OPERATING INSTRUCTIONS

Flexi Soft gateways in Safety Designer

Configuration software

SICK
Sensor Intelligence.

Described product

Flexi Soft Gateways in the Safety Designer

Configuration software

Manufacturer

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

Legal information

This work is protected by copyright. Any rights derived from the copyright shall be reserved for SICK AG. Reproduction of this document or parts of this document is only permissible within the limits of the legal determination of Copyright Law. Any modification, abridgment or translation of this document is prohibited without the express written permission of SICK AG.

The trademarks stated in this document are the property of their respective owner.

© SICK AG. All rights reserved.

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 Safety 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.

Document identification

Document part number:

- This document: 8018172
- Available language versions of this document: 8018170

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 Safety Designer configuration software. They are intended for the following target groups: project developers (planners, developers, designers), installers, electricians, programmers, operators, and maintenance personnel.

NOTE

These operating instructions do not cover how to configure the gateways using the Flexi Soft Designer configuration software. The operating instructions titled “Flexi Soft Gateways in the Flexi Soft Designer Configuration Software” (SICK part number 8012483) contain information on this.

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
- EDS file for the FX0-GCAN for CANopen
- ESI file for the FX0-GETC for EtherCAT
- EDS file for the FX3-GEPR for EFI-pro

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 V4.00.0; the Safety Designer configuration software must be at least V1.6.x.

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 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”.

Exception: The FX3-GEPR EFI-pro gateway

The FX3-GEPR EFI-pro gateway can act as both an originator (master) and a target (slave) in the EFI-pro network. Since it is used as a master in many application cases, the term “input data” in this case describes the data that the device receives from the network, and the term “output data” refers to data sent by the device to the network.

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 Safety Designer configuration software. Safety Designer can be used to establish connections to a Flexi Soft system via the following interfaces:

- RS-232 interface of the main module
- USB interface of an FX3-GEPR
- Ethernet interface of an FX3-GEPR

For detailed information on configuring the Flexi Soft system, please refer to the operating instructions titled “Flexi Soft in the Safety Designer Configuration Software” (SICK part number 8013926).

3.1 Device variants

Table 2: Device variants and their main features

Gateway	Network type	Ethernet TCP/IP socket interface	TCP/IP configuration interface [1]
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 [2]
FX0-GCAN	CANopen slave	–	–
FX3-GEPR [3]	EFI-pro master and slave operation EtherNet/IP™ CIP Safety™ master EtherNet/IP™ slave	–	TCP port 2122 UDP port 30718

1) The TCP/IP configuration interface of the gateway FX0-GETC can only be used with the Flexi Soft Designer configuration software. The TCP/IP configuration interface is available in Safety Designer for FX0-GENT, FX0-GPNT and FX0-GMOD from version 1.7.0.

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

3) To configure the FX3-GEPR, version V1.6.x or higher of the Safety Designer configuration software is required. Flexi Soft systems that include an FX3-GEPR cannot be configured with the Flexi Soft Designer configuration software.

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

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 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
- The FX0-GMOD Modbus® TCP gateway
- The FX0-GPNT PROFINET IO gateway
- The 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.

NOTE

- Safety Designer supports Flexi Soft EtherNet/IP™ gateways with firmware ≥ V3.00.0.
- You will find the firmware version on the device type label.

4.1.1.1 Basic configuration – assigning the device name and IP address

► Start Safety Designer and load the hardware configuration, including the EtherNet/IP™ gateway.
► Under **Configuration**, click on the FX0-GENT to open the dialog box for the gateway configuration.
► In the navigation area, click on **Gateway configuration**. The following dialog box is displayed:

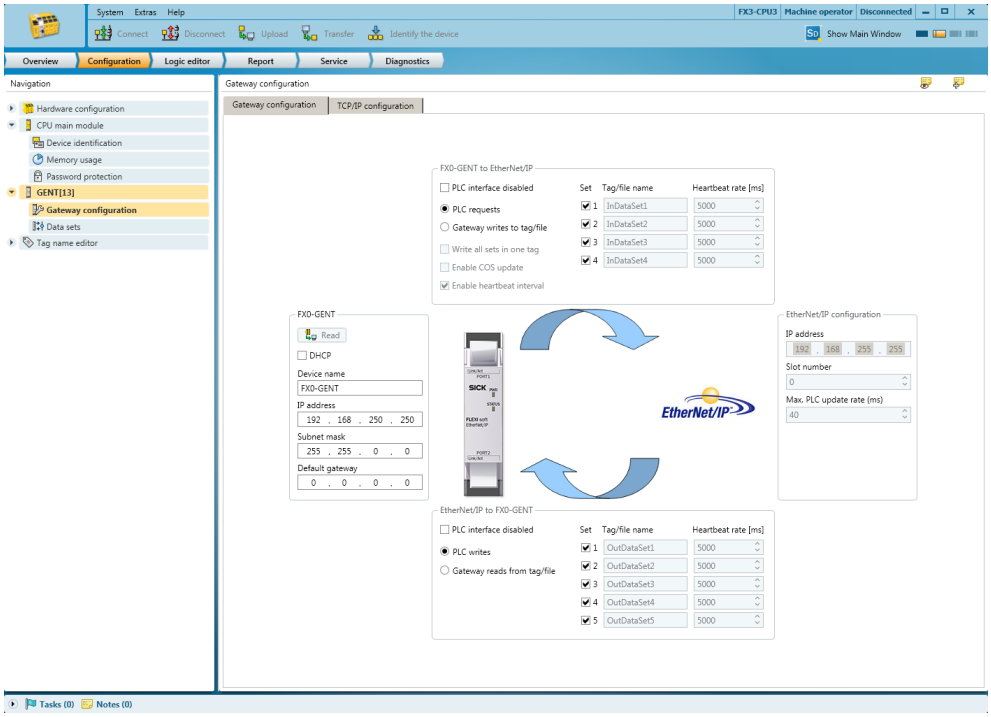

Figure 1: Configuration window for the EtherNet/IP™ gateway

► 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 device** to transfer the settings to the Flexi Soft system.

4.1.1.2 EtherNet/IP™ Class 1 communication – implicit messaging

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

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.

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 4: 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

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

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 5: 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 6: 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

Assembly instance	Byte index	Length [bytes]	Output data set	Description	Valid lengths for write access [bytes]
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 7: 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 8: 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

Assembly instance	Description	Data type	Data values	Access type	Corresponding attributes of the Full Data Set Transfer object
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
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 9: 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 10: 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.

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

The FX0-GENT supports a total number of 8 connections.

Configuration

The following table describes the configuration process depending on the transmission type:

Table 11: Configuration guideline – gateway as master

Gateway is a master (gateway writes to tag/file and/or gateway reads from tag/file)	
Required 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 the PLC or is to be read from it.	–
Determine the location in the PLC memory to which the selected data is to be written: Enter tag name. Example: InDataSet1 Determine the location in the PLC memory from which the selected data is to be read: Enter tag name. Example: OutDataSet1	Enter this exact tag name in the PLC program. Example: InDataSet1 INT[25] OutDataSet1 INT[5] The data type must be INT.
Select how often this data is to 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 12: Configuration guideline – gateway as slave

Gateway is a slave (PLC requests and/or PLC writes)	
Required 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.
–	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 PLC program to read data from the gateway or to write it to the gateway.
–	Program the trigger for sending the explicit messages.

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

With this transmission type, the FX0-GENT is the master and writes all activated data sets to the specified memory areas of the PLC. The PLC programmer only needs to define a tag name in the control which corresponds to the tag name in the gateway configuration.

Configuring the gateway as a master

- ► Start Safety Designer and load the hardware configuration, including the EtherNet/IP™ gateway.
- ► Under **Configuration**, click on the FX0-GENT to open the dialog box for the gateway configuration.
- ► In the navigation area, click on **Gateway configuration**. The following dialog box is displayed:

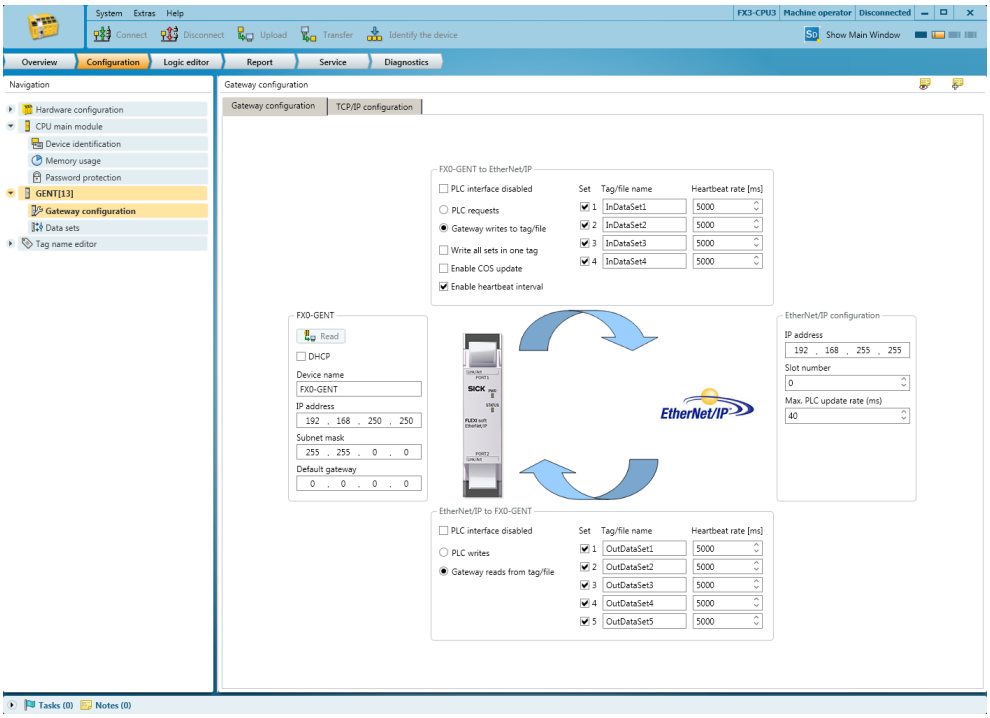

Figure 2: Configuring the EtherNet/IP™ gateway as a master

- ► In the **Gateway configuration** dialog box, activate the **Gateway writes to tag/file** option for the Flexi Soft to network direction and the **Gateway reads from tag/file** option 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 PLC memory bank, in which the selected data is to be written or the location from which it is to be 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. In this case, the tag defined for data set 1 is used.
- ► 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 with the set **heartbeat rate**.
  - Both options can be selected 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** to update the selected data sets at the specified interval.

- Specify the location in the EtherNet/IP™ in which the selected data is to be written or the location from which it is to be read. Under EtherNet/IP™ configuration, enter the **IP address** and the **slot number** of the PLC.

NOTE

The configuration is incorrect if the PLC IP address is zero and **Gateway writes to tag/file** is checked 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 highest or leftmost byte of the integer.

- Click on **Connect** to switch to online mode.
- Click on **Transfer to device** to transfer the settings 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:

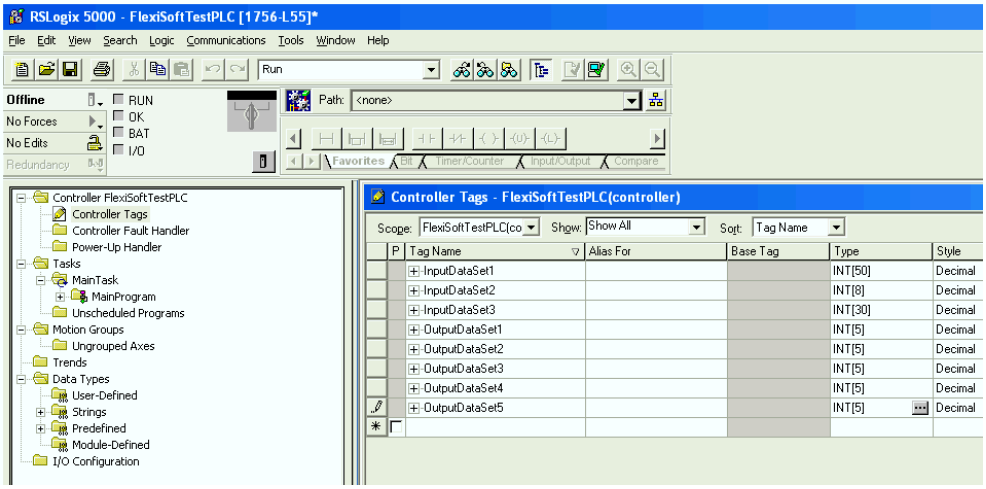

Figure 3: Example of tag names in a PLC program

NOTE

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

Transmission type 2: Polling mode – the PLC requests the data or writes the 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 Safety Designer and load the hardware configuration, including the EtherNet/IP™ gateway.
- ► Under **Configuration**, click on the FX0-GENT to open the dialog box for the gateway configuration.
- ► In the navigation area, click on **Gateway configuration**. The following dialog box is displayed:

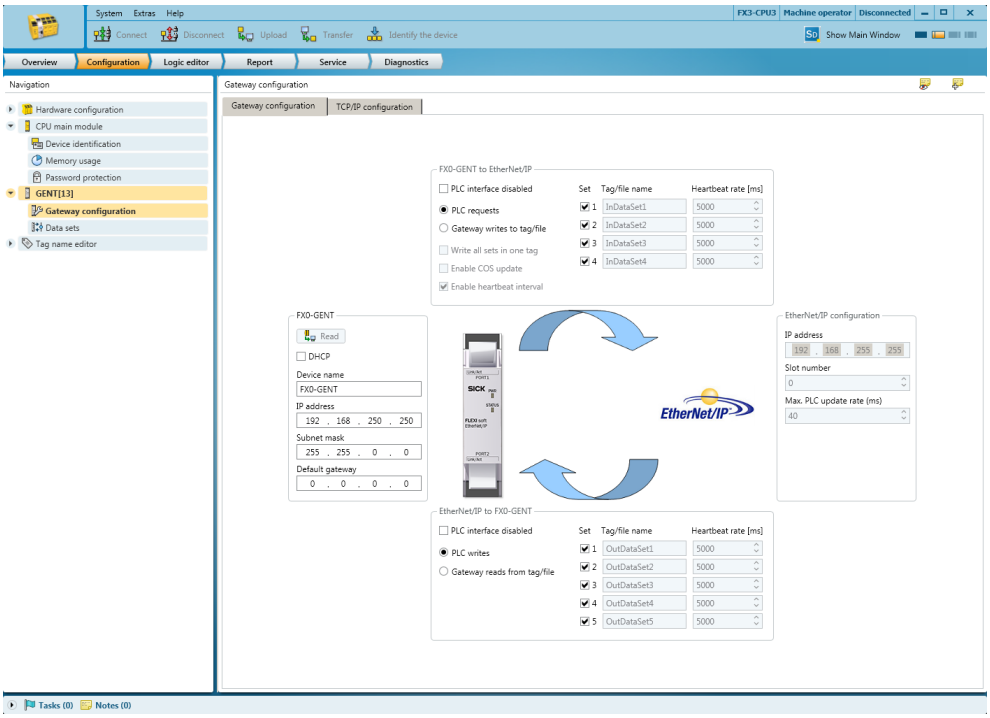

Figure 4: Configuring the EtherNet/IP™ gateway as a slave

- ► In the **Gateway configuration** dialog box, activate the **PLC requests** option for the Flexi Soft to network transmission direction and the **PLC writes** option for the network to Flexi Soft transmission 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 device** 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 13: 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 14: 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 15: 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 16: 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 17: 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 18: 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 19: 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 20: 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 21: 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 22: 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]

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. Under **Configuration**, click on the FX0-GMOD to open the dialog box for the gateway configuration.
3. Click on **Gateway configuration**. The following dialog box is displayed:

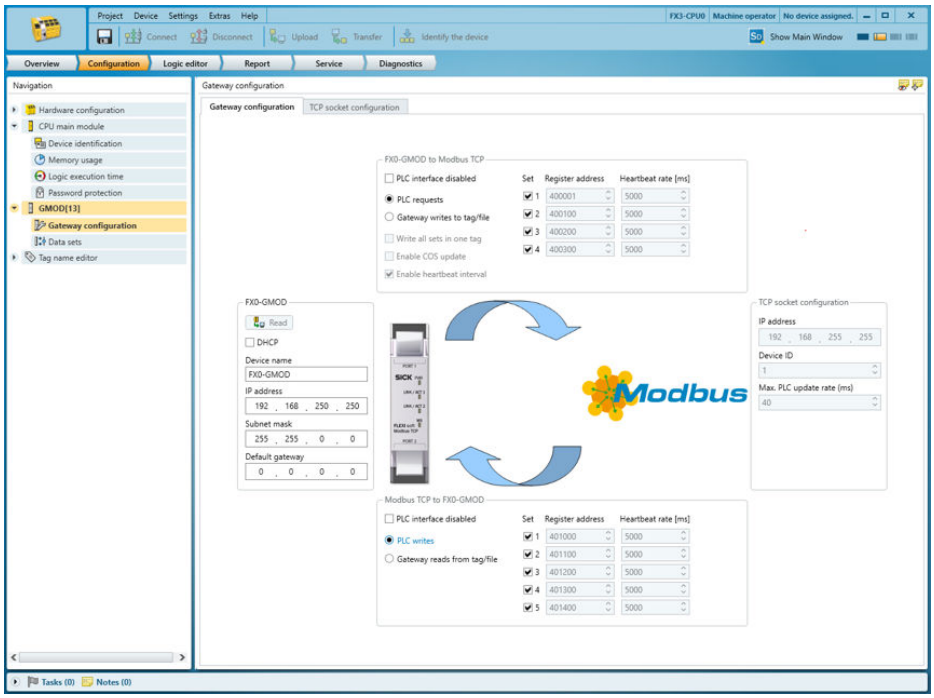

Figure 5: Configuration window for the Modbus® TCP gateway

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
- Data can also be read 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 23: 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 24: 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.	–
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 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. Under **Configuration**, click on the FX0-GMOD to open the dialog box for the gateway configuration.
3. Click on **Gateway configuration**. The following dialog box is displayed:

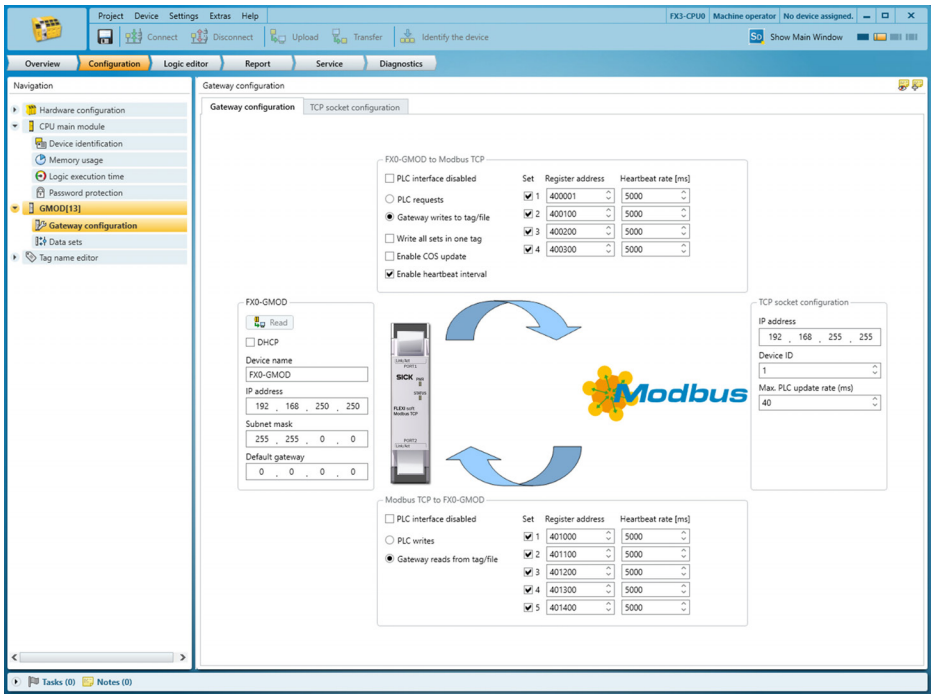

Figure 6: Configuration of the Modbus® TCP gateway as a client

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 109).
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 **Write all sets in one tag** option if all data sets are to be written to the PLC memory in a single address. In this case, the register address defined for data set 1 is used.
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.
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 25: 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 26: 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
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. Under **Configuration**, click on the FX0-GMOD to open the dialog box for the gateway configuration.
3. Click on **Gateway configuration**. The following dialog box is displayed:

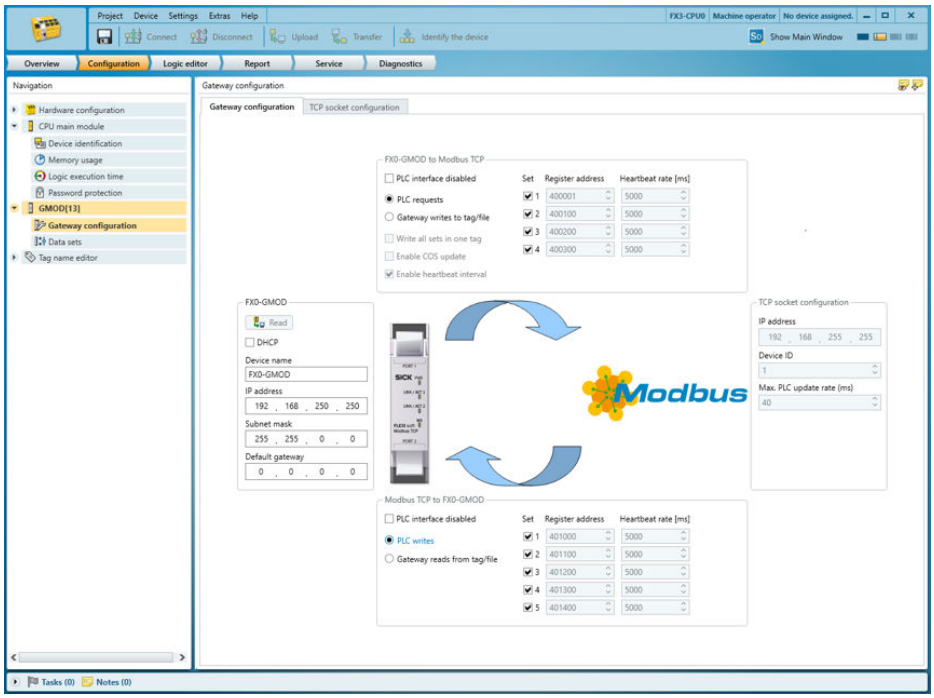

Figure 7: Configuration of the Modbus® TCP gateway as a server

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 109).
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 27: Modbus commands

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

Error messages

Table 28: 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 The FX0-GPNT PROFINET IO gateway

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

NOTE

- Safety Designer supports Flexi Soft PROFINET IO gateways with firmware ≥ V3.00.0.
- You will find the firmware version on the device type label.

The FX0-GPNT supports

- PROFINET IO, conformance class B
- Network load class 1
- MRP client
- LLDP
- SNMP
- MIB-II
- Fast integrated switching
- Auto MDI
- Auto-negotiation
- Cyclical I/O communication

4.1.3.1 Basic configuration – assigning the device name and IP address

It is possible to configure and diagnose the FX0-GPNT with both the Flexi Soft configuration software and 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 the device name using Safety Designer

- ► Start Safety Designer and load the hardware configuration, including the PROFINET IO gateway. Make sure that the project is offline.
- ► Under **Configuration**, click on the FX0-GPNT to open the dialog box for the gateway configuration.
- ► In the navigation area, click on **Gateway configuration**. The following dialog box is displayed:

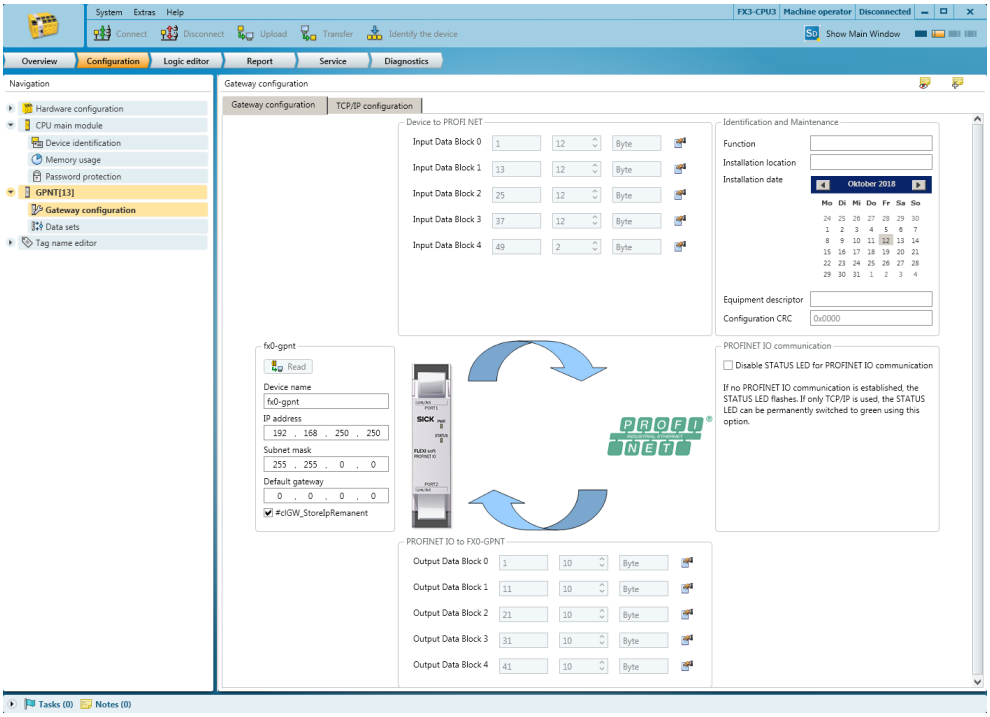

Figure 8: Configuration window for the PROFINET IO gateway

- ► Enter the device name in the **Device name** field (maximum length 255 characters).
- ► Click on **Connect** to switch to online mode.
- ► Click on **Transfer to device** to transfer the settings 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, the SIEMENS SIMATIC STEP 7 uses the same IP address for the default gateway as for the FX0-GPNT.

Assigning the IP address using Safety Designer

The IP address is usually assigned by the PROFINET IO controller (e.g., PLC). However, the FX0-GPNT also allows you to configure the entire Flexi Soft system via Ethernet TCP/IP. In this case, it may be necessary to assign an IP address to the gateway before setting up the PROFINET IO network. You can also do this via the configuration page (see figure 8, page 36).

Saving the connection data remanently

The connection data can be saved remanently in FX0-GPNT. The gateway will be reset to the most recent, remanently saved connection data each time it is restarted.

Saving the connection data remanently

- Select the **Save remanently** checkbox.
- Click on **Connect** to switch to online mode.
- Click on **Transfer to device** to transfer the configuration to the Flexi Soft system.

Deactivating the STATUS LED for PROFINET IO communication

You can 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).

► Under **Configuration**, click on the FX0-GPNT to open the dialog box for the gateway configuration.
► In the navigation area, click on **Gateway configuration**. You can find the following area in the configuration window:

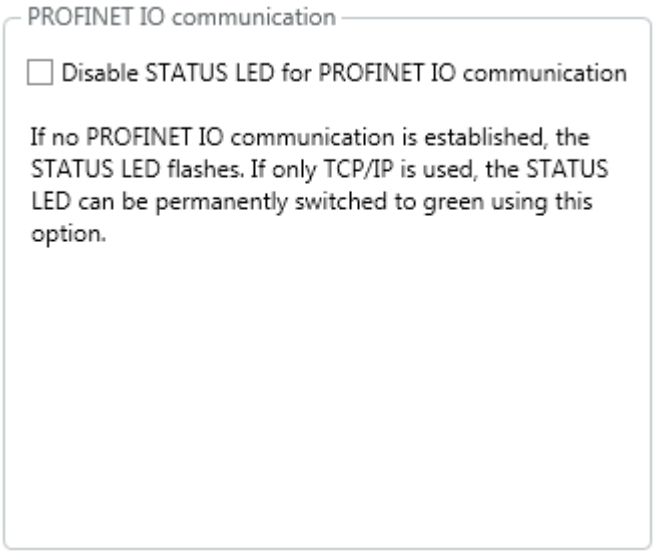

Figure 9: Deactivating the FX0-GPNT STATUS LED

► 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.

4.1.3.2 PROFINET IO configuration – selecting the data

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 10, page 39.

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 29: 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

For information on the content of the process image, see "Data transferred to the network (network input data sets)", page 109.

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.

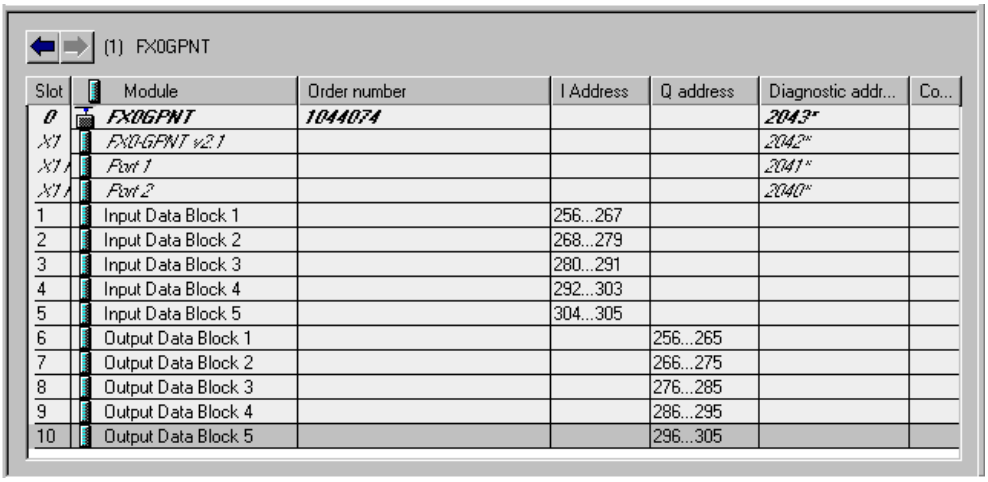

Figure 10: Projecting the FX0-GPNT

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 30: 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 31: 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>

For an interpretation of the module status bits in data set 3: see "Error and status information of the modules", page 113

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 32: 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

The following I&M information can be displayed and edited using Safety 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 in the **Identification and maintenance** area.
► Click on **Connect** to switch to online mode.
► Click on **Transfer to device** 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 33: 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 The 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:

- 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. More detailed 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 Safety Designer and load the hardware configuration, including the FX0-GETC or create a new Flexi Soft system with an FX0-GETC in Safety Designer.

Configuring the gateway in the Flexi Soft system

- ► Under **Configuration**, click on the FX0-GETC to open the window for the gateway configuration.
- ► In the navigation area, click on **Gateway configuration**.
- ► In the **Flexi Soft to network** 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 "Configuring the process image", page 120).
- ► In the **Network 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 **Device configuration** area, you can change the **device name** of the gateway in the Flexi Soft system. The preset name of the gateway is “GETC”.

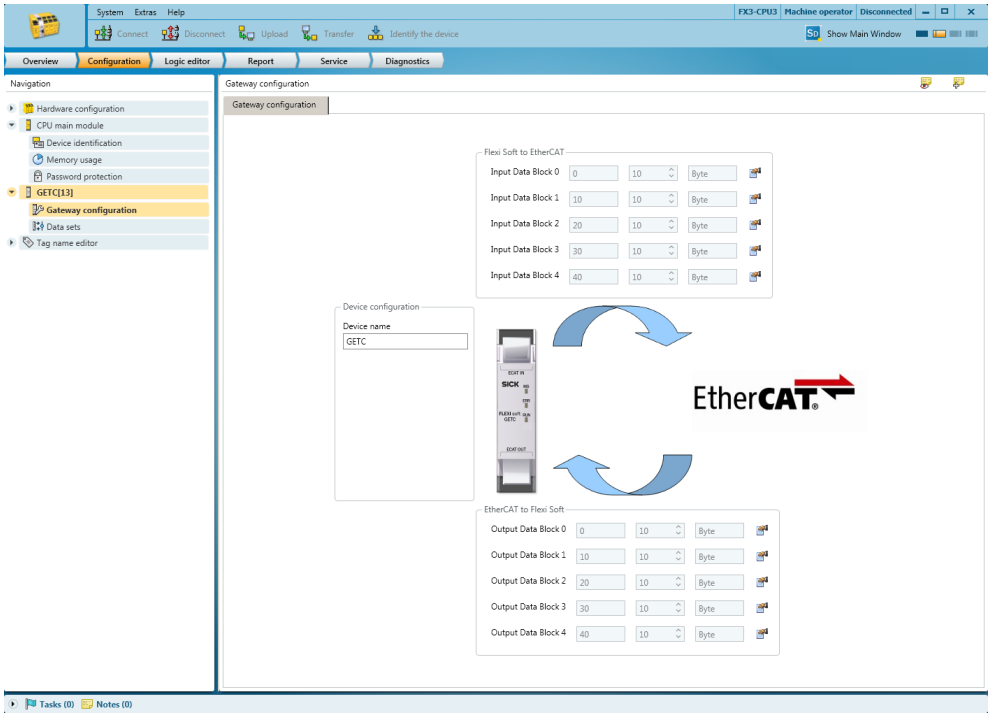

Figure 11: Gateway configuration dialog box for the FX0-GETC

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.
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.

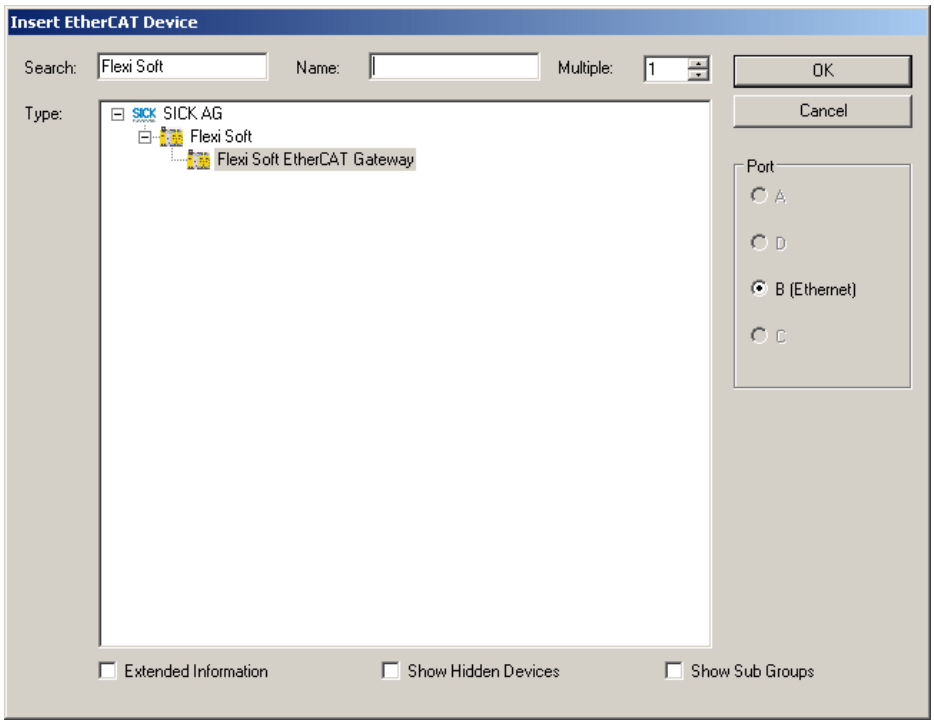

Figure 12: Example for integrating FX0-GETC into an EtherCAT network

► 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 34: 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.

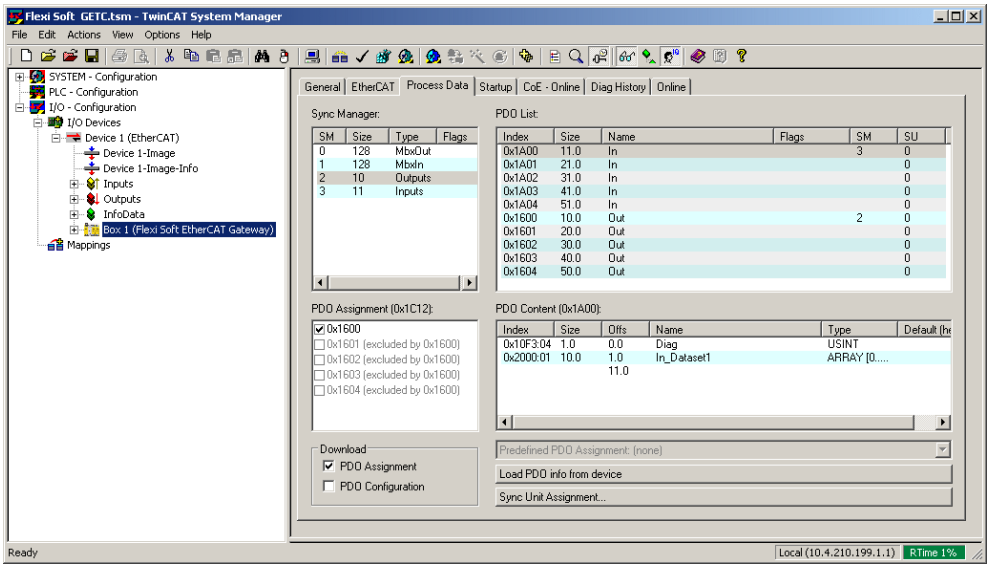

Figure 13: PDO configuration in the EtherCAT network configuration tool

4.1.4.3 Input data – Flexi Soft to network

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 35: 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

For detailed information on the content of the process image, see "Data transferred to the network (network input data sets)", page 109.

For information on how the process image is configured, see "Configuring the process image", page 120 and the operating instructions titled “Flexi Soft in the Safety Designer Configuration Software” (SICK part number 8013926).

NOTE

The process data can also be read using the CoE objects 2000h and 2001h (see "CAN application layer over EtherCAT (CoE)", page 55). We recommend simple access via SDO for diagnostic purposes. In normal operation, the quicker PDO communication should be used.

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

Safety 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 in 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 to individual bits in the EtherCAT PDOs.

Exporting tag names

- ► The tag names for the input and output data sets must be exported separately. You can export the tag names of the input data sets by opening the configuration page for **Flexi Soft to network**. You can export the tag names of the output data sets by opening the configuration page for **Network 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

- For each data set used, the export function creates a ten-byte structure with the byte or module name 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 address of the data sets in the TwinCAT PLC process image can be changed in the gateway configuration menu.

Changing the start addresses of the data sets

- ► Start Safety Designer and load the hardware configuration, including the EtherCAT gateway. Make sure that the project is offline.
- ► Under **Configuration**, click on the FX0-GETC to open the dialog box for the gateway configuration.
- ► Click on **Gateway configuration**. The following dialog box is displayed:

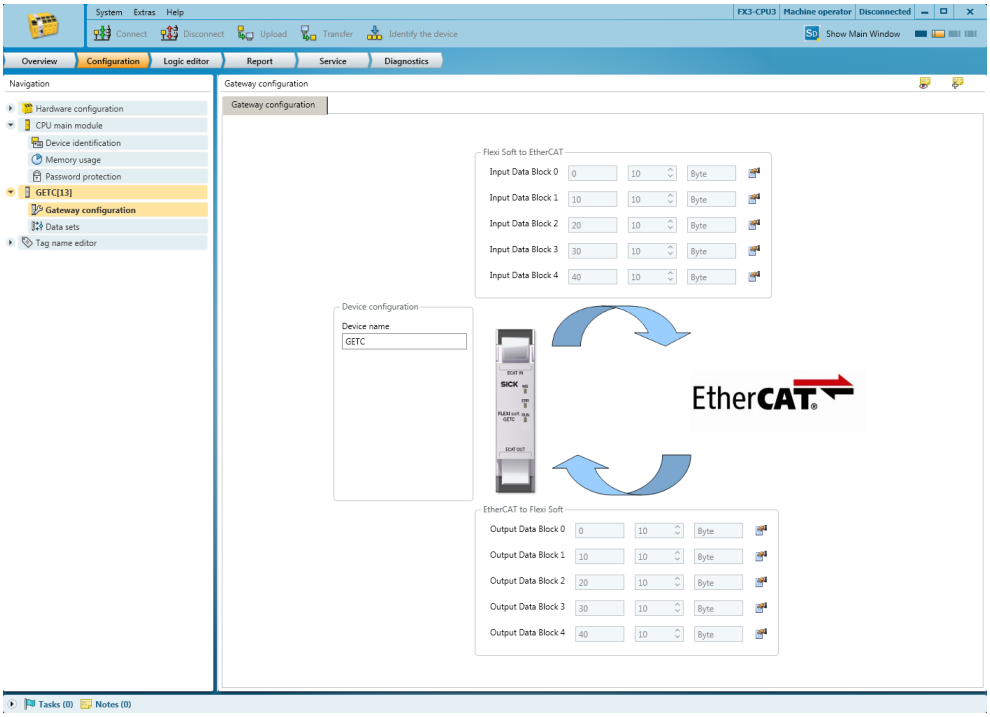

Figure 14: Configuration window for the EtherCAT gateway

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

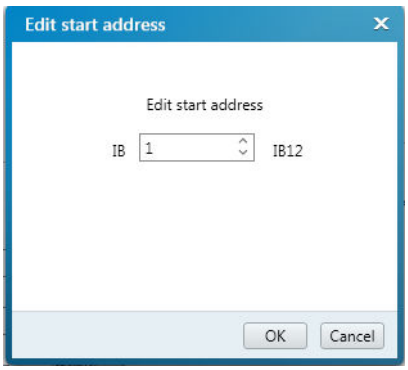

Figure 15: Changing the data set start address

► 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.

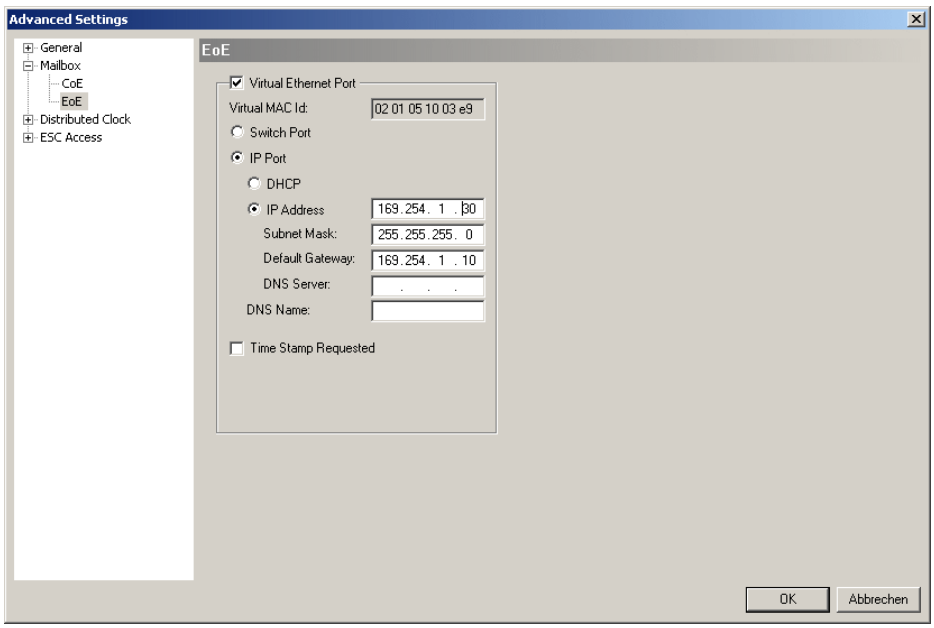

Figure 16: Activating EoE for the FX0-GETC in TwinCAT

► 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 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.

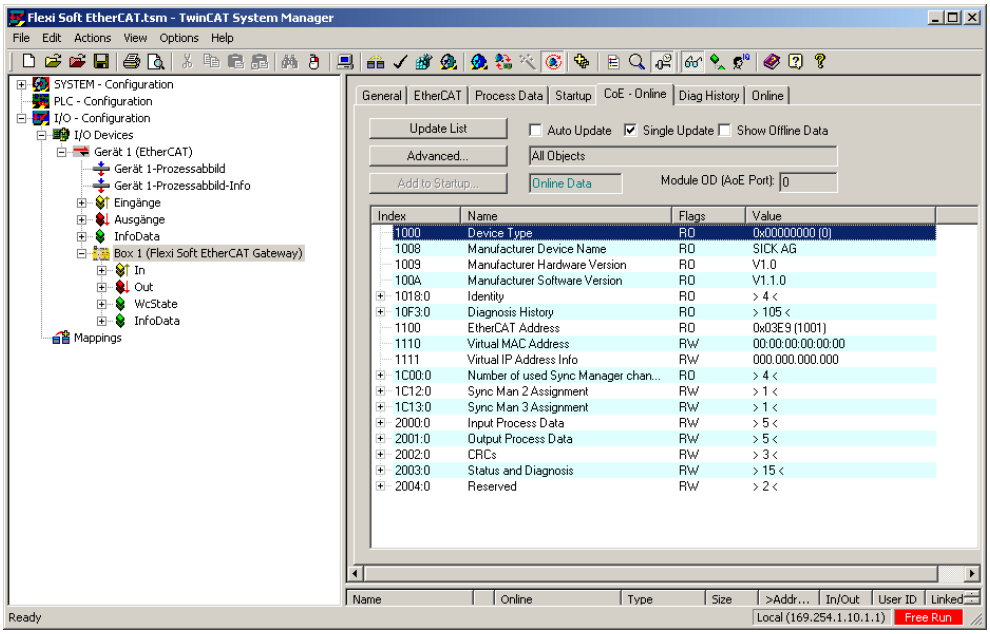

Figure 17: CoE object catalog of the FX0-GETC in TwinCAT

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 36: 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 37: 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 checksums (for a description, see "Configuration checksums", page 113).

Table 38: 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 for the Flexi Soft system. Each Flexi Soft module has 32 status bits, each of which represents a possible error message of the module. The meaning of each bit depends on the type of module (see "Error and status information of the modules", page 113).

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

Table 39: 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 40: 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 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.

For each TCP/IP socket, six connections are possible – in other words, a total of 24 connections per gateway.

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.

4.1.5.1 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. Under **Configuration**, click on the desired gateway to open the dialog box for the gateway configuration.
3. Click on **Gateway configuration**.
4. Click on **TCP/IP configuration**. The following dialog box is displayed:

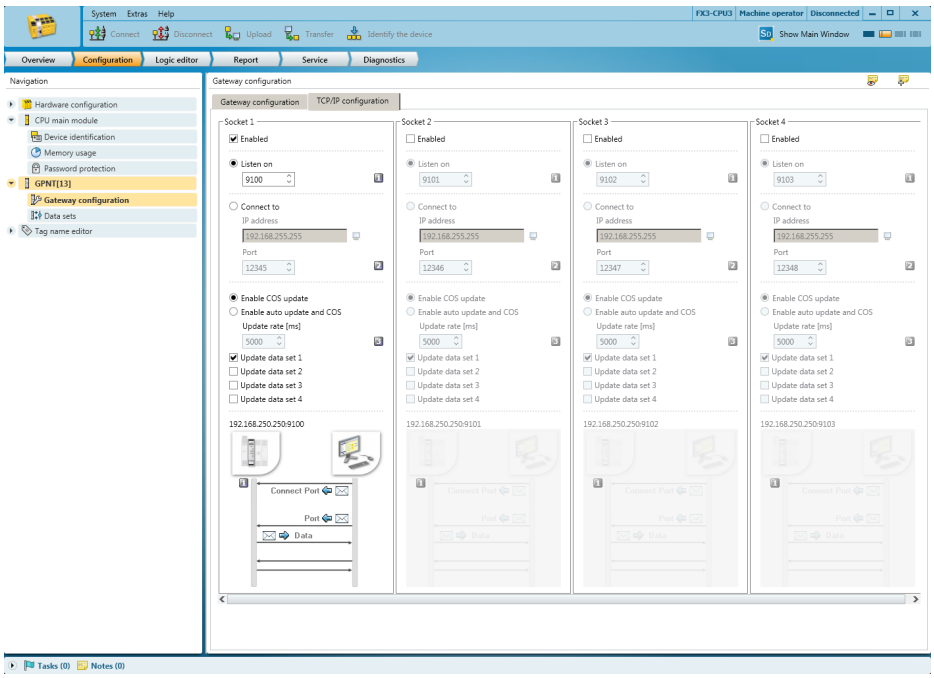

Figure 18: TCP/IP configuration dialog box

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 109

4.1.5.2 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 41: 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 42: Error response

Parameter	Length	Description
Command	WORD	Bit 15 of the received command is set (i.e. the command 00F2h becomes 80F2h).
Following data	Length depends on command	Unchanged. Returned as received

4.1.5.3 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. Under **Configuration**, click on the desired gateway to open the dialog box for the gateway configuration.
3. Click on **Gateway configuration**.
4. Click on **TCP/IP configuration**. The following dialog box is displayed:

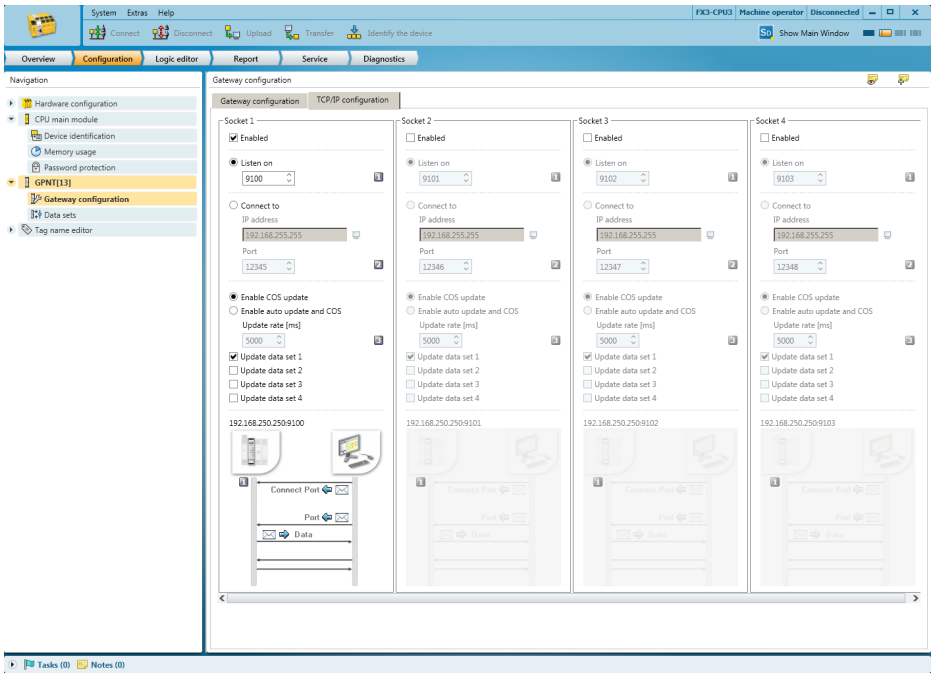

Figure 19: TCP/IP configuration for polling mode (application requests)

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

4.1.5.3.1 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 43: 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 44: 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

Parameter	Length	Value
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

4.1.5.3.2 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 45: 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
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 46: 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

4.1.5.4 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 47: 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
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 48: Response to configuring auto update mode

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

4.1.5.5 Configuring auto update mode

Procedure

1. Start the configuration software and load the hardware configuration, including the gateway.
2. Under **Configuration**, click on the desired gateway to open the dialog box for the gateway configuration.
3. Click on **Gateway configuration**.
4. Click on **TCP/IP configuration**. The following dialog box is displayed:

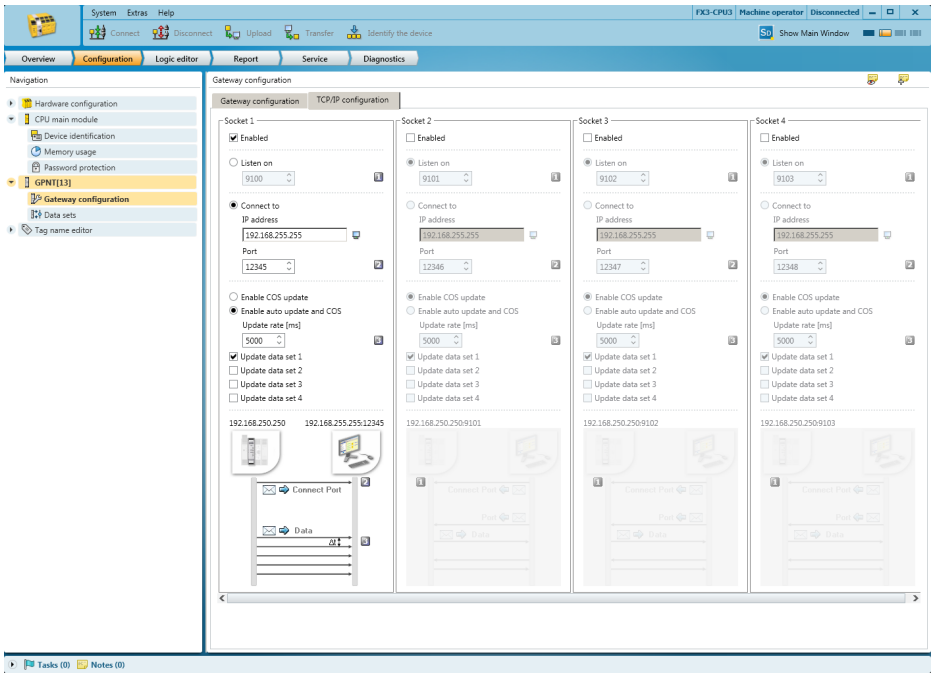

Figure 20: TCP/IP configuration for auto update

5. Select the **Connect to** option.
6. Enter the IP address and the port to which the gateway is to write.
7. Select update mode.
8. If you wish to use auto update and COS, enter the update rate in milliseconds.
9. Select which data is to be updated: Select the **Update data set n** checkbox.

4.1.5.5.1 Auto update mode message

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

Table 49: 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

4.1.6 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 50: 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
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		

Byte values [Hex]	Part of the message	Significance
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)
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 gateway:

- The FX0-GCAN CANopen gateway

4.2.1 FX0-GCAN CANopen gateway

4.2.1.1 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 Safety Designer and load the hardware configuration, including the CANopen gateway. Make sure that the project is offline.
3. Under **Configuration**, click on the FX0-GCAN to open the dialog box for the gateway configuration.
4. In the navigation area, click on **Gateway configuration**. The following dialog box is displayed:

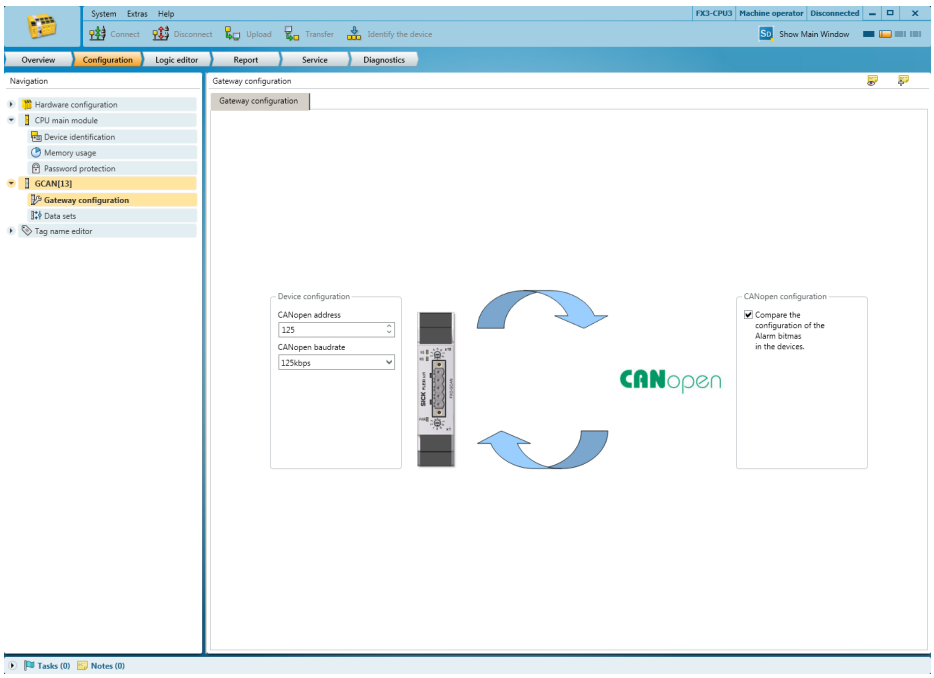

Figure 21: Configuration window for the CANopen gateway

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

4.2.1.2 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.

4.2.1.2.1 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 51: Structure of the CAN identifier

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

Table 52: 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.

4.2.1.2.2 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 53: Network management for an NMT device with address N

CAN-ID	DLC	Data							
00h	2	OP	N						

Broadcast service for all NMT devices

Table 54: 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 55: Example NMT object for resetting all communication

CAN-ID	DLC	Data							
00h	2	82h	0						

4.2.1.2.3 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 56: 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 83

4.2.1.2.4 Emergency

Emergency

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

Table 57: 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 58: 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 59: 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 60: 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 61: 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 62: 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 63: 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 64: 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 65: 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							

4.2.1.2.5 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 66: 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 67: Device response

CAN-ID	DLC	Data							
700h + N	1	Byte1							

Table 68: 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 78, page 77), 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 78, page 77), then at least one node guarding message must be received within the configured heartbeat consumer time (typically from an NMT controller).

4.2.1.2.6 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 69: 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 70: 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 54, page 70).

RxPDO1 ... 4

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

Table 71: 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.

4.2.1.2.7 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 72: 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 73: 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 74: 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 75: 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 76: Read SDO (example)

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

The server responds:

Table 77: 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.

4.2.1.2.8 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 78: 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 71.
- 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 85).

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 85).

SDO 1018h: Identification

This SDO contains basic information about the gateway.

Table 79: 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 80: 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 71. 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 81: 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 71, page 76).

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 82: 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 83: 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 69, page 75 and see table 70, page 75).

SDO 3100h: Module status bits

SDO 3100h contains the module status bits of the Flexi Soft system (see "Emergency", page 71). Active bits have the value 0.

Table 84: 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 85: 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 71, page 76). The mapping is as follows:

Table 86: 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 70, page 75). The mapping is as follows:

Table 87: 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.

4.2.1.3 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 88: 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**		

4.2.1.4 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 71) 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 71).

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

Table 89: 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 90, page 87).

Overview of error objects

Table 90: 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 71
Gateway detects outgoing error, other errors present	–	2	FF02h 81h M1 = Module index M2 ... M5 = Module diagnostic data	M2, M3, M4, M5	"Emergency", page 71
Gateway detects incoming error, other errors present	–	3	FF03h 81h M1 = Module index M2 ... M5 = Module diagnostic data	M2, M3, M4, M5	"Emergency", page 71
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 91: 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 71)

Reading the current error from SDO 3100h

PLC requests:

Table 92: 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 84, page 82)

Response of the gateway:

Table 93: 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 84, page 82)

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 94: 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 95: 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 96: 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 71)

Reading the current error from SDO 3100h

PLC requests:

Table 97: 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 84, page 82)

Response of the gateway:

Table 98: 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 84, page 82)

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 71)

Reading the error from the error history in SDO 1003h

PLC requests:

Table 99: 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 100: 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

4.3 Safe gateways

4.3.1 The FX3-GEPR EFI-pro gateway

4.3.1.1 Basic configuration

CAUTION

Malfunction due to inconsistent 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.

- ► Before configuring a new FX3-GEPR, reset the device so that all previous configuration data in the device memory is deleted. This also applies when replacing a device. (ODVA SRS52)

CAUTION

Malfunction due to faulty configuration or communication

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.

- ► Before commissioning a Flexi Soft system with one or several FX3-GEPRs, ensure that all gateways are configured correctly.
- ► Test the safety-related communication between the gateways to ensure that the connections function as intended. (ODVA FRS103, SRS92)

For further information on commissioning a Flexi Soft system and the necessary checks, see the operating instructions titled “Flexi Soft Modular Safety Controller Hardware” (SICK part number 8012999) and the operating instructions titled “Flexi Soft in the Safety Designer Configuration Software” (SICK part number 8013926).

CAUTION

Malfunction due to faulty replacement devices

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.

► After replacing a defective FX3-GEPR, ensure that the replacement device is configured correctly and functions without any problems. (ODVA FRS112)

For further information on replacing defective devices, see the operating instructions titled “Flexi Soft in the Safety Designer Configuration Software” (SICK part number 8013926).

NOTE

- After configuring an originator with connection data and/or a target, you must transfer the configuration to the target to check and verify it. Otherwise, SCIDs (Safety Configuration Identifiers) cannot be confirmed by the target. (ODVA SRS44)
- After the transfer, carry out a visual check to see whether all configuration data has been transferred correctly. (ODVA SRS204)

NOTE

The FX3-GEPR EFI-pro gateway can act as both an originator (master) and a target (slave) in the EFI-pro network. Since it is used as a master in many application cases, the term “input data” in this case describes the data that the device receives from the network, and the term “output data” refers to data sent by the device to the network.

Setting up a Flexi Soft system with an FX3-GEPR

This section describes how to create a Flexi Soft system with an FX3-GEPR for the first time in Safety Designer.

1. Start Safety Designer and open a new project.
2. Add a Flexi Soft safety controller from the device catalog. A main module with the firmware version 4.00.0 (Step 4.xx) or above is required for the FX3-GEPR to be used.
3. Click on the device tile to open the device window.
4. Click on **Configuration**.
5. Click on **Hardware configuration** in the navigation tree.
6. Add an FX3-GEPR from the **catalog** to the Flexi Soft system.

✓ The FX3-GEPR appears in the navigation tree and can be configured further there.

✓ Only one FX3-GEPR can be used per station.

NOTE

To display the current settings for the FX3-GEPR, select the FX3-GEPR under Configuration in the navigation tree.

- Under **Device identification**, you can edit the tag name of the FX3-GEPR and the Safety Network Number (SNN).
- Under **Network settings**, you can edit the IP settings (IP address, subnet, and router).
- The device's MAC ID is printed on the type label of the gateway. It is also displayed in the Safety Designer configuration dialog for the FX3-GEPR under **BOM info** as well as in the report.

(ODVA SRS53)

Also see the operating instructions titled "Flexi Soft in the Safety Designer Configuration Software" (SICK part number 8013926).

Connecting the computer to the FX3-GEPR

Establishing a connection with the FX3-GEPR

► Create a Flexi Soft system with a main module, any expansion modules, and an FX3-GEPR.

► Connect the computer to one of the interfaces of the Flexi Soft system.
The following interfaces are available:
  - Serial RS-232 interface of the main module
  - USB interface of the FX3-GEPR
  - Ethernet interface of the FX3-GEPR

NOTE

The following functions are not possible if the Flexi Soft system is connected via the serial interface of the main module (RS-232):

- EFI-pro diagnostics (connection status and data)
- Detailed diagnostics of the FX3-GEPR. The diagnostics function of the main module only provides the standard diagnostics messages as in the case of the other gateways.

For this reason, it is a good idea to always connect Flexi Soft systems containing an FX3-GEPR via the USB or Ethernet connection of the FX3-GEPR.

► Switch on the Flexi Soft system.

► Start the Safety Designer configuration software that has been installed on the computer.

► On the Safety Designer start screen, click on **Search for devices**. A new project is opened and the available interfaces of the computer are scanned.

► Click on **Configuration**. A list of the devices found is displayed on the **Device search** tab.

NOTE

- If, in the Safety Designer online settings, the **Search for devices** window is deactivated, click on the icon for **Search once for devices** on the **Device search** tab to start the search.
- On the **Device search** tab, only the main module of the found Flexi Soft systems is displayed, even if the connection was established via a gateway connected to the main module.

► Add the Flexi Soft main module from the **Device search** tab to the project (double-click or drag and drop). The device is displayed in the device overview as a device tile and is marked as **Accessible**.

NOTE
If several similar devices are displayed in the device catalog, the **Identify device through the flashing of its indicating element** function can be used for assignment.

► Click the **Connect** button on the toolbar or in the device tile. Safety Designer establishes a connection with the Flexi Soft system and then loads the hardware configuration of the Flexi Soft system (main module, gateway, and expansion modules).

Assigning a Safety Network Number (SNN)

This section describes how to enter or generate the Safety Network Number using Safety Designer.

CAUTION
Malfunction due to incorrect safety network number (SNN)
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.

► Use the same uniform SNN across the system for all devices in an EFI-pro network. (ODVA FRS154)
► In safety-related applications, do not use the function for automated assignment of SNNs available in some originators. (ODVA SRS193)

1. Open the device window. The SNN is displayed under **Configuration**, **GEPR**, **Device identification**.
2. In the **SNN** field, enter the desired Safety Network Number or click on **Take from project settings** to apply the standard SNN generated by Safety Designer (ODVA SRS193).
3. Use the **Generate** button to generate a Safety Network Number by entering a decimal number or based on the current system time.
4. To change the Safety Network Number in the device, transfer the configuration to the device.

NOTE
For this action, you must log in as an authorized client. The login dialog will open where applicable.

5. If a Flexi Soft system is connected, you can also load the Safety Network Number for the device and apply it to the project.

Assign IP address

This section describes how to enter the IP address using Safety Designer.

1. Open the device window. The network configuration in the project file and – if there is a connection – the network configuration in the device are displayed under **Configuration**, **GEPR**, **Network settings**.
2. Enter the desired **IP address** for the gateway, the **subnet**, and the IP address for the **router**. The IP settings predefined for the entire project can be applied for the **subnet mask** and the IP address of the **router**.
3. Click on **Transfer** to transfer the settings to the gateway. The computer must be connected to the Flexi Soft system for this.

NOTE
For this action, you must log in as an authorized client. The login dialog will open where applicable.

4. If a Flexi Soft system is connected, you can also load the IP settings for the device and apply them to the project.

4.3.1.2 FX3-GEPR EFI-pro communication

Description

The FX3-GEPR works as an interface between two EFI-pro systems or an EFI-pro system and a higher-level application. It allows SICK-specific network management and EFI-pro functions to be applied in both directions:

- Secure exchange of data
- SOPAS scanning function to identify and address devices using SICK-specific services

The FX3-GEPR can act as both an originator (master) and a target (slave) in the EFI-pro network. As an originator it supports safety connections, and as a target it supports both standard and safety connections.

NOTE
For detailed information on EFI-pro, see the “EFI-pro system” technical information (SICK part number 8022340).

Setting up EFI-pro communication between two FX3-GEPRs

This section describes how an EFI-pro data connection between two FX3-GEPRs is set up.

Setting up an automated connection in the connection view

1. Start Safety Designer, open a new project, and create two Flexi Soft systems, each with a main module and an FX3-GEPR, as described in the Basic configuration section.
2. Connect the two Flexi Soft systems together in Safety Designer under **Connections**, as described in the Configuring EFI-pro connections in Safety Designer section.

✓ A dynamic safety assembly is established in both the master and slave for input data and for output data, and connected with the corresponding assembly of the other device.

✓ The automatically established assemblies are 4 bytes in size and are 8-bit or Boolean types.

NOTE

- This is the recommended procedure.
- The connection view cannot be used to create any multi-connections or any assemblies that are of a different size or that have a different data type. This is only possible manually in the device window.

Manually setting up a connection in the device window

1. Start Safety Designer, open a new project, and create two Flexi Soft systems, each with a main module and an FX3-GEPR, as described in the Basic configuration section.
2. In the relevant device window under **Configuration**, **GEPR**, **Device identification**, assign a **Safety Network Number (SNN)** that is consistent across the entire project to each of the two gateways.
3. In the relevant device window under **Configuration**, **GEPR**, **Network settings**, assign an individual **IP address** to each of the two gateways.
4. In the device window of one of the two Flexi Soft systems under **Configuration**, **GEPR**, **Connection creation**, double-click on one of the two plus symbols to open the **assembly editor** and create a new assembly.

- You can create assemblies for received data (output by network) using the left-hand plus symbol.
- You can create assemblies for data to be sent (input into the network) using the right-hand plus symbol.
- In the **Assembly editor**, you can assign a **name** to the new assembly and select the **size** of the assembly. In addition, you must assign a **data type** (e.g., 8 bit, INT 8, etc.) to every byte of the assembly using drag and drop. Assemblies with multiple data types are possible.
- Click on **Save** to close the **Assembly editor** and create the new assembly.

5. Create an input and an output assembly of the desired size (e.g., 4 byte), assign each byte with the 8-bit data type and name the assemblies accordingly (e.g., **RxAssembly** for received data and **TxAssembly** for data to be sent). All bit of the assemblies will then be available as inputs or outputs of the FX3-GEPR in the logic editor of the main module.
6. Switch to the device window of the second Flexi Soft system and go to **Configuration**, **GEPR**, **Connection creation**.
7. Create one output and one input assembly in the second Flexi Soft system. The size and type of the input assembly for the second system must correspond to the output assembly of the first system and vice versa.
8. Under **Available devices** at the bottom left, double-click on the first Flexi Soft system to include this in the project. The system is displayed in the **Connection creation** window on the top left as an input and on the top right as an output with the relevant assembly.
9. Drag and drop the output assemblies of each system to the input assembly of the other system. The assemblies will be connected to one another.

Deleting connections and assemblies

Deleting a connection

► Open the device window of the master for the connection you wish to delete. From there, switch to **Configuration**, **GEPR**, **Connection creation**.
► Hover the mouse pointer over the connection line for the desired connection. A context window appears with basic information about the connection in question. In this context window, click on the **Delete** button.

Deleting an assembly

► If the assembly being deleted is connected, you must first delete the connection.
► Double-click on the assembly to open the **Assembly Detail Editor**.
► In **Assembly-Detail-Editor**, click on the **Delete assembly** button.

Setting up EFI-pro communication between an FX3-GEPR and another SICK device

This section describes how an EFI-pro data connection between an FX3-GEPR and another EFI-pro-enabled SICK device is set up.

Setting up a connection in the connection view

1. Start Safety Designer, open a new project, and create a Flexi Soft system with a main module and an FX3-GEPR, as described in the Basic configuration section.
2. Add an EFI-pro-enabled SICK device (e.g., a microScan3 Pro EFI-pro safety laser scanner) to the project in Safety Designer.
3. Connect the Flexi Soft system as the originator to the other device as the target in Safety Designer under **Connections**, as described in the Configuring EFI-pro connections in Safety Designer section.

✓ The predefined safety assemblies of the microScan3 Pro EFI-pro will be automatically applied to the FX3-GEPR during this procedure.

NOTE
This is the recommended procedure.

Setting up a connection in the device window

1. Start Safety Designer, open a new project, and create a Flexi Soft system with a main module and an FX3-GEPR, as described in the Basic configuration section.
2. Add an EFI-pro-enabled SICK device (e.g., a microScan3 Pro EFI-pro safety laser scanner) to the project in Safety Designer.
3. In the Flexi Soft system device window under **Configuration**, **GEPR**, **Device identification**, assign a **Safety Network Number (SNN)** that is consistent across the entire project to the EFI-pro gateway.
4. In the Flexi Soft system device window under **Configuration**, **GEPR**, **Network settings**, assign an **IP address** to the EFI-pro gateway.
5. Also assign an **IP address** and the **Safety Network Number (SNN)** to the EFI-pro-enabled SICK device. The steps required for this are described in the operating instructions for the device in question.
6. Open the Flexi Soft system device window and double-click on the EFI-pro-enabled SICK device under **Configuration**, **GEPR**, **Connection creation** at the bottom left under **Available devices**. The device is then displayed on the top left as an input and on the top right as an output with its respective input and output assemblies that were pre-configured ex works.
7. Drag the desired input and output assemblies for the EFI-pro-enabled SICK device to the Flexi Soft system. The input and output assemblies will be applied and will then be available as inputs or outputs of the FX3-GEPR in the logic editor for the main module of the Flexi Soft system.

4.3.1.3 Editing the EFI-pro connection settings

In the device window under Configuration, GEPR, Connection overview, you can manage the following parameters of each individual connection:

- Connection name
  Double-click on the desired connection name to change it.
- Data transmission rate
  The data transmission rate (also known as the requested packet interval (RPI)) specifies the frequency for updating an assembly. It is expressed in milliseconds.
- Max. lost packages
  This parameter specifies how many data packages may be lost, i.e., how often the update process can fail before the device goes into the error state.
- Network Delay
  This parameter allows an additional tolerance for avoiding faults/errors due to data transmission delays.

The response time of the relevant connection in milliseconds is calculated on the basis of the last three parameters and displayed.

NOTE

You can also edit the EFI-pro connection settings of an individual connection under Configuration, GEPR, Connection creation:

- ► Hover the mouse pointer over the connection line for the desired connection. A context window appears with basic information about the connection in question.
- ► In this context window, click on the **Edit** button. The dialog box for editing the parameters described above opens.

Capacity utilization

In the **Capacity utilization** area, the cycle time required by S bus+ and the total number of routed bytes are displayed. This data is used to calculate the response time.

The response time of the FX3-GEPR from the network to the logic of the main module is a result of the sum of the network response time and the cycle time of the S bus+.

Other factors such as the response time of the sensors and actuator used are relevant for the calculation of the total response time of the Flexi Soft system.

WARNING
Faulty calculation of the total response time 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.

► When calculating the total response time of the Flexi Soft system, always consider all relevant factors.

For detailed information on calculating the total response time of a Flexi Soft system, please refer to the operating instructions titled “Flexi Soft Modular Safety Controller Hardware” (SICK part number 8012999).

4.3.1.4 Setting up EtherNet/IP™ communication

Possible EtherNet/IP™ connections

- EtherNet/IP™ with CIP Safety™
  FX3-GEPR as the originator with an EtherNet/IP™ CIP Safety™ device as the target
- EtherNet/IP™ standard communication
  FX3-GEPR as the target with an EtherNet/IP™ standard device as the originator

FX3-GEPR as the originator (EtherNet/IP™ with CIP Safety™)

This section describes how a data connection between an FX3-GEPR as the originator and an EtherNet/IP™ CIP Safety™ device from a third-party supplier as the target is set up.

► Start Safety Designer, open a new project, and create a Flexi Soft system with a main module and an FX3-GEPR, as described in the Basic configuration section.
► In the Flexi Soft system device window under **Configuration**, **GEPR**, **Device identification**, assign a **Safety Network Number (SNN)** that is consistent across the entire project to the EFI-pro gateway, as described in the Basic configuration section.
► In the Flexi Soft system device window under **Configuration**, **GEPR**, **Network settings**, assign an individual **IP address** to the EFI-pro gateway, as described in the Basic configuration section.
► In Safety Designer, create a **generic EtherNet/IP CIP safety device** for the third-party EtherNet/IP™ CIP Safety™ device and configure it so that it is suitable for the project, see "Generic EtherNet/IP CIP safety device", page 105.
  - Assign a suitable **IP address** and the **SNN** which is consistent across the entire project to the device.
  - Define the assemblies of the device.
► Connect the Flexi Soft system as the originator to the EtherNet/IP™ CIP Safety™ device as the target in Safety Designer under **Connections** (see "Configuring EFI-pro connections in Safety Designer", page 107).

Or:

► In the Flexi Soft system device window under **Configuration**, **GEPR**, **Connection creation**, double-click on the EtherNet/IP™ CIP Safety™ device from the third-party supplier to include it in the project and connect the input and output assemblies of the device to the corresponding assemblies of the FX3-GEPR.
✓ The input and output assemblies of the device will then be available as inputs or outputs of the FX3-GEPR in the logic editor of the main module.

FX3-GEPR as the target (EtherNet/IP™)

This section describes how to set up a data connection between an FX3-GEPR as the target and an EtherNet/IP™ device from a third-party supplier as the originator.

1. Start Safety Designer, open a new project, and create a Flexi Soft system with a main module and an FX3-GEPR, as described in the Basic configuration section.
2. Assign an **IP address** to the FX3-GEPR.

✓ The FX3-GEPR can be addressed by other EtherNet/IP™ devices under the assigned IP address.

3. Open the device window and switch to **Configuration**, **GEPR**, **EtherNet/IP™ access**.
4. Under **Enable alternative access**, select and activate all desired static assemblies:
   - Communication type (EtherNet/IP™ or CIP Safety™)
   - Data transmission direction (Flexi Soft to network or network to Flexi Soft)
   - Size of the assembly (4 or 10 bytes per direction)
     In total, up to four static assemblies (one assembly per communication type and transmission direction) can be activated.
   - Using the assembly:
     - Not used
     - For EFI-pro (only available for safety assemblies)
     - For external communication

     Static assemblies that are used for **EFI-pro** or **for external communication** are available in the logic editor of the main module under **Inputs**, **GEPR** or under **Outputs**, **GEPR**.
     Static safety assemblies that are used for **EFI-pro** are available under **Configuration**, **GEPR**, **Connection creation** for connections with other EFI-pro-capable devices.

✓ Safety Designer calculates the cycle time required by S bus+ and the number of routed bytes from the number and size of the activated assemblies. This information is displayed under **Connection overview** in the **Capacity utilization** area.

EtherNet/IP™ overview

On the **EtherNet/IP™ overview** page, all data that is relevant to EtherNet/IP™ connections is displayed. The EDS file of the FX3-GEPR also contains this information, which is required to connect into external controls.

NOTE
The EDS file for the FX3-GEPR can be downloaded from www.sick.com.

Device services

Under Device services, EtherNet/IP™ service functions (SNCT services) can be used in connection with an FX3-GEPR:

- Remove target owner link to a safety controller
- Reset the safety configuration
- Set a Target Unique Node Identifier (TUNID)

For detailed information on using the device services, see "Device services", page 106.

4.3.1.5 CIP Standard

The Common Industrial Protocol (CIP) is a network-independent application protocol which works independently of the transmission medium. Control units can access common device profiles and object libraries via the CIP.

NOTE
The FX3-GEPR uses a manufacturer-specific device profile.

Implicit messaging is used for real-time communication between two network nodes in EtherNet/IP™. With implicit messaging, the I/O data is exchanged cyclically between two devices, for example, using the communication manager within the CIP. Implicit messaging uses UDP/IP via the port 2222.

For event-controlled communication which does not have to occur in real time, explicit messaging is used in EtherNet/IP™. Explicit messaging uses TCP/IP. It is used to read the parameters of a CIP object, for example.

The functionalities for EtherNet/IP™ standard communication supported by the FX3-GEPR are described in more detail below.

CIP Standard connections

The FX3-GEPR supports both master functionalities (originator) and slave functionalities (target). The maximum number of standard connections is as follows:

Table 101: Max. number of standard connections for the FX3-GEPR

Type of connection	Maximum number of connections
Standard originator	0
Standard target	8 (max. 64 implicit and explicit in total)

CIP object model

An object is an abstract representation of functionalities in a device. An object is described by its data, variables, functions, services, and behavior. The object model for the FX3-GEPR is explained in more detail in the section on CIP Safety™.

Assembly object

The assembly object provides data containers, known as assemblies, for cyclical data exchange. The FX3-GEPR only supports static standard assemblies. The instance number is predefined.

Table 102: Predefined standard input assemblies

Instance	Size	Contents
130	4 bytes	Process data, freely definable
131	10 bytes	Process data, freely definable

Table 103: Predefined standard output assemblies

Instance	Size	Contents
140	4 bytes	Process data, freely definable
141	10 bytes	Process data, freely definable
142	50 bytes	With the FX3-GEPR, data set 1 is generated dynamically in line with the configured connections and cannot be modified.
143	32 bytes	Data set 2 contains the checksums of the system configuration.
144	60 bytes	Data set 3 contains the status and diagnostic data for the individual modules with four bytes per module.
145	60 bytes	Data set 4 is filled with reserved values.

The freely definable standard input assemblies (instance 130 and 131) and standard output assemblies (instance 140 and 141) are configured as follows:

- ► In the **EtherNet/IP™** area under **Configuration**, **GEPR**, **EtherNet/IP™ access**, go to the left drop-down menu and select the desired size of the assemblies (**4 bytes – assembly 130** or **10 bytes – assembly 131** or (**4 bytes - assembly 140** or **10 bytes – assembly 141**).
- ► In the right drop-down menu for each assembly you wish to use, select **For external communication**.
- ► The selected assemblies are available in the **Logic editor** under **Inputs**, **GEPR**, or under **Outputs**, **GEPR**.

Table 104: Standard assemblies in the logic editor

Instance	Size	Type	Name in the logic editor
130	4 bytes	Input assembly	Standard input 1
131	10 bytes	Input assembly	Standard input 2
140	4 bytes	Output assembly	Standard output 1
141	10 bytes	Output assembly	Standard output 2

4.3.1.6 CIP Safety™

CIP Safety™ extends the Common Industrial Protocol (CIP) with safety functions.

The FX3-GEPR transfers safety-related data using CIP Safety™ via real-time Ethernet with a safety integrity level (SIL) of up to SIL3 in accordance with IEC 61508 and IEC 62061.

It receives data from other network nodes via CIP Safety™, checks these in accordance with the safety mechanisms of the CIP Safety™ specification (ODVA CIP Network Library, Volume 5 – CIP Safety™), and provides valid safety data to the application in line with SIL3.

When application data is sent, CIP Safety™ frames are generated in accordance with SIL3 and distributed to other network nodes via the network at a previously configured interval.

CIP Safety™ connections

The FX3-GEPR supports single-cast safety connections. A single-cast safety connection can be found between exactly two network nodes (point-to-point connection).

The FX3-GEPR supports both master functionalities (originator) and slave functionalities (target). The maximum number of safety connections is as follows:

Table 105: Max. number of safety connections for the FX3-GEPR

Type of connection	Maximum number of connections
Safety originator	22
Producing safety target	11
Consuming safety target	11

NOTE

The FX3-GEPR does not support the SNCT interface (Safety Network Configuration Tool). Instead, Safety Designer configuration software, which is specific to SICK, is used to configure safety connections.

When a connection is established, this is always initiated by a SafetyOpen service from the originator to the target. In the process, the originator sends the TUNID (Target Unique Node Identifier), OUNID (Originator Unique Node Identifier), and the SCID (Safety Configuration Identifier) to the target, as well as the configuration data.

When the connection is being established, Safety Designer does not send any configuration data to the device. The FX3-GEPR uses the SafetyOpen service type 2b for this, with which the SCID is always zero (SCID=0).

CIP Safety™ objects

The FX3-GEPR supports the following CIP Safety™ objects:

Table 106: CIP Safety™ objects supported by the FX3-GEPR

Class code	Class	Description	Instances
0x01	Identity Object	Information about the network nodes	1

Class code	Class	Description	Instances
0x02	Message router object	Processes all messages and routes them to the suitable objects	1
0x04	Assembly object (I/O assembly class)	Combines attributes (data) of different objects to form a single object. Used for I/O messages	Max. 32 [1]
0x06	Connection manager object	Contains connection-specific attributes for triggering, transmission, and connection type	1
0x39	Safety supervisor object	Central object of a CIP Safety™ device. It includes the behavior of the state machine and the optional SNTC support.	1
0x3A	Safety validator object	Contains all the information required to coordinate and maintain reliable safety connections between client and server applications	Max. 48 [2]
0x47	Device level ring (DLR) object	Contains the configuration and status information of the DLR protocol	1
0x48	Quality of service (QoS) object	Used to manage data traffic with different relative priorities	1
0xF5	TCP/IP interface object	Contains all attributes for configuring the TCP/IP interface	1
0xF6	Ethernet link object	Contains connection-specific attributes, such as transmission rate, MAC address, or duplex mode	2

1) Corresponds to the number of assemblies in the device.
2) Corresponds to the number of CIP Safety™ connections.

Assembly object

Class code: 0x04

The assembly object provides data containers, known as assemblies, for cyclical data exchange. The FX3-GEPR supports static and dynamic safety assemblies.

Table 107: Static safety assemblies of the FX3-GEPR

Number and type of static assemblies	Description
2 predefined safety input assemblies	• Instance 100, 4 bytes of process data, freely definable • Instance 101, 10 bytes of process data, freely definable
2 predefined safety output assemblies	• Instance 120, 4 bytes of process data, freely definable • Instance 121, 10 bytes of process data, freely definable

The freely definable static safety input assemblies (instance 100 and 101) and static safety output assemblies (instance 120 and 121) are configured as follows:

► Under **Configuration**, **GEPR**, **EtherNet/IP™ access** in the **CIP Safety™** area in the left drop-down menu, select the desired size of the assemblies (**4 bytes – assembly 100** or **10 bytes – assembly 101** or (**4 bytes – assembly 120** or **10 bytes – assembly 121**).
► In the right drop-down menu for each assembly you wish to use, select either **for EFI-pro** or **for external communication**.
► The selected static safety assemblies are available in the **Logic editor** under **Inputs**, **GEPR**, or under **Outputs**, **GEPR**.

Table 108: Standard assemblies in the logic editor

Instance	Size	Type	Name in the logic editor
100	4 bytes	Input assembly	Safety input 1
101	10 bytes	Input assembly	Safety input 2
120	4 bytes	Output assembly	Safety output 1
121	10 bytes	Output assembly	Safety output 2

Static safety assemblies that are used for **EFI-pro** are available under **Configuration**, **GEPR**, **Connection creation** for connections with other EFI-pro-capable devices.

Dynamic safety assemblies

Table 109: Dynamic safety assemblies of the FX3-GEPR

Number and type of dynamic assemblies	Description
Maximum 10 dynamic safety input (Tx) assemblies	Instance 150...189, maximum 48 bytes (Rx assemblies) or 50 bytes (Tx assemblies) of process data in total, freely definable
Maximum of 10 dynamic safety output (Rx) assemblies	

Configuring dynamic safety assemblies

To configure the dynamic safety assemblies see "Setting up EFI-pro communication between two FX3-GEPRs", page 95.

Safety supervisor object

Class code: 0x39

The safety supervisor object supports the following services and instance attributes:

Table 110: General services of the safety supervisor object

Service code	Service name	Implemented in class	Implemented in instance	Description
0x0E	Get_Attributes_Single	Yes	Yes	Supplies of the content of the specified attribute
0x10	Set_Attribute_Single	No	Yes	Changes an attribute value
0x54	Reset	–	Yes	Resets the device. Function in line with the identity object; however, a password is also required. Only type 0 is supported.

Table 111: Instance attributes of the safety supervisor object

Attribute ID	Access type	Name	Data type	Description	Default
11	Read	Device status	USINT	Status of the safety device	–
12	Read	Exception status	BYTE	Bit-coded diagnostic data of the device	–
15	Read/ Write	Alarm enable	BOOL	Activates/deactivates alarms 0: Deactivated 1: Activated	1
16	Read/ Write	Warning enable	BOOL	Activates/deactivates warnings 0: Deactivated 1: Activated	1

Attribute ID	Access type	Name	Data type	Description	Default
25	Read	Configuration UNID (CFUNID)	10 octets	Identifies the source of the device configuration 0: Undetermined, accepts all All = 0xFF: Configuration only via tool	All = 0xFF
26	Read	Safety Configuration Identifier (SCID)	10 octets	Made up of Safety Configuration CRC + Safety Configuration Time Stamp	0
27	Read	Target UNID [1]	10 octets	The current UNID of the device made up of the SNN [2] and IP address	All = 0xFF
28	Read	Output Connection Point Owners (OCPUNID)	STRUCT	Contains the number of output resources, their UNIDs, and the data paths of the assemblies	–

1) The Unique Node Identifier (UNID) identifies a node uniquely in the entire network. It is made up of 10 bytes (6 bytes for SNN +4 bytes for IP address).

2) Every safety EtherNet/IP™ node must be assigned a Safety Network Number (SNN). The SNN is made up of the date (2 bytes) and the time (4 bytes).
The decimal date values 1 to 11,687 (January 1, 1972 to December 31, 2003) are reserved for manual SNN settings.

Safety validator object

Class code: 0x3A

The safety validator object supports the following services and instance attributes:

Table 112: General services of the safety validator object

Service code	Service name	Implemented in class	Implemented in instance	Description
0x0E	Get_Attributes_Single	Yes	Yes	Supplies of the content of the specified attribute
0x10	Set_Attribute_Single	No	Yes	Changes the specified attribute
0x4B	Reset error code	Yes	No	Resets attribute 8 (safety connection fault count)

Table 113: Class attributes of the safety validator object

Attribute ID	Access type	Name	Data type	Description
8	Read	Safety connection fault count	UINT	Counter for safety connection faults

Table 114: Instance attributes of the safety validator object

Attribute ID	Access type	Name	Data type	Description
1	Read	Validator state	USINT	Status of the safety connection 0 = Not connected 1 = Initialization 2 = Connected 3 = Fault

Attribute ID	Access type	Name	Data type	Description
2	Read	Validator type	USINT	Currently executed validator type of this instance 0 = Not connected 1 = Singlecast 2 = Multi-cast
3	Read	Ping interval EPI multiplier	UINT	Provides the Ping_Count_Interval of this connection
7	Read	Max consumer number	USINT	Maximum number of possible consumers with this connection
12	Read/ Write	Max data age	UINT	Diagnostics: Highest detected data age in increments of 128 µS. The only valid value during write access is 0.
13	Read	Application data path	EPATH	Safety data path for this connection
15	Read	Producer/ consumer fault counters	STRUCT of:	
		Producer/ consumer counter array size	USINT	For multi-cast producers = max. consumer number +1 For single-cast or multi-cast consumers = 1
		Producer/ consumer fault counter	Array of USINT	Number of faults which occurred in the last hour

Failsafe values

In the event of a fault/error, i.e., when no valid input data is received, the data content of the input assemblies is treated as follows:

- All data bytes are set to 0x00.
- The input data is marked as “Invalid”.

In the event of a fault/error, the output data is treated as follows:

- All data bytes are set to 0x00.
- The Run_Idle bit in the CIP Safety™ frame (mode byte) is set to 0 (idle).

4.3.2 Generic EtherNet/IP CIP safety device

Create a **generic Ethernet/IP CIP safety device** in the device catalog in the main Safety Designer window. It is possible to integrate devices from third-party manufacturers into a Flexi Soft system via EtherNet/IP™ in this way.

Possible EtherNet/IP™ connections

- EtherNet/IP™ with CIP Safety™
  FX3-GEPR as the originator with an EtherNet/IP™ CIP Safety™ device as the target

4.3.2.1 Creating generic Ethernet/IP CIP safety device

► Double-click on **generic Ethernet/IP CIP safety device** in the device catalog.
✓ A device tile for the generic Ethernet/IP CIP safety device is added in the device overview.

4.3.2.2 Configuring generic Ethernet/IP CIP safety device

Procedure

1. Click on the device tile for the generic Ethernet/IP CIP safety device to open the associated device window.
2. In the **General** navigation, make the following settings:
   - ► Enter a **device name** and assign an image to the device where applicable.
   - ► Enter the device **IP address** and the project **safety network number**.
   - ► Enter the general characteristics of the new device (**supplier**, **product type**, **product code**, **main version**, and **minor version**). You can find this data in the manual or the EDS file for the device.
   - ► Optionally, enter a **user** and/or a **comment**.
3. Under **Default settings**, enter the general connection parameters. This includes the **data format**, as well as the **maximum fault number** and the **standard RPI rate** where applicable. You can find this data in the manual or the EDS file for the device.
4. Optionally, you can also enter a **configuration signature** under **Default settings**.
5. In the **Connections** navigation, configure the connection paths of the new device. These can be calculated from the assembly data of the device (can be found in the manual or the EDS file). Alternatively, you can enter the hexadecimal connections paths directly under **Define connection path**.

NOTE
You must always enter the **assembly size**, even if a hexadecimal connection path is being used.

Complementary information

There are pre-configured generic profiles for different robot controls available for import in the configuration of the generic EtherNet/IP CIP safety device. You can select the profiles in the **General** navigation via **Import example**.

4.3.2.3 Device services

Under Device services, EtherNet/IP™ service functions (SNCT services) can be used in connection with an FX3-GEPR:

- Remove target owner link to a safety controller
- Reset the safety configuration
- Set a Target Unique Node Identifier (TUNID)

NOTE
To use the SNCT services for a device from a third-party manufacturer, the device must support these functions.

The SNCT services can only be executed if Safety Designer is connected to the Flexi Soft system.

To execute the SNCT services, the current Safety Network Number (SNN) and the IP address of the device must be known.

To execute the SNCT services, the device must not have an active safety connection. All connections to previous safety networks must be interrupted or stopped.

Remove target owner link to a safety controller

If the device has already been connected to a safety controller and is now to be connected to another safety controller, the link to the previous safety controller must be explicitly removed.

NOTE
When the link is removed, the safety configuration of the device loses its verification.

- ► In the device window of the device in question, click on **Open device window with removal function** under **Device services**. The **Services for external devices** page is opened in the FX3-GEPR device window, and the **Target IP address** as well as the **safety network number** of the device in question are transferred.
- ► Under **Target owner link**, click on **Remove target owner link to [IP address]**.

Reset the safety configuration

This command can be used to reset the safety configuration of the device. The IP address and the password are retained.

NOTE
By resetting, the device loses its existing safety configuration. If required, the device must then be reconfigured using the configuration software provided by the manufacturer.

- ► In the device window of the device in question, click on **Open device window with reset function** under **Device services**. The **Services for external devices** page is opened in the FX3-GEPR device window, and the **Target IP address** as well as the **safety network number** of the device in question are transferred.
- ► Under **Reset safety configuration**, click on **Reset safety configuration to [IP address]**.

Set a Target Unique Node Identifier (TUNID)

This command can be used to reset the device TUNID, after the link to the previous safety controller has been removed and the safety configuration has been reset.

- ► In the device window of the device in question, click on **Open device window with TUNID function** under **Device services**. The **Services for external devices** page is opened in the FX3-GEPR device window, and the **Target IP address** as well as the **safety network number** of the device in question are transferred. The displayed device status is **waiting for target unique node identifier (TUNID)**.
- ► Enter the new **Target IP address** (if modified) and the Flexi Soft project **SNN** for the device here.
- ► Under **Set Target Unique Node Identifier (TUNID)**, click on **Write TUNID to [IP address]**.

NOTE
Depending on the type of third-party manufacturer device in question, it may then need to be reconfigured, using the configuration software provided by the manufacturer.

4.3.3 Configuring EFI-pro connections in Safety Designer

The Safety Designer is able to establish a network between the following systems:

- Flexi Soft systems with an FX3-GEPR EFI-pro gateway
- Other EFI-pro-enabled SICK devices (e.g., microScan3 Pro EFI-pro)
- Devices from third-party manufacturers that support EtherNet/IP™ CIP Safety™

NOTE
EtherNet/IP™-enabled devices from third-party manufacturers can be integrated as **generic EtherNet/IP CIP safety devices** in projects in Safety Designer, see "Generic EtherNet/IP CIP safety device", page 105.

4.3.3.1 Configuring network

Important information

NOTE

- An originator must always be networked to a target. Networking two originators or two targets is not possible.
- If you click on a networking icon, then the networking icons with which a network can be established are highlighted.
- Double-clicking on the EFI-pro interface of a device tile opens the associated configuration page in the device window for this device.
- Double-clicking on the EFI-pro networking line opens the associated configuration page in the device window for the originator of this network.

Procedure

1. In the main Safety Designer window, create a project with the desired devices and configure these as required.
2. Click on **Networking** in the Safety Designer main window.

✓ The networking view opens.
In the networking view, the tile of every EFI-pro-enabled device contains one or two networking icons for EFI-pro networks.
   - A networking icon on the upper edge of the device tile indicates that the device can act as a CIP Safety™ target.
   - A networking icon on the lower edge of the device tile indicates that the device can act as a CIP Safety™ originator.

3. Click on a networking icon and establish a network by dragging it to another networking icon. During the drag and drop process, possible connection points for networks are highlighted in blue.

4.3.3.2 Deleting network

► In this context window of the networking line, click on the Delete button.

4.3.4 Data transmission rate for EFI-pro connections

Table 115: Standard data transmission rate for EFI-pro connections

Type of connection	Data transmission rate for outputs	Data transmission rate for inputs
FX3-GEPR with FX3-GEPR	12 ms	12 ms
FX3-GEPR with EFI-pro microScan 3	12 ms	10 ms
FX3-GEPR with generic EtherNet/IP CIP safety device	12 ms	Depends on the generic Ether-Net/IP CIP safety device

NOTE

In order to ensure plant availability, it may be necessary, depending on the number of connections and the system topology, to adapt the data transmission rate or the network delay.

See also "Editing the EFI-pro connection settings", page 97.

5 The process image

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

Available data

The Flexi Soft gateways can provide the following data:

- Operating data
  - **Input values** (1/0) of all Flexi Soft input expansions (see "Direct gateway output values", page 111)
  - **Output values** (1/0) of all Flexi Soft input/output expansions (see "Direct gateway output values", page 111)
  - **Logic results** from the Flexi Soft main module (FX3-CPUx) (see "Logic results", page 111)
  - **Output data** from another network, i.e., data which was received by a second gateway in the Flexi Soft system (see "Routing data from a second network", page 113)
- Diagnostics
  - **Fault/error and status information** for all modules except UE410-2RO and UE410-4RO (see "Error and status information of the modules", page 113)
  - **Checksums** (see "Configuration checksums", page 113)

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 split into four input data sets.

- **Data set 1** (max. 50 bytes) contains the operating data. The content of data set 1 can be modified using Safety Designer. It is preset when the gateway is delivered (for details, see table 117, page 110).
  With the FX0-GPNT, 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 corresponding section on the gateway in question.
  With the FX3-GEPR, data set 1 is generated dynamically in line with the configured connections and therefore does not correspond to the schema in table 117.
- **Data set 2** (32 bytes) contains the checksums of the system configuration (see "Configuration checksums", page 113).
- **Data set 3** (60 bytes) contains the status and diagnostic data for the individual modules with four bytes per module (for details, see "Error and status information of the modules", page 113).
- **Data set 4** (60 bytes) is currently filled with reserved values and should therefore not be used for the application.

The table below gives an overview of which data sets are provided by which gateway.

Table 116: 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-GCAN	CANopen	CANopen (SDOs)	CANopen (SDOs) [1)]	–

	Data set 1	Data set 2	Data set 3	Data set 4
FX0-GETC	Input and output objects 2000h and 2001h	Checksums Object 2002h	Status and diagnostics Object 2003h	Reserved Object 2004h
FX3-GEPR	EtherNet/IP™ (CIP Standard)	EtherNet/IP™ (CIP Standard)	EtherNet/IP™ (CIP Standard)	EtherNet/IP™ (CIP Standard)

1) The FX0-GCAN provides status and diagnostic data via CANopen SDOs (service data objects), see "SDO communication", page 76.

Table 117: Overview of input data sets 1–3 (default setting for EtherNet/IP™, Modbus® TCP, and TCP/IP)

	Data set 1 [1)]	Data set 2	Data set 3
Byte 0	Logic result 0	Overall checksum value	Status module 0. Module 0 is always the main module. For detailed information about the module status: see "Error and status information of the modules", page 113.
Byte 1	Logic result 1		
Byte 2	Logic result 2		
Byte 3	Logic result 3		
Byte 4	Input values of module 1	Flexi Soft checksum	Status of module 1
Byte 5	Input values of module 2		
Byte 6	Input values of module 3		
Byte 7	Input values of module 4		
Byte 8	Input values of module 5	FX3-CPU0 and FX3-CPU1: Reserved FX3-CPU2 and FX3-CPU3: ACR checksum	Status of module 2
Byte 9	Input values of module 6		
Byte 10	Input values of module 7		
Byte 11	Input values of module 8		
Byte 12	Input values of module 9	Flexi Soft checksum (verified)	Status of module 3
Byte 13	Input values of module 10		
Byte 14	Input values of module 11		
Byte 15	Input values of module 12		

	Data set 1 [1]	Data set 2	Data set 3
Byte 16	Output values of module 1	Reserved	Status of module 4
Byte 17	Output values of module 2		
Byte 18	Output values of module 3		
Byte 19	Output values of module 4		
Byte 20	Output values of module 5		Status of module 5
Byte 21	Output values of module 6		
Byte 22	Output values of module 7		
Byte 23	Output values of module 8		
Byte 24	Output values of module 9		Status of module 6
Byte 25	Output values of module 10		
Byte 26	Output values of module 11		
Byte 27	Output values of module 12		
Byte 28	Direct gateway output values 1		Status of 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 of module 8 ... 13
Byte 50 ... 55	Not available		
Byte 56			Status of module 14 (module 13 and 14 are always the gateways)
Byte 57			
Byte 58			
Byte 59			
Length	50 bytes	32 bytes	60 bytes

[1] With the FX3-GEPR, data set 1 is generated dynamically in line with the configured connections and therefore does not correspond to the schema shown here.

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 "Configuring the process image", page 120 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 into a gateway. Four bytes are reserved for this purpose in the default setting for data set 1; however, all 50 bytes from data set 1 can be used as direct gateway output values. For additional information, see "Direct gateway output values", page 123.

5.1.3 Module status, input and output values

The Flexi Soft gateways can transfer the status plus the input and output values of all Flexi Soft modules connected to the Flexi Soft system to the network. Data set 1, which contains the input and output values, can be adjusted as needed. For detailed information, see "Configuring the process image", page 120 and the chapter on the gateway in question.

Module status

The Flexi Soft gateways can transfer the status of the connected modules to the network. A total of six bytes are available for this purpose.

Table 118: Module status

Module status	Size	Meaning	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 etc. 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	
Location 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

For information on the meaning of the status bits, please refer to the operating instructions titled “Flexi Soft in the Safety Designer Configuration Software” (SICK part number 8013926).

Input values 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 at an I/O module, then only the low-order bit represents the status of the preliminary evaluation for the affected 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).
  In this case, the high-order bit (bit 1, 3, 5, and 7) indicates the status of the preliminary evaluation:
  - 0 = Error
  - 1 = No error
- Output values for I/O modules:

For each module with outputs, one byte is available for data set 1. The output values show the status of the control information from the logic of the main module for the affected element on the I/O module. The level of the associated terminal is not guaranteed to be identified from this as the output can be disabled due to the cross-circuit detection or overload detection.
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.

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.

- Total checksum:
  The same value as the Flexi Soft 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.
- Flexi Soft checksum (verified):
  This is the Flexi Soft checksum that applied at the time of the most recent verification. During verification, the Flexi Soft checksum of the current configuration in Safety Designer is transferred and compared to the checksum of the configuration stored in the device. If these two checksums are identical, the configuration of the Flexi Soft system is classed as verified (LED CV lights up ● yellow).

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

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 119: 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
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

Table 120: 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 121: 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

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 3	Reserved							

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

Table 122: 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 123: 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 124: 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 125: 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 126: 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 127: 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 120, page 114).

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 128: 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 120, page 114).

Example process image: see "Example of a TCP/IP process image", page 66.

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

The data received from the network is split into output data sets (max. 50 bytes). With the FX0-GENT, FX0-GMOD, FX0-GPNT, and FX0-GETC, these data sets are divided into five data blocks, each with ten bytes. The FX0-GCAN has four process data objects, each with eight bytes. With the FX3-GEPR, up to 11 safety assemblies can be freely configured with up to 48 bytes in total.

Table 129: 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-GCAN	8 bytes	8 bytes	8 bytes	8 bytes	–
FX3-GEPR	Up to 11 safety assemblies with up to 48 bytes in total				

The content of the output data sets can be used in the logic editor of the main module and provided 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 130: 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 split into one or several data blocks depending on the network protocol. For detailed information on the modularization of the data which is sent into the network, see table 131, page 119 and the section on the relevant gateway.

The content of data set 1 is preset when delivered but can be freely configured if required (see "Default settings for the operating data", page 118 and see "Configuring the process image", page 120).

The diagnostic data in data sets 2 to 4 depends on the network protocol used and is described in the section on the relevant 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 131: 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)

For corresponding information for the FX0-GETC, see table 35, page 52 and for the FX0-GCAN, see table 69, page 75.

With the FX3-GEPR, no assemblies are defined in the as-delivered state. This is done by the user, see "Setting up EFI-pro communication between two FX3-GEPRs", page 95.

The assignment of data sets or assemblies can be configured for all gateways, as shown in the following section.

5.5 Configuring the process image

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 Safety Designer Configuration Software” (SICK part number 8013926).

The configuration of the Flexi Soft gateway for data routing is displayed in the dialog box for the gateway configuration.

- ► In the device window, click on the **Configuration** button.
- ► In the navigation area, select the desired gateway from the list of configurable modules.
- ► Click on the **Data sets** button underneath.
- ► In the configuration area, click on the **Flexi Soft to network** tab to display the routing configuration.

The default is as follows (example for Modbus® TCP):

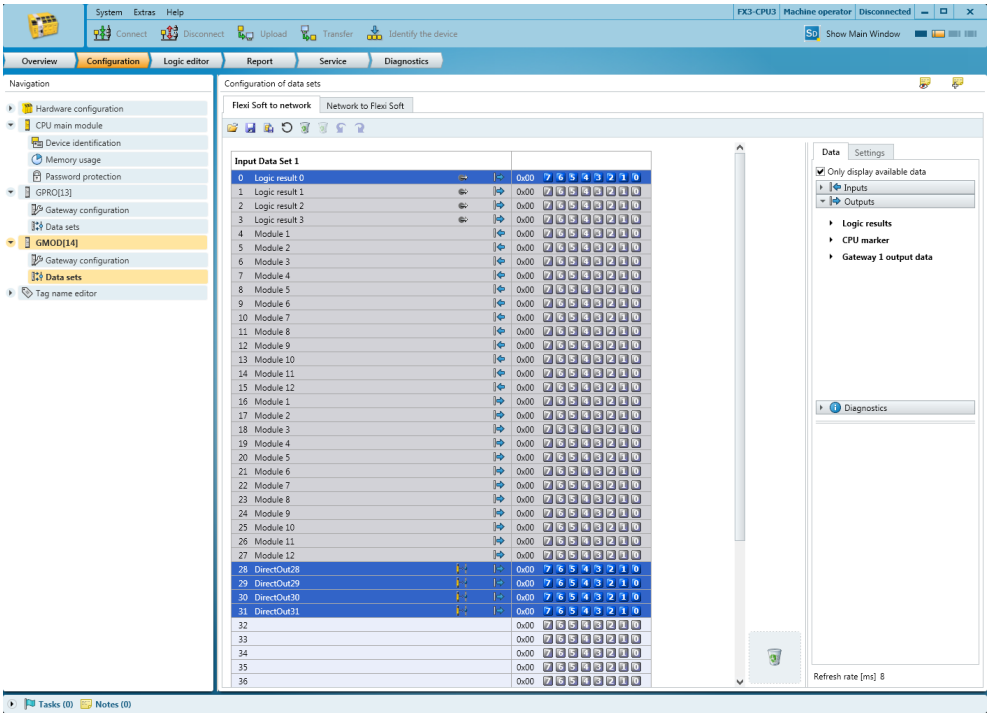

Figure 22: Preset configuration for the operating data transferred to the network

In the routing configuration area, the data sets or assemblies of the gateway for data transmission into the network are displayed, as well as the data sources available in the system under the **Data** tab and the tag names under the **Settings** tab. The toolbar can be found above the data sets.

NOTE
Features of the FX3-GEPR

The **Data** tab is not available in the FX3-GEPR. In the FX3-GEPR, the mapping of data sources to individual bits of the process image is carried out exclusively in the logic editor.

5.5.1 The toolbar

Figure 23: Routing configuration toolbar for the Flexi Soft to network transmission direction

Figure 24: Routing configuration toolbar for the network to Flexi Soft transmission direction

The toolbar contains buttons for the following actions (working from left to right):

- The **Load user configuration** and **Save user configuration** buttons are used to load or save the routing configuration in XML format. When you load a configuration, any unsaved changes to the routing configuration will be lost. This command cannot be undone.
- With the **Import** and **Export** buttons, you can export 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 you to import the tag names into a PLC program and use them there.

NOTE
The **Import** button is only available when the routing is configured for the network to Flexi Soft direction.

- **Reset to factory setting** restores the default routing configuration. This command cannot be undone.
- **Clear all** deletes the configuration, i.e., all bytes assigned to the data sets are deleted.
- **Delete routing** deletes the byte that is currently selected.
- The **Undo** and **Redo** buttons allow you to undo or redo the last changes that you have made to the configuration.

NOTE
Features of the FX3-GEPR

Using the Clear all and Delete routing buttons in the FX3-GEPR, you can delete the tag names only.

- The **Clear all** button deletes all the tag names of all the bytes in the routing table.
- The **Delete routing** button deletes all the tag names of all the currently selected bytes.
- For the Flexi Soft to network transmission direction, the **Clear all** and **Delete routing** buttons are not available. The tag names can only be deleted using the input mask.

5.5.2 The Data tab

NOTE
Features of the FX3-GEPR

The **Data** tab is not available in the FX3-GEPR. In the FX3-GEPR, the mapping of data sources to individual bits of the process image is carried out exclusively in the logic editor.

The **Data** tab contains all the sources from which data can be routed into the network. The data is split into three groups: the available **inputs** and **outputs**, as well as **diagnostics**.

- The **inputs** include the input values of the connected Flexi Soft modules. 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 **outputs** include the output values of the connected Flexi Soft modules and the **logic results** from the logic editor of the main module.
- **Diagnostics** includes the status information of the connected Flexi Soft modules.

All sources supported by the current configuration are shown in black:

- Connected Flexi Soft modules
- Connected EFI-pro devices or CIP Safety™ devices
- Configured logic results [1)]
- Data which is provided by another gateway in the system

Sources that are not supported by the current configuration are shown in gray. Checking the **Show only available data** box in the top left-hand corner hides the unused sources.

Sources which provide live data are indicated with an icon to the left of the text.

5.5.3 The configuration area for Flexi Soft to network

NOTE
Features of the FX3-GEPR

The configuration steps described in this section are not available in the FX3-GEPR. In the FX3-GEPR, the mapping of data sources to individual bits of the process image is carried out in the logic editor.

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 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 **Data** area to a free position in a data set or assembly (drag and drop). 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 byte from the routing table

► Drag the byte to be deleted onto the recycle bin icon (drag and drop).

1) In the default configuration, only the first byte of the logic results (logic result 0) is active and available. You can activate more output bits for logic results as required in the logic editor (see the operating instructions “Flexi Soft in the Flexi Soft Designer Configuration Software” (SICK part number 8012998)).

Or:

► Select the byte to be deleted by clicking on it. Then click the **Delete routing** button on the toolbar.

Or:

► In the context menu of the byte to be deleted, select the **Delete routing** command.

Moving a data byte to another position in the routing table

► Drag the byte to be moved to the desired position (drag and drop). 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 Tag names

Under the **Settings** tab, the tag names of all bits for the bytes currently selected in the routing configuration area are displayed. You can enter, edit, and delete the tag names in the logic editor or the tag name editor.

Under the Settings tab, you can only edit or delete the following tag names:

- Standard gateways: Tag names for the direct gateway output values
- FX3-GEPR: All tag names

Bits without a tag name are not visible in the logic editor and cannot be used.

5.5.5 Direct gateway output values

Values (e.g., logic results) can be written directly from the logic editor to a gateway. These direct gateway output values can found in the logic editor under **Outputs**.

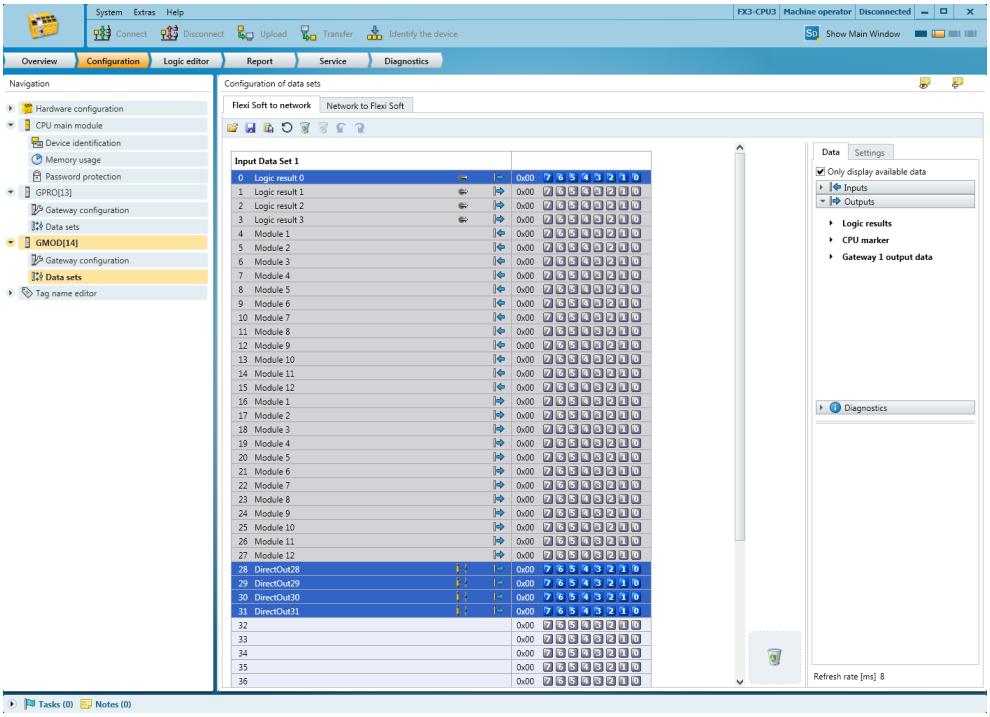

Figure 25: Direct gateway output values in the process image (default setting)

To use one or several bits of a byte as direct gateway output values, each bit which is to be used must be assigned a tag name.

Configuring direct gateway output values

► Select a free byte.

► On the **Configuration** tab, check **Use direct processing.**

- ► Enter a tag name for the selected byte.
- ► Enter tag names for the individual bits of the selected byte.

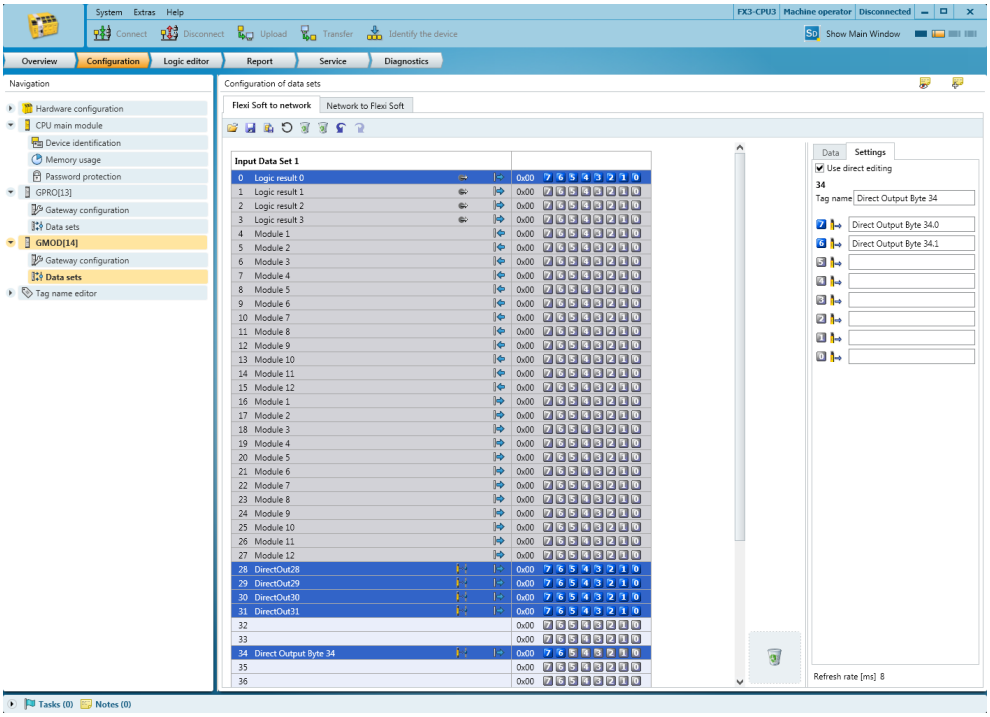

Figure 26: Configuring direct gateway output values in the process image

All bits with a tag name are available in the logic editor under **Outputs**.

NOTE

By default, the process images of the standard gateways contain four predefined bytes for direct gateway output values. These can be processed in the same way.

NOTE

Features of the FX3-GEPR

The process image of the FX3-GEPR works entirely using the principle of direct gateway output values. In the FX3-GEPR, the mapping of data sources to individual bits of the process image is carried out exclusively in the logic editor.

5.5.6 Configuring the output data (Network to Flexi Soft)

Activating incoming data bits

- ► Click on the button for the **Network to Flexi Soft** transmission direction.
  In the routing configuration area, the data sets or assemblies of the gateway for the data received from the network are displayed, as well as the tag names for the selected bytes respectively.
- ► Select a byte.
- ► Enter the desired 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 under **Inputs** and as a data source for a second gateway where applicable. Bits without a tag name are not visible in the logic editor and cannot be used.

5.5.7 Loading and saving a configuration

The **Load user configuration** and **Save user configuration** buttons are used to load or save the routing configuration in XML format. When you load a routing configuration, 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 you to import the tag names used in a Flexi Soft project into a PLC program and use them there.

When you import a configuration, 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.

- ► In the device window under **Configuration**, select the desired gateway and click on **Data sets** in the navigation tree.
- ► Click on the **Flexi Soft to network** tab or on the **network to Flexi Soft** tab to display the routing tables 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:

Input Data Set 3		IB20...IB29
IB20	Module 9	0x00 7 6 5 4 3 2 1 0
IB21	Module 10	0x00 7 6 5 4 3 2 1 0
IB22	Module 11	0x00 7 6 5 4 3 2 1 0
IB23	Module 12	0x00 7 6 5 4 3 2 1 0
IB24	Logic result 0	0x00 7 6 5 4 3 2 1 0
IB25	Logic result 1	0x00 7 6 5 4 3 2 1 0
IB26	Logic result 2	0x00 7 6 5 4 3 2 1 0
IB27	Logic result 3	0x00 7 6 5 4 3 2 1 0
IB28	DirectOutIB28	0x00 7 6 5 4 3 2 1 0
IB29	DirectOutIB29	0x00 7 6 5 4 3 2 1 0

Figure 27: Active and inactive bits in the online process image

In the **Flexi Soft to network** view, 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:

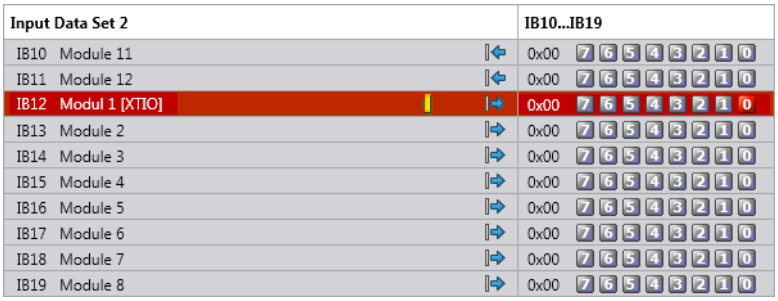

Figure 28: Inactive network input bits as a result of a fault/error

In the **Network to Flexi Soft** direction view, 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 29: Network output bits without assigned tag names 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 Safety Designer Configuration Software” (SICK part number 8013926).

6 Troubleshooting

For information on diagnosing the Flexi Soft system, please refer to the operating instructions titled “Flexi Soft in the Safety Designer Configuration Software” (SICK part number 8013926).

6.1 The FX0-GENT EtherNet/IP™ gateway

Table 132: Troubleshooting the FX0-GENT

Error	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 and 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).

Error	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.	Switch the voltage supply for the Flexi Soft system off and then back on again. Check whether the FX0-GENT is positioned correctly in the Flexi-Soft system. Use the configuration software to check the diagnostic messages. 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 dirty 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-GENT 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 133: Troubleshooting the FX0-GMOD

Error	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.

Error	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 and 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.	Switch the voltage supply for the Flexi Soft system off and then back on again. Use the configuration software to check the diagnostic messages. If the fault persists, replace the gateway.

Error	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 dirty 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 134: Troubleshooting the FX0-GPNT

Error	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 has a fixed IP address.	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	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 and 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).

Error	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 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.	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. 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) 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 135: Troubleshooting the FX0-GETC

Error	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.
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).

Error	Possible cause	Possible measures
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.	Switch the voltage supply for the Flexi Soft system off and then back on again. Use the configuration software to check the diagnostic messages. 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 dirty 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-GCAN CANopen gateway

Table 136: Troubleshooting the FX0-GCAN

Error	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 heartbeat 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.
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.

Error	Possible cause	Possible measures
The FX0-GCAN is not supplying any PDO data. PWR ● green NS ○ OFF/● Red/● Green MS ☼ Green (1 Hz)	The FX0-GCAN is in the “Idle” state. The node guarding or heartbeat messages are being 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 PDO 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.
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 heartbeat 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.

Error	Possible cause	Possible measures
The FX0-GCAN is in the “Serious error” state. PWR ● green NS ● Red MS ☼ Red (2 Hz)	Internal device error on the FX0-GCAN.	Switch the voltage supply for the Flexi Soft system off and then back on again. Use the configuration software to check the diagnostic messages. 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 dirty 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.

6.6 The FX3-GEPR EFI-pro gateway

CAUTION

Unexpected machine startup

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

► Do not use LED indicators for safety-relevant functions; they must only be used for general diagnostic purposes during commissioning or for troubleshooting. (ODVA SRS105)

NOTE

- In online mode, the current status of the diagnostic bits of the FX3-GEPR is displayed under **Configuration**.
- Error messages and the diagnostic history of the FX3-GEPR are displayed under **Diagnostics**.
- The status of each configured incoming connection of the FX3-GEPR can be found in the logic editor of the main module under **Inputs** => **GEPR** => **Status** as a diagnostic bit and can be evaluated in the logic program.

Table 137: Troubleshooting the FX3-GEPR

Fault	Possible cause	Possible measures
The Flexi Soft gateway cannot be found in the Safety Designer.	The supply voltage of the FX0-GEPR is too low or missing. The Ethernet connection has been interrupted.	Switch on the voltage supply. Check the Ethernet cabling.
The FX3-GEPR is not supplying any data. PWR ●/☼ Green MS ●/☼ Green NS ☼ Green	The FX3-GEPR has been configured for transmitting data but no connection has been established yet. The Ethernet cabling is not correct or the configuration is incompatible with the communication partner which means that the connection is rejected.	Check the Ethernet cabling. Check the diagnostic entries of the FX3-GEPR for information about a rejected connection. Check and retransmit the configuration of both connection partners.
The FX3-GEPR is not supplying any data. NS ○ OFF	The Ethernet connection has been interrupted.	Check the Ethernet cabling.

Fault	Possible cause	Possible measures
The FX3-GEPR is not supplying any data. PWR ☼ Red (1 Hz) MS ☼ Red/green	Configuration required. The configuration has not yet been transferred in full.	Configure the FX3-GEPR and transfer the configuration to the device. Wait until the configuration has been fully transferred.
The FX3-GEPR is not supplying any data. PWR ● Green Link/Act ●/☼ Green MS ☼ Red (1 Hz) NS ● Red	There is a second device with an identical IP address on the network.	Correct the IP address of the FX3-GEPR and then switch the device off and back on again, or change the IP address of the other device.
The FX3-GEPR is not supplying any data. PWR ●/☼ Green Link/Act ●/☼ Green MS ● Green NS ☼ Red (1 Hz)	At least one EFI-pro or CIP connection has reached the maximum data age limit or has timed out.	Check the physical connection running to the connected devices. Check the data transmission rate and network delay parameters for the connections. Check the compatibility of the minimum RPIs that have been set for the devices. Use the configuration software to check the diagnostic messages.
The FX3-GEPR is not supplying any data. PWR ● Green Link/Act ●/☼ Green MS ● Green NS ● Green	The FX3-GEPR is in the “Run” state. The assemblies have not been configured correctly (e.g., assemblies configured incorrectly in the logic).	Configure the FX3-GEPR correctly and transfer the configuration to the device.
The FX3-GEPR is not supplying any data. PWR ☼ Green (1 Hz) MS ☼ Green (1 Hz)	The FX3-GEPR is in the “Stopped” state.	Start the main module (switch to the “Run” state).
The FX3-GEPR is in the “Serious error” state. PWR ☼ Red (2 Hz) Link/Act ●/☼ Green MS ● Red	Internal device error on the FX3-GEPR	Use the configuration software to check the diagnostic messages. Switch the voltage supply for the Flexi Soft system off and then back on again. If the fault persists, replace the gateway.
The FX3-GEPR/ Flexi Soft system is in the “Serious error” state. PWR ● Red MS ☼ Red (2 Hz)	The FX3-GEPR has not been connected correctly to the other Flexi Soft modules. The module connector is dirty or damaged. There is an internal serious error on another Flexi Soft module.	Plug in the FX0-GEPR 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 List of figures

9 List of tables

9 LIST OF TABLES

8018172/1KOF/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.