OPERATING INSTRUCTIONS

Flexi Soft in Flexi Soft Designer

Configuration software

SICK
Sensor Intelligence.

Described product

Flexi Soft in Flexi Soft Designer

Configuration software

Manufacturer

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

Legal information

Original document

This document is an original document of SICK AG.

Contents

1 About this document

1.1 Purpose of this document

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 of application and modification level

Product

These operating instructions apply to version V1.9.6 of the Flexi Soft Designer configuration software when used in conjunction with the following devices:

- FX3-CPU0 (firmware version V1.00.0–V4.05.0)
- FX3-CPU1 (firmware version V1.00.0–V4.05.0)
- FX3-CPU2 (firmware version V1.00.0–V4.05.0)
- FX3-CPU3 (firmware version V1.00.0–V4.05.0)
- FX3-MOC0 (firmware version V1.00.0–V1.10.0)
- FX3-MOC1 (firmware version V2.00.0–V3.01.0)
- FX3-ANA0 (firmware version V1.01.0–V2.00.0)

Document identification

Document part number:

- This document: 8012480
- Available language versions of this document: 8012998

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

1.3 Information depth

These operating instructions are intended to provide technical personnel working for the machine manufacturer or machine operator with instructions so that they can configure, operate and perform diagnostics on a Flexi Soft system with the Flexi Soft Designer software. They are only valid in conjunction with the “Flexi Soft Modular Safety Controller Hardware” operating instructions.

Please note that technical skills not covered by this document are also required when planning and using SICK protective devices.

General safety notes: see "On safety", page 16. Please make sure to read these instructions.

The official and legal regulations for operating the Flexi Soft modular safety controller must always be complied with.

1.4 Target groups

These operating instructions are intended for planning engineers, developers, and operators of plants and systems that are to be protected by means of a Flexi Soft modular safety controller. They are also intended for people who integrate the Flexi Soft safety controller into a machine, carry out its commissioning, or who are in charge of maintenance.

These operating instructions do not provide information on operating the machine, plant, or system in which the Flexi Soft safety controller is integrated. For information about this, refer to the operating instructions of the machine, plant, or system concerned.

1.5 Further information

www.sick.com

The following information is available via the Internet:

- Data sheets and application examples
- CAD files and dimensional drawings
- Certificates (such as the EU declaration of conformity)
- Guide for Safe Machinery Six steps to a safe machine
- Safety Designer (software for configuring safety solutions made by SICK AG)
- Flexi Soft Designer (software for configuring the Flexi Soft safety controller)

1.6 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.

ℹ **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.

Menus and commands

The names of software menus, submenus, options and commands, selection boxes, and windows are all emphasized. Example:

1. Go to the **File** menu and click on **Edit**.

The term “dangerous state”

The figures in this document always show the dangerous state (standard term) of the machine as movement of a machine part. In practice, there are various types of dangerous state:

- Machine movements
- Live electrical parts
- Visible and invisible beams
- A combination of multiple hazards

2 On safety

This chapter contains general safety information about the Flexi Soft modular safety controller.

More safety information about specific usage situations for the Flexi Soft modular safety controller is provided in the respective chapters.

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 Designer software is intended for configuring a modular Flexi Soft safety controller.

The product may be used in safety functions.

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 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 Version, compatibility, and features

There are different firmware versions and function packages (so-called “Steps”) for the Flexi Soft product family that permit realization of the different functions. This section provides an overview of which firmware version, which function package and/or which version of the Flexi Soft Designer configuration software or Safety Designer configuration software is needed to use a certain function or a certain device.

Table 2: Modules, firmware versions, and software versions you will need

	Necessary module with firmware from version	Available from Flexi Soft Designer	Available from Safety Designer
Function blocks and logic			
Offline simulation of logic	Unrestricted	V1.2.0	V1.6.x
Import and export of partial applications	Unrestricted	V1.3.0	V1.6.x
Automatic circuit diagrams	Unrestricted	V1.3.0	V1.6.x
Central tag name editor	Unrestricted	V1.3.0	V1.6.x
Documentation for function blocks of main modules in logic editor	Unrestricted	V1.3.0	N. a. [1)]
Matrix of input and output connections	Unrestricted	V1.3.0	V1.6.x
Invertible inputs for the function blocks AND, OR, RS Flip-Flop and Routing n:n	FX3-CPUx V2.00.0 (Step 2.xx)	V1.3.0	V1.6.x
Function block for ramp-down detection	FX3-CPUx V1.11.0 (Step 1.xx)	V1.3.0	V1.6.x
Function blocks for configurable switch-on delay and configurable switch-off delay	FX3-CPUx V2.00.0 (Step 2.xx)	V1.3.0	V1.6.x
Speed to Bool function block	FX3-MOC0 V1.10.0	V1.7.0	V1.6.x
Motion Status to Bool function block	FX3-MOC0 V1.10.0	V1.7.0	V1.6.x
Function block for type IIIA two-hand control: Configurable discrepancy time	FX3-CPUx V4.05.0	V1.9.6 SP1	V2023.01
Function block for type IIIC two-hand control: Configurable synchronization time	FX3-CPUx V4.05.0	V1.9.6 SP1	V2023.01
Verification possible even without identical hardware	FX3-CPUx V2.00.0 (Step 2.xx)	V1.3.0	V1.6.x
Status input data and status output data in logic	FX3-CPUx V2.00.0 (Step 2.xx) and FX3-XTIO, FX3-XTDI, or FX3-XTDS, each V2.00.0 (Step 2.xx)	V1.3.0	V1.6.x
Easy applications for FX3-MOC0	FX3-MOC0 V1.10.0	V1.7.1	N. a.
Special functions			
Two S3000 safety laser scanners on one EFI interface	FX3-CPU1 V1.00.0	V1.2.2	N. a.
Flexi Link	FX3-CPU1 V2.00.0 (Step 2.xx)	V1.3.0	N. a.
Flexi Loop	FX3-CPUx V3.00.0 (Step 3.xx) and FX3-XTIO, FX3-XTDI, or FX3-XTDS, each V3.00.0 (Step 3.xx)	V1.6.0	V1.8.0
Flexi Line	FX3-CPU3 V3.00.0 (Step 3.xx)	V1.6.0	N. a.
Automatic configuration of connected EFI-enabled safety sensors (automatic configuration recovery)	FX3-CPU2 V3.00.0 (Step 3.xx)	V1.5.0 (FX3-CPU2) V1.6.0 (FX3-CPU3)	N. a.
Deactivation of test signals Q1 to Q4 on the FX3-XTIO possible	FX3-XTIO V2.00.0 (Step 2.xx)	V1.3.0	V1.6.x

	Necessary module with firmware from version	Available from Flexi Soft Designer	Available from Safety Designer
Fast shut-off with bypass at FX3-XTIO	FX3-CPUx and FX3-XTIO, each V2.00.0 (Step 2.xx)	V1.3.0	V1.6.x
Multiple safety mats at FX3-XTIO/FX3-XTDI	FX3-XTIO or FX3-XTDI, each V1.13.0	V1.3.0	V1.6.x
Data recorder	FX3-CPUx V2.00.0 (Step 2.xx)	V1.5.0	V1.6.x
Extended cross-circuit detection time for the switching of increased capacitive loads at FX3-XTIO	FX3-XTIO V3.00.0 (Step 3.xx)	V1.6.0	V1.6.x
Configurable filter time for in/out filters and out/in filters at inputs I1 to I8 at FX3-XTIO/FX3-XTDI/FX3-XTDS	FX3-XTIO, FX3-XTDI, or FX3-XTDS, each V3.00.0 (Step 3.xx)	V1.6.0	V1.6.x
Optimization of logic execution time	FX3-CPUx V4.00.0 (Step 4.xx)	V1.7.1	V1.6.x
Automated download	No limitation	V1.9.1	n.a.
Checksums for logic pages and user-defined function blocks	No limitation	V1.9.3	n.a.
Licensed SICK safety systems	FX3-CPUx V4.00.0 (Step 4.xx)	V1.9.4	n.a.
Devices			
FX3-CPU0	No limitation	V1.2.0	V1.6.x
FX3-CPU1	No limitation	V1.2.0	N. a.
FX3-CPU2	No limitation	V1.2.0	N. a.
FX3-CPU3	No limitation	V1.2.0	N. a.
FX3-XTIO	No limitation	V1.2.0	V1.6.x
FX3-XTDI	No limitation	V1.2.0	V1.6.x
Gateways for PROFINET IO, Modbus® TCP and EtherNet/IP™	FX3-CPUx V1.11.0 (Step 1.xx)	V1.2.0	V1.6.x
CC-Link gateway	FX3-CPUx V1.11.0 (Step 1.xx)	V1.3.0	N. a.
CANopen gateway	FX3-CPUx V1.11.0 (Step 1.xx)	V1.3.0	V1.6.x
EtherCAT gateway	FX3-CPUx V2.00.0 (Step 2.xx)	V1.3.0	V1.6.x
EFI-pro gateway	FX3-CPU0 V4.00.0 (Step 4.xx)	N. a.	V1.6.x
SIM1000 FXG [2]	FX3-CPUx V1.11.0 (Step 1.xx)	V1.9.2	n.a.
Speed Monitor MOC3SA	Unrestricted	V1.3.0	V1.6.x
FX3-MOC0	FX3-CPUx V2.50.0	V1.5.0	N. a.
FX3-MOC1	FX3-CPUx V2.50.0	V1.8.0	V1.6.x
FX3-XTDS	Unrestricted	V1.6.0	V1.6.x
FX0-STIO	Unrestricted	V1.6.0	V1.6.x
FX3-ANA0	FX3-CPUx V4.00.0 (Step 4.xx)	V1.8.0	V1.7.0
Conformities			
RoHS conformity FX3-XTIO	FX3-XTIO V1.01.0	–	–

[1] N. a. = Not available

[2] All other modules as from market introduction.

[2] You can find information on this gateway in the SIM1000 FXG operating instructions.

NOTE

- More recent modules are backward compatible so that each module can be replaced by one with a higher firmware version.
- Flexi Soft Designer Version ≥ V1.4.0 can also be used to configure devices with a later version of the firmware, even if Flexi Soft Designer does not yet recognize the new firmware. In such cases, the user will only be able to access the function packages (Step 1.xx, Step 2.xx, Step 3.xx, or Step 4.xx) that are supported by the available version of Flexi Soft Designer.
- A corresponding new version of the configuration software is needed in order to use the full functional scope of modules with a later firmware version.
- The configuration software is not upwards-compatible. In other words, a project created with a more recent version of the configuration software cannot be opened with an older version.
- The function package (Step 1.xx, Step 2.xx, Step 3.xx, or Step 4.xx) must be selected in the hardware configuration menu of the configuration software. The availability of a desired function package in the configuration software can be found in the table.
- To use the Step N.xx function package, the relevant module must have a minimum firmware version of VN.00.0. If you try to transfer a configuration in a module with a lower firmware version, an error message is displayed.
- The hardware version of the Flexi Soft modules can be seen in the hardware configuration of the configuration software in online status or in the report if the system was previously online.
- You will find the **firmware version** of the Flexi Soft modules on the type label of the Flexi Soft modules in the firmware version field.
- The date of manufacture of a device can be found 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).
- The version of the configuration software can be found by selecting **Info** in the **Extras** menu.
- The latest version of the configuration software can be found on the Internet at www.sick.com.

4 Installation

4.1 System requirements

Recommended system configuration:

- Operating system: Windows 10, Windows 11
- Microsoft .NET Framework 3.5 SP1, Microsoft .NET Framework 4.6.2
- 1 GHz processor
- 2 GB RAM
- Screen resolution of 1024 × 768 pixels
- 400 MB of free memory on the hard drive

Flexi Soft Designer is a .NET Framework application. For information about the latest .NET Framework versions and details of the operating systems supported, please visit http://www.microsoft.com/.

Microsoft .NET Framework and any other necessary components can also be downloaded from http://www.microsoft.com/downloads/.

4.2 Installing and updating the software

Overview

The latest version of the Flexi Soft Designer software can be obtained on the Internet at www.sick.com.

Prerequisites

- Administrator rights

Installing and updating the software

Download the installation program, then run the installation program and follow the instructions.

To update the software, deinstall the old software version and install the new software version. The folder containing the project data is retained during the deinstallation.

Errors during installation

Table 3: Troubleshooting during installation

Error/error message	Possible cause	Troubleshooting measures
When you start Flexi Soft Designer, the following error message (or a similar one) appears: “Unable to locate DLL – Dynamic Link Library mscoree.dll could not be found in the specified path. Please set registry key HKLM\Software\Microsoft\.NETFramework\InstallRoot to point to the .NET Framework install location.”	Microsoft .NET Framework is not installed on the computer.	Install the appropriate version of Microsoft .NET Framework; if necessary, ask your system administrator. .NET Framework is available for download from Microsoft's web pages. **Note**: Install .NET Framework 3.5.

Complementary information

- To find out what version of Flexi Soft Designer is installed, select **Extras** and then look under **Info**.
- The Flexi Soft Designer installation will also install the CodeMeter software, which is used to license additional functions.

Further topics

- "Licensing and activating functions requiring a license", page 23

4.3 Unsupervised installation

Unsupervised installation

During unsupervised installation, the Flexi Soft Designer installation routine runs without additional input by the user. Unsupervised installation is configured with parameters in the command line.

The Inno Setup parameters are available for the Flexi Soft Designer. You can find the parameters and their description under: http://www.jrsoftware.org/ishelp

The following syntax applies for the command line:

```
FlexiSoftDesginer.exe [Parameter1] [Parameter2] [Parameter n]
```

Example

For unsupervised installation of Flexi Soft Designer version 1.9 without calling up message boxes and restarting the system, the following invocation is required in the command line:

```
FlexiSoftDesigner1_9_setup.exe/SUPPRESSMSGBOXES /VERYSILENT/NORESTART
```

4.4 Licensing of safety systems

Overview

You can find information on adding and licensing SICK safety systems in the respective operating instructions of the safety system.

Important information

! **NOTICE**
When deleting the Flexi Soft system plug, the device licenses for safety systems can also be deleted.

- ► Pay attention to the Flexi Soft Designer notes.
- ► If necessary, restore device licenses:
  - o Activate device licenses with the same computer in the device with which they were deleted.
  - o Read the configuration before deletion.

4.5 Licensing and activating functions requiring a license

Overview

Contact SICK Sales to license a function requiring a license such as the **Automatic Download** function. You can then purchase a Ticket ID (licensing code) that you can use to activate the desired function.

Procedure

1. In the **Extras** menu, select the **Software licenses** command.
2. In the **Software licenses** dialog box, click on the **Activate** button.
3. Enter the Ticket ID and confirm with **OK.**

✓ The CodeMeter License Central WebDepot opens in your browser.

4. Select the binding for the licenses:

   - Binding to a PC
   - Binding to a dongle
5. Select the desired licenses.

   ℹ **NOTE |** Note the specified number of licenses. If you have purchased a license package and only want to activate specific licenses in it, you need to first distribute the licenses.
6. Activate the licenses by clicking the **Now activate the selected licenses** button.
7. Click on **Close** to apply the changes and close the **Software licenses** window.

Complementary information

- It is also possible to activate a license offline. To do so, following the **File-based license transfer** instructions in the WebDepot.
- Activated licenses can be returned in CodeMeter License Central WebDepot. To do so, enter the Ticket ID and then return the license by clicking the **Re-Host Licenses** button. The license will then be available again.

Further topics

- "Distributing licenses", page 24
- "Ordering information for Flexi Soft Designer functions requiring a license", page 526

4.6 Distributing licenses

Overview

If you have purchased a license package and only want to activate specific licenses in it, you need to first distribute the licenses.

Prerequisites

- A license package has been purchased and a Ticket ID is available.

Procedure

1. Opening the **CodeMeter License Central WebDepot:** https://license.sick.com/
2. Enter the Ticket ID and click on **Next.**

✓ The **My Licenses** tab is displayed.

3. Select **Distribute licenses.**
4. Follow the instructions for distributing the licenses in WebDepot.

Further topics

- "Licensing and activating functions requiring a license", page 23

5 Connecting the computer to the Flexi Soft system

5.1 Initial steps for setting up a connection

Interfaces of the safety controller for connecting to a computer:

- RS-232
- USB (requires FX3-CPU3)
- TCP/IP (via Ethernet gateway)

Flexi Soft Designer uses port 9000 when connecting to a computer.

The operating instructions “Flexi Soft Gateways in Flexi Soft Designer Configuration Software” provide information on establishing a connection via TCP/IP.

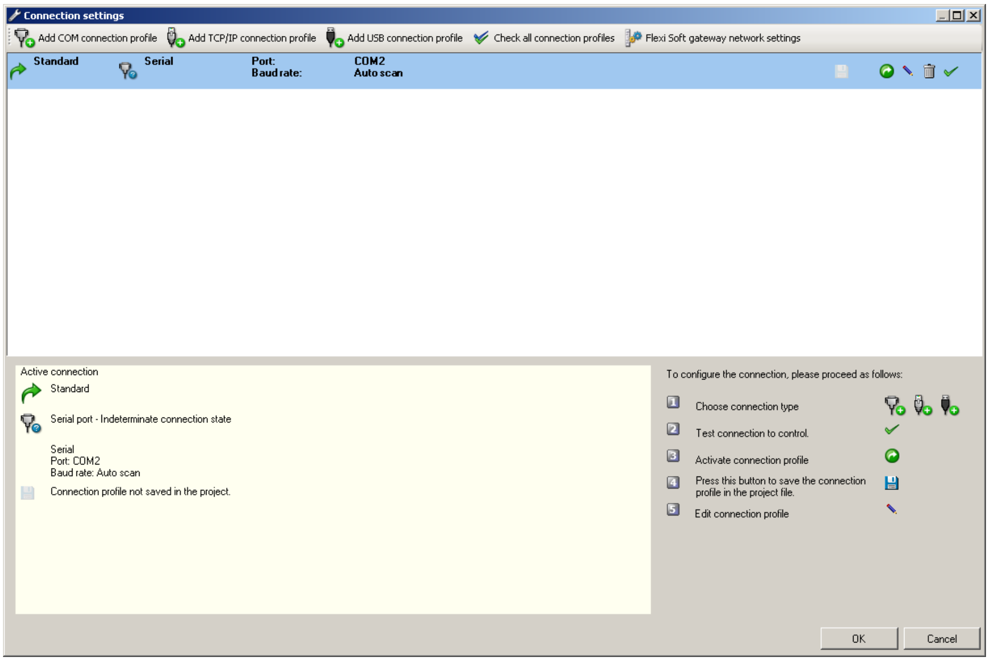
5.1.1 Connecting the computer to the Flexi Soft system via RS-232

► Connect the computer with the RS-232 interface of the main module either via the serial COM interface or using an RS-232/USB adapter.

► Switch on the Flexi Soft system.

► Open the Flexi Soft Designer configuration software that has been installed on the computer.

► On the Flexi Soft Designer start screen, click **Adjust interface parameters** to open the **Communication settings** dialog box. The connection profiles currently installed on the computer are displayed. The active connection profile is marked by a green arrow on the left side of the box.


Connection settings
Add COM connection profile
Add TCP/IP connection profile
Add USB connection profile
Check all connection profiles
Flexi Soft gateway network settings
OK
Cancel

Standard | Serial | Port: COM2 | Baud rate: Auto scan

Active connection
Standard
Serial port - Indeterminate connection state
Serial
Port: COM2
Baud rate: Auto scan
Connection profile not saved in the project.

To configure the connection, please proceed as follows:
1 Choose connection type
2 Test connection to control.
3 Activate connection profile
4 Press this button to save the connection profile in the project file.
5 Edit connection profile

Figure 1: Communication settings dialog box

► If no COM connection profile is displayed, click the **Add COM connection profile** button to create a new profile:

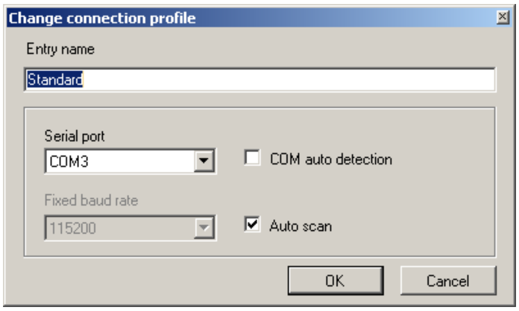
  ► Enter a name for the new COM connection profile in the **Create new connection profile** dialog box.

  ► Select the serial connection for the new connection profile or mark the **COM auto detection** box.

  ► Select a fixed data transmission rate or mark the **Auto scan** box.

  ► Click on **OK**. The dialog box closes and the new connection profile appears in the selection list.

- ► To change the settings of a COM connection profile, select it and click on the pencil symbol on the right. The following dialog box opens:



Entry name: Standard

Serial port: COM3

☐ COM auto detection

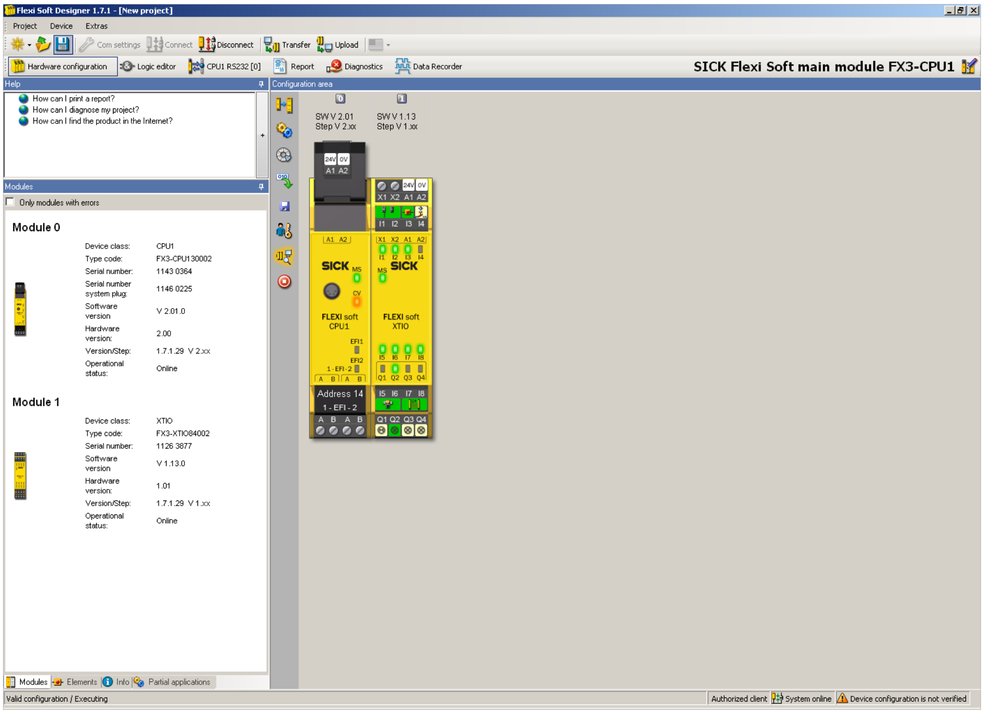
Fixed baud rate: 115200

☑ Auto scan

Figure 2: Change connection profile dialog box

- ► Change the settings as needed and click on **OK**. The **Change connection profile** dialog box closes.
- ► Select the required connection profile and click on the green icon **Activate connection profile**. Flexi Soft Designer will now start using this connection profile.
  Or, click on * behind the Save symbol. The connection profile is automatically activated when loading the configuration.
- ► Click on **OK**. The **Communication settings** dialog box closes.
- ► Click **Connect to physical device**. Flexi Soft Designer now starts searching for connected Flexi Soft devices and loads the hardware configuration into the **Hardware configuration** dialog box. As soon as all the modules have been identified, the software asks whether you want the configuration to be loaded.
- ► Click **Yes** to load the configuration.

Below is an example of a possible Hardware configuration view:



How can I print a report?
How can I diagnose my project?
How can I find the product in the Internet?

☐ Only modules with errors

Module 0

Device class:	CPU1
Type code:	FX3-CPU130002
Serial number:	1143 0364
Serial number system plug:	1146 0225
Software version	V 2.01.0
Hardware version:	2.00
Version/Step:	1.7.1.29 V 2.xx
Operational status:	Online

Module 1

Device class:	XTIO
Type code:	FX3-XTIO84002
Serial number:	1126 3877
Software version	V 1.13.0
Hardware version:	1.01
Version/Step:	1.7.1.29 V 1.xx
Operational status:	Online

Valid configuration / Executing | Authorized client | System online | Device configuration is not verified

Figure 3: Example for hardware configuration view

- ► To change the configuration for the Flexi Soft modules, click **Disconnect** to switch to offline mode. Alternatively, use the **Edit** button to change to edit mode to make minor changes without having to disconnect every time.

NOTE

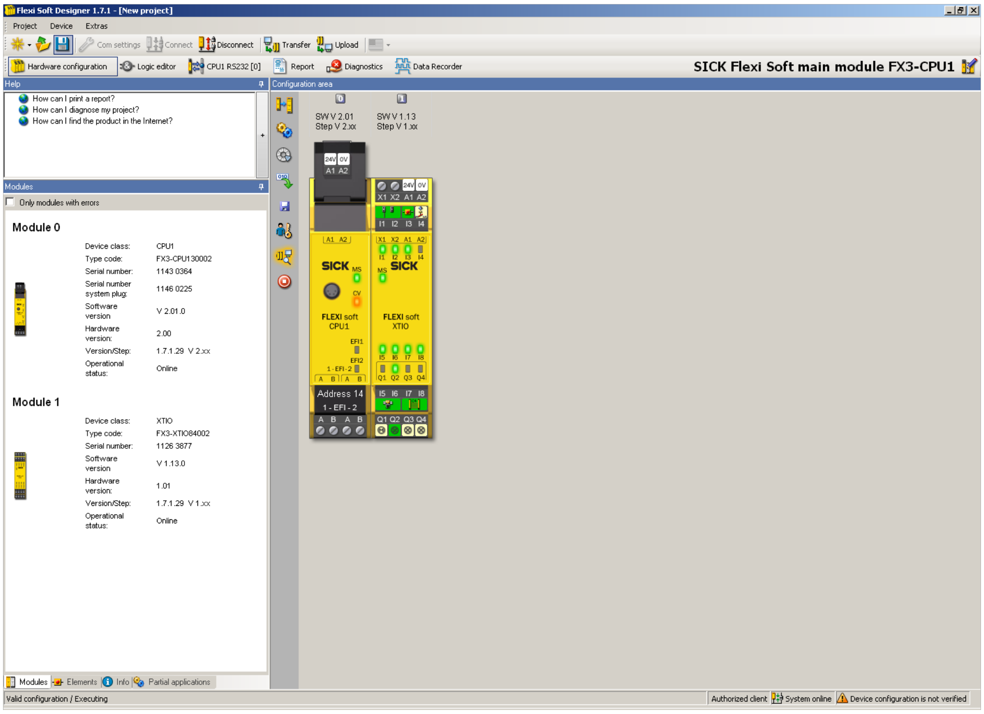
Flexi Soft Designer cannot be used to configure and verify devices that are connected to the Flexi Soft system, even if these devices can be addressed via the RS-232 interface or the USB interface of a Flexi Soft main module. The connected devices have their own configuration and verification mechanisms.

There is one exception to this rule: EFI-enabled devices that can be connected to an FX3-CPU1 Flexi Soft main module or a later version of this module (EFI elements in element window). These devices can either be configured directly in Flexi Soft Designer by double-clicking the relevant symbols or they can be configured and verified locally via their own RS-232 interface. This involves using the SICK Configuration & Diagnostic Software (CDS).

5.1.2 Connecting the computer to the Flexi Soft system via USB

- ► Connect the computer to the USB interface of the main module.
- ► Switch on the Flexi Soft system.
- ► Open the Flexi Soft Designer configuration software that has been installed on the computer.
- ► On the Flexi Soft Designer start screen, click **Adjust interface parameters** to open the **Communication settings** dialog box. The connection profiles currently installed on the computer are displayed. The active connection profile is marked by a green arrow on the left side of the box.
- ► If no USB profile is displayed, click the **Add USB connection profile** button to create a new profile:
  - ► Select the Flexi Soft main module displayed under **Connected Devices** in the **Create new connection profile** dialog box.
  - ► Enter a name for the new USB connection profile and click on **OK**. The dialog box closes and the new connection profile appears in the selection list.
- ► Select the required connection profile and click on the green icon **Activate connection profile.** Flexi Soft Designer will now start using this connection profile.
  Or, click on * behind the Save symbol. The connection profile is automatically activated when loading the configuration.
- ► Click on **OK**. The **Communication settings** dialog box closes.
- ► Click **Connect to physical device.** Flexi Soft Designer now starts searching for connected Flexi Soft devices and loads the hardware configuration into the **Hardware configuration** dialog box. As soon as all the modules have been identified, the software asks whether you want the configuration to be loaded.
- ► Click **Yes** to load the configuration.

Below is an example of a possible Hardware configuration view:



Module 0

Device class: CPU1
Type code: FX3-CPU130002
Serial number: 1143 0364
Serial number system plug: 1146 0225
Software version: V 2.01.0
Hardware version: 2.00
Version/Step: 1.7.1.29 V 2.xx
Operational status: Online

Module 1

Device class: XTIO
Type code: FX3-XTIO84002
Serial number: 1126 3877
Software version: V 1.13.0
Hardware version: 1.01
Version/Step: 1.7.1.29 V 1.xx
Operational status: Online

Valid configuration / Executing

Authorized client | System online | Device configuration is not verified

Figure 4: Example for hardware configuration view

► If you want to change the configuration for the Flexi Soft modules, click **Disconnect** to switch to offline mode. Alternatively, click the **Edit** button to switch to editing mode, allowing you to make minor changes without having to disconnect every time.

NOTE

Flexi Soft Designer cannot be used to configure and verify devices that are connected to the Flexi Soft system, even if these devices can be addressed via the RS-232 interface or the USB interface of a Flexi Soft main module. The connected devices have their own configuration and verification mechanisms.

There is one exception to this rule: EFI-enabled devices that can be connected to an FX3-CPU1 Flexi Soft main module or a later version of this module (EFI elements in element window). These devices can either be configured directly in Flexi Soft Designer by double-clicking the relevant icons or they can be configured and verified locally via their own RS-232 interface. This involves using the SICK Configuration & Diagnostic Software (CDS).

5.1.3 Status and background color

The background color in Flexi Soft Designer indicates the current status of the software as per the table below:

Table 4: Meaning of the background color

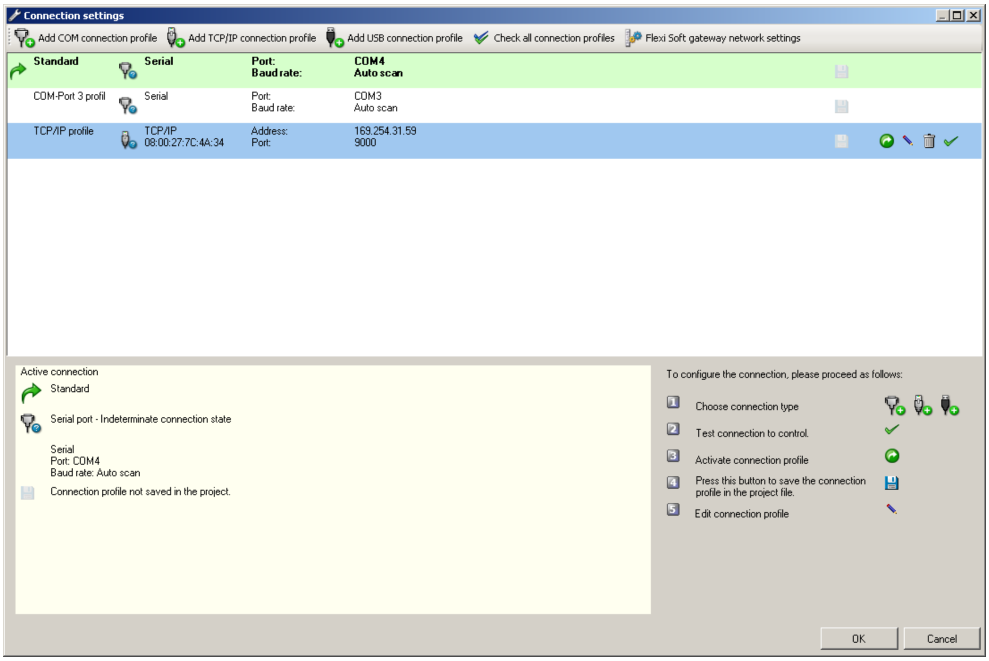
Background color	Status	Configuration status in Flexi Soft Designer
Light yellow	Offline	Any
Blue	Online	Invalid and/or different from device configuration
Gray	Online	Valid and identical to device configuration

5.2 Editing communication settings

You can use the **COM settings** command to create, edit, and delete connection profiles.

The software must be in offline mode to edit the communication settings.

- ► If Flexi Soft Designer is currently in online mode, click the **Disconnect** button to switch to offline mode.
- ► Click on **COM settings**. The **Communication settings** dialog box opens:


Connection settings
Add COM connection profile
Add TCP/IP connection profile
Add USB connection profile
Check all connection profiles
Flexi Soft gateway network settings
OK
Cancel

Standard	**Serial**	**Port:** **Baud rate:**	**COM4** **Auto scan**
COM-Port 3 profil	Serial	Port: Baud rate:	COM3 Auto scan
TCP/IP profile	TCP/IP 08:00:27:7C:4A:34	Address: Port:	169.254.31.59 9000

Active connection

Standard

Serial port - Indeterminate connection state

Serial
Port: COM4
Baud rate: Auto scan

Connection profile not saved in the project.

To configure the connection, please proceed as follows:

1. Choose connection type
2. Test connection to control.
3. Activate connection profile
4. Press this button to save the connection profile in the project file.
5. Edit connection profile

Figure 5: Communication settings dialog box – existing connection profiles

All existing connection profiles are displayed here. The currently active connection profile is highlighted in light green with the text in bold font, and marked by an arrow on the left-hand side. The connection profile selected for editing is highlighted in blue.

An overview of the current settings can be found in the bottom section of the dialog box.

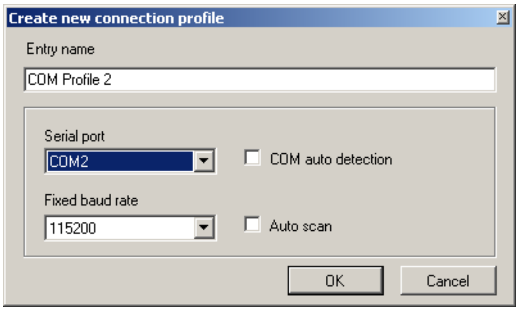
Below is a description of what the buttons for editing the connection profiles mean:

Table 5: Buttons for editing the connection profiles in the Communication settings box

Button	Meaning
💾	Save connection profile in current project file
*	Automatically activate the saved connection profile when loading the configuration.
➦	Activate connection profile
✎	Edit connection profile
🗑	Remove connection profile
✓	Test connection to Flexi Soft system

5.2.1 Adding a COM connection profile (serial interface)

- ► Click the **Add COM connection profile** button. The **Create new connection profile** dialog box opens.



Create new connection profile

Entry name: COM Profile 2

Serial port: COM2

☐ COM auto detection

Fixed baud rate: 115200

☐ Auto scan

Figure 6: Create new connection profile dialog box (serial interface)

- ► Enter a name for the new connection profile.
- ► Select the serial connection for the new connection profile or mark the **COM auto detection** box.
- ► Select a fixed data transmission rate or mark the **Auto scan** box.
- ► Click on **OK**. The dialog box closes and the new connection profile appears in the selection list.
- ► Select the required connection profile and click on the green icon **Activate connection profile.** Flexi Soft Designer will now start using this connection profile.

5.2.2 Add USB connection profile

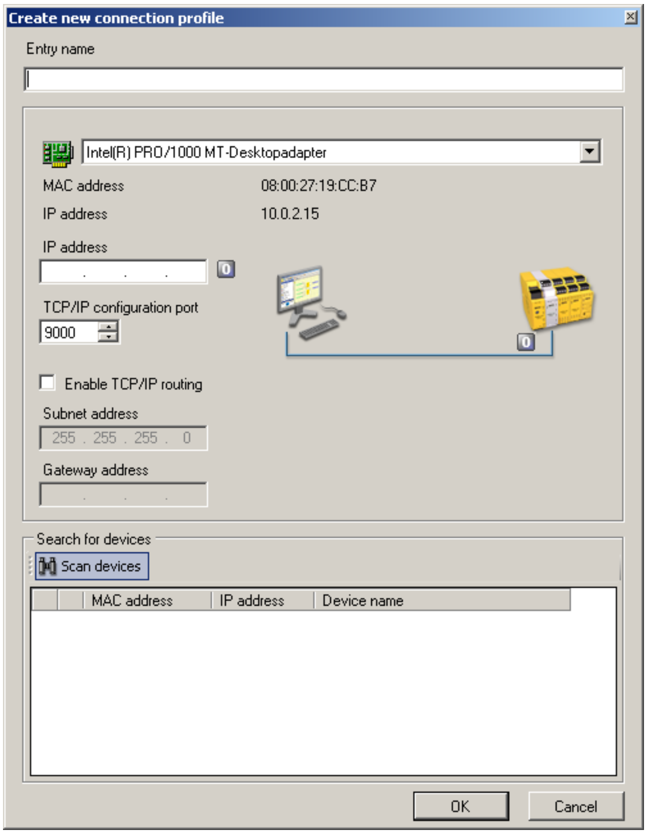
- ► Click the **Add USB connection profile** button. The **Create new connection profile** dialog box opens.
- ► Enter a name for the new connection profile.
- ► Select the main module for the new connection profile.
- ► Click on **OK**. The dialog box closes and the new connection profile appears in the selection list.
- ► Select the required connection profile and click on the green icon **Activate connection profile.** Flexi Soft Designer will now start using this connection profile.

NOTE
With connections made via the USB interface, the Flexi Soft system will only be recognized correctly when a supply voltage is present.

5.2.3 Add TCP/IP connection profile

NOTE
Before you can set up a TCP/IP connection profile, an Ethernet based gateway (e.g., FX0-GENT, FX0-GPNT, or FX0-GMOD) must be integrated into the Flexi Soft system and configured with a valid IP address. For detailed information on configuring gateways, please refer to the operating instructions titled “Flexi Soft Gateways in the Flexi Soft Designer Configuration Software” (SICK part number 8012483).

- ► Click the **Add TCP/IP connection profile** button. The **Create new connection profile** dialog box opens.



Create new connection profile

Entry name

Intel(R) PRO/1000 MT-Desktopadapter

MAC address 08:00:27:19:CC:B7

IP address 10.0.2.15

IP address

TCP/IP configuration port 9000

□ Enable TCP/IP routing

Subnet address 255 . 255 . 255 . 0

Gateway address

Search for devices

Scan devices

		MAC address	IP address	Device name

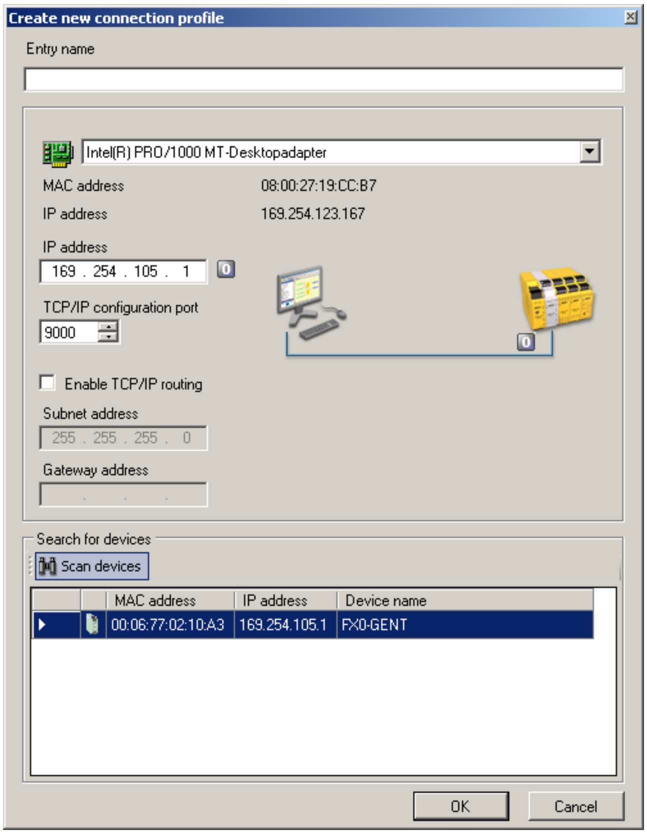
OK Cancel

Figure 7: Create new connection profile dialog box (TCP/IP)

► Select the network card that connects your computer to the relevant network.

NOTE
If you wish to establish an Ethernet connection via VPN, you must also select the relevant network card.

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



Create new connection profile

Entry name

Intel(R) PRO/1000 MT-Desktopadapter

MAC address 08:00:27:19:CC:B7

IP address 169.254.123.167

IP address

169 . 254 . 105 . 1

TCP/IP configuration port

9000

□ Enable TCP/IP routing

Subnet address

255 . 255 . 255 . 0

Gateway address

Search for devices

Scan devices

	MAC address	IP address	Device name
▶	00:06:77:02:10:A3	169.254.105.1	FX0-GENT

OK Cancel

Figure 8: Create new connection profile dialog box – list of detected gateways

- ► Click the required gateway. The IP address of the device is transferred to the **IP address** field.
- ► Enter a name for the new connection profile.
- ► Click on **OK**. The dialog box closes and the new connection profile appears in the selection list.
- ► Select the required connection profile and click on the green icon **Activate connection profile**. Flexi Soft Designer will now start using this connection profile.

5.2.4 Checking the connection profile

- ► Click the green check mark to the right of the connection profile that you want to check.
- ► To check all the connection profiles, click the **Check all connection profiles** button.

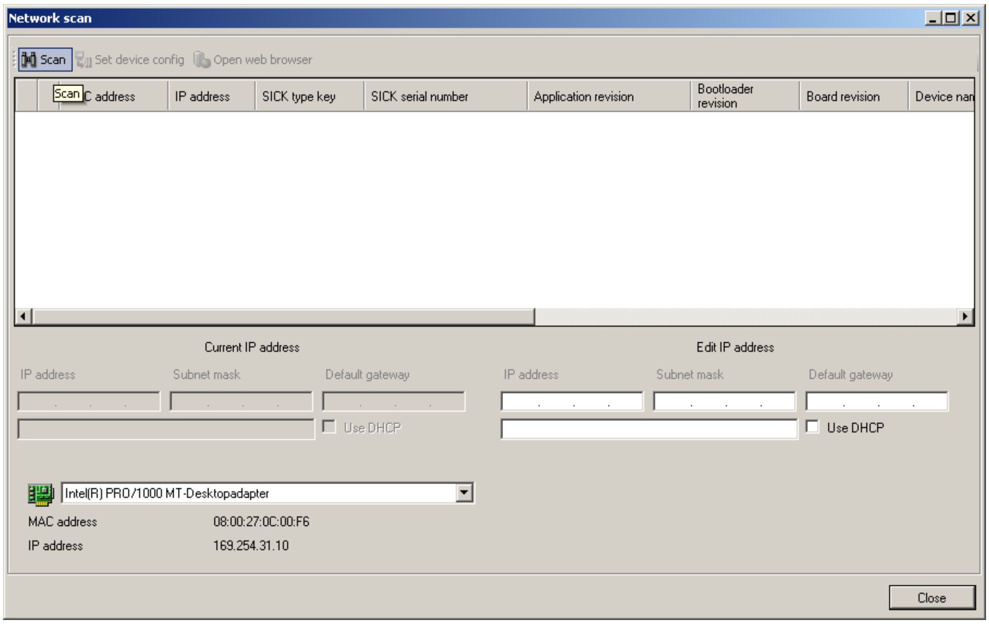
Flexi Soft Designer checks the communication settings and highlights any faulty connection profiles.

Table 6: Symbols for indicating the connection profile status

Connection profile	Profile not checked	Profile OK	Profile faulty
COM (serial)	?	✓	✗
USB	?	✓	✗
TCP/IP	?	✓	✗

5.2.5 Changing the network settings for a Flexi Soft gateway

- ► Click the **Flexi Soft gateway network settings** button. The **Search for devices** dialog box opens.
- ► If necessary, select the correct network adapter from the drop-down list below.
- ► Click the **Search for devices** button. The software scans your network for connected gateways and displays any devices that it finds in the list below.



Intel(R) PRO/1000 MT-Desktopadapter

MAC address 08:00:27:0C:00:F6

IP address 169.254.31.10

Figure 9: Search for devices dialog box – list of detected gateways

- ► Click the required gateway.
- ► Enter the new settings in the **Edit IP address** area.
- ► Click the **Set device config** button to transfer the new settings to the gateway.

NOTE

If a gateway from the Flexi Classic product family (i.e., UE410-EN1) is detected on the network by Flexi Soft Designer, this is also displayed in the list. These gateways feature an internal web server and can be addressed using the **Open web browser** button.

5.3 Establishing a connection to the Flexi Soft system

WARNING

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

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

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

- ► Click the **Connect** button. Flexi Soft Designer attempts to establish a connection to the Flexi Soft system using the currently active connection profile.

If a connection can be established, the software switches to online mode and the following actions can be performed, depending on the user group:

- Log in (see "User groups in Flexi Soft Designer", page 34)
- Transfer configuration to device, load configuration from device, or verify configuration (see "Transferring the configuration", page 493)
- Start or stop device (see "Changing the device status", page 516)
- Start force mode (see "Force mode", page 240)

5.4 Detecting a project

The **Detect project** command performs the same function as the **Connect to physical device** command that is available when Flexi Soft Designer is started up.

► In the **Device** menu, select the **Detect project** command. This closes the project that is currently open.
► Flexi Soft Designer now starts searching for connected Flexi Soft devices and loads the hardware configuration into the **Hardware configuration** dialog box. As soon as all the modules have been identified, the software asks whether to load the configuration.
► To load the configuration, click **Yes**.

5.5 User groups in Flexi Soft Designer

When Flexi Soft Designer is connected to the devices in a project (i.e., if it is in online mode), it is possible to switch between the Flexi Soft Designer user groups. The various user groups have different authorizations:

Table 7: User group authorizations

User group	Password	Authorization
Machine operator	None	Permitted to create and edit configurations offline Permitted to connect to the system (for diagnostics only) Not permitted to transfer any configurations Not permitted to verify any configurations
Maintenance personnel	Default setting: None (i.e., no login possible) Can be changed by authorized clients	Permitted to create and edit configurations offline Permitted to transfer verified configurations Permitted to connect to the system (for the purposes of transfer and diagnostics) Permitted to start and stop the system Not permitted to verify any configurations
Authorized client	Default setting: SICKSAFE Can be changed by authorized clients	Permitted to create and edit configurations offline Permitted to transfer verified and unverified configurations Permitted to connect to the system (for the purposes of transfer and diagnostics) Permitted to start and stop the system Permitted to use forcing Permitted to verify configurations Permitted to edit passwords

WARNING
Unauthorized configuration changes

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

► Before leaving the computer, switch to the **Machine operator** user group so that unauthorized persons cannot transfer configurations to the devices.

NOTE

- When logging in to a main module for the first time as an **Authorized client** using a system plug in delivery condition (new or erased system plug), you will be requested to assign a new password.
- The password protection applies to the configuration for the current devices.
- The password is stored in the system plug. The password is retained if the main module is replaced.

Changing user group

► In the **Hardware configuration** view, click the **Login** button on the left of the configuration area. To do this, the Flexi Soft Designer must be connected to the Flexi Soft system (i.e., be in online mode).

► In the **Login** dialog box, select the required user group, enter the password, and click **Log in**.

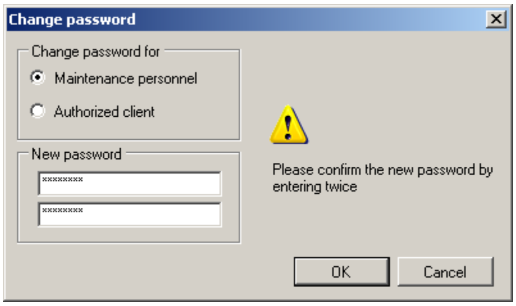
Setting or changing the password for a user group

► Switching to online mode.

► Open the **Hardware configuration** in Flexi Soft Designer.

► In the context menu of the main module, select the **Change password...** command. If no one in the authorized client user group is logged in, the login dialog for this user group is now displayed.

► In the **Change password** dialog box, select the user group whose password you want to change, enter the new password, then click **OK** to confirm.


Change password

Change password for
● Maintenance personnel
○ Authorized client

New password
xxxxxxxx
xxxxxxxx

Please confirm the new password by entering twice

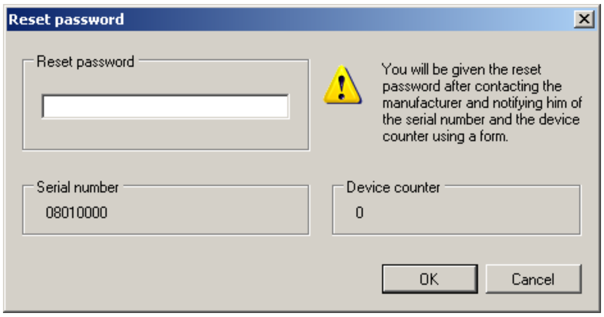
Figure 10: Change password dialog box

Resetting the password

► Switching to online mode.

► Open the **Hardware configuration** in Flexi Soft Designer.

► In the context menu of the main module, select the **Reset password...** command. The **Reset password** dialog box opens.



You will be given the reset password after contacting the manufacturer and notifying him of the serial number and the device counter using a form.

Serial number
08010000

Device counter
0

Figure 11: Reset password dialog box

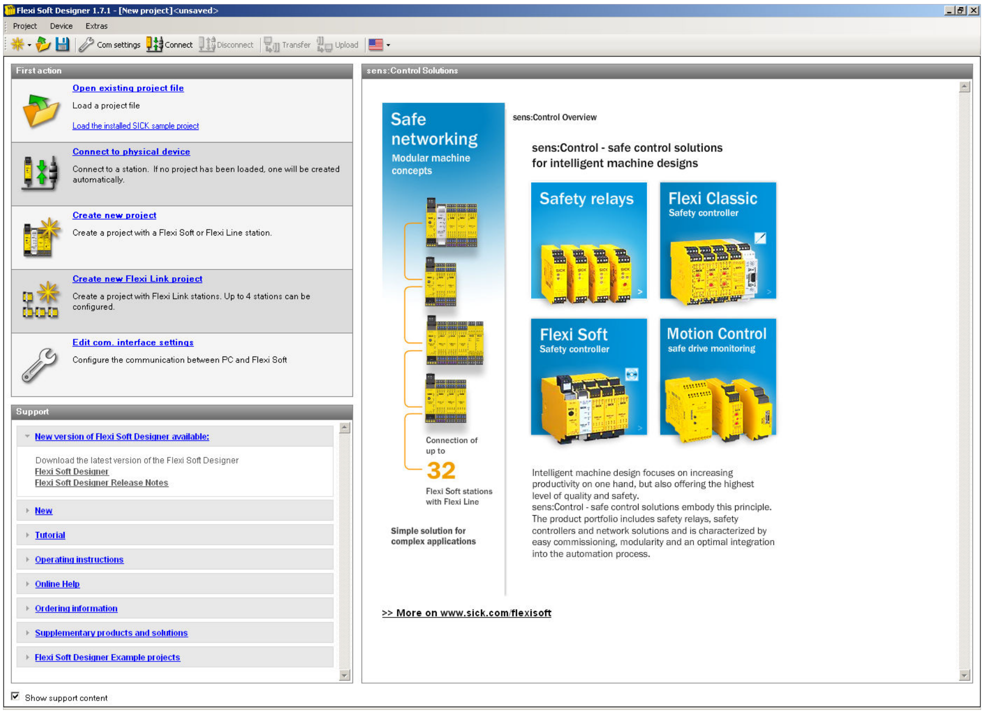
- ► Make a note of the serial number displayed for the system plug and the number displayed under “Device counter”; then contact SICK Support. SICK Support will provide a reset password that can be used to reset all passwords to their factory settings.
- ► Enter the reset password in the **Reset password** dialog box and click **OK** to confirm.

6 The user interface

6.1 Home screen

When you start the software, a home screen appears. Here, users can get started by choosing between the following actions:

- Open an existing project file or an example project
- Establish a connection to a physically connected device
- Create a new project
- Create a new Flexi Link project
- Adjust the parameters of the serial interface


Flexi Soft Designer 1.7.1 - [New project]<unsaved>
Project
Device
Extras
Com settings
Connect
Disconnect
Transfer
Upload
First action
Open existing project file
Load the installed SICK sample project
Connect to physical device
Create new project
Create new Flexi Link project
Edit com. interface settings
Support
New version of Flexi Soft Designer available:
Flexi Soft Designer
Flexi Soft Designer Release Notes
New
Tutorial
Operating instructions
Online Help
Ordering information
Supplementary products and solutions
Flexi Soft Designer Example projects
☑ Show support content
sens:Control Solutions
Safe networking
Modular machine concepts
Connection of up to
32
Flexi Soft stations with Flexi Line
Simple solution for complex applications
sens:Control Overview
Safety relays
Flexi Classic
Safety controller
Flexi Soft
Safety controller
Motion Control
safe drive monitoring
>> More on www.sick.com/flexisoft

Load a project file

Connect to a station. If no project has been loaded, one will be created automatically.

Create a project with a Flexi Soft or Flexi Line station.

Create a project with Flexi Link stations. Up to 4 stations can be configured.

Configure the communication between PC and Flexi Soft

Download the latest version of the Flexi Soft Designer

sens:Control - safe control solutions for intelligent machine designs

Intelligent machine design focuses on increasing productivity on one hand, but also offering the highest level of quality and safety.
sens:Control - safe control solutions embody this principle. The product portfolio includes safety relays, safety controllers and network solutions and is characterized by easy commissioning, modularity and an optimal integration into the automation process.

Figure 12: Home screen with choice of available actions

6.2 Language selection

► Click the flag icon on the toolbar and selected the desired language.

6.3 Standard views

Flexi Soft Designer offers the following views, which can be accessed via the buttons underneath the toolbar.

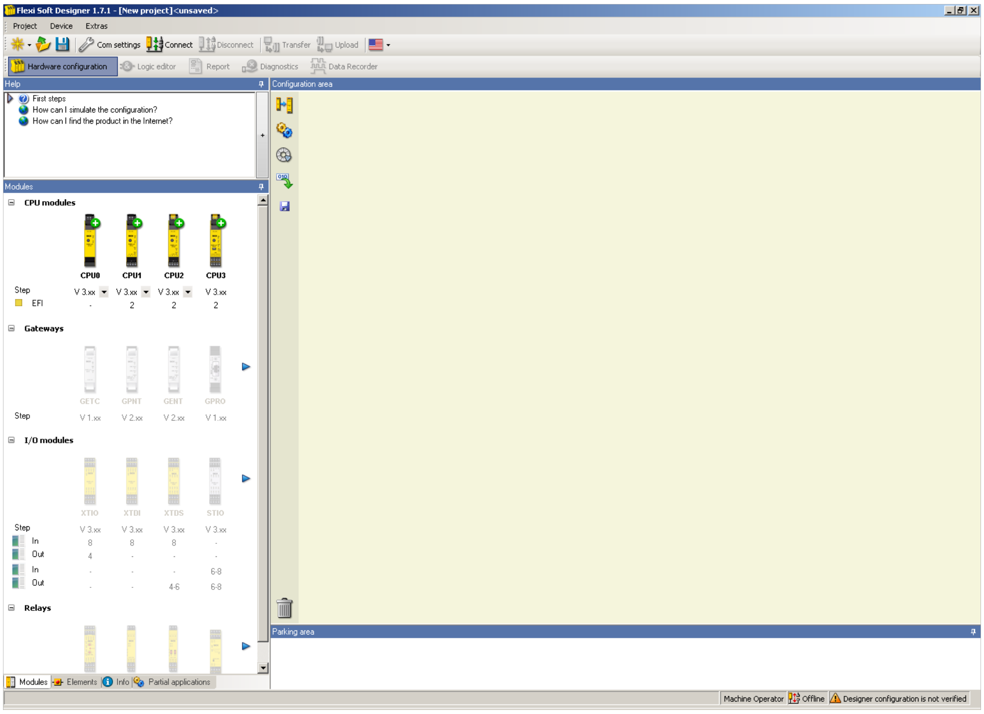


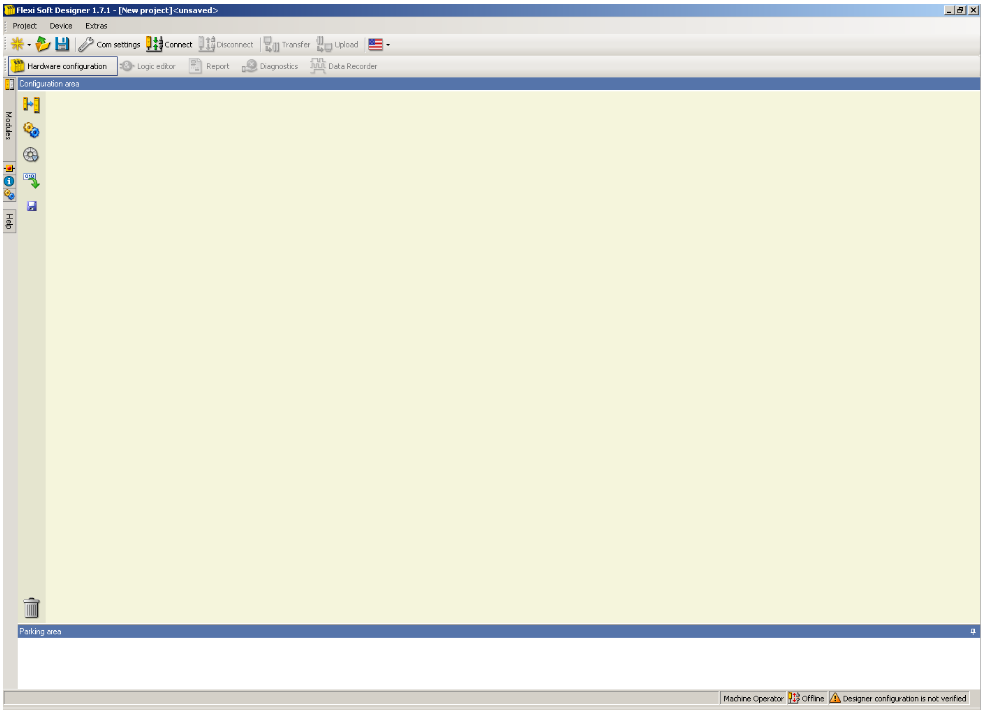
Figure 13: Buttons for selecting various views

- The **Hardware configuration** view allows you to define the layout of a Flexi Soft system comprising various hardware modules and to configure the inputs and outputs of the connected elements.
- In the **Logic editor** view, you can use logical function blocks and application-specific function blocks to configure the function logic. This view only becomes available once a main module has been selected in the Hardware configuration view.
- Depending on the configuration, the **Interfaces** view may or may not be available. This view is used to configure data exchange via the various network interfaces, such as gateways, RS-232 communication, Flexi Loop, Flexi Line, etc. The type of network used determines which configuration options are available. These options are described in the relevant chapters of these operating instructions and in the operating instructions titled “Flexi Soft Gateways in the Flexi Soft Designer Configuration Software”.
- The **Report** view provides comprehensive information about the currently loaded project and all settings, including the logic programming and wiring diagrams. The report can be saved as a PDF or printed out. You can customize the contents and scope of the report.
- The **Diagnostics** view shows a history of the error messages saved for a connected Flexi Soft system.
- You can use the **Data Recorder** view to record and visualize input and output signals for a Flexi Soft system.

6.4 Arranging windows

Each view consists of several subwindows that you can arrange however you like:

- ► To change the height, width and position of the subwindow, use the mouse to move the frame or title bar of the subwindow.
- ► Click the **Auto hide** button (thumbtack icon) on the right-hand side of the title bar to turn a subwindow into a flyout window. The flyout menu will then be located on the left edge of the Flexi Soft Designer window (see figure),
- ► Click the thumbtack icon in the flyout window to turn the flyout window back into a normal subwindow.



Designer configuration is not verified

Figure 14: Flyout menu for flyout window

6.5 Hardware configuration

The **Hardware configuration** window consists of the following elements:

- **Modules** selection window: This shows all the Flexi Soft hardware modules that can be combined to create a Flexi Soft safety controller. Any modules that are not available for selection with the current configuration are grayed out. Any modules that can be added to the current configuration are identified by a green “+” symbol.
  Under each module, you can see how many EFI connections are available for the module or how many safe and non-safe inputs and outputs are associated with the module. Safe inputs and outputs are identified by means of icons with a yellow background; non-safe inputs and outputs by means of icons with a gray background.
  The Step version or function package can be selected for each module. The selected function package determines the minimum firmware version that has to be used. Step 2.xx requires a firmware version ≥ V2.00.0 (see "Version, compatibility, and features", page 19).
- **Elements** selection window: This lists all the devices (e.g., sensors, actuators, encoders) that can be connected to the inputs and outputs of a Flexi Soft safety controller. The devices can be configured and renamed. In addition, you can create and save user-defined devices.
  EFI elements can be connected by dragging them onto either of the two EFI interfaces of an EFI-enabled main module (see "Connecting EFI-enabled devices", page 51).

In addition, the **Elements** selection window also contains the connections for Flexi Line and Flexi Loop (see "Flexi Line", page 481 or the operating instructions “Flexi Loop Safe Series Connection Hardware”)

- **Info** selection window: This displays general information about the current configuration or – if Flexi Soft Designer is connected to the Flexi Soft main module – the current status of the main and connected modules.
- **Partial applications** selection window: see "Exporting and importing partial application", page 53
- **Safety system** selection window: SICK safety systems can be added here, then information about the safety system can be displayed. You can find information on adding and licensing SICK safety systems in the respective operating instructions of the safety system.
- **Configuration area:** The entire hardware configuration for the Flexi Soft safety controller and connected devices is created here and displayed in graphical format. The individual modules and connected devices can be named, given a tag name, and parameterized using the context menus for the devices. In addition, the context menu of the main module offers various other options, such as the ability to export or import a configuration (hardware configuration and logic) and – if Flexi Soft Designer is connected to the system – to change and reset the password, or perform a system reset via the software.
  Buttons for the following functions are available to the left of the arranged modules: **Zoom in/out**, **Settings**, **Edit tag names**, and **Export tag names to Pro-face GP-Pro EX**. Once a connection to a Flexi Soft main module has been established, further functions become available: **Log in** (switch user group), **Verify** (load and compare configuration), and **Start** or **Stop** main module.
- **Parking area:** This area allows users to create a collection of devices for a specific application and store them here temporarily.

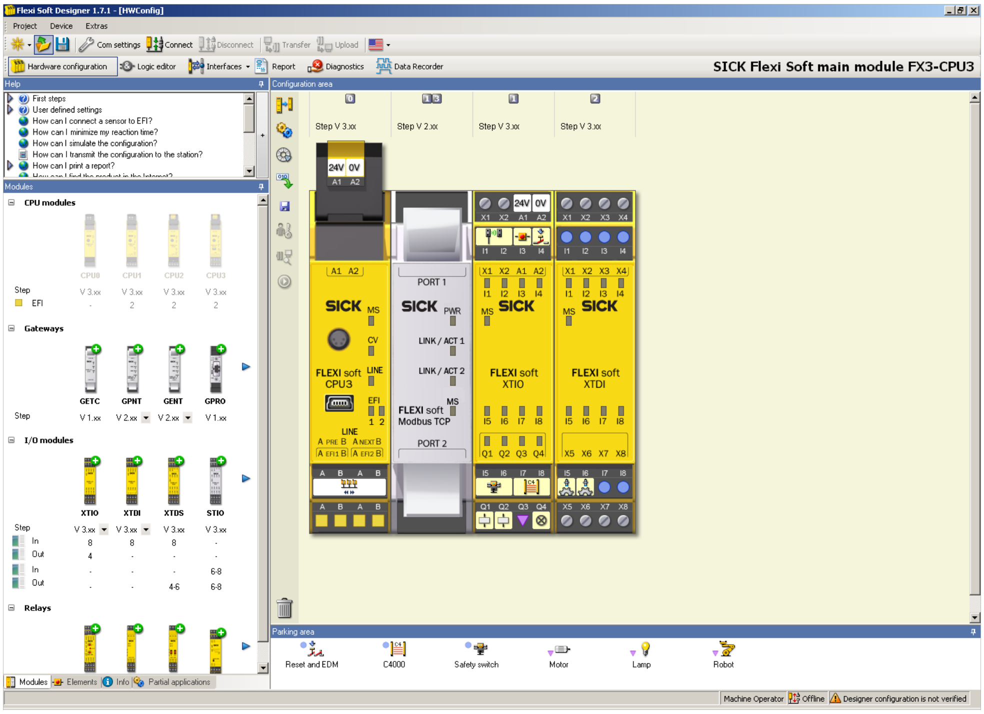


Figure 15: Hardware configuration

NOTE

- Double-click the main module in the configuration area to open the logic editor.
- Double-click a gateway in the configuration area to open the configuration view for the gateway concerned.

Zoom in/zoom out

Figure 16: Zoom in/zoom out button

The **Zoom in/zoom out** button enlarges or reduces the view of the configuration area.

Settings

Figure 17: Settings button

The **Settings** button opens the dialog box for the project settings. The following actions, amongst others, can be performed here:

- Define your own tag name format
- Activate or deactivate customized elements (see "Customized elements", page 47)
- Activate or deactivate the import of customized function blocks
- Activate or deactivate RS-232 routing for the main module
- Enable additional XT modules (see "Configuring the Flexi Soft modules", page 42)
- Save the current display and/or activate a saved display
- Change the paths for the folders that are used to store customized elements
- Export the module status bytes as a CSV file, e.g., for use in a PLC

Edit tag names

Figure 18: Button for editing tag names

The **Edit tag name** button opens the central Tag name editor (see "Tag name editor", page 71).

Export tag names to Pro-face GP-Pro EX

Figure 19: Button for exporting tag names to Pro-face GP-Pro EX

The **Export tag names to Pro-face GP-Pro EX** button is used to export a list of the tag names for use in a Pro-face HMI (see "Exporting tag names for use in Pro-face GP-Pro EX", page 72).

Edit button

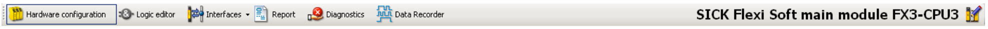


Figure 20: Edit button in Hardware configuration

The **Edit** button in the top right-hand corner of the screen above the configuration area can be used to switch to editing mode while Flexi Soft Designer is connected to the Flexi Soft system. This enables the configuration to be edited without having to first disconnect from the system.

NOTE
The **Edit** button is only visible when Flexi Soft Designer is connected to the system.

6.5.1 Configuring the Flexi Soft modules

- ► Create a new project (**Project**, **New**, **Single-station project** menu command). All the Flexi Soft modules are displayed in the **Modules** selection window. All modules are grayed out apart from the FX3-CPUx main modules.
- ► In the selection list, select the function package under the relevant main module (FX3-CPUx). Function package Step V 2.xx requires a main module with a firmware version ≥ 2.00, for example (see "Version, compatibility, and features", page 19).
- ► Using the mouse, drag the main module into the **Configuration area**. A larger representation of the main module is displayed there. The inputs and outputs (terminals) are visible. Now the main modules are grayed out in the **Modules** selection window and the other modules (gateways, I/O modules, analog input module, Speed Monitor, relay modules) can be selected.
- ► Drag other Flexi Soft modules into the **Configuration area**. Green arrows indicate where the new module is positioned. Gray arrows indicate other possible positions. The main module is always located on the far left. Directly to the right of the main module, up to two gateways then follow. It is only after these that the other expansion modules are arranged (I/O modules, analog input module, Speed Monitor). Relay modules must be placed on the far right.
- ► In the context menus of the various modules, click **Edit...**, enter a new name for the respective module, and click **OK** to close the dialog box again.
- ► Change the position of the module by dragging it with the mouse.
- ► Select the **Delete module** command in the context menu of a module to delete it again from the configuration area. Alternatively, use the mouse to drag the module into the recycle bin at the bottom left of the **Configuration area**.

NOTE

- A maximum of two gateways can be connected to one Flexi Soft main module.
- A maximum of twelve expansion modules can be connected to one Flexi Soft main module.
- Gateways are not classed as expansion modules.
- Each FX3-MOCx module occupies the space of two expansion modules.

Enabling configurations with more than twelve expansion modules

Flexi Soft Designer version ≥ V1.5.0 allows for configurations with up to 22 expansion modules, enabling you to create a maximum configuration that can be used with several similar plants/systems, for example. To adapt it to the plant/system concerned, all you have to do is delete those modules that are not required.

Enabling configurations with more than twelve expansion modules

- ► In the **Hardware configuration**, click the **Settings** button on the left of the **Configuration area** to open the **Settings** dialog box.
- ► On the **General** tab, select the **Enable additional XT modules (more than 12)** option.
- ► Click on **OK**.

NOTE

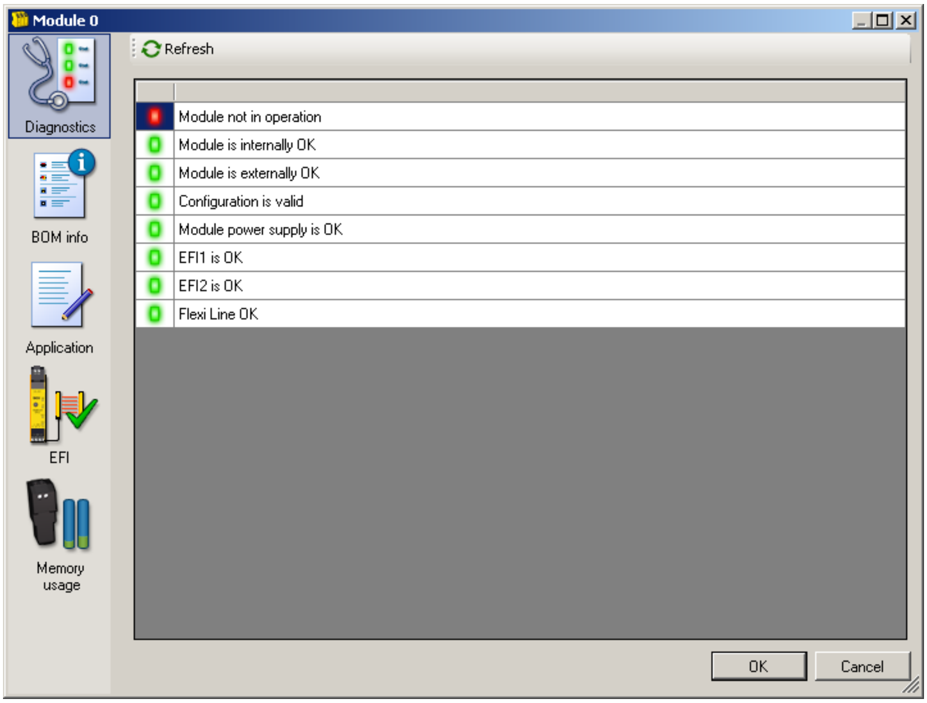
- If a configuration contains more than twelve expansion modules, the following restrictions apply:
  - It is not possible to establish a connection to a Flexi Soft system.
  - It is not possible to transfer the configuration to a Flexi Soft system.
  - It is not possible to perform a simulation.
- An expanded configuration cannot contain any more than two gateways either.

6.5.2 Module status bits in the Hardware configuration view

When Flexi Soft Designer is online, i.e., connected to the system, the status bits for each module along with their current values can be displayed.

► In the context menu of any of the modules (main module, gateway or expansion module), click **Edit...**. If the system is online, the dialog box for the selected module will contain an additional **Diagnostics** tab. This displays all the available status bits for the selected module along with their associated values.

► Click the **Refresh** button to update the values of the module status bits.



🟥	Module not in operation
🟩	Module is internally OK
🟩	Module is externally OK
🟩	Configuration is valid
🟩	Module power supply is OK
🟩	EFI1 is OK
🟩	EFI2 is OK
🟩	Flexi Line OK

Figure 21: Status bits of the main module in the Hardware configuration view

Exporting module status bits

► In the **Hardware configuration**, click the **Settings** button on the left of the **Configuration area** to open the **Settings** dialog box.

► Go to the **Export module status** tab and click on the **Export** button. A file selection window opens.

► Select the folder where the export file is to be saved, enter a name for the export file, and click **Save**. The module status bits are saved as a CSV file.

6.5.3 Connecting elements

- ► You can expand the structure in the **Elements** selection window with a click of the mouse.
- ► Optional: In the context menu of an element, select the **Edit template...** and assign a customized **Internal device number** to the element.
- ► Select multiple elements and drag them into the **Parking area**.

ℹ **NOTE**
The **Parking area** is there so that you can keep track of everything. This is where you can first collect all the required elements. Alternatively, the elements can be taken directly from the **Elements** selection window and dragged into the **Configuration area**.

- ► Drag an element from the **Parking area** (or from the **Elements** selection window) into the **Configuration area**.
- ► If the **Configuration area** does not contain a module with free inputs or outputs of an appropriate kind, you will not be able to position the element inside it. In this case, you need to first add at least one module with inputs or outputs to the Flexi Soft system in the configuration area, e.g., an FX3-XTIO or FX3-XTDI.
- ► Using the mouse, move the element over free inputs or outputs, if the element can be positioned on an input or output, it will turn green. The software automatically takes account how many inputs or outputs are required.
- ► Drag and drop the element at a suitable position. The element symbol is then integrated at this location.

ℹ **NOTE**
Some elements can only be connected to certain modules:

- Dual-channel elements can only be connected to safe inputs or outputs.
- Safety elements – such as an emergency stop or safety switch – can only be connected to safe modules; however, they cannot be connected to an FX0-STIO, for example.

- ► Elements can be dragged with the mouse to other suitable inputs or outputs, or back into the **Parking area**.
- ► To delete an element, click **Delete...** in the context menu of the element. Alternatively, use the mouse to drag the element into the recycle bin at the bottom left of the **Configuration area**.
- ► Once an element is inside the **Parking area** or the **Configuration area**, it can be configured, see "Configuring connected elements", page 45.

6.5.4 Safe and non-safe elements in the hardware configuration

Safe and non-safe elements are color-coded in the hardware configuration so that you can tell them apart:

- Safe elements are highlighted in yellow.
- Non-safe elements are highlighted in gray.
- Safe elements that are connected to a non-safe input or output are highlighted in red.

Most elements are only marked as safe or non-safe when they are dragged onto a corresponding input or output:

- Safe elements that are dragged onto a safe input or output are highlighted in yellow.
- Elements that are dragged onto a non-safe input or output are highlighted in gray.

- If a gray element is dragged onto a safe input or output, it remains gray but can be highlighted in yellow by editing it.
- If a yellow element is moved from a safe input or output to one that is not safe, it is highlighted in red to begin with. In this case, the configuration cannot be transferred. Before you can transfer the configuration, you need to go to the element settings for the relevant element and remove the check mark that identifies it as a safety element.

How to identify an element as a safety element

- ► Double-click on an element that is highlighted in gray or red, or select **Edit ...** in the context menu. The **Element settings** window opens.
- ► Mark the **Safety element** box.
- ► Close the **Element settings** window by clicking **OK**. The element is now highlighted in yellow to indicate that it is a safety element.

6.5.5 Expanding elements

Some elements are actually made up of a group of two or more subelements, e.g., a guard locking device consisting of a safety switch (which serves as the input element) and a guard locking element with interlocking (which serves as an output element). Normally, these elements must be connected to a module (e.g., FX3-XTIO) but some of them can be expanded so that the individual subelements can be connected to various different modules.

Expanding elements

- ► Place an element (e.g., a guard locking element) in the parking area.
- ► In the context menu of the element, select the **Expand** command. The element in the parking area is now replaced by its associated subelements, which can be handled in exactly the same way as individual elements.

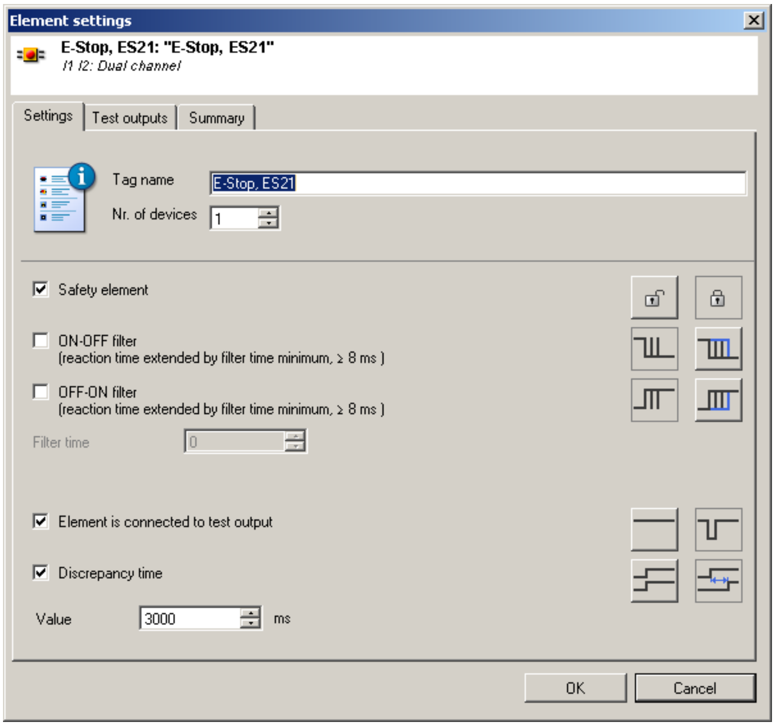
6.5.6 Configuring connected elements

Input and output elements can be configured once they are inside the **Parking area** or the **Configuration area**. Depending on the type of element, the following actions can be performed:

- Assign tag names (identifying names for the element)
- Set evaluation parameters for the element, e.g., discrepancy time, on-off filter or off-on filter, connection to a test output, whether test pulses should be activated or deactivated, and so on

Configuring elements

- ► Double-click on an element in the **Parking area** or **Configuration area** or select the **Edit ...** command in the context menu of the element. The **Element settings** window opens.



E-Stop, ES21: "E-Stop, ES21"
I1 I2: Dual channel

Tag name E-Stop, ES21
Nr. of devices 1

☑ Safety element

☐ ON-OFF filter
(reaction time extended by filter time minimum, ≥ 8 ms)

☐ OFF-ON filter
(reaction time extended by filter time minimum, ≥ 8 ms)

Filter time 0

☑ Element is connected to test output

☑ Discrepancy time

Value 3000 ms

Figure 22: Element settings window for an ES21 emergency stop pushbutton

Tag name

- ► Enter a **tag name** for the element if you want to assign one. Otherwise, the default tag name will be used instead.

Nr. of devices

- ► Enter the **No. of devices**, if necessary. If a cascade of several testable L21 type 2 sensors is connected to an input, for example, you can use this function to set the correct number of devices which are then shown in the bill of materials in the report.

Safety element

The **Safety element** option can be deactivated if the element is not used for safety functions. The element can then also be connected to non-safe inputs (e.g. a FX0-STIO).

See "Safe and non-safe elements in the hardware configuration", page 44.

ON-OFF filter and OFF-ON filter

- ► If necessary, activate the **ON-OFF filter** or **OFF-ON filter** and set the desired **filter time.**

WARNING
Extended response time

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

- ► Select the filter time to be as short as possible.
- ► Take into account the extended response time.

Effect of the filter on the response time:

- The switch-off response time is extended by at least as long as the selected filter time, if the ON-OFF filter is active.
- The switch-on response time is extended by at least as long as the selected filter time, if the OFF-ON filter is active.
- If the signal changes within the selected filter time, the response time may be extended by longer than the selected filter time, i.e. until a constant signal has been detected for at least as long as the selected filter time.

Element is connected to test output

The **Element is connected to test output** activates the testing of the relevant element:

- Sensor wiring short-circuits to 24 V that could interfere with a switch-off condition can be detected.
- Electronic sensors with test inputs (e.g., L21) can be tested.

► To activate or deactivate the connection to the test outputs, either click the check box or the 3D buttons on the right-hand side of it.

If an element has been connected to test outputs, the **Test outputs** tab will also be available in the Element settings dialog box. The **test period** and **test interval** can be configured here.

NOTE
An FX3-XTDI only has two test sources even if it features eight test output terminals.

WARNING
Ineffectiveness of the protective device due to unexpected pulses or delayed falling signal edges at single-channel inputs

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.

► Protect single-channel inputs against short-circuits and cross-circuits.
  ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines).
  ► No short-circuit detection, i.e., no referencing to test outputs.

Discrepancy time

In the case of dual-channel elements, a **discrepancy time** can be configured.

6.5.7 Customized elements

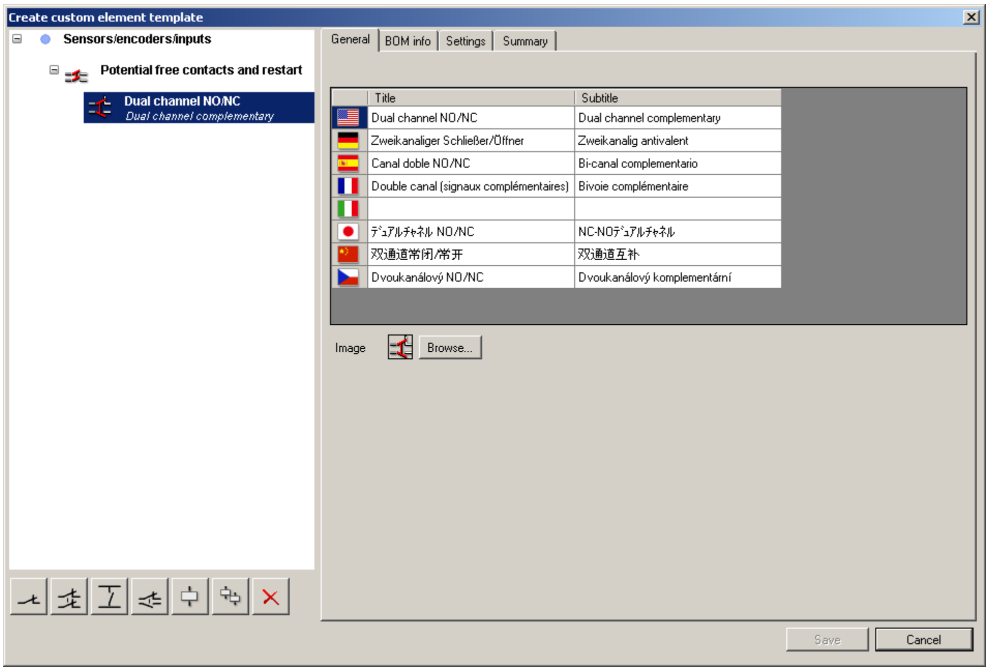
In addition to the standard input and output elements that are installed in conjunction with Flexi Soft Designer, it is also possible to create, configure, import, and export customized elements. This function allows you to create your own elements using preset configuration options (e.g., single-channel or dual-channel evaluation, discrepancy time, on-off filtering, connection to test outputs) to meet the requirements of your particular equipment.

Activating customized elements

► In the **Hardware configuration**, click the **Settings** button on the left of the **Configuration area** to open the **Settings** dialog box.
► On the **General** tab, select the **Enable customized elements** option.
► Click on **OK**.

Creating customized elements

► In the context menu of any element in (in the **Elements** window, in the **Configuration area** or in the **Parking area**) select the **Save as element template** command. The **Create custom element template** window opens.
We recommend selecting an element that bears as much similarity as possible to the customized element that you want to create.



	Title	Subtitle
	Dual channel NO/NC	Dual channel complementary
	Zweikanaliger Schließer/Öffner	Zweikanalig antivalent
	Canal doble NO/NC	Bi-canal complementario
	Double canal (signaux complémentaires)	Bivoie complémentaire
	ﾃﾞｭｱﾙﾁｬﾈﾙ NO/NC	NC-NOﾃﾞｭｱﾙﾁｬﾈﾙ
	双通道常闭/常开	双通道互补
	Dvoukanálový NO/NC	Dvoukanálový komplementární

Figure 23: Create custom element template window

► Rename the element and configure as required (see below for details).

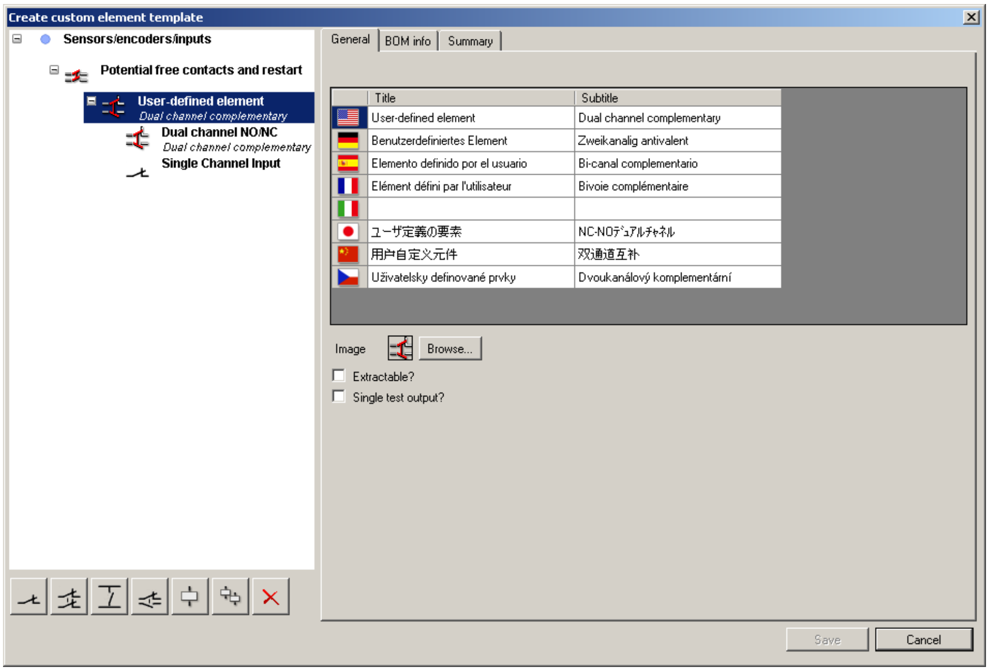
► When finished configuring the element, click **Save** to save the new element and close the window.

NOTE

- A new name must be entered for the element so that it can be saved.
- Before saving, ensure that the settings are complete and correct. It is not possible to edit an element once it have been saved.

Configuring customized elements

► Highlight the new customized element in the element tree and use the buttons for the subelements underneath the tree to add extra inputs or outputs. You can choose between single-channel inputs/outputs and a variety of dual-channel types. When you add subelements, they appear in the element tree one level below the customized element.



Create custom element template

- Sensors/encoders/inputs
  - Potential free contacts and restart
    - User-defined element *Dual channel complementary*
      - Dual channel NO/NC *Dual channel complementary*
      - Single Channel Input

General | BOM info | Summary

	Title	Subtitle
🇺🇸	User-defined element	Dual channel complementary
🇩🇪	Benutzerdefiniertes Element	Zweikanalig antivalent
🇪🇸	Elemento definido por el usuario	Bi-canal complementario
🇫🇷	Elément défini par l'utilisateur	Bivoie complémentaire
🇮🇹		
🇯🇵	ユーザ定義の要素	NC-NOﾃﾞｭｱﾙﾁｬﾈﾙ
🇨🇳	用户自定义元件	双通道互补
🇨🇿	Uživatelsky definované prvky	Dvoukanálový komplementární

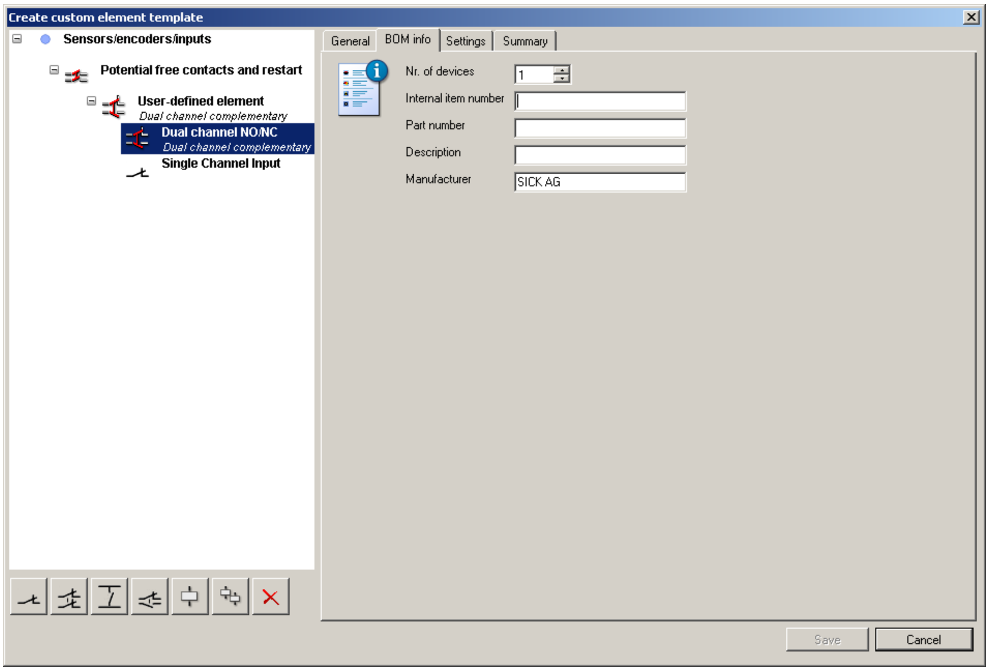
Image Browse...

☐ Extractable?

☐ Single test output?

Figure 24: Adding and deleting subelements for a customized element

- ► Highlight the new element in the element tree and enter a new name for it on the **General** tab. It is not possible to save an element under a name that already exists. However, it is not necessary to enter the new name of the element in all of the languages displayed. The name of the element must be entered in the language that is currently set for Flexi Soft Designer.
- ► Click the **Browse...** button to open a file selection window and assign a custom graphic to the element or subelement.
- ► If an element contains two or more subelements, the **Extractable?** option will be available. If a template has been created with this option checked, any elements that are based on it can be extracted or “split” into subelements. These can then be handled in exactly the same way as individual elements (see "Expanding elements", page 45).
- ► If the **Single-channel test output?** option is selected, all the subelements that make up the element must be connected to the same test output. Examples of such elements include the tested operating mode selector switches that either have to be connected to inputs I1/I3/I5/I7 (if test output X1 is used) or to inputs I2/I4/I6/I8 (if test output X2 is used).
- ► On the **BOM info** tab, enter the details about the elements and subelements to be included in the bill of materials. This information will be used in the bill of materials within the Flexi Soft Designer report.

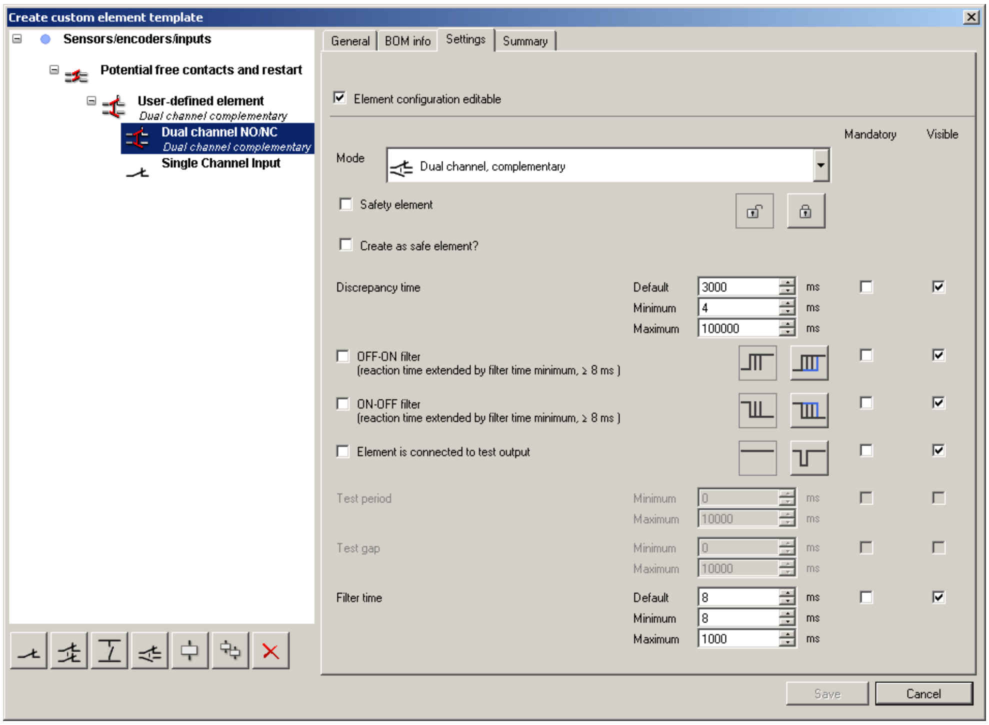


Sensors/encoders/inputs
Potential free contacts and restart
User-defined element
Dual channel complementary
Dual channel NO/NC
Dual channel complementary
Single Channel Input

Nr. of devices: 1
Internal item number:
Part number:
Description:
Manufacturer: SICK AG

Figure 25: Entering the BOM information for a customized element

- ► Select the customized element or subelement to be configured and click on the **Settings** tab to edit the configuration settings.



Sensors/encoders/inputs
Potential free contacts and restart
User-defined element
Dual channel complementary
Dual channel NO/NC
Dual channel complementary
Single Channel Input

☑ Element configuration editable

				Mandatory	Visible
Mode: Dual channel, complementary					
□ Safety element					
□ Create as safe element?					
Discrepancy time	Default	3000	ms	□	☑
	Minimum	4	ms		
	Maximum	100000	ms		
□ OFF-ON filter (reaction time extended by filter time minimum, ≥ 8 ms)				□	☑
□ ON-OFF filter (reaction time extended by filter time minimum, ≥ 8 ms)				□	☑
□ Element is connected to test output				□	☑
Test period	Minimum	0	ms	□	□
	Maximum	10000	ms		
Test gap	Minimum	0	ms	□	□
	Maximum	10000	ms		
Filter time	Default	8	ms	□	☑
	Minimum	8	ms		
	Maximum	1000	ms		

Figure 26: Editing the configuration settings of a customized element

- ► If you want the created element to be configurable within the limits preset on this tab, the **Element configuration editable** box must be checked.
- ► Adjust the settings (e.g., discrepancy time, ON-OFF filter, OFF-ON filter, etc., see "Configuring connected elements", page 45). You can specify a particular value to serve as a default, and can configure minimum and maximum values.

- ► If a created element **has to** be used with a module that supports a specific function, the **Mandatory** box for the function concerned must be checked (e.g. to create an element that has to be connected to a module with test outputs).
- ► If the **Visible** box is checked for a particular option, the configured default for this option can be changed later on by the user. If the **Visible** box for a particular option is not checked, the default setting is fixed and cannot be changed.

Transferring customized element to another computer

- ► Save the project and open it on the other computer. Any customized elements within the project are imported automatically.

NOTE

However, please note that customized elements can only be imported with Flexi Soft Designer version ≥ V1.3.0.

Deleting customized elements

- ► In the context menu of the element to be deleted, select the **Delete template ...** command and confirm with **Yes**.

Exporting customized elements as an XML file

- ► In the context menu of the customized element to be exported, select the **Export ...** command. A file selection window opens.
- ► Select the folder where the customized element is to be saved and click **Save**. The customized element is saved as an XML file.

Importing customized elements from an XML file

- ► In the **Elements** window, select the **Import...** command in the context menu of any element or element group. A file selection window opens.
- ► Select the XML file with the customized element to be imported and click **OK**. The customized element is imported.

6.5.8 Connecting EFI-enabled devices

EFI-enabled devices can be connected to main modules of type FX3-CPU1, FX3-CPU2 or FX3-CPU3.

- ► Use the mouse to drag the required EFI-enabled device (e.g., a C4000 safety light curtain) out of the **Elements** selection window and onto the EFI connection of the main module. The **Device selection wizard** opens. Here, you can either select the exact variant of the device or can enter the type code directly.
- ► Click **Finish** to confirm your choice and to connect the EFI-enabled device. The EFI status bits for the connected EFI-enabled device are now available in the logic editor as inputs and outputs of the main module.
- ► Double-clicking on an EFI-enabled device opens its configuration window.

NOTE

- The configuration for an EFI-enabled device must be loaded or transferred separately in the configuration window of the EFI-enabled device. For this purpose, the Flexi Soft Designer must be connected to the Flexi Soft system.
- For information about cascaded connections for EFI-enabled devices, please refer to the EFI technical description (SICK part number 8012611).
- Depending on which devices are already connected, restrictions may apply concerning which devices can be connected to the other EFI connection of the main module.

Switching over the EFI address

With some combinations of EFI-enabled devices it is absolutely essential for the EFI address of the Flexi Soft system to be set to 13, because EFI address 14 is already being used by another EFI-enabled device (e.g., an EFI gateway or UE403).

► To switch between EFI addresses 13 and 14, select **Address 13** or **Address 14** in the context menu of the main module.

NOTE
Once the EFI address has been switched over, the Flexi Soft system performs a restart, i.e., all outputs are deactivated.

EFI system integrity test

EFI-enabled devices that are connected to the EFI interfaces can be tested by the Flexi Soft main module whenever a reset is performed by disconnecting the power source. The following parameters can be compared with those stored when the main module was last configured:

- Type code: A device with the same type code is expected.
- Serial number: A device with the same serial number is expected.
- Configuration date: A device with the same configuration date is expected.

If the parameters of the connected device do not match the stored values, the main module sets the input and output data of the EFI-enabled device concerned to 0 and the associated EFI LED (EFI1 or EFI2) starts flashing ✹ red (1 Hz).

NOTE
If the configuration date is used as the basis for the EFI system integrity test, the configuration for the connected EFI-enabled devices must be transferred before the configuration for the main module.

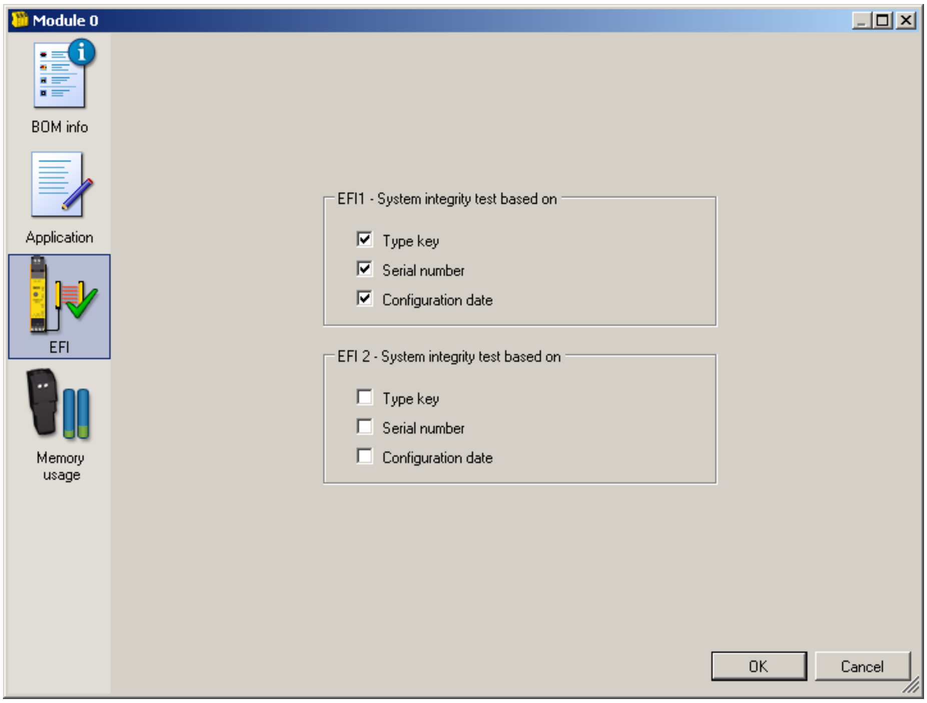
If the configuration does not match the devices that are physically present, a question mark appears next to the relevant EFI connection in the Hardware configuration view of Flexi Soft Designer.

- An EFI-enabled device is physically present on this EFI connection, but is missing from the configuration for the main module. If the configuration is now uploaded using the **Transfer** command, this device is added to the configuration for the main module. Exception: If the configuration in the main module has been verified, the sensor is not synchronized. In this case, the configuration in the main module will remain unchanged.
- An EFI-enabled device is configured on this EFI connection, but is not physically present. If the configuration is now uploaded with the **Transfer** command, this device is removed from the configuration in the main module. Exception: If the configuration in the main module has been verified, the sensor is not synchronized. In this case, the configuration will remain unchanged.

Configuring the EFI system integrity test

► If Flexi Soft Designer is connected to the system, click **Disconnect** or switch to **Editing mode.**

► In the context menu of the main module, select the **Edit...** command. In the dialog that appears next, click the **EFI** button on the left-hand side.



EFI1 - System integrity test based on

☑ Type key
☑ Serial number
☑ Configuration date

EFI 2 - System integrity test based on

□ Type key
□ Serial number
□ Configuration date

Figure 27: Configuring the EFI system integrity test

- ► Activate all the parameters that are to be used for the EFI system integrity test.
- ► Click on **OK** to accept the settings and close the dialog box.

6.5.9 Exporting and importing partial application

Overview

It is possible to export or import a partial application. In the process, all modules except for the main module are exported with the inputs and outputs connected to them, as well as the logic.

When a partial application is imported into an existing project, the saved modules, elements, and logic are added to the project but the rest of the project remains unchanged. This is particularly useful if you want to replace a main module within an existing project without having to reconfigure all the hardware and logic.

Important information

NOTE

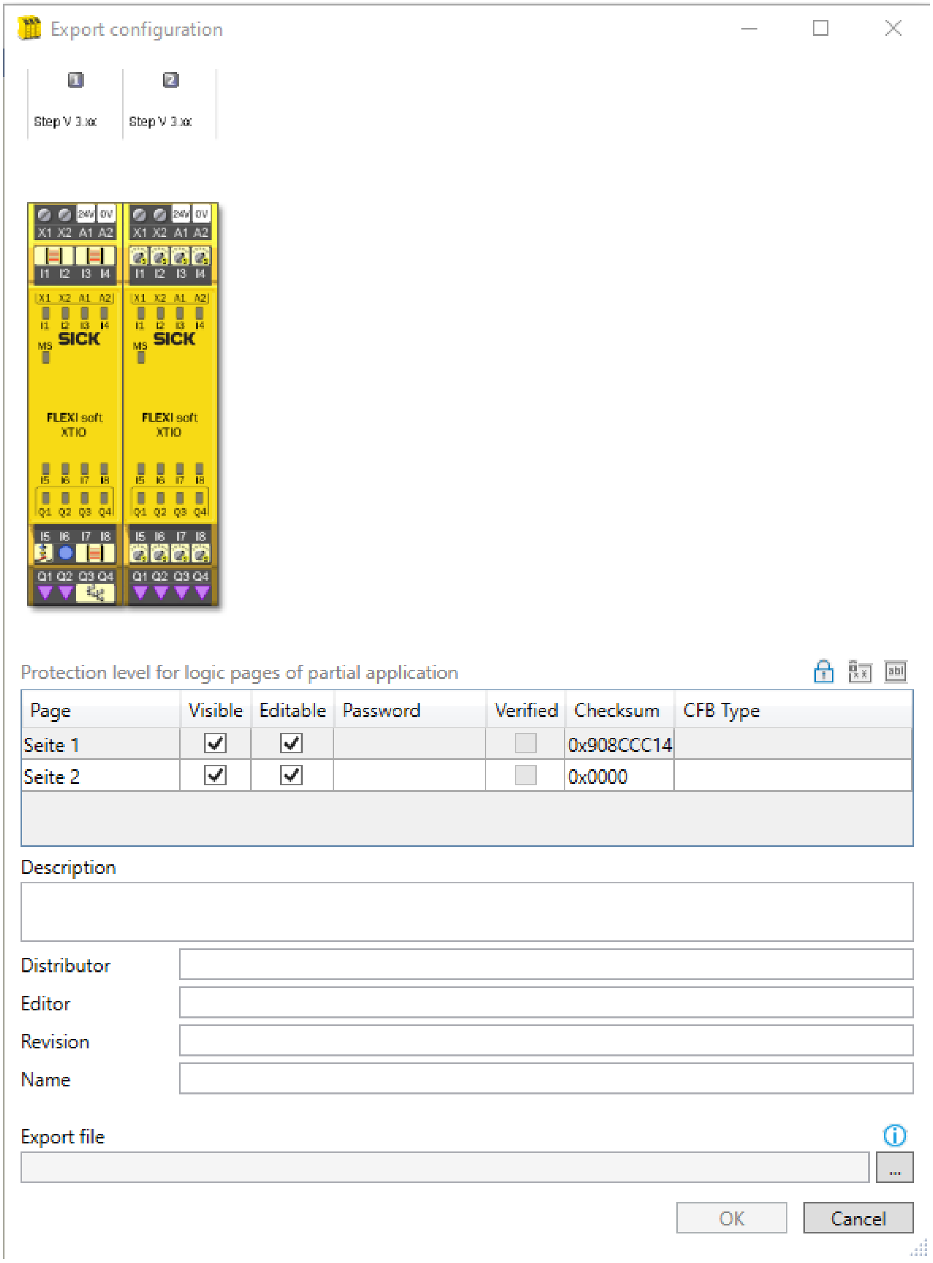
- The exported partial application does not contain the configuration for any EFI-enabled devices that may be connected. Therefore, these devices must be reconfigured. However, the connections are retained within the logic for these devices.
- When partial applications with any connected MOC devices are exported, the logic access levels and passwords of the MOC logic are not exported at the same time. Therefore, you may have to set up password protection again after importing a partial application.
- Partial applications that have been created using Safety Designer and partial applications that have been created using Flexi Soft Designer are not compatible with one another and cannot be used in the other program.
- Exporting and importing partial applications is only upward compatible. A partial application exported with FSD version 1.9.6 cannot be imported with FSD versions < 1.9.6.

Procedure

Exporting partial application

1. In the **Partial applications** selection window, click on **Save as a new partial application**. As an alternative, you can select the **Export configuration...** command in the context menu.

✓ The **Export configuration** dialog box opens.



Page	Visible	Editable	Password	Verified	Checksum	CFB Type
Seite 1	☑	☑		□	0x908CCC14	
Seite 2	☑	☑		□	0x0000	

Figure 28: Exporting configuration dialog box

2. If necessary, select logic page protection level.
3. Specify additional information if necessary, e.g. in the **Description** field.
4. Click on the **...** button to the right of the **Export file** field.

✓ The file selection window opens.

5. Select the folder where the export file is to be saved, enter a name for the export file, and click on **Save** to close the file selection window.
6. Click on **OK**.

Importing partial application

1. In the **Partial applications** selection window, click on **Open partial application**. As an alternative, in the context menu of the main module, select the **Import module configuration...** command.

✓ The **Import configuration** dialog box opens.

2. Click on the ... button to the right of the **Import file** field.

✓ The file selection window opens.

3. Select the folder that contains the file you want to import.

✓ All the Flexi Soft import files (*.fsi) inside the selected folder are displayed.

4. Select the desired file and click on **Open**.

✓ The partial application within the selected .fsi file is displayed along with its description.

5. Click on **OK**.

✓ The hardware configuration of the partial application is added to the hardware configuration of the current project. The logic program of the partial application is integrated into the logic editor of the current project on one or several separate new pages.

Example

There is a project containing a main module and an FX3-XTIO module, a deTec4, an emergency stop pushbutton, a robot, and a page with the necessary logic in the logic editor. The partial application that is to be imported contains an additional FX3-XTIO module with a two-hand control device and a motor, plus a page in the logic editor with the logic for controlling these devices. Once the import process is complete, the project contains both FX3-XTIO modules along with devices connected to each one and the two logic programs on two separate pages.

6.5.9.1 Replacing a main module in a Flexi Soft project

Overview

The Export and Import functions allow you to replace a main module within an existing project (e.g., to replace an FX3-CPU0 with an FX3-CPU1 or a module with a different firmware version) without having to reconfigure the project (hardware configuration, logic).

Important information

NOTE

The exported partial application does not contain the configuration for any EFI-enabled devices that may be connected. Therefore, these devices must be reconfigured. However, the connections are retained within the logic for these devices.

Procedure

1. Open the project with the main module to be replaced.
2. Export the project as a partial application as described above.
3. Close the project and open a new project (in the **Project** menu, select the **New, Single-station project** command).
4. In the **Hardware configuration** section, add the desired new main module to the new project.
5. Import the new project in the previously saved partial application.

6.5.9.2 Replacing an I/O module in a Flexi Soft project

Important information

NOTE

- This method does not work for elements that are used in conjunction with a **Fast shut off** function block, because these elements can no longer be moved across to another I/O module.
- This method is also not possible in the case of grouped elements, such as operating mode selector switches and switches with guard locking.

Procedure

1. Load the project with the I/O module to be replaced.
2. In the hardware configuration, add the desired new I/O module.
3. Move the connected elements across from the old I/O module to the new one. With this procedure, the connections are retained within the logic.
4. Delete the old I/O module.

6.5.10 RS-232 routing

The input and output data of the Flexi Soft system are available, amongst other places, at the RS-232 interface of the main module. This means, for example, that communication can take place between the Flexi Soft system and a connected PLC without you having to use a gateway or connect an HMI.

WARNING

Non-secure data in the RS-232 interface

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.

► Do not use the RS-232 interface for safety-related functions.

Using the RS-232 interface, you can read up to 100 bytes from the Flexi Soft system and write up to four bytes to the Flexi Soft system.

The bits received via RS-232 are available as inputs in the logic editor under **Diagnostics** => **RS-232**.

Activating RS-232 routing

► In the hardware configuration, click the **Settings** button on the left of the configuration area.

► On the **General** tab, select the **Activate RS-232 routing for CPU** option.

► Click on **OK**. RS-232 routing is now enabled. In the **Interfaces** menu, you can now open the configuration window for the data that is to be transmitted via RS-232.

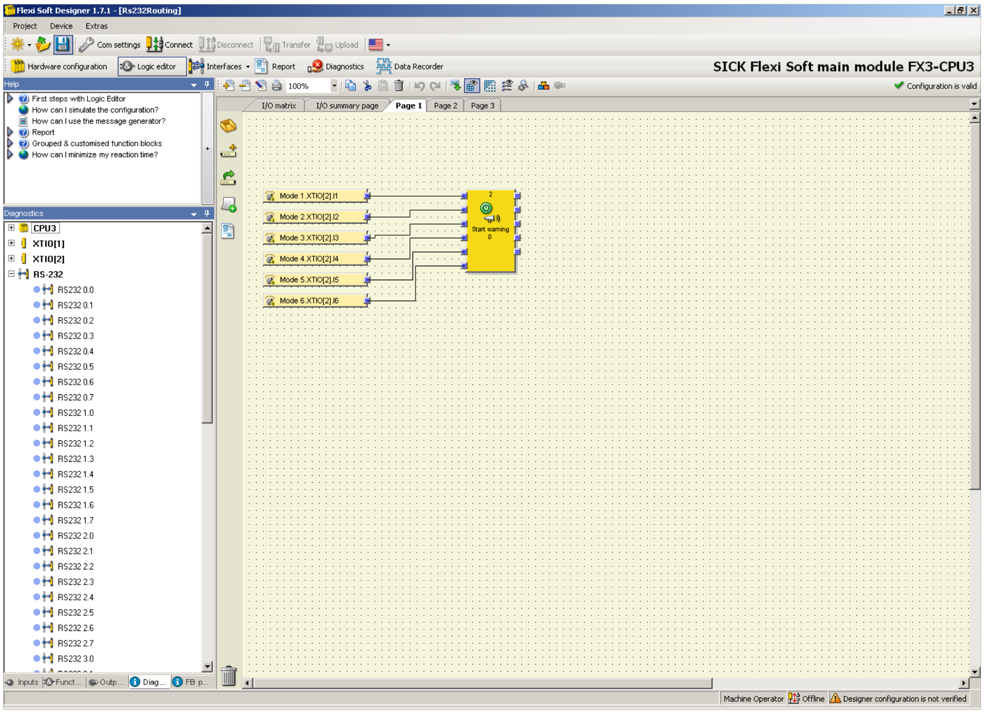
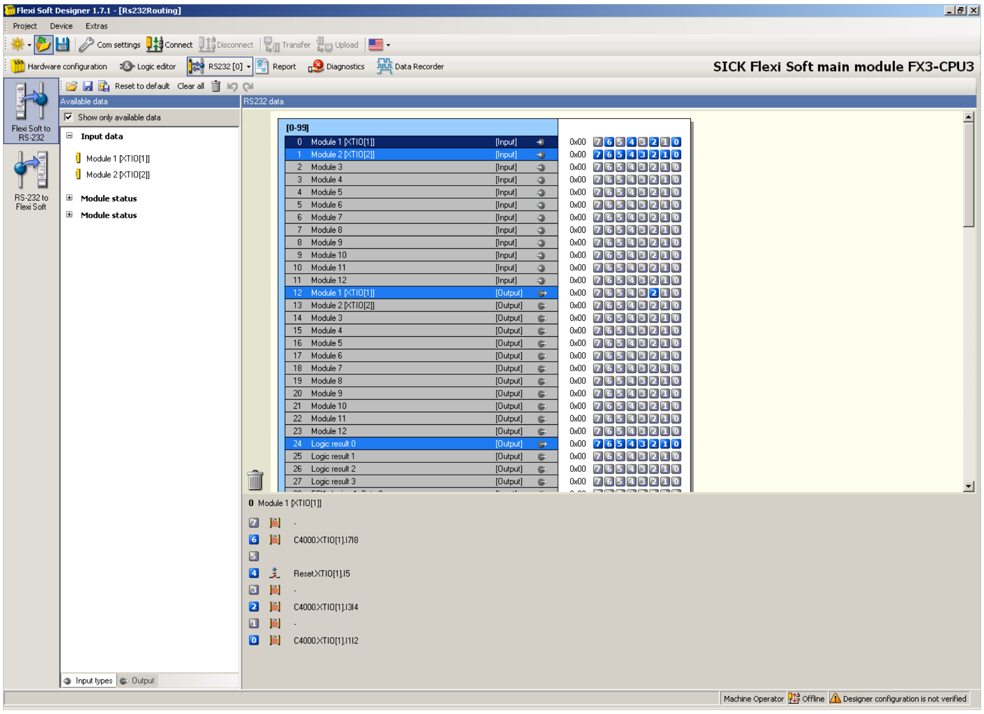


Figure 29: RS-232 output data in the logic editor

Configuring the input data for RS-232 routing

- ► In the **Interfaces** menu, click **RS-232 [0]** to open the dialog box for the RS-232 configuration.
- ► Click on the **Flexi Soft to RS-232** button to display the routing configuration for the input data.



☑ Show only available data

[0-99]

0	Module 1 [XTIO[1]]	[Input]	0x00	7 6 5 4 3 2 1 0
1	Module 2 [XTIO[2]]	[Input]	0x00	7 6 5 4 3 2 1 0
2	Module 3	[Input]	0x00	7 6 5 4 3 2 1 0
3	Module 4	[Input]	0x00	7 6 5 4 3 2 1 0
4	Module 5	[Input]	0x00	7 6 5 4 3 2 1 0
5	Module 6	[Input]	0x00	7 6 5 4 3 2 1 0
6	Module 7	[Input]	0x00	7 6 5 4 3 2 1 0
7	Module 8	[Input]	0x00	7 6 5 4 3 2 1 0
8	Module 9	[Input]	0x00	7 6 5 4 3 2 1 0
9	Module 10	[Input]	0x00	7 6 5 4 3 2 1 0
10	Module 11	[Input]	0x00	7 6 5 4 3 2 1 0
11	Module 12	[Input]	0x00	7 6 5 4 3 2 1 0
12	Module 1 [XTIO[1]]	[Output]	0x00	7 6 5 4 3 2 1 0
13	Module 2 [XTIO[2]]	[Output]	0x00	7 6 5 4 3 2 1 0
14	Module 3	[Output]	0x00	7 6 5 4 3 2 1 0
15	Module 4	[Output]	0x00	7 6 5 4 3 2 1 0
16	Module 5	[Output]	0x00	7 6 5 4 3 2 1 0
17	Module 6	[Output]	0x00	7 6 5 4 3 2 1 0
18	Module 7	[Output]	0x00	7 6 5 4 3 2 1 0
19	Module 8	[Output]	0x00	7 6 5 4 3 2 1 0
20	Module 9	[Output]	0x00	7 6 5 4 3 2 1 0
21	Module 10	[Output]	0x00	7 6 5 4 3 2 1 0
22	Module 11	[Output]	0x00	7 6 5 4 3 2 1 0
23	Module 12	[Output]	0x00	7 6 5 4 3 2 1 0
24	Logic result 0	[Output]	0x00	7 6 5 4 3 2 1 0
25	Logic result 1	[Output]	0x00	7 6 5 4 3 2 1 0
26	Logic result 2	[Output]	0x00	7 6 5 4 3 2 1 0
27	Logic result 3	[Output]	0x00	7 6 5 4 3 2 1 0

0 Module 1 [XTIO[1]]

7	-
6	C4000.XTIO[1].I7I8
5	
4	Reset.XTIO[1].I5
3	-
2	C4000.XTIO[1].I3I4
1	-
0	C4000.XTIO[1].I1I2

Figure 30: Configuration for the operating data transmitted to the network via RS-232

This dialog box is divided into the following areas: the **Available data** area on the left, the **RS-232 data** area on the right. The available bytes are shown at the top, and the **Tag names** for the currently selected bytes. The toolbar is located above the dialog box.

The toolbar

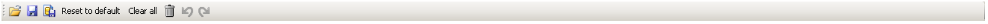


Figure 31: Toolbar for the routing configuration

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

- The **Load user-specific configuration** and **Save user configuration** buttons can be 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.
- The **Import** and **Export** buttons are used to import and export the currently used tag names as a CSV file. 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 **RS-232 to Flexi Soft** direction.

- **Reset to default** restores the default routing configuration. The command must be confirmed with **Yes**. In this case, any unsaved changes to the routing configuration will be lost. This command cannot be undone.
- **Clear all** deletes the routing configuration, i.e., all bytes assigned in the **RS-232 data** are cleared.
- **Remove routing** deletes the byte that is currently selected in the **RS-232 data** area.
- The **Undo** and **Redo** buttons can be used to undo or redo any changes that have been made to the routing configuration.

Available data area

This area contains all the sources from which data can be routed into the network. It is divided into two views, which contain the available input and output data. The tabs at the bottom can be used to switch between these two views.

- The **Input types** view contains the input values of the connected Flexi Soft modules and EFI-enabled devices, plus the module status data. If the Flexi Soft system contains gateways, the input data for the gateways (i.e., the data received by the gateways from the network) will also be available here.
- The **Output** view contains the output values of the connected Flexi Soft modules and EFI-enabled devices, plus the **Logic results** and **CPU marker** from the logic editor.

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

- Connected Flexi Soft modules
- Connected EFI-enabled devices
- Configured logic results and CPU marker [1)]
- Gateway input data and gateway output data

Sources that are not supported by the current configuration are shown in gray. You can use the **Show only available data** checkbox at the top left to hide sources which are not used.

Sources which provide live data are indicated with an icon to the left of the text (see figure 30, page 57).

[1)] In the default configuration, only the first byte of the logic results (logic result 0) and CPU marker is active and available. Additional output bits can be activated in the logic editor.

The RS-232 data area

This area contains the routing table. It displays the current contents of the data sent via the RS-232 interface. 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 do not currently have any data assigned to them because the hardware configuration does not support the sources.

Adding a data byte to the routing table

► Drag an element (e.g., a byte) from the **Available data** area to a free location in the **RS-232 data** area (drag & drop). If the desired position is not free, it must first be cleared by deleting the byte that is currently assigned to it or by moving this byte to another position in the table.

ℹ **NOTE**
The same byte can be used multiple times within the routing table.

Deleting a data byte from the routing table

► Drag the byte to be deleted to the trash can icon in the bottom left corner for the **RS-232 file** area (drag and drop).

Or:

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

Or:

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

Moving a data byte to another position in the routing table

► Drag the byte to be moved to the desired position (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.

The tag names area

This area displays the tag names of all the bits of the byte currently selected in the **Available data** or **RS-232 data** areas. You can edit the tag names in the tag name editor and – to a certain extent – also in the logic editor and the hardware configuration dialog (e.g. for expansion modules). It is not possible to edit the tag names in the **Tag names** area of the configuration window for the **Flexi Soft to RS-232** routing direction.

Configuring the tag names for the received data

► Click the **RS-232 to Flexi Soft** button. The following dialog box is displayed:

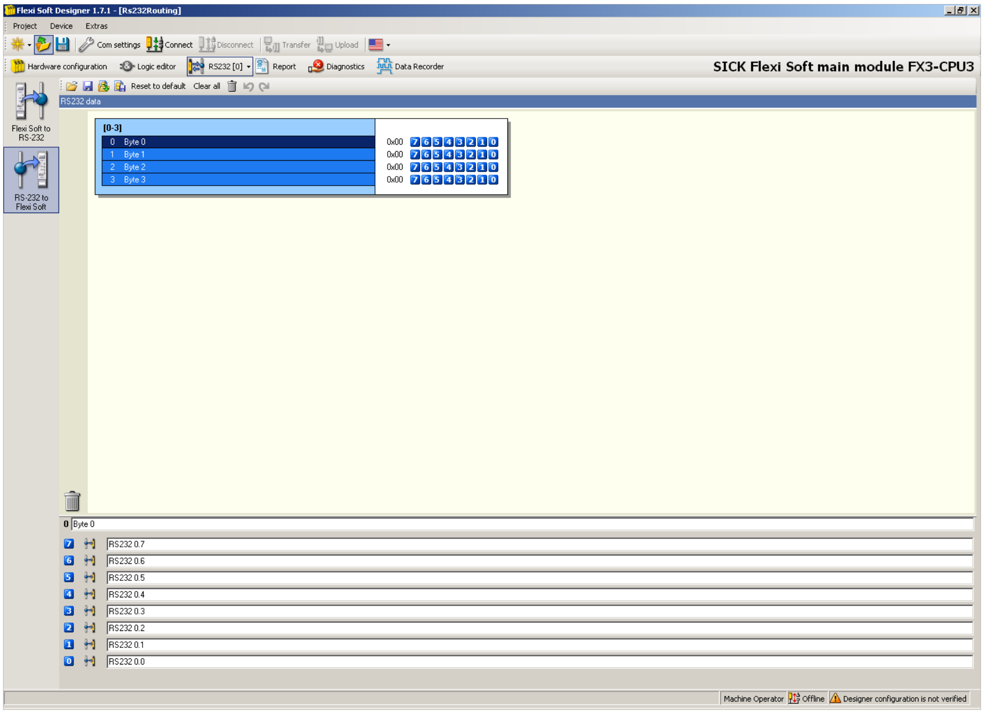


Figure 32: Configuration for the operating data received from the network via RS-232

The **RS-232 data** area displays the current configuration for the output data. The area below this shows the tag names that have been assigned to the byte selected above.

► Select a byte in the **RS-232 data** area.

► Enter the desired tag name for each bit of the selected byte that is to be used.

Loading and saving a routing configuration

The **Load user-specific configuration** and **Save user configuration** buttons can be 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.

Importing and exporting the tag names

The **Import** and **Export** buttons are used to import and export the currently used tag names as a CSV file. This allows you to import the tag names into a PLC program and use them there.

When you import the tag names, all unsaved changes will be lost. This command cannot be undone.

NOTE

The **Import** button is only available when the routing is configured for the **RS-232 to Flexi Soft** direction.

6.5.11 Optimization of the logic execution time

Flexi Soft main modules with a firmware version ≥ V4.00.0 feature firmware optimizations that may affect the logic execution time. To ensure compatibility with older models, these optimizations can be activated or deactivated by the user.

To benefit from the improved performance of this firmware, select **Logic execution time optimization** in Flexi Soft Designer and deactivate any functions that are not being used (Flexi Line, Flexi Loop, EFI including Flexi Link).

When the **Logic execution time optimization** option is activated, the logic program is executed more quickly in the main module. This can reduce the logic execution time. Particularly in the case of complex applications, this means that a shorter processing time can be achieved and, in turn, a shorter response time.

⚠ WARNING

Change of the logic execution time

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

► After each activation or deactivation of the **Logic execution time optimization** option, check the correct function of the entire application.

ℹ NOTE

The minimum logic execution time of a Flexi Soft system is always 4 ms. This cannot be reduced any further – not even by applying the optimization settings.

Changes to the logic execution time may make it necessary to change the configuration for function blocks whose parameters are based on the logic execution time.

To use the feature for optimizing the logic execution time, you must have the following installed: an FX3-CPUx main module with a firmware version ≥ V4.00.0 (Step 4.xx) and Flexi Soft Designer version ≥ V1.7.1.

Optimizing the logic execution time

► In the context menu of the main module, select the **Edit...** command.

► In the next dialog box, click on the **Logic execution time** button.

► Select the **Logic execution time optimization** option.

To further increase the performance of the main module, you can now deactivate other functions that are not required:

- EFI including Flexi Link (FX3-CPU1 or later)
- Flexi Loop (FX3-CPU0 or later)
- Flexi Line (FX3-CPU3 or later)

ℹ NOTE

- The aforementioned functions (Flexi Loop, Flexi Line, EFI including Flexi Link) can only be deactivated if they are not being used in the current project. If any of these functions has already been configured, it must first be removed from the project before it can be deactivated.
- Optimization of the logic execution time occasionally affects function block parameters, e.g., those of the Clock generator function block.
- Optimization of the logic execution time also affects calculation of the logic execution time.

6.6 Logic editor

Flexi Soft Designer features a graphical **logic editor**. The function logic is programmed using logical and application-specific function blocks. The inputs, function blocks, and outputs are arranged on a worksheet and connected as appropriate.

As soon as a Flexi Soft main module is placed inside the configuration area, the **logic editor** becomes accessible via the corresponding button.

FX3-MOCx modules feature their own logic editor with special function blocks for drive monitoring. As soon as an FX3-MOCx module is placed inside the configuration area, the associated logic editor likewise becomes accessible via the **logic editor** button.

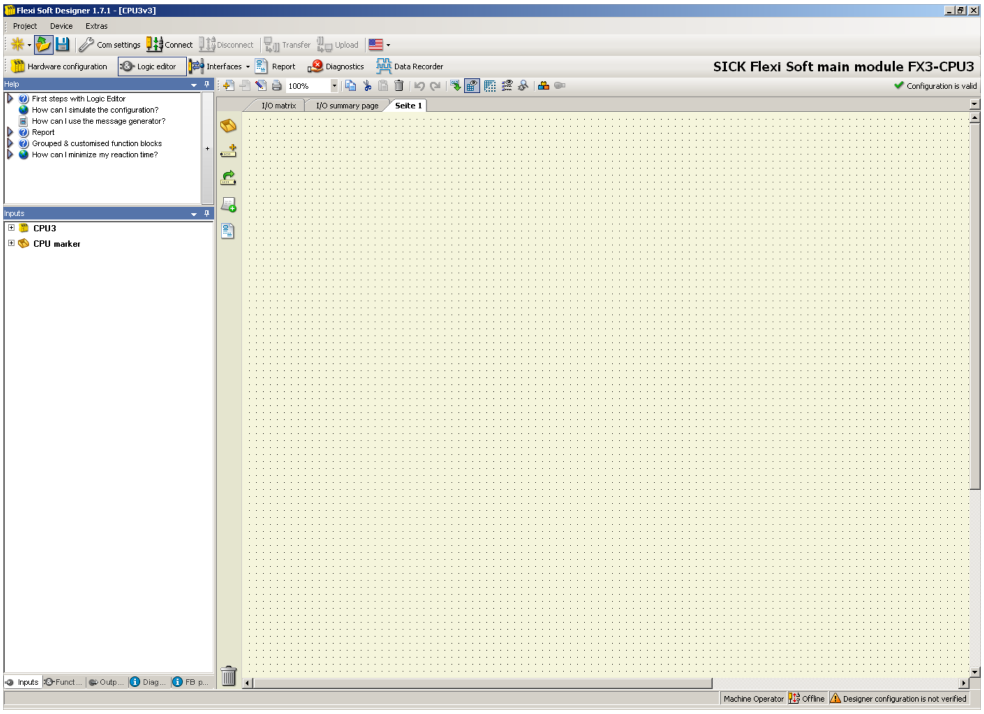

Flexi Soft Designer 1.7.1 - [CPU3v3]
Project
Device
Extras
Com settings
Connect
Disconnect
Transfer
Upload
Hardware configuration
Logic editor
Interfaces
Report
Diagnostics
Data Recorder
SICK Flexi Soft main module FX3-CPU3
Help
100%
Configuration is valid
First steps with Logic Editor
How can I simulate the configuration?
How can I use the message generator?
Report
Grouped & customised function blocks
How can I minimize my reaction time?
I/O matrix
I/O summary page
Seite 1
Inputs
CPU3
CPU marker
Inputs
Funct...
Outp...
Diag...
FB p...
Machine Operator
Offline
Designer configuration is not verified

Figure 33: The logic editor

The **Logic editor** window consists of the following elements:

- The logic editor toolbar, which offers the functions: **Add new page**, **Delete/Rename current page**, **Print current page**, **Zoom**, **Copy/Cut/Paste/Delete elements**, **Undo/Redo last action**, **Open dialog box for editing logic results**, **Show grid**, **Show line grid/Point grid**, **Show description of function block I/Os**, **Search for function block**, **Refresh the connections on the logic page**, **Start simulation mode**, and **Start force mode**
- Window for selecting **Inputs**, **Function blocks**, **Outputs**, and **Diagnostics**
- **FB group info** window on the bottom left for displaying key system resource information such as the number of function blocks used/available or the current logic execution time (logic cycle time). When you move the mouse pointer over a function block on the worksheet, the **FB group info** window displays additional information about the function block concerned.
- Worksheets (**pages**) for creating the logic – **I/O summary page** and **I/O matrix** – each of which can be selected via the relevant tab
- The **Help** window on the top left-hand side, which contains information on selected topics. Double-clicking an entry opens the corresponding article in the browser.

6.6.1 Using the Logic editor

- ► In the **Hardware configuration**, create a Flexi-Soft-System with one main module, at least one FX3-XTIO module, and an input or output element.
- ► Open the **Logic editor.**
- ► Drag an input from the **Inputs** selection window onto the worksheet. To select multiple inputs at the same time, either click on each of these in turn while pressing the `Ctrl` key, or first press the `Shift` key then click on the top input and then the bottom input in a sequence of inputs. The selected inputs can now be dragged over to the workspace in one go.

- ► In the window for selecting **Inputs**, **Function blocks**, and **Outputs**, click **Function blocks** and drag an application-specific or a logical function block over to the worksheet from the selection list. The function block will be shown in red until all of its inputs have been connected.

ℹ **NOTE**
The function block index is displayed at the top of each function block in the work area. This specifies the position of the function block in the execution sequence.

- ► In the window for selecting **Inputs**, **Function blocks**, and **Outputs**, click **Outputs** and drag an output over to the worksheet from the selection list.

ℹ **NOTE**
The inputs and outputs are color-coded in the **Logic editor** according to their function:

- Gray: Non-safe
- Yellow: Safe
- Blue: Diagnostics

- ► Connect the input to an input of the function block. To do this, click on the input node, hold down the mouse button and drag the mouse pointer to the input node of the function block. Connect one output of the function block to the output accordingly. As soon as all the inputs of the function block have been connected, the function block turns yellow.
- ► Alternatively, you can place and connect inputs or outputs in a single step. Drag an input or an output to the area directly above the input/output node of the function block to which it is to be connected. The node becomes highlighted as soon as the mouse pointer is above it. Then drag the input or output to the position on the worksheet where it is to be placed (drag and drop).
- ► By holding down the `Strg` key, you can move the end of an existing connection line across from one node to another. This means you can change a connection without deleting it first.
- ► Select the input, the function block, the output, and the connections by clicking or dragging them with the left mouse button. The selection can be moved anywhere in the work area.
- ► In the window for selecting **Inputs**, **Function blocks**, and **Outputs**, select **FB group info.** When moving the mouse over an element or a function block, detailed information appears in the **FB group info**window.
- ► To replace one function block with another, drag the new function block across from the selection list, hover it over the existing function block until that one becomes highlighted in green, and then release the mouse button.
- ► Right-click on an input or output element in the logic editor to display the logic pages where the element you have clicked is used.
- ► To delete an element, select the **Delete** command from the element’s context menu.

6.6.2 CPU markers

CPU markers are available in the logic editor as inputs and outputs. They can, for example, be used to create logical loopbacks or to connect the output of a function block on one page of the logic editor to the input of a function block on another page.

A CPU marker consists of an output marker and an input marker. After a delay of one logic cycle (i.e., the logic execution time), the input marker always assumes the same value (1 or 0) as the associated output marker.

WARNING
Extension of the response time

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

► Note the response time extension due to the extended logic execution time when using CPU markers.

Using CPU markers

► Connect a CPU output marker (e.g., Marker 0.0) from the **Outputs** tab of the logic editor to the output of a function block.
► Take the associated CPU input marker (e.g., Marker 0.0) from the **Inputs** tab of the logic editor and connect it to the input of a different function block.

NOTE
CPU input markers can be used more than once within the same project.

6.6.3 Jump addresses

Jump addresses can essentially be used in the same way as CPU markers. They consist of a source jump address and a destination jump address. The destination jump address assumes the same value (1 or 0) as the associated source jump address without any delay time whatsoever – unless it is a logical loopback. This is what distinguishes jump addresses from CPU markers.

A maximum of 256 jump addresses can be used in a project.

Logical loopback

A logical loopback occurs when a function block input is connected to a destination jump address but the associated source jump address is linked to an output of the same function block or to an output of a function block with a higher function block index number. [2]

In this case, the logic result of the current logic cycle will only become available at the destination jump address during the subsequent logic cycle, i.e., it is subject to a delay that is equivalent to the logic execution time. If a jump address does cause a logical loopback, this will be indicated automatically by the appearance of an additional clock symbol at the destination jump address. The resulting delay time corresponds to the logic execution time.

WARNING
Response time extension due to the logical loopback

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

► The delay caused by the logical loopback must be factored in when calculating the response time and functionality.

[2] The function block index number is displayed at the top of each function block and indicates the position occupied by the function block within the execution sequence.

Using jump addresses

- ► Add a **source jump address** to the project using drag & drop. A dialog box appears in which the new source jump address has to be assigned a name. The name must be unique and can only be used once within the same project. A source jump address is usually connected to any function block output.
- ► Next add one or more **destination jump addresses** using drag & drop. A dialog box appears with a selection list of available source jump addresses. A source jump address can have multiple destination jump addresses within the same project. A destination jump address is usually connected to any function block input.

6.6.4 Validating the configuration

The configuration software checks the logic program automatically. If an error is detected, the configuration is marked as invalid and a warning appears in the toolbar and on the tab of the faulty logic page. Any function blocks that are not connected correctly are highlighted in red.

NOTE
The configuration software only checks the logic program for connection errors.

Until the configuration becomes valid, it will not be possible to start simulation mode or transfer the configuration to the Flexi Soft system.

Correcting an invalid configuration

- ► Connect any function block inputs that are currently disconnected. Correctly connected function blocks are shown in yellow.

WARNING
Inadequate safety check

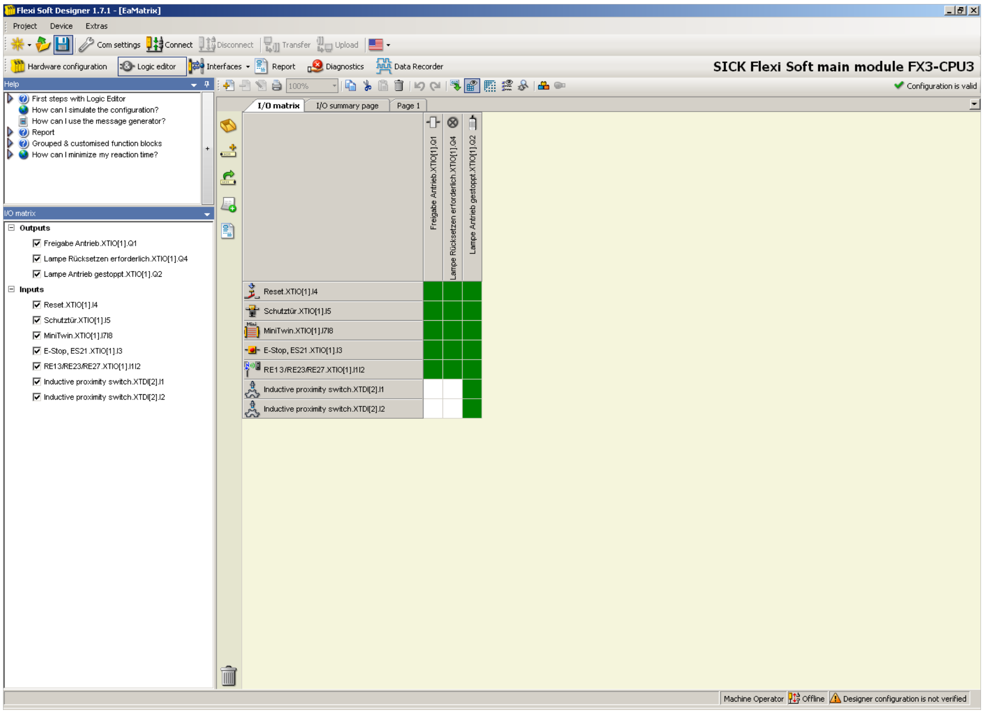
The target safety-related level may not be achieved in the event of non-compliance.

- ► After all connection errors have been rectified, the configuration must be thoroughly checked to ensure it corresponds to the risk analysis and the risk mitigation strategy and that it complies with all applicable standards and directives.

6.6.5 I/O matrix

The **I/O matrix** tab in the logic editor shows which inputs affect which outputs. This may be useful, for example, for checking whether the logic program is complete.

A green field indicates that the relevant input affects the relevant output; a white field indicates that there is no relationship between the respective input and output.



Flexi Soft Designer 1.7.1 - [EaMatrix]
Project
Device
Extras
Com settings
Connect
Disconnect
Transfer
Upload
Hardware configuration
Logic editor
Interfaces
Report
Diagnostics
Data Recorder
SICK Flexi Soft main module FX3-CPU3
Help
First steps with Logic Editor
How can I simulate the configuration?
How can I use the message generator?
Report
Grouped & customised function blocks
How can I minimize my reaction time?
100%
Configuration is valid
I/O matrix
I/O summary page
Page 1
Machine Operator
Offline
Designer configuration is not verified

I/O matrix

⊟ **Outputs**

- ☑ Freigabe Antrieb.XTIO[1].Q1
- ☑ Lampe Rücksetzen erforderlich.XTIO[1].Q4
- ☑ Lampe Antrieb gestoppt.XTIO[1].Q2

⊟ **Inputs**

- ☑ Reset.XTIO[1].I4
- ☑ Schutztür.XTIO[1].I5
- ☑ MiniTwin.XTIO[1].I7I8
- ☑ E-Stop, ES21.XTIO[1].I3
- ☑ RE13/RE23/RE27.XTIO[1].I1I2
- ☑ Inductive proximity switch.XTDI[2].I1
- ☑ Inductive proximity switch.XTDI[2].I2

	Freigabe Antrieb.XTIO[1].Q1	Lampe Rücksetzen erforderlich.XTIO[1].Q4	Lampe Antrieb gestoppt.XTIO[1].Q2
Reset.XTIO[1].I4	🟩	🟩	🟩
Schutztür.XTIO[1].I5	🟩	🟩	🟩
MiniTwin.XTIO[1].I7I8	🟩	🟩	🟩
E-Stop, ES21.XTIO[1].I3	🟩	🟩	🟩
RE13/RE23/RE27.XTIO[1].I1I2	🟩	🟩	🟩
Inductive proximity switch.XTDI[2].I1			🟩
Inductive proximity switch.XTDI[2].I2			🟩

Figure 34: I/O matrix in offline mode

All the inputs and outputs are listed in the **I/O matrix** window. Check the corresponding boxes to select which inputs and outputs should be displayed in the I/O matrix. In complex projects with a large number of inputs and outputs, this can be used to reduce the amount of information displayed to the most important aspects.

I/O matrix in simulation mode

In simulation mode (see "Simulating the configuration", page 238), the I/O matrix shows the values of the inputs and outputs used. Inputs and outputs with the value 1 are shown in green.

Clicking on an input switches its value between 1 and 0. This makes it possible to observe the effect of an input on the outputs.

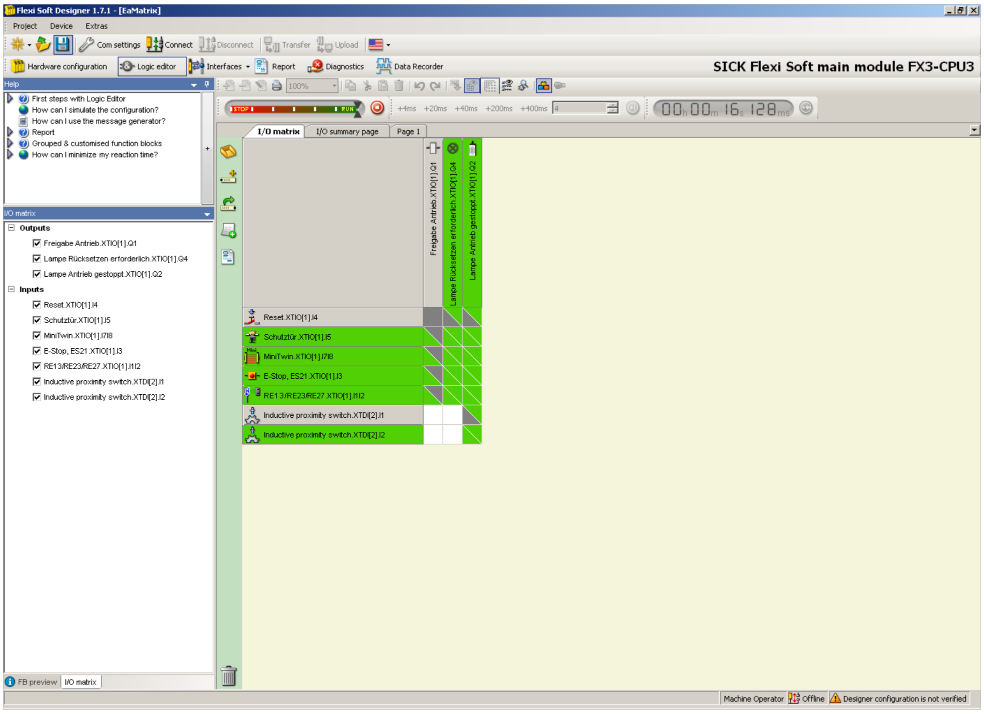

Flexi Soft Designer 1.7.1 - [EaMatrix]
Project
Device
Extras
Com settings
Connect
Disconnect
Transfer
Upload
Hardware configuration
Logic editor
Interfaces
Report
Diagnostics
Data Recorder
SICK Flexi Soft main module FX3-CPU3
Help
First steps with Logic Editor
How can I simulate the configuration?
How can I use the message generator?
Report
Grouped & customised function blocks
How can I minimize my reaction time?
100%
STOP
RUN
+4ms
+20ms
+40ms
+200ms
+400ms
00h 00m 16s 128ms
I/O matrix
I/O summary page
Page 1
Outputs
☑ Freigabe Antrieb.XTIO[1].Q1
☑ Lampe Rücksetzen erforderlich.XTIO[1].Q4
☑ Lampe Antrieb gestoppt.XTIO[1].Q2
Inputs
☑ Reset.XTIO[1].I4
☑ Schutztür.XTIO[1].I5
☑ MiniTwin.XTIO[1].I7I8
☑ E-Stop, ES21.XTIO[1].I3
☑ RE13/RE23/RE27.XTIO[1].I1I2
☑ Inductive proximity switch.XTDI[2].I1
☑ Inductive proximity switch.XTDI[2].I2
Freigabe Antrieb.XTIO[1].Q1
Lampe Rücksetzen erforderlich.XTIO[1].Q4
Lampe Antrieb gestoppt.XTIO[1].Q2
Reset.XTIO[1].I4
Schutztür.XTIO[1].I5
MiniTwin.XTIO[1].I7I8
E-Stop, ES21.XTIO[1].I3
RE13/RE23/RE27.XTIO[1].I1I2
Inductive proximity switch.XTDI[2].I1
Inductive proximity switch.XTDI[2].I2
FB preview
I/O matrix
Machine Operator
Offline
Designer configuration is not verified

Figure 35: I/O matrix in simulation mode

Export I/O matrix

You can export the I/O matrix as a CSV file, e.g., for documentation purposes.

- ► Click on the **Export I/O matrix as CSV file** button in the toolbar.
- ► Select a storage location, enter a file name, and click on Save. The I/O matrix is saved as a CSV file. The data can be viewed and evaluated using Microsoft Excel, for example.

Format of the CSV file:

- The first line contains the names of the outputs
- For each input, a line follows with the name of the input and its effect on the respective output in the first line
- “X” means: The input has an effect on the respective output.
- “–” means: The input has no effect on the respective output.
- The semicolon is used as the separator.

NOTE

The I/O matrix can also be exported while the Flexi Soft Station is online or in simulation mode. In this case, the values (0 or 1) of the effective inputs and outputs on which the respective input acts are also exported.

Example 1: I/O matrix without values

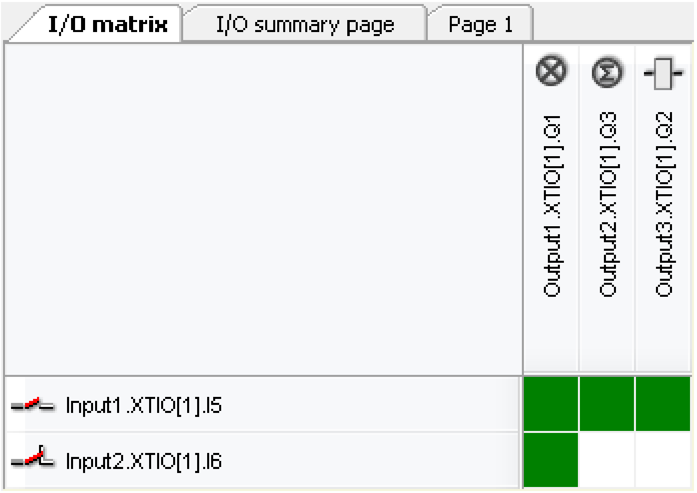


Figure 36: I/O matrix offline

Generated CSV file:

```
Inputs/Outputs     ;Output1.XTIO[1].Q1 ;Output2.XTIO[1].Q3 ;Output3.XTIO[1].Q2
Input1.XTIO[1].I5 ;X                   ;X                  ;X
Input2.XTIO[1].I6 ;X                   ;-                  ;-
```

Example 2: I/O matrix with values (system online or simulation mode)

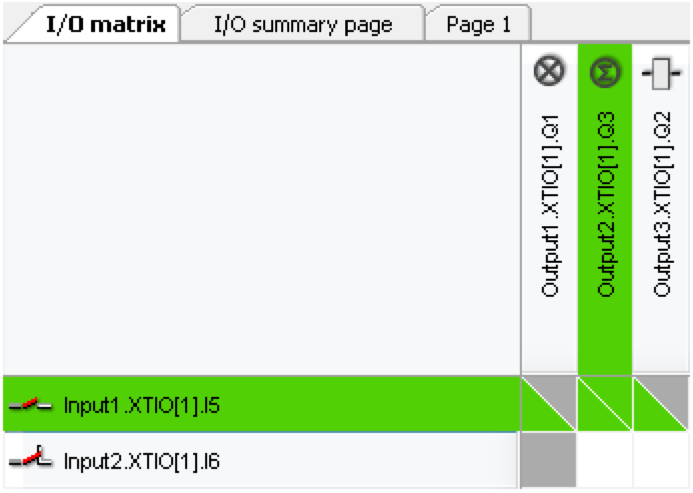


Figure 37: I/O matrix in simulation mode

Generated CSV file:

```
Inputs/Outputs     ;Output1.XTIO[1].Q1 ;Output2.XTIO[1].Q3 ;Output3.XTIO[1].Q2
Input1.XTIO[1].I5 ;X (1|0)             ;X (1|1)            ;X (1|0)
Input2.XTIO[1].I6 ;X (0|0)             ;-                  ;-
```

6.6.6 Logic access levels

Overview

The **Logic access level** function can be used to password protect individual pages of the logic editor. In this way, you can prevent unauthorized persons from making changes to logic pages or even looking at them.

The access levels are available for the following logics:

- CPU logic (access protection and visibility protection)
- MOC logic (access protection)

Logic access levels

Table 8: Access levels in the logic editor

Access level	Authorization
Logic access level 0 (not logged in)	• Permission to view and edit unprotected pages • Permission to view pages that are access protected but whose visibility has not been protected
Logic access level 1	• Permission to view and edit unprotected pages • Permission to view and edit pages that have been access protected and/or whose visibility has been protected with access level 1 • Permission to view protected pages that have been access protected with access level 2 and whose visibility has been protected up to access level 1 • Permission to set up access protection for unprotected pages (access level 1 only) • Permission to set up visibility protection for unprotected pages (access level 1 only) • Permission to remove access or visibility protection from protected pages (access level 1 only)
Logic access level 2 (administrator)	• Permission to view and edit all protected and unprotected pages • Permission to set up access protection at all levels • Permission to set up visibility protection at all levels • Permission to remove access or visibility protection at all levels • Permission to deactivate page protection completely

6.6.6.1 Activating access protection for logic page

Important information

!

NOTICE

Loss of password

The password cannot be reset or recovered, even by the SICK service.

► Make a note of the password and keep it safe.

Procedure

1. In the **Hardware configuration**, in the context menu of the main module, select the **Access change for logic pages** command.
2. Select the **Enable password protection for logic pages** option.
3. Assign passwords for logic access levels 1 and 2.
4. Click on **OK**.

6.6.6.2 Logging on to a logic access level

Procedure

Logging on to a logic access level

1. In the **Hardware configuration** in the context menu of the main module, select the **Change access for logic pages** command.
2. In the **Current logic access level** area, click on **Log in**.

3. In the login dialog, select the required logic access level, enter the password, and click on **Log in**.
4. Click on **OK**.

Logging out

1. In the **Hardware configuration** in the context menu of the main module, select the **Change access for logic pages** command.
2. In the **Current logic access level** area, click on **Log out**.
3. Click on **OK**.

6.6.6.3 Setting up access protection for a logic page

Procedure

Setting up access protection for a logic page

1. Open the desired page in the **Logic editor** view.
2. Right-click on the page, and in the context menu of the page select the **Set up access protection for logic page** submenu and the required logic access level.
3. If necessary, enter the password for the required logic access level and click on **Log in**.

Removing the access protection from a logic page

1. Open the desired page in the **Logic editor** view.
2. Right-click on the page, and in the context menu of the page select the **Remove access protection for logic page** command.
3. If necessary, enter the password for the required logic access level and click on **Log in**.

Complementary information

- Users who are not logged in with the required logic access level can see the content of the logic page in question but cannot edit it. In this case, the required logic access level will be displayed in red font.

6.6.6.4 Applying visibility protection to a logic page

Important information

NOTE

- Unverified projects that contain unverified logic pages with visibility protection cannot be transmitted to the Flexi Soft system and cannot be verified.
- When partial applications are exported, the logic access levels and passwords of the MOC logic are not exported at the same time. Therefore, you may have to set up password protection again after importing a partial application.
- Applications containing protected MOC logic pages can only be exported after logging into the relevant logic access level.

Procedure

Applying visibility protection to a logic page

1. Open the desired page in the **Logic editor** view.
2. Right-click on the page, choose the **Protect page visibility...** submenu in the context menu of the page, and select the required logic access level.
3. Enter the password for the desired logic access level where applicable and click on **Log in**.

Removing visibility protection from a logic page

1. Open the desired page in the **Logic editor** view.

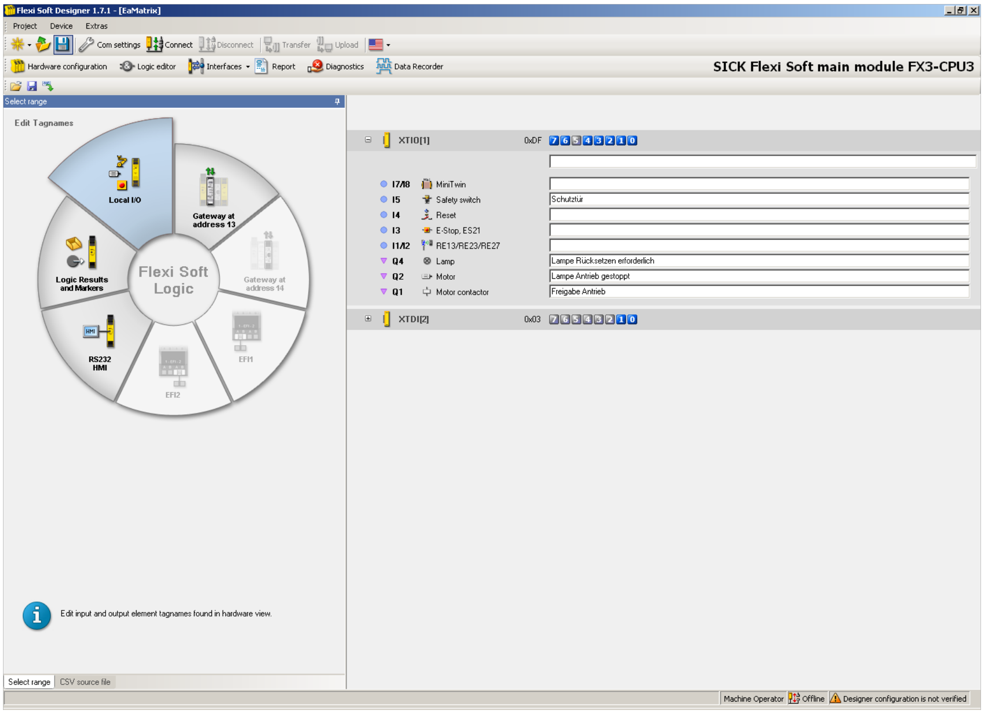
2. Right-click on the page and select the **Remove page visibility protection...** command in the context menu of the page.
3. Enter the password for the desired logic access level where applicable and click on **Log in**.

Complementary information

- Users who are not logged in with the required logic access level cannot see the content of the logic page in question or edit it. In this case, the required logic access level will be displayed in red font.

6.7 Tag name editor

The Tag name editor allows you to modify all of the tag names within a project. It can be opened either using the **Edit tag names** button in the **Hardware configuration**, or using the **Open dialog box for editing logic results** button in the logic editor toolbar.



XTIO[1] 0xDF 7 6 5 4 3 2 1 0

I7/I8	MiniTwin	
I5	Safety switch	Schutztür
I4	Reset	
I3	E-Stop, ES21	
I1/I2	RE13/RE23/RE27	
Q4	Lamp	Lampe Rücksetzen erforderlich
Q2	Motor	Lampe Antrieb gestoppt
Q1	Motor contactor	Freigabe Antrieb

XTDI[2] 0x03 7 6 5 4 3 2 1 0

Edit input and output element tagnames found in hardware view.

Figure 38: The tag name editor

The different types of tag name in the tag name editor

- **Logic Results and Markers**: Tag names for the main module in the logic editor
- **Local I/O**: Tag names for the input and output elements of expansion modules in the **Hardware configuration** view
- **Gateway at address 13/14**: Tag names for the input and output data sets of the gateways
- **EFI1/EFI2**: Tag names for the inputs and outputs of the devices connected to EFI interfaces 1 and 2
- **RS232 HMI**: Tag names for RS-232 inputs and outputs

The tag names that correspond to the selected type are listed in a tree view on the right-hand side of the screen.

If a particular type of tag name is not available in your project (e.g., if no EFI-enabled device is connected), the associated segment will be shown in gray, i.e., will be deactivated.

Editing tag names

- ► Select the segment associated with the tag you wish to modify.
- ► In the tree view on the right-hand side of the screen, select a bit and enter the desired tag name in the relevant input field.

6.7.1 Importing and exporting tag names

The **Import tag names** and **Export tag names** buttons on the top left-hand side of the tag name editor can be used to import the tag names from a CSV or Excel file, or save the tag names as a text file in CSV format.

Table 9: Buttons for importing and exporting tag names

Button	Meaning
📂	Import tag names
💾	Export tag names
⬇	Export to Pro-face GP-Pro EX

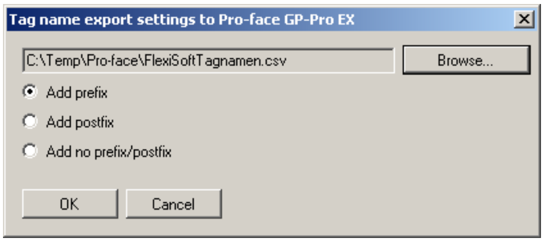
6.7.2 Exporting tag names for use in Pro-face GP-Pro EX

The **Export to Pro-face GP-Pro EX** button on the top left-hand side of the tag name editor can be used to export the tag names as a CSV file for subsequent use in Pro-face GP-Pro EX.

NOTE
In Pro-face GP-Pro EX, the tag names must not exceed the maximum length of 32 characters. Longer tag names will be truncated. This may result in several identical tag names when the tag names are exported. For this reason, the export process allows you to specify whether an additional prefix or postfix should be attached to each exported tag name. This serves as a unique identifier for the tag name.

Exporting the tag names for Pro-face GP-Pro EX

- ► Click on the **Export to Pro-face GP-Pro EX** button. The **Tag name export settings to Pro-face GP-Pro EX** window opens.
- ► Click on **Browse ....** The **Save as** window opens.
- ► Select the destination folder for the tag names, enter a file name and click on **OK** to close the **Save as** window again.



Tag name export settings to Pro-face GP-Pro EX

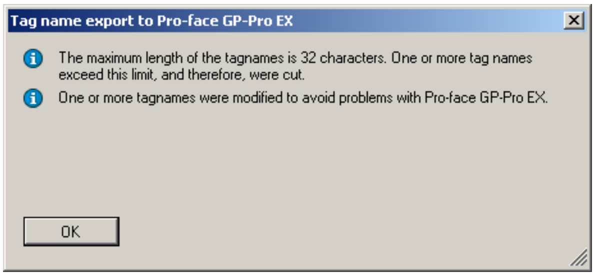
C:\Temp\Pro-face\FlexiSoftTagnamen.csv Browse...

● Add prefix
○ Add postfix
○ Add no prefix/postfix

OK Cancel

Figure 39: Tag name export settings for Pro-face GP-Pro EX

- ► Select whether a **prefix**, a **postfix**, or neither should be added to the tag names. The Pro-face HMI only supports tag names with a length of up to 32 characters. Therefore, tag names that exceed this limit will be truncated to the maximum length (including the prefix or postfix).
- ► Click on **OK** to start the export process. The tag names are saved as a CSV file using the selected file name. At the end of the export process, a dialog box appears to inform you of the result and, where applicable, any modifications that have been made to the exported tag names.



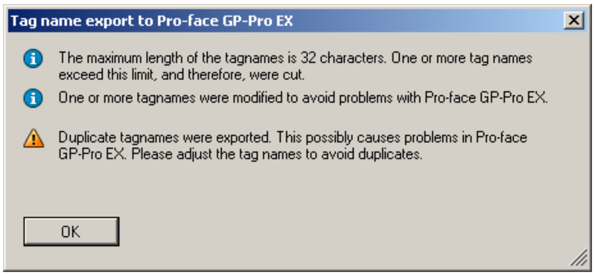
Tag name export to Pro-face GP-Pro EX

The maximum length of the tagnames is 32 characters. One or more tag names exceed this limit, and therefore, were cut.

One or more tagnames were modified to avoid problems with Pro-face GP-Pro EX.

Figure 40: Tag name export successful

If the software was unable to generate unique tag names, the following warning appears:



Tag name export to Pro-face GP-Pro EX

The maximum length of the tagnames is 32 characters. One or more tag names exceed this limit, and therefore, were cut.

One or more tagnames were modified to avoid problems with Pro-face GP-Pro EX.

Duplicate tagnames were exported. This possibly causes problems in Pro-face GP-Pro EX. Please adjust the tag names to avoid duplicates.

Figure 41: Warning displayed when identical tag names are exported

► Check the exported CSV file to see if this issue affects any of the tag names that are to be used in Pro-face. If it does, the following options are available:
   - Assign shorter tag names. The default tag name format can be configured in the **Hardware configuration** by going to **Settings** on the **Tag name format** tab. Now start a new export.

   Or:
   - Change the problematic tag names manually in the exported CSV file.

NOTE

As well the tag names, the alarm messages of the Flexi Soft system are also saved in the same folder during the export process. These are saved as CSV files in all the available languages. For this reason, we recommend using a separate folder when exporting tag names.

For further information about connecting a Pro-face HMI to a Flexi Soft system, please see the operating instructions titled “Flexi Soft Modular Safety Controller Hardware”.

For further information about using tag names and carrying out programming in Pro-face GP-Pro EX, please refer to the manual or the online help for Pro-face GP-Pro EX.

6.7.3 Coding Pro-face prefixes and postfixes

If the **Add prefix** or **Add postfix** option is selected when exporting the tag names, each tag name has a coded prefix or postfix added to it that identifies the data source of the tag name concerned. The following table describes the individual components that make up the prefix or postfix.

Table 10: Coding prefixes and postfixes

	Possible values			
Data type	**Station**	**Source**	**Byte, input or output**	**No. or bit**
Module status	A … D [1]	00 … 14 (no. of module in Flexi Soft station)	I (input I#)	1 … 8
			Q (output Q#)	1 … 4
EFI1 or EFI2	A … D	EFI1, EFI2	0 … 3	0 … 7
Flexi Soft to RS-232 (100 byte input) [2]	A … D	F2R	00 … 99	0 … 7
RS-232 to Flexi Soft (4 byte output) [2]	A … D	R2F	0 … 3	0 … 7
CPU type code array	A … D	CTYP	00 … 17	0 … 7
Expansion module type code array	A … D	MTYP	000 … 255	0 … 7
Operating data block	A … D	ODB	0 … 9	0 … 7
Checksum	A … D	CRC	00 … 19	0 … 7

[1] The station coding relates to Flexi Link. In the case of standalone systems, the station is always A.
[2] Information about configuring data exchange via the RS-232 interface: see "RS-232 routing", page 56.

NOTE
The Pro-face HMI only supports tag names with a length of up to 32 characters. Therefore, tag names that exceed this limit will be truncated to the maximum length (including the prefix or postfix).

Examples

- The prefix or postfix **A01I1** denotes station A, module 01, input I1.
- The prefix or postfix **AEFI100** denotes station A, connection EFI1, byte 0, bit 0.
- The prefix or postfix **BF2R023** denotes station B, RS-232 input, byte 02, bit 3.

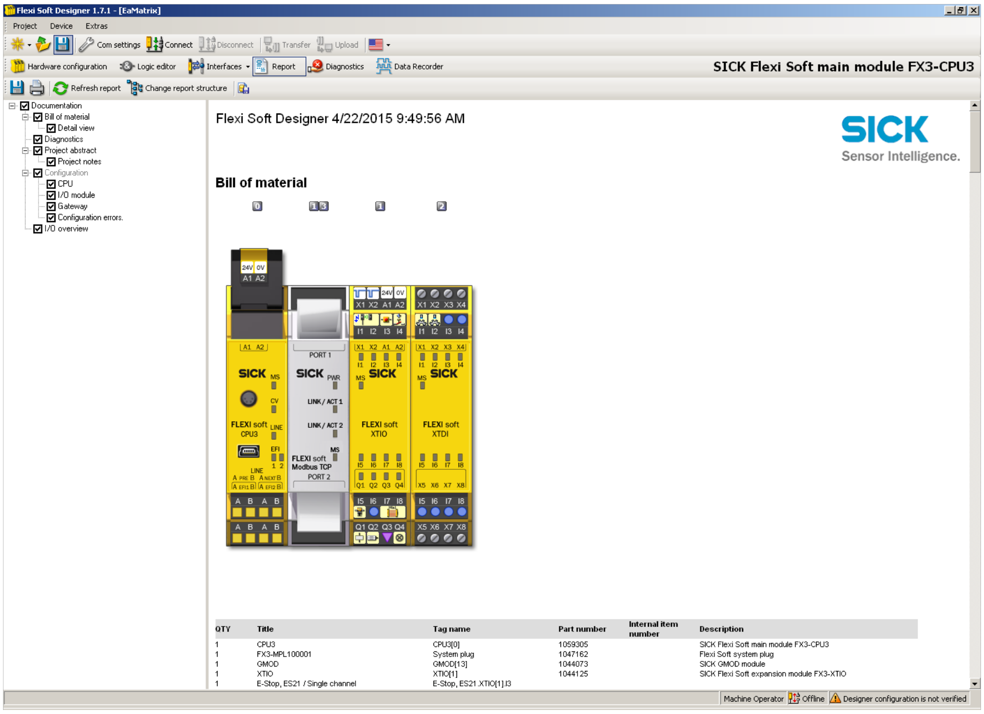
For further information about connecting a Pro-face HMI to a Flexi Soft system, please see the operating instructions titled “Flexi Soft Modular Safety Controller Hardware”.

For further information about using tag names and carrying out programming in Pro-face GP-Pro EX, please refer to the manual or the online help for Pro-face GP-Pro EX.

6.8 Report

Report

The **Report** view contains a clear summary of all the information for the project in question, including all configuration settings, the logic program, and detailed wiring information. You can customize the contents of the report.



☑ Documentation
☑ Bill of material
☑ Detail view
☑ Diagnostics
☑ Project abstract
☑ Project notes
☑ Configuration
☑ CPU
☑ I/O module
☑ Gateway
☑ Configuration errors.
☑ I/O overview

Flexi Soft Designer 4/22/2015 9:49:56 AM

Bill of material

QTY	Title	Tag name	Part number	Internal item number	Description
1	CPU3	CPU3[0]	1059305		SICK Flexi Soft main module FX3-CPU3
1	FX3-MPL100001	System plug	1047162		Flexi Soft system plug
1	GMOD	GMOD[13]	1044073		SICK GMOD module
1	XTIO	XTIO[1]	1044125		SICK Flexi Soft expansion module FX3-XTIO
1	E-Stop, ES21 / Single channel	E-Stop, ES21.XTIO[1].I3			

Figure 42: Report

On the left-hand side, there is an expandable selection list so that you can select which items of information you want to appear in the report. To select the items, check the relevant boxes.

The toolbar for the **Report** contains the following commands:

- **Save**: Saves the report on a data card in PDF format
- **Print**: Opens the report in PDF format. To use these functions, you must have a PDF viewing program (e.g., Acrobat Reader) installed on the computer.
- **Refresh report**: Updates the report after changing the report structure.
- **Change report structure**: Switches between a hardware-based and a function-based report structure

Complementary information

For detailed information about using the wiring information at the end of the report, see the operating instructions titled “Flexi Soft Modular Safety Controller Hardware”.

6.8.1 Create report

Procedure

1. Click on the **Report** button.
2. Use the **Change report structure** button to select one of the two available report structures (hardware-oriented or function-oriented).
3. In the selection list on the left-hand side, select the checkboxes for the components that you want to include in the report.
4. Once you have finished making your selections, click the **Refresh report** button on the toolbar. The report is now generated and displayed.

6.8.2 Save or print the report

Procedure

- ► To save the report as a PDF, click the **Save** button.
- ► To print the report, click the **Print** button. A preview of the report is generated and can then be printed.

6.8.3 Comparing projects

Overview

You can compare the configuration of the current project with a saved configuration.

Prerequisites

- The project must be saved with software version V1.9.0 or higher. Older projects must first be updated with a newer version.
- Recommendation: Set the same language in both projects.

Procedure

1. Click on the **Report** button.
2. Use the **Change report structure** button to select one of the two available report structures (hardware-oriented or function-oriented).
3. In the selection list on the left-hand side and check the boxes for the components that you want to include in the report.
4. Once you have finished making your selections, click the **Project comparison** button on the toolbar.
5. Select a saved **comparison project** in the file selection window.

✓ A report is created. Differences between the project are labeled in color.

6. Optional: You can hide and display parts of the project comparison with the boxes.

Complementary information

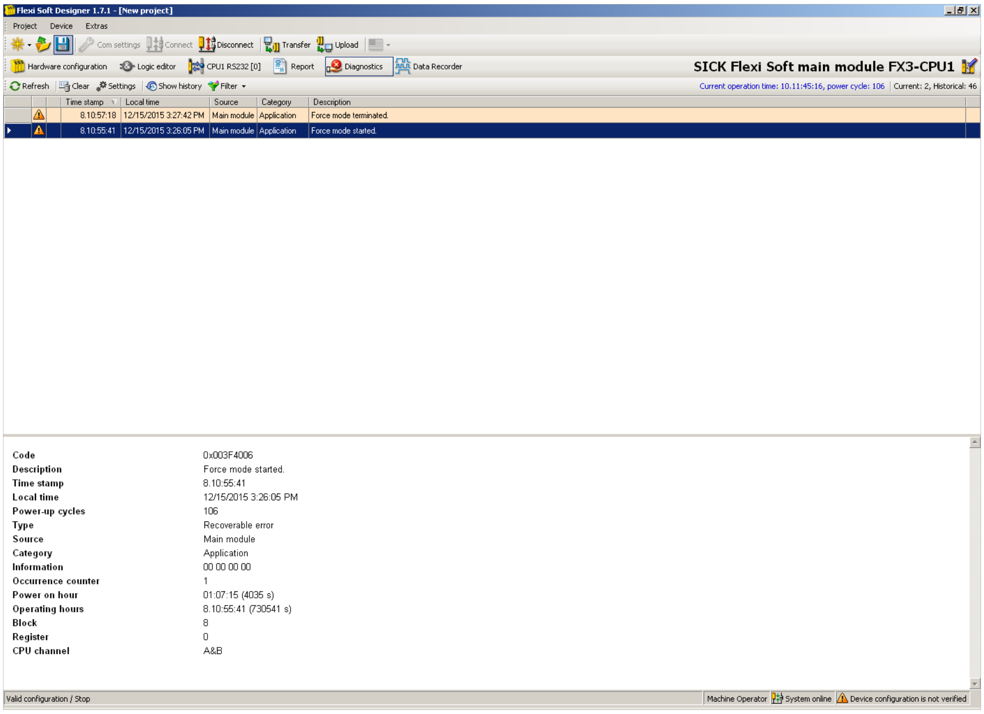
Meaning of colors:

- White: Identical in both projects
- Yellow: Change in current project
- Blue: Change in comparison project
- Green: Not available in current project
- Red: Only available in current project

6.9 Diagnostics

Diagnostics

Once the project has been completed and there is a connection to the Flexi Soft system in place, then you can carry out diagnostics on the system. The **Diagnostics** window contains a list of all the messages, information, warnings, and error messages in the system. Clicking an entry in the list displays the details of the selected message in the lower half of the window.


Flexi Soft Designer 1.7.1 - [New project]
Project
Device
Extras
Com settings
Connect
Disconnect
Transfer
Upload
Hardware configuration
Logic editor
CPU1 RS232 [0]
Report
Diagnostics
Data Recorder
SICK Flexi Soft main module FX3-CPU1
Refresh
Clear
Settings
Show history
Filter
Current operation time: 10.11:45:16, power cycle: 106
Current: 2, Historical: 46
Valid configuration / Stop
Machine Operator
System online
Device configuration is not verified

Time stamp	Local time	Source	Category	Description
8.10:57:18	12/15/2015 3:27:42 PM	Main module	Application	Force mode terminated.
8.10:55:41	12/15/2015 3:26:05 PM	Main module	Application	Force mode started.

Code	0x003F4006
Description	Force mode started.
Time stamp	8.10:55:41
Local time	12/15/2015 3:26:05 PM
Power-up cycles	106
Type	Recoverable error
Source	Main module
Category	Application
Information	00 00 00 00
Occurrence counter	1
Power on hour	01:07:15 (4035 s)
Operating hours	8.10:55:41 (730541 s)
Block	8
Register	0
CPU channel	A&B

Figure 43: Diagnostics view

Table 11: Meaning of diagnostic information

Keyword	Description
Code	Hexadecimal error code
Description	Error description
Time stamp	Total operating time of the main module up until the point when the error occurred (days:hours:minutes:seconds)
Local time	Time when the error occurred (system time of the computer). This value is not displayed in the case of historical errors.
Power-up cycles	Total number of times the main module has been powered up to date
Type	Error type (e.g., information, warning, recoverable error, serious error)
Source	Module that detected the error
Category	Module component that detected the error
Information	Internal information about the error
Occurrence counter	Number of times this error has occurred If an error occurs several times in succession, the most recent instance is saved and the occurrence counter incremented accordingly.
Power on hours	Operating time since the main module was last switched on. This value is reset whenever a restart is performed.
Operating hours	Total operating time of main module
Block	Diagnostic memory area in the main module 8 = RAM (volatile memory, error occurred during the current operating phase) 88 = EEPROM (non-volatile memory, error occurred during an earlier operating phase)
Register	Index number in diagnostic memory area
CPU channel	Internal hardware channel (A or B) of the module that detected the error

Complementary information

For a list of the most important error codes, possible causes, and possible troubleshooting measures, see the operating instructions titled “Flexi Soft Modular Safety Controller Hardware”.

6.9.1 Performing diagnosis

Procedure

► On the toolbar, click the **Diagnostics** button to open the **Diagnostics** window.

You have the following options:

► Click **Refresh** to read the current list of messages out from the system.
► **Clear** deletes all messages from the system. It is necessary to log in as an authorized client to do this.
► Automatic refreshing of diagnostics can be configured under **Settings**. In the **Diagnostic settings** window, select the **Refresh automatically** checkbox, and enter the desired refresh interval in seconds.
► The **Show history** button can be used to show or hide older messages that are still saved in the Flexi Soft system.
► The **Filter** drop-down menu makes it possible to show or hide certain types of message according to requirements. In the menu, click the various types of message to select them.

Complementary information

The Report function can be used to save or print the diagnostic messages.

Further topics

- "Save or print the report", page 76

6.10 Data recorder

Important information

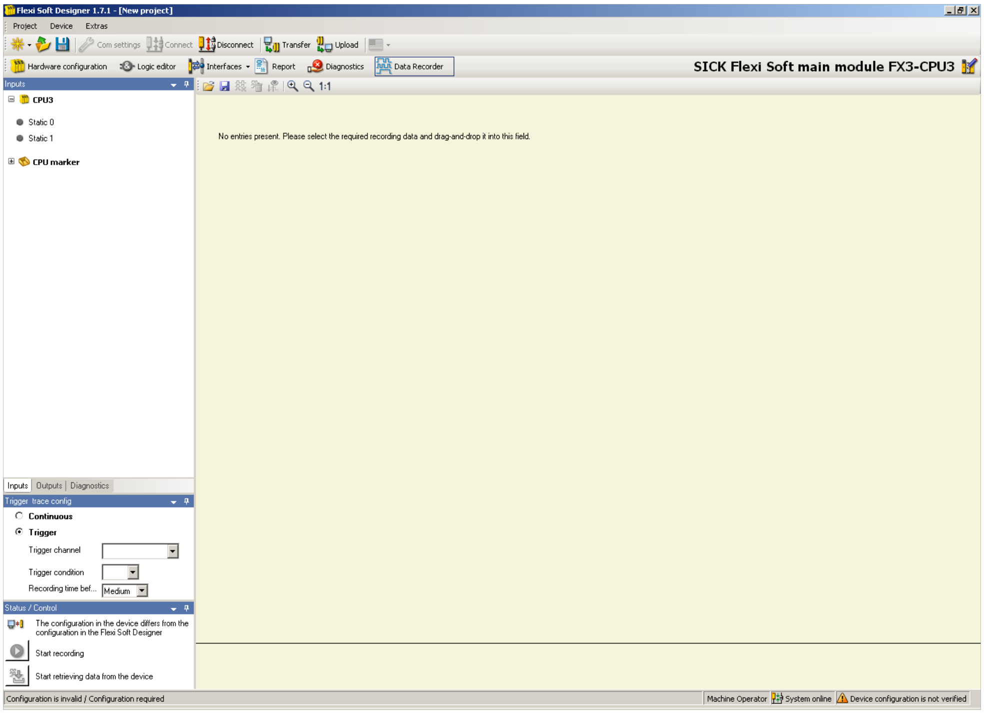
⚠ **WARNING**
Configuration, diagnostics or operation errors due to several simultaneous configuration connections

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

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

Data recorder

You can use the **Data Recorder** to record the input and output signals of a Flexi Soft system during operation. This function can be used, for example, to document the Flexi Soft system validation process or to troubleshoot a system that starts behaving strangely.



No entries present. Please select the required recording data and drag-and-drop it into this field.

○ Continuous

● Trigger

Trigger channel

Trigger condition

Recording time bef... Medium

The configuration in the device differs from the configuration in the Flexi Soft Designer

Configuration is invalid / Configuration required

Machine Operator | System online | Device configuration is not verified

Figure 44: Data recorder

The **Data Recorder** contains the following elements:

- Toolbar
- Window for selecting which input, output, and diagnostic data (“channels”) you want the data recorder to record.
- Window for configuring the data recorder (trigger and trace configuration). Recording can either commence right away or subject to the fulfillment of a trigger condition that you can configure here.
- **Status/Control** window for starting and stopping the recording process
- Visualization window, which displays the recorded input and output signals on a timescale. You can display markers for the purpose of measuring the time between two points of the recording.

6.10.1 Toolbar

Table 12: The data recorder toolbar

📂	Import recorded data from file
💾	Export recorded data to file
✕	Delete recorded data
🗑	Delete data recorder configuration
👁	Show or hide markers
🔍+	Zoom in
🔍−	Zoom out
1:1	Reset view to original size, i.e., the entire recording time is displayed

6.10.2 Status and control

Table 13: Data recorder status displays

	Recording status: Recording running
	Recording status: Waiting for trigger
	Recording status: Recording stopped
	Recording status: Data not yet retrieved from device
	Not connected
Ø	No data recorder configuration present
≠	Configuration for recording differs from configuration in device
≠	Project configuration differs from configuration in device
	Imported data displayed
⚠	Invalid data recorder configuration

Table 14: Data recorder controls

▶	Start recording
■	Stop recording
	Start retrieving data from the device

6.10.3 Data recording type

Overview

There are two types of data recording available: continuous recording and triggered recording.

Continuous recording

The data is recorded continuously. Older data in the device is overwritten by more recent data. Consequently, it only makes sense to use this setting if the Flexi Soft system is connected to Flexi Soft Designer and the Data Recorder view is open.

If the data is not retrieved quickly enough from the Flexi Soft system, partial data loss may occur. Lost data is highlighted on the display by means of the **Non-guaranteed values** status. This data is identified using a pale line.

To stop continuous recording, you must press the **Stop recording** button.

Triggered recording

Recording only commences when a configurable **trigger condition** is fulfilled.

Please be aware of the following when using trigger-based recording:

- Trigger-based recording does not require the Flexi Soft system to remain permanently connected to Flexi Soft Designer. Once recording has commenced, Flexi Soft Designer may be disconnected from the system.
- The Flexi Soft system records the data continuously, but only generates a snapshot once the trigger event actually occurs. A certain amount of data surrounding the trigger event is saved during this process.
- The length of the saved recording depends on how many channels and what types of channels are being recorded and the number of signal changes that occur. Non-Boolean data, for example, take up more memory space than Boolean data. When recording two speed values, for instance, 2,500 signal change points may be recorded with trigger-based recording – resulting in a recording time of at least 10 seconds. When recording four position values, 1,000 signal change points may be recorded – resulting in a recording time of at least four seconds.
- At the end of the recording process, the data recorder stops. If the trigger event occurs again, no new recording process is triggered. If you do want a new recording process to commence, you must first restart the data recorder using Flexi Soft Designer.
- The recording can be retrieved by pressing the **Start retrieving data from the device** button after you have re-established the connection to Flexi Soft Designer. This deletes it from the Flexi Soft system. A recording can therefore only be retrieved once.
  When a new recording commences, any previous recording that is still stored in the device will be deleted.

6.10.4 Configuring the data recorder

Overview

- You can record a maximum of 16 channels.
- In addition to recording data for the main module, you can record data for a maximum of two expansion modules (e.g., FX3-MOCx, FX3-ANA0).
- You can record a maximum of four channels per expansion module.
- You can record one analog value per FX3-ANA0.
- Maximum recording time per csv export: 60 min. For continuous recording, Flexi Soft Designer creates a new file for the trace data every 60 min.

Procedure

1. In the window for selecting **Inputs**, **Outputs**, and **Diagnostics**, select the channels whose values you want to record and drag them over to the visualization window.

NOTE

- All channels under **Inputs** and **Outputs**, plus all those marked with an “i” under **Diagnostics** are shown from the perspective of the main module. Changes to the values of these channels are only registered when they arrive at the main module.
- The channels that are shown from the perspective of the FX3-MOC0 modules can be found under **Diagnostics**.
- The data recorder records the values that are sent and received by the respective module (main module or FX3-MOC0). This may mean that more values are recorded than are processed by the logic program of the module. For example, very brief changes in the input signals are still recorded by the data recorder even though they are not registered by the logic.
- You can tell which values have been processed by the logic program by looking at the logic cycles. These are visualized by means of colored bars.

2. Go to the visualization window and sort the selected channels into the required order using drag & drop.
3. Channels can be removed by dragging and dropping them into the recycle bin.
4. Go to the **Trigger trace config** selection window and specify whether the recording should commence right away (**Continuous**) or only when a particular **trigger condition** is fulfilled.
   If the recording is to be started by a trigger condition, steps 5 to 7 must be followed as well.
5. Select the **Trigger channel** from the selection list. You can select any channel in the visualization window. The selected trigger channel is identified by a green arrow in the visualization window. By default, the first channel that you dragged into the visualization window is selected as the trigger channel.
6. Configure the **trigger condition**. This depends on the data type of the selected trigger channel:
   Possible trigger conditions for Boolean data:
   - Trigger channel is 1.
   - Trigger channel is 0.
   - Rising signal edge (transition from 0–1) on trigger channel.
   - Falling signal edge (transition from 1–0) on trigger channel.

   Possible trigger conditions for non-Boolean data:
   - Trigger channel value is below a configurable threshold.
   - Trigger channel value is above a configurable threshold.
   - Trigger channel value rises above a configurable threshold.
   - Trigger channel value drops below a configurable threshold.
7. Configure the **Recording time before trigger event.**
   This setting refers to the number of signal changes. The actual recording time prior to the trigger event (like the overall recording time) cannot be defined in advance. It depends on the:
   - Overall recording time
   - The number of signal changes between when the data recorder is started and when the trigger event occurs
   - The number of signal changes between when the trigger event occurs and when recording ends

Complementary information

Continuous recording: If you want to export the recorded data, then you need to select a storage location before starting the recording, see "Exporting data", page 83.

6.10.5 Deleting the data recorder configuration

Important information

NOTE

- The software immediately deletes the data recorder configuration in Flexi Soft Designer without displaying a confirmation prompt.
- Only the data recorder configuration in Flexi Soft Designer is deleted. If a data recorder configuration has been saved in the Flexi Soft system, this will be retained and you can reload it by pressing the **Start retrieving data from the device** button.

Procedure

► In the toolbar, click the **Delete data recorder configuration** button to delete the entire data recorder configuration.

6.10.6 Recording data

Requirements

- The Flexi Soft system must be in the Run status and be connected to Flexi Soft Designer.
- The configuration in Flexi Soft Designer must match the one saved in the Flexi Soft system.

Start recording

► Go to the **Status/Control** window and click the **Start recording** button. If continuous recording has been configured, recording commences immediately. Otherwise, recording will commence when the configured trigger condition is fulfilled.
If Flexi Soft Designer is connected to the Flexi Soft system during recording, the recorded data is displayed in the visualization window immediately.

NOTE
Whenever a recording process commences, any existing data in the RAM is deleted. If this data is still required, you must back it up with the Export function before starting a new recording process.

Stop recording

► Go to the **Status/Control** window and click the **Stop recording** button.

NOTE
Trigger-based recording stops automatically as soon as the available memory in the device is full.

6.10.7 Exporting data

Procedure

► Go to the toolbar and click the **Export trace data to file** button.
► Select a storage location, enter a file name, and click on **Save**. The current data in RAM is saved as a CSV file and can be viewed and evaluated using a program such as Microsoft Excel.

6.10.8 Importing data

Important information

ℹ **NOTE**
When data is imported, the current data in the data recorder gets overwritten and the data recorder configuration is deleted.

Procedure

► In the toolbar, click on the **Import trace data from file** button.
► Select the file that you want to import and click on **Open**. The saved data is loaded and displayed. The **Imported data** status is displayed in the **Status/Control** window.

6.10.9 Deleting data

Procedure

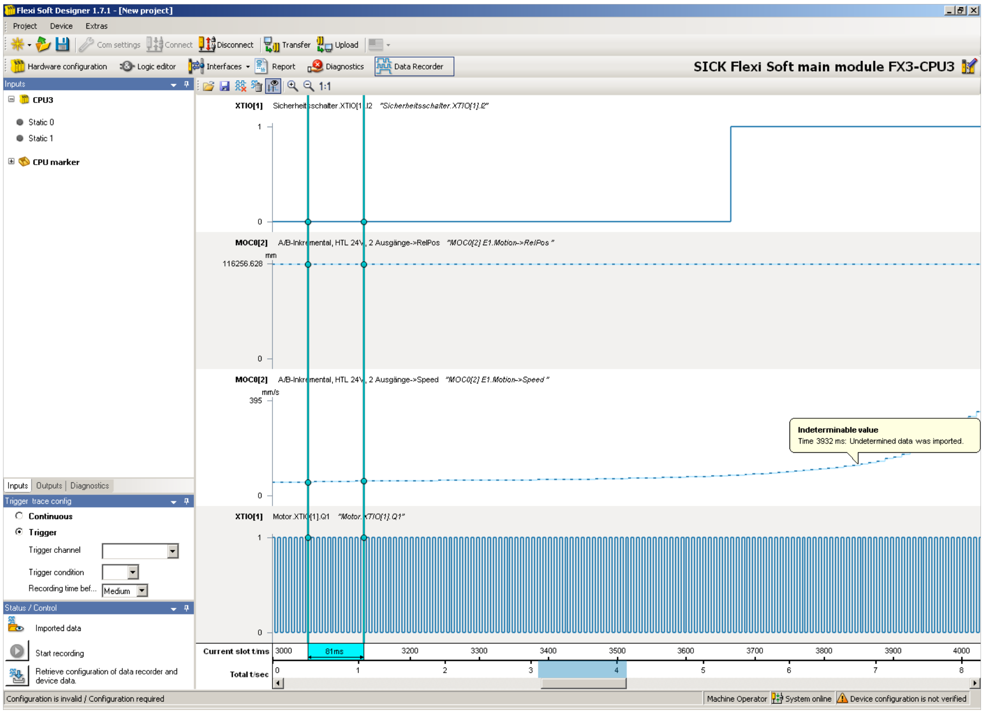
► In the toolbar, click on the **Delete recorded data** button to delete the data stored in RAM.

6.10.10 Visualizing data

The recorded data is displayed in the visualization window. The scale for the recorded values is located on the left of each channel. Non-Boolean data is automatically scaled to the available magnitude of the channel.

When you hover the mouse pointer over a point on one of the recorded curves, a pop-up window appears. This displays the exact point in time in milliseconds and the measured value.

If data is missing, e.g., because it could not be retrieved quickly enough from the device, the curve continues as normal but the status of the affected data is identified as **Non-guaranteed** by using a paler line. When you hover the mouse pointer over a measuring point of this kind, a pop-up window appears. This gives the reason for the **Non-guaranteed** status. (This information is lost when the data is exported, but the **Non-guaranteed** status is retained.)



Indeterminable value
Time 3932 ms: Undetermined data was imported.

○ Continuous
● Trigger
Trigger channel:
Trigger condition:
Recording time bef...: Medium

Configuration is invalid / Configuration required

Device configuration is not verified

Figure 45: Visualization of the recorded data in the data recorder

You can use the three zoom icons on the toolbar to zoom in, zoom out, or reset the view to its original size (entire recording time is displayed).

You can also access the zoom functions by right-clicking in the display area:

- ► Right-click and drag the mouse to the right to zoom in on the highlighted area.
- ► Right-click and drag the mouse to the left to reset the view to its original size.

In addition to displaying the recorded data, the data recorder also shows the duration of the logic cycles as vertical bars (depending on the zoom level). This enables you to work out which output signal has been affected by which input signal. The duration of the logic cycles may vary from one module to another. So that you can tell them apart more easily, the logic cycles of each module are displayed in different colors.

NOTE

It is the status of the inputs at the start of the logic cycle that determines what happens in the logic program from a processing perspective. This input status goes on to affect the status of the outputs at the end of the same logic cycle. If a signal change occurs at an input while the logic is being executed, this change cannot be applied to the outputs until the subsequent logic cycle. By contrast, a signal change at an input is registered in the recording by the data recorder at the exact moment when it occurs.

Consequently, changes to output signals cannot be interpreted based on the status of the concurrent input signals because these may have changed while the logic was being executed. Instead, you must look to see what the status of the input signals was at the start of the respective logic cycle.

There is a timescale underneath the channels. The time of the current slot is displayed above the scale and the total recording time appears below it. Depending on the zoom level, the currently displayed slot may also be shown as a blue area.

You can use the **Show/Hide markers** button on the toolbar to display two markers. You can move these with the mouse for the purpose of measuring the exact time between two points of the recording.

7 Logic programming in the main module

The function logic of the Flexi Soft system is programmed using function blocks. These function blocks are certified for use in safety-related functions provided that all the safety standards are observed during their implementation. The logic pages of the CPU logic have their own checksum.

7.1 Safety notes for logic programming

Standards and safety regulations

All safety-related parts of the system (wiring, connected sensors and control devices, configuration) must conform to the relevant standards (e.g., IEC 62061 or ISO 13849) and safety regulations.

WARNING
Incorrect configuration of the safety application

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

- ► Observe all applicable standards and safety regulations.
- ► Check that the operating principle of the Flexi Soft hardware and the logic program react in accordance with the risk avoidance strategy.
- ► Only use safety-related signals for safety-related applications.
- ► Always use the correct signal sources for the function blocks.

Safe value

The safe value of process data and outputs is 0 or Low and this is set when an error is identified.

WARNING
Inadequate safety measures

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

If the safe value (signal = Low) could lead to a dangerous state in the application, additional measures must be implemented. This applies in particular to inputs with signal edge detection.

- ► Analyze the status of the process data.
- ► Switch off the affected outputs if the status analysis detects an error.

Unexpected rising or falling signal edges

An error at an input can result in unexpected rising or falling signal edges (e.g. an interruption in network communication, a cable break at a digital input, a short-circuit at a digital input that is connected to a test output). The safe value remains set until the conditions for resetting the error have been met. For this reason, the affected signal may behave as follows:

- It temporarily switches to 1 instead of remaining set to 0 as it normally would in the fault-free status (rising signal edge and falling signal edge, i.e. 0–1–0).
- It temporarily switches to 0 instead of remaining set to 1 as it normally would in the fault-free status (falling signal edge and rising signal edge, i.e. 1–0–1).
- It remains set to 0 instead of switching to 1 as it normally would in the fault-free status.

WARNING
Unexpected rising or falling signal edges

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

- ► Take into account unexpected rising or falling signal edges.

Delays caused by CPU markers and jump addresses

CPU markers and jump addresses can extend the logic processing time and thus the response time. A CPU marker generally causes a delay that is equal to one logic cycle. A jump address can extend the logic execution time if it results in a logical loopback.

A logical loopback occurs when a function block input is connected to a destination jump address but the associated source jump address is linked to an output of the same function block or to an output of a function block with a higher function block index number. [3] In this case, the input comprises not the output values of the current logic cycle, but rather the output value of the previous logic cycle. This must be taken into account in terms of the functionality and, in particular, when calculating the response time.

Logical loopbacks can occur when using a CPU marker or a jump address.

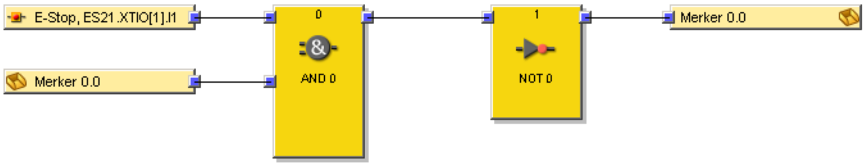


Figure 46: Logical loopback when using a CPU marker

A jump address only causes a delay of one logic cycle if it results in a logical loopback. In this case, the input of the jump address is displayed with a clock symbol (for Flexi Soft Designer ≥ V1.3.0).

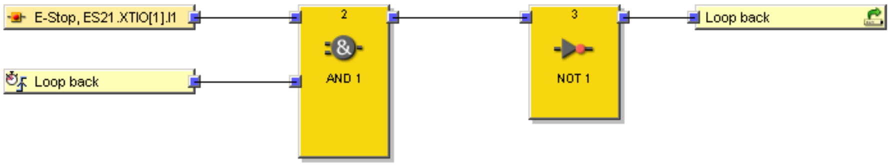


Figure 47: Logical loopback when using a jump address

WARNING
Extension of the response time

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

- ► When using CPU markers, take into account the extended logic execution time and the resultant increase in response time.
- ► The delays caused by the logical loopbacks must be factored in when calculating the response time and functionality.

[3] The function block index number is displayed at the top of each function block and indicates the position occupied by the function block within the execution sequence.

7.2 Function blocks in the main module

Overview

The Flexi Soft system uses function blocks to define the safety-related logic. There are two types of function block: logical function blocks and application-specific function blocks. The following table lists all the function blocks that are available in the main modules.

Function blocks in the main module

Table 15: Overview of the function blocks in the main module

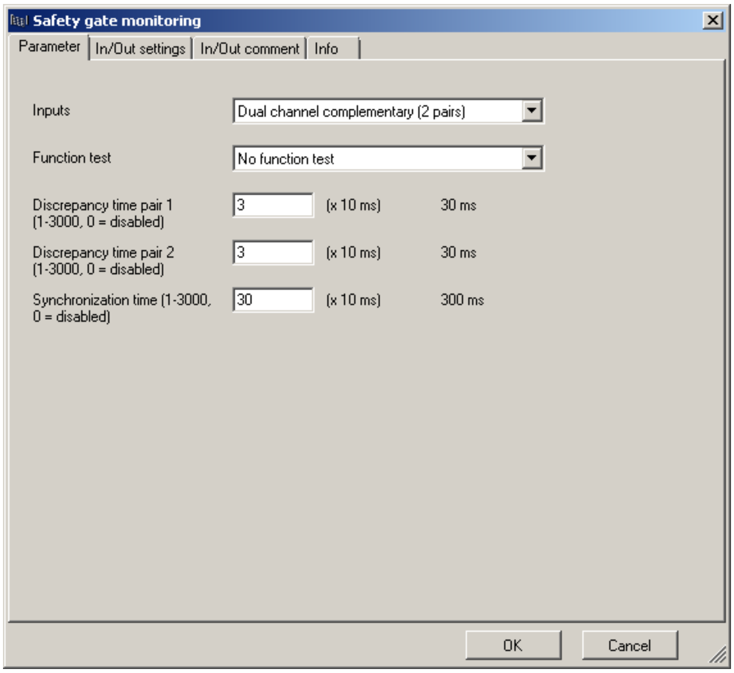
Logic	
• NOT (negation) • AND (conjunction) • OR (disjunction) • XOR (exclusive OR) • XNOR (exclusive NOR) • Multiple release • RS Flip-Flop	• JK Flip-Flop • Multiple memory • Binary decoder • Binary encoder • Routing 1:n (signal duplication) • Routing n:n (n inputs to n outputs in parallel)
Start/signal edge	
• Reset • Restart	• Start warning • Edge detection
Delays	
• Switch-on delay • Switch-off delay	• Adjustable switch-on delay timer • Adjustable switch-off delay timer
Event counter and clock	
• Event counter (up, down, up and down) • Clock generator	• Ramp down detection • Frequency monitoring • Message generator
EDM/output function blocks	
• External device monitoring • Valve monitoring	• Fast shut off with bypass • Fast shut off
Muting/presses	
• Sequential muting • Parallel muting • Cross muting • Universal press contact • Press single stroke	• Press setup • Press automatic • PSDI mode • Eccentric press contact
Other	
• Operating mode selector switch • Emergency stop • Safety gate monitoring • Magnetic switch • Light grid monitoring • Tolerant dual-channel evaluation	• Two-hand control type IIIA • Two-hand control type IIIC • Multi operator • Switch synchronization • Error output combination
Customized function blocks	
• Grouped function block	• Customized function block
Safety system function blocks SICK safety system function blocks requiring a license.	

Complementary information

A configuration can include up to 255 function blocks. The logic execution time is dependent on the type and number of function blocks used. Therefore, the number of function blocks used should always be kept as low as possible.

7.3 Configuring the function blocks

Most of the function blocks have configurable parameters. These vary from one function block to another. Double-clicking on a function block opens the configuration dialog of the function block. The following example shows the configuration dialog for the Safety gate monitoring function block:



Inputs: Dual channel complementary (2 pairs)

Function test: No function test

Discrepancy time pair 1 (1-3000, 0 = disabled): 3 (x 10 ms) 30 ms

Discrepancy time pair 2 (1-3000, 0 = disabled): 3 (x 10 ms) 30 ms

Synchronization time (1-3000, 0 = disabled): 30 (x 10 ms) 300 ms

Figure 48: Configurable function block parameters

Besides the input type (e.g., single-channel, dual-channel equivalent, etc.), function blocks may have other parameters that are defined on the function block properties page.

Depending on the function block, the configurable parameters can be found on either the **Parameter** or **I/O settings** tabs. The **I/O comment** tab can be used to replace the default names of the function block inputs and outputs with use-defined designations and, if required, also give the function blocks their own name. A description of the function block and its parameters can be found under **Info**.

7.3.1 Inputs of the function blocks

All the input elements listed in the input selection tree of the logic editor and all function block outputs can serve as possible signal sources for function block inputs.

7.3.2 Inverting inputs

The inputs of some function blocks can be inverted. At inverted inputs, the value 1 is evaluated as 0 and the value 0 as 1.

► Open the function block configuration dialog.
► Mark the desired input as **Inverted** and close the configuration dialog with **OK**.

Inverted inputs are marked with a white circle:

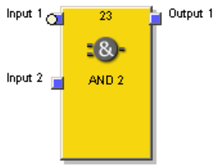


Figure 49: Example of an AND function block with inverted input 1

Function blocks with inverted inputs can include the following (for example):

- AND
- OR
- Routing n:n
- RS Flip-Flop
- JK Flip-Flop
- Switch synchronization

7.3.3 Outputs of the function blocks

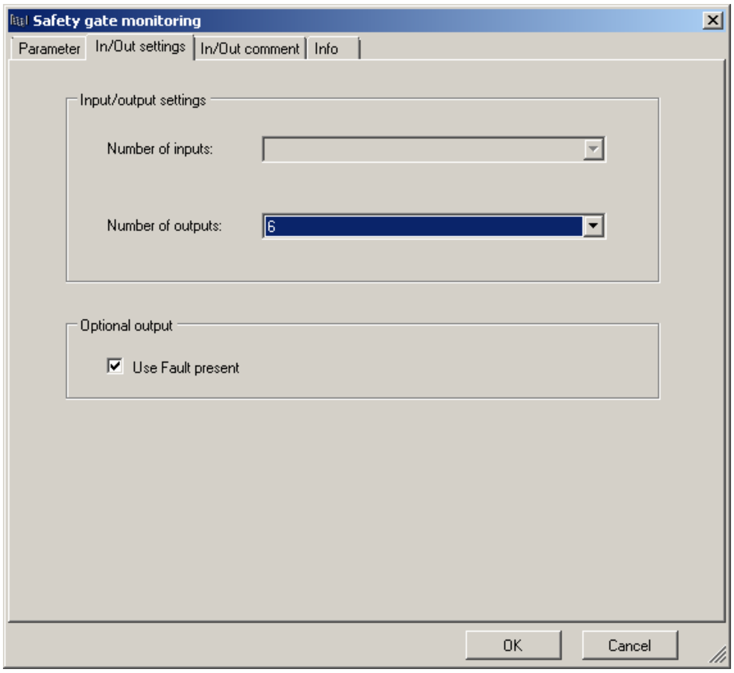
Function blocks provide various outputs for connection to physical outputs or to other function blocks.

The output of a function block can be connected to several downstream function blocks, but not to multiple output elements (physical outputs or EFI outputs). If a single function block is to be used to control several physical outputs, this the **Routing 1:n** function block must be used. The behavior of the outputs is explained in the descriptions of the individual function blocks.

Fault and diagnostic outputs can optionally be displayed. With the default configuration setting for the function blocks, only the **Enable** output and several other outputs are selected (e.g., reset required). Further error and diagnostic outputs can be activated in the configuration dialog of the relevant function block.

7.3.4 The Fault present output

Various function blocks have the optional **Fault present** output. This can be activated by checking the **Use Fault present** box on the **I/O settings** tab of the configuration dialog.


Safety gate monitoring

Input/output settings

Number of inputs:

Number of outputs: 6

Optional output

☑ Use Fault present

Figure 50: Activating the Fault present diagnostic output

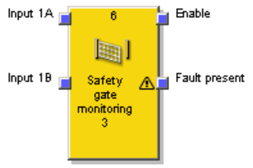


Figure 51: Fault present output

The Fault present output switches to 1 when a fault is detected on the basis of the configured function block parameters (e.g., discrepancy error, function test error, synchronization error, etc.). When the Fault present output is set to 1, the main output (e.g., the Enable output) is set to 0.

The Fault present output switches to 0 once all errors/faults have been reset. The conditions for resetting a fault/error are described in the section that deals with the function block concerned.

7.3.5 Changing the execution sequence

Overview

You can configure the execution sequence of the function blocks with this function.

Procedure

1. Right-click on the function block and select **Change the execution sequence...**.

✓ The dialog window opens.

2. Change the sequence with the **Forwards** / **Backwards** button. Or enter a number (a place in the sequence) to which the block is to be moved in the **New execution sequence** field.
3. Click **OK** to confirm.

✓ The function block is moved in the execution sequence.

Complementary information

- If the modified execution sequence is not possible, the software shows warnings that the execution sequence cannot be applied, or only with changes. You can suppress these warnings by deactivating the **Display warning when the desired execution sequence is not possible** checkbox.

7.4 Module data in the Logic editor

The following data is available in the logic editor of the main module:

- Input and output data of all modules of the Flexi Soft system
- Diagnostic status bits of all modules of the Flexi Soft system
- Input and output data of the connected EFI-enabled devices
- Diagnostic status bits of the connected EFI-enabled devices

NOTE

The inputs and outputs are color-coded according to their function:

- Gray: Non-safe
- Yellow: Safe
- Blue: Diagnostics

7.4.1 Inputs

The **Inputs** selection window contains the inputs of the modules and the connected EFI-enabled devices. These can be used as inputs for the logic program.

Static 0 and Static 1

The **Inputs** selection window of the main module contains the inputs **Static 0** and **Static 1**.

The **Static 0** input can be used to set a function block input permanently to 0. Similarly, the **Static 1** input can be used to set a function block input permanently to 1. This might be necessary, for example, to achieve a valid logic configuration if the relevant function block contains inputs that are not required but cannot be deactivated.

7.4.2 Outputs

The **Outputs** selection window contains the outputs of the modules and the connected EFI-enabled devices. These can be controlled using the logic program.

Logic results

The **Outputs** selection window also contains the user-defined output bits for the **logic results**. These can be used to forward the results of the logic program to other controllers via a network, e.g., with the help of a gateway or via RS-232, Flexi Line, or Flexi Link.

7.4.3 Module status bits

The status bits for the modules can be found in the **Diagnostics** selection window. These can also be used as logic inputs.

7.4.3.1 Module status bits of the main module

The following module status bits can be found under **Diagnostics** in the Logic editor in the main module.

Table 16: Module status bits of the main module

Module status bit	Description
Configuration is valid	0 = Configuration is invalid 1 = Configuration is valid
Module supply voltage OK	0 = Supply voltage outside the specified range 1 = Supply voltage OK
Verify status	0 = Configuration is not verified 1 = Configuration is verified (CV LED of main module lights up a steady yellow)
First logic cycle	This module status bit is set to 1 during the very first logic cycle that is performed after each transition from Stop to Run. Throughout all other logic cycles, it remains set to 0. This bit can be used to trigger initialization functions in the logic program.
Simulation bit	0 = Simulation mode is deactivated 1 = Simulation mode is active

EFI status bits

These status bits are only available on main modules with EFI interfaces, i.e., FX3-CPU1 or later. The status bits **EFI1 status** and **EFI2 status**, and their equivalents **EFI1 OK** and **EFI2 OK**, are located in the **Diagnostics** window of the Logic editor, under the main module.

Table 17: Meaning of the EFI status bits in the logic editor

Value	Description
0	• Communication with one or more EFI-enabled devices is not yet possible. **Or:** • There is an error affecting communication with one or more EFI-enabled devices.

Value	Description
1	• EFI input and output process data is being successfully exchanged with all the EFI-enabled devices as per the EFI interface configuration (EFI1 or EFI2). **Or:** • No device has been configured for this EFI connection.

NOTE
With AOPD senders, no process data is exchanged via EFI. Consequently, no communication errors can occur in relation to these devices, i.e., interruptions in communication cannot be detected.

Flexi Link status bits

In a Flexi Link system, additional status bits are available on the main module (see "Flexi Link status bits", page 478).

Flexi Line status bits

In a Flexi Line system, additional status bits are available on the main module (see "Flexi Line status bits", page 486).

7.4.3.2 Input data status and output data status

Important information

NOTE
The **Input data status** and **Output data status** diagnostics bits for the FX3-XTIO and FX3-XTDI modules are available with firmware version ≥ V2.00.0.

Input data status and output data status

The **Input data status** and **Output data status** diagnostic bits of the connected gateways and expansion modules are available under **Diagnostics** in the logic editor and can be used as inputs for the logic program. In some applications, it may be important to evaluate this status information to determine the behavior of the logic functions performed by the safety controller.

The status indicates the following:

- The data transferred from a connected device into the main module are 0 because this is the initial value on the connected device.
- The data transferred from a connected device into the main module are 0 because there is an error on the connected device.

Table 18: Description of the diagnostic bits

Diagnostic bit	Value	Description
Input data status	0	One or more input bits of the associated module have been set to 0 because an error has been detected (e.g., cross-circuit or communication error detected). This means that the values of the input bits may be different from those that would normally occur during error-free operation.
	1	The inputs of the associated module are free of errors.
Output data status	0	An error has been detected at one or more outputs of the associated module (e.g., overload detected, short-circuit detected, or communication error detected). This means that the values of the outputs may be different from those that would normally occur during error-free operation.
	1	The outputs of the associated module are free of errors.

The refresh rate of the **Input data status** and **Output data status** diagnostics bits corresponds to the refresh rate of the process data for the module.

7.4.3.3 Module status bits of the expansion modules

Overview

The module status bits for the expansion modules contain diagnostics data. This data is refreshed approximately every 200 ms. Due to the longer refresh interval, this data may not be consistent with the latest process data for the module.

Important information

⚠ **WARNING**
Non-secure or inconsistent data

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

- ► Only use secure data for safety-related applications.
- ► Only use the module status bits of the expansion modules for diagnostic purposes.

I/O module FX3-XTIO

Table 19: Module status bits of the FX3-XTIO I/O module

Module status bit	Description
Configuration is valid	0 = Configuration invalid 1 = Configuration valid
Module supply voltage OK	0 = Supply voltage outside the specified range 1 = Supply voltage OK
Fast shut-off control is OK	0 = Error or timeout in fast shut-off logic 1 = Fast shut-off logic OK
Inputs I1 / I2 dual channel evaluation is OK	0 = Discrepancy error input I1 / I2 1 = Dual channel evaluation input I1 / I2 OK
Inputs I3 / I4 dual channel evaluation is OK	0 = Discrepancy error input I3 / I4 1 = Dual channel evaluation input I3 / I4 OK
Inputs I5 / I6 dual channel evaluation is OK	0 = Discrepancy error input I5 / I6 1 = Dual channel evaluation input I5 / I6 OK
Inputs I7 / I8 dual channel evaluation is OK	0 = Discrepancy error input I7 / I8 1 = Dual channel evaluation input I7 / I8 OK
Input I1 … I8 OK	0 = Error at input 1 = Input OK
Output Q1 … Q4 OK. Testing deactivated	0 = Error at output 1 = Output OK
Output Q1 … Q4 OK. Testing active	0 = Error at output 1 = Output OK

I/O module FX3-XTDI

Table 20: Module status bits of the FX3-XTDI I/O module

Module status bit	Description
Configuration is valid	0 = Configuration invalid 1 = Configuration valid
Inputs I1 / I2 dual channel evaluation is OK	0 = Discrepancy error input I1 / I2 1 = Dual channel evaluation input I1 / I2 OK
Inputs I3 / I4 dual channel evaluation is OK	0 = Discrepancy error input I3 / I4 1 = Dual channel evaluation input I3 / I4 OK

Module status bit	Description
Inputs I5 / I6 dual channel evaluation is OK	0 = Discrepancy error input I5 / I6 1 = Dual channel evaluation input I5 / I6 OK
Inputs I7 / I8 dual channel evaluation is OK	0 = Discrepancy error input I7 / I8 1 = Dual channel evaluation input I7 / I8 OK
Input I1 … I8 OK	0 = Error at input 1 = Input OK

I/O module FX3-XTDS

Table 21: Module status bits of the FX3-XTDS I/O module

Module status bit	Description
Configuration is valid	0 = Configuration invalid 1 = Configuration valid
Module supply voltage OK	0 = Supply voltage outside the specified range 1 = Supply voltage OK
Output current is OK	0 = Output current outside the specified range 1 = Output current OK
Inputs I1 / I2 dual channel evaluation is OK	0 = Discrepancy error input I1 / I2 1 = Dual channel evaluation input I1 / I2 OK
Inputs I3 / I4 dual channel evaluation is OK	0 = Discrepancy error input I3 / I4 1 = Dual channel evaluation input I3 / I4 OK
Inputs I5 / I6 dual channel evaluation is OK	0 = Discrepancy error input I5 / I6 1 = Dual channel evaluation input I5 / I6 OK
Inputs I7 / I8 dual channel evaluation is OK	0 = Discrepancy error input I7 / I8 1 = Dual channel evaluation input I7 / I8 OK
Input I1 … I8 OK	0 = Error at input 1 = Input OK

I/O module FX0-STIO

Table 22: Module status bits of the FX0-STIO I/O module

Module status bit	Description
Configuration is valid	0 = Configuration invalid 1 = Configuration valid
Module supply voltage OK	0 = Supply voltage outside the specified range 1 = Supply voltage OK
Output current is OK	0 = Output current outside the specified range 1 = Output current OK

Motion Control FX3-MOC0

See "Module status bits of the FX3-MOC0", page 256.

Motion Control FX3-MOC1

See "Module status bits of the FX3-MOC1", page 322.

Analog input module FX3-ANA0

See "Module status bits of the FX3-ANA0", page 449.

Gateways

See the operating instructions titled “Flexi Soft Gateways in the Flexi Soft Designer Configuration Software” (SICK part number 8012483).

7.4.4 EFI I/O error status bits in the logic editor

For each connected EFI-enabled device/each Flexi Link station, there is an I/O error status bit available under the relevant EFI-enabled device on the **Inputs** tab of the Logic editor. This can be used as an input for the logic program. The I/O error status bit is set to 1 if the data or the process image of the connected EFI-enabled device/Flexi Link station has been set to 0. This may be the case if, for example, an error has been detected, or if the Flexi Link station is in the Stop status or is being reconfigured.

Table 23: Meaning of the EFI I/O error status bits

Status bit	Value	Meaning
I/O error	0	The associated EFI-enabled device or the Flexi Link station is free of errors (e.g., is in the Run status).
	1	The process image of the associated EFI-enabled device or of the Flexi Link station has been set to 0 for one of the following reasons: • Error detected in EFI-enabled device • Flexi Link station not in the Run status • A suspended Flexi Link station has been found. • A Flexi Link station with a divergent Flexi Link ID has been found.

Additional information: see "Flexi Link status bits", page 478.

7.5 Time values and logic execution time

WARNING
Malfunction 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.

► Configure the monitoring functions with suitable times.
► Carefully check the configured monitoring functions.

Time values for the monitoring of the discrepancy time, synchronization time, pulse duration, muting time, etc. must meet the following conditions:

- The monitoring time must exceed the logic execution time.
- The times have an accuracy of ± 10 ms (evaluation plus logic execution time).

The logic execution time is dependent on the type and number of function blocks used. It is a multiple of 4 ms. Under **FB group info** in the logic editor, the logic execution time as well as the percentage of this time that has actually been used are displayed. The logic execution time specified has an accuracy of ± 100 ppm (parts per million). If the amount of time used exceeds 100% of the logic execution time, the logic execution time is automatically increased by 4 ms.

7.6 Logical function blocks

7.6.1 NOT

Function block diagram

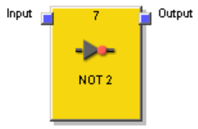


Figure 52: Inputs and outputs of the NOT function block

General description

The value at the output is the inverted value of the input. If, for example, the input is set to 1, the output is set to 0.

Truth table

Table 24: Truth table for the NOT function block

Input	Output
0	1
1	0

7.6.2 AND

Function block diagram

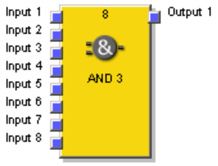


Figure 53: Inputs and outputs of the AND function block

General description

The output is set to 1 when all the evaluated inputs are 1. Up to eight inputs are evaluated.

Function block parameters

Table 25: Parameters of the AND function block

Parameter	Possible values
Number of inputs	2 to 8
Invert input x	Every input of this function block can be inverted. At an inverted input, a 0 works in the same way as a 1 and vice versa.

Truth tables

The following explanations apply to the truth tables in this section:

- “x” signifies “any” (0 or 1).

NOTE
Truth tables apply when the function blocks are configured without inverted inputs.

Table 26: Truth table for AND evaluation with two inputs

Input 1	Input 2	Output
0	x	0
x	0	0
1	1	1

Table 27: Truth table for AND evaluation with eight inputs

Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	Output
0	x	x	x	x	x	x	x	0
x	0	x	x	x	x	x	x	0
x	x	0	x	x	x	x	x	0
x	x	x	0	x	x	x	x	0
x	x	x	x	0	x	x	x	0
x	x	x	x	x	0	x	x	0
x	x	x	x	x	x	0	x	0
x	x	x	x	x	x	x	0	0
1	1	1	1	1	1	1	1	1

7.6.3 OR

Function block diagram

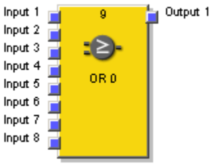


Figure 54: Inputs and outputs of the OR function block

General description

The output is set to 1 when any of the evaluated inputs are 1. Up to eight inputs are evaluated.

Function block parameters

Table 28: Parameters of the OR function block

Parameter	Possible values
Number of inputs	2 to 8
Invert input x	Every input of this function block can be inverted. At an inverted input, a 0 works in the same way as a 1 and vice versa.

Truth tables

The following explanations apply to the truth tables in this section:

- “x” signifies “any” (0 or 1).

NOTE
Truth tables apply when the function blocks are configured without inverted inputs.

Table 29: Truth table for OR evaluation with two inputs

Input 1	Input 2	Output
0	0	0
1	x	1
x	1	1

Table 30: Truth table for OR evaluation with eight inputs

Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	Output
0	0	0	0	0	0	0	0	0
1	x	x	x	x	x	x	x	1
x	1	x	x	x	x	x	x	1
x	x	1	x	x	x	x	x	1
x	x	x	1	x	x	x	x	1
x	x	x	x	1	x	x	x	1
x	x	x	x	x	1	x	x	1
x	x	x	x	x	x	1	x	1
x	x	x	x	x	x	x	1	1

7.6.4 XOR (exclusive OR)

Function block diagram

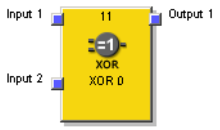


Figure 55: Inputs and outputs of the XOR function block

General description

The output is only set to 1 if the two inputs are complementary (i.e., the values are opposites; one input is 1 and one input is 0).

Truth table for the XOR function block

Table 31: Truth table for XOR evaluation

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

7.6.5 XNOR (exclusive NOR)

Function block diagram

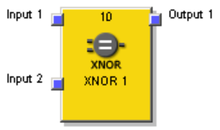


Figure 56: Inputs and outputs of the XNOR function block

General description

The output is only set to 1 if the two inputs are equivalent (i.e., they have the same values; both inputs are either 1 or 0).

Truth table for the XNOR function block

Table 32: Truth table for XNOR evaluation

Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	1

7.6.6 Multiple release

Function block diagram

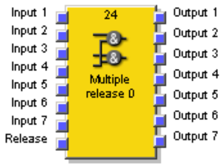


Figure 57: Inputs and outputs of the Multiple release function block

General description

Using the Multiple release function block, up to 7 inputs can be logically ANDed with the **Release** input (7 times AND).

Function block parameters

Table 33: Parameters of the Multiple release function block

Parameter	Possible values
Number of inputs (not including Release input)	1 to 7
Invert input x	Every input of this function block can be inverted. At an inverted input, a 0 works in the same way as a 1 and vice versa.
Invert Release	

Truth table for the Multiple release function block

NOTE
Truth tables apply when the function blocks are configured without inverted inputs.

Table 34: Truth table for the Multiple release function block

Release	Output x
0	0
1	Input x

7.6.7 RS Flip-Flop

Function block diagram

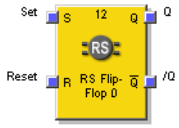


Figure 58: Inputs and outputs of the RS Flip-Flop function block

General description

The RS Flip-Flop function block saves the most recent value for the **Set** or **Reset** inputs. It is used as a simple memory cell. **Reset** has a higher priority than **Set**. If the most recent value for **Set** was 1, output **Q** is 1 and output /**Q** (Q inverted) is 0. If the most recent value for the **Reset** input was 1, output **Q** is 0 and output /**Q** is 1.

Function block parameters

Table 35: Parameters of the RS Flip-Flop function block

Parameter	Possible values
Invert Set	Every input of this function block can be inverted. At an inverted input, a 0 works in the same way as a 1 and vice versa.
Invert Reset	

Truth table for the RS Flip-Flop function block

The following explanations apply to the truth table in this section:

- "n–1" refers to the previous value.
- "n" refers to the current value.
- "x" signifies "any" (0 or 1).

NOTE
Truth tables apply when the function blocks are configured without inverted inputs.

Table 36: Truth table for the RS Flip-Flop function block

Set	Reset	Output Q $_{n-1}$	Output Q $_{n}$	Output /Q $_{n}$
0	0	0	0	1
0	0	1	1	0
0	1	x	0	1
1	0	x	1	0
1	1	x	0	1

7.6.8 JK Flip-Flop

Function block diagram

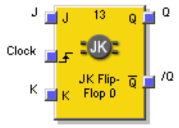


Figure 59: Inputs and outputs of the JK Flip-Flop function block

General description

The JK Flip-Flop function block has three inputs. The **J** and **K** inputs only affect the outputs if a rising signal edge is detected at the **PSDI** input:

- If input **J** is 1 and input **K** is 0, output **Q** switches to 1 and output/**Q** (= Q inverted) switches to 0.
- If input **J** is 0 and input **K** is 1, output **Q** switches to 0 and output/**Q** switches to 1.
- If both inputs are 0, outputs **Q** and/**Q** remain set to the most recent value.
- If both inputs are 1, the outputs switch over, i.e., their most recent values are inverted.

Function block parameters

Table 37: Parameters of the JK Flip-Flop function block

Parameter	Possible values
Number of outputs	• 1 (Q) • 2 (Q and /Q)
Invert J	Every input of this function block can be inverted. At an inverted input, a 0 works in the same way as a 1 and vice versa.
Invert Clock	
Invert K	

Truth table for the JK Flip-Flop function block

The following explanations apply to the truth table in this section:

- "↑" signifies that a rising signal edge has been detected at the input.
- "↓" signifies that a falling signal edge has been detected at the input.
- "n−1" refers to the previous value.
- "n" refers to the current value.
- "x" signifies "any" (0 or 1).

NOTE
Truth tables apply when the function blocks are configured without inverted inputs.

Table 38: Truth table for the JK Flip-Flop function block

J	K	Clock	Output Q_{n-1}	Output Q_n	Output $/Q_n$
x	x	0, 1, or ↓	0	0	1
x	x	0, 1, or ↓	1	1	0
0	0	↑	0	0	1
0	0	↑	1	1	0
0	1	↑	0	0	1
0	1	↑	1	0	1
1	0	↑	0	1	0
1	0	↑	1	1	0
1	1	↑	0	1	0
1	1	↑	1	0	1

7.6.9 Multiple memory

Function block diagram

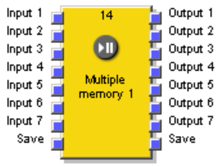


Figure 60: Inputs and outputs of the Multiple memory function block

General description

Depending on the **Save** input, the status of up to 7 inputs can be forwarded or saved at the respective outputs.

If the **Save** input is set to 0, the status of inputs 1 to 7 is forwarded to outputs 1 to 7 unchanged.

If the **Save** input switches from 0 to 1, the current status of inputs 1 to 7 is saved and continues to be output at outputs 1 to 7 for as long as the **Save** input remains set to 1. Any change in status of inputs 1 to 7 during this time will not affect the outputs 1 to 7.

If the **Save** input is already set to 1 during the first cycle when the system transitions from the Stop to the Run status, this has the same effect as a switch from 0 to 1, i.e., the current status of inputs 1 to 7 is saved and continues to be output at outputs 1 to 7 for as long as the **Save** input remains set to 1.

If the **Save** input is not inverted, then the status of the **Save** output will always correspond to the status of the **Save** input.

If the **Save** input is inverted, then the status of the **Save** output will always correspond to the inverted status of the **Save** input.

Function block parameters

Table 39: Parameters of the Multiple memory function block

Parameter	Possible values
Number of inputs (not including Save input)	1 to 7
Invert input x	Every input of this function block can be inverted. At an inverted input, a 0 works in the same way as a 1 and vice versa.

Truth table for the Multiple memory function block

The following explanations apply to the truth table in this section:

- "↑" signifies that a rising signal edge (switch from 0 to 1) has been detected at the input.
- "n–1" refers to the previous value.
- "n" refers to the current value.

NOTE
Truth tables apply when the function blocks are configured without inverted inputs.

Table 40: Truth table for the Multiple memory function block

Save input	Save output	Output x_n
0	0	Input x
↑	↑	Input x
1	1	Output x_{n-1}

7.6.10 Clock generator

Function block diagram

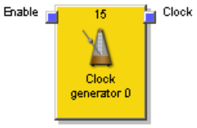


Figure 61: Inputs and outputs of the Clock generator function block

General description

The Clock generator function block allows you to generate a pulsed signal. When the **Enable clock** input is set to 1, the **Clock** output pulsates from 0 to 1 and back to 0 in accordance with the function block parameter settings. The **Clock** output switches to 0 when the **Enable clock** input is set to 0.

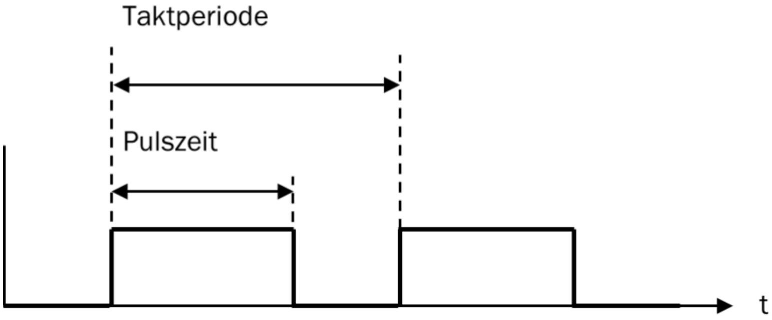


Figure 62: Parameter diagram for the Clock generator function block

Pulse time < elementary period (clock period)

The pulse time and elementary period are configured as a multiple of the logic execution time.

Function block parameters

WARNING

The clock period and pulse time change when the logic execution time is adjusted

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

- ► Check the whole application to ensure it functions without errors after any change to the configuration.

Table 41: Parameters of the Clock generator function block

Parameter	Possible values
Stop mode	• Immediate • After last clock pulse
Elementary period (clock period)	2 to 65,535 Duration = parameter value × logic execution time
Pulse time (pulse duration)	1 to 65,534 Duration = parameter value × logic execution time The pulse time must be shorter than the elementary period.

Sequence/timing diagram

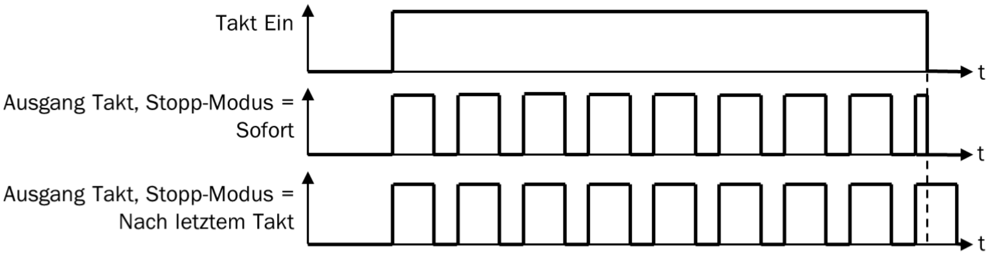

Takt Ein
t
Ausgang Takt, Stopp-Modus = Sofort
t
Ausgang Takt, Stopp-Modus = Nach letztem Takt
t

Figure 63: Sequence/timing diagram for the Clock generator function block

7.6.11 Event counter (up, down, and up and down)

Function block diagram

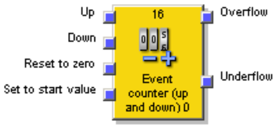


Figure 64: Inputs and outputs of the Event counter function block (up and down)

General description

You can use the Event counter function blocks to count events (in an upward and/or downward direction) so that the **Overflow** output indicates when a preset overflow value is reached and the **Underflow** output indicates when a value of zero is reached. The following function blocks are available for the various counting directions required: Event counter (up), Event counter (down) and Event counter (up and down).

Functionality

A rising signal edge (0–1) at the **Up** input increases the value of the internal counter by "1".

A rising signal edge (0–1) at the **Down** input reduces the value of the internal counter by "1".

If a rising signal edge (0–1) occurs at both the **Up** input and the **Down** input (only applies in the case of the up and down event counter function block), the value of the internal counter remains unchanged.

Function block parameters

Table 42: Parameters of the Event counter function blocks (up, down, and up and down)

Parameter	Possible values
Reset to zero after overflow	• Manual • Automatic
Set to start value after underflow	• Manual • Automatic
Overflow value	Integer of between 1 and 65,535. The overflow value must be greater than or equal to the start value.
Start value	Integer of between 1 and 65,535
Min. pulse time for reset to zero	• 100 ms • 350 ms
Min. pulse time for set to start value	• 100 ms • 350 ms

NOTE

If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal for **Reset to zero** or for **Set to start value** may produce a pulse if the signal is reset as a result of short-circuit detection.

WARNING

Undesired reset following short-circuit to high

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

- ► Make sure that the transitions of the signals for **Reset to zero** and **Set to start value** are in line with requirements according to the safety standards and regulations.
  - ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines)
  - ► No short-circuit detection, i.e., no referencing to test outputs

Reset to zero

A valid pulse sequence with a 0–1–0 transition at the **Reset to zero** input sets the internal counter to “0”. This happens whether or not the **Overflow value** has been reached and regardless of whether **Reset to zero after overflow** has been configured as Manual or Automatic.

The **Min. pulse time for reset to zero** determines the minimum duration of the pulse at the **Reset to zero** input. The valid values are 100 ms and 350 ms. If the pulse duration is shorter than the minimum pulse time configured or if it is longer than 30 s, the pulse is ignored.

Set to start value

A valid pulse sequence with a 0–1–0 transition at the **Set to start value** input sets the internal counter to the value that has been configured for the **Start value** parameter. This happens whether or not **Set to start value after underflow** has been configured as Manual or Automatic.

The **Min. pulse time for set to start value** determines the minimum duration of the pulse at the **Set to start value** input. The valid values are 100 ms and 350 ms. If the pulse duration is shorter than the minimum pulse time configured or if it is longer than 30 s, the pulse is ignored.

Overflow value and Reset to zero after overflow

The **Reset to zero after overflow** parameter determines what happens when the counter reaches the **Overflow value.** If this parameter is configured as Automatic and the internal counter is equal to the **Overflow value**, the **Overflow** output remains 1 for the duration of the logic execution time. After that, the value of the internal counter is reset to zero.

If the **Reset to zero after overflow** parameter is configured as Manual and the **Overflow value** has been reached, the **Overflow** output is set to 1 and remains 1 until the counter value changes again due to a countdown operation, a valid pulse sequence at the **Reset to zero** input, or a valid pulse sequence at the **Set to start value** input when the start value is lower than the overflow value. Until then, all other “up” counting pulses are ignored.

Start value and Set to start value after underflow

The **Set to start value after underflow** parameter determines what happens when the counter reaches a value of zero. If this parameter is configured as Automatic and the internal counter is equal to zero, the **Underflow** output remains 1 for the duration of the logic execution time. After that, the value of the internal counter is set to the configured **Start value.**

If the **Set to start value after underflow** is configured as Manual and the lower limit (i.e., zero) has been reached, the **Underflow** output is set to 1 and remains 1 until the counter value changes again due to a count-up operation or a valid pulse sequence at the **Set to start value** input. Until then, all other “down” counting pulses are ignored.

Truth table for the Event counter function blocks (up, down, and up and down)

The following explanations apply to the truth table in this section:

- “↑” signifies that a rising signal edge has been detected at the signal input.
- “↓” signifies that a falling signal edge has been detected at the signal input.
- “n–1” refers to the previous value.
- “n” refers to the current value.
- “y” refers to the value of the internal counter.
- “x” signifies “any”. For example, the **Reset to zero** and **Set to start value** inputs have priority over the **Up** and **Down** inputs.

Table 43: Truth table for the Event counter function blocks (up, down, and up and down)

Up	Down	Reset to zero	Set to start value	Counter value $_{n-1}$	Counter value $_n$	Overflow $_n$	Underflow $_n$
↑	0, 1, or ↓	0	0	y	y+1	0	0
↑	0, 1, or ↓	0	0	y	y+1 = overflow value	1	0
↑	0, 1, or ↓	0	0	y = overflow value	y = overflow value	1	0
0, 1, or ↓	↑	0	0	y	y–1	0	0
0, 1, or ↓	↑	0	0	y	y–1 = 0	0	1
0, 1, or ↓	↑	0	0	y = 0	y = 0	0	1
↑	↑	0	0	y	y	0	0
x	x	1	0	y	Reset to zero	0	0
x	x	0	1	y	Set to start value	0	0
x	x	1	1	y	Reset to zero	0	0

7.6.12 Fast shut off and Fast shut off with bypass

Function block diagram

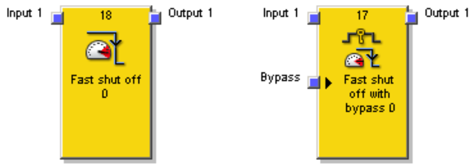


Figure 65: Inputs and outputs of the Fast shut off and Fast shut off with bypass function blocks

General description

The Fast shut off and Fast shut off with bypass function blocks are used to minimize the response time of a safety switching path within the Flexi Soft system. These function blocks can only be used if both the inputs and the outputs of the switching path are connected to the same expansion module (i.e., FX3-XTIO). This is necessary because both Fast shut off function blocks trigger direct shut-off on the expansion module, in turn resulting in a shorter shut-off time that is independent of the logic execution time.

As far as the Fast shut off function block is concerned, this means that the logic between the fast shut-off input and output cannot prevent a shut-off once the Fast shut off is activated.

By contrast, the Fast shut off with bypass function block makes it possible to bypass the Fast shut off function temporarily with the help of the Bypass input.

NOTE
The Fast shut off with bypass function block is only available with FX3-XTIO modules that feature a firmware version ≥ V2.00.0.

Example: In this logic example, the safety light curtain shuts off the motor.

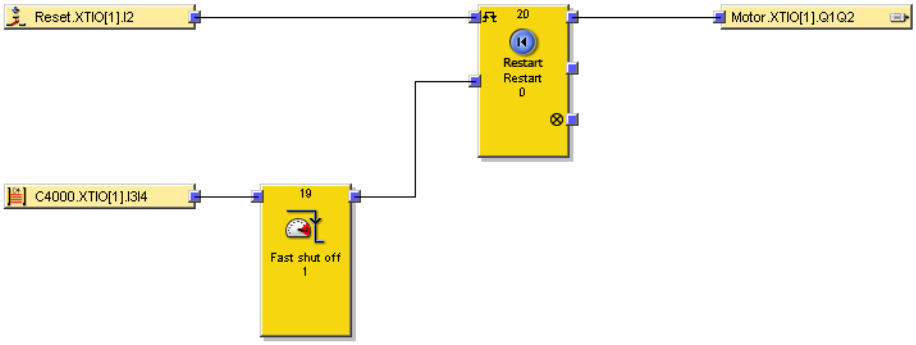


Figure 66: Example of Fast shut off

A simple form of logic such as this can be implemented within the Fast shut off function block (see configuration instructions below).

NOTE
The signal path running from the output of the Fast shut off function block to the physical output selected in the Fast shut off function block must be organized so that the physical output is also deactivated immediately along with the output of the Fast shut off function block. Typically, the AND, Restart, or External device monitoring function blocks can be used as part of the signal chain within this context. By contrast, an OR function block does not comply with the rules and so is not suitable.

Response time

The response time of the Fast shut off function block is not the same as the total response time of the complete safety function. The total response time involves several parameters that are not part of this function block.

WARNING
Incorrect calculation of the total response time

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

- ► Always observe the total response time of the complete safety function when configuring the Fast shut off.

NOTE
For a table on calculating the total response time of the Flexi Soft system, please refer to the operating instructions titled “Flexi Soft Modular Safety Controller Hardware”.

Function block parameters

Table 44: Parameters of the Fast shut off and Fast shut off with bypass function blocks

Parameter	Possible values
Number of inputs	Fast shut off: 1 to 8 Fast shut off with bypass: 1 to 7
Output for Fast shut off	Any output of the expansion module whose inputs are connected to the function block, unless the output concerned is already being used for the Fast shut off function

Configuring Fast shut off

The following example illustrates the function on the basis of three light curtains that are connected to one Fast shut off function block.

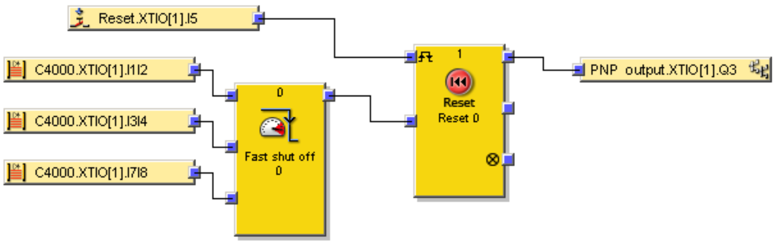


Figure 67: Example configuration of Fast shut off with three light curtains

- ► Connect the required input elements and output elements to the FX3-XTIO module.
- ► Connect the elements to the function block. In the configuration dialog of the function block, you can select the number of required inputs under **I/O settings**.
- ► Then, select the box for the inputs under **Parameters** to select the area.

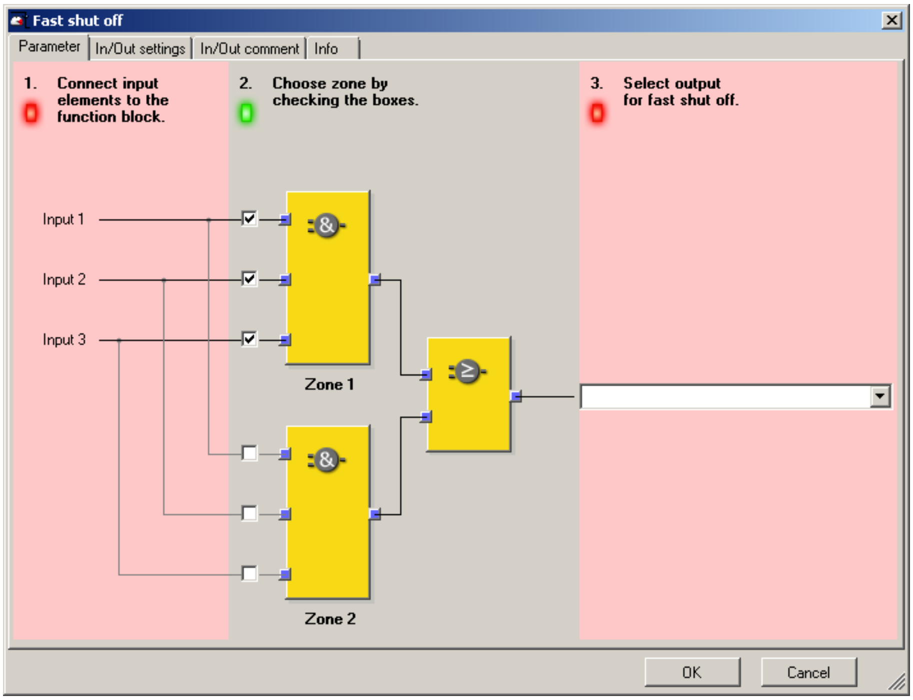

Fast shut off
Parameter
In/Out settings
In/Out comment
Info
1. Connect input elements to the function block.
2. Choose zone by checking the boxes.
3. Select output for fast shut off.
Input 1
Input 2
Input 3
Zone 1
Zone 2
OK
Cancel

Figure 68: Parameter settings for the Fast shut off function block

NOTE

If you only need to use AND logic, you can leave the AND function block inputs for Zone 2 deactivated. However, if you need to use OR logic as well, you can combine the inputs with the help of the Zone 1 and Zone 2 function blocks and then connect them to the internal OR function block.

► Finally, select the output for Fast shut off.

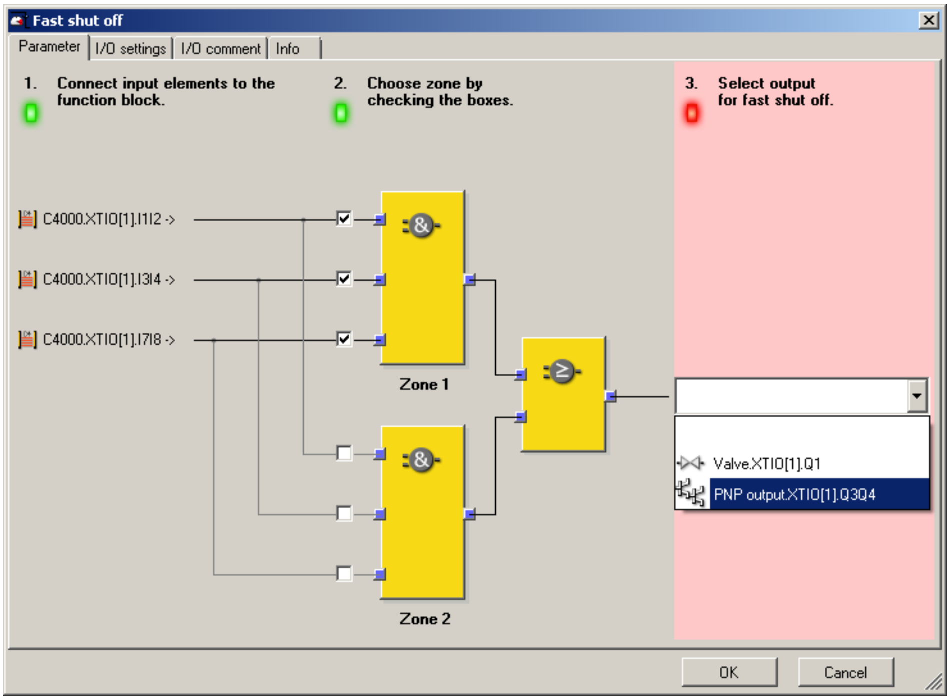


Figure 69: Select the output for Fast shut off

The selected inputs and outputs are now connected to one another in such a way that the outputs in the hardware configuration can no longer be moved to any other position and the inputs must remain connected to the same FX3-XTIO module. The elements that have been connected in this way are shown in orange in the hardware configuration.

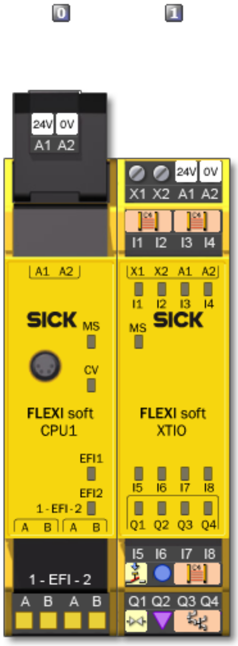


Figure 70: What the inputs and outputs connected to the Fast shut off function block look like in the hardware configuration

These connections will be undone if the Fast shut off function block is edited or deleted.

Fast shut off with bypass

In some applications, it may be necessary to bypass the Fast shut-off function, e.g., in safe setup mode when the machine can only be operated in inching mode. The Fast shut-off with bypass function block is available for this purpose. This is used and configured in exactly the same way as the Fast shut-off function block. The only difference is that one of the inputs of the Fast shut-off with bypass function block is used for the Bypass function.

If the **Bypass** input is 1, the Fast shut-off with bypass function block is bypassed. A stop condition such as breaching a protective field does **not** cause the machine to switch off while the bypass is active.

If the **Bypass** input is deactivated while a switch-off condition is present, the outputs will not be deactivated until the end of the standard application response time. The shorter response time for the Fast shut off function does not apply to the **Bypass** input.

WARNING

Restricted safety during the Bypass

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

- ▶ Make sure that the system or machine is in a safe status when using the Bypass function.
- ▶ Ensure the mandatory use of other protective measures while the Bypass function is active, e.g., that the machine is in safe setup mode so that it cannot endanger people or parts of the system while the bypass is active.
- ▶ Take into account the longer response time when the bypass is deactivated when planning the application.

NOTE

- Unlike the other inputs and outputs of this function block, the **Bypass** input can be connected to an output of another function block and also to any other input element that can be moved across to another module in the hardware configuration as well.
- The **Bypass** input is subject to a switch-on delay of three logic cycles to compensate for any delays associated with the logic processing time and the FLEXBUS+ transmission time. This delay makes sure that the I/O module has received the bypass signal before this is used for further logic processing in the Fast shut off with bypass function block. As a result of this delay, the **Bypass** input must remain set to 1 for three logic cycles before the Fast shut off function can be successfully bypassed. If this condition is met, the Fast shut off output of the function block remains set to 1, and the physical output on the I/O module remains set to High.
- The Fast shut off function block immediately deactivates the output of the FX3-XTIO module that is connected to it and the subsequent logic programming is ignored. For this reason, additional bypass conditions cannot be programmed in the logic editor between the output of the Fast shut off function block and the FX3-XTIO output that is connected to it.
- The value of the connected FX3-XTIO output in the online monitor may deviate from the actual value of the FX3-XTIO module's physical output. For example, the connected output may be 0 because of the downstream logic while the output of the Fast shut off with bypass function block is 1 and the FX3-XTIO module's physical output is High, because the **Bypass** input is 1.
- If one of the requirements of the application is the ability to deactivate the output of the FX3-XTIO module independently of any existing bypass condition (e.g., emergency stop), the underlying logic must be implemented in the manner illustrated below so that the relevant shutdown signal (e.g., emergency stop) also deactivates the **Bypass** input of the function block (see example).

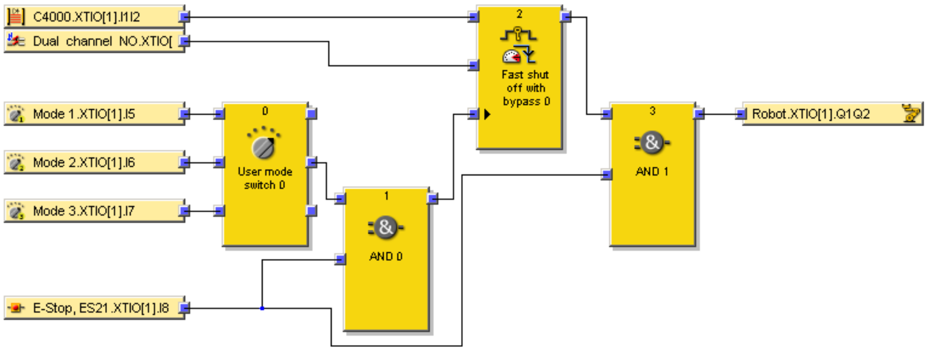


Figure 71: Example of Fast shut off with bypass with more than one bypass condition

7.6.13 Edge detection

Function block diagram

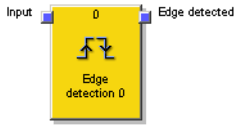


Figure 72: Inputs and outputs of the Edge detection function block

General description

The Edge detection function block makes it possible to detect a positive (rising) or negative (falling) input signal edge. The function block can be configured to detect a positive signal edge, a negative signal edge, or both. If a signal edge corresponding to the parameter settings is detected, the **Edge detected** output switches to 1 for the duration of the logic execution time.

Function block parameters

Table 45: Parameters of the Edge detection function block

Parameter	Possible values
Edge detection	• Positive • Negative • Positive and negative

Sequence/timing diagram

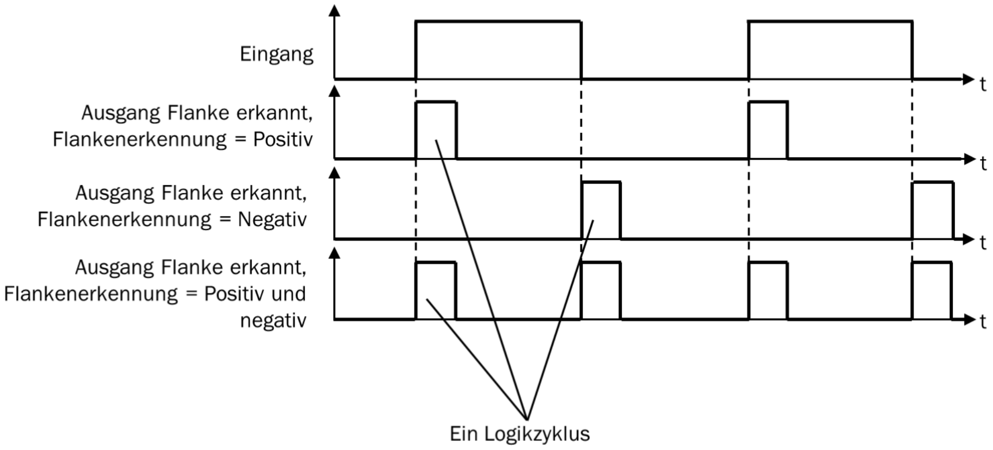


Figure 73: Sequence/timing diagram for the Edge detection function block

7.6.14 Binary encoder

Function block diagram

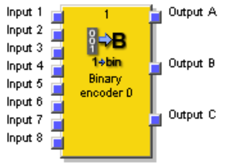


Figure 74: Inputs and outputs of the Binary encoder function block

General description

Depending on the current configuration, the Binary encoder function block converts a 1-of-n code or a priority code into a binary code (output A = 2^0output B = 2^1, output C = 2^2). Between two and eight inputs can be configured. The number of outputs is determined by the number of inputs. An optional **Fault present** output is also available.

Function block parameters

Table 46: Parameters of the Binary encoder function block

Parameter	Possible values
Number of inputs	2 to 8
Coding mode	• 1-of-n • Priority • Priority-to-binary (Input 1 dominant)
Use Fault present	• With • Without

Fault present output

WARNING
Undetected faults

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

► Remember to evaluate the **Fault present** output if the Binary encoder function block is being used for safety purposes.

If the Binary encoder function block is used for safety-related logic, then the evaluation of the **Fault present** is the only way to determine whether it is just a case of input 1 being set to 1 or whether the input status is invalid. In both instances, all outputs are set to 0.

1-of-n

In **1-of-n** mode, only one input may be set to 1 at any given time. The outputs are set on the basis of the index number of the input concerned (input 1 = 1, input 2 = 2, etc.). If all the inputs are 0 or if multiple inputs are 1 at the same time, all the outputs are set to 0 and the **Fault present** output switches to 1.

Priority

In Priority mode, multiple inputs can be set to 1 at the same time. The outputs are set on the basis of the input with the highest index (input 1 = 1, input 2 = 2, etc.). If all the inputs are 0 at the same time, all the outputs are set to 0 and the **Fault present** output switches to 1.

Priority-to-binary (Input 1 dominant)

In this mode, all the outputs are set to 0 when input 1 is 1. All the remaining inputs are ignored. When input 1 is 0, the function block behaves in the same way as for Priority mode. If all the inputs are 0 at the same time, all the outputs are set to 0 and the **Fault present** output switches to 1.

Truth tables for the Binary encoder function block

The following explanations apply to the truth tables in this section:

- “x” signifies “any” (0 or 1).

Table 47: Truth table for the Binary encoder function block with two inputs in 1-of-n mode

Input 2	Input 1	Output A	Fault present
0	0	0	1
0	1	0	0
1	0	1	0
1	1	0	1

Table 48: Truth table for the Binary encoder function block with eight inputs in 1-of-n mode

Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1	Output C	Output B	Output A	Fault present
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	1	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	1	0	1	0
0	1	0	0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	0	0	1	1	1	0
More than one input = 1								0	0	0	1

Table 49: Truth table for the Binary encoder function block with two inputs in Priority mode

Input 2	Input 1	Output A	Fault present
0	0	0	1
0	1	0	0
1	x	1	0

Table 50: Truth table for the Binary encoder function block with eight inputs in Priority mode

Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1	Output C	Output B	Output A	Fault present
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	x	0	0	1	0
0	0	0	0	0	1	x	x	0	1	0	0
0	0	0	0	1	x	x	x	0	1	1	0
0	0	0	1	x	x	x	x	1	0	0	0
0	0	1	x	x	x	x	x	1	0	1	0
0	1	x	x	x	x	x	x	1	1	0	0
1	x	x	x	x	x	x	x	1	1	1	0

Table 51: Truth table for the Binary encoder function block with two inputs in Priority-to-binary (Input 1 dominant) mode

Input 2	Input 1	Output A	Fault present
0	0	0	1
x	1	0	0
1	0	1	0

Table 52: Truth table for the Binary encoder function block with eight inputs in Priority-to-binary (Input 1 dominant) mode

Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1	Output C	Output B	Output A	Fault present
0	0	0	0	0	0	0	0	0	0	0	1
x	x	x	x	x	x	x	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	1	x	0	0	1	0	0
0	0	0	0	1	x	x	0	0	1	1	0
0	0	0	1	x	x	x	0	1	0	0	0
0	0	1	x	x	x	x	0	1	0	1	0
0	1	x	x	x	x	x	0	1	1	0	0
1	x	x	x	x	x	x	0	1	1	1	0

7.6.15 Binary decoder

Function block diagram

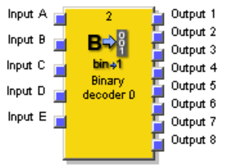


Figure 75: Inputs and outputs of the Binary decoder function block

General description

Depending on how it is currently configured, the Binary decoder function block decodes a binary code by converting it into either a 1-of-n code or a priority code. Up to five inputs can be configured. The number of outputs is determined by the number of inputs. When inputs A, B, and C are evaluated, a single Binary decoder function block can be used to decode binary codes with decimal values ranging from 0 to 7 (input A = 2^0, input B = 2^1, input C = 2^2). Optional inputs D and E allow you to combine up to four Binary decoder blocks for the purpose of decoding binary codes with decimal values ranging from 0 to 31.

Function block parameters

Table 53: Parameters of the Binary decoder function block

Parameter	Possible values
Coding mode	• 1-of-n • Priority
Inputs	• Not inverted • Inverted
Number of inputs	1 to 5
Value range	• 0 to 7 • 8 to 15 (only available if more than three inputs are used) • 16 to 23 (only available if five inputs are used) • 24 to 31 (only available if five inputs are used)

1-of-n

In **1-of-n** mode, the only output set to 1 is the one whose number matches the current input values.

Priority

In Priority mode, the output whose number matches the current input values is set to 1, along with all outputs with lower numbers.

Inputs inverted/not inverted

This parameter can be used to invert all the inputs.

Truth tables for the Binary decoder function block

Table 54: Truth table for the Binary decoder function block with one input in 1-of-n mode

Input A	Output 2	Output 1
0	0	1
1	1	0

Table 55: Truth table for the Binary decoder function block with two inputs in 1-of-n mode

Input B	Input A	Output 4	Output 3	Output 2	Output 1
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Table 56: Truth table for the Binary decoder function block with three inputs in 1-of-n mode

Inp. C	Inp. B	Inp. A	Outp. 8	Outp. 7	Outp. 6	Outp. 5	Outp. 4	Outp. 3	Outp. 2	Outp. 1
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Table 57: Truth table for the Binary decoder function block with one input in Priority mode

Input A	Output 2	Output 1
0	0	1
1	1	1

Table 58: Truth table for the Binary decoder function block with two inputs in Priority mode

Input B	Input A	Output 4	Output 3	Output 2	Output 1
0	0	0	0	0	1
0	1	0	0	1	1
1	0	0	1	1	1
1	1	1	1	1	1

Table 59: Truth table for the Binary decoder function block with three inputs in Priority mode

Inp. C	Inp. B	Inp. A	Outp. 8	Outp. 7	Outp. 6	Outp. 5	Outp. 4	Outp. 3	Outp. 2	Outp. 1
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	1
0	1	0	0	0	0	0	0	1	1	1
0	1	1	0	0	0	0	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1
1	0	1	0	0	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

Evaluating more than three inputs

If four or five inputs are used, up to four Binary decoder blocks can be combined to decode binary codes with values ranging from 0 to 31.

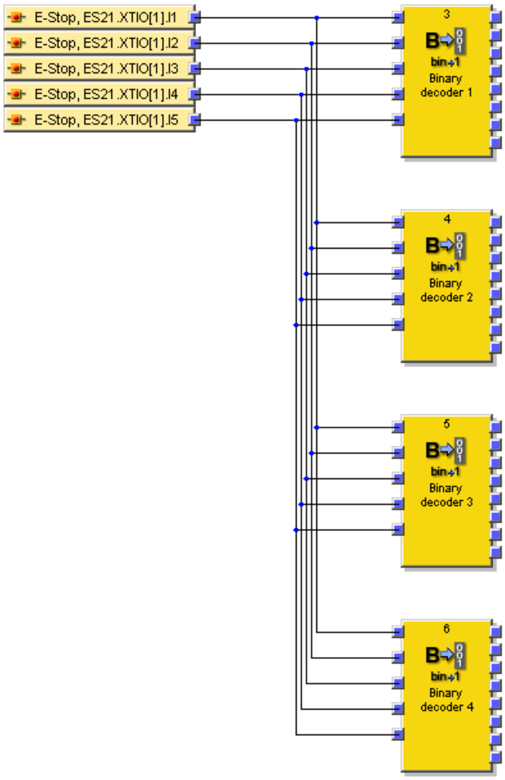


Figure 76: Four Binary decoder blocks combined

If you combine multiple Binary decoder function blocks, you must use the **Value range** option to configure the range of values that is to be covered by each one. This range is determined by the value of inputs D and E.

Table 60: Value range of Binary decoder function block based on input D

Input D	Value range
0	0 to 7
1	8 to 15

Table 61: Value range of Binary decoder function block based on inputs D and E

Input E	Input D	Value range
0	0	0 to 7
0	1	8 to 15
1	0	16 to 23
1	1	24 to 31

- If **input D** and **input E** have the same value as the **Value range** parameter (e.g., if **input E** = 1, **input D** = 0, and the **value range** is set to 16–23), the function block behaves as shown in the truth tables, depending on the value of inputs A, B, and C, and the configured coding mode (**1-of-n** or **Priority**).
- If **input D** and **input E** have a lower value than the **Value range** parameter (e.g., if **input E** = 0, **input D** = 1, and the **value range** = 16–23), all the outputs are set to 0 regardless of which coding mode is configured (**1-of-n** or **Priority**).
- If **Input D** and **Input E** have a higher value than the **Value range** parameter (e.g., **Input E** = 1, **Input D** = 1, and the **Value range** = 16–23), then...
  - all the outputs are set to 0 in **1-of-n** mode,
  - all the outputs are set to 1 in **Priority** mode.

7.6.16 Message generator

Function block diagram

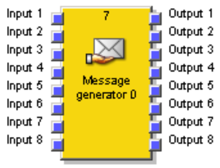


Figure 77: Inputs and outputs of the Message generator function block

General description

The Message generator function block evaluates up to eight inputs. If a signal edge is detected at one of these inputs in accordance with the configuration, the function block sets the associated output to 1 for the duration of the logic execution time and adds a customized text message to the diagnostic history. This can be read out in online mode using the Diagnostics function in the configuration software.

NOTE
These messages get deleted if the voltage supply of the Flexi Soft system is interrupted.

Function block parameters

Table 62: Parameters of the Message generator function block

Parameter	Possible values
Number of inputs	1 to 8
Messages	Up to 64 user-defined messages per project
Input condition	• Rising signal edge • Falling signal edge • Rising or falling signal edge

The following example shows the Message generator function block with three emergency stop pushbuttons connected to it.

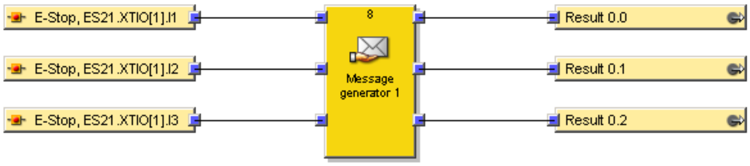
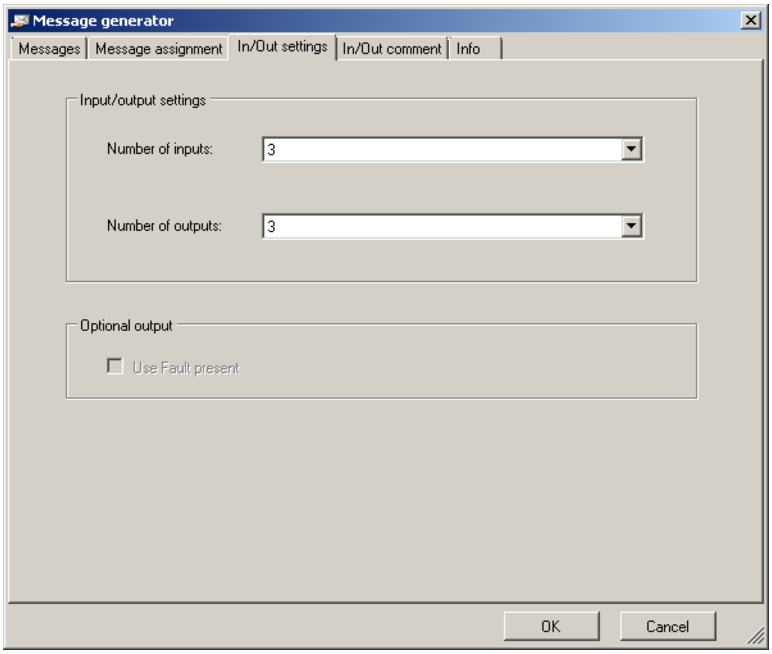


Figure 78: Example configuration for Message generator with three emergency stop pushbuttons

Configuring the Message generator function block

► Connect the input elements to the function block. In the configuration dialog of the function block, you can select the required number of inputs under **I/O settings**.



Input/output settings

Number of inputs: 3

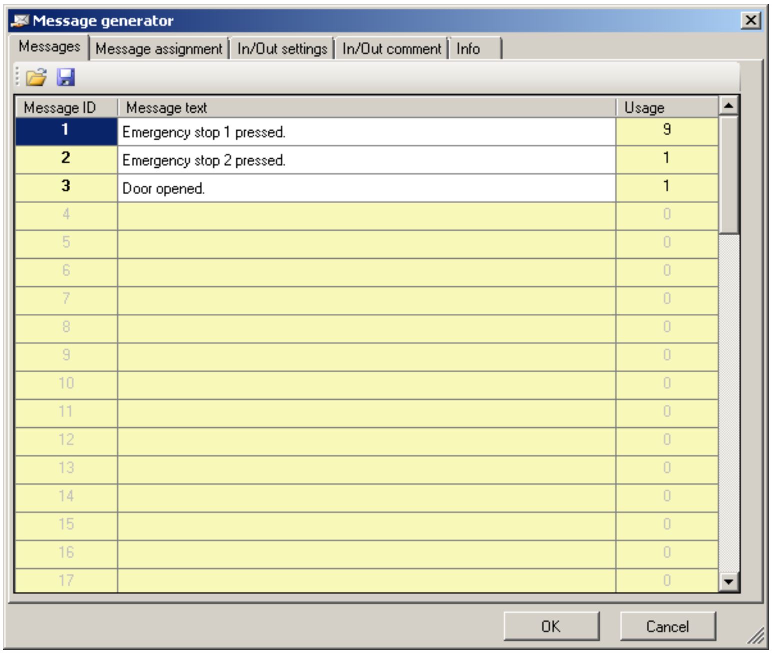
Number of outputs: 3

Optional output

□ Use Fault present

Figure 79: In/Out settings for the Message generator function block

► Then click on the **Messages** tab and enter the messages that you want to be output in the Diagnostics view.



Message ID	Message text	Usage
1	Emergency stop 1 pressed.	9
2	Emergency stop 2 pressed.	1
3	Door opened.	1
4		0
5		0
6		0
7		0
8		0
9		0
10		0
11		0
12		0
13		0
14		0
15		0
16		0
17		0

Figure 80: Messages of the Message generator function block

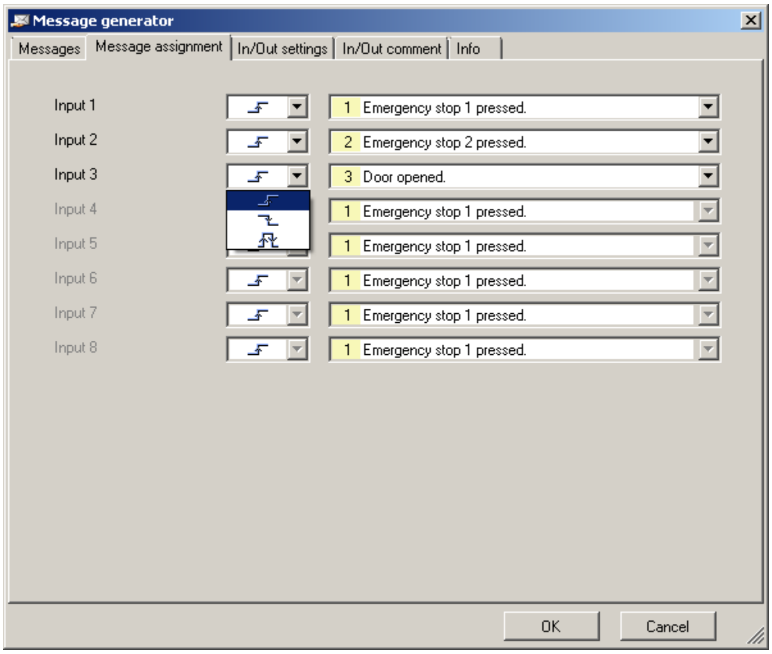
NOTE

- You can enter up to 64 different messages per project, each with a maximum length of 32,767 characters.
- The messages entered apply across all the Message generator function blocks that are used in one project.
- The messages are saved in the project and remain there even if you delete the Message generator function block from the workspace.
- You can use the **Import from CSV** and **Export to CSV** buttons to save the messages as a text file in CSV format or to import them from a CSV file.

► Finally, on the **Message assignment** tab, assign the desired message to each of the inputs used. For each input, select the input condition that will cause the relevant message to be output when it is met (rising signal edge, falling signal edge, or rising or falling signal edge).

NOTE

The message assignment cannot be exported or imported.



Input 1	1 Emergency stop 1 pressed.
Input 2	2 Emergency stop 2 pressed.
Input 3	3 Door opened.
Input 4	1 Emergency stop 1 pressed.
Input 5	1 Emergency stop 1 pressed.
Input 6	1 Emergency stop 1 pressed.
Input 7	1 Emergency stop 1 pressed.
Input 8	1 Emergency stop 1 pressed.

Figure 81: Message assignment for the Message generator function block

Priority of messages

If more than one condition is met at the same time, the following priorities apply:

- If there is only one Message generator function block, the input with the lowest number has priority, i.e., the message generated by this input is logged first.
- If multiple Message generator function blocks are used, the function block with the lowest function block index number has priority, i.e., the messages generated by this function block are logged first.

7.6.17 Routing 1:n

Function block diagram

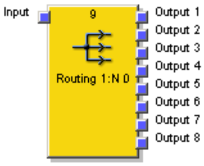


Figure 82: Inputs and outputs of the Routing 1:n function block

General description

The Routing 1:n function block routes one input signal to a maximum of eight output signals. This function block can be used to connect one output of a function block or one input element to multiple output elements at the same time (e.g. outputs of an I/O module, CPU markers). However, it is not required if you want to create a connection to multiple function block inputs, as this can be done directly.

Function block parameters

Table 63: Parameters of the Routing 1:n function block

Parameter	Possible values
Number of outputs	1 to 8

7.6.18 Routing n:n

Function block diagram

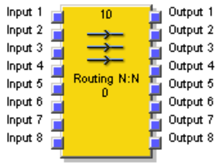


Figure 83: Inputs and outputs of the Routing n:n function block

General description

The Routing n:n function block routes up to eight input signals to a maximum of eight outputs in parallel. This function block makes it possible to connect input elements (e.g., inputs of an FX3-XTIO or FX3-XTDI module) to output elements on a one-to-one basis.

Function block parameters

Table 64: Parameters of the Routing n:n function block

Parameter	Possible values
Number of inputs and outputs	1 to 8
Invert input x	Every input of this function block can be inverted. At an inverted input, a 0 works in the same way as a 1 and vice versa.

7.7 Application-specific function blocks

7.7.1 Reset

Function block diagram

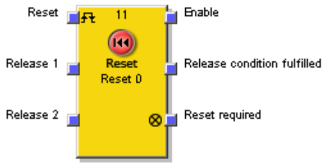


Figure 84: Inputs and outputs of the Reset function block

General description

The Reset function block can be used to meet the safety application requirements laid down by standards for acknowledging a manual safety stop with a subsequent request to restart the application. Typically, the safety logic for a Flexi Soft modular safety controller will always include a Reset function block.

NOTE

If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal for the Reset function may produce a pulse if the signal is reset as a result of short-circuit detection.

WARNING
Undesired reset following short-circuit to high

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

- ► Make sure that the transitions of the signals for the **Reset** function meet the requirements of safety standards and regulations.
  - ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines)
  - ► No short-circuit detection, i.e., no referencing to test outputs

Function block parameters

Table 65: Parameters of the Reset function block

Parameter	Possible values
Min. reset pulse time	• 100 ms • 350 ms
Number of inputs	2 to 8 (= 1 to 7 Release inputs activated)

Release condition fulfilled output

The **Release condition fulfilled** output indicates the result of an AND operation involving all activated **Release** inputs. It is set to 1 when all the activated **Release** inputs are 1.

Reset required output

The **Reset required** output pulsates at 1 Hz to indicate that the function block is expecting a valid reset pulse at the **Reset** input so that the **Enable** output can switch to 1. This happens when the **Release condition fulfilled** output is 1, i.e., when all activated **Release** inputs are 1 but the **Enable** output is still set to 0. This output is usually used to control an indicator lamp.

Enable output

The **Enable** output switches to 1 when the **Release condition fulfilled** output is 1 and a valid reset pulse has been detected at the **Reset** input, provided that all activated **Release** inputs remain set to 1.

The **Min. reset pulse time** determines the minimum duration of the pulse at the **Reset** input. The valid values are 100 ms and 350 ms. If the pulse duration is shorter than the minimum pulse time configured or if it is longer than 30 s, the pulse is ignored.

The **Enable** output switches to 0 when one or more of the **Release** inputs change to 0.

Sequence/timing diagram

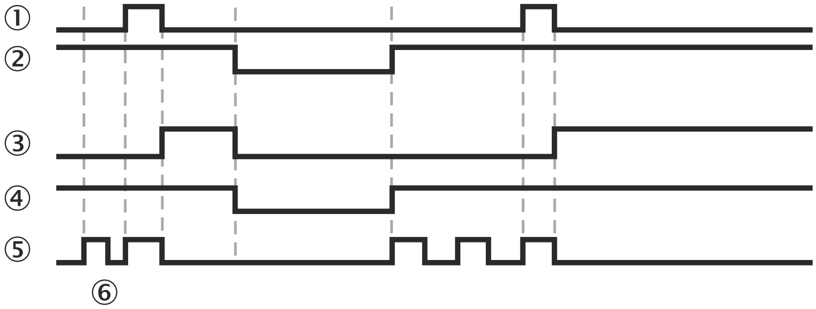


Figure 85: Sequence/timing diagram for the Reset function block

①	Reset
②	Release 1 input
③	Enable output
④	Release condition fulfilled
⑤	Reset required
⑥	Stop ➜ Run

7.7.2 Restart

Function block diagram

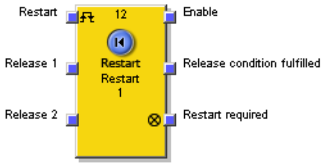


Figure 86: Inputs and outputs of the Restart function block

General description

The internal logic of the Restart function block works in exactly the same way as that of the Reset function block. The Restart function block makes it possible to distinguish between the function blocks graphically while still adhering to the application standards for acknowledging a manual restart request.

NOTE
If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal for restarting may produce a pulse if the signal is reset as a result of short-circuit detection.

WARNING
Undesired restart following short-circuit to high

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

► Make sure that the transitions of the signals for the Restart function meet the requirements of safety standards and regulations.
  ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines)
  ► No short-circuit detection, i.e., no referencing to test outputs

Function block parameters

Table 66: Parameters of the Restart function block

Parameter	Possible values
Min. restart pulse time	• 100 ms • 350 ms
Number of inputs	2 to 8 (= 1 to 7 Release inputs activated)

Release condition fulfilled output

The **Release condition fulfilled** output indicates the result of an AND operation involving all activated **Release** inputs. It is set to 1 when all the activated **Release** inputs are 1.

Restart required output

The **Restart required** output pulsates at 1 Hz to indicate that the function block is expecting a valid restart pulse at the **Restart** input so that the **Enable** output can switch to 1. This happens when the **Release condition fulfilled** output is 1, i.e., when all activated **Release** inputs are 1 but the **Enable** output is still set to 0. This output is typically used to control an indicator lamp.

Enable output

The **Enable** output switches to 1 when the **Release condition fulfilled** output is 1 and a valid restart pulse has been detected at the **Restart** input, provided that all activated **Release** inputs remain set to 1.

The **Min. restart pulse time** determines the minimum duration of the pulse at the **Restart** input. The valid values are 100 ms and 350 ms. If the pulse duration is shorter than the minimum pulse time configured or if it is longer than 30 s, the pulse is ignored.

The **Enable** output switches to 0 when one or more of the **Release** inputs change to 0.

Sequence/timing diagram

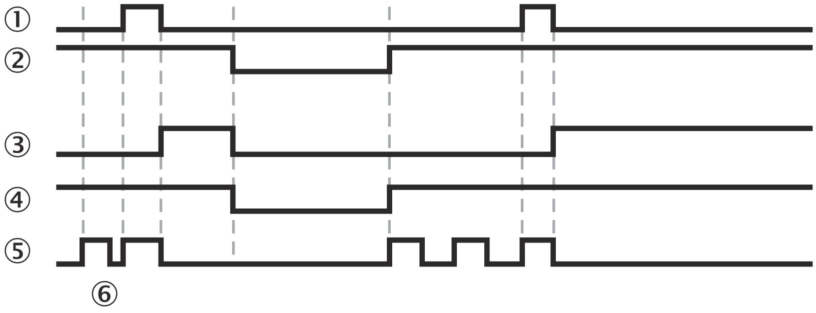


Figure 87: Sequence/timing diagram for the Restart function block

① Restart

② Release 1 input

③ Enable output

④ Release condition fulfilled

⑤ Restart required

⑥ Stop ➜ Run

7.7.3 Off-delay timer

Function block diagram

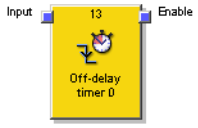


Figure 88: Inputs and outputs of the Switch-off delay timer function block

General description

The Off-delay timer function block delays deactivation of the **Enable** output by a configurable period of time.

Function block parameters

Table 67: Parameters of the Off-delay timer function block

Parameter	Possible values
Delay time	0 to 300 s in 10 ms increments. If the value is anything other than 0, it must be greater than the logic execution time.

The timer starts the delay sequence when the input transitions from 1 to 0. When the timer reaches the end of the configured **delay time**, the **Release** output also switches to 0, provided that the input is still set to 0. If the input changes to 1, the **Release** output immediately switches to 1 and the timer is reset.

Sequence/timing diagram

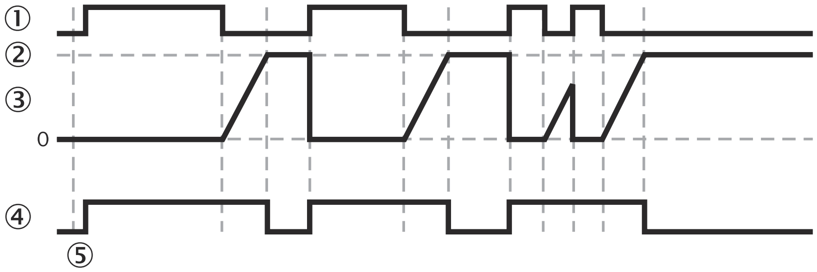


Figure 89: Sequence/timing diagram for the Off-delay timer function block

① Input
② Set value
③ Timer value
④ Enable output
⑤ Stop ➜ Run

7.7.4 Adjustable off-delay timer

Function block diagram

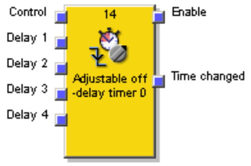


Figure 90: Inputs and outputs of the Adjustable switch-off delay timer function block

General description

The Adjustable off-delay timer function block delays deactivation of the **Enable** output by an adjustable period of time. It is possible to configure four custom off-delay times, each of which can be activated by means of an associated **Delay** input. The total delay time is equal to the sum of all the activated delay times.

Function block parameters

Table 68: Parameters of the Adjustable off-delay timer function block

Parameter	Possible values
Off delay time 1	1 to 60 s in 10 ms increments. If the value is anything other than 0, the associated input is activated. In this case, the value must be greater than the logic execution time. The total delay time (sum of all off-delay times) is limited to 600 seconds.
Off delay time 2	
Off delay time 3	
Off delay time 4	

The timer starts the delay sequence when a falling signal edge (1–0) occurs at the **Control** input. When the timer reaches the end of the selected total delay time, the **Enable** output also switches to 0, provided that the **Control** input is still set to 0. If the **Control** input changes to 1, the **Enable** output immediately switches to 1 and the timer is reset.

If any of the **Delay** inputs assume a different value while a delay sequence is running, the **Time changed** output switches to 1 and remains set to 1 until the **Control** input switches back to 1.

The effective total delay time is dependent on which **Delay** inputs were set to 1 when the falling signal edge occurred at the **Control** input. This means that if a change does occur at the **Delay** inputs during a delay sequence, it does not have any effect on the delay sequence that is currently running.

If the **Control** input is 0 during the first logic cycle after a transition from the Stop status to the Run status, the **Enable** output also remains set to 0.

Sequence/timing diagram

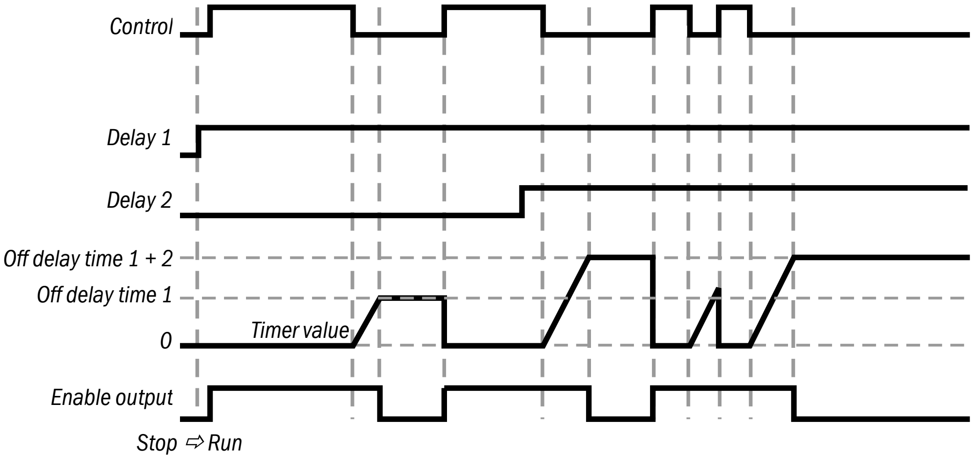


Figure 91: Sequence/timing diagram for the Adjustable off-delay timer function block with Off delay time 1 and Off delay time 2

7.7.5 On-delay timer

Function block diagram

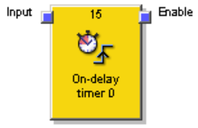


Figure 92: Inputs and outputs of the On-delay timer function block

General description

The On-delay timer function block delays activation of the **Enable** output by a configurable period of time.

Function block parameters

Table 69: Parameters of the On-delay timer function block

Parameter	Possible values
Delay time	0 to 300 s in 10 ms increments. If the value is anything other than 0, it must be greater than the logic execution time.

The timer starts the delay sequence when the input transitions from 0 to 1. When the timer reaches the end of the configured **delay time**, the **Release** output also switches to 1, provided that the input is still set to 1. If the input changes to 0, the **Release** output immediately switches to 0 and the timer is reset.

Sequence/timing diagram

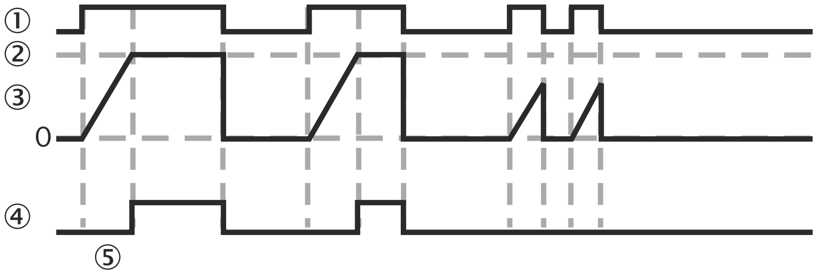


Figure 93: Sequence/timing diagram for the On-delay timer function block

① Input

② Set value

③ Timer value

④ Enable output

⑤ Stop ➜ Run

7.7.6 Adjustable on-delay timer

Function block diagram

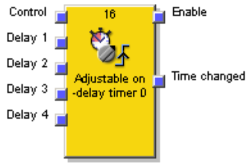


Figure 94: Inputs and outputs of the Adjustable on-delay timer function block

General description

The Adjustable on-delay timer function block delays activation of the **Enable** output by an adjustable period of time. It is possible to configure four custom delay times, each of which can be activated by means of an associated **Delay** input. The total delay time is equal to the sum of all the activated delay times.

Function block parameters

Table 70: Parameters of the Adjustable on-delay timer function block

Parameter	Possible values
On delay time 1	0 to 600 s in 10 ms increments. If the value is anything other than 0, the associated input is activated. In this case, the value must be greater than the logic execution time. The total delay time (sum of all on-delay times) is limited to 600 seconds.
On delay time 2	
On delay time 3	
On delay time 4	

The timer starts the delay sequence when a rising signal edge (0–1) occurs at the **Control** input. When the timer reaches the end of the selected total delay time, the **Enable** output also switches to 1, provided that the **Control** input is still set to 1. If the **Control** input changes to 0, the **Enable** output immediately switches to 0 and the timer is reset.

If any of the **Delay** inputs assumes a different value while a delay sequence is running, the **Time changed** output switches to 1 and remains set to 1 until the **Control** input switches back to 0.

The effective total delay time is dependent on which **Delay** inputs were set to 1 when the rising signal edge occurred at the **Control** input. This means that if a change does occur at the **Delay** inputs during a delay sequence, it does not have any effect on the delay sequence that is currently running.

If the **Control** input is 1 during the first logic cycle after a transition from the Stop status to the Run status, the **Enable** output switches to 1 after expiry of the selected total delay time.

Sequence/timing diagram

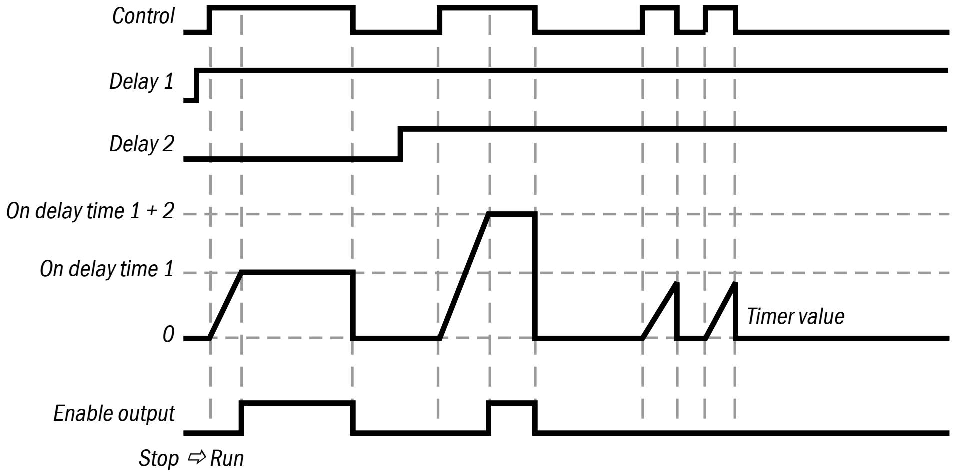


Figure 95: Sequence/timing diagram for the Adjustable on-delay timer function block with On delay time 1 and On delay time 2

7.7.7 External device monitoring

Function block diagram

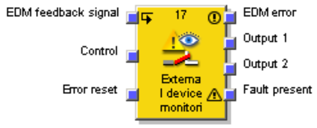


Figure 96: Inputs and outputs of the External device monitoring function block

General description

The External device monitoring function block can be used to control an external device (e.g., a contactor) and check – on the basis of its feedback signal – whether it has switched as expected. This involves connecting the external device to **Output 1** and/or **Output 2**. The feedback signal is connected to the **EDM feedback signal** input. The **Control** input is connected to the logic signal that represents the desired status for the external device, e.g., the **Enable** output of a Reset function block.

Function block parameters

Table 71: Parameters of the External device monitoring function block

Parameter	Possible values
Max. feedback delay	• With firmware < 4.00.0: 10 to 1,000 ms in 10 ms increments. • With firmware ≥ 4.00.0: 10 to 60,000 ms in 10 ms increments. The value must be greater than the logic execution time.
Using the Error reset input	• With • Without
Use Fault present	• With • Without

Output 1 and Output 2

Both outputs always have the same value. This means that two outputs are available for direct connection to two output elements.

Output 1 and **Output 2** change to 1 if the **EDM feedback signal** is 1 and the **Control** input then switches from 0 to 1.

Output 1 and **Output 2** switch to 0 if the **Control** input is 0 or if an error occurs (**EDM error** output set to 1).

NOTE

If you want the signals of **Output 1** and **Output 2** to be delayed, you need to implement this delay using another function block that is located **before** – and not after – of the External device monitoring function block. Otherwise, errors may result.

EDM error and Fault present

The general expectation is that the **EDM feedback signal** will always assume the inverted value of the **Control** input within the maximum feedback delay time (T_{EDM}) that has been configured.

The **EDM error** and **Fault present** outputs change to 1 in the following situations:

- The **Control** input switches from 0 to 1 and the **EDM feedback signal** is set to 0 (regardless of T_{EDM}).
- The **Control** input switches from 0 to 1 and the **EDM feedback signal** does not switch from 1 to 0 within T_{EDM}.
- The **Control** input switches from 1 to 0 and the **EDM feedback signal** does not switch from 0 to 1 within T_{EDM}.
- The **Control** input is set to 0 and the **EDM feedback signal** switches to 0 for longer than T_{EDM}.
- The **Control** input is set to 1 and the **EDM feedback signal** switches to 1 for longer than T_{EDM}.

The **EDM error** and **Fault present** outputs switch to 0 on detection of a signal sequence that sets **Output 1** and **Output 2** to 1.

Error reset

An error can also be reset by means of the **Error reset** input (with firmware ≥ V3.00.0). The **EDM error** and **Fault present** outputs change to 0 if the **Error reset** input switches from 0 to 1 and either of the following conditions is met:

- The **Control** input is set to 0 and the **EDM feedback signal** is set to 1.
- The **Control input** is set to 1 and the **EDM feedback signal** is set to 0.

Output 1 and **Output 2** only switch to 1 as well if the second of these two possible conditions is met. This can lead an undesired switch-on if there is a defective contactor (**EDM feedback signal** is set to 0 instead of 1 as the contactor has not dropped out properly).

WARNING

Undesired machine switch-on

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

► Only activate the **Error reset** input if the **Control** input is set to 0.

Sequence/timing diagram

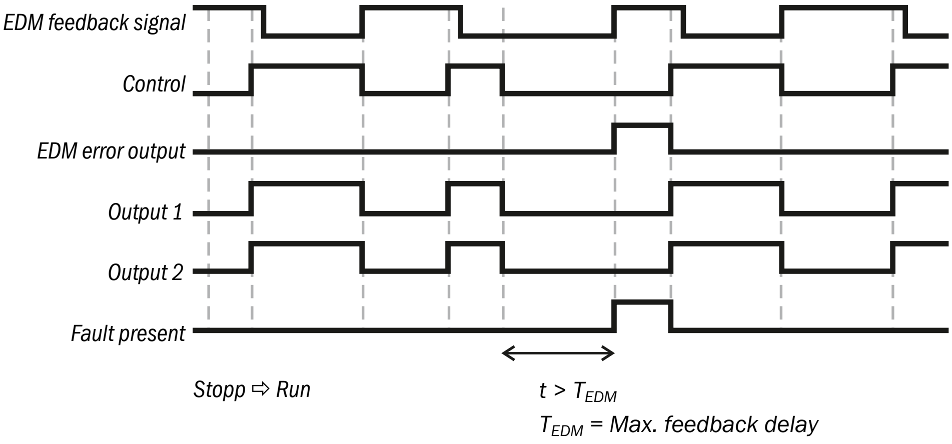


Figure 97: Sequence/timing diagram for the External device monitoring function block

7.7.8 Valve monitoring

Function block diagram

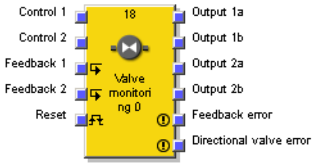


Figure 98: Inputs and outputs of the Valve monitoring function block, configured for a directional valve

General description

The Valve monitoring function block can be used to control valves and check – on the basis of their feedback signals – whether they have switched as expected.

This involves connecting the valves to **Output 1A** through to **Output 2B**. The feedback signals are connected to the **Feedback 1** and **Feedback 2** inputs. The **Control input 1** and **Control input 2** inputs are connected to the logic signal that represents the desired status for the valve, e.g. the **Release** output of a Reset function block. Depending on the type of valve, some of the signals may not be required.

Three different types of valve are available: single valves, double valves, and directional valves.

Function block parameters

Table 72: Parameters of the Valve monitoring function block

Parameter	Possible values
Reset condition	• Manual reset • Automatic reset
Continuous monitoring when valve is active	• Active • Deactivated

Parameter	Possible values
Valve type	• Single valve (Control input 1, Output 1A, Output 1B, Feedback 1 activated) • Double valve (Control input 1, Output 1A, Output 1B, Feedback 1, Output 2A, Output 2B, Feedback 2 activated) • Directional valve (Control input 1, Output 1A, Output 1B, Feedback 1, Control input 2, Output 2A, Output 2B, Feedback 2, Directional valve error activated)
Max. switch-on feedback delay	• With firmware < 4.00.0: 50 to 1,000 ms in 10 ms increments. • With firmware ≥ 4.00.0: 50 to 60,000 ms in 10 ms increments. 0 = infinite (only with FX3-CPUx firmware version ≥ V2.00.0) If this parameter is set to 0, the **Continuous monitoring when valve is active** option must be deactivated. If the value is anything other than 0, it must be greater than the logic execution time.
Max. switch-off feedback delay	• With firmware < 4.00.0: 50 to 1,000 ms in 10 ms increments. • With firmware ≥ 4.00.0: 50 to 60,000 ms in 10 ms increments. 0 = infinite (only with FX3-CPUx firmware version ≥ V2.00.0) If the value is anything other than 0, it must be greater than the logic execution time.
Min. reset pulse time	• 100 ms • 350 ms
Use Fault present	• With • Without

WARNING

Incorrect function due to short-circuit of the feedback signals

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

► Prevent a short-circuit of the feedback signals, e.g., through protected cable laying or wiring of this signal only within the control cabinet.
  - Short-circuit of the feedback signals **Feedback 1** and **Feedback 2** with each other
  - Short-circuit of the feedback signals to the signals for outputs

Output 1A through Output 2B

The two outputs that make up each pair (**Output 1A** and **Output 1B**/**Output 2A** and **Output 2B**) always have the same value. This means that two outputs are available for each valve to allow direct connection to two outputs elements.

Output 1A / **1B** and **Output 2A** / **2B** change to 1 if the associated input (**Feedback 1** or **Feedback 2**) is set to 1 and the associated **Control** input then switches from 0 to 1.

Output 1A / **1B** and **Output 2A** / **2B** change to 0 if the associated **Control** input is set to 0 or if an error is present (**Feedback error** output is set to 1 or **Directional valve error** is set to 1).

The control input associated with **Output 1A** / **1B** is always **Control input 1**.

The control input associated with **Output 2A** / **2B** depends on which valve type has been configured:

- For double valve: **Control 1**
- For directional valve: **Control 2**

Feedback error, Directional valve error, and Fault present

The general expectation is that the **Feedback 1/2** input will always assume the inverted value of the associated **Control** input within the **Max. switch-on feedback delay** (T_{ON}) or **Max. switch-off feedback delay** (T_{OFF}) that has been configured.

The **Feedback error** output is 1 if one of the following conditions is met:

- The **Control input** switches from 0 to 1 and the associated **Feedback signal** is set to 0 (regardless of T_{ON} and T_{OFF}).
- T_{ON} is greater than zero, the **Control input** switches from 0 to 1, and the associated **Feedback signal** fails to switch from 1 to 0 within T_{ON}.
- T_{OFF} is greater than zero, the **Control input** switches from 1 to 0, and the associated **Feedback signal** fails to switch from 0 to 1 within T_{OFF}.
- **Continuous monitoring when valve is active** is active, the **Control input** is set to 1, and the associated **Feedback signal** switches to 1.

The **Directional valve error** output switches to 1 if the **Valve type** parameter = Directional valve and the **Control input 1** and **Control input 2** inputs are both set to 1 at the same time.

The **Fault present** output switches to 1 if the **Feedback error** and/or **Directional valve error** is 1.

The **Feedback error**, **Directional valve error**, and **Fault present** outputs switch to 0 when all the activated Control inputs are 0 and all the activated Feedback inputs are 1. If manual reset has been configured as a reset condition, a valid reset pulse must also be triggered at the **Reset** input.

The **Min. reset pulse time** determines the minimum duration of the pulse at the **Reset** input. The valid values are 100 ms and 350 ms. If the pulse duration is shorter than the minimum pulse time configured or if it is longer than 30 s, the pulse is ignored.

NOTE
If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal for Reset may produce a pulse if the signal is reset as a result of short-circuit detection.

WARNING
Undesired reset following short-circuit to high

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

- ► Make sure that the transitions of the signals for **Reset** meet the requirements of safety standards and regulations.
  - ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines)
  - ► No short-circuit detection, i.e., no referencing to test outputs

Sequence/timing diagrams

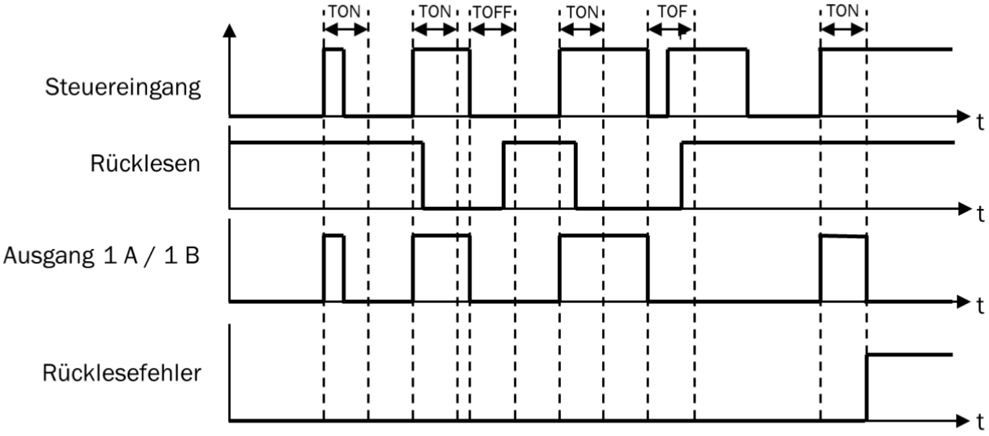


Figure 99: Sequence/timing diagram for single valve in manual reset mode

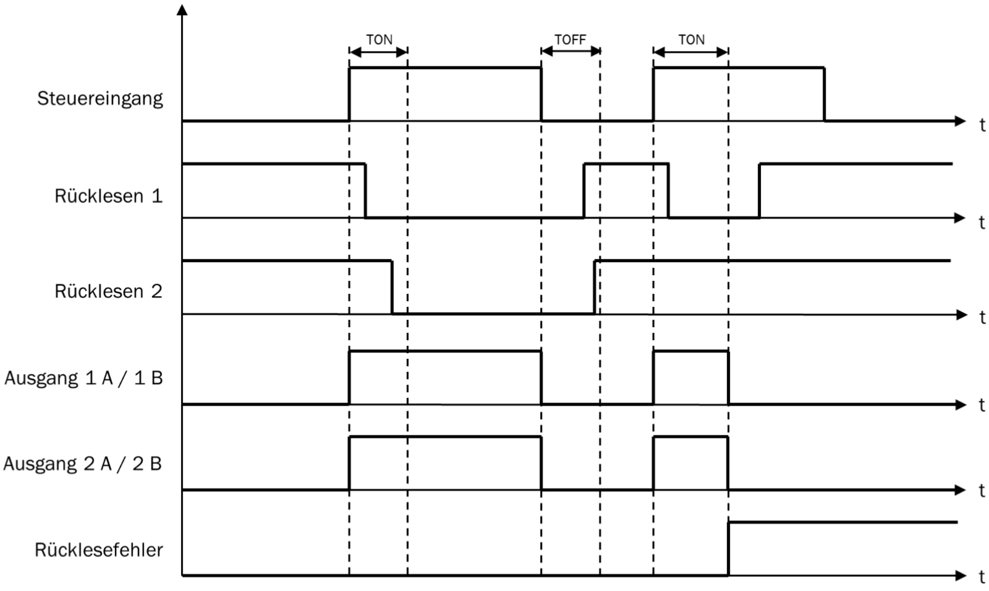


Figure 100: Sequence/timing diagram for double valve in manual reset mode

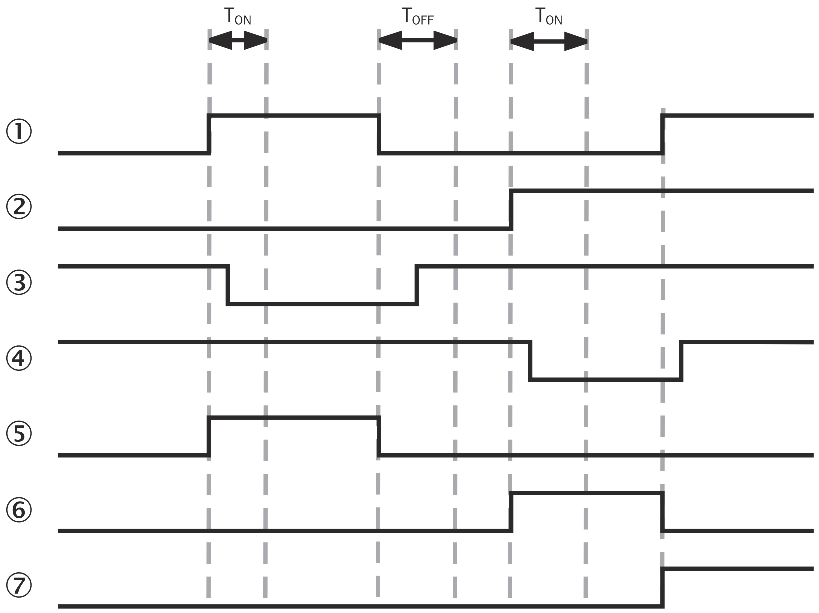


Figure 101: Sequence/timing diagram for directional valve

①	Control 1
②	Control 2
③	Feedback 1
④	Feedback 2
⑤	Output 1A / 1B
⑥	Output 2A / 2B
⑦	Directional valve error

7.7.9 User mode switch

Function block diagram

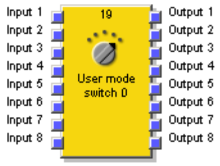


Figure 102: Inputs and outputs of the Operating mode selector switch function block

General description

The User mode switch function block selects an output on the basis of an input value. If input x is 1, then output x is 1.

The function block supports between two and eight inputs plus the corresponding outputs.

Only one input can ever be set to 1 at once. If more than one input or if no input is set to 1, the last output to be set to 1 remains 1 for the duration of the set discrepancy time. At the end of the discrepancy time, the outputs are set to the values defined in the error output combination and the **Fault present** output switches to 1.

If there is no valid input combination during the first logic cycle after a transition from the Stop status to the Run status, the outputs are immediately set to the values defined in the error output combination and the **Fault present** output switches to 1.

Function block parameters

Table 73: Parameters of the User mode switch function block

Parameter	Possible values
Discrepancy time	0 to 10 seconds in 10 ms increments
Error output combination	When Fault present is 1, any outputs with a check mark switch to 1 and any without a check mark switch to 0.
Number of inputs or Number of outputs	2 to 8
Use Fault present	• With • Without

Testing

WARNING

Restricted cross-circuit detection when using tested inputs

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

► Prevent an input cross-circuit, e.g., by using protected cable laying.

NOTE

- If the function block is connected to tested inputs and a test pulse error (short-circuit to High) leads to a faulty input combination (0 input value), the test pulse error must first be reset. One way to do this is to briefly disconnect the affected line at the input or at the test output.
- If the function block is connected to tested inputs, a cross-circuit between the inputs used can only be detected if the selected operating mode causes one of these inputs to be activated.

Truth table for the User mode switch function block

Table 74: Truth table for the User mode switch function block

Inputs								Outputs								
1	2	3	4	5	6	7	8	Fault present	1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
More than one input set to 1 or no input set to 1 for less than the configured discrepancy time								0	= last output combination							
More than one input set to 1 or no input set to 1 for longer than the configured discrepancy time								1	= error output combination							

Sequence/timing diagram

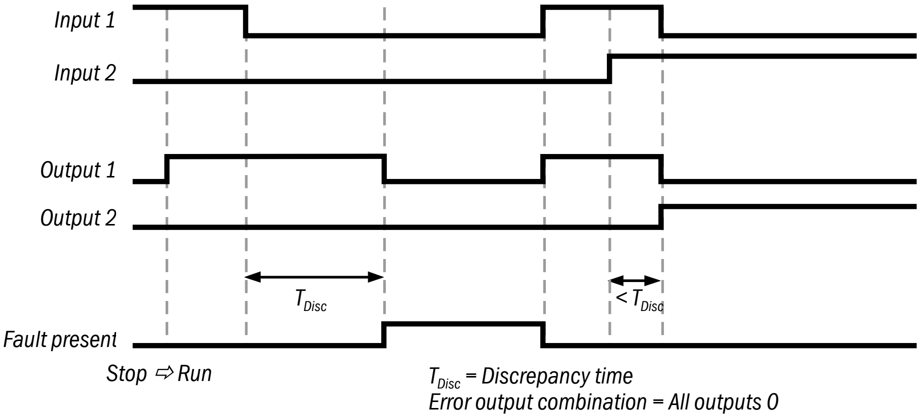


Figure 103: Sequence/timing diagram for the User mode switch function block

7.7.10 Switch synchronization

Overview

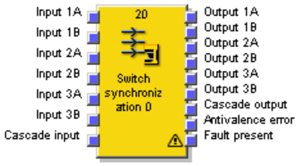


Figure 104: Inputs and outputs of the Switch synchronization function block

The Switch synchronization function block was developed to enable better integration of SICK safety laser scanners (e.g., S3000). It monitors the input signals for changes. If a change to any input signal is detected, the outputs of the function block retain their current values until the configurable **Hold time for outputs** has expired.

Function block parameters

Table 75: Parameters of the Switch synchronization function block

Parameter	Possible values
Cascade input	• With • Without
Antivalence check	• Active • Deactivated When this function is active, it is possible to use the optional Antivalence error output.
Hold time for outputs	10 ms … 10 s in 10 ms increments. The value must be greater than the logic execution time.
Invert input 1A … invert input 3B	Every input of this function block can be inverted. At an inverted input, a 0 works internally in the same way as a 1 and vice versa.
Number of inputs or Number of outputs	1 to 6
Use Fault present	• With • Without

NOTE

An inverted input also inverts the signal of the associated output. If, for example, input 1A is 1 but has been configured as inverted, it will be evaluated as 0 and output 1A will be set to 0.

Hold time for outputs

The **Hold time for outputs** determines the delay time between the moment when any input signal first undergoes a change and the moment when the input signals “kick in” by “latching”, i.e., when the outputs respond. It can be used to compensate for delays between the various contacts of mechanical switches, for example.

Non-cascaded mode – Without Cascade input

If the Switch synchronization function block is configured without the **Cascade input**, it supports the evaluation of up to three input pairs. A change in any of the input signals starts the timer. Outputs 1A through 3B retain their values until the end of the configured **Hold time for outputs.** Once the timer has finished counting down, outputs 1A through 3B assume the current values of inputs 1A through 3B whatever the result of the antivalence check. The outputs retain these values until the next synchronization process takes place.

Cascaded mode – With Cascade input

Several Switch synchronization function blocks can be combined to create a cascade so that all the outputs are switched at exactly the same time.

Cascading several Switch synchronization function blocks makes it possible to synchronize more than six inputs. When the function block is configured with the **Cascade input**, the **Cascade output** also becomes available.

NOTE

All cascaded function blocks must be configured with the same **Hold time for outputs.**

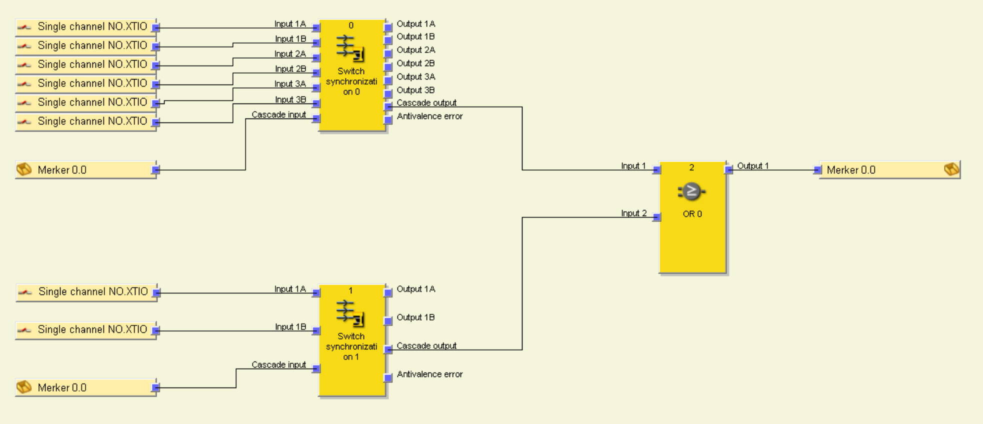


Figure 105: Logic example for two cascading Switch synchronization function blocks

The signals of all **Cascade outputs** must be fed back to the **Cascade inputs** of all the Switch synchronization function blocks that are used by means of an OR function block and a CPU marker.

NOTE

- To feed the signals back to the **Cascade inputs**, you must use a CPU marker and not a jump address. This ensures that all the associated Switch synchronization function blocks process the signal in the same logic cycle.
- When a **Cascade input** is linked by means of a CPU marker, it creates a delay. Therefore, the timer is increased by the value required to compensate for this.

A rising signal edge at the **Cascade input** starts the timer (the timer start value is the system time of the last logic cycle). Once the timer has finished counting down, outputs 1A through 3B assume the current values of inputs 1A through 3B whatever the result of the antivalence check. The outputs retain these values until the next synchronization process takes place.

Antivalence check

When this function is active, an antivalence check is performed whenever the timer has finished counting down (i.e., every time the outputs assume the current values of the inputs). If any of the input pairs used (**Input 1A** / **Input 1B** through **Input 3A** / **Input 3B**) do not have complementary values at this time (one of the inputs within each pair must be 0 and the other 1), the **Antivalence error** output is set to 1. It switches back to 0 when another synchronization process is completed without an antivalence error occurring. However, the behavior of outputs 1A through 3B is not affected by the result of the antivalence check.

NOTE

To ensure a defined combination of output values in the event of an antivalence error, you can use the Error output combination function block.

Behavior on system startup

In the event of a transition from the Stop status to the Run status, the outputs are immediately set in accordance with the input values and the antivalence check is performed (if configured). In this case, the function block does not wait for the **Hold time for outputs** to expire.

Sequence/timing diagram

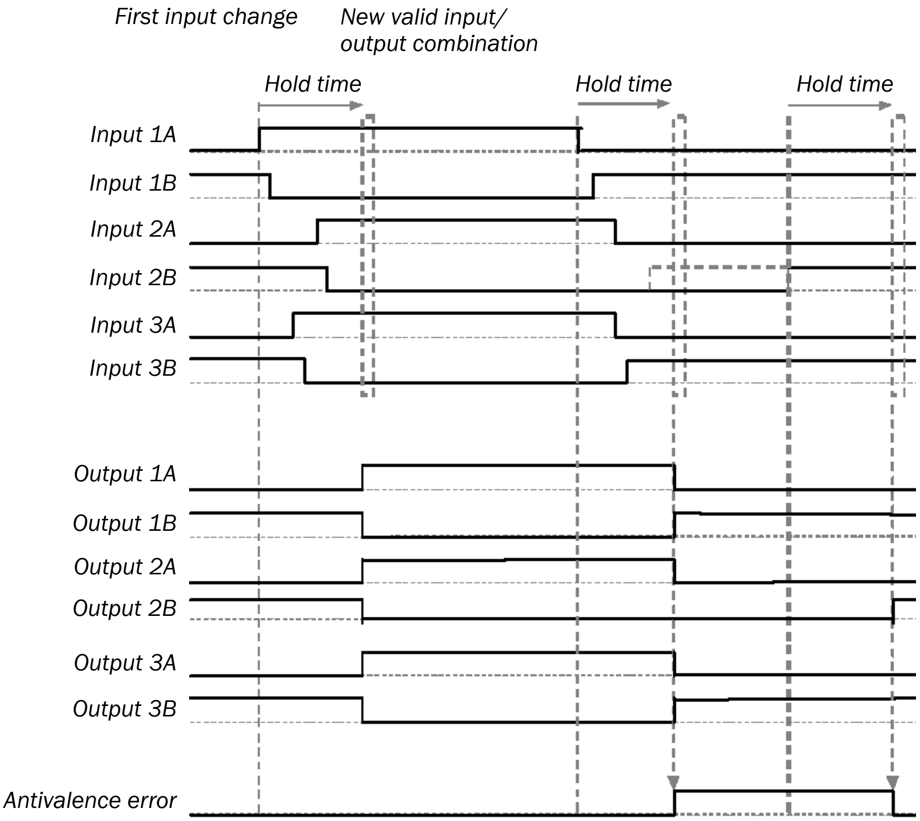


Figure 106: Sequence/timing diagram for the Switch synchronization function block without cascading

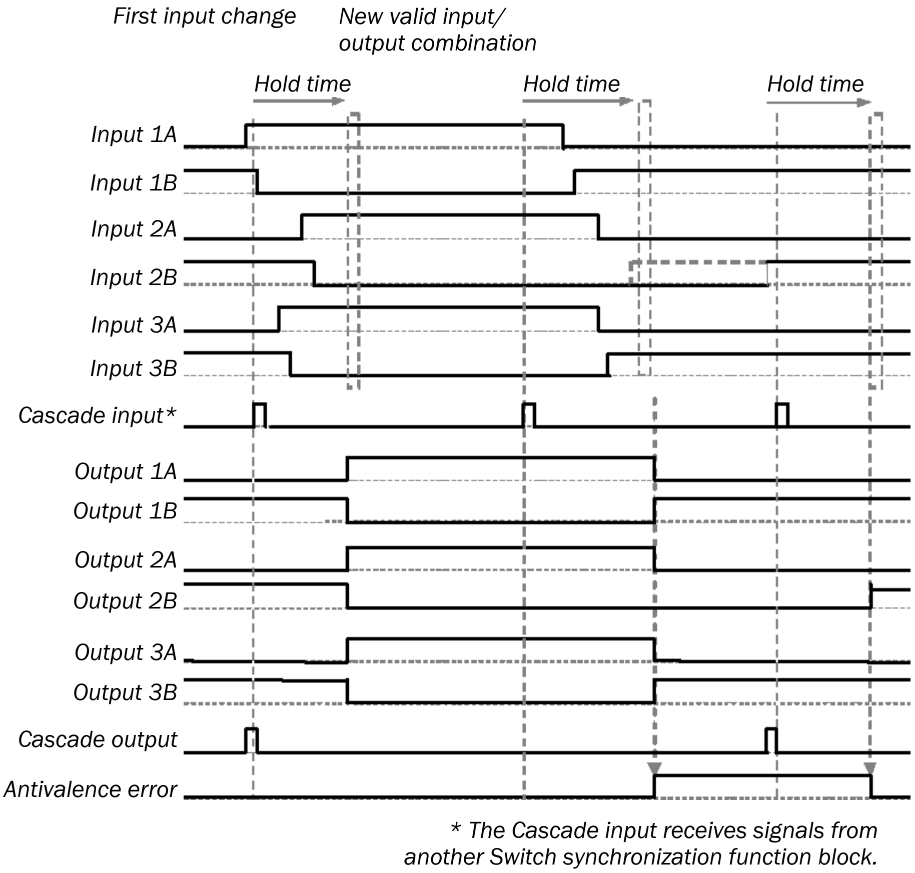


Figure 107: Sequence/timing diagram for the Switch synchronization function block with cascading

7.7.11 Error output combination

Function block diagram

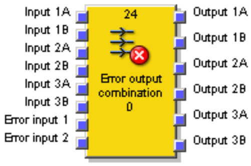


Figure 108: Inputs and outputs of the Error output combination function block

General description

The Error output combination function block was developed to enable better integration of SICK safety laser scanners (e.g. S3000). It can be used to set the outputs to preconfigured values under certain conditions, e.g. so that a dedicated error output combination is output when the Switch synchronization function block is subject to an antivalence error.

Function block parameters

Table 76: Parameters of the Error output combination function block

Parameter	Possible values
Number of error inputs	• 1 error input • 2 error inputs
Number of inputs or Number of outputs	1 to 6
Error output combination	Individually for each output: • 1 • 0

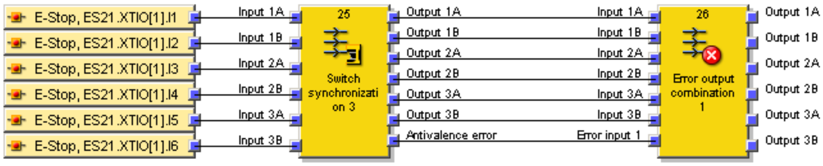


Figure 109: Logic example for the Error output combination function block

Truth table for the Error output combination function block

The following explanations apply to the truth table in this section:

- "x" signifies "any" (0 or 1).

Table 77: Truth table for the Error output combination function block

Error input 1	Error input 2	Output 1A	Output 1B	Output 2A	Output 2B	Output 3A	Output 3B
1	x	Error output combination					
x	1	Error output combination					
0	0	Input 1A	Input 1B	Input 2A	Input 2B	Input 3A	Input 3B

7.7.12 Ramp down detection

Function block diagram

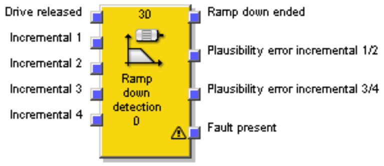


Figure 110: Inputs and outputs of the Ramp down detection function block

General description

The Ramp down detection function block checks whether a connected drive has stopped, i.e., no pulses from the incremental encoder system have been detected (e.g., from an HTL encoder or from proximity switches) for a configurable period of time. Depending on the result of this check, a safety door interlock might be released, for example.

Ramp down detection is triggered by a falling edge of the **Drive released** input signal. A drive stop is detected if no signal change (rising or falling signal edge) occurs at any of the **Incremental x** inputs for at least as long as the **Min. time between signal changes** that has been configured. In this case, the **Ramp down ended** output switches to 1. When the **Drive released** input changes to 1, this immediately sets the **Ramp down ended** output to 0 and terminates any Ramp down detection function that is currently running.

While the drive is running (**Drive released** input is 1), the **Incremental x** inputs are not monitored for signal changes. The same applies if a stop has been detected (**Ramp down ended** output 1) (see figure 115, page 149).

The function block can be used to perform an optional plausibility check on the **Incremental x** inputs to detect breaks in the wiring, provided that the incremental encoder is supplying suitable signals, e.g., complementary outputs or proximity switches and a gear wheel with a 270° tooth width and a 180° phase separation. When the plausibility check is active, at least one of the signals in each signal pair must be set to 1 at any given time. The **Plausibility error incremental** output switches to 1 when this condition is not fulfilled for two consecutive logic cycles. This means that both inputs in a pair are allowed to remain 0 for the duration of the logic execution time without this being classed as an error (see figure 116, page 149).

The **Plausibility error incremental** output is reset to 0 if at least one signal in a signal pair is 1 and the **Drive released** input is 0.

The **Fault present** output changes to 1 when any of the **Plausibility error incremental** outputs switches to 1. The **Fault present** output switches to 0 once all error outputs are 0.

Function block parameters

Table 78: Parameters of the Ramp down detection function block

Parameter	Possible values
Number of incremental inputs	• 1 single-channel incremental encoder input • 1 pair of incremental encoder inputs • 2 pairs of incremental encoder inputs
Input plausibility check	• Deactivated • Activated If this parameter is active, the number of incremental encoder inputs must either be set to 1 pair or 2 pairs.
Min. time between signal changes	100 ms to 10 s in 10 ms increments. The value must be greater than the logic execution time.
Use Fault present	• With • Without

⚠ **WARNING**

Malfunction due to incorrect configuration of incorrect incremental encoder connection

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

- ► Configure the duration of the incremental encoder signals to be at least as long as the logic execution time.
- ► Connect the signal that controls the physical output for the drive to the **Drive released** input. Care must be taken to ensure that the drive torque is definitely switched off when this input is 0.
- ► Connect the incremental encoders **locally** to an FX3-XTIO or FX3-XTDI module on the same Flexi Soft station (not via a network).

Configuration steps

- Check the maximum signal frequency of the incremental encoder signals (see step 1).
- Determine how much time is required between signal changes for the speed limit (see step 2).

Step 1: Check the maximum signal frequency of the incremental encoder signals

The minimum duration of the t_{high} and t_{low} incremental encoder signals must be greater than the logic execution time. This limits the permissible signal frequency and incremental encoder speed in accordance with the type of incremental encoder. The following figures show typical signal patterns for various types of incremental encoder:

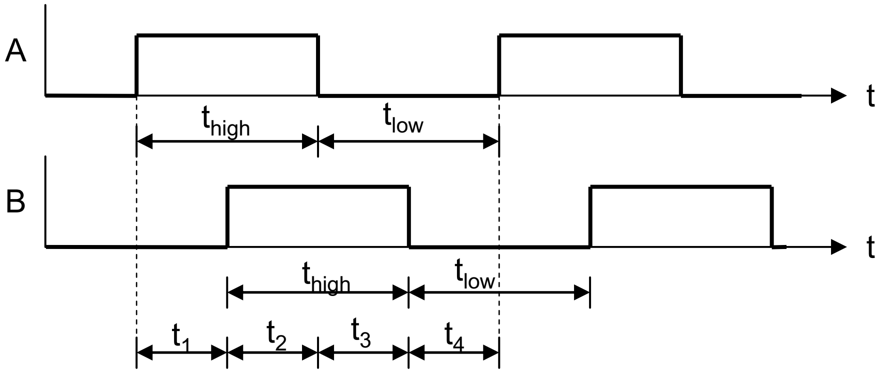


Figure 111: Signal pattern for an A/B incremental encoder with a 90° phase separation

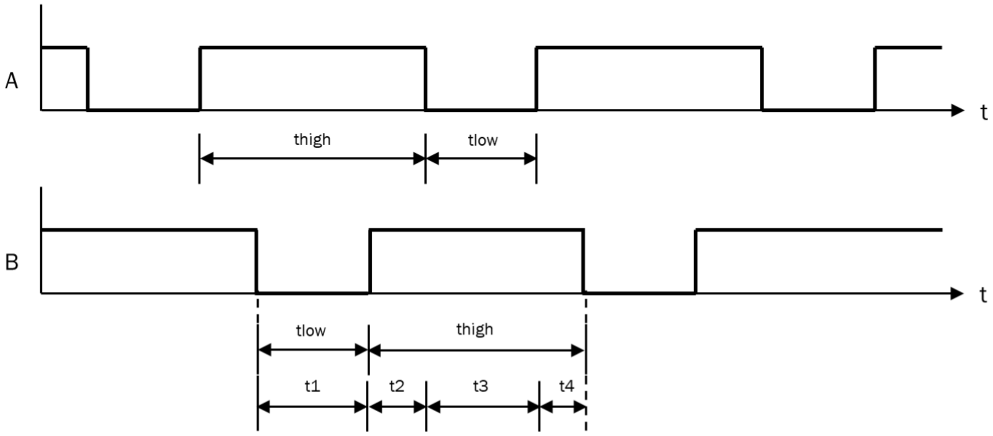


Figure 112: Signal pattern for a 1/3 gap incremental encoder with a 180° phase separation

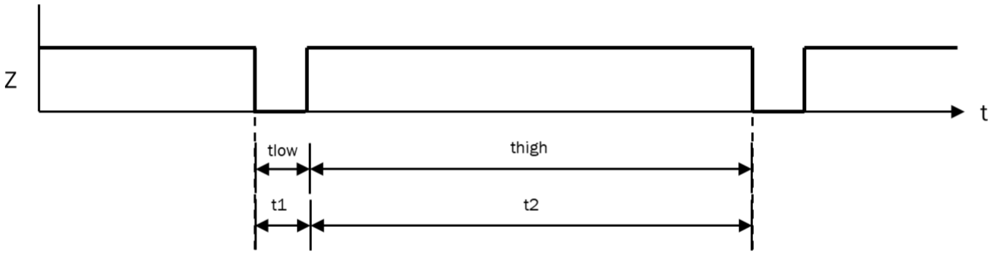


Figure 113: Signal pattern for a single incremental encoder signal

The system must be designed to ensure that the minimum duration of the t_{high} and t_{low} incremental encoder signals is always greater than the logic execution time. As part of this process, remember to take all the possible tolerance values into account, e.g., switching tolerances, gear wheel tolerances, and so on. The following table shows typical values for various types of incremental encoder:

Table 79: Maximum permissible signal frequency and speed (rpm) of incremental encoders according to type and logic execution time

Type of incremental encoder	Max. permissible incremental encoder signal frequency (Hz) for logic execution time									
	4 ms	8 ms	12 ms	16 ms	20 ms	24 ms	28 ms	32 ms	36 ms	40 ms
A/B, 90° phase separation	125.0	62.5	41.7	31.3	25.0	20.8	17.9	15.6	13.9	12.5
1/3 gap [1]	83.3	41.7	27.8	20.8	16.7	13.9	11.9	10.4	9.3	8.3
1/4 gap [1]	62.5	31.3	20.8	15.6	12.5	10.4	8.9	7.8	6.9	6.3
180° pulse	125.0	62.5	41.7	31.3	25.0	20.8	17.9	15.6	13.9	12.5

[1] 180° phase separation, at least 1 signal always set to 1.

Step 2: Determine how much time is required between signal changes for the speed limit

- ► Determine the speed at which the **Ramp down ended** output is to be activated, e.g., for the purpose of unlocking a safety door.
- ► Determine the maximum time between two signal changes for this speed (highest values from t_1 through t_4). As part of this process, remember to take all the possible tolerance values into account, e.g., switching tolerances, gear wheel tolerances, and so on.
  Min. time between signal changes = highest values from t_1 through t_4 + 10 ms

The **min. time between signal changes** must always be greater than the logic execution time **and** has to be rounded up to the next multiple of 10 ms.

WARNING
Extended logic execution time

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

- ► Every time you change the logic program, check if the logic execution time has been extended.
- ► It may be necessary to recalculate the maximum signal frequency of the incremental encoders.

Example 1: A/B, 90° phase separation

- 4 teeth per revolution
- Switching tolerances ±5° ➜ teeth 175° to 185° (corresponds to t_{low}, t_{high}); signal change 85° to 95° (corresponds to t_1 through t_4)
- Maximum drive speed = 750 rpm = 12.5 Hz
- Drive speed for release = 15 rpm = 0.25 Hz
- Logic execution time = 8 ms

- ► Check the maximum signal frequency of the incremental encoder signals:
  Max. signal frequency = 12.5 Hz × 4 teeth/revolution = 50 Hz
  Lowest t_{low} = 1/50 Hz × 175°/360° = 9.7 ms
  ➜ Greater than logic execution time ✔
  Lowest t_{high} = 1/50 Hz × 175°/360° = 9.7 ms
  ➜ Greater than logic execution time ✔
- ► Determine how much time is required between signal changes for the speed limit:

Signal frequency for release = 0.25 Hz × 4 teeth/revolution = 1 Hz
Max. duration of input pattern = 1/1 Hz × 185°/360° = 514 ms
Time between signal changes = 514 ms + 10 ms = 524 ms
➜ Min. time between signal changes = 530 ms (rounded up to next multiple of 10 ms)

Example 2: 1/3 gap, 180° phase separation

- 8 teeth per revolution
- Switching tolerances ±2° ➜ teeth 118° to 122° (corresponds to t_{low}, t_{high}); signal change 118° to 122° (corresponds to t_1 through t_4)
- Maximum drive speed = 120 rpm = 2 Hz
- Drive speed for release = 12 rpm = 0.2 Hz
- Logic execution time = 16 ms

► Check the maximum signal frequency of the incremental encoder signals:
Max. signal frequency = 2 Hz × 8 teeth/revolution = 16 Hz
Lowest t_{low} = 1/16 Hz × 118°/360° = 20.5 ms
➜ Greater than logic execution time ✔
Lowest t_{high} = 1/16 Hz × 238°/360° = 41.3 ms
➜ Greater than logic execution time ✔

► Determine how much time is required between signal changes for the speed limit:
Signal frequency for release = 0.2 Hz × 8 teeth/revolution = 1.6 Hz
Max. duration of input pattern = 1/1.6 Hz × 122°/360° = 212 ms
Time between signal changes = 212 ms + 10 ms = 222 ms
➜ Min. time between signal changes = 230 ms (rounded up to next multiple of 10 ms)

Example 3: Zero pulse 10°

- 1 tooth per revolution
- Switching tolerances ±1° ➜ tooth 9° to 11° (corresponds to t_{low}, t_{high}); signal change 349° to 351° (corresponds to t_1 through t_4)
- Maximum drive speed = 300 rpm = 5 Hz
- Drive speed for release = 3 rpm = 0.05 Hz
- Logic execution time = 4 ms

► Check the maximum signal frequency of the incremental encoder signals:
Max. signal frequency = 5 Hz × 1 tooth/revolution = 5 Hz
Lowest t_{low} = 1/5 Hz × 9°/360° = 5 ms
➜ Greater than logic execution time ✔
Lowest t_{high} = 1/5 Hz × 351°/360° = 195 ms
➜ Greater than logic execution time ✔

► Determine how much time is required between signal changes for the speed limit:
Signal frequency for release = 0.05 Hz × 1 tooth/revolution = 0.05 Hz
Max. duration of input pattern = 1/0.05 Hz × 11°/360° = 611 ms
Time between signal changes = 611 ms + 10 ms = 621 ms
➜ Min. time between signal changes = 630 ms (rounded up to next multiple of 10 ms)

Example logic

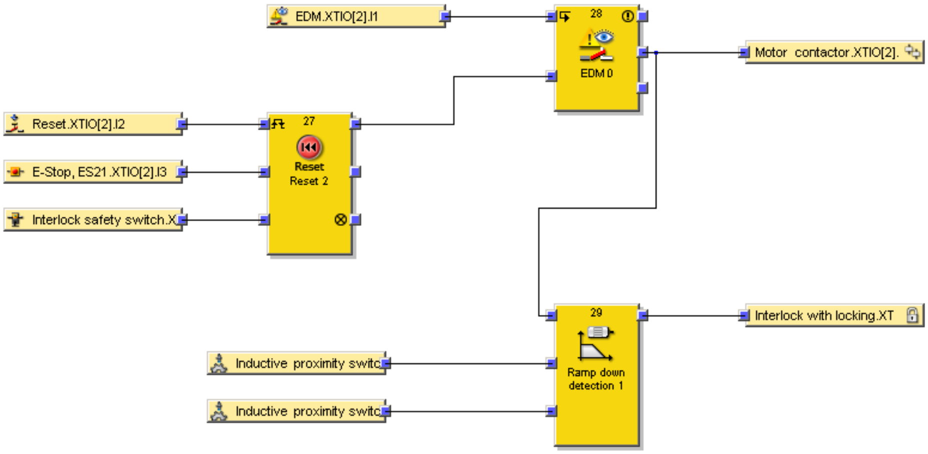


Figure 114: Logic example for the Trail detection function block

Sequence/timing diagrams

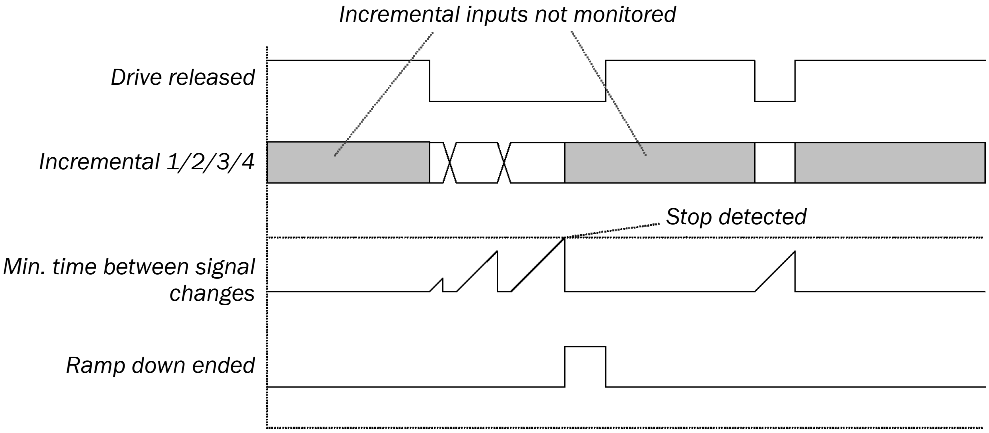

Incremental inputs not monitored
Drive released
Incremental 1/2/3/4
Stop detected
Min. time between signal changes
Ramp down ended

Figure 115: Sequence/timing diagram for the Ramp down detection function block

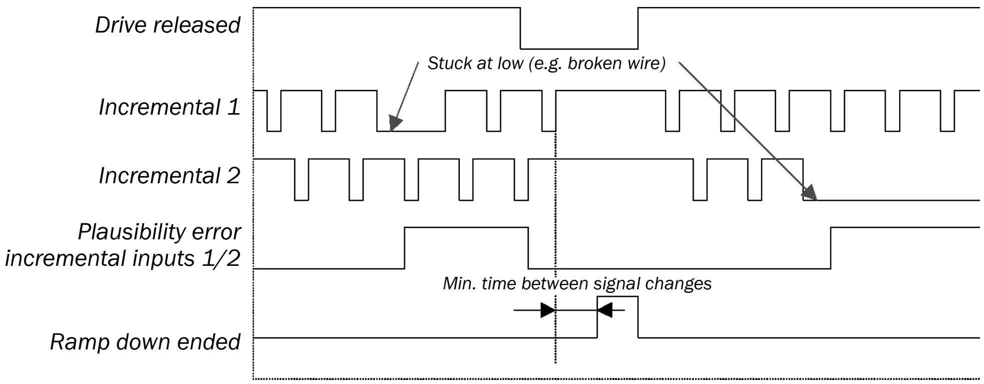

Drive released
Stuck at low (e.g. broken wire)
Incremental 1
Incremental 2
Plausibility error incremental inputs 1/2
Min. time between signal changes
Ramp down ended

Figure 116: Sequence/timing diagram for the Ramp down detection function block with plausibility check

7.7.13 Frequency monitoring

Function block diagram

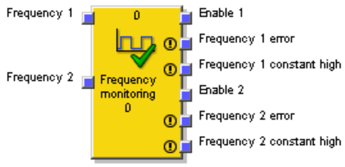


Figure 117: Inputs and outputs of the Frequency monitoring function block

General description

The Frequency monitoring function block can be used to monitor the frequency or period duration of up to two signals separately. In addition, the pulse duration can be optionally monitored (t_{high}). For instance, this can be used to evaluate signal sources that output a pulse signal with a particular frequency to serve as an enable signal.

Function block parameters

Table 80: Parameters of the Frequency monitoring function block

Parameters	Possible values for Frequency 1 and Frequency 2
Min. period time	20 ms – 2.54 s in 10 ms increments The value must be at least as great as two times the logic execution time.
Max. period time	30 ms – 2.55 s in 10 ms increments The value must be at least as great as the min. period time + the logic execution time.
Average pulse duration (t_{high})	0 = infinite, 10 ms – 2.53 s in 10 ms increments When the setting is 0 = infinite, the pulse duration is not evaluated. This means that the pulse duration is always classed as valid for evaluation purposes. If the value is anything other than 0, it must fulfill the following conditions: • > 2 × logic execution time and • < (min. period duration – pulse duration tolerance)
Pulse duration tolerance (t_{high})	0 ms to 310 ms in 10 ms increments The value may only be 0 if the average pulse duration is 0 as well. If the value is not 0, it must be greater than the logic execution time.
Frequency x error output	• All errors • Only when Frequency x is constantly High
Use Fault present	• With • Without This parameter applies to the function block and so covers both Frequency 1 and Frequency 2 jointly.

Monitoring accuracy

WARNING
Malfunction due to unsuitable incremental encoder signals

The dangerous state may not be stopped at all or in a timely manner in the event non-compliance, since a higher frequency (lower period duration) is not recognized.

► Only use incremental encoder signals with a pulse duration (t_{high}) and pulse gap (t_{low}) which are greater than the logic execution time.

Limits for a reliably valid signal

WARNING
Malfunction due to unsuitable incremental encoder signals

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

► Make sure the incremental encoder signals used are valid.

The limits for the average period duration that a signal must achieve in order to be reliably classed as valid are actually narrower than the limits selected via the parameters. The effective narrower limits are always the next multiple of the logic execution time. Within this context, what the average period duration means is that although individual periods of the signal are allowed to exhibit extremes (jitter), these must be evened out over the course of several periods.

Table 81: Examples of effective limits for the period duration

Logic execution time	Set parameters		Effective limits for a reliably valid signal	
	Min. period time	Max. period time	Min. period time	Max. period time
4 ms	120 ms	160 ms	120 ms	160 ms
12 ms	120 ms	160 ms	120 ms	156 ms
32 ms	120 ms	160 ms	128 ms	160 ms

Limits for a reliably invalid signal

The limit for the average period duration that a signal must exceed in order to be reliably classed as invalid corresponds to the tolerance for the associated parameters.

Essentially, this means that a permanently 0 or 1 state will be detected as an invalid signal at the latest after the max. period duration + the logic execution time +10 ms. The response time of the signal path used is increased by this amount.

If the average period duration of the signal exceeds the limit for the signal to be reliably classed as valid but is still below the limit for it to be reliably classed as invalid, it may take several periods for the deviations to accumulate sufficiently. As a result, the signal may only become classed as invalid at this point:

Number of periods = (logic execution time +10 ms) / (real averaged period duration – effective limit for signal to be reliably classed as valid)

NOTE
In the following description, the “x” in the signal names either means 1 or 2, i.e., the index number for either of the separate monitoring functions in the function block.

The function block is capable of detecting the following invalid signals:

- ① The detected period duration is too short: The time between the rising or falling signal edges at the **Frequency x** input is shorter than the **Min. period duration**. This monitoring function is triggered by the first rising signal edge after a transition from the Stop status to the Run status.
- ② The detected period duration is too long: The time between the rising or falling signal edges at the **Frequency x** input is longer than the **Max. period duration**. This monitoring function is triggered by the first rising signal edge after a transition from the Stop status to the Run status.
- ③ The detected pulse duration is too short: Pulse duration monitoring is activated (**Average pulse duration** is set to a value other than 0) and the time between the last rising signal edge and the last falling signal edge at the **Frequency x** input is shorter than **Average pulse duration – Pulse duration tolerance**. This monitoring function is triggered by the first rising signal edge after a transition from the Stop status to the Run status.
- ④ The detected pulse duration is too long: Pulse duration monitoring is activated (**Average pulse duration** is set to a value other than 0) and the time since the last rising signal edge at the **Frequency x** input is longer than **Average pulse duration + Pulse duration tolerance**. In other words, no falling signal edge has been detected within the expected time. This monitoring function is triggered by the first rising signal edge after a transition from the Stop status to the Run status.
- ⑤**Frequency x** input is constantly 1: The **Frequency x** input remains set to 1 for longer than the **Max. period duration**. This monitoring function is triggered immediately after a transition from the Stop status to the Run status.

The **Enable x** output switches to 1 once two periods with a valid period duration and a valid pulse duration have been detected at the **Frequency x** input. If pulse duration monitoring is deactivated, the pulse duration is always classed as valid for evaluation purposes.

The **Enable x** output switches to 0 when an invalid signal is detected at the **Frequency x** input, i.e., if ...

- ① the detected period duration is too short or
- ② the detected period duration is too long or
- ③ the detected pulse duration is too short and pulse duration monitoring has been activated or
- ④ the detected pulse duration is too long and pulse duration monitoring has been activated.

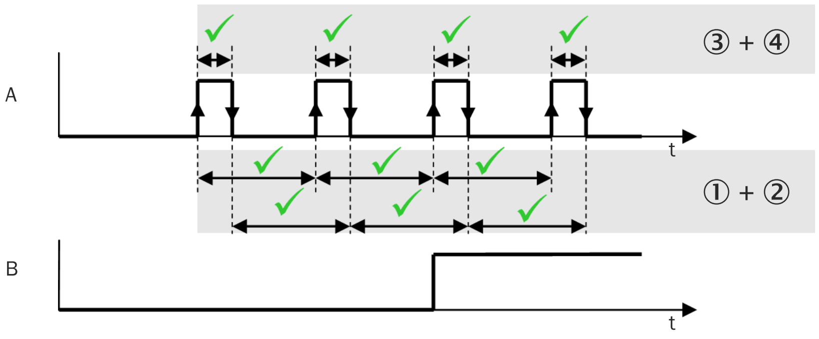


Figure 118: Sequence/timing diagram for the Frequency monitoring function block, with activation of the Enable x output

A: Frequency x

B: Enable x

The **Frequency x error** output switches to 1 if ...

- (a) the **Frequency x error output** parameter = **All errors** and

  - ① the detected period duration is too short or
  - ② the detected period duration is too long or
  - ③ the detected pulse duration is too short and pulse duration monitoring has been activated or
  - ④ the detected pulse duration is too long and pulse duration monitoring has been activated or
  - ⑤ the **Frequency x constant high** input is set to 1
- (b) the **Frequency x error output** parameter = **Only when Frequency x is constantly High** and
  - ⑤ the **Frequency x constant high** input is set to 1.

The **Frequency x constant high** output switches to 1 if ...

- ⑤ the **Frequency x** input is constantly 1.

The **Fault present** output switches to 1 if ...

- the **Frequency 1 error** output is set to 1, or
- the **Frequency 2 error** output is set to 1, or
- the **Frequency 1 constant high** output is set to 1, or
- the **Frequency 2 constant high** output is set to 1.

The **Frequency x error** and **Fault present** outputs switch back to 0 when the **Enable x** output changes to 1, i.e., once two periods with a valid period duration and a valid pulse duration have been detected at the **Frequency x** input.

The **Frequency x constant high** output switches back to 0 when the **Frequency x** input changes to 0.

After a transition from the Stop status to the Run status, all outputs are set to 0.

Sequence/timing diagrams

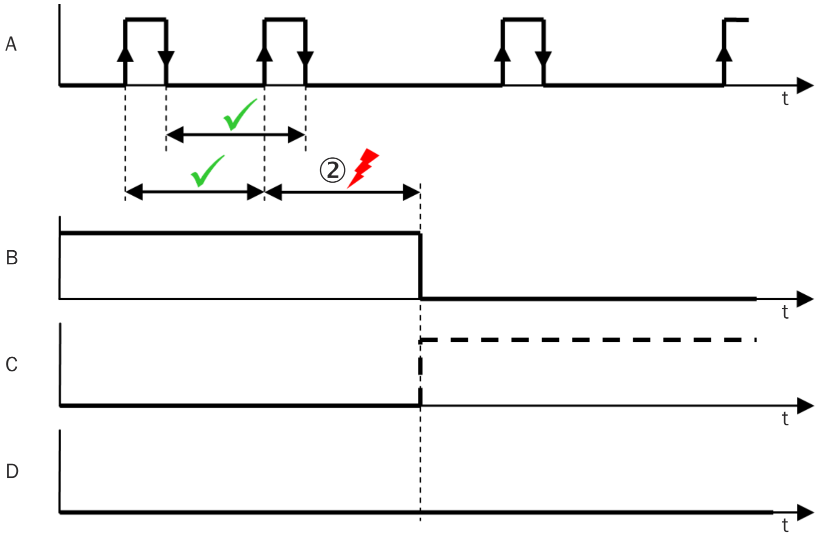


Figure 119: Sequence/timing diagram for the Frequency monitoring function block, period duration too long

A: Frequency x

B: Enable x

C: Frequency x error

D: Frequency x constant high

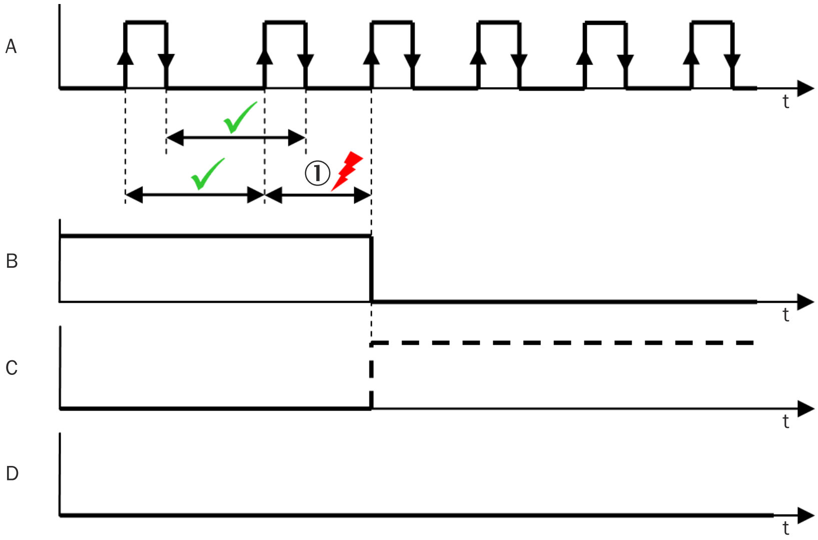


Figure 120: Sequence/timing diagram for the Frequency monitoring function block, period duration too short

A: Frequency x

B: Enable x

C: Frequency x error

D: Frequency x constant high

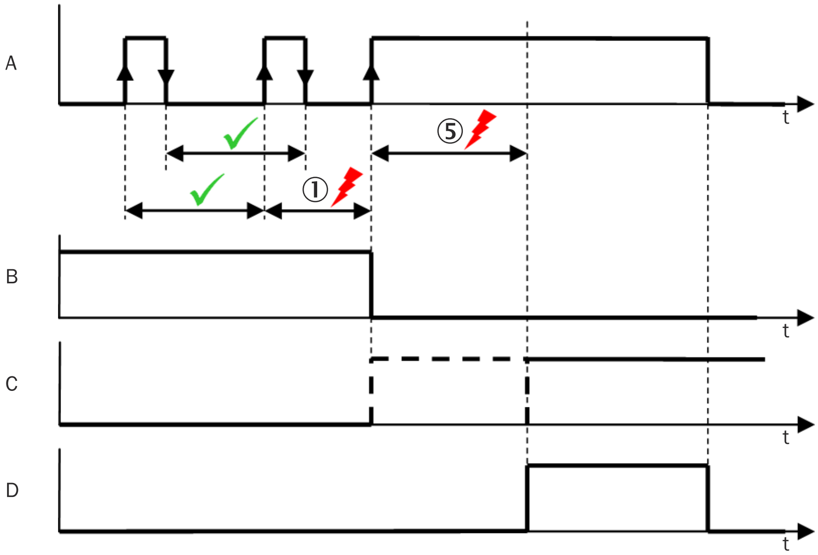


Figure 121: Sequence/timing diagram for the Frequency monitoring function block, frequency x constant 1

A: Frequency x

B: Enable x

C: Frequency x error

D: Frequency x constant high

7.7.14 Start warning

Function block diagram

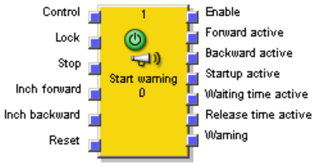


Figure 122: Inputs and outputs of the Start warning function block

General description

Many machines have to feature a start warning, e.g., if the machine is so big that the machine operator cannot survey all the hazardous areas from a single location.

As soon as a start button is pressed, the waiting time commences and a warning signal is triggered. At the end of the waiting time, the release time commences and a second warning signal is triggered. While the release time is running, the machine can be started by pressing the start button again.

WARNING
Unexpected machine startup

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

► Configure a startup warning for every operational status which could put the machine in a dangerous state.

Start sequence

1. At startup, the function block is in deactivated mode. The **Startup active** output is 1 but all the other outputs are 0.
2. If the **Control** input switches to 0 and the **Lock** and **Stop** inputs are set to 1, the start sequence is enabled and the function block switches to Wait for start mode.
3. A rising signal edge at the **Inch forward** input or at the **Inch backward** input triggers the start sequence:
   - The **Startup active** output switches to 0, the waiting time and signal time both start running, and the **Waiting time active** and the **Warning** outputs switch to 1 for the duration of the signal time.
   - At the end of the **Waiting time**, the **Release time** and the **Pulse time** both start running. The **Waiting time active** output switches back to 0, the **Release time active** output changes to 1, and the **Warning** output also returns to 1 for the duration of the pulse time.
4. If a second rising edge occurs at the **Inch forward** input during the release time, the function block switches to inching (forward) mode and the **Enable** and **Forward active** outputs change to 1. A similar situation applies if a second rising edge occurs at the **Inch backward** input during the release time: In this case, the function block switches to inching (backward) mode and the **Enable** and **Backward active** outputs change to 1.

NOTE
This rule is subject to a restriction when the mode is set to “Direction switching locked” (see below).

5. If the release time expires but there has been no transition to inching mode within this period, the function block switches back to the “Wait for start” mode and the entire start sequence must be performed again from scratch.
6. There is no limit to how long inching mode can remain active. It is terminated when the active (1) input (**Inch forward** or **Inch backward**) switches back to 0. In this case, the **Enable** and **Forward active** or **Backward active** outputs return to 0. Inching mode is also terminated when both inputs (**Inch forward** and **Inch backward**) are set to 1 at the same time. On termination of inching mode, the release time starts running again. This means that another rising signal edge at the **Inch forward** or **Inch backward** input will restart inching mode immediately without the need for a new start sequence. If the release time expires but there has been no transition to inching mode within this period, the function block switches back to the Wait for start mode and the entire start sequence must be performed again from scratch.
7. Inching mode is likewise terminated when a falling signal edge occurs at the **Reset** or **Stop** inputs. In this case, the function block switches back to the “Wait for start” mode and the entire start sequence must be performed again from scratch.

Function block parameters

Table 82: Parameters of the Start warning function block

Parameter	Possible values
Direction switching	• Locked • Not locked
Waiting time	1 … 60 s in 10 ms increments. The value must be greater than the logic execution time.
Release time	1 … 600 s in 10 ms increments. The value must be greater than the logic execution time.
Signal time	0 … 60 s in 10 ms increments. If the value is anything other than 0, it must be longer than the logic execution time but shorter than the waiting time.
Pulse time	0 … 600 s in 10 ms increments. If the value is anything other than 0, it must be longer than the logic execution time but shorter than the release time.

Direction switching

This parameter determines whether it is possible to switch between forward and backward operation without having to perform a complete start sequence first. When the parameter is set to Not locked, it is possible to initiate the start sequence with either of the two inputs (e.g., **Inch forward**) and confirm it with the other input (e.g., **Inch backward**). In Not locked mode, it is also possible to switch the direction during inching mode without having to perform a complete start sequence from scratch.

With the Locked setting, the same input (**Inch forward** or **Inch backward**) that was used to initiate the start sequence must be used to confirm it (during the release time). A rising signal edge at the other input will restart the waiting time instead of the start sequence. This makes it impossible to switch direction in inching mode. To switch direction, you must perform the complete start sequence from scratch (see figure 125, page 160).

Waiting time

The **Waiting time** parameter determines how much time should elapse from when the first rising signal edge occurs at the **Inch forward** or **Inch backward** input until the **Release time** starts running.

Release time

At the end of the **Waiting time**, the **Release time** starts running. During the **Release time**, a rising signal edge at either the **Inch forward** or **Inch backward** inputs starts up the machine (regardless of the setting for the **Direction switching** parameter).

Signal time

The **Signal time** starts running in parallel with the **Waiting time.** During the **Signal time**, the **Warning** output is set to 1 to indicate that a start sequence has been initiated.

Pulse time

The **Pulse time** starts running in parallel with the **Release time.** During the **Pulse time**, the **Warning** output switches back to 1 to indicate that inching mode can now be started. If inching mode is started during the **Pulse time**, this has no impact on the **Pulse time**, i.e., the **Warning** output remains set to 1 until the configured **Pulse time** has expired.

NOTE
The second warning pulse is not mandatory and can be deactivated by setting the pulse time to 0 s.

Control input

A start sequence can only be initiated if the **Control** input is set to 0. If the **Control** input switches to 1 during a start sequence, the current start sequence is aborted and a new one can only be initiated once the **Control** input has returned to 0.

Lock input

A start sequence can only be initiated if the **Lock** input is set to 1. If the **Lock** input switches to 0 during a start sequence, the current start sequence is aborted and a new one can only be initiated once the **Lock** input has returned to 1. This input can be used for safety stops.

If inching mode is active, a falling signal edge at the **Lock** input terminates inching mode and resets the function block to the “Wait for start” mode.

Stop input

A start sequence can only be initiated if the **Stop** input is set to 1. If the **Stop** input switches to 0 during a start sequence, the current start sequence is aborted and a new one can only be initiated once the **Stop** input has returned to 1. This input can be used for safety stops.

If inching mode is active, a falling signal edge at the **Stop** input terminates inching mode and resets the function block to the “Wait for start” mode.

Inch forward/Inch backward inputs

If a rising signal edge (0–1) is detected at either the **Inch forward** input or the **Inch backward** input but the other one remains set to 0, the start sequence commences.

NOTE
If a rising signal edge is detected at both inputs or if a rising signal edge is detected at one input while the other input is set to 1, the input status is classed as invalid. If a status of this kind occurs during a start sequence (during the waiting time or during the release time), the waiting time starts running again from the beginning. If a status of this kind occurs during inching mode, inching mode is terminated and the release time starts running again from the beginning.

Reset input

A falling signal edge at the **Reset** input forces the start sequence to begin again. If inching mode is active, it is terminated and the function block switches back to the “Wait for start” mode. The **Release** output plus the **Forward active** and **Backward active** outputs switch to 0 while the **Startup active** output changes to 1.

Startup active output

The **Startup active** output is set to 0 during a start sequence (during the waiting time or the release time) or if inching mode is active (the **Release** output is 1). The **Startup active** output can be used as an interlock to prevent other instances of the Start warning function block from running in parallel. For this, the **Startup active** output must be connected to the **Lock** input of the other instance of the function block using a CPU marker.

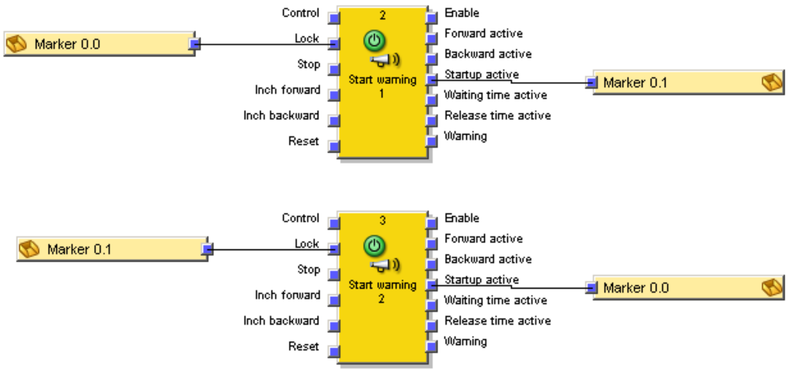


Figure 123: Logic example for the combination of two Start warning function blocks

Waiting time active output and Release time active output

These outputs indicate whether the waiting time or release time is active.

Sequence/timing diagrams

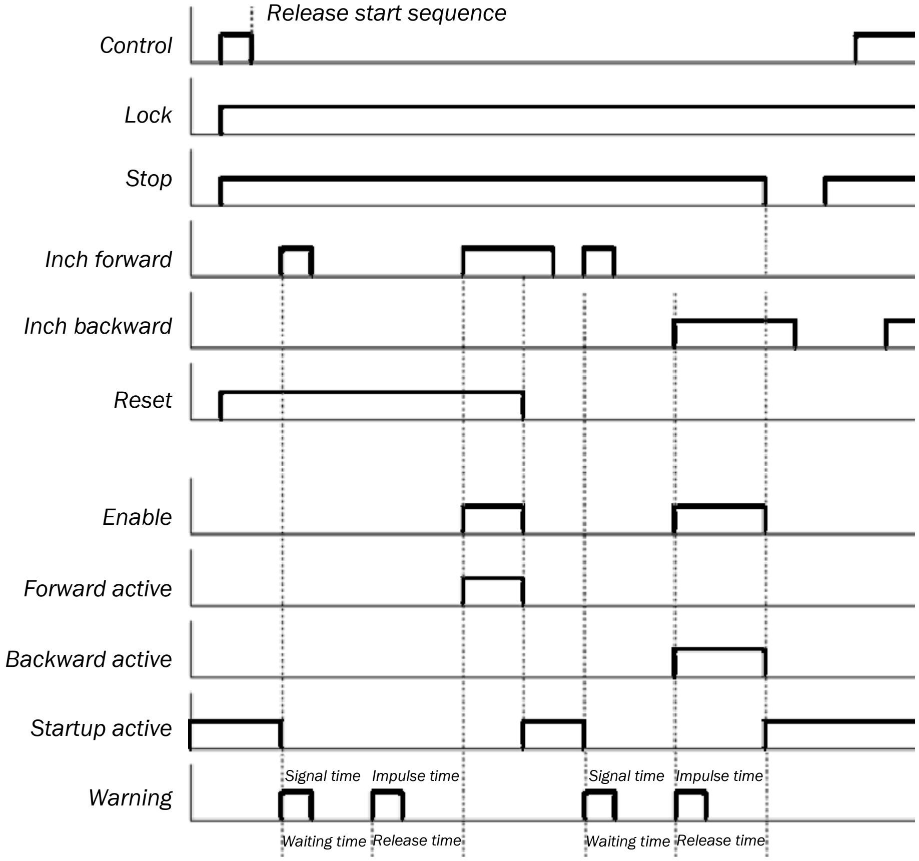


Figure 124: Sequence/timing diagram for the Start warning function block in Not locked mode

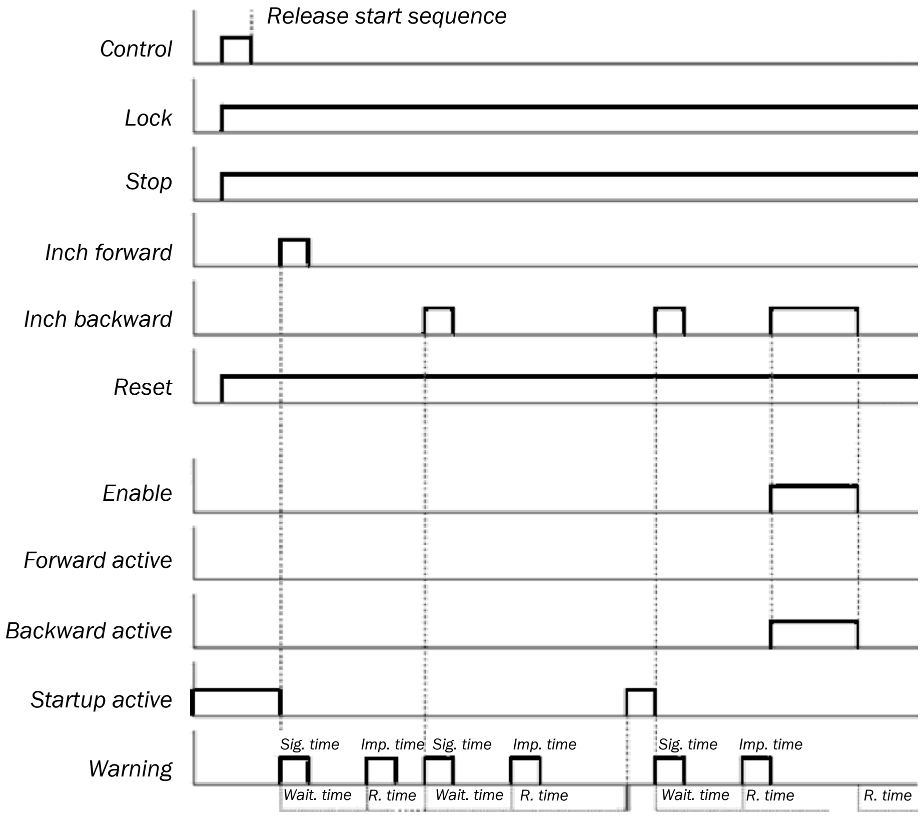


Figure 125: Sequence/timing diagram for the Start warning function block in Locked mode

NOTE

- The start sequence is initiated by a rising signal edge at the **Inch forward** input.
- A rising signal edge at the **Inch backward** input causes the **Waiting time** to restart during the start sequence.
- A rising signal edge at the **Inch backward** input terminates inching mode if the **Inch forward** input is set to 1 at the same time.

7.8 Function blocks for dual-channel evaluation

The Flexi Soft system supports applications up to SIL3 (IEC 61508, IEC 62061) and performance level PL e (ISO 13849). One or two safety signals connected locally to the Flexi Soft safety controller can serve as possible sources for the function block inputs.

The following types of input evaluation can be selected (depending on the function block):

- Single-channel
- Dual-channel
  - Dual-channel equivalent (1 pair)
  - Dual-channel complementary (1 pair)
  - Dual-channel equivalent (2 pairs)
  - Dual-channel complementary (2 pairs)

The following truth tables summarize how each input signal evaluation method for the Flexi Soft safety controller is evaluated internally.

Truth tables

The following explanations apply to the truth tables in this section:

- “x” signifies “any” (0 or 1).

NOTE

The **Fault present** output is set to 1 if the logic processing component of the Flexi Soft safety controller detects an erroneous combination or sequence of input signals.

7.8.1 Single-channel evaluation

NOTE

The section below relates to the Safety gate monitoring and Emergency stop function blocks.

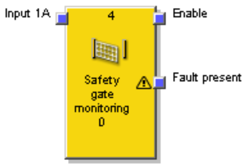


Figure 126: Example of single-channel analysis

The single-channel evaluation of these function blocks has no functional use when the function test is deactivated. However, a function block configured in this way can help you to create a clearer layout for the logic program. If this is not necessary, this function block can be omitted.

7.8.2 Dual-channel evaluation (1 pair) and discrepancy time

NOTE

- This section deals with the following function blocks: Safety gate monitoring, Emergency stop, Light curtain monitoring, Magnetic switch, Two hand control type IIIA, and Two hand control type IIIC.
- It does not cover the Tolerant dual-channel evaluation function block.

Please note that expansion modules such as FX3-XTIO or FX3-XTDI are capable of performing dual-channel evaluation if predefined input elements from the element window (e.g., RE27, deTec4, etc.) are connected to them. When you use these input elements, you do not need to have a separate function block for dual-channel evaluation (e.g., light grid monitoring, safety gate monitoring, or magnetic switch).

Alternatively, input signals that are not subject to preliminary evaluation can be connected to both input channels of a function block with a dual-channel input configuration. In this case, dual-channel evaluation takes place within the function block.

The disadvantage of this alternative is that it requires an additional function block, which may result in a longer logic execution time. The advantage is that the function block output makes a discrepancy error available in the logic and this can be evaluated.

The following function blocks generate the same output value for a dual-channel input signal that has undergone preliminary evaluation by the I/O module.

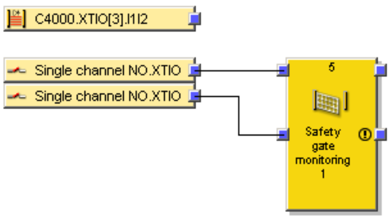


Figure 127: Dual-channel evaluation with I/O module or with function block

Dual-channel evaluation checks whether the sequence of the two input signals is correct. If either of the two signals has triggered a switch-off, the other signal is expected to follow accordingly. The value that the two signals are required to have is dependent on the type of dual-channel evaluation. There are two options:

- Equivalent evaluation
- Complementary evaluation

An optional discrepancy time can be defined. The discrepancy time defines how long the two inputs can continue to have discrepant values after a change in either of the input signals without this being regarded as an error.

The following truth table describes the discrepancy conditions for dual-channel equivalent and dual-channel complementary input evaluation:

Table 83: Dual-channel evaluation

Evaluation type	Input A	Input B	Discrepancy timer [1]	Status of dual-channel evaluation	Enable output	Discrepancy error output
Equivalent	0	0	0	Deactivated	0	Unchanged [2]
	0	1	< discrepancy time	Discrepant	0	Unchanged [2]
	1	0	< discrepancy time	Discrepant	0	Unchanged [2]
	1	1	0	Active [3]	1	0
	x	x	≥ discrepancy time (timeout)	Error	0	1
Complementary	0	1	0	Deactivated	0	Unchanged [2]
	0	0	< discrepancy time	Discrepant	0	Unchanged [2]
	1	1	< discrepancy time	Discrepant	0	Unchanged [2]
	1	0	0	Active [3]	1	0
	x	x	≥ discrepancy time (timeout)	Error	0	1

[1] If the discrepancy time is active (> 0), the discrepancy timer is restarted on the first signal change that leads to a discrepant status. If the discrepancy time is deactivated (= 0), the discrepancy timer is not started, i.e., a timeout never occurs.
[2] Unchanged = the last status is retained.
[3] If the correct sequence has been observed.

A dual-channel evaluation can only switch to Active (Release output changes from 0 to 1) if the following conditions are met:

- The status has been set to Deactivated at least once since it was last Active. It is not possible to switch from Active to Discrepant and then back to Active.
- The discrepancy time has either not yet elapsed or is fully deactivated.

A discrepancy error (timeout) is reset when the Active status is achieved, i.e., when the **Release** output switches to 1.

NOTE

The following must be taken into account when defining values for the discrepancy time:

- The discrepancy time must exceed the logic execution time.
- The discrepancy time has an accuracy of ± 10 ms plus the logic execution time. The logic execution time is dependent on the type and number of function blocks used. It is displayed on the **FB group info** tab of the logic editor and also in the report.
- If signals from tested sensors are connected to FX3-XTIO or FX3-XTDI modules, the discrepancy time must exceed the **test gap (ms)** + the **max. OFF-ON delay (ms)** of the test output that is being used, as a signal change at the input of the modules may be delayed by this amount of time. You will find these values listed in the report under **Configuration**, **I/O module**, **Test pulse parameters.**
- If both inputs in a pair are connected to the same input signal, the evaluation function works in the same way as single-channel evaluation, i.e., no equivalent or complementary evaluation takes place and the discrepancy time is not monitored.

Sequence/timing diagram

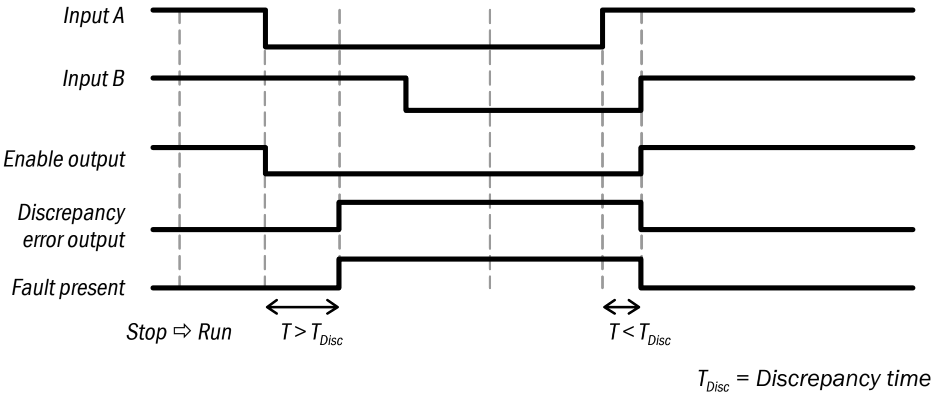


Figure 128: Sequence/timing diagram for the Emergency stop function block – dual-channel equivalent logic

7.8.3 Double dual-channel evaluation and synchronization time

Overview

This section deals with the following function blocks: Safety gate monitoring and Two hand control type IIIC.

Important information

NOTE

All time parameters: To ensure the accuracy of the total signal path, the 1 x logic execution time must also be taken into consideration.

Double dual-channel evaluation and synchronization time

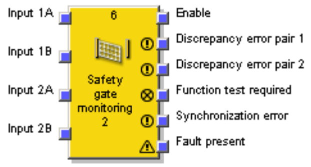


Figure 129: Double dual-channel evaluation with the Safety gate monitoring function block

Double dual-channel evaluation checks each pair of inputs to see whether the sequence of the two input signals is correct in both cases. In addition, both dual-channel evaluation functions are monitored in relation to one another to make sure they are performed in the correct sequence. If either of the two dual-channel evaluation functions has triggered a switch-off, the other dual-channel evaluation function is expected to follow accordingly.

An optional synchronization time can be defined. The synchronization time defines how long the two dual-channel evaluation functions can continue to have non-synchronous statuses without this being regarded as an error.

The synchronization time is different from the discrepancy time in that it evaluates the relationship between the two dual-channel evaluation functions. By contrast, the discrepancy time relates to one pair of inputs for one dual-channel evaluation function.

The following table describes the synchronization conditions for double dual-channel evaluation (2 pairs):

Table 84: Double dual-channel evaluation (synchronization evaluation)

Status of dual-channel evaluation pair 1	Status of dual-channel evaluation pair 2	Synchronization timer [1]	Synchronization status	Enable output	Synchronization error output
Deactivated or Discrepant	Deactivated or Discrepant	0	Deactivated	0	Unchanged [2]
Deactivated or Discrepant	Active	< synchronization time	Discrepant	0	Unchanged
Active	Deactivated or Discrepant	< synchronization time	Discrepant	0	Unchanged
Active	Active	0	Active [3]	1	0
x	x	≥ synchronization time (timeout)	Error	0	1

[1] If the synchronization time is active (> 0), the synchronization timer is restarted following the first status change that leads to a discrepant synchronization status. If the synchronization time is deactivated (= 0), the synchronization timer is not started, i.e., a timeout never occurs.

[2] Unchanged = the last status is retained.

[3] If the correct sequence has been observed.

Synchronization evaluation

The synchronization evaluation can only switch to Active (Release output changes from 0 to 1) if the following conditions are met:

- The synchronization status has been set to Deactivated at least once since it was last Active. In the case of the Two hand control type IIIC function block, both dual-channel evaluation functions must be set to Deactivated at the same time, but with the Safety gate monitoring function block this Deactivated status can be staggered. It is not possible to switch from Active to Discrepant and then back to Active.
- The synchronization time has either not yet elapsed or is fully deactivated.
- The synchronization status has been set to Disabled at least once since the Flexi Soft system switched from the Stop status to the Run status. Consequently, if the input signals are already present for the Active status when the system transitions to the Run status, the **Release** output still remains set to 0.

A synchronization error (timeout) is reset when the Active synchronization status is achieved, i.e., when the **Release** output switches to 1.

NOTE

The following must be taken into account when defining values for the synchronization time:

- The synchronization time must exceed the logic execution time.
- The synchronization time has an accuracy of ± 10 ms plus the logic execution time. The logic execution time is dependent on the type and number of function blocks used. It is displayed on the **FB group info** tab of the logic editor and also in the report.
- If signals from tested sensors are connected to FX3-XTIO or FX3-XTDI modules, the synchronization time should be at least as long as the set **test gap (ms)** plus the **max. OFF-ON delay (ms)**, as a signal change at the input of the modules may be delayed by this amount of time. Both values of the test output used are displayed in the report.

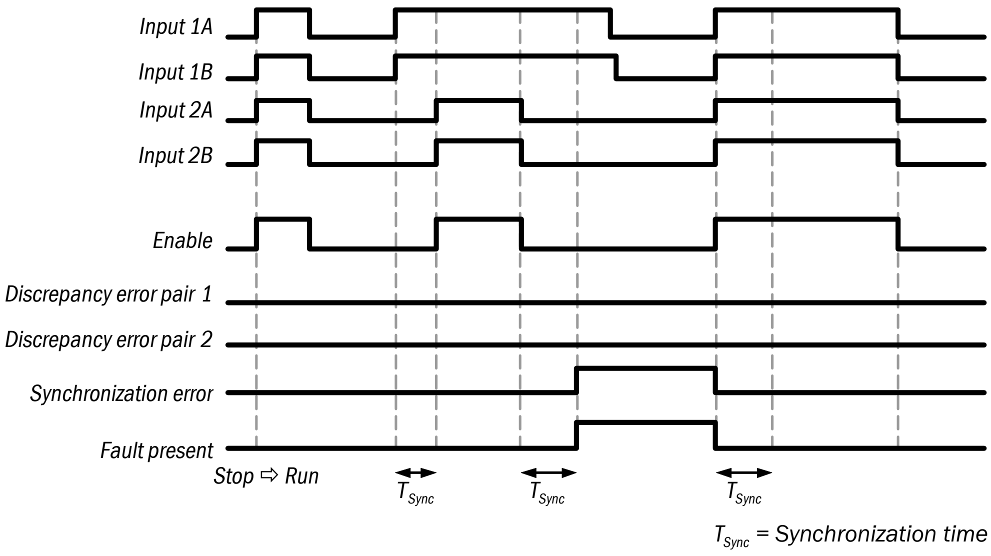


Figure 130: Sequence/timing diagram for the Safety gate monitoring function block, category 4, double dual-channel without function test – synchronization monitoring

7.8.4 Emergency stop

Function block diagram

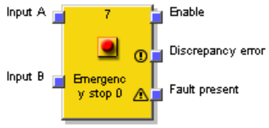


Figure 131: Inputs and outputs of the Emergency stop function block

General description

The Emergency stop function block can be used to implement an Emergency stop function with an emergency stop pushbutton.

If a corresponding dual-channel input element is configured in the hardware configuration, this function block is no longer required in the logic because, in this case, preliminary evaluation takes place directly on the expansion module (e.g., the FX3-XTIO or FX3-XTDI module). However, this function block can be used if the **Fault present** output is required for further processing. This involves configuring both input signals as single-channel signals and connecting them to the inputs of the function block.

In the case of emergency stop pushbuttons, a Reset and/or Restart function block is required to handle the processing of the reset/restart conditions for the safety chain if the **Release** output switches to 0. This may also be necessary in the case of emergency stop pushbuttons with a combined push/pull release mechanism.

Function block parameters

Table 85: Parameters of the Emergency stop function block

Parameter	Possible values
Inputs	• Single-channel • Dual-channel equivalent • Dual-channel complementary
Discrepancy time	0 = infinite, 10 … 30,000 ms in 10 ms increments. If the value is anything other than 0, it must be greater than the logic execution time.
Number of outputs	• 1 (Enable output) • 2 (Enable output and Discrepancy error output)
Use Fault present	• With • Without

Additional information about the behavior of this function block: see "Dual-channel evaluation (1 pair) and discrepancy time", page 161.

7.8.5 Magnetic switch

Function block diagram

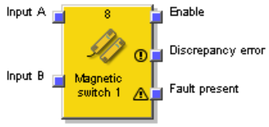


Figure 132: Inputs and outputs of the Magnetic switch function block

General description

The internal logic of the Magnetic switch function block works in exactly the same way as for the Emergency stop function block. The only difference is that the choice of parameters is more restricted. This function block is used to differentiate graphically between the various elements in the logic program.

The Magnetic switch function block is a predefined function block for reed switches or other sensors that call for discrepancy time monitoring. If the evaluation of the inputs is 1, the **Release** output is 1.

Function block parameters

Table 86: Parameters of the Magnetic switch function block

Parameter	Possible values
Inputs	• Dual-channel equivalent • Dual-channel complementary
Discrepancy time	10 … 3.000 ms in 10 ms increments. The value must be greater than the logic execution time.
Number of outputs	• 1 (Enable output) • 2 (Enable output and Discrepancy error output)
Use Fault present	• With • Without

Additional information about the behavior of this function block: see "Dual-channel evaluation (1 pair) and discrepancy time", page 161.

7.8.6 Light curtain monitoring

Function block diagram

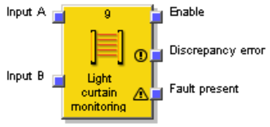


Figure 133: Inputs and outputs of the Light grid monitoring function block

General description

The Light curtain monitoring function block can be used to implement a semiconductor protective device function with an electro-sensitive protective device.

The internal logic of the Light grid monitoring function block works in exactly the same way as for the Emergency stop function block. The only difference is that the choice of parameters is more restricted. It is not possible to select the single-channel input type for the Light grid monitoring function block. If the evaluation of the inputs is 1, the **Release** output is 1.

NOTE

If a corresponding dual-channel input element is configured in the hardware configuration, this function block is no longer required in the logic because, in this case, preliminary evaluation takes place directly on the expansion module (e.g., the FX3-XTIO or FX3-XTDI module). However, this function block can be used if the **Fault present** output is required for further processing. This involves configuring both input signals as single-channel signals and connecting them to the inputs of the function block.

Function block parameters

Table 87: Parameters of the Light curtain monitoring function block

Parameter	Possible values
Input type	Dual-channel equivalent
Discrepancy time	0 = infinite, 10 … 500 ms in 10 ms increments. If the value is anything other than 0, it must be greater than the logic execution time.
Number of outputs	• 1 (Enable output) • 2 (Enable output and Discrepancy error output)
Use Fault present	• With • Without

Additional information about the behavior of this function block: see "Dual-channel evaluation (1 pair) and discrepancy time", page 161.

7.8.7 Safety gate monitoring

Function block diagram

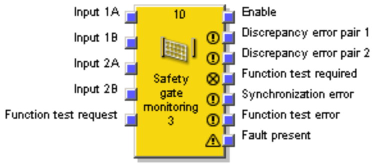


Figure 134: Inputs and outputs of the Safety gate monitoring function block

General description

This function block can be used to evaluate dual-channel switches (on safety gates). You can choose between one pair or two pairs. Information on the behavior of dual-channel evaluation: see "Dual-channel evaluation (1 pair) and discrepancy time", page 161 and see "Double dual-channel evaluation and synchronization time", page 163.

In addition, the function block can be optionally used for function test monitoring.

Important information

NOTICE

If both inputs change from 0 → 1 without discrepancy and exactly one logic execution time was = 0 beforehand, the **Enable** output is only set to 1 every second time.

Function block parameters

Table 88: Parameters of the Safety gate monitoring function block

Parameter	Possible values
Inputs	• Single-channel • Dual-channel equivalent (1 pair) • Dual-channel complementary (1 pair) • Dual-channel equivalent (2 pairs) • Dual-channel complementary (2 pairs)
Function test	• No function test • Function test required

Parameter	Possible values
Discrepancy time pair 1 Discrepancy time pair 2	Can be set separately for inputs 1A / 1B and 2A / 2B. 0 = infinite, 10 … 30,000 ms in 10 ms increments. If the value is not 0, it must be greater than the logic execution time.
Synchronization time	0 = infinite, 10 … 30,000 ms in 10 ms increments. If the value is not 0, it must be greater than the logic execution time.
Number of outputs	1 … 6
Use Fault present	• With • Without

Function test

In some applications, protective devices have to undergo a cyclic physical check (function test).

If the Safety gate monitoring function block is configured with the **Function test required** parameter, the input signal must change once per machine cycle so that the release condition stops being met and then it must switch back again (e.g., as a result of a safety door being opened and closed).

The **Function test request** input is typically connected to the machine cycle contact.

If a function test is required by the configuration, it must be performed in the following cases:

- After the Flexi Soft system switches from the Stop status to the Run status
- After each rising signal edge (0–1) at the **Function test request** input.

This is indicated by the **Function test required** output switching to 1. The **Function test required** output returns to 0 if the following happens before the next rising signal edge occurs at the **Function test request** input: A signal sequence is detected at the inputs causing the **Release** output to switch from 0 to 1.

The **Function test error** output switches to 1 if the next machine cycle begins before a function test has been performed, i.e., if the **Function test required** output is still set to 1 and another rising signal edge (0–1) occurs at the **Function test request** input. The **Release** output is 0 while a function test error has occurred.

The **Function test error** output returns to 0, if a signal sequence is detected that causes the **Release** output to switch from 0 to 1.

Sequence/timing diagram

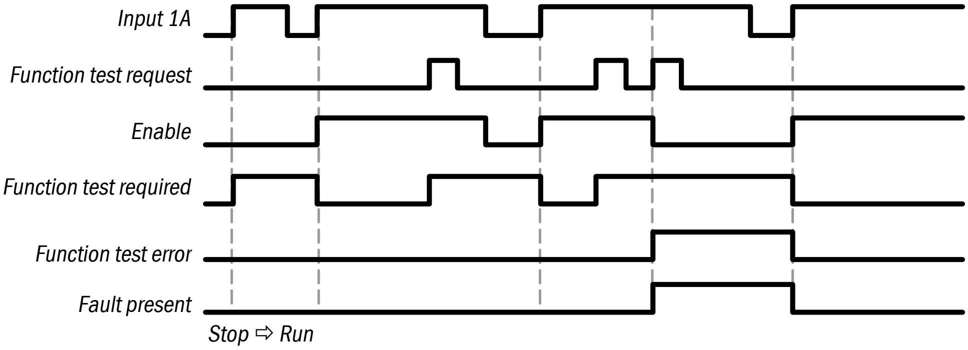
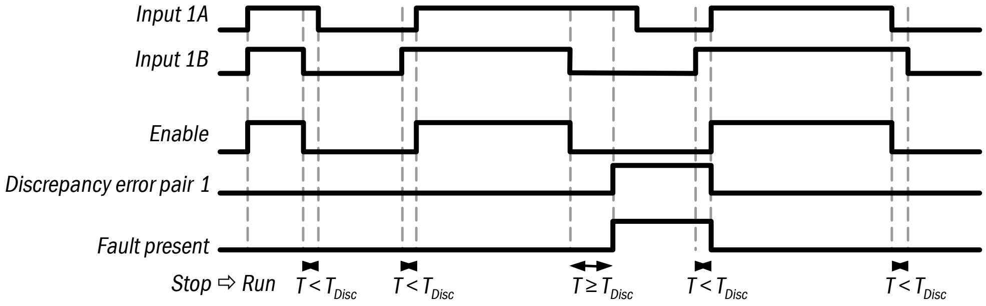


Figure 135: Sequence/timing diagram for the Safety gate monitoring function block, category 2, single-channel with function test



T_{Disc} = Discrepancy time

Figure 136: Sequence/timing diagram for the Safety gate monitoring function block, category 4, dual-channel without function test

7.8.8 Tolerant dual-channel evaluation

Overview

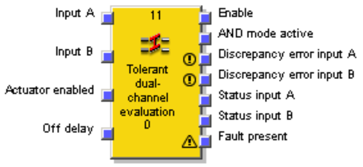


Figure 137: Inputs and outputs of the Tolerant dual-channel evaluation function block

The Tolerant dual-channel evaluation function block can be used to evaluate dual-channel switches or sensors. It also offers a form of dual-channel evaluation that is less restrictive than the standard type that is available with expansion modules such as the FX3-XTIO or FX3-XTDI, or with the Safety gate monitoring, Emergency stop, Light grid monitoring, Magnetic switch, Two-hand control type IIIA, and Two-hand control type IIIC function blocks.

Tolerant dual-channel evaluation checks whether the sequence of the two input signals is correct. If either of the two signals has triggered a switch-off, the other signal is expected to follow accordingly.

Tolerant dual-channel evaluation differs from standard dual-channel evaluation in the following respects:

- Fulfillment of the switch-off condition at the two inputs can be staggered. The switch-off condition does not have to be simultaneously fulfilled at both inputs on one occasion.
- An AND mode can be optionally activated to make the evaluation even more tolerant under certain conditions. In this case, deactivation of a single input is accepted as a correct sequence without the other input having to follow. This may be acceptable, as the dangerous machine parts (actuators) will have been shut down safely by this time. For this purpose, the optional **Actuator enabled** input must be connected to the signal within the logic that is responsible for controlling the output for the safety-related release of the actuator. If required, a time limit can be applied to restrict the duration of the AND mode.
- An optional switch-off delay can be applied to ignore a situation whereby one or both inputs are deactivated temporarily. It must be enabled by the **switch-off delay** input.
- Discrepancy time monitoring can be activated separately for switching on and switching off.

Function block parameters

Table 89: Parameters of the Tolerant dual-channel evaluation function block

Parameter Parameter	Possible values
Input mode	• Equivalent • Complementary
Evaluation mode	• Dual channel • Dual-channel / AND mode
Max. time for AND mode	0 = infinite, 1 to 60,000 s, adjustable in 1 s increments
Discrepancy time monitoring when switching on	• Disabled • Active
Discrepancy time monitoring when switching off	• Disabled • Active
Discrepancy time	0 = infinite, 10 ms to 60 s in 10 ms increments If the value is anything other than 0, it must be greater than the logic execution time.
Off delay time	0 to 10 s in 10 ms increments If the value is anything other than 0, it must be greater than the logic execution time.
Use input for switch-off delay	• Disabled • Active
Use outputs status input A and status input B	• Disabled • Active
Use Fault present	• Without • With

Dual-channel evaluation

The selected **Input mode** determines what values the two signals must have in order for the desired status to be achieved. There are two options:

- Equivalent evaluation
- Complementary evaluation

Table 90: Status of tolerant dual-channel evaluation according to input mode

Input mode	Input A	Input B	Status of tolerant dual-channel evaluation
Equivalent	0	0	Deactivated
	0	1	Discrepant, input A switched off
	1	0	Discrepant, input B switched off
	1	1	Active, if correct sequence has been observed
Complementary	0	1	Deactivated
	0	0	Discrepant, input A switched off
	1	1	Discrepant, input B switched off
	1	0	Active, if correct sequence has been observed

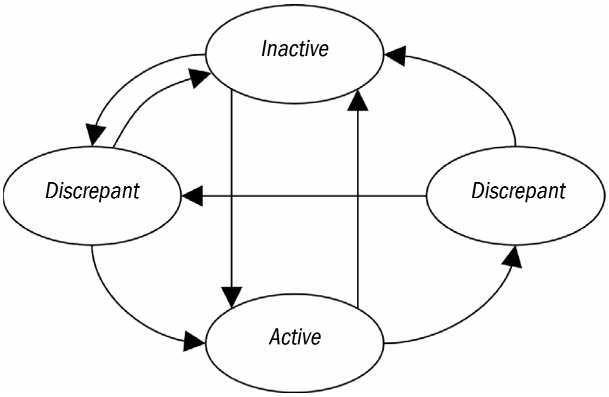


Figure 138: State diagram for the Tolerant dual-channel evaluation function block

Discrepancy time

An optional discrepancy time can be defined. The discrepancy time defines how long the two inputs can continue to have discrepant values after a change in either of the input signals without this being regarded as an error. The **Discrepancy error input A** and **Discrepancy error input B** outputs are used to indicate which input did not follow within the anticipated period of time.

A discrepancy error (timeout) is reset when the Active status is achieved, i.e., when a correct sequence has been observed so that the **Release** output switches to 1.

Sequence/timing diagrams

Tolerant dual-channel evaluation can only switch to Active (**Release** output changes from 0 to 1) if the following conditions are met:

- both inputs have switched off at least once since the last Active status and
- The discrepancy time has not expired or discrepancy time monitoring for switching on is deactivated.

This means that it is not possible to switch from Active to Discrepant and then back to Active if only one input has switched off.

NOTE
The sequence/timing diagrams shown in this section relate to the equivalent input mode. For the complementary input mode, input B is to be regarded as inverted.

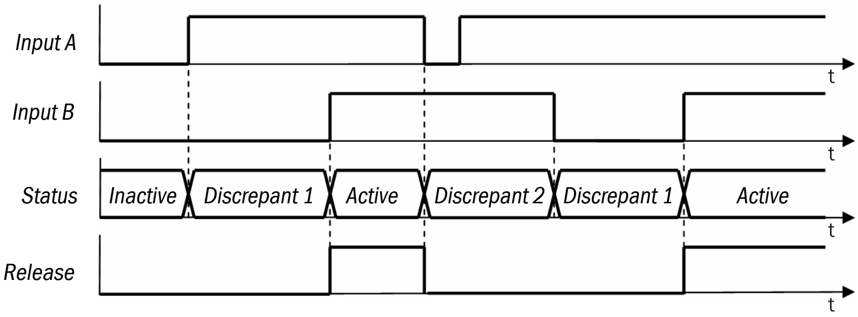


Figure 139: Sequence/timing diagram for the Tolerant dual-channel evaluation function block – switch to Active

Fault present and Error reset

The **Fault present** output switches to 1 if one of the following scenarios occurs:

- The discrepancy time for switching on has been activated and has expired.
- The discrepancy time for switching off has been activated and has expired.

All error statuses and error outputs (**Discrepancy error input A**, **Discrepancy error input B**, **Error present**) are reset following a successful switch to the Active status (**Release** output switches from 0 to 1). For this to happen, both inputs must have switched off beforehand, but not necessarily simultaneously.

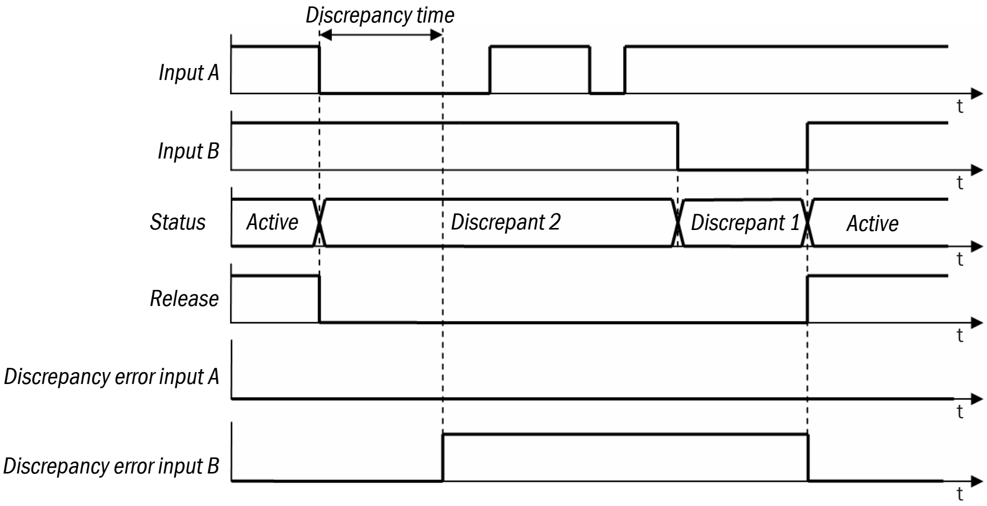


Figure 140: Sequence/timing diagram for the Tolerant dual-channel evaluation function block – Error reset

AND mode

If the **Evaluation mode** is set to **Dual-channel/AND mode**, there are two possibilities depending on the **Actuator enabled** input: Either both inputs are monitored in accordance with the rules of tolerant dual-channel evaluation or they are simply linked via a logical AND operation.

If AND mode is active, only one input has to switch off and on again for the status to switch back to Active; it is not necessary for the other input to switch as well. Whenever one or both of the inputs switch off, the **Enable** output is always forced to switch off as well. In AND mode, the value for switch-off via **Input B** is still dependent on the input mode.

AND mode is activated when a falling signal edge (1–0) occurs at the **Actuator enabled** input and the **Release** output is 1.

AND mode is deactivated again when the **Actuator enabled** input is set to 1 or when the **max. time for AND mode** expires. Expiry of the **max. time for AND mode** has no effect on the **Fault present** output.

No discrepancy time monitoring is performed in AND mode.

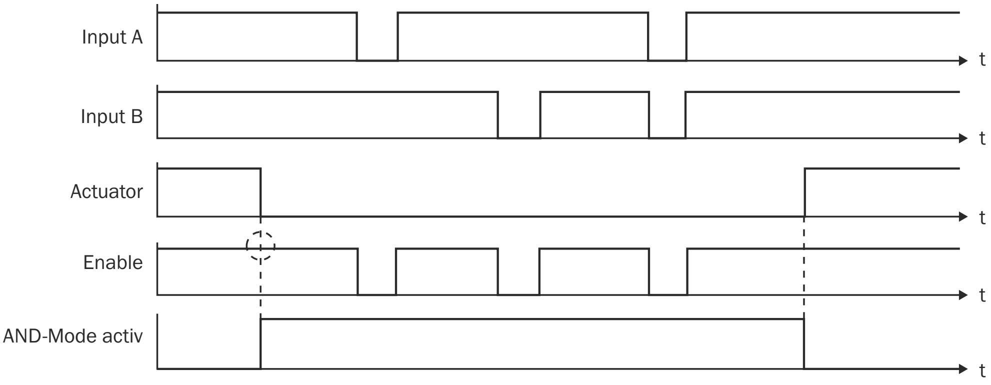


Figure 141: Sequence/timing diagram for the Tolerant dual-channel evaluation function block – AND mode

Off delay

The switch-off delay can be applied to ignore temporary deactivation of one or both inputs while keeping the **Release** output set to 1. If one or both inputs are still switched off at the end of the off delay time, the **Release** output switches to 0.

The switch-off delay is only effective when the **Switch-off delay** input is set to 1. If the **Switch-off delay** input is 0 and one or both of the inputs switch off, the effect is immediate.

The switch-off delay is effective in dual-channel mode as well as in AND mode.

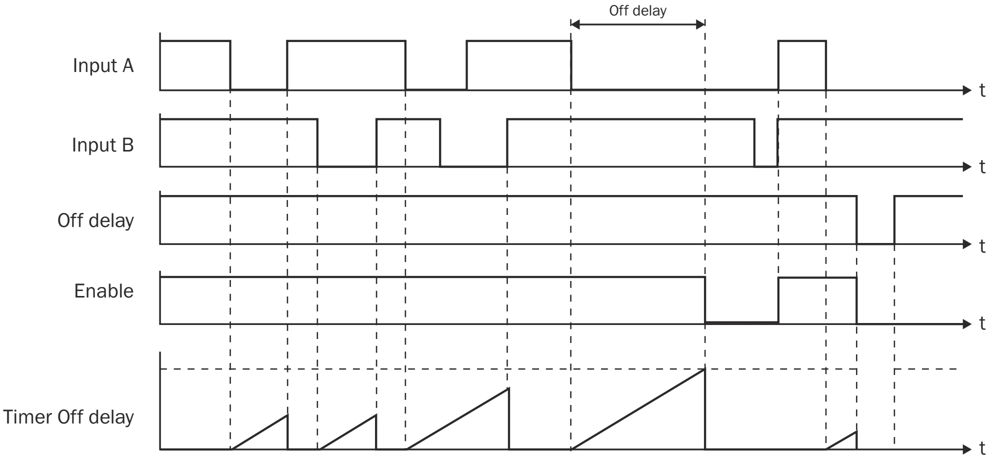


Figure 142: Sequence/timing diagram for the Tolerant dual-channel evaluation function block – Off delay

Status input A/B

The two outputs **Status input A** and **Status input B** show the internal values of inputs A and B. These are the same as the **Input A** and **Input B** values apart from the following exceptions:

- The status output shows the “Off” value even though the associated input is switched on (with input mode = Equivalent: 0 instead of 1), as the other input still has to switch off first before switch-on becomes possible again (**Release** output switches to 1).
- The status output shows the “On” value even though the associated input is switched off (with input mode = Equivalent: 1 instead of 0). This is because the switch-off delay is active and switch-off is currently being prevented internally.

Further topics

- "Dual-channel evaluation (1 pair) and discrepancy time", page 161

7.8.9 Two hand control type IIIA

Function block diagram

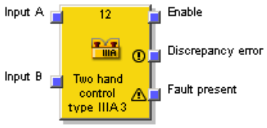


Figure 143: Inputs and outputs of the Two hand control type IIIA function block

General description

The Two hand control type IIIA function block is a predefined function block for two-hand control devices, where discrepancy time monitoring is required for equivalent inputs. Discrepancy time monitoring enables synchronous actuation monitoring for type IIIA two-hand control devices according to ISO 13851.

The internal logic of the Two hand control type IIIA function block works in exactly the same way as for the Emergency stop function block. The only difference is that the choice of parameters is more restricted. This function block enables a distinction to be made graphically in accordance with the application concerned.

Together, **Input A** and **Input B** form a dual-channel evaluation function and must be equivalent. If the evaluation of the inputs is 1, the **Release** output is 1 (see "Dual-channel evaluation (1 pair) and discrepancy time", page 161).

Function block parameters

Table 91: Parameters of the Two hand control type IIIA function block

Parameter	Possible values
Inputs	Fixed value: dual-channel equivalent
Discrepancy time	For main module firmware version up to V4.04.0: Fixed value: 500 ms For main module firmware version V4.05.0 and above: 10 ms … 500 ms in 10 ms increments. The value must be greater than the logic execution time. (Corresponds to synchronization time according to ISO 13851)
Number of outputs	• 1 (Enable output) • 2 (Enable output and Discrepancy error output)
Use Fault present	• With • Without

7.8.10 Two hand control type IIIC

Function block diagram

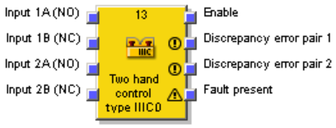


Figure 144: Inputs and outputs of the Two hand control type IIIC function block

General description

The Two hand control type IIIC function block provides the logic that is required to monitor the inputs of a two-hand control device in accordance with ISO 13851.

The inputs used must be configured as single-channel signals in the hardware configuration, i.e., no dual-channel input evaluation on the expansion module.

Function block parameters

Table 92: Parameters of the Two hand control type IIIC function block

Parameter	Possible values
Discrepancy time (pair 1)	0 = infinite, 10 ms … 500 ms in 10 ms increments. If the value is not 0, it must be greater than the logic execution time.
Discrepancy time (pair 2)	0 = infinite, 10 ms … 500 ms in 10 ms increments. If the value is not 0, it must be greater than the logic execution time.

Parameter	Possible values
Synchronization time	For main module firmware version up to V4.04.0: Fixed value: 500 ms For main module firmware version V4.05.0 and above: 0 = 500 ms, 10 ms … 500 ms in 10 ms increments. The value must be greater than the logic execution time.
Number of outputs	• 1 (Enable output) • 2 (Enable output and Discrepancy error pair 1 output) • 3 (Enable output, Discrepancy error pair 1 output, and Discrepancy error pair 2 output)
Use Fault present	• With • Without

The function block evaluates its input signals in pairs. Together, **Input 1A** and **Input 1B** form one dual-channel evaluation function and they have to be complementary. Together, **Input 2A** and **Input 2B** form one dual-channel evaluation function and they also have to be complementary. A discrepancy time can be specified for each pair of inputs.

The synchronization time is the amount of time for which the input pairs are allowed to have different values. As stipulated in standards and regulations, the synchronization time for evaluating the two-hand control devices must not exceed 500 ms. That means the synchronization time can be edited in a range between 10 ms and 500 ms.

Information on the behavior of double dual-channel evaluation: see "Dual-channel evaluation (1 pair) and discrepancy time", page 161 and see "Double dual-channel evaluation and synchronization time", page 163.

The Two-hand control type IIIC function block uses a different type of synchronization evaluation from the Safety gate monitoring function block in terms of what condition must be met for the Disabled synchronization status. With the Two-hand control type IIIC function block, both dual-channel evaluation functions must be set to Disabled, i.e., the A inputs must be 0 and the B inputs must be 1 for both pairs of inputs at the same time.

Furthermore, there is no **Synchronization error** output for the Two-hand control type IIIC function block. This is because a failure to actuate both hand switches of a two-hand control device within the fixed time of 500 ms is not classed as an error. However, this synchronization time must not be exceeded, because otherwise the **Release** output will not switch to 1.

Sequence/timing diagram

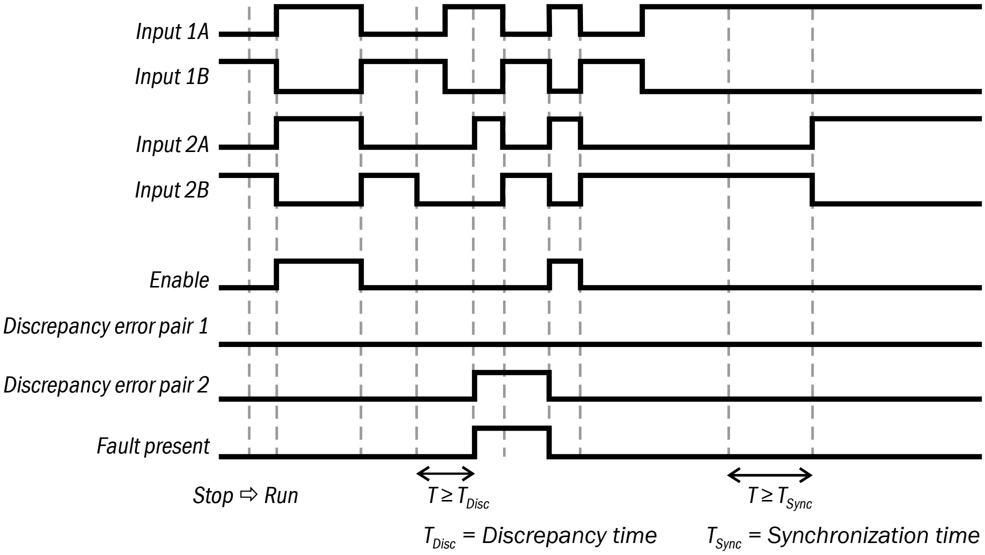


Figure 145: Sequence/timing diagram for the Two hand control type IIIC function block

7.8.11 Multi operator

Function block diagram

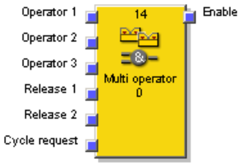


Figure 146: Inputs and outputs of the Multi operator function block

General description

The Multi operator function block makes it possible to monitor the simultaneous operation of up to three two-hand control systems. For example, in a press application where there is more than one operator, it may be necessary to have multiple two-hand control systems or foot switches so that the downward movement of the press can be triggered in unison. Each **Operator** input is usually connected to the **Enable** output of a Two hand control function block.

In additional, two optional **Release** inputs (e.g., safety light curtains) can be connected to make sure that the assigned devices are 1 before the **Release** output can switch to 1. The Reset and Restart functions must be dealt with independently of this function block.

The **Cycle request** input can be used to impose a requirement so that the operators have to let go of each two-hand control system at least once before a restart is possible. Typically, this input is connected to a signal that generates a pulse during each machine cycle. In this way, you can prevent one or more of the two-hand control systems from remaining permanently actuated.

NOTE

- Never connect anything other than safe signals that have undergone preliminary evaluation to the **Operator** inputs, e.g.,the **Release** output of a Two-hand control type IIIA or Two-hand control type IIIC function block. There are two ways to perform safety-related evaluation on the inputs of a two-hand control device. They can either be evaluated by another function block (e.g., Two-hand control device or Light curtain monitoring) or evaluation can be built into how the safety capable inputs are configured (e.g., by configuring the inputs with dual-channel evaluation).
- The **Cycle request** input must not be used for safety functions. This input is intended exclusively for automation control.

WARNING

Incorrect configuration or incorrect use of the inputs

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

- ► Only connect signals that have undergone preliminary evaluation to the Operator and Release inputs.
- ► Only use suitable inputs for safety functions.

Function block parameters

Table 93: Parameters of the Multi operator function block

Parameter	Possible values
Cycle request condition	• Rising signal edge • Falling signal edge
Number of operators	• 2 operators • 3 operators
Number of release inputs	• 0 • 1 • 2

The Release output switches to 1 if the following conditions are met:

- All **Release** inputs are and remain set to 1.
- Each activated **Operator** input has been set to 0 at least once (can also be staggered) since the Flexi Soft system switched from the Stop status to the Run status.
  This second condition does not need to be fulfilled if one of the three following cases applies:
  a) The **Release** output was never set to 1 after the Flexi Soft system switched from the Stop status to the Run status.
  b) A rising or falling signal edge (depending on the configuration) has been detected at the **Cycle requirement** input.
  c) One or more **Release** inputs were previously set to 0.
- All activated **Operator** inputs have subsequently switched to 1.

The Release output is 0 if one of the following conditions is met:

- One or more **Release** inputs are 0.
- One or more **Operator** inputs are 0.
- A rising or falling signal edge (depending on the configuration) has been detected at the **Cycle request** input.

Sequence/timing diagram

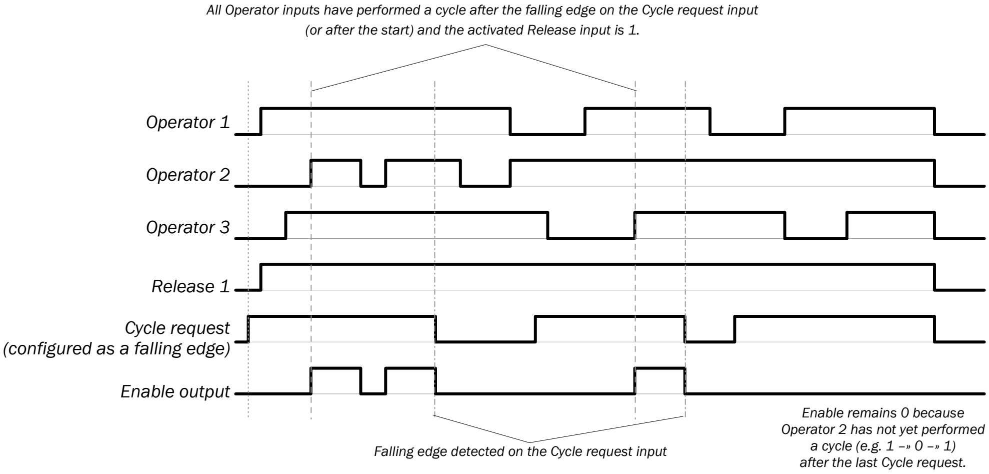


Figure 147: Sequence/timing diagram for the Multi operator function block

7.9 Parallel muting, Sequential muting, and Cross muting function blocks

7.9.1 Overview and general description

Muting is an automated process that temporarily bypasses safety functions of a control system or protective equipment. Muting allows certain objects (e.g., pallets loaded with material) to pass through electro-sensitive protective equipment (ESPE) such as a safety light curtain and into a hazardous area. During this transport operation, the Muting function bypasses monitoring by the electro-sensitive protective equipment.

Three different function blocks are available for muting:

- Parallel muting

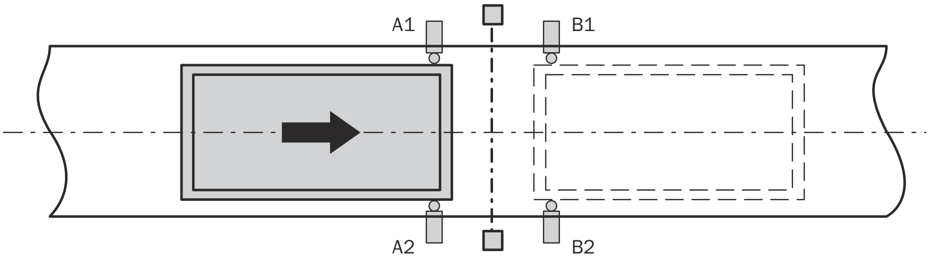


Figure 148: Muting with two sensor pairs arranged in parallel (A1 / A2 and B1 / B2)

- Sequential muting

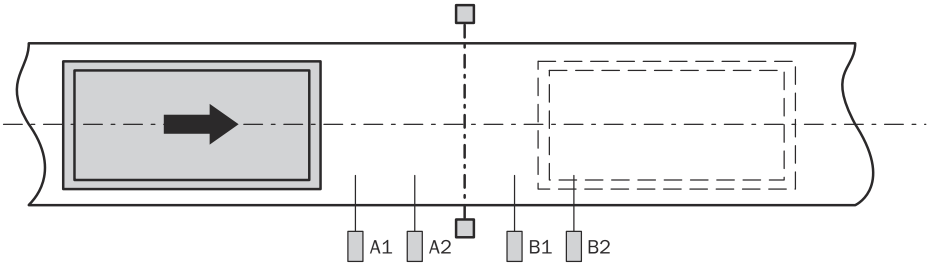


Figure 149: Muting with two sensor pairs arranged in sequence (A1 / A2 and B1 / B2)

- Cross muting

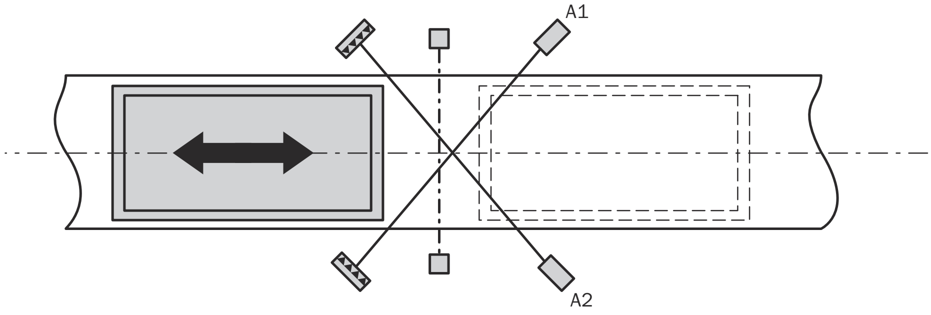


Figure 150: Muting with a sensor pair arranged crosswise (A1 / A2)

Muting sensors

Muting sensors monitor the presence of the material while it is being transported. Careful selection of the sensor type and how the sensors are arranged makes it possible to distinguish between objects and people.

In conjunction with the muting sensors and the electro-sensitive protective device, the object that is being transported generates a precisely defined signal sequence while it is traveling into the hazardous area. The muting sensors must ensure that all dangers are eliminated in the event of someone entering the area protected by the electro-sensitive protective device (i.e., a dangerous state must be terminated immediately). It is absolutely essential to ensure that a person cannot generate the same signal sequence as a transported object.

The placement of the muting sensors is determined by the shape of the object being detected. There are various options involving different numbers of sensor input signals. These include the following:

- Two sensors
- Two sensors and one additional C1 signal
- Four sensors (two pairs of sensors)
- Fours sensors (two pairs of sensors) and one additional C1 signal

Muting signals can be generated by the following sources:

- Optical sensors
- Inductive sensors
- Mechanical switches
- Controller signals

If optical sensors are used for muting applications, choose sensors with background suppression to ensure that only the material being transported fulfills the muting conditions. These sensors are only capable of detecting material up to a certain distance. Consequently, the input conditions for the muting sensors cannot be met by objects that are located any further away than this. This applies in particular to sequential muting.

Conditions for muting

While the muting status is active, the **Release** output remains at 1, even if the **Electro-sensitive protective equipment** input switches to 0.

Depending on the selected muting type and configuration, different conditions are tested for a correct muting cycle, i.e., the correct initiation, maintenance and termination of the muting status.

In general, at least one muting sensor signal pair (**A1** / **A2** or **B1** / **B2**) must always be active to maintain the muting status.

You can achieve a higher level of safety and improved protection against manipulation using the following functions:

Table 94: Monitoring functions for muting

Monitoring	Parallel muting	Sequential muting	Cross muting	Additional information
Sequence monitoring	–	✓	–	"Sequence monitoring", page 186
Direction detection	✓	✓	–	"Direction detection", page 186
Optional C1 input	✓	✓	✓	"Input C1", page 187
Concurrence monitoring	✓	✓	✓	"Concurrence monitoring time", page 185
Monitoring of the total muting time	✓	✓	✓	"Total muting time", page 185
End of muting by ESPE	✓	✓	✓	"End-of-muting condition", page 185

7.9.2 Safety notes for muting applications

The safety functions of a protective device are bypassed by muting.

WARNING

Restricted safety through muting

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

- ► Observe the general safety specifications and protective measures.
- ► You must always observe the following notes about how to use the muting function correctly.

General safety notes for muting

- ► Always observe the applicable national, regional and local regulations and standards.
- ► Make sure that the application is in line with an appropriate risk analysis and risk avoidance strategy.
- ► Muting must be setup to be carried out automatically but not be dependent on a single electrical signal.
- ► Never use muting to transport a person into the hazardous area.
- ► Make sure that muting is only activated for as long as access to the hazardous area remains blocked by the object responsible for triggering the muting condition.
- ► Make sure that the muting condition is terminated as soon as the object has finished passing through so that the protective device returns to its standard non-bypassed status (i.e., it must be reactivated).
- ► In the case of long muting cycles (i.e., those lasting more than 24 hours) or long machine downtimes, check the muting sensors to make sure they are functioning correctly.
- ► If the total muting time is set to infinite (inactive), use additional measures to prevent anyone from entering the hazardous area while muting is active.
- ► If safety-related information (i.e., remote safety capable input values and/or remote safety output values) is transmitted via a safety fieldbus network, always take the associated delay times into account. These delay times may influence both the system behavior and the minimum safety distance requirements associated with the response times.

Safety notes for the electro-sensitive protective equipment (ESPE)

- ► Access to the hazardous area must be reliably detected by the ESPE or other measures must be taken to prevent a person from bypassing, exceeding, crawling under or crossing the ESPE undetected.
- ► Observe the operating instructions for the electro-sensitive protective device that explain how to install and use the device correctly.
- ► Secure the area between the electro-sensitive protective device and the muting sensors as follows to prevent anyone standing behind:
  - ○ With parallel muting – between the electro-sensitive protective device and sensors A1 / A2 as well as between the electro-sensitive protective device and sensors B1 / B2 (see figure 156, page 194).
  - ○ With sequential muting – between the electro-sensitive protective device and sensor A2 as well as between the electro-sensitive protective device and sensor B1 (see figure 159, page 196).
  - ○ With cross muting – between the electro-sensitive protective equipment and sensor A1 as well as between the electro-sensitive protective equipment and sensor A2 (see figure 162, page 198).

Safety notes for the muting sensors

- ► Set up muting so that it is triggered by at least two signals (e.g., from muting sensors) that are wired independently of one another and it is not fully dependent on software signals (e.g., from a PLC).
- ► Arrange the muting sensors so that if an intervention in the protective field occurs, the hazardous area can only be reached once the dangerous state has been eliminated. A condition for this is that the necessary minimum distances between the ESPE and the hazardous area are maintained, typically in accordance with EN ISO 13855.
- ► Arrange the muting sensors so that material can pass unhindered but so no one can enter the hazardous area by fulfilling the muting conditions themselves (i.e., by activating both muting sensors and thereby meeting the muting requirements).

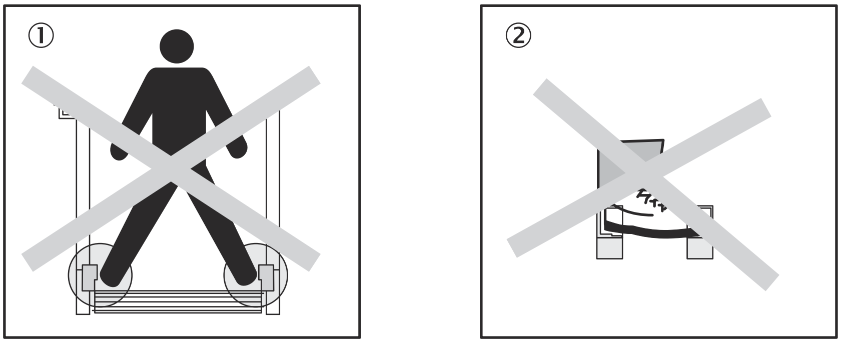


Figure 151: Safety requirements when mounting the muting sensors

① It must not be possible to activate sensors that are located opposite one another at the same time.

② It must not be possible to activate sensors that are located next to one another at the same time.

- ► Arrange the muting sensors so that only the moving material is detected, and not the transportation equipment (pallet or vehicle).

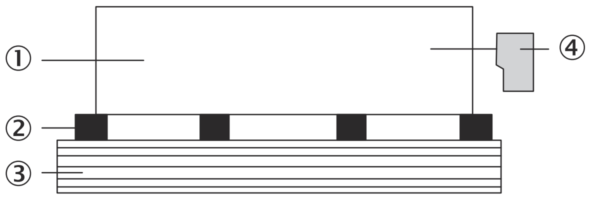


Figure 152: Detection of material during muting

①	Transported material
②	Transportation equipment
③	Transport level
④	Muting sensor

- ► Set up the muting so that the material to be transported is detected over the entire stretch. The output signal must not be interrupted (see "Suppression of sensor signal gaps", page 186).
- ► Arrange the muting sensors so that a minimum distance is observed in relation to the detection zone of the electro-sensitive protective equipment (e.g., in relation to the light beams of a light curtain) whenever material is detected. The minimum distance ensures the required processing time until muting is activated.
- ► For wiring the muting sensors, see section titled "Notes on wiring", page 192.

Safety notes for override

- ► Mount the control switches for the Override functions outside of the hazardous area so that they cannot be actuated by anyone who is located inside the hazardous area. In addition, the operator must have a complete overview of the hazardous area when actuating a control switch.
- ► For wiring the signal for Override, see the section titled "Notes on wiring", page 192 .
- ► Before activating the Override function, make sure that the equipment is in perfect working order, particularly the muting sensors (visual inspection).
- ► Make sure that the hazardous area is clear of people both before the Override function is activated and while it is active.
- ► If you have had to activate the Override function, check the functionality of the equipment and the arrangement of the muting sensors after the event.

Safety notes for the muting/override lamp

- ► Use a muting and/or override lamp to signal that the Muting or Override functions are active. You can either use an external muting/override lamp or one that is integrated into the electro-sensitive protective device (ESPE).
- ► Always attach the muting and/or override lamps so that they are clearly visible. The muting/override lamp must be visible from every side all the way around the hazardous area and must be clearly visible to the system operator.
- ► Depending on local, regional, and national regulations and standards, it may be necessary to monitor the muting/override lamp(s). If this is the case, implement additional measures for this purpose. The FX3-XTIO and FX3-XTDI modules do not support any form of lamp monitoring.

7.9.3 Inputs, outputs and parameters of function blocks

The following table shows which configuration parameters are possible with the various muting function blocks.

Table 95: Parameters of the muting function blocks

Parameter	Possible values
Direction detection	• Deactivated **Parallel muting and sequential muting only:** • Forward (A1/A2 first) • Backward (B1/B2 first)
Condition of other sensor pair for start of muting	• Both sensors clear **Parallel muting and sequential muting only:** • At least one sensor clear
End-of-muting condition	• With muting sensor pair • With electro-sensitive protective device (ESPE)
Total muting time	0 = infinite, 5 … 3,600 s, adjustable in 1 s increments
Concurrence monitoring time	0 = infinite, 10 … 3,000 ms, adjustable in 10 ms increments. If the value is not 0, it must be greater than the logic execution time.
Suppression of sensor signal gaps	0 = infinite, 10 … 1,000 ms, adjustable in 10 ms increments. If the value is not 0, it must be greater than the logic execution time.
Sequence monitoring	**Cannot be selected. Defined by the choice of muting function block:** • With sequential muting: Active • With parallel muting and cross muting: Deactivated
Additional muting time when ESPE is clear	0 ms, 200 ms, 500 ms, 1,000 ms
Input C1	• With • Without
Override input	• With • Without
Conveyor	• With • Without
Min. override pulse time	• 100 ms • 350 ms

Condition of other sensor pair for start of muting

NOTE

This parameter can only be changed for parallel muting and sequential muting. With cross muting, the parameter is always set to **Both sensors clear.**

The **Condition of other sensor pair for start of muting** parameter determines when the next valid muting sequence can begin after a previous muting sequence. The **Condition of other sensor pair for start of muting** can be defined as follows:

- **Both sensors clear:** All muting sensor signal inputs are 0 and the **Electro-sensitive protective equipment** input is 1 (i.e., the protective field is clear).

Or:

- **At least one sensor clear:** All muting sensor signal inputs except the last one are 0 and the **Electro-sensitive protective equipment** input is 1 (i.e., the protective field is clear).

If a higher throughput is required, it may be advisable to let the next muting sequence begin as soon as the transported material has traveled past the protective device and past all the muting sensors except the last one (i.e., at least one sensor is clear).

End-of-muting condition

The **Condition for end of muting** parameter determines when a valid muting status is over:

- **With muting sensor pair**: When a muting sensor signal input in the last muting sensor pair switches to 0 (sensor clear).

Or:

- **With ESPE**: When the **Electro-sensitive protective equipment** input switches to 1 and therefore indicates that the protective field is clear again.

If the **Electro-sensitive protective equipment** input switches to 0 at the end of muting (e.g., because the ESPE protective field has been breached) before the next valid muting sequence begins, the **Release** output of the function block switches to 0. In this case, the next muting cycle can only begin once the **End-of-muting condition** has been met.

Additional muting time when ESPE is clear

The **Additional muting time when ESPE is clear** parameter can be used when the **End-of-muting condition** parameter has been set to **With ESPE**. Sometimes, irregularities in the material or transportation equipment may mean that the ESPE cannot always detect the end of muting precisely. If this happens, you can increase the availability of the machine by configuring an additional muting time of up to 1,000 ms.

In this case (and in this case only), the **Additional muting time after the ESPE frees up** parameter determines the additional muting time once the **Electro-sensitive protective equipment** input has switched back to 1.

NOTE
If one of the muting sensors relevant for muting end frees up, the muting sequence is ended immediately, even if the **Additional muting time after the ESPE frees up** has not yet expired.

Total muting time

The **total muting time** is used to limit the maximum duration of the muting sequence. If the value set for the **total muting time** is exceeded, the **Muting error** and **Fault present** outputs change to 1 and the **Release** output switches to 0.

The timer for the **Total muting time** starts running when a valid start condition for muting exists; this is indicated by the **Muting status** output transitioning to 1. The timer for the **Total muting time** stops running and is reset to 0 if the muting sequence is ended again; this is indicated by the **Muting status** output transitioning to 0.

If the optional **Conveyor** input is used, the timer for the **Total muting time** pauses when the **Conveyor** input is set to 0, i.e., if a conveyor system stop has been detected.

Concurrence monitoring time

The **Concurrence monitoring time** is used to check whether the muting sensors are activated at the same time. This value relates to the two muting sensor signal inputs that are subject to dual-channel evaluation and specifies how long they are allowed to have different values without this being regarded as an error. This means that input pair **A1** / **A2** or input pair **B1** / **B2** must assume equivalent values before the end of the **Concurrence monitoring time**.

Concurrence monitoring starts as soon as a value of a muting sensor signal input changes for the first time. If the **Concurrence monitoring time** expires and both inputs of an input pair still have different values, an error occurs and the muting sequence is canceled.

If the Concurrence monitoring function of at least one input pair detects an error, the function block indicates this by setting the **Muting error** output to 1.

NOTE
With sequential muting, it must be taken into account that the two sensors of each pair switch at different times. The difference depends on the distance between the two sensors and on the speed of the material transport.

Suppression of sensor signal gaps

Occasionally, muting sensors are affected by output signal faults that are of no significance as far as muting is concerned. The **Suppression of sensor signal gaps** function makes it possible to filter out brief faults without interrupting muting.

When **Suppression of sensor signal gaps** is active, a change of a muting sensor signal input to 0 will be ignored for the length of time that has been set for **Suppression of sensor signal gaps**. The function block continues to interpret this as an uninterrupted 1 signal provided that only one muting sensor signal input from each sensor pair (**A1** / **A2** or **B1** / **B2**) is affected by a signal gap.

If a signal gap has already been detected on one muting sensor signal input of a sensor pair within the pair and then another signal gap occurs on the other muting sensor signal input of the same sensor pair at the same time, muting is terminated.

NOTE
To avoid machine downtimes during sequential muting, the configured time for **Suppression of sensor signal gaps** should be less than the length of time between deactivation of the first sensor and deactivation of the second sensor of a muting sensor pair (e.g., A1 / A2 or B1 / B2) when the transported material leaves the range of this sensor pair. Otherwise, the signal of the first sensor is still active at the time of deactivation of the second sensor due to the **Suppression of sensor signal gaps** and an error occurs in the sequence monitoring.

Direction detection

NOTE
Direction detection is only possible with parallel muting and sequential muting.

The **Direction detection** function can be used to tighten muting conditions if the material being transported is only to be moved in one particular direction. The possible movement direction depends on the order in which the muting sensors are activated.

If **Direction detection** is deactivated, the transported material can be moved in either direction to meet the muting conditions. In this case, it does not matter which sensor pair is activated first.

If the **Forward (A1 / A2 first)** direction is selected, the inputs for the muting sensor pairs must be activated in the order **A1** / **A2** before **B1** / **B2**. Muting is not possible in the opposite direction.

If the **Backward (B1 / B2 first)** direction is selected, the inputs for the muting sensor pairs must be activated in the order **B1** / **B2** before **A1** / **A2**. Muting is not possible in the opposite direction.

Sequence monitoring

NOTE
This parameter is only available in the case of sequential muting.

Sequence monitoring allows you to define a mandatory order in which the muting sensors have to be activated.

Table 96: Requirements for sequence monitoring according to direction detection setting

Direction detection	Valid sequence for muting sensor input signals
Deactivated	A1 before A2 before B1 before B2 or B2 before B1 before A2 before A1
Forward	A1 before A2 before B1 before B2
Backward	B2 before B1 before A2 before A1

Deviations from the sequence result in a muting error, which is indicated at the **Muting error** output. This applies both to the sequence of activation (muting sensor signal inputs switch from 0 to 1) and to deactivation (muting sensor signal inputs switch from 1 to 0).

Input C1

The optional **C1** input can be used as additional protection against manipulation. If it is used, the **C1** input must have switched to 0 after a previous muting cycle, and to 1 at the latest when both muting sensor signal inputs switch to 1 at the same time. A failure to meet this condition results in a muting error, which is indicated at the **Muting error** output.

The **C1** input must then switch back to 0 before the subsequent muting cycle is permitted. The **C1** input is not relevant for the duration of the muting status.

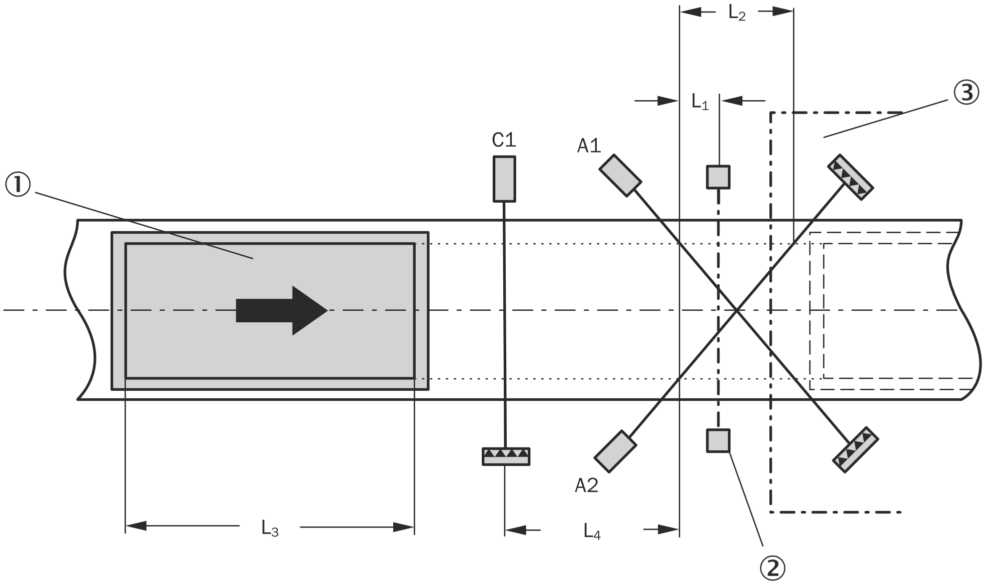


Figure 153: Example for the use of the optional C1 input with cross muting

① Transported material

② Electro-sensitive protective device (e.g., safety light curtain)

③ Hazardous area

In this example, the protection afforded by the protective equipment is bypassed when the sensors are actuated in a defined order. **C1** must be activated before the **two** muting sensor signal inputs in the first sensor pair (**A1** and **A2** in the example) switch to 1. This requires the length of the material in the conveying direction (L_3) to be greater than the distance between **C1** and the detection line of the muting sensors **A1** and **A2** (L_4).

Override input

An **Override** input signal allows you to remove transported objects that have been left stranded in the protective field of the protective device (e.g., safety light curtain) as a result of a power failure, an emergency stop, muting errors, or similar circumstances.

The **Override** function allows you to activate the **Release** output of the muting function block even though no valid muting sequence has been detected and the protective equipment (e.g., safety light curtain) is signaling that a dangerous state may exist. The **Override** input should only be used if the hazardous area has been visually inspected beforehand, there is no one within the hazardous area, and nobody will be able to access the hazardous area while the **Override** input is in use.

WARNING
Restricted safety with Override

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

► Only use override if the hazardous area has been visually inspected beforehand, there is no one within the hazardous area, and nobody will be able to access the hazardous area while the **Override** input is in use.

The Override status output (available with firmware ≥ V3.00.0) switches to 1 and the Override required output pulsates at 2 Hz if all of the following conditions are met:

- Muting is set to 0 (i.e., the **Muting status** output is 0).
- At least one of the muting sensor signal inputs **A1**, **A2**, **B1**, **B2** is set to 1.
- The **Electro-sensitive protective equipment** input is set to 0 (e.g., safety light curtain has been interrupted).
- The **Release** output is set to 0.

If the conditions for the **Override required** output are met and a valid override sequence involving a 0–1–0 transition (at least 100 ms or 350 ms but not exceeding 3 s; longer or shorter pulses will be ignored) occurs at the **Override** input, the **Release** output switches to 1 in exactly the same way as if the muting conditions had been met. Once all the muting sensor signal inputs have switched back to 0 and the **Electro-sensitive protective equipment** input is set to 1 (e.g., is indicating that the protective field of a safety light curtain is now clear), the next valid muting cycle is expected. If the next object does not meet the conditions for a muting cycle but does meet the conditions for the **Override required** output, then another override cycle can be used to remove the transported material. The number of override cycles is limited (see table 98, page 190).

NOTE
A reset pushbutton may also be suitable for the Override function.

Table 97: Conditions for Override required and when override is possible

Muting status	At least one of the muting sensor signal inputs A1, A2, B1, B2 is set to 1	Electro-sensitive protective equipment input	Override required output	Override possible
0	No	0	0	No
0	No	1	0	No
0	Yes	0	Pulsates (2 Hz)	Yes, unless the maximum permissible number of override cycles has been exceeded
0	Yes	1	0	No
1	No	0	0	No
1	No	1	0	No
1	Yes	0	0	No
1	Yes	1	0	No

Example sequence for **Override** and **Override required:**

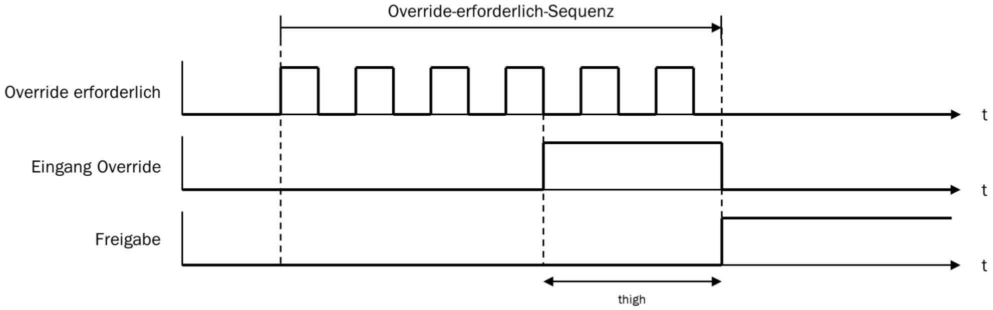


Figure 154: Sequence/timing diagram for Override and Override required

NOTE

t_{high} must be greater than or equal to the minimum override pulse time (100 ms or 350 ms), but less than or equal to 3 seconds. Otherwise the pulse on the **Override** input is ignored.

During an override cycle, the **Release** output is set to 1 in the same way as during a valid muting sequence. To prevent excessive use of the Override function, the number of permissible override cycles is limited. The number of permissible override cycles is dependent on the value for the **total muting time.**

Table 98: Number of permissible override cycles

Total muting time	Number of permissible override cycles	Comments
5 s	360	Maximum number of override cycles = 360
10 s	360	
20 s	180	= 60 min/total muting time
30 s	120	
1 min	60	
5 min	12	
15 min	5	Minimum number of override cycles = 5
30 min	5	
60 min	5	
Deactivated (0 = unlimited)	5	

The number of override cycles is saved in the function block. This value is incremented whenever the **Override required** output starts pulsating or whenever the **Override status** output switches to 1. The value is reset to 0 on completion of a valid muting cycle, after a system reset or after a transition from the Stop status to the Run status.

Once the **Override required** output has started pulsating at 2 Hz and a subsequent **Override** signal has switched to 1, muting begins again and the **Release** output changes to 1.

If the muting cycle is stopped because of a faulty muting sensor input signal, **Override required** switches to 1 for the duration of the logic execution time if the remaining conditions for **Override required** are met. If the faulty muting sensor signal input switches back to 1 first and then returns to 0, the muting cycle is once again stopped and **Override required** switches to 1 if the remaining conditions for **Override required** are met.

While there is a valid override status, none of the following are performed for the duration of one override cycle: direction detection, sequence monitoring (depending on function block) and concurrence monitoring.

Min. override pulse time

The **min. override pulse time** determines the minimum amount of time for which the **Override** input must remain set to 1 in order for the override signal to be valid.

Conveyor

If the transported material stops moving during the muting cycle, the total muting time and other parameters that can lead to muting errors could be exceeded. This problem can be avoided by using the **Conveyor** input. This input allows you to stop the time-dependent functions associated with muting if the material being transported comes to a halt.

- **Conveyor** input is 0: conveyor system stopped
- **Conveyor** input is 1: conveyor system running

The following timer functions are affected by the **Conveyor** input:

Table 99: Effect of the Conveyor input on timer functions

Monitoring function	Effect of Conveyor input
Monitoring of the total muting time	• The detection of a conveyor system stoppage pauses the timer functions. • When the conveyor system starts up again, the timer continues running with the value that was stored before the stoppage was detected. When this happens for the first time, a one-time increase of 5 seconds is added onto the total muting time.
Concurrence monitoring	

NOTE
The **Suppression of sensor signal gaps** is not affected by the **Conveyor** input.

Muting status output

The **Muting status** output indicates the status of the Muting function as per the table below:

Table 100: Muting status output values

Condition	Muting status output
Muting cycle inactive, no error or muting error detected	0
Muting cycle active, no error or override active, no error	1

Muting lamp output

The **Muting lamp** output can be used to indicate when a muting cycle is active. The value for the **Muting lamp** output is directly dependent on the values for **Muting status**, **Override status**, and **Override required**, as shown in the table below:

*Table 101: Output values for the **Muting lamp** output*

Status of the muting function block	Muting lamp output
Muting status output is set to 0	0
Muting status output is set to 1 or **Override status** output is set to 1	1
Override required output is set to 1	Pulsates at 2 Hz

Muting error output

The **Muting error** output indicates when an error associated with the muting function block has been detected. The **Muting error** output is set to 1 if the **Electro-sensitive protective equipment** input is set to 0 and any muting error has been detected and not yet reset.

The following muting errors are possible:

- Total muting time monitoring error
- Concurrence monitoring error
- Direction detection error
- Sequence monitoring error
- Error with transition from Stop status to Run status

NOTE
If the **Electro-sensitive protective equipment** input is set to 1, the display of muting errors at the **Muting error** output is suppressed.

To reset a muting error, it is necessary for the **Electro-sensitive protective equipment** input to be set to 1 and for all used muting sensor signal inputs to be set to 0. Alternatively, a muting error is reset with a valid override cycle.

Fault present output

The **Error flag** output has the same state as the **muting error**output.

Enable output

The Release output is 1 if one of the following conditions is met:

- The **Electro-sensitive protective equipment** input is set to 1 and no error/fault condition is active.
- A valid muting condition exists.
- A valid override cycle takes place.

In all other cases, the **Release** output is set to 0.

7.9.4 Notes on wiring

If muting functions are to be implemented, potential errors must be taken into account as part of the wiring process. If certain signal combinations are to be transmitted via the same cable, additional precautions must be taken to ensure that the respective signals are correct. Suitable measures must be implemented (e.g., protected cable laying) to make sure that no errors can occur as a result of this wiring.

Table 102: Muting wiring combinations and requirements

Signal	A1	A2	B1	B2	C1	Conveyor	Electro-sensitive protective device	Override	Enabled	Muting lamp	Muting status	Override required	Voltage supply 24 V DC
A1	–	Z	Y	Y	Z	Z	Z	Z	Z	Z	Z	X	Y
A2	Z	–	Y	Y	Z	Z	Z	Z	Z	Z	Z	X	Y
B1	Y	Y	–	Z	Z	Z	Z	Z	Z	Z	Z	X	Y
B2	Y	Y	Z	–	Z	Z	Z	Z	Z	Z	Z	X	Y
C1	Z	Z	Z	Z	–	Z	Z	Z	Z	X	X	X	Z
Conveyor	Z	Z	Z	Z	Z	–	X	Z	Z	X	X	X	Z
Electro-sensitive protective device	Z	Z	Z	Z	Z	X	–	X	Z	X	X	X	Z
Override	Z	Z	Z	Z	Z	Z	X	–	Z	Z	X	Z	Z

Z The specified signals may only be installed in the same cable if a short-circuit between these signals can be excluded, e.g., by means of protected cable laying.

Y The specified signals may only be installed in the same cable if sequence monitoring is used or a short-circuit between these signals can be excluded, e.g., by means of protected cable laying.

X The specified signals may be installed in the same cable.

– Not applicable

Short-circuit to 24 V supply voltage

If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal may produce a pulse if the signal for **Override** is reset as a result of short-circuit detection.

WARNING
Undesired override following short-circuit to high

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

- ► Make sure that the transitions of the signals that expect an input pulse (for the muting function blocks: input **Override**) meet the requirements of safety standards and regulations.
  - ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines)
  - ► No short-circuit detection, i.e., no referencing to test outputs

7.9.5 Transition from Stop status to Run status

If there are objects in the area of the muting sensors during the transition of the Flexi Soft safety controller from the Stop status to the Run status and therefore one or more muting sensor signal inputs are set to 1, this generates a muting error.

NOTE
The signaling of the error state at the **Muting error** output is suppressed if the **Electro-sensitive protective equipment** input is set to 1.

Before a new valid muting cycle can be executed, this error must be reset, see "Muting error output", page 191.

7.9.6 Error statuses and reset information

Table 103: Error statuses and reset information for muting function blocks

Diagnostic outputs	Resetting the error status	Comments
Muting error: • Total muting time monitoring error • Concurrence monitoring error • Direction detection error • Sequence monitoring error • Error with transition from Stop status to Run status	Before a muting error of any kind can be reset, a valid muting cycle must be performed in full. This either involves using the Override function, or all of the muting sensor signal inputs must be set to 1 and the **Electro-sensitive protective device** input must be set to 0. A valid muting sequence must follow this. When either of these conditions is met, the Muting error output returns to 0 provided that there is no other error pending.	The **Release** output switches to 0 and the **Fault present** output switches to 1 when the **Muting error** output is set to 1.

7.9.7 Parallel muting

Function block diagram

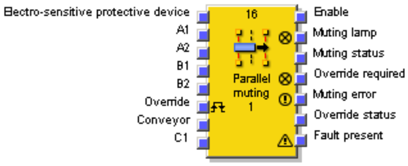


Figure 155: Inputs and outputs of the Parallel muting function block

WARNING
Restricted safety through muting

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

► Observe the notes in the section "Safety notes for muting applications", page 181.

Example of how to arrange sensors for parallel muting

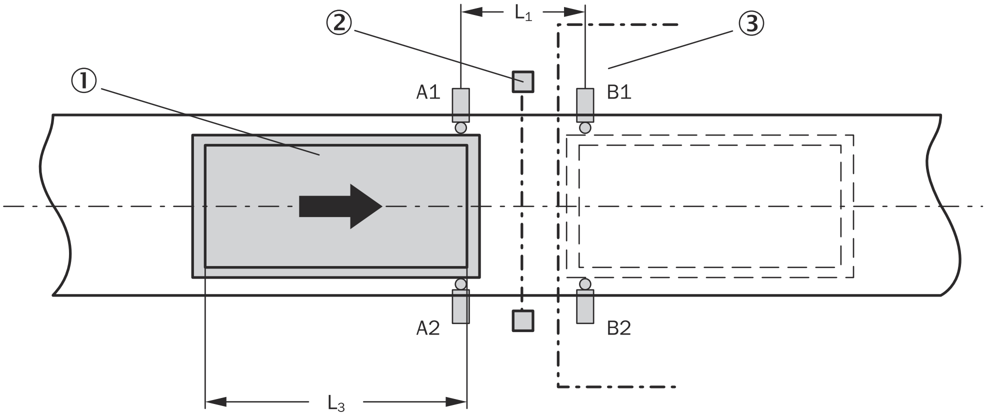


Figure 156: Example for parallel muting

① Transported material

② ESPE (e.g., safety light curtain)

③ Hazardous area

In this example, the material moves from left to right. As soon as the first pair of muting sensors (A1 and A2) is activated, the protection afforded by the protective equipment (electro-sensitive protective equipment) is bypassed.

Calculation of distance L_1

NOTE
In the example, four muting sensors with identical response times are used. The two muting sensor pairs are mounted symmetrically, i.e., at the same distance from the detection range of the ESPE. Different configurations require separate consideration.

The distance L_1 is calculated using the following formula:

$L_1 \geq v \times 2 \times T_{\text{IN muting sensor}}$

The following prerequisites must be met:

- $v \times t > L_1 + L_3$
- $L_1 < L_3$

where...

- L_1 = distance between sensors (arranged symmetrically in relation to detection zone of electro-sensitive protective device)
- L_3 = length of material in conveying direction
- v = speed of material (e.g., of conveyor system)
- t = total muting time set (s)
- $T_{\text{IN muting sensor}}$ = response time until the muting sensors signal is available in the Flexi Soft system. The response time of the slowest muting sensor used to initiate a muting status is decisive. (See section titled “Flexi Soft system response times” in the “Flexi Soft Modular Safety Controller Hardware” operating instructions).

NOTE

- The material can either be moved in both directions or only one transport direction can be allowed using the **Direction detection** configuration parameter.
- When the sensors are arranged in parallel, the position of the muting sensors is also used to monitor the width of the permissible object. Whenever objects move past the muting sensors, the width must always be the same.
- If optical sensors are used for parallel muting, pushbuttons with background suppression are typically used here to prevent a person from unintentionally activating both sensors at the same time.
- Prevent mutual interference of the sensors.
- Notes on wiring: see "Notes on wiring", page 192.

Sequence/timing diagram

The sequence/timing diagram shows a valid muting sequence based on the default parameter setting for this function block.

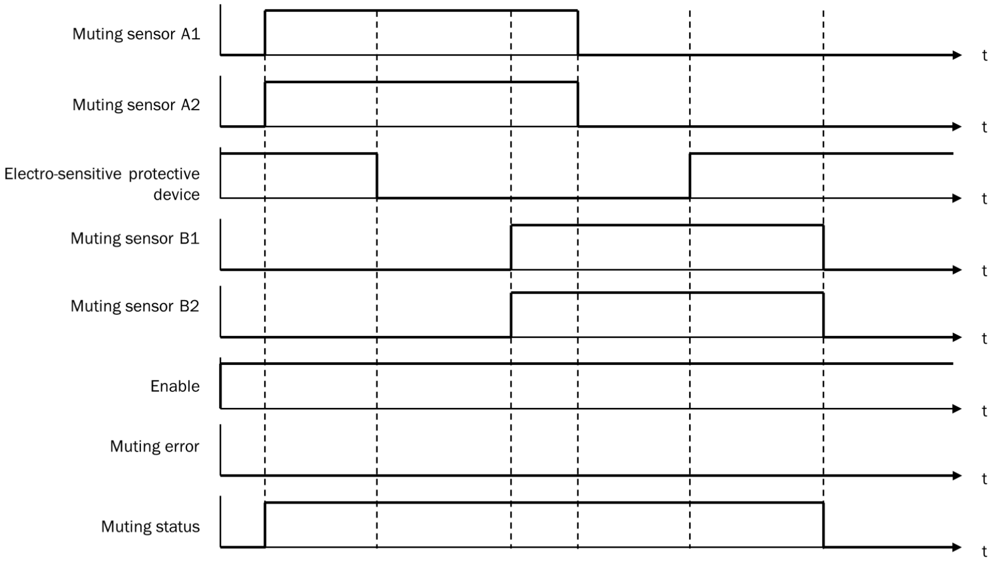


Figure 157: Valid muting sequence when the default parameter setting is used

7.9.8 Sequential muting

Function block diagram

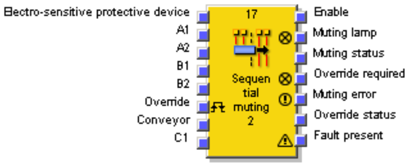


Figure 158: Inputs and outputs of the Sequential muting function block

WARNING

Restricted safety through muting

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

► Observe the notes in the section "Safety notes for muting applications", page 181.

Example of how to arrange sensors for sequential muting

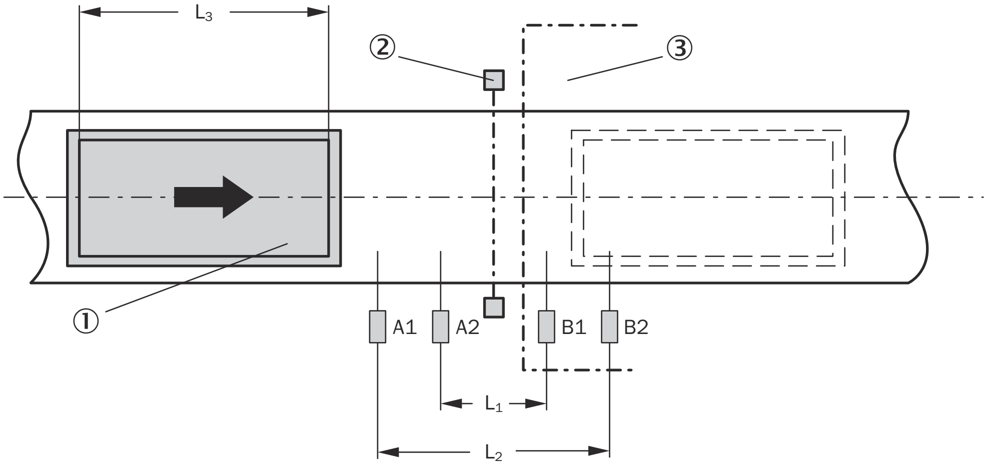


Figure 159: Example for sequential muting

① Transported material
② ESPE (e.g., safety light curtain)
③ Hazardous area

In this example, the material moves from left to right. As soon as muting sensors A1 and A2 are activated, the protection afforded by the protective equipment (electro-sensitive protective equipment) is bypassed.

Calculation of distance L_1

NOTE
In the example, four muting sensors with identical response times are used. The two muting sensor pairs are mounted symmetrically, i.e., at the same distance from the detection range of the ESPE. Different configurations require separate consideration.

The distance L_1 is calculated using the following formula:

$L_1 \geq v \times 2 \times T_{\text{IN muting sensor}}$

The following prerequisites must be met:

- $v \times t > L_1 + L_3$
- $L_2 < L_3$

where...

- L_1 = distance between inner sensors (arranged symmetrically in relation to detection zone of electro-sensitive protective device)
- L_2 = distance between outer sensors (arranged symmetrically in relation to detection zone of electro-sensitive protective device)
- L_3 = length of material in conveying direction
- v = speed of material (e.g., of conveyor system)
- t = total muting time set (s)
- $T_{\text{IN muting sensor}}$ = response time until the muting sensors signal is available in the Flexi Soft system. The response time of the slowest muting sensor used to initiate a muting status is decisive. (See section titled “Flexi Soft system response times” in the “Flexi Soft Modular Safety Controller Hardware” operating instructions.

NOTE

- The material can either be moved in both directions or only one transport direction can be allowed using the **Direction detection** configuration parameter.
- The sensor arrangement shown in this example is suitable for all types of sensor.
- Prevent mutual interference of the sensors.
- Notes on wiring: see "Notes on wiring", page 192.

Sequence/timing diagram

The sequence/timing diagram shows a valid muting sequence based on the default parameter setting for this function block.

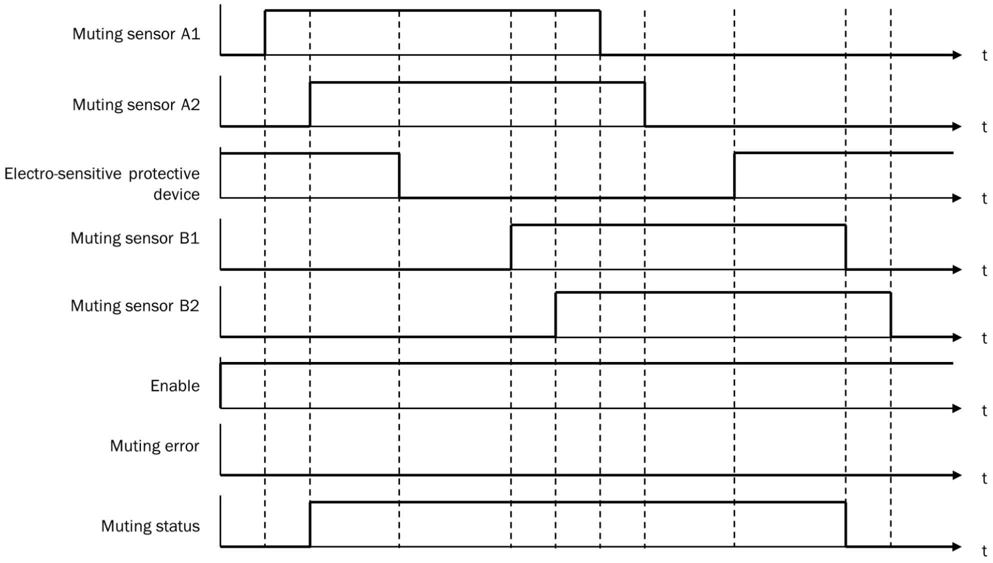


Figure 160: Valid muting sequence when the default parameter setting is used

7.9.9 Cross muting

Function block diagram

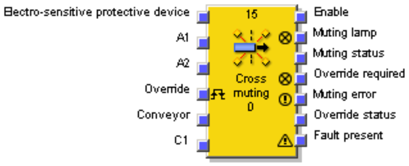


Figure 161: Inputs and outputs of the Cross muting function block

WARNING

Restricted safety through muting

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

► Observe the notes in the section "Safety notes for muting applications", page 181.

Example of how to arrange sensors for cross muting

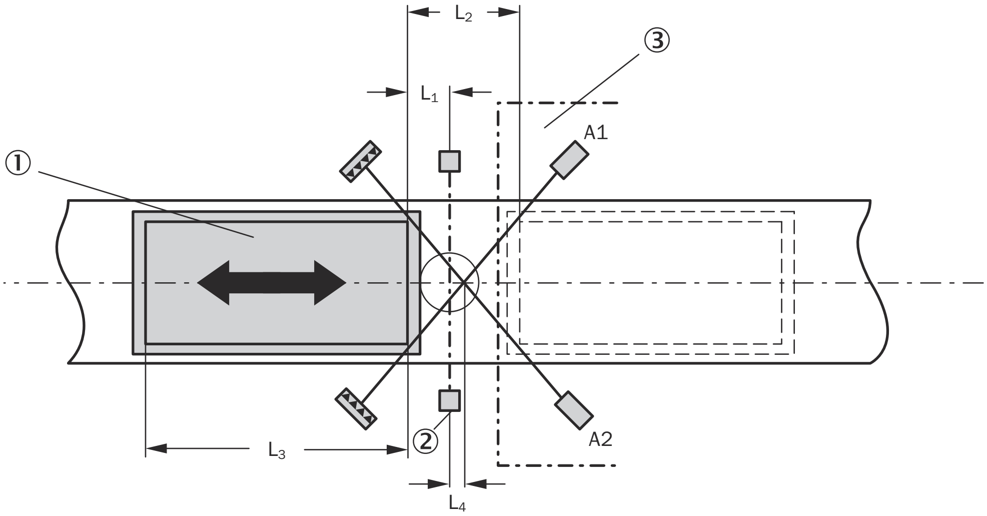


Figure 162: Example for cross muting

① Transported material

② ESPE (e.g., safety light curtain)

③ Hazardous area

In this example, the material can move in both directions. As soon as the pair of muting sensors (A1 and A2) is activated, the protection afforded by the protective equipment (electro-sensitive protective equipment) is bypassed.

Calculation of distance L_1

The distance L_1 is calculated using the following formula:

$L_1 \geq v \times T_{\text{IN muting sensor}}$

The following prerequisites must be met:

- $v \times t > L_2 + L_3$
- $L_4 \geq 0$

where...

- L_1 = minimum distance between detection line of the ESPE and detection by A1 and A2
- L_2 = distance between the two detection lines of the A1 and A2 sensors (sensors activated/sensors clear)
- L_3 = length of material in conveying direction
- L_4 = Distance between the detection line of the ESPE and the point where the muting sensors intersect
- v = speed of material (e.g., of conveyor system)
- t = total muting time set (s)
- $T_{\text{IN muting sensor}}$ = response time until the muting sensors signal is available in the Flexi Soft system. The response time of the slowest muting sensor used to initiate a muting status is decisive. (See section titled “Flexi Soft system response times” in the “Flexi Soft Modular Safety Controller Hardware” operating instructions.

NOTE

- In this example, the material is able to flow in both directions.
- The point where the muting sensors intersect should be placed behind the light beams of the ESPE in the hazardous area. If this is not possible, the point of intersection may be placed exactly in the path of the ESPE light beams, but not in front of it.
- The sensor arrangement shown in the example is suitable for through-beam photoelectric sensors and for photoelectric retro-reflective sensors.
- Prevent mutual interference of the sensors.
- Notes on wiring: see "Notes on wiring", page 192.

Sequence/timing diagram

The sequence/timing diagram shows a valid muting sequence based on the default parameter setting for this function block.

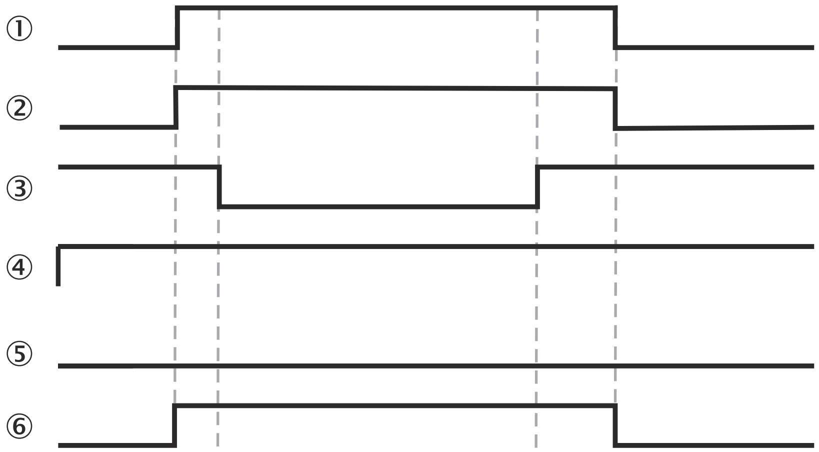


Figure 163: Valid muting sequence when the default parameter setting is used

① A1
② A2
③ Electro-sensitive protective equipment
④ Enable
⑤ Muting error
⑥ Muting status

7.10 Function blocks for press contact monitoring

7.10.1 Overview and general description

There are two types of function block available for press applications, with each one complementing the other. This chapter describes the function blocks for press contact monitoring. These function blocks provide signals for a second type of function block that is responsible for press cycle control.

Two different function blocks are available for press contact monitoring. These can be used to monitor whether the signal sequence of the contacts is correct and whether the press is brought to a stop correctly (ramp down/overrun). The outputs of these function blocks signal what phase of the press cycle the press is currently in (e.g., upstroke or top dead center). Typically, the **Enable** output, the **Top** (top dead center) output, and the **Upstroke** output of a function block for press contact monitoring are connected to the corresponding inputs of one or more press cycle control function blocks.

Table 104: Features of the function blocks for press contact monitoring

	Eccentric press contact	Universal press contact
Typical press types	Eccentric press	Eccentric press Hydraulic press
Direction of movement of press	Forward	Forward and backward
Contacts	Overrun cam Upstroke cam Dynamic cam	TDC BDC SCC
Condition for TDC	If Overrun cam = 1	If TDC = 0
Upstroke condition	If Upstroke cam = 1	If BDC = 1
Overrun monitoring	Optional	Optional
Disable monitoring	Optional	Optional

7.10.2 Eccentric press contact

Function block diagram

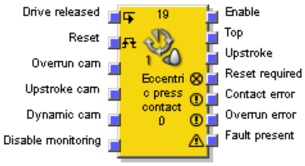


Figure 164: Inputs and outputs of the Eccentric press contact function block

General description

The Eccentric press contact function block can be used for certain types of eccentric press (i.e., mechanical presses). For the minimum configuration, an **Overrun cam** and the **Upstroke cam** are required. Optionally, a **Dynamic cam** can also be connected.

Function block parameters

Table 105: Parameters of the Eccentric press contact function block

Parameter	Possible values
Dynamic cam input	• With • Without
Min. reset pulse time	• 100 ms • 350 ms
Reset input	• With • Without
Disable monitoring input	• With • Without
Use Fault present	• With • Without

Enable output

The **Release** output is used to stop the press and is connected to another press function block with a complementary role, e.g., Press setup or Press single stroke. If no error has been detected, the **Release** output of the function block is set to 1.

If an error is detected in the contact signal sequence, the **Release** output switches to 0, the relevant error/fault output switches to 1, and the **Reset required** output changes to 1. A valid reset sequence is then required at the **Reset** input.

The **Release** output also switches to 0 when monitoring is disabled.

Reset input

A valid reset sequence at the **Reset** input corresponds to a 0–1–0 transition with a pulse duration of at least 100 ms or 350 ms but lasting no longer than 30 s. Pulses any shorter or longer than these limits are ignored.

NOTE

- If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal for the **Reset** function may produce a pulse if it is reset as a result of short-circuit detection.
- Type C standards, such as EN 692 and EN 693, contain requirements for the use of safety-related signals. For example, within the context of overrun errors, it may be necessary to provide the restart signal with a suitable form of protection (e.g., by using a key switch or installing it inside a locked control cabinet).

WARNING
Undesired reset following short-circuit to high

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

► Make sure that the transitions of the signals for the **Reset** function meet the requirements of safety standards and regulations.
  ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines).
  ► No short-circuit detection, i.e., no referencing to test outputs.
► Observe the applicable standards and regulations for safety-related signals.

If the **Reset** input is deactivated, an error can only be reset by stopping the logic program, e.g., by briefly switching the power off and on again, or by using configuration software to make the system transition from the Run status to the Stop status and then back to the Run status.

Top output and Upstroke output

The **Top** (top dead center) output is typically used to stop the press. It is connected to another press function block with a complementary role, e.g., Press setup or Press single stroke.

The **Upstroke** output is typically connected to another press function block with a complementary role, e.g., Press setup or Press single stroke. It can also be used to trigger upstroke muting.

The Eccentric press contact function block sets the **Upstroke** and **Top** outputs based on changes to the contact input values. If the function block detects an error, both outputs are set to 0.

Without Dynamic cam

The **Upstroke** output switches to 1 when a rising signal edge (0–1) occurs at the **Upstroke cam** input and switches to 0 when a rising signal edge occurs at the **Overrun cam** input.

The **Top** output switches to 1 when the **Overrun cam** input is set to 1.

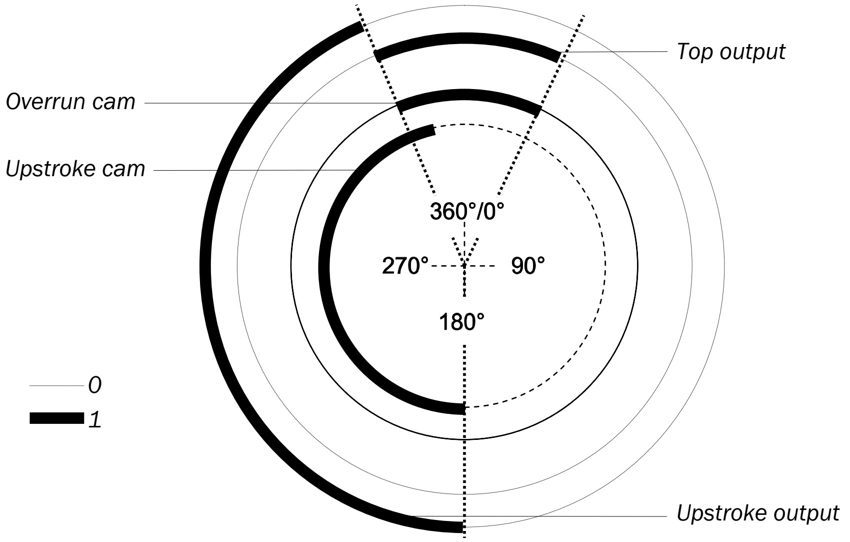


Figure 165: Press cycle for the Eccentric press contact function block without Dynamic cam

With Dynamic cam

If **Dynamic cam** is configured for this function block, the start of the TDC phase can be brought forward by means of a falling signal edge (1–0) at the **Dynamic cam** input.

The **Upstroke** output switches to 1 when a rising signal edge (0–1) occurs at the **Upstroke cam** input. It switches to 0 either when a rising signal edge occurs at the **Overrun cam** input or when a falling signal edge occurs at the **Dynamic cam** input (depending on which one occurs first).

The **Top** output switches to 1 when a rising signal edge occurs at the **Overrun cam** input or when a falling signal edge occurs at the **Dynamic cam** input (depending on which one occurs first). The **Top** output switches to 0 when a falling signal edge occurs at the **Overrun cam** input.

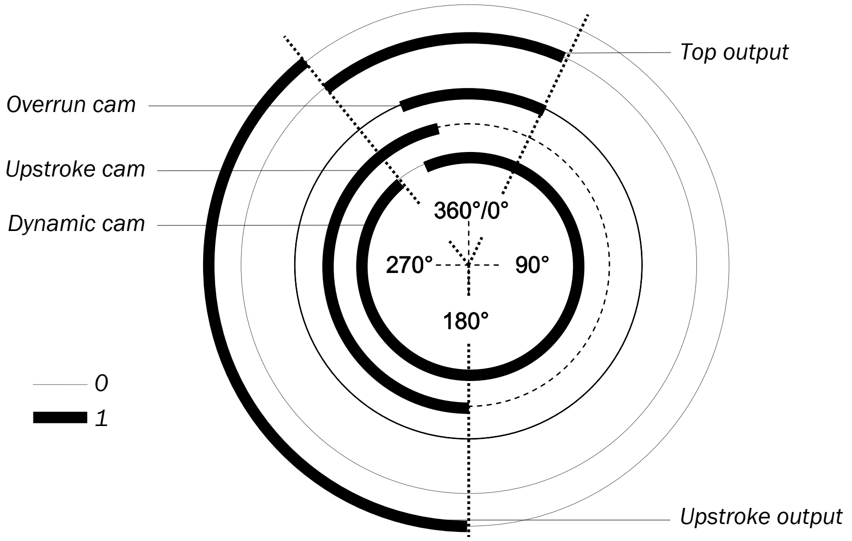


Figure 166: Press cycle for the Eccentric press contact function block with Dynamic cam during upstroke

If a falling signal edge occurs at the **Dynamic cam** input while the **Upstroke cam** input is set to 0 (i.e., during the downward phase of the press cycle), the **Top** output switches to 1 until a rising signal edge is detected at the **Upstroke cam** input. The **Upstroke** output remains set to 0 for the rest of the press cycle.

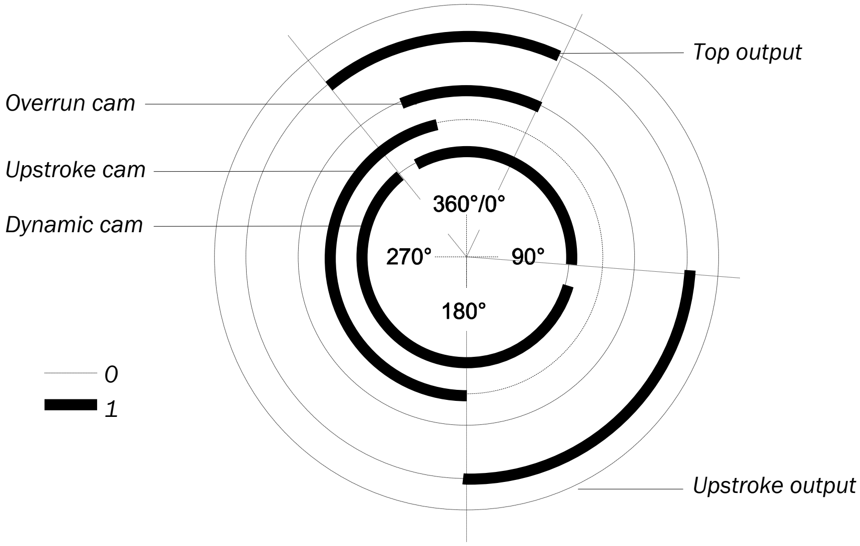


Figure 167: Press cycle for the Eccentric press contact function block with Dynamic cam during upstroke and downward movement

NOTE

If the **Upstroke cam** input is already set to 1 when the contact inputs start being monitored (e.g., during the first logic cycle after an error has been reset or after monitoring has been activated via the **Disable monitoring** input), the **Upstroke** output remains set to 0 until the first actual transition from 0 to 1 is detected at the **Upstroke cam** input.

Contact monitoring

The input signals for the **Overrun cam**, **Upstroke cam**, and **Drive released** inputs must obey the rules illustrated in the figure and described below.

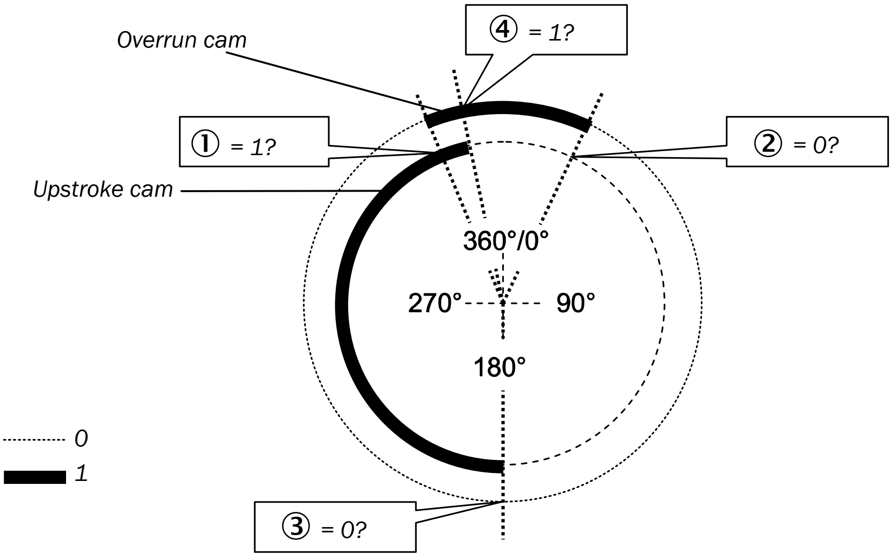


Figure 168: Contact monitoring with the Eccentric press contact function block

- ① The overrun has to commence during the upstroke phase: The rising signal edge at the **Overrun cam** input (0–1) has to occur while the **Upstroke cam** input is set to 1.
- ② The overrun has to finish on completion of the upstroke phase: The falling signal edge at the **Overrun cam** input (1–0) has to occur when the **Upstroke cam** input is set to 0.

- ③ The upstroke phase has to commence once the overrun has finished: The rising signal edge at the **Upstroke cam** input (0–1) has to occur while the **Overrun cam** input is set to 0.
- ④ The upstroke phase has to finish during the overrun: The falling signal edge at the **Upstroke cam** input (1–0) has to occur while the **Overrun cam** input is set to 1.

A failure to meet even one of these conditions during operation is sufficient to set the **Release** output to 0 and the **Contact error** output to 1.

A valid sequence meeting these conditions would be:

1. Start condition: **Overrun cam** input = 1
   Upstroke cam input = 0
2. **Overrun cam** input: 1–0
3. **Upstroke cam** input: 0–1
4. **Overrun cam** input: 0–1
5. **Upstroke cam** input: 1–0

WARNING
Non-safe signals

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

► Make sure that the application conforms to all applicable standards and regulations.
► Only use safety-related signals for safety-related applications.
► In the case of the **Upstroke cam** input, this is particularly important if the **Upstroke** output is used for upstroke muting, e.g., in conjunction with a function block for press cycle control.

To comply with the safety regulations it may be necessary to use tested switches that rely on different test sources for the contact inputs in each case. This means that the **Overrun cam**, **Upstroke cam**, and **Dynamic cam** inputs must be connected to different FX3-XTIO or FX3-XTDI modules.

NOTE
An FX3-XTDI module only has two test sources even though it features eight test output terminals.

Overrun monitoring

The Eccentric press contact function block monitors the overrun of the press. If contact stops being made with the **Overrun cam** contact but the press should actually have come to a stop by now, the function block detects an overrun error.

The **Drive released** input then has to obey the rules illustrated in the figure and described below, see figure 169, page 205.

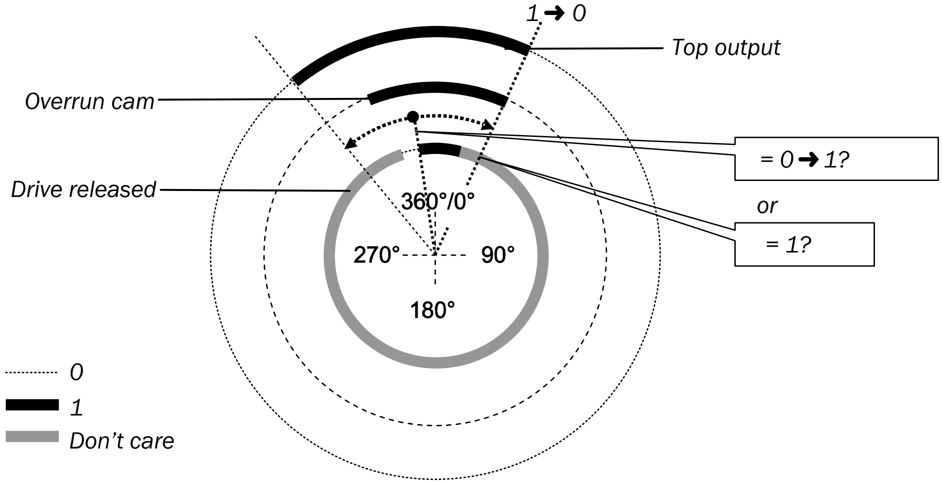


Figure 169: Overrun monitoring with the Eccentric press contact function block

Either a transition from 0 to 1 must occur at the **Drive released** input while the **Top** output is set to 1 or the **Drive released** input must be set to 1 once the **Overrun cam** (1–0) has finished. A failure to meet either of these conditions causes the **Release** output to switch to 0 and the **Overrun error** output to change to 1.

The **Drive released** input must be connected to the signal that controls the physical output of the press drive so that the function block can detect whether the press is currently running or has been stopped. Typically, this will be the **Enable** output of a subsequent Press setup or Press single stroke function block.

NOTE
The signal that controls the physical output for the press drive must be controlled with a jump address or a CPU marker.

- When using a jump address, this signal must form a logical loopback. To achieve this, connect the outputs of this function block to the inputs of the subsequent function blocks and then connect the jump address to the **Drive released** input. This is particularly important if all connections to the subsequent function blocks are established using jump addresses.
- If using a CPU marker, you must use a routing function block to route the signal not only to the physical output for the press drive but also to the output of the CPU marker.

Disable monitoring

This optional input allows you to deactivate the Monitoring function under certain conditions so that the function block is prevented from switching to an error status. This may be useful for certain operating modes, e.g., while setting up the machine or when the press is running in reverse.

If the **Disable monitoring** input is set to 1, the **Release** output of the Eccentric press contact function block is 0, and contact signal sequence and overrun monitoring are deactivated, provided that there is no error pending. This does not affect the error/fault outputs.

If the **Disable monitoring** input is set to 1 and an error is pending at the same time, the error can be reset.

If the **Disable monitoring** input switches from 1 to 0, the function block behaves in the same way as during a transition from the Stop status to the Run status, i.e., the **Release** output switches back to 1.

7.10.3 Universal press contact

Function block diagram

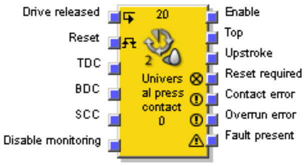


Figure 170: Inputs and outputs of the Universal press contact function block

General description

The Universal press contact function block can be used for different types of press (e.g., hydraulic presses and eccentric presses, i.e., mechanical presses). For the minimum configuration, you only need to use the **TDC** contact. However, you also have the option of connecting the **BDC** and **SCC** inputs as well.

- The **Upstroke** output is only available when the **BDC** input is activated.
- Overrun monitoring is only possible when the **SCC** input is activated.
- If **BDC** and **SCC** are not used, no plausibility check can be performed for this function block. In this case, overrun monitoring is not possible. This only leaves the function for making the **Top** output signal available.

WARNING
No plausibility check

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

► Always use the **BDC** and **SCC** inputs for safety applications.

Function block parameters

Table 106: Parameters of the Universal press contact function block

Parameter	Possible values
SCC input	• With • Without
BDC input	• With • Without
Number of BDC signals per cycle	• 1 (e.g., eccentric press) • 0 to 2 (e.g., hydraulic press)
Min. reset pulse time	• 100 ms • 350 ms
Reset input	• With • Without
Disable monitoring input	• With • Without
Use Fault present	• With • Without

Enable output

The **Release** output is used to stop the press and is connected to another press function block with a complementary role, e.g., Press setup or Press single stroke. If no error has been detected, the **Release** output of the function block is set to 1.

If an error is detected in the contact signal sequence, the **Release** output switches to 0, the relevant error/fault output switches to 1, and the **Reset required** output changes to 1. A valid reset sequence is then required at the **Reset** input.

The **Release** output also switches to 0 when monitoring is disabled.

Reset input

A valid reset sequence at the **Reset** input corresponds to a 0–1–0 transition with a pulse duration of at least 100 ms or 350 ms but lasting no longer than 30 s. Pulses any shorter or longer than these limits are ignored.

NOTE

- If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal for the **Reset** function may produce a pulse if it is reset as a result of short-circuit detection.
- Type C standards, such as EN 692 and EN 693, contain requirements for the use of safety-related signals. For example, within the context of overrun errors, it may be necessary to provide the restart signal with a suitable form of protection (e.g., by using a key switch or installing it inside a locked control cabinet).

WARNING
Undesired reset following short-circuit to high

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

- ► Make sure that the transitions of the signals for the **Reset** function meet the requirements of safety standards and regulations.
  - ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines).
  - ► No short-circuit detection, i.e., no referencing to test outputs.
- ► Observe the applicable standards and regulations for safety-related signals.

If the **Reset** input is deactivated, an error can only be reset by stopping the logic program, e.g., by briefly switching the power off and on again, or by using configuration software to make the system transition from the Run status to the Stop status and then back to the Run status.

Top output and Upstroke output

The **Top** (top dead center) output is typically used to stop the press and is connected to another press function block with a complementary role, e.g., Press setup or Press single stroke.

The **Upstroke** output is typically connected to another press function block with a complementary role, e.g., Press setup or Press single stroke. It can also be used to trigger upstroke muting.

The Universal press contact function block sets the **Upstroke** and **Top** outputs based on changes to the contact input values. If the function block detects an error, both outputs are set to 0.

The **Top** output switches to 1 when the **TDC** input is set to 0. The **Upstroke** output switches to 1 when a rising signal edge (0–1) occurs at the **BDC** input. It switches to 0 when a falling signal edge occurs at the **TDC** input or when a falling signal edge occurs at the **BDC** input (depending on which one occurs first).

If the **BDC** input is set to 1 when the function block starts (switch-on, deactivated ➜ activated), the **Upstroke** output remains set to 0 throughout the first press cycle.

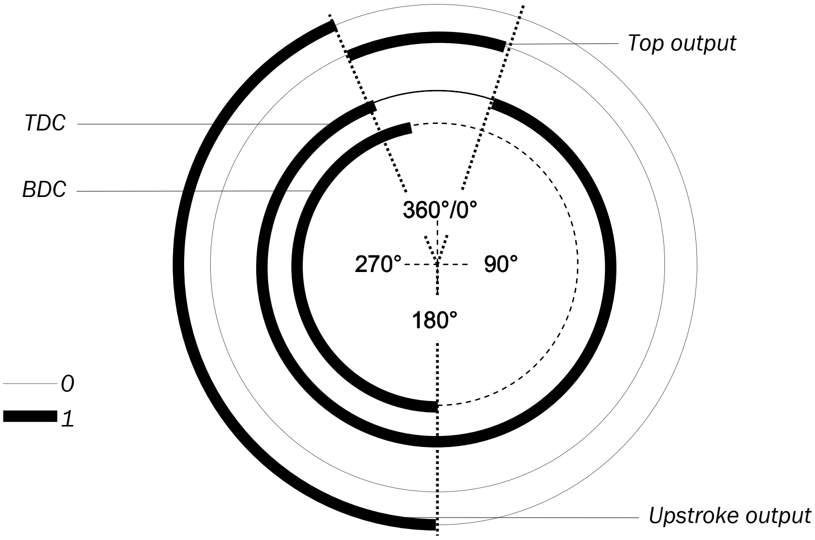


Figure 171: Press cycle for the Universal press contact function block with falling signal edge at TDC input before BDC input

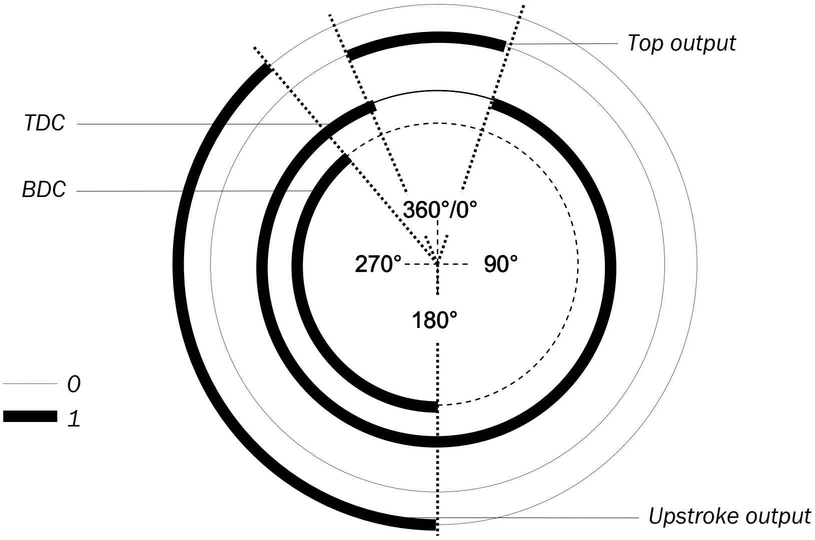


Figure 172: Press cycle for the Universal press contact function block with falling signal edge at BDC input before TDC input

A second rising signal edge at the **BDC** input does not start the upstroke phase again. This applies when the **Number of BDC signals per cycle** parameter is set to 0 to 2 (e.g., hydraulic press) and the press is moving forward and backward in the lower range.

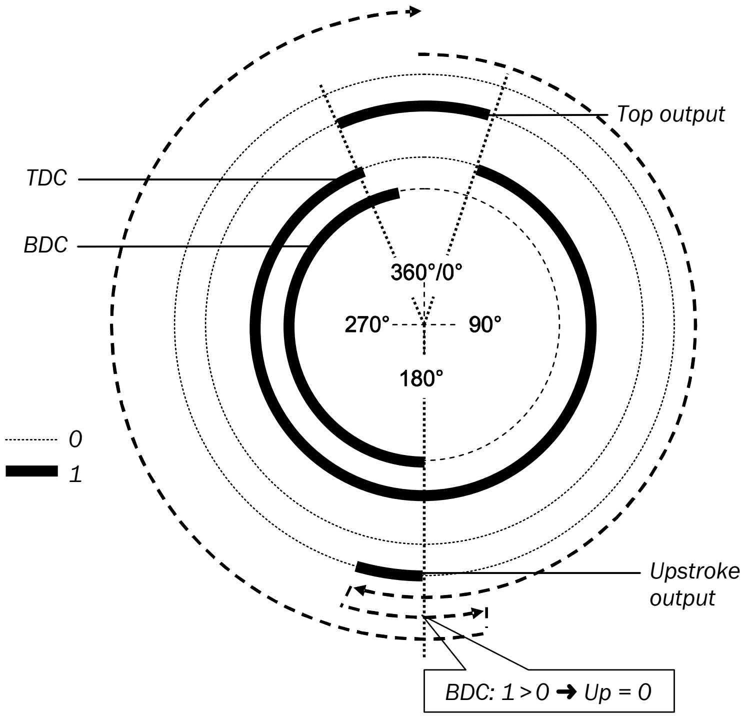


Figure 173: Press cycle for the Universal press contact function block with two BDC transitions

If this setting is configured and no pulse at all occurs at the **BDC** input during the cycle, the **Upstroke** output remains set to 0 throughout the entire cycle.

NOTE
If the **BDC** input is already set to 1 when the contact inputs start being monitored (e.g., during the first logic cycle after an error has been reset or after monitoring has been activated via the **Disable monitoring** input), the **Upstroke** output remains set to 0 throughout the first logic cycle. The next transition from 0 to 1 at the **BDC** input is only accepted if this has been preceded by a transition from 1 to 0 at the **Top** output.

Monitoring of TDC

In the course of each cycle, exactly one pulse must occur at the **TDC** input. A violation of this rule can only be detected if the **SCC** input is activated and/or the **BDC** input is activated plus the **Number of BDC signals per cycle** parameter has been set to a value of 1 (e.g., eccentric press).

Monitoring of SCC

If the **SCC** input is activated, the input signals for **SCC** must obey the rules illustrated in the figure and described below.

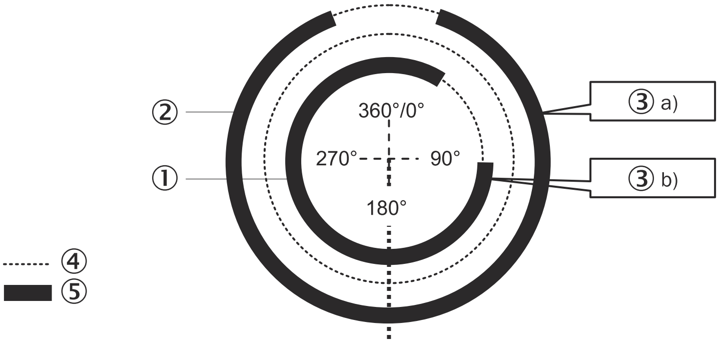


Figure 174: Contact monitoring with the Universal press contact function block when SCC is activated

① SCC

② TDC

③ a) = 1?

or

b) = 1?

④ 0

⑤ 1

In the course of each cycle, exactly one pulse must occur at the **SCC** input. The rising signal edge at the **SCC** input (0–1) must precede the falling signal edge at the **TDC** input. The falling signal edge at the **SCC** input (1–0) must occur after the rising signal edge at the **TDC** input. This means that at least one of the two inputs must be set to 1 at any given time.

Monitoring of BDC

If the **BDC** input is activated and the **SCC** input is deactivated, the input signals for **BDC** must obey the rules illustrated in the figure and described below.

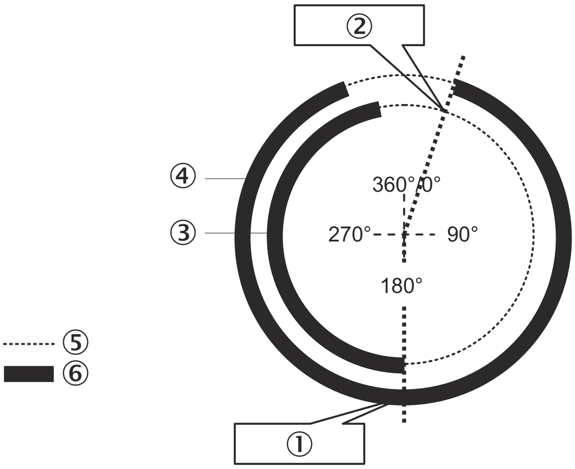


Figure 175: Contact monitoring with the Universal press contact function block when BDC is activated

① = 1?

② = 0?

③ BDC

④ TDC

⑤ 0

⑥ 1

- ① When the signal at **BDC** begins (0–1), it must be close to 180° and must occur while the **TDC** input is set to 1.
- ② The signal at **BDC** must end (1–0) before the rising signal edge (0–1) occurs at the **TDC** input. In other words, the **BDC** input must be set to 0 when a rising signal edge (0–1) occurs at the **TDC** input.

Monitoring of BDC and SCC

If the **BDC** and **SCC** inputs are both activated, the signals at the **BDC** input must obey the rules illustrated in the figure and described below.

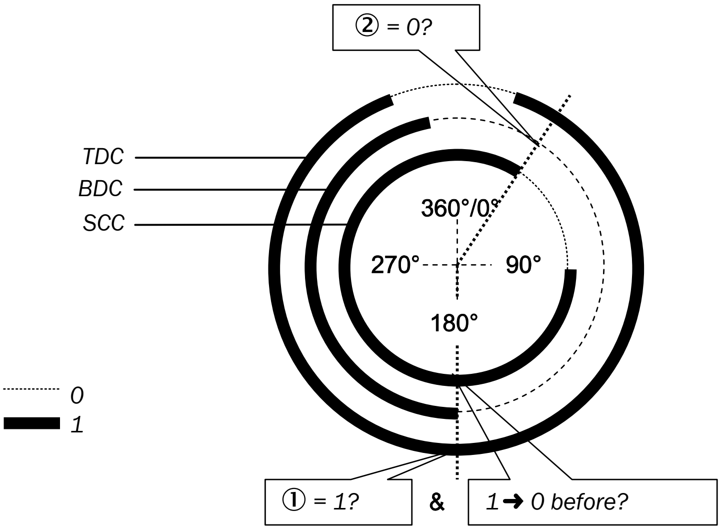


Figure 176: Contact monitoring with the Universal press contact function block when the BDC and SCC inputs are both activated

- ① When the signal at **BDC** begins (0–1), it must be close to 180° and must occur while the **TDC** input is set to 1 and after a falling signal edge (1–0) has occurred at the **SCC** input (although the **SCC** input is allowed to have returned to 1 in the meantime).
- ② The signal at **BDC** must end (1–0) before the falling signal edge (1–0) occurs at the **SCC** input. In other words, the **BDC** input must be set to 0 when a falling signal edge (1–0) occurs at the **SCC** input.

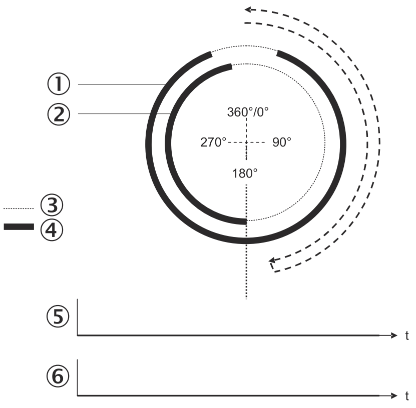
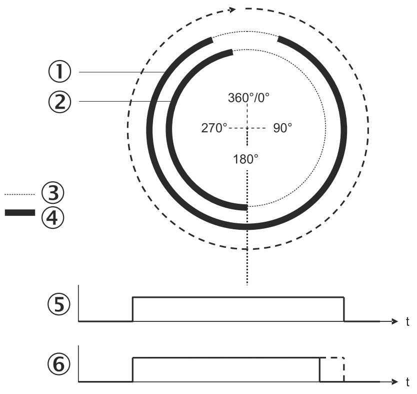
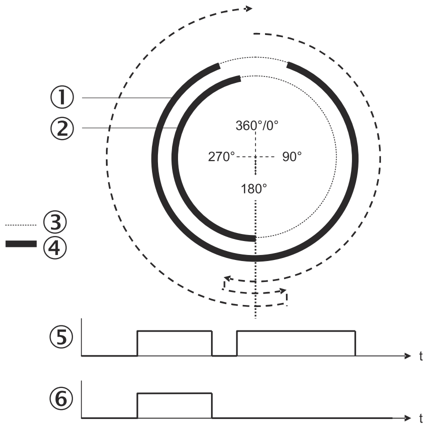
A valid sequence meeting the conditions for **BDC** and **SCC** would be:

1. Start condition: TDC = 0, BDC = 0, SCC = 1
2. TDC: 0–1
3. Drive released = 1 (fulfills the condition for overrun monitoring)
4. SCC: 1–0
5. BDC: 0–1
6. SCC: 0–1
7. TDC: 1–0 and TDC: 1–0 (order is irrelevant)

Depending on the type of press (e.g., hydraulic press) the **BDC** signal (step 5) may occur twice or not at all, rather than just once. To prevent this from causing a contact error, the **Number of BDC signals per cycle** must be set to 0 to 2 (e.g., hydraulic press). With this setting, the conditions for **BDC** continue to apply for every pulse at the **BDC** input with the exception of a falling signal edge at the **SCC** input (step 4).

In addition, the number of signals (0–1–0) that occur at the **BDC** input must match the value configured, i.e., either exactly one signal or any number of signals between 0 and 2.

Table 107: Timing diagrams for 0, 1, and 2 BDC signals per cycle

0 BDC signals per cycle	1 BDC signal per cycle	2 BDC signals per cycle
 ① ② 360°/0° 270° 90° 180° ③ ④ ⑤ t ⑥ t	 ① ② 360°/0° 270° 90° 180° ③ ④ ⑤ t ⑥ t	 ① ② 360°/0° 270° 90° 180° ③ ④ ⑤ t ⑥ t
① TDC	① TDC	① TDC
② BDC	② BDC	② BDC
③ 0	③ 0	③ 0
④ 1	④ 1	④ 1
⑤ TDC input	⑤ TDC input	⑤ TDC input
⑥ Upstroke output	⑥ Upstroke output	⑥ Upstroke output

A failure to meet even one of these conditions during operation is sufficient to set the **Release** output to 0 and the **Contact error** output to 1.

WARNING
Non-safe signals

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

- ► Make sure that the application conforms to all applicable standards and regulations.
- ► Only use safety-related signals for safety-related applications.
- ► In the case of the **BDC** input, this is particularly important if the **Upstroke** output is used for upstroke muting, e.g., in conjunction with a function block for press cycle control.

Setting the **Number of BDC signals per cycle** parameter to 0 to 2 (e.g., hydraulic press) reduces the fault detection capabilities of the function block with the result that not all input errors can be detected (e.g., short-circuit to 0 V at the **BDC** input).

To comply with the safety regulations it may be necessary to use tested switches that rely on different test sources for the contact inputs in each case. This means that the **TDC**, **BDC**, and **SCC** inputs must be connected to different FX3-XTIO or FX3-XTDI modules.

NOTE
An FX3-XTDI module only has two test sources even though it features eight test output terminals.

Overrun monitoring

If the **SCC** input is activated, the Universal press contact function block monitors the press overrun. If contact stops being made with the **SCC** contact but the press should actually have come to a stop by now, the function block detects an overrun error.

The **Drive released** input then has to obey the rules illustrated in the figure and described below.

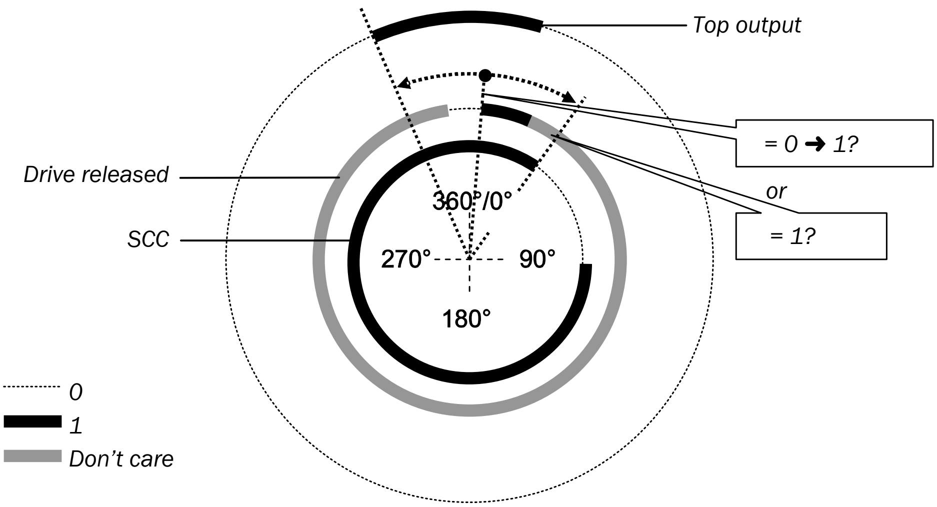


Figure 177: Overrun monitoring with the Universal press contact function block

Either a transition from 0 to 1 must occur at the **Drive released** output between the moment when the **Top** output transitions from 0 to 1 and when the **SCC** signal ends (1–0) or, alternatively, the **Drive released** input must be 1 when the **SCC** signal ends (1–0). A failure to meet either of these conditions causes the **Release** output to switch to 0 and the **Overrun error** output to change to 1.

The **Drive released** input must be connected to the signal that controls the physical output of the press drive so that the function block can detect whether the press is currently running or has been stopped. Typically, this will be the **Enable** output of a subsequent Press setup or Press single stroke function block.

NOTE
The signal that controls the physical output for the press drive must be connected using a jump address or a CPU marker.

- When using a jump address, this signal must form a logical loopback. This is indicated by a clock symbol at the input of the jump address. To achieve this, connect the outputs of this function block to the inputs of the subsequent function blocks and then connect the jump address to the **Drive released** input. This is particularly important if all connections to the subsequent function blocks are established using jump addresses.
- If using a CPU marker, you must use a routing function block to route the signal not only to the physical output for the press drive but also to the output of the CPU marker.

Disable monitoring

This optional input allows you to deactivate the Monitoring function under certain conditions so that the function block is prevented from switching to an error status. This may be useful for certain operating modes, e.g., while setting up the machine or when the press is running in reverse.

If the **Disable monitoring** input is set to 1, the **Release** output is 0, and contact signal sequence and overrun monitoring are deactivated, provided that there is no error pending. This does not affect the error/fault outputs.

If the **Disable monitoring** input is set to 1 and an error is pending at the same time, the error can be reset.

If the **Disable monitoring** input switches from 1 to 0, the function block behaves in the same way as during a transition from the Stop status to the Run status, i.e., the **Release** output switches back to 1.

7.11 Function blocks for press cycle control

7.11.1 Press setup

Function block diagram

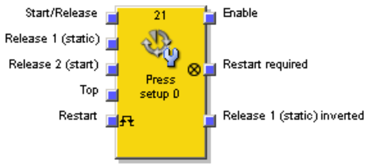


Figure 178: Inputs and outputs of the Press setup function block

General description

The Press setup function block is generally used together with the Universal press contact or Eccentric press contact function blocks for the purpose of setting up the press and so that the information from the **Top** output can be made available as an input for this function block. The **Top** output is required for single-stroke mode. The press can, for example, be controlled by means of a two-hand control system.

Function block parameters

Table 108: Parameters of the Press setup function block

Parameter	Possible values
Restart interlock condition	• Never • When Release 1 or Start/Release is 0 • When Release 1 is 0 or Top changes to 1 • Always
Release 2 (start) input	• With • Without
Single stroke monitoring	• Active • Deactivated
Min. restart pulse time	• 100 ms • 350 ms

NOTE
If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal for the **Restart** function may produce a pulse if it is reset as a result of short-circuit detection.

WARNING
Undesired restart following short-circuit to high

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

- ► Make sure that the transitions of the signals for the **Restart** function meet the requirements of safety standards and regulations.
  - ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines).
  - ► No short-circuit detection, i.e., no referencing to test outputs.

Function block input signals

The Press setup function block supports the following input signals:

Start/Release

The **Start/Release** input signal is used to indicate the beginning and the end of the press movement. A rising signal edge (0–1) at the **Start/Release** input signals that the press is starting. The value 0 at the **Start/Release** input signals that the press is stopping. If the **Restart interlock condition** configuration parameter is set to **When Release 1 or Start/Release is 0**, a valid **restart** sequence must be performed following a stop that was caused by a 0 signal at the **Start/Release** input.

Release 1 (static)

The **Release 1 (static)** input signal is mandatory. The **Release** output always switches to 0 as soon as **Release 1 (static)** is set to 0.

If this function block is used together with a press contact function block (e.g., Eccentric press contact or Universal press contact), the **Release** output of the relevant press contact function block must be connected to the **Release 1 (static)** input of the Press setup function block.

Release 2 (start)

WARNING
Incorrect use of the **Release 2 (start)** input

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 **Release 2 (start)** input for safety purposes, e.g. to trigger an emergency stop.

The **Release 2 (start)** input is optional. If **Release 2 (start)** is used, the **Release** output can only switch to 1 (e.g., during switch-on) if **Release 2 (start)** is set to 1. When the **Release** output is set to 1, **Release 2 (start)** stops being monitored.

Top

The **Top** input is used for single stroke monitoring. It is used to determine the end of the press cycle (i.e., when the press has reached the top dead center position). The **Top** input must only be connected to a **Top** output of a Universal press contact or Eccentric press contact function block, or to an equivalent signal source – but never to anything else.

If the **Single stroke monitoring** configuration parameter is set to Active, the **Release** output changes to 0 when the **Top** input switches from 0 to 1.

⚠ **WARNING**
Incorrect use of the **Top** input

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 **Top** input for safety purposes, e.g. to trigger an emergency stop.

Restart input

If the **Restart interlock condition** configuration parameter has been set to Never, no **Restart** signal is required to restart the press following a stop. The **Restart interlock condition** parameter can also be set to the following values:

- **When Release 1 or Start/Release is 0**
- **When Release 1 is 0 or Top changes to 1**
- **Always**

This parameter determines when a **Restart** signal is expected as an input signal for the function block.

If the **Release** output switches to 0 because the specified **Restart interlock condition** configuration parameter has been set, the **Release** output can only be reset once a valid **restart** sequence has been completed with a 0–1–0 transition (at least 100 ms or 350 ms; shorter pulses and pulses lasting longer than 30 s will be ignored).

Function block output signals

The Press setup function block supports the following output signals:

Restart required

The **Restart required** output is set to 1 if a valid **restart** sequence is expected at the **Restart** input.

Enable

The **Release** output is 1 when **Restart required** is 0 (i.e., no restart is required) and the following conditions are met:

- **Single stroke monitoring** is set to Deactivated, **Release 1 (static)** is set to 1, **Release 2 (start)** (if configured) is also set to 1, and a rising signal edge (0–1) is detected at the **Start/Release** input.

Or:

- **Single stroke monitoring** is set to Active, **Start/Release** switches from 0 to 1, **Release 1 (static)** is 1, and **Release 2 (start)** (if configured) is also 1. In this case, the **Release** output changes to 0 if the **Top** input switches from 0 to 1.

Release 1 (static) inverted

The **Release 1 (static) inverted** output indicates whether there is a release signal present at the Press setup function block. When **Release 1 (static)** is 1, **Release 1 (static) inverted** is 0, and vice versa.

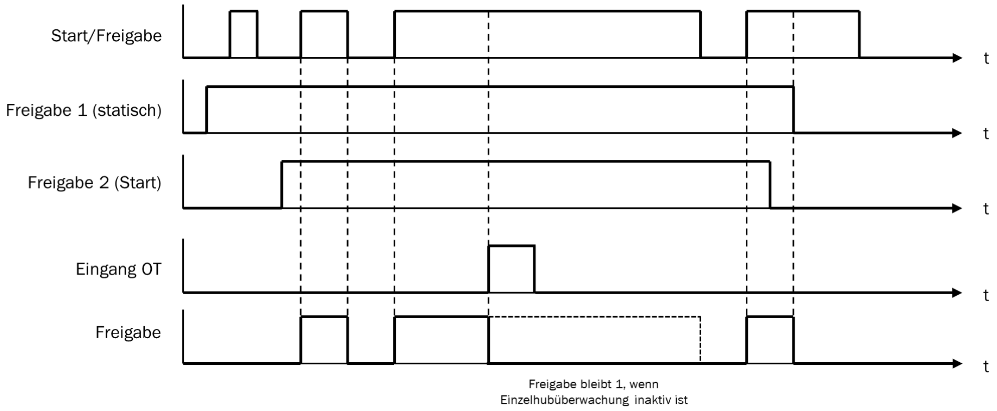


Figure 179: Sequence/timing diagram for the Press setup function block

7.11.2 Press single stroke

Function block diagram

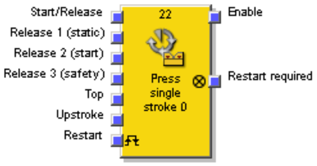


Figure 180: Inputs and outputs of the Press single stroke function block

General description

The Press single stroke function block is generally used together with the Universal press contact or Eccentric press contact function blocks so that the information from the **Top** and **Upstroke** outputs can be made available as an input for this function block. The **Top** output is required for single-stroke mode. The press can, for example, be controlled by means of a two-hand control system or by using a PSDI mode function block in conjunction with a safety light curtain.

Single stroke monitoring is always active and is not configurable. Consequently, the **Release** output always switches to 1 whenever the **Top** input changes to 0. The requirements for a restart depend on how the **Restart interlock condition** parameter is configured.

Function block parameters

Table 109: Parameters of the Press single stroke function block

Parameter	Possible values
Restart interlock condition	• Never • When Release 1 or Release 3 or Start/Release is 0 • When Release 1 or Release 3 is 0 or Top changes to 1 • Always • When Release 1 or Release 3 is 0
Release 2 (start) input	• With • Without
Release 3 (safety) input	• With • Without
Mode for Start/Release input	• Start and static release (jog mode) • Start only (stopping not possible)
Mode for upstroke muting	• Deactivated • For Release 3 • For Release 3 and Start/Release
Max. time for upstroke muting	0 = infinite, 1 to 7,200 s. The Upstroke input is only present when this value is set to a value other than 0.
Min. restart pulse time	• 100 ms • 350 ms
Ignore Release 3 (safety) for restart interlock while in top position	• Yes • No

NOTE

If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal for the **Restart** function may produce a pulse if it is reset as a result of short-circuit detection.

WARNING

Undesired restart following short-circuit to high

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

- ► Make sure that the transitions of the signals for the **Restart** function meet the requirements of safety standards and regulations.
  - ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines).
  - ► No short-circuit detection, i.e., no referencing to test outputs.

Function block input parameters and input signals

The Press single stroke function block supports the following input signals:

Start/Release

The **Start/Release** input signal is used to indicate the beginning and the end of the press movement. A rising signal edge (0–1) at the **Start/Release** input signals that the press is starting. The value 0 at the **Start/Release** input signals that the press is stopping.

If the **Mode for Start/Release input** parameter is set to **Start only (stopping not possible)**, the press cannot be stopped with the **Start/Release** input.

WARNING
Restricted safety in **Start only (stopping not possible)** mode

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

► Implement additional safety measures (e.g. secure the hazardous point using a light curtain) if the **mode for the Start/Release input** parameter is set to **Start only (stopping not possible).**

After a stop caused by a 0 at the Start/Release input, a valid Restart sequence is required if the function block has been configured as follows:

- The **Mode for Start/Release input** parameter is set to **Start and static release (jog mode).**
- The **Restart interlock condition** parameter is set to either **Always** or **When Release 1 or Release 3 or Start/Release is 0.**

The enable signal of a two-hand control system or a PSDI mode function block is the ideal signal to connect to the **Start/Release** input.

Release 1 (static)

The **Release 1 (static)** input signal is mandatory. The **Release** output always switches to 0 as soon as **Release 1 (static)** is set to 0.

If the Press single stroke function block is used together with a press contact function block (e.g., Eccentric press contact or Universal press contact), the release signal of the relevant press contact function block must be connected to the **Release 1 (static)** input of the Press single stroke function block.

Release 2 (start)

WARNING
Incorrect use of the **Release 2 (start)** input

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 **Release 2 (start)** input for safety purposes, e.g. to trigger an emergency stop.

The **Release 2 (start)** input is optional. If **Release 2 (start)** is used, the **Release** output can only switch to 1 (e.g., during switch-on) if **Release 2 (start)** is set to 1. When the **Release** output is set to 1, **Release 2 (start)** stops being monitored.

Release 3 (safety)

The **Release 3 (safety)** input signal is optional. The **Release** output can only transition from 0 to 1 when **Release 3 (safety)** is set to 1. If **Release 3 (safety)** is 0 and **Upstroke** is also 0, the **Release** output is set to 0 and a **restart** sequence must be performed in accordance with the settings.

If **Release 1 (static)** and **Upstroke** are 1 and the **max. time for upstroke muting** has been set to a value greater than 0, the **Release 3 (safety)** signal is bypassed (muted).

Top

The **Top** input is used for single stroke monitoring. It is used to determine the end of the press cycle (i.e., when the press has reached the top dead center position). The **Top** input must only be connected to a **Top** output of a Universal press contact or Eccentric press contact function block, or to an equivalent signal source – but never to anything else.

The **Release** output changes to 0 if the **Top** input switches from 0 to 1.

WARNING
Incorrect use of the **Top** input

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 **Top** input for safety purposes, e.g. to trigger an emergency stop.

Mode for upstroke muting

If the **max. time for upstroke muting** is set to a value other than 0, the **Upstroke** input must be connected.

NOTE
The **Upstroke** input is usually connected to the **Upstroke** output of a Universal press contact or Eccentric press contact function block.

WARNING
Restricted safety with upstroke muting

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

► Eliminate all possible dangers that could arise during the press upstroke.

This results in the **Release 3 (safety)** and **Start/Release** input signals being bypassed (muting of the **Start/Release** input is dependent on the parameter settings) when the **Release** output is set to 1 and the **Upstroke** input is also set to 1. The Press single stroke function block does not carry out a plausibility check on the **Upstroke** input signal. If the **Upstroke** input switches to 1 several times during the same press cycle, the relevant input of the function block can be bypassed (muted) several times as well.

If you do not want a signal to be bypassed (muted), it should be connected to the **Release 1 (static)** input along with all the other signals that need to be connected to the **Release 1 (static)** input using an AND function block.

Max. time for upstroke muting

The **max. time for upstroke muting** is configurable. This time starts running when a rising signal edge (0–1) occurs at the **Upstroke** input. If the timer reaches the value configured for **max. time for upstroke muting** before a falling signal edge (1–0) occurs at the **Upstroke** input, the function block terminates muting of the **Release 3 (safety)** and **Start/Release** inputs. If either of these two inputs switches to 0 as of this point, the **Release** output is likewise set to 0.

Restart input

If the **Restart interlock condition** configuration parameter has been set to Never, no **Restart** signal is required to restart the press following a stop. The **Restart interlock condition** parameter can also be set to the following values:

- When Release 1 or Release 3 or Start/Release is 0
- When Release 1 or Release 3 is 0 or Top changes to 1
- Always
- When Release 1 or Release 3 is 0

This parameter determines when a **Restart** signal is expected as an input signal for the function block.

If the **Release** output switches to 0 because the specified **Restart interlock condition** configuration parameter has been set, the **Release** output can only be reset once a valid **restart** sequence has been completed with a 0–1–0 transition (at least 100 ms or 350 ms; shorter pulses and pulses lasting longer than 30 s will be ignored).

Ignore Release 3 (safety) for restart interlock while in top position

If the **Ignore Release 3 (safety) for restart interlock while in top position** parameter is configured to Yes, the restart interlock is not activated when the **Release 3 (safety)** input switches to 0 during a regular press stop operation.

That is, if the **Release** output switches to 0 because the **Top** input switched to 1, and then the **Release 3 (safety)** input switches to 0, the **Restart required** output does not switch to 1 unless the press has been restarted.

Function block output signals

The Press single stroke function block supports the following output signals:

Restart required

The **Restart required** output is set to 1 if a valid **restart** sequence is expected at the **Restart** input.

Sequence/timing diagrams

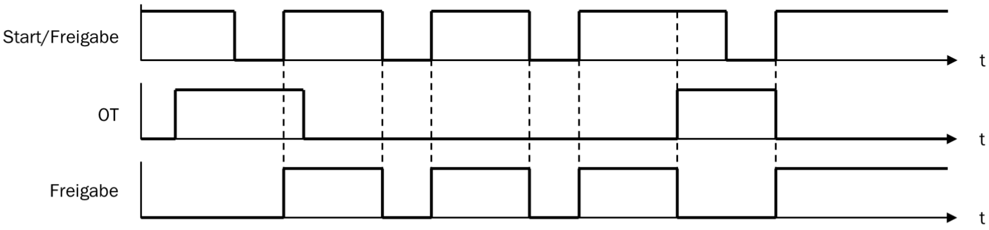


Figure 181: Sequence/timing diagram for the Press single stroke function block when Start/ Release is set to Start and static release (jog mode)

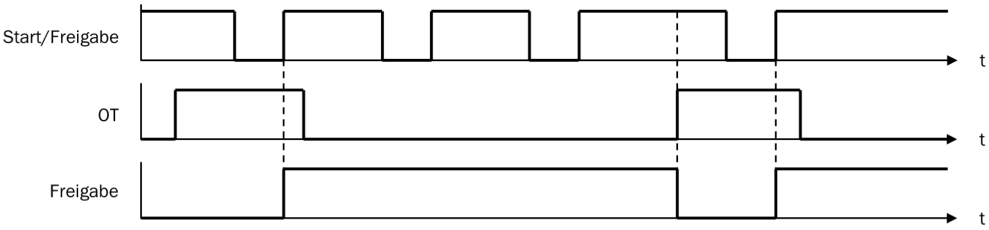


Figure 182: Sequence/timing diagram for the Press single stroke function block when Start/ Release is set to Start only (stopping not possible)

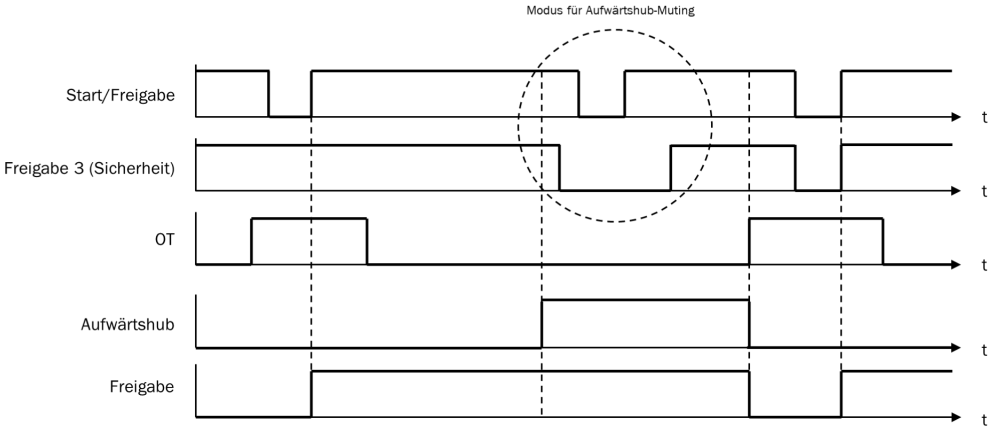


Figure 183: Sequence/timing diagram for the Press single stroke function block with upstroke muting applied to Start/Release and Release 3 (safety)

7.11.3 Press automatic

Function block diagram

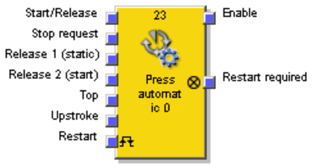


Figure 184: Inputs and outputs of the Press automated function block

General description

The Press automatic function block is used in conjunction with press applications where the movements of the workpieces toward and away from the press are automated and where occasional access to the press is required, e.g., for tool changes.

For this purpose, the function block can generate a stop signal for the press (i.e., the **Release** output switches to 0) when it reaches a position that makes it easy to change the tool (e.g., in the top position), but only if a stop operation has been requested first.

Function block parameters

Table 110: Parameters of the Press automatic function block

Parameter	Possible values
Restart interlock condition	• After every stop • Never
Stop request condition	• When Start/Release input is 0 • When Stop input is 1
Upstroke input	• With • Without
Release 2 (start) input	• With • Without
Min. restart pulse time	• 100 ms • 350 ms

NOTE

If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal for the **Restart** function may produce a pulse if it is reset as a result of short-circuit detection.

WARNING

Undesired restart following short-circuit to high

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

- ► Make sure that the transitions of the signals for the **Restart** function meet the requirements of safety standards and regulations.
  - ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines).
  - ► No short-circuit detection, i.e., no referencing to test outputs.

Function block input parameters and input signals

The Press automatic function block supports the following input signals:

Stop request condition

The **Stop requirement condition** parameter determines the stop mode for the Press automatic function block. When this parameter is set to When Start/Release input is 0, the **Start/Release** input is used to control the **Release** output directly. When it is set to When Stop input is 1, the **Release** output switches to 0 when the **Stop requirement** input is 1.

In both cases, the **Release** output switches to 1 when all of the following conditions are met:

- A transition from 0 to 1 occurs at the **Start/Release** input.
- The **Stop requirement** input is 0 (if connected).
- There is no other factor present that would normally trigger a stop signal, e.g., **Release 1 (static)** is 0.

Upstroke input

If the **Upstroke input** parameter is set to With, a 1 signal at the **Upstroke** input makes it possible to stop the press during the downward movement as well as when it is in the top position. If this parameter is set to Without, regular stop operations are only possible when the press is in the top position.

NOTE

The **Upstroke** input is usually connected to the **Upstroke** output of a Universal press contact or Eccentric press contact function block.

Start/Release

WARNING

Incorrect use of the **Start/release** input

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 **Start/release** input for safety purposes, but rather only to trigger stop requests from the automation control.
- ► Only connect signals for initiating a safety stop (e.g., emergency stop) to the **Release 1 (static)** input of the function block.

The **Start/Release** input signal is used to signal the beginning and the end of the press movement. If a rising signal edge (0–1) is detected at the **Start/Release** input, the **Release** output switches to 1 provided that the **Stop requirement** input is 0 and there is no other factor present that would normally trigger a stop signal, e.g., **Release 1 (static)** is 0. A valid restart sequence may be required prior to the transition of the **Start/Release** signal if the **Restart interlock condition** parameter has been set to After every stop. If control switches (e.g., a two-hand control device) are connected to the **Start/Release** input, you must make sure that a restart cannot be triggered accidentally.

Stop request

WARNING
Incorrect use of the **Stop request** input

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 **Stop request** input for safety purposes, but rather only to trigger stop requests from the automation control.
► Only connect signals for initiating a safety stop (e.g., emergency stop) to the **Release 1 (static)** input of the function block.

When the **Stop requirement condition** parameter is set to When Stop input is 1, the **Stop requirement** input is used to send a stop signal to the press. When the **Stop requirement** input is 1, the **Release** output is set to 0.

This input should only be used if the **Stop requirement condition** parameter has been set to When Stop input is 1. The **Stop requirement** input is not used when the **Stop requirement condition** parameter is set to When Start/Release input is 0. A valid restart sequence may be required prior to the transition of the **Start/Release** signal if the **Restart interlock condition** parameter has been set to After every stop. The **Stop requirement** input is intended for non-safety-related signals (e.g., for connecting signals from a programmable logic controller (PLC)). Safety-related signals may only be connected to the **Release 1 (static)** input and not to the **Stop requirement** input.

Release 1 (static)

The **Release 1 (static)** input signal is mandatory. The **Release** output always switches to 0 as soon as **Release 1 (static)** is set to 0.

If the Press automatic function block is used together with a press contact function block (e.g., Eccentric press contact or Universal press contact), its **Release** output must be connected to the **Release 1 (static)** input of the Press automatic function block.

Release 2 (start)

WARNING
Incorrect use of the **Release 2 (start)** input

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 **Release 2 (start)** input for safety purposes, e.g. to trigger an emergency stop.

The **Release 2 (start)** input is optional. If **Release 2 (start)** is used, the **Release** output can only switch to 1 (e.g., during switch-on) if **Release 2 (start)** is set to 1. When the **Release** output is set to 1, **Release 2 (start)** stops being monitored.

Top

The **Top** input is used to determine the end of the press cycle (i.e., when the press has reached the top dead center position). The **Top** input must only be connected to a **Top** output of a Universal press contact or Eccentric press contact function block, or to an equivalent signal source – but never to anything else.

WARNING

Incorrect use of the **Top** input

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 **Top** input for safety purposes, e.g. to trigger an emergency stop.

Restart

If the **Restart interlock condition** configuration parameter is set to Never, no **Restart** signal is required to restart the press following a stop.

If the **Restart interlock condition** is set to After every stop and the **Release** output switches to 0, the **Release** output can only be reset once a valid restart sequence has been completed with a 0–1–0 transition (at least 100 ms or 350 ms; shorter pulses and pulses lasting longer than 30 s will be ignored).

Function block output signals

The Press automatic function block supports the following output signals:

Restart required

The **Restart required** output is set to 1 if a valid **restart** sequence is expected at the **Restart** input.

Sequence/timing diagram

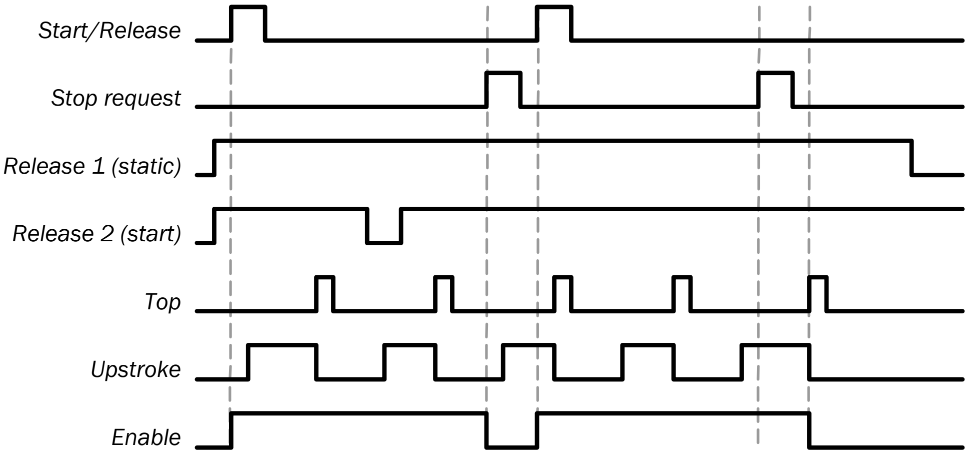


Figure 185: Sequence/timing diagram for the Press automatic function block with the Stop request and Upstroke inputs

7.11.4 PSDI mode

Function block diagram

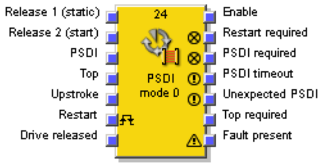


Figure 186: Inputs and outputs of the PSDI mode function block

General description

The PSDI mode function block is used for press applications that rely on PSDI mode (= Presence Sensing Device Initiation).

The requirements for PSDI mode are described in local, regional, national, and international standards. PSDI mode applications must always be implemented in accordance with these standards and regulations as well as in accordance with the relevant risk analysis and risk mitigation strategy to ensure the safety of the application.

WARNING
Restricted safety with PSDI mode

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

- ► Follow the safety regulations for PSDI mode.
- ► Observe the following notes about how to use PSDI mode correctly:
  - If the electro-sensitive protective device (e.g., safety light curtain) is not used in more than one of the operating modes that has been set up, the electro-sensitive protective device must be deactivated in this operating mode to make it clear that protective operation is not currently active for this device.
  - If an application involves using more than one electro-sensitive protective device (e.g., safety light curtain) that is reliant on PSDI functions, only one of these electro-sensitive protective devices may be used to meet the requirements for PSDI mode.
  - In accordance with EN 692 and EN 693, which cover press applications, the number of interventions is limited to 1 or 2. Other applications are dependent on the applicable standards.
  - With PSDI mode, people must not be able to enter, pass through, and leave the protective field of an electro-sensitive protective device. Consequently, press systems that are configured to permit this are not allowed.
- ► Prevent access to dangerous movements.

The PSDI mode function block defines a specific sequence of events that trigger a press cycle. These are called “interruptions” and may be defined as a transition from 1 to 0 to 1 by the **PSDI** input signal. When a press is in PSDI mode, a press cycle is triggered manually using an indirect process that is based on a predefined number of interruptions in the electro-sensitive protective device. Once the electro-sensitive protective device (e.g., a safety light curtain) detects that the operator has finished performing the movements required to insert or remove parts and that the operator has withdrawn all parts of his or her body from the protective field of the electro-sensitive protective device, automated triggering of the press is allowed.

The PSDI mode function block can be used in conjunction with the Universal press contact or Press single stroke function blocks and an input for a safety light curtain. The **Release** output of this function block can be used to control the **Start**/**Release** input of a Press single stroke function block, for example.

The PSDI mode function block checks whether the start sequence is valid and when the intervention counter or function block must be reset.

Function block parameters

Table 111: Parameters of the PSDI mode function block

Parameter	Possible values
Number of PSDI pulses	1 to 8
Mode	• Standard • Sweden
Max. time for upstroke muting	0 = infinite, 1 to 7,200 s. The Upstroke input is only present when the value is set to a value other than 0.
Max. time for PSDI pulses (timeout)	0 = infinite, 1 to 500 s
Condition for Release 2 (start) input	• Without • Necessary for first start • Necessary for every start
Start of first PSDI pulse (PSDI input 0 -> 1)	• After TDC has been reached • After the start of upstroke
Restart interlock	• For all stops • For stops during downstroke and at TDC (ignored during upstroke) • Without
Min. restart pulse time	• 100 ms • 350 ms
Valid start position (for restart and PSDI pulses)	• Everywhere • TDC only
Min. PSDI pulse time (0 time)	• 100 ms • 350 ms
Use Fault present	• With • Without

NOTE

If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal for the **Restart** function may produce a pulse if it is reset as a result of short-circuit detection.

WARNING

Undesired restart following short-circuit to high

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

► Make sure that the transitions of the signals for the **Restart** function meet the requirements of safety standards and regulations.
  ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines).
  ► No short-circuit detection, i.e., no referencing to test outputs.

Function block input parameters and input signals

The PSDI mode function block supports the following input signals:

Mode (Standard or Sweden)

The **Mode** parameter defines the full start sequence for the PSDI mode function block. In Standard mode, the number of electro-sensitive protective device interventions defined by the **Number of PSDI pulses** parameter must be completed first and then a valid **Restart** sequence must follow.

In Sweden mode, a valid **Restart** sequence is required first and then the configured number of interventions must follow.

Requirements for the start sequence

If the **Release** output switches to 0 due to one of the conditions listed below, a full start sequence may be required:

- **Release 1 (static)** is 0
- **Unexpected PSDI** output is 1 while PSDI = 0 and there is no active upstroke muting and no stop at TDC
- In the event of a PSDI timeout
- After **Drive released** input has switched to 1

If the **Unexpected PSDI** output is 1, the **Release** output and the **PSDI** input are both 0, and the **Restart interlock** parameter is set to Without, a restart is possible without the need for a full **restart** sequence. This may also apply during a press upstroke when the **Restart interlock** parameter is set to For stops during downstroke and at TDC (ignored during upstroke).

The minimum interruption time at the **PSDI** input is 100 ms or 350 ms. Shorter interruptions are not classed as valid, i.e., they are ignored. If the **Condition for Release 2 (start) input** parameter is set to Necessary for first start or Necessary for every start, the **Release 2 (start)** input must likewise be 1 when a full start sequence is required.

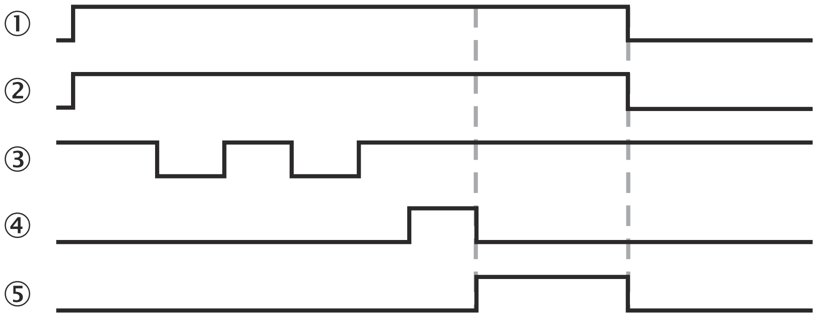


Figure 187: Sequence/timing diagram for a full start sequence in Standard mode with 2 PSDIs

① Release 1 (static)

② Release 2 (start)

③ PSDI

④ Restart

⑤ Enable

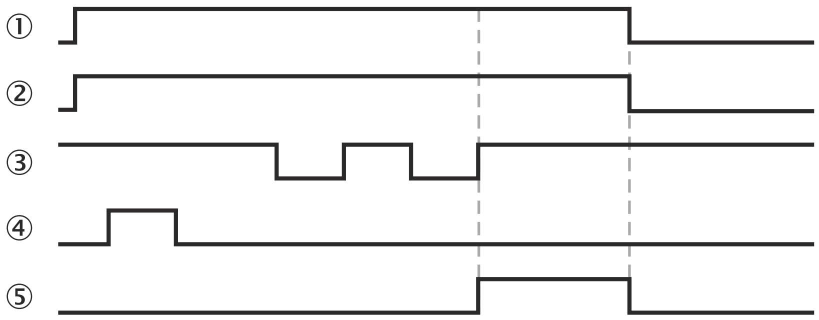


Figure 188: Sequence/timing diagram for a full start sequence in Sweden mode with 2 PSDIs

① Release 1 (static)
② Release 2 (start)
③ PSDI mode
④ Restart
⑤ Enable

Once the initial full start sequence has finished and the press has completed a press cycle, the **Top** input must indicate that the press is in the TDC position. This is indicated by a rising signal edge (0–1) at the **Top** input. When this happens, the internal interruption counter is reset.

For a subsequent cycle to be triggered, a cycle start sequence is required. In this case, the **Release** output switches to 1 following the configured number of interruptions if the other configured conditions are also met (e.g., **Condition for Release 2 (start) input** may be set to Necessary for every start).

Max. time for PSDI pulses (timeout)

The **Max. time for PSDI pulses (timeout)** parameter defines the time required for a full start sequence and also for a cycle start sequence. If the **max. time for PSDI pulses (timeout)** is exceeded, the **PSDI timeout** output switches to 1. In this case, a full start sequence is required in order for the **Release** output to switch back to 1 (e.g., to start a press). The timer starts running when the press is stopped at TDC (i.e., when the **Top** input switches from 0 to 1) and once all the other stop conditions apply.

The default setting for the **max. time for PSDI pulses (timeout)** is 30 s in accordance with the maximum PSDI time permitted for eccentric presses (defined in EN 692). Setting the **max. time for PSDI pulses (timeout)** to 0 disables PSDI time monitoring.

Start of first PSDI pulse (PSDI input 0 -> 1)

The **Start of first PSDI pulse** parameter determines the circumstances under which an intervention is classed as valid.

If the **Start of first PSDI pulse** parameter is set to After the start of upstroke, the interruption is classed as valid if it begins (i.e., a falling signal edge (1–0) at the **PSDI** input) after the rising signal edge occurs at the **Upstroke** input. It does not matter whether the **Top** input has already switched to 1.

If the **Start of first PSDI pulse** parameter is set to After TDC has been reached, the interruption is only classed as valid if it begins (i.e., a falling signal edge (1–0) at the **PSDI** input) after the rising signal edge occurs at the **Top** input.

In both cases, the interruption must end (i.e., a rising signal edge (0–1) at the **PSDI** input) after the rising signal edge occurs at the **Top** input. It does not matter whether the **Top** input is still set to 1 or whether it has already switched back to 0.

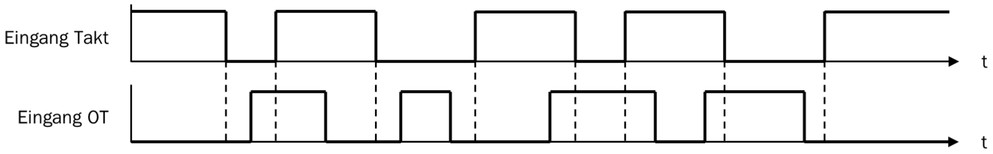


Figure 189: Valid interventions when the Start of first PSDI pulse parameter is set to After the start of upstroke

NOTE

If the **Start of first PSDI pulse** parameter is set to After the start of upstroke, upstroke muting must be activated. Otherwise, the **Release** output will change to 0 as soon as the **PSDI** input switches to 0 (i.e., at the beginning of the interruption).

Upstroke muting and maximum time for upstroke muting

Upstroke muting can be used to bypass the **PSDI** input (e.g., the OSSDs of a safety light curtain) during the upstroke of the press cycle. Upstroke muting is activated when the **Max. time for upstroke muting** parameter is set to a value greater than 0. Upstroke muting is deactivated when the **Max. time for upstroke muting** parameter is set to 0.

WARNING

Restricted safety with upstroke muting

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

► Eliminate all possible dangers that could arise during the press upstroke.

If upstroke muting is activated, the following points must be observed:

- It is vital to ensure that the **Upstroke** input is connected to a suitable signal. This could be the **Upstroke** output of the Eccentric press contact or the Universal press contact function block, for example.
- The **PSDI** input of the function block is bypassed (muted), if the **Upstroke** input is 1 and the **Top** input remains set to 0.

The function block does not check the **Upstroke** input for plausibility. This means that the **PSDI** input can be bypassed (muted) several times if the **Upstroke** input is activated multiple times within the same press cycle.

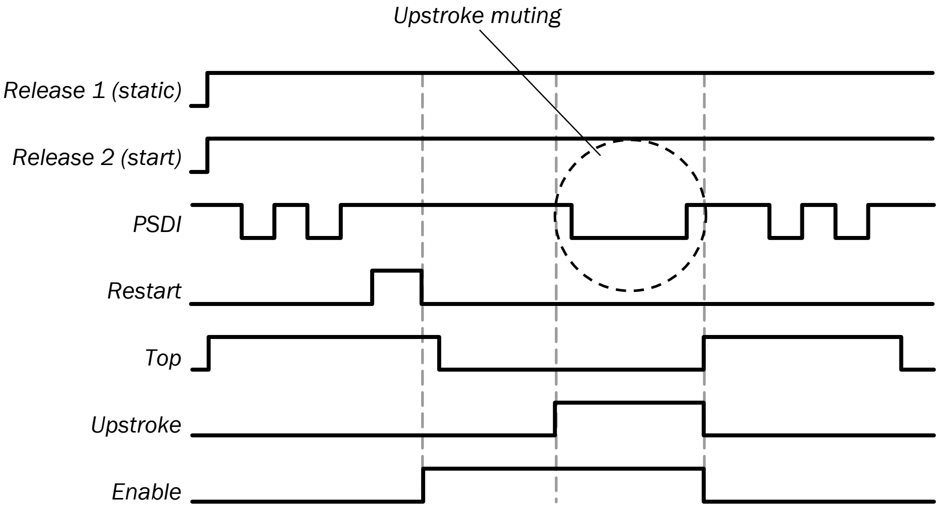


Figure 190: Sequence/timing diagram for upstroke muting in Standard mode with 2 PSDIs

The **max. time for upstroke muting** is configurable. The timer for upstroke muting starts running when a rising signal edge (0–1) occurs at the **Upstroke** input. If the timer reaches the value configured for **max. time for upstroke muting** before another rising signal edge occurs at the **Upstroke** input, upstroke muting is interrupted. When the **PSDI** input is 0 in this case, the **Release** output is also set to 0. If a second rising signal edge then occurs at the **Upstroke** input, upstroke muting starts again.

If the **Release** output switches to 0 – because either the **Release 1 (static)** input or the **PSDI** input has changed to 0 – the **Top required** diagnostic output switches to 1. A press restart operation is prevented from taking place until the **Top** input has switched back to 1 without a restart having been performed in any other operating mode.

Valid start position (for restart and PSDI pulses)

If the **Valid start position (for restart and PSDI pulses)** parameter is set to TDC only, the press can only be restarted when it is in the top dead center position. If it is in any other position, a restart operation is prevented from taking place. If, for example, the press has been stopped partway through the downward movement due to an interruption in the protective field of the light curtain, you must switch to another operating mode (e.g., in conjunction with the Press setup function block) so that the press can be brought back to the top dead center position. This is necessary because the PSDI mode function block does not allow a restart to be performed with the TDC only parameter setting.

If the **Valid start position (for restart and PSDI pulses)** parameter is set to TDC only, the optional **Drive released** input must be connected to determine whether the press is running or has been brought to a stop. This must be the same signal that controls the press directly. Usually, a jump address or a CPU marker is used to link the **Drive released** input to the logic editor output signal that is connected to the physical output for the press.

NOTE

Do not connect any physical input signals to the **Drive released** input. The signal that controls the physical output for the press drive must be controlled with a jump address or a CPU marker.

- When using a jump address, this signal must form a logical loopback. To achieve this, connect the outputs of this function block to the inputs of the subsequent function blocks. After that, connect the jump address to the **Drive released** input. This is particularly important if all connections to the subsequent function blocks are established using jump addresses.
- If using a CPU marker, you must use a routing function block to route the signal not only to the physical output for the press drive but also to the output of the CPU marker.

Release 1 (static) input

The **Release 1 (static)** input signal is mandatory. The **Release** output always switches to 0 as soon as **Release 1 (static)** is set to 0.

If the PSDI mode function block is used together with a press contact function block (e.g., Eccentric press contact or Universal press contact), its **Release** output must be connected to the **Release 1 (static)** input of the PSDI mode function block.

Release 2 (start)

WARNING
Incorrect use of the **Release 2 (start)** input

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 **Release 2 (start)** input for safety purposes, e.g. to trigger an emergency stop.

The **Release 2 (start)** input is optional. If **Release 2 (start)** is used, the **Release** output can only switch to 1 (e.g., during switch-on) if **Release 2 (start)** is set to 1. When the **Release** output is set to 1, **Release 2 (start)** stops being monitored.

Top

WARNING
Incorrect use of the **Top** input

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 **Top** input for safety purposes, e.g. to trigger an emergency stop.

The **Top** input is used to determine the end of the press cycle (i.e., when the press has reached the top dead center position). The **Top** input must only be connected to a **Top** output of a Universal press contact or Eccentric press contact function block, or to an equivalent signal source – but never to anything else.

Upstroke input

When upstroke muting is active (i.e., if the **max. time for upstroke muting** is greater than 0), the **PSDI** input of the function block is bypassed (muted), if the **Upstroke** input is 1 and the **Top** input remains set to 0.

NOTE
The **Upstroke** input is usually connected to the **Upstroke** output of a Universal press contact or Eccentric press contact function block.

Restart input

If the **Restart interlock** configuration parameter is set to Without, no **Restart** signal is required to restart the press after the **Release** output has switched to 0.

If the **Restart interlock** parameter has been set to For all stops and the **Release** output switches to 0, the **Release** output can only be reset once a valid **Restart** sequence has been completed with a 0–1–0 transition (at least 100 ms or 350 ms; shorter pulses and pulses lasting longer than 30 s will be ignored). The only exception to this rule is the start of a cycle. In this case, the **Restart interlock** parameter has no effect on the function block.

If the **Restart interlock** parameter has been set to For all stops and the **max. time for upstroke muting** has been set to 0 s, a 0 signal at the **PSDI** input during the upstroke will immediately set the **Release** output to 0.

If the **Restart interlock** parameter has been set to For all stops and upstroke muting is active, the **Release** output remains set to 1 until the **Top** input switches to 1 (thereby indicating that the press cycle is complete). In this case, a full **restart** sequence is required.

If the **Restart interlock** parameter has been set to For stops during downstroke and at TDC (ignored during upstroke) and the **Upstroke** input is 1, the **Release** output remains set to 1 until the **Top**input switches to 1 (thereby indicating that the press cycle is complete). In this case, a cycle start sequence is required.

If the **PSDI** input switches from 1 to 0 and then back to 1 at the end of the **max. time for upstroke muting**, the **Release** output then does the same by switching from 1 to 0 and then back to 1. The setting of this parameter has no effect if the **Restart** and **Upstroke** inputs are left unconnected.

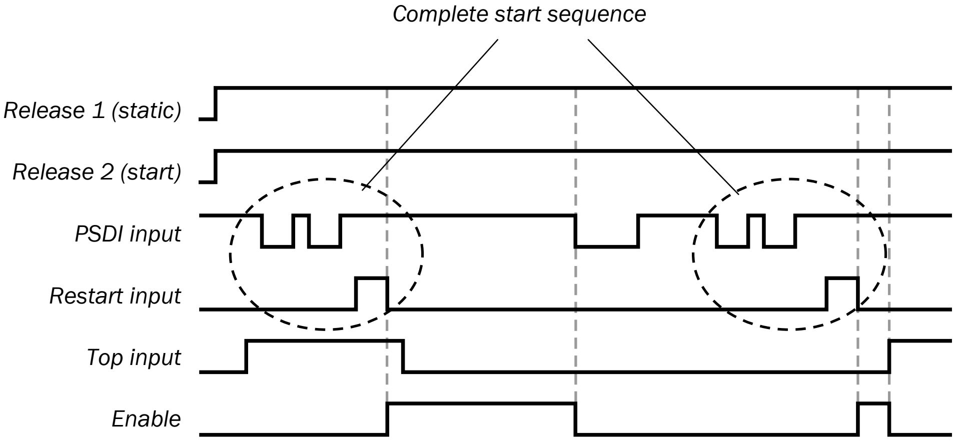


Figure 191: Sequence/timing diagram when the PSDI input is 0, upstroke muting is disabled and the Restart interlock parameter is set to “For all stops”

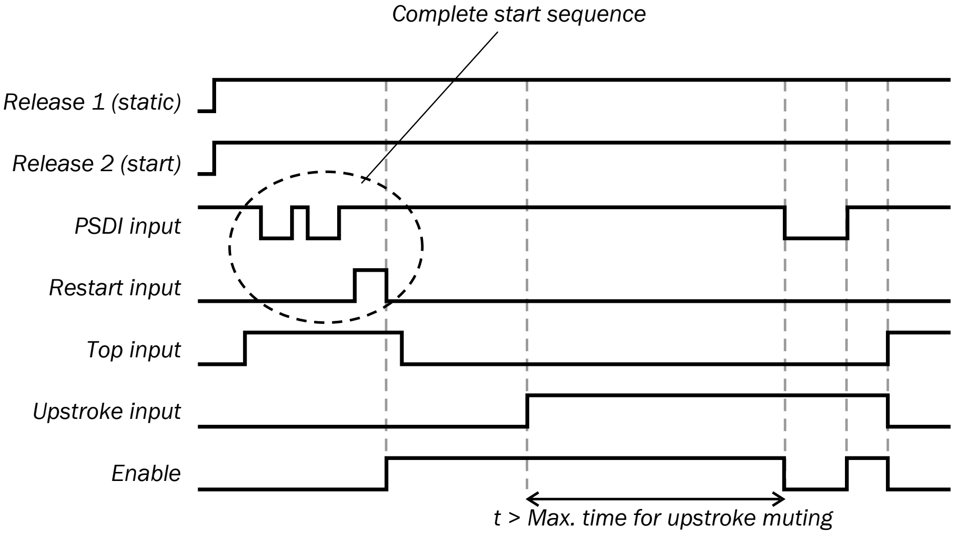


Figure 192: Sequence/timing diagram when the PSDI input is 0, the max. time for upstroke muting is > 0, and the Restart interlock parameter is set to “For stops during downstroke and at TDC (ignored during upstroke)”

Function block output signals

The PSDI mode function block supports the following output signals:

Restart required output

The **Restart required** output is set to 1 if a valid **restart** sequence is expected at the **Restart** input.

PSDI required output

The **PSDI required** output is set to 1 if an interruption is expected at the **PSDI** input.

Unexpected PSDI output

The **Unexpected PSDI** output is set to 1 if a valid start sequence has been performed and the **PSDI** input switches from 1 to 0 while muting is deactivated and no interruption is expected. If **Unexpected PSDI** is 1, a valid restart sequence must usually be performed before the **Release** output can switch back to 1.

If the **Unexpected PSDI** output is 1, the **Release** output and the **PSDI** input are both 0, and the **Restart interlock** parameter is set to Without, a restart is possible without the need for a full **restart** sequence. This may also apply during a press upstroke when the **Restart interlock** parameter is set to For stops during downstroke and at TDC (ignored during upstroke).

Error statuses and reset information

Table 112: Error statuses and reset information for the PSDI mode function block

Diagnostic outputs	Resetting the error status	Comments
Unexpected PSDI	In the case of **Unexpected PSDI**, the **PSDI** input must usually switch back to 1 and a valid **restart** sequence must then follow in order for the error to be reset. If the **Unexpected PSDI** output is 1, the **Release** output and the **PSDI** input are both 0, and the **Restart interlock** parameter is set to Without or For stops during downstroke and at TDC (ignored during upstroke), a restart is possible without the need for a full **restart** sequence. In the event of a PSDI timeout, the error is reset by a valid restart sequence.	The **Release** output switches to 0 and the **Fault present** output switches to 1 when the **Unexpected PSDI** diagnostic output or **PSDI timeout** is set to 1.
PSDI timeout		

7.12 Grouped function blocks

Important information

NOTE
When calculating the total number of function blocks in a project, a grouped function block is not counted as a single block. Instead, the number of function blocks within it are counted.

Grouped function blocks

Function blocks can be grouped. This makes it easier to use the same logic groups multiple times and it reduces the number of function blocks displayed on the workspace.

A grouped function block has the following characteristics:

- It can have a maximum of 32 inputs and 32 outputs.
- It must not contain the Fast shut off function block.
- It must not contain another grouped function block.
- It must not contain a customized function block.
- It can be saved as a customized function block and can therefore be used in other projects.
- It has its own checksum in the CPU logic.
- It has its public parameters in the CPU logic.

Complementary information

Grouped function blocks of the CPU logic are verified when the Flexi Soft system is verified. The report shows the grouped function blocks together with the checksum and the verification status.

Further topics

- "Configuration checksums", page 496
- "Verifying configuration", page 494

7.12.1 Grouping function blocks

Procedure

1. Drag the function blocks to be grouped onto the logic editor workspace, configure them as required, and connect them to one another.
2. Select all function blocks to be grouped (e.g. by clicking while pressing the Ctrl pushbutton).
3. Open the context menu of one of the selected function blocks and click on **Group ...**.
4. Enter a name for the grouped function block.
5. To assign an icon to the grouped function block, click on **Selection...**, choose your desired icon, and confirm with **OK**.
6. Click on **OK** to confirm the changes and close the dialog box. The selected function blocks are grouped. The grouped function block looks like an individual function block on the workspace.

Complementary information

The content of the grouped function block is displayed on a separate logic page with an orange background and can be edited there.

On the **Group info inputs** and **Group info outputs** tab, you can change the name and the icon for the grouped function block.

7.12.2 Adding inputs and outputs

Procedure

1. Switch to the logic page of the grouped function block.
2. Drag inputs and outputs (can be found under **Group info inputs** and **Group info outputs**) onto the workspace and connect as required.
3. Double-click on an input or an output to edit its tag name.
   The inputs and outputs that have been added are displayed as inputs and outputs of the grouped function block in the main program of the logic. Devices, inputs and outputs of function blocks, jump addresses, etc., can be connected to these. If a device, etc., has been connected to an input or output of the grouped function block, then you can also see if the external view is activated in the logic of the grouped function block.
4. Click on **Switch view** in the toolbar to switch between the internal and external view.
   - The internal view shows the tag names for the inputs and outputs of the grouped function block.
   - The external view shows what the inputs and outputs of the grouped function block are connected to in the logic program.

7.12.3 Adding public parameters

Overview

Public parameters are accessible outside the logic page of a grouped function block (GFB) or a user-defined function block (CFB). For example, you can use public parameters to edit protected function blocks via the general logic page.

Procedure

Adding public parameters

1. Switch to the logic page of the grouped function block.

2. Right-click on a function block.
3. Select **Adding public parameters**.
✓ The parameter list opens.
4. Select a parameter.
✓ The parameter is added at the bottom left in the **Function block group info** tab.

Editing public parameters

► Right-click on a function block to edit the public parameter.

Complementary information

Public parameters are available for the clock generator and counter (up, down, up and down) function blocks.

7.13 Customized function blocks

Important information

! **NOTICE**
Loss of password

The password cannot be reset or recovered, even by the SICK service.

► Make a note of the password and keep it safe.

i **NOTE**
When calculating the total number of function blocks in a project, a customized function block is not counted as a single function block. Instead, the number of function blocks within it are counted.

Customized function blocks

Customized function blocks are created from grouped function blocks.

Customized function blocks have their own checksum in the CPU logic.

They differ from these in the following ways:

- Customized function blocks are displayed in the selection list for the function blocks. Like other function blocks, they can also be used multiple times in a project.
- It is not possible to edit customized function blocks.
- Customized function blocks are available in all projects on the computer on which they were created.
- Customized function blocks can be transferred to other computers.
- Customized function blocks can be protected with a password. A customized function block with password protection can be used in the logic editor as normal. However, the configuration for this function block cannot be viewed unless you have the password.

Complementary information

Customized function blocks of the CPU logic are verified when the Flexi Soft system is verified. The report shows the grouped function blocks together with the checksum and the verification status.

If the configuration contains customized function blocks protected with an unknown password, they have to be checked using the checksum when the Flexi Soft system is verified.

Further topics

- "Configuration checksums", page 496

7.13.1 Creating customized function blocks

Procedure

1. Create a grouped function block with the desired functionality.
2. On the logic page of the grouped function block, click on the **Save as CFB** button.
3. Enter a name for the customize function block.
4. Optionally, assign an icon to the customized function block.
   - ► Click on **Selection ...** to choose an icon from the library.

   Or:
   - ► Click on **Browse ...** to use a graphic which you created yourself.
5. Optionally, activate password protection and enter a password.
6. Click on **OK** to confirm the changes and close the dialog box.

✓ The grouped function block is converted into a customized function block. This is available immediately in the selection list of the function blocks and can be used in all projects on the same computer.

Complementary information

It is not possible to overwrite customized function blocks which are already available.

7.13.2 Changing customized function blocks

Overview

Unlike grouped function blocks, you cannot edit customized function blocks. To change a customized function block, you must convert it back into a grouped function block again beforehand.

Procedure

1. Double-click on the customized function block and enter the password if necessary to open the logic page of the customized function block.
2. Click on **Edit ...** in the toolbar and confirm with **Yes**.

✓ The customized function block is converted back into a grouped function block, which can be edited as usual.

7.13.3 Transferring customized function blocks

Overview

You can transfer customized function blocks to another computer.

Procedure

1. Create and save a project, in which the desired customized function blocks are used.
2. Open the project on another computer. If the project contains customized function blocks which are not available on the computer, then you have two options.
   - ► Import the customized function blocks. These are then added to the function block selection list and are available for use in all projects on the same computer.
   - ► Do not import the customized function blocks. In this case, the project still opens. However, the customized function blocks included in this project are not added to the selection list of function blocks and they can only be used in the current project.

7.13.4 Deleting customized function blocks

Important information

NOTE

- This command cannot be undone.
- You can still use projects which contain the deleted customized function block. When you open a project containing the customized function blocks that have been deleted, the project is treated in the same way as one that has been transferred from another computer.

Procedure

1. Remove all occurrences of the customized block to be deleted from the project or convert into grouped function blocks.
2. In the drop-down menu of function blocks, open the context menu for the customized function block and select the **Delete customized function block** command.

7.14 Simulating the configuration

It is possible to simulate the programmed logic offline in the **logic editor**. Inputs can be set to 1 or 0 and you can observe the subsequent switching of the outputs. The respective timer and counter values for the function blocks used are displayed on the function blocks during simulation.

- Click the **Start simulation mode** () button on the toolbar to activate simulation mode. The background of the logic editor turns green and the simulation toolbar appears.

NOTE

Simulation mode can only be started if there is a valid configuration.

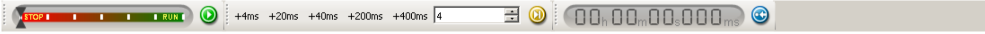


Figure 193: Simulation toolbar before a simulation is started

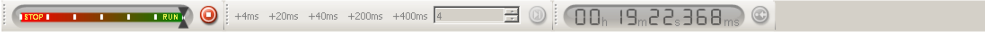


Figure 194: Simulation toolbar while a simulation is in progress

The green **Start** button starts the simulation process at full speed. [4] The timer indicates how much time has elapsed. To reset the timer, click on the blue **Reset** button. To stop the current simulation, click on the red **Stop** button.

[4] Real time or slower, depending on the performance of the computer.

Slowing down the simulation

If the logic processes are too fast to follow at real-time speed, there are two things you can do:

- Use the slider to slow down the simulation.
- Use the buttons on the right of the slider to perform the simulation step by step. The following time stages are available by default: +4 ms, +20 ms, +40 ms, +200 ms, and +400 ms. These values are adjusted automatically in accordance with the programmed logic because they represent multiples of the respective logic execution time. When you click one of these buttons, the simulation jumps forward by the relevant amount of time.
  In addition, the input field to the right of these buttons allows you to enter a customized time in milliseconds. When you click on the yellow button next to the input field, the simulation jumps forward by this amount of time. This feature can be used to shorten the waiting time before a timer elapses, for example.

NOTE

The time entered is rounded to the nearest logic execution time.

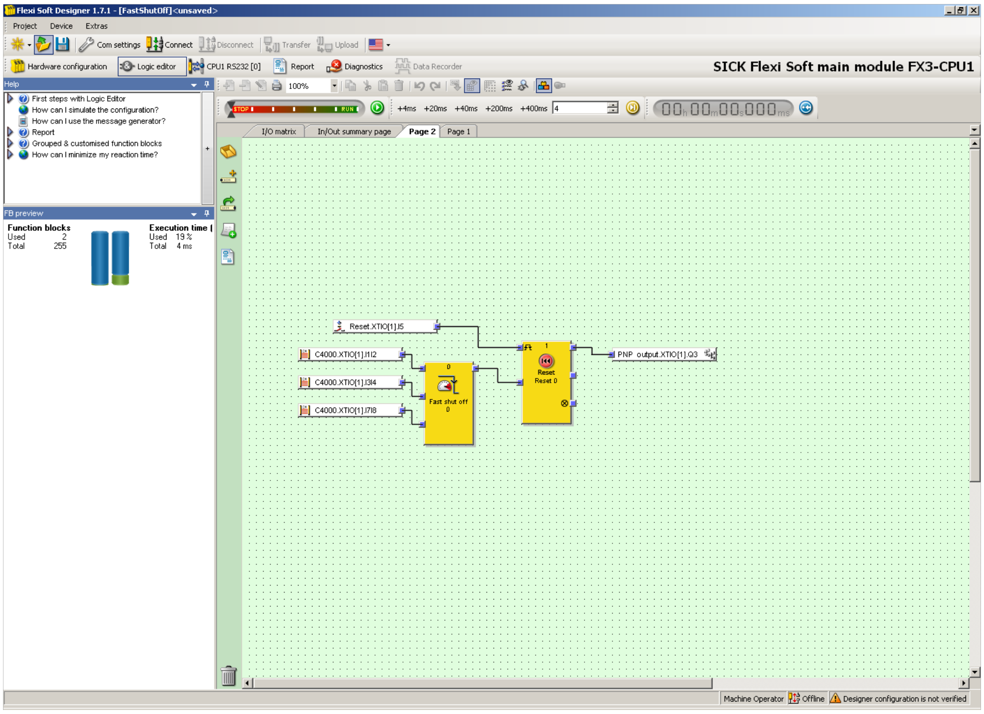


Figure 195: Simulation mode started, simulation stopped

During an ongoing simulation, inputs can be set to 1 by clicking on them once. When inputs are set to 1, they turn green and a blue frame appears around them. Clicking again resets the input to 0.

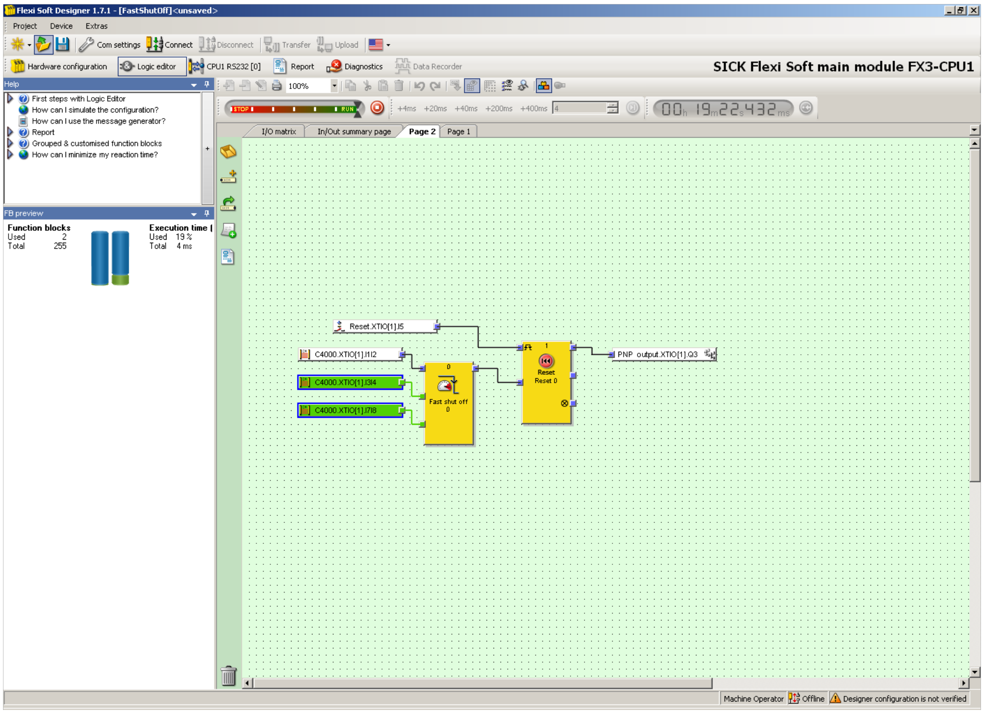


Figure 196: Simulation mode started, simulation running

If the simulation has been stopped, you can select inputs to make them switch at the next possible point in time. When you click on an input after the simulation has been stopped, a blue frame appears around the input concerned to indicate that it is going to switch during the next cycle of the simulation. This allows multiple elements to be switched simultaneously so that you can immediately see how this affects the logic.

Once the required inputs have been set, the simulation must be resumed in order for the logic and outputs to switch accordingly. To do this, either click the green **Start** button or one of the buttons for stage-by-stage execution.

NOTE

If you are using the **External device monitoring** or **Valve monitoring** function blocks, it is advisable to delete these from the logic before starting the simulation. These function blocks expect to receive a 1 signal at their feedback inputs within 300 ms of the associated outputs switching to 1. This cannot be simulated in real time. Instead, small time stages must be used.

7.15 Force mode

In force mode, the user can influence the logic program of the Flexi Soft system during operation. For this, the configuration software must be connected to the Flexi Soft system, which must be in the Run status.

In force mode, the Flexi Soft system inputs can be set to 1 or 0, regardless of the actual value of the physical inputs. In this case, the Flexi Soft system logic program behaves in exactly the same way as if the physical inputs had really assumed the respective values.

This allows you, for example, to test the wiring of the system in online mode during commissioning or maintenance, and to check whether the logic program is functioning correctly.

WARNING
Restricted safety in force mode

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

- ► Before you activate force mode, make sure that no one is in the hazardous area of the machine or system.
- ► Eliminate all possible hazards that could endanger people or objects while force mode is active.
- ► Make sure that no one is able to enter the hazardous area of the machine or system while force mode is active.
- ► If necessary, implement additional safety measures.
- ► Do not activate force mode from multiple computers at the same time.

NOTE

- Force mode only enables the user to directly influence the inputs in the logic of a Flexi Soft system. It does not allow the user to influence outputs and logic results, such as function blocks or jump addresses.
- The status of the outputs and logic results of the Flexi Soft system may change due to the forced input values. This may disable the protective function of the safety device.
- Force mode only affects function block inputs and the further processing of these. It is not possible to influence signals which are independent of the outputs of a function block. This applies to the values of inputs of I/O modules that are routed directly to a PLC using a gateway, for example.

Activating force mode

The following requirements must be met to be able to activiate force mode:

- The firmware version of the Flexi Soft main module must be ≥ V1.10.0.
- The user must be logged in to the system as an authorized client.
- The configuration of the Flexi Soft project must not have been verified (CV LED of the main module flashes ☀ **yellow** at 1 Hz).

NOTE

- To use force mode, the computer must be connected to the Flexi Soft system via the communication interface of the main module (RS-232, USB).
- If you attempt to activate force mode even though the configuration has been verified already (CV LED of the main module lights up ● **yellow**), a dialog box appears so that you can reset the status of the configuration to not verified.

- ► Establish a connection with the Flexi Soft system and start the application.
- ► Switch to the **Logic editor** view and click on the **Start force mode** ( ) button. The system opens a dialog box so that you can enter how much time should elapse before force mode is exited automatically if no further actions are triggered.
- ► Select the desired time and click **OK**. Force mode is started and the background color of the logic editor changes to red.

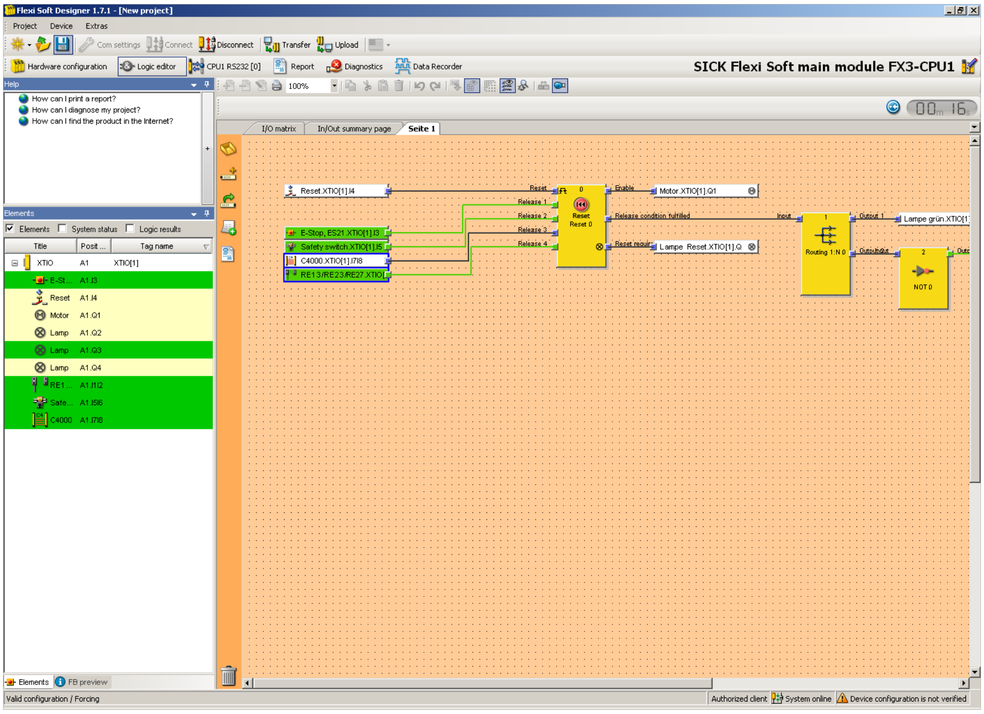


Figure 197: View of the logic editor with force mode activated

NOTE

While force mode is active, it is not possible to log out, to receive and compare a configuration, or to stop the device.

Switching inputs in force mode

► Click on an input. A menu appears with the following options:
  - **Force 0**: The input is evaluated as 0 in the logic program, regardless of its actual physical value in the Flexi Soft system.
  - **Force 1**: The input is evaluated as 1 in the logic program, regardless of its actual physical value in the Flexi Soft system.
  - **Deselect forcing**: The input is evaluated with its actual physical value in the Flexi Soft.

A forced input is identified by means of a dark blue border. An active input (1) is shown in green and a disabled input (0) in white. If there are any inputs whose forced value is different from their actual physical value, they are shown in light blue.

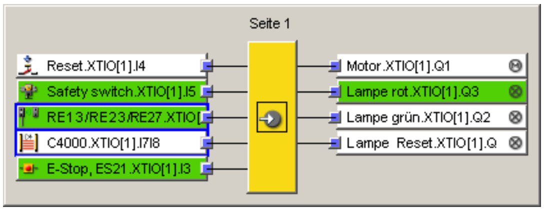


Figure 198: Forced and non-forced inputs

NOTE

- When an input is being forced in the logic, the real value of the physical input is not shown in the logic editor. Instead, it is shown in the **Hardware configuration** view only.
- Force mode only affects the inputs in the logic program, not the physical inputs of expansion modules.
  Examples:
  - Forcing does not affect the inputs of an FX3-XTIO module that are used for Fast shut off. This means that the output may remain set to 0 in the hardware even though the inputs in the logic have been forced to 1 because Fast shut off in the FX3-XTIO module is controlled directly by the physical inputs.
  - Forcing does not affect those inputs whose values are transmitted directly to a PLC via a gateway instead of being controlled by the logic program.
- Force mode always applies across the complete project. Where logic programs cover several pages in the logic editor, this means that a forced input is not just set to the same value on the page of the logic editor that is currently being displayed, but is set wherever it is used.
- If more than 16 outputs are switched simultaneously as a result of one input being forced in a logic program, the switching of some of these outputs will be delayed by the logic execution time (or a multiple thereof) due to the limited transmission rate of the RS-232 interface. The logic execution time is dependent on the size of the logic program. This is calculated automatically in the logic editor and displayed on the top right-hand side of the **FB group info** information window.
- Unlike in simulation mode, you can also use the **External device monitoring** or **Valve monitoring** function blocks when you are in force mode, provided that appropriate devices are actually connected and these send the necessary feedback signal when the outputs are activated.
- When using a Flexi Soft gateway, make sure that the process image of the Flexi Soft gateway always reflects the actual physical value of the inputs and outputs for the connected devices and not the (purely virtual) forced value of an input in the logic program. If the value of an output changes (e.g., 1–0) because an input has been forced in the logic program (e.g., 1–0), the physical value of the output in the process image (i.e., the one that has actually been changed; in the example: 0) is transmitted to the PLC. However, the forced 0 value of the input in the logic program does not get transmitted. Instead, the actual physical value of the input on the device (in the example: 1) still gets transmitted. This must be taken into account when transmitted data is being evaluated in the PLC.

Terminating force mode

Force mode can be terminated in the following ways:

- Manually by the user
- Automatically once the time defined at the start has elapsed
- Automatically after 30 seconds if the Flexi Soft system detects an error (e.g., if the connection to the computer is interrupted)

When force mode is terminated, all outputs of the Flexi Soft system are set to 0 and the active application is stopped.

⚠ **WARNING**
Restricted safety when ending force mode

In the case of non-compliance, it is possible that the dangerous state of the machine may not be stopped or not stopped in a timely manner.

- ► Before ending force mode, make sure that there is no risk of a dangerous situation arising.
- ► Make sure that the machine or system goes into a safe status when force mode is terminated and that it cannot sustain any damage.
- ► Before restarting the machine or system, make sure that it does not pose any risks.

- ► Click on the **Stop force mode** button. A safety prompt appears. Select **Yes** to confirm this and quit force mode or click **No** to remain in force mode.
  On expiry of the time defined at the start, force mode is terminated automatically if no action (e.g., Force input) is performed. In force mode, there is a timer on the top right-hand side to indicate how much time is left before force mode is terminated automatically. This timer is reset whenever an action is performed. The timer can also be reset by clicking on the **Trigger force mode** button on the left next to it. A dialog box appears 15 seconds before the timer runs out to tell you that force mode is about to be terminated.

- ► Click on **Cancel**. The dialog box closes and force mode is terminated on expiry of the set time.

Or:

- ► Click **OK** to close the dialog box, reset the timer, and remain in force mode.

Or:

- ► If you do not respond, force mode is terminated on expiry of the set time.

8 I/O modules

8.1 Dual-channel evaluation and discrepancy time monitoring

Dual-channel evaluation

The FX3-XTIO, FX3-XTDI, and FX3-XTDS safe I/O modules are capable of performing dual-channel evaluation if predefined input elements from the element window (e.g., RE27, C4000, ...) are connected to them. If an input element of this kind is selected, you do not need to have a separate function block for dual-channel evaluation (e.g., light grid monitoring, safety gate monitoring, or magnetic switch).

Dual-channel evaluation checks whether the sequence of the two input signals is correct. If either of the two signals has triggered a switch-off, the other signal is expected to follow accordingly. The value that the two signals are required to have is dependent on the type of dual-channel evaluation. There are two options:

- Equivalent evaluation
- Complementary evaluation

Discrepancy time

Dual-channel elements can be evaluated with or without a **Discrepancy time**. The discrepancy time defines how long the two inputs can continue to have discrepant values after the value of one of the two input signals changes without this being regarded as an error.

► Double-click the desired element to open the **Element settings** window.
► On the right-hand side of the **Element settings** window, click on either the check box or the 3D button to activate or deactivate the **discrepancy time**.

The following restrictions apply to elements that are connected to a FX3-XTIO, FX-3XTDI, or FX3-XTDS:

- The **value** of the discrepancy time can be set to 0 (= infinite) or to a value of between 4 ms and 30 s. Due to the internal evaluation frequency of the modules, this is automatically rounded up to the next multiple of 4 ms.
- If signals from tested sensors are connected to a FX3-XTIO, FX3-XTDI, or FX3-XTDS, the discrepancy time must exceed the **test gap (ms)** + the **max. OFF-ON delay (ms)** of the test output that is being used, as a signal change at the input of the modules may be delayed by this amount of time. You will find these values listed in the report under **Configuration**, **I/O module**, **Test pulse parameters**.

The following truth table describes the discrepancy conditions for dual-channel equivalent input evaluation and dual-channel complementary input evaluation:

Table 113: Dual-channel evaluation

Evaluation type	Input A (I1, I3, I5, I7)	Input B (I2, I4, I6, I8)	Discrepancy timer [1]	Status of dual-channel evaluation	Input of I/O module in logic editor	Discrepancy error
Equivalent	0	0	0	Deactivated	0	0
	0	1	< discrepancy time	Discrepant	0	Unchanged [2]
	1	0	< discrepancy time	Discrepant	0	Unchanged [2]
	1	1	0	Active [3]	1	0
	x	x	≥ discrepancy time (timeout)	Error	0	1

Evaluation type	Input A (I1, I3, I5, I7)	Input B (I2, I4, I6, I8)	Discrepancy timer [1]	Status of dual-channel evaluation	Input of I/O module in logic editor	Discrepancy error
Complementary	0	1	0	Deactivated	0	0
	0	0	< discrepancy time	Discrepant	0	Unchanged [2]
	1	1	< discrepancy time	Discrepant	0	Unchanged [2]
	1	0	0	Active [3]	1	0
	x	x	≥ discrepancy time (timeout)	Error	0	1

[1] If the discrepancy time is active (> 0), the discrepancy timer is restarted on the first signal change that leads to a discrepant status. If the discrepancy time is deactivated (= 0), the discrepancy timer is not started, i.e., a timeout never occurs.

[2] Unchanged = the last status is retained.

[3] If the correct sequence has been observed.

Sequence error

A dual-channel evaluation process can only switch to Active (input of the I/O module in the logic editor changes from 0 to 1) if all of the following conditions are met:

- The status has been set to Deactivated at least once since it was last Active. I.e., it is not possible to switch from Active to Discrepant and then back to Active.
- The discrepancy time has either not yet elapsed or is fully deactivated.

NOTE
If the correct sequence for achieving the Active status has not been observed (i.e., if the status has changed from Active to Discrepant and then straight back to Active), FX3-XTIO, FX3-XTDI, and FX3-XTDS modules with a firmware version ≥ V2.00.0 will indicate this sequence error within a maximum of 100 ms, unless the discrepancy time has already expired (i.e., if the discrepancy time has been set to 0 or a value > 100 ms). Older modules do not indicate this sequence error, but their input in the logic editor still remains set to 0.

In the event of a discrepancy or sequence error, the module responds as follows:

- The MS LED of the affected module flashes...
  - with firmware V1.xx.0: ● red (1 Hz)
  - with firmware ≥ V2.00.0: ● red/green (1 Hz)
- The LEDs of the affected inputs flash ● green (1 Hz).
- The **input data status** of the module in the logic editor is set to 0.

Resetting the error

A discrepancy error (timeout) or sequence error is reset once the Deactivated status has been reached.

8.2 ON-OFF filter and OFF-ON filter

When a component with contacts opens or closes, the bouncing of the contacts results in undesirable behavior in the form of several short signal changes. This can affect the evaluation of the inputs. The **ON-OFF filter** for falling signal edges (i.e., 1–0) and the **OFF-ON filter** for rising signal edges (i.e., 0–1) are intended to eliminate this effect.

The **ON-OFF filter** and the **OFF-ON filter** for an element can be activated in the element settings of the element. To do this, go to the **Settings** tab and either click the relevant checkbox or one of the 3D buttons on the right.

If the **ON-OFF filter** or the **OFF-ON filter** is active, a change in the signal is only recognized as such if the status of the signal remains the same for at least as long as the selected filter time. For this purpose, the status of the input is evaluated at 4 ms intervals.

In the case of FX3-XTIO and FX3-XTDI modules with a firmware version ≤ V2.xx.0, the filter time is not adjustable and is fixed at 8 ms.

WARNING
Extended response time

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

- ► Select the filter time to be as short as possible.
- ► Take into account the extended response time.

Effect of the filter on the response time:

- The switch-off response time is extended by at least as long as the selected filter time, if the ON-OFF filter is active.
- The switch-on response time is extended by at least as long as the selected filter time, if the OFF-ON filter is active.
- If the signal changes within the selected filter time, the response time may be extended by longer than the selected filter time, i.e. until a constant signal has been detected for at least as long as the selected filter time.

NOTE
In the case of dual-channel elements with complementary evaluation, the respective filter (ON-OFF or ON-OFF) always relates to the leading channel. The filter for the complementary channel is automatically active.

NOTICE
For larger Flexi Soft systems and when using ON-OFF filters with a filter time > 500 ms at inputs of the FX3-XTIO, the error 0x002AC006 can occur.

- ► If possible, reduce the filter time (≤ 500 ms)

8.3 Deactivating test pulses at FX3-XTIO outputs

With firmware versions ≥ V2.00.0, it is possible to deactivate the test pulses at one or more outputs of FX3-XTIO modules.

Deactivating the test pulses at one or more of the outputs (Q1 to Q4) of an FX3-XTIO reduces the safety parameters of all the outputs (Q1 to Q4) of the module concerned.

WARNING
Reduced safety parameters by deactivating the test pulse

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

- ► This must be taken into account in the risk analysis and risk avoidance strategy.

NOTE
For detailed information about safety parameters, see the operating instructions titled “Flexi Soft Modular Safety Controller Hardware”.

Deactivating test pulses

- ► Connect an output element to the FX3-XTIO module.
- ► Double-click on the output element to open the element settings.
- ► Deactivate the **Enable the test pulses of this output** option.
- ✓ The test pulses of this output are disabled. A corresponding note is displayed underneath the relevant FX3-XTIO in the hardware configuration.

8.4 Extended fault detection time for cross-circuits at outputs Q1 to Q4 of the FX3-XTIO for switching higher capacitive loads

With the FX3-XTIO Step ≥ 3.xx (firmware version V3.00.0), it is possible to configure an extended fault detection time for cross-circuits that affect outputs Q1 to Q4 of FX3-XTIO modules.

This may be necessary for switching loads in cases where the voltage at the load does not drop to the Low level as quickly as expected and would – if the normal fault detection time were to be used – result in a cross-circuit fault immediately after deactivation (switch from High to Low). Example cases include:

- Loads with a capacitance that is higher than the standard level permitted for the output, such as the supply voltage of PLC output cards that require safety-related switching.
  For this application, the test pulse of the output must also be deactivated.
  Safety-capable inputs on fail-safe PLCs generally also have capacitance at the inputs.
- Inductive loads that cause an overshoot in the positive voltage range once the induction voltage has decayed.

Table 114: Maximum permissible time until Low level is reached after output (Q1 to Q4) is deactivated

FX3-XTIO firmware version	Switching higher capacitive loads	Maximum permissible time until Low level (≤ 3.5 V) is reached after output (Q1 to Q4) is deactivated
≤ V2.11.0	Not possible	3 ms
≥ V3.00.0	Deactivated	3 ms
	Activated	43 ms

Once the output has been deactivated, the capacitance that exceeds the standard value permitted for the output must be discharged by the user until the Low level is reached. If this condition is not met within the maximum permissible time, it results in a cross-circuit fault at the relevant output regardless of whether test pulses are activated or deactivated for the output concerned.

WARNING
Loss or impairment of the safety-related switch-off capability due to PLC output card errors

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

- ► Use a PLC output card that is suitable for safety-related deactivation of the outputs by means of supply voltage switching.
- ► Take suitable measures to prevent a cross-circuit, e.g., using protected cable laying.
- ► When using a buffer capacitor in the voltage supply of the PLC output card, observe the possibly extended response time.

Activating the switching of higher capacitive loads

- ► Connect an output element to the FX3-XTIO module.
- ► Double-click the output element.
- ► Select the **Enable switching of higher capacitive loads with this output** option.

⚠ **WARNING**

Extended error recognition time due to switching of higher capacitive loads

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

► Pay attention to the extended error recognition time.

9 Logic programming in Motion Control FX3-MOC0

9.1 General description

The Motion Control module FX3-MOC0 module is used for drive monitoring. It can be used for the safe monitoring of different drives (electric, pneumatic, hydraulic, etc.), provided that a suitable sensor system is installed.

The FX3-MOC0 has its own logic editor. If a project includes one or more FX3-MOC0 modules, you can open the associated logic editors by double-clicking on the relevant module or by using the **Logic editor** menu.

The functions blocks described in this chapter are only available in the logic editor of an FX3-MOC0. They have been specifically tailored for drive monitoring applications. Two types of function block are generally used. Firstly, there are the actual monitoring function blocks that can be used to monitor the speed, position, or stopping and braking functions. Secondly, there are the data conversion function blocks. These are required because the Motion Control modules differ from the rest of the Flexi Soft system in that they are also capable of processing integer data types.

With Flexi Soft Designer Version ≥ V1.7.1, Easy Applications are available in the FX3-MOC0 logic editor. These are customized function blocks with their own configuration dialog that have been prepared by SICK ready for immediate use. Easy Applications make it much easier to configure standard applications. If necessary, Easy Applications can be adapted for a specific application at various levels by going beyond the default settings.

NOTE

The encoder elements are configured outside of the logic editor. For more information, see chapter "Encoder in Motion Control FX3-MOC0", page 302.

Differences between FX3-MOC0 and FX3-MOC1

In the FX3-MOC0, two motion signals can be plausibility-checked using a speed comparison. The relative positions of both signal sources may continuously diverge (e.g., in a vehicle with a sensor on both a left wheel and a right wheel).

In the FX3-MOC1, two motion signals can be plausibility-checked either by means of a position comparison or, as from FX3-MOC1 V3, also be means of a speed cross check. The position comparison is less sensitive to short-term differences in speed. Both detected positions must change at an even rate (allowing for slight drift).

The Motion Control FX3-MOC1 also features additional functions for monitoring absolute positions. It also includes the logic functions AND, NOT and OR to facilitate signal processing within the FX3-MOC1 logic.

9.2 Safety notes for logic programming

Standards and safety regulations

All safety-related parts of the system (wiring, connected sensors and control devices, configuration) must conform to the relevant standards (e.g., IEC 62061 or ISO 13849) and safety regulations.

⚠ **WARNING**
Incorrect configuration of the safety application

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

- ► Observe all applicable standards and safety regulations.
- ► Check that the operating principle of the Flexi Soft hardware and the logic program react in accordance with the risk avoidance strategy.
- ► Only use safety-related signals for safety-related applications.
- ► Always use the correct signal sources for the function blocks.

Safe value

The safe value of process data and outputs is 0 or Low and this is set when an error is identified.

⚠ **WARNING**
Inadequate safety measures

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

If the safe value (signal = Low) could lead to a dangerous state in the application, additional measures must be implemented. This applies in particular to inputs with signal edge detection.

- ► Analyze the status of the process data.
- ► Switch off the affected outputs if the status analysis detects an error.

Unexpected rising or falling signal edges

An error at an input can result in unexpected rising or falling signal edges (e.g. an interruption in network communication, a cable break at a digital input, a short-circuit at a digital input that is connected to a test output). The safe value remains set until the conditions for resetting the error have been met. For this reason, the affected signal may behave as follows:

- It temporarily switches to 1 instead of remaining set to 0 as it normally would in the fault-free status (rising signal edge and falling signal edge, i.e. 0–1–0).
- It temporarily switches to 0 instead of remaining set to 1 as it normally would in the fault-free status (falling signal edge and rising signal edge, i.e. 1–0–1).
- It remains set to 0 instead of switching to 1 as it normally would in the fault-free status.

⚠ **WARNING**
Unexpected rising or falling signal edges

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

- ► Take into account unexpected rising or falling signal edges.

Delays due to jump addresses

Jump addresses can extend the logic execution time and thus the response time if a logical loopback occurs through them.

A logical loopback occurs when a function block input is connected to a destination jump address but the associated source jump address is linked to an output of the same function block or to an output of a function block with a higher function block index number. [5] In this case, the input comprises not the output values of the current logic cycle, but rather the output value of the previous logic cycle. This must be taken into account in terms of the functionality and, in particular, when calculating the response time.

If there is a logical loopback due to a jump address, then this effects a delay of one logic cycle. In this case, the input of the jump address is displayed with a clock symbol (with Flexi Soft Designer ≥ V1.3.0).

WARNING
Extension of the response time

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.

► The delays caused by the logical loopbacks must be factored in when calculating the response time and functionality.

9.3 Parameterization of function blocks

The function blocks have configurable parameters. Double-clicking on a function block opens the configuration window of the function block. The configurable parameters are distributed on different tabs here.

In the case of function blocks for which speeds or positions have to be configured, you can set the units that are to be used on the **Units** tab; e.g., for the purpose of calculating speeds (mm/s, km/h, rpm, etc.).

The **I/O Comment** tab allows you to replace the default designations of the function block inputs and outputs with your own. It also enables you to add a name or descriptive text for the function block, which is displayed under the function block in the logic editor.

The remaining configurable parameters for the function block can be found on the other tabs in accordance with the type of function block.

The **Report** tab shows a summary of the function block configuration. This includes all the links between the inputs and outputs, and the configured parameters.

9.4 Inputs and outputs in the logic editor

Inputs

The following input data is available in the logic editor of the FX3-MOC0:

- Data routed from the main module of the Flexi Soft system, see "Exchange of process data between main module and FX3-MOC0", page 255
- Motion data of the connected encoders, see "Data types used in the logic of the FX3-MOC0", page 253 and "Encoder in Motion Control FX3-MOC0", page 302
- General data sources of the FX3-MOC0, see "General data sources", page 253

NOTE
The inputs and outputs are color-coded according to their function:

- Gray: Non-safe
- Yellow: Safe
- Blue: Diagnostics

[5] The function block index number is displayed at the top of each function block and indicates the position occupied by the function block within the execution sequence.

Outputs

The following output data is available in the logic editor of the FX3-MOC0:

- Data routed to the main module of the Flexi Soft system, see "Exchange of process data between main module and FX3-MOC0", page 255
- 4 customized MOC status bits, see "Module status bits of the FX3-MOC0", page 256

9.4.1 General data sources

Static 0 and Static 1

The inputs **Static 0** and **Static 1** are available in the **Inputs** selection window.

The **Static 0** input can be used to set a function block input permanently to 0. Similarly, the **Static 1** input can be used to set a function block input permanently to 1. This might be necessary, for example, to achieve a valid logic configuration if the relevant function block contains inputs that are not required but cannot be deactivated.

First logic cycle

This input has the value 1 in the first cycle in which the logic of the Motion Control is executed, otherwise it has the value 0.

NOTE
The value of the **First logic cycle** input refers to the logic of the Motion Control. This starts before the logic of the main module. As a result, this input is set to 1 before the corresponding input in the logic editor of the main module.

9.5 Time values and logic execution time

The logic execution time of the Motion Control is 4 ms.

the accuracy of which is ±100 ppm (parts per million).

Table 115: Precision of times (parameters and invariable values) in accordance with increment and absolute value

Configuration increment	Value range for the function block	Precision
4 ms	≤ 5,000 ms	±0.5 ms
	> 5,000 ms	±100 ppm of the configured time
1 s	≤ 40 s	±4 ms
	> 40 s	±100 ppm of the configured time

9.6 Data types used in the logic of the FX3-MOC0

The function blocks in the FX3-MOC0 are capable of processing various data types. This makes them different from the function blocks in the main module, which can only process Boolean values. The type of data that is expected or output depends on which function block input or output is used in each case.

Boolean

Boolean data is binary. It can only be 1 or 0.

Motion V1

Motion V1 data encompasses all the information provided by an encoder. It consists of the following elements:

Table 116: Motion V1 data type structure

Element	Size	Internal value range (number of digits)	Internal resolution for rotary movement type	Internal resolution for linear movement type
Speed	16 bits with sign	−32,768 … +32,767	1 digit = 0.5 rpm	1digit = 1 mm/s
Speed status	1 bit	0 = invalid 1 = valid	–	–
Relative position [1]	32 bits with sign	–2,147,483,648 … +2,147,483,647	1 digit = 1/30,000 rev.	1 digit = 1/250 mm
Relative position status	1 bit	0 = invalid 1 = valid	–	–
Absolute position [2]	32 bits with sign	–2,147,483,648 … +2,147,483,647	1 digit = 1/30,000 rev.	1 digit = 1/250 mm
Absolute position status	1 bit	0 = invalid 1 = valid	–	–

[1] A relative position means that the traveled path can be reproduced, but the position in relation to the mechanical position is not clear. This primarily occurs because the start value for the relative position in the encoder's Motion V2 data always starts with a 0, regardless of the mechanical position.

[2] An absolute position means that the position value is clear for any possible mechanical position in the application. This also applies following a restart of the measurement system.

NOTE

The following rules apply in relation to the status:

- When a status bit 0 = invalid, then the corresponding value is 0.
- The **Relative position status** can only be 1 if the **Speed status** is 1.
- The FX3-MOC0 cannot be used to evaluate absolute positions. For this reason, the **Absolute position** and **Absolute position status** are always 0 in the FX3-MOC0.
- When the status of a value is set to 0 = invalid, it can become 1 = valid again only after 1 s has elapsed (**Error Recovery Time**) as soon as valid data is available again.

The **Motion V1** data type is used by the **Motion In** inputs and the **Motion Out** output of various FX3-MOC0 function blocks. The individual elements of the data type are evaluated automatically in the respective FX3-MOC0 function block.

Internal resolution of the speed and position information

The smallest unit for the detected speed and position information is determined by the internal resolution of this data (see table 116). It may be further restricted by the resolution of the encoder system.

UI8

Data of the UI8 type can, for example, be used to select or display a speed or position range.

Table 117: Possible values for UI8 data

Element	Size	Values for speed ID
UI8	8 bits	0 = invalid 1 … 31 = range index

NOTE

Inputs and outputs that expect or that output data types other than Boolean are marked accordingly on the function block icons. Within this context, **M** stands for **Motion V1** and **UI8** stands for **Unsigned Integer 8-Bit**.

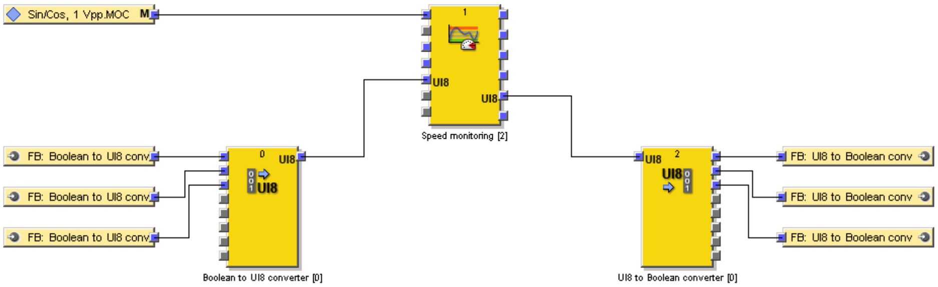


Figure 199: How data types are displayed at the inputs and outputs of the function blocks in FX3-MOC0

9.7 Exchange of process data between main module and FX3-MOC0

Given that the main modules and the FX3-MOCx modules are capable of processing different types of data and that more complex signal preprocessing and logic can be programmed in the FX3-MOCx, data exchange between the modules must be properly organized. 18 bits can be sent from the main module to the FX3-MOCx and 16 bits can be sent from the FX3-MOCx to the main module. These bits must be linked in the logic editor.

The bits from the FX3-MOCx to the main module appear ...

- In the Logic editor of the FX3-MOCx under **Outputs** and
- In the Logic editor of the main module under **Inputs** for the relevant FX3-MOCx module.

The bits from the main module to the FX3-MOCx appear ...

- In the Logic editor of the FX3-MOCx under **Inputs** and
- In the logic editor of the main module under **Outputs** for the relevant FX3-MOCx.

The tag name for these bits is pre-populated with the name of the input + block + module (default name). This name can be changed if required.

The data is exchanged via the internal FLEXBUS+ bus.

WARNING

Unrecognized signals from the FX3-MOCx module

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

► Observe the time requirements for signals that are sent from the FX3-MOCx to the main module.

The signals that are sent from an FX3-MOCx to the main module must meet the same time requirements as all other signals. If the logic execution time of the main module is more than 4 ms, a signal sent from the FX3-MOCx to the main module must continue to have the same status for at least as long as the logic execution time of the main module. This is necessary to ensure that this status can always be detected in the main module logic.

Following a transition to the Run status, there is a delay of up to 80 ms before logic processing by the main module commences. This ensures that the logic of the main module always has the latest valid signals of the expansion modules to work with. As a result of this, all the data sent from the main module to the FX3-MOCx remains set to 0 once the system has transitioned to the Run status for up to 80 ms plus the logic execution time of the main module. This applies in particular to signals that are transmitted by inputs to expansion modules and whose status is sent to the FX3-MOCx via the main module.

9.8 Module status bits of the FX3-MOC0

Overview

The module status bits for the expansion modules contain diagnostics data. This data is refreshed approximately every 200 ms. Due to the longer refresh interval, this data may not be consistent with the latest process data for the module.

Important information

WARNING
Non-secure or inconsistent data

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

- ► Only use secure data for safety-related applications.
- ► Only use the module status bits of the expansion modules for diagnostic purposes.

Module status bits of the FX3-MOC0

The module status bits of the FX3-MOC0 are available at the following locations:

- In the logic editor of the main module under **Diagnostics**, for use as inputs in the logic program of the main module
- In the Flexi Soft gateways in data set 3; see the operating instructions titled “Flexi Soft Gateways in the Flexi Soft Designer Configuration Software” (SICK part number 8012483).
- In the RS-232 routing of the main module.

Table 118: Module status bits of the FX3-MOC0

Module status bit	Description
Configuration is valid	0 = Configuration invalid 1 = Configuration valid
Encoder 1 is OK	0 = Error 1 = No error or not used
Encoder 2 is OK	0 = Error 1 = No error or not used
Customized MOC status bit 1 … 4	• User-defined module status bits • Alarm generation

Further topics

- "Input data status and output data status", page 93
- "RS-232 routing", page 56

9.9 Overview of the function blocks in the FX3-MOC0

NOTE
For information on connecting and configuring encoders, see "Encoder in Motion Control FX3-MOC0", page 302.

The Logic editor of the FX3-MOC0 uses function blocks to define the safety-related logic. A configuration can include up to 10 function blocks.

Depending on the type of function blocks used, even a small number can result in the available computing time in the logic execution cycle or the available memory capacity being exceeded. These values are displayed in the Logic editor of the FX3-MOC0, on the **FB group info** tab.

There are some function blocks for monitoring and some others for data conversion. The following table lists all the function blocks that are available for the FX3-MOC0:

Table 119: Overview of the function blocks in the FX3-MOC0

Name of the function block	Description
Function blocks for monitoring	
Speed cross check V1	Compares speed values from two different signal sources. This makes it possible to achieve a higher level of safety.
Speed monitor V1	Enables speed and direction monitoring. The basic functions are: • Safe speed monitor (SSM) • Safely limited speed (SLS) • Safe direction (SDI) • Safe operating stop (SOS) • Monitoring of up to four different speed ramps during the transition from a monitored speed to a lower speed
Safe Stop V1	This is used to trigger and monitor a Safe Stop for a drive system. The drive has to be shut down in a controlled manner. The stop ramp of a drive system is not usually safe. For this reason, the Safe stop V1 function block monitors the actual reduction in speed until the drive comes to a standstill. Functions: • Safe Stop 1 (SS1) • Safe Stop 2 (SS2) Typical application: • Monitoring the shutdown and stopping behavior of a machine
Function blocks for data conversion	
UI8 to Bool V1	Converts an 8-bit integer value into a Boolean value. Possible application: For connecting the **Speed status ID** output of the **Speed monitor V1** function block to Boolean signals so that they can be forwarded to the main module.
Bool to UI8 V1	Converts a Boolean value into an 8-bit integer value. Possible application: For connecting the **Speed enable ID** input of the **Speed monitor V1** function block to Boolean signals from the main module.
Motion status to Bool V1	Converts the **Speed status**, the **Relative position status**, and the **Absolute position status** from the data type Motion V1 to Boolean.
Speed to Bool V1	Converts the **Speed** and the **Speed status** from the data type Motion V1 to Boolean.
Speed to laser scanner V1	Converts the **Speed** from the data type Motion V1 into a format with cm/s scaling that is suitable for a SICK laser scanner.

NOTE

The suffix **V1** or **V2** is used to differentiate between the different versions of the function blocks. The version of the function block that is used depends on the module type used (FX3-MOC0 or FX3-MOC1) and the module version.

9.10 Function blocks for monitoring

9.10.1 Speed cross check V1

Function block diagram

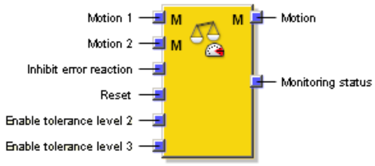


Figure 200: Inputs and outputs of the Speed Cross Check V1 function block

General description

The **Speed cross check V1** function block compares speed values from two different signal sources. The associated performance checks are used to achieve a higher level of safety, particularly when working with non-safe encoders.

Deviations in both measured values may occur continuously or temporarily due to phenomena such as slip, abrasion, or mechanical coupling behavior. Consequently, this function block offers various parameters that can be used to tolerate deviations of this kind. In this way, unintended shutdowns can be avoided and machine availability can be ensured.

The following factors can be taken into account during evaluation:

- Permanently tolerated absolute speed difference or permanently tolerated speed ratio (relative speed difference in %), e.g. caused by different levels of component wear)
- Temporarily increased tolerance limits for the speed ratio, e.g. due to automation process requirements such as cornering by an AGV
- Signs of the speed values when calculating the speed difference

Fault detection

WARNING
Incorrect configuration

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

- ► As part of the safety assessment, consider the parameters used to configure the **Speed cross check V1** function block.
- ► It must also be noted here that fault detection by the speed cross check is prevented by use of the optional **Inhibit error reaction** input.

In accordance with generally recognized testing principles, test authorities usually stipulate that the application must ensure the monitored unit performs a movement at least once within the space of 24 hours. This movement must generate a signal change on the encoder system so that the speed cross check function can use it as a basis for detecting the relevant faults.

Inputs of the function block

Table 120: Inputs of the function block Speed cross check V1

Input	Description	Possible values
Motion In 1	For connecting two encoders. Type Motion V1 data is expected (see "Data types used in the logic of the FX3-MOC0", page 253).	Data of type Motion V1
Motion In 2		

Input	Description	Possible values
Reset	Optional input for error reset by an external signal	- Deactivated - Active
Inhibit error reaction	Optional input, allows the error reaction to be suppressed if a speed cross check fails	0 = no suppression 1 = error reaction inhibited
Enable tolerance limit 2	Optional inputs that can be used to select increased tolerances for the speed cross check if required	- Deactivated - Active
Enable tolerance limit 3		

Outputs of the function block

Table 121: Outputs of the function block Speed cross check V1

Output	Description	Possible values
Motion Out	Output of the checked Motion V1 data for use in another function block, e.g. **Speed monitoring V1**. Output of the values may take place after a delay depending on the configuration of the function block.	Depending on the **Speed output mode** parameter. The **Speed status** is set depending on the result of the speed cross check.
Monitoring status	Indicates whether a speed cross check has failed. The initial status when the Flexi Soft system transitions to the Run status is 1. The output switches to 0 if the determined deviation exceeds the selected speed tolerance in a speed cross check. The output switches back to 1 when the determined deviation is again lower than or equal to the selected tolerance, but only after the **Error recovery time** of 1 s has elapsed.	0 = Error detected 1 = OK (no error detected or status unknown)

Function block parameters

Table 122: Parameters of the function block Speed cross check V1

Parameter	Description	Possible values
Inputs		
Reset input	The optional **Reset** input can be activated so that errors are reset by an external signal.	- Inactive - Active
Inhibit error reaction	The optional **Inhibit error reaction** input can be activated.	- Inactive - Active
Automated reset is dependent upon the absolute speed	Automated reset of an error with or without taking into account the current speed at the time when an inadmissible speed ratio was detected. This option is only available when the **Inhibit error reaction** input is active.	- Inactive - Active
Speed cross check mode		
Speed cross check mode	Specifies whether a tolerance speed is to be calculated and whether the sign should be taken into account for the calculation.	- No speed cross check - With sign - Without sign
Limits for speed cross check		
Absolute tolerance limit for speed difference	Permanently permissible absolute speed difference between **Motion In 1** and **Motion In 2**. Speed differences that exceed this limit are taken fully into account.	**0 … 32,767 digits =** - 0.5 … 16,383 rpm - 1 … 32,767 mm/s - 0 = infinite

Parameter	Description	Possible values
Speed ratio Tolerance limit 1	Permanent tolerance for the speed ratio (permissible relative speed difference in %) between **Motion In 1** and **Motion In 2** based on the higher of the two values	0 … 100%
Speed ratio Tolerance limit 2 Speed ratio Tolerance limit 3	Conditional increased tolerance for the speed ratio (permissible relative speed difference in %) between **Motion In 1** and **Motion In 2** based on the higher of the two values. • Enable via **Enable tolerance limit 2** and **Enable tolerance limit 3** input • Optional time limitation	0 … 100%
Max. time for tolerance limit 2	Maximum length of time for which **Speed ratio tolerance limit 1** may be exceeded while tolerance limit 2 is valid	0 = infinite 4 … 60,000 ms in 4 ms increments
Max. time for tolerance limit 3	Maximum length of time for which **Speed ratio tolerance limit 2** may be exceeded while tolerance limit 3 is valid	0 = infinite 4 … 60,000 ms in 4 ms increments
Compensation for signal propagation delay		
Signal propagation delay of Motion In 2	Propagation delay compensation for a delayed signal received at the **Motion In 2** input during speed cross check	0 … 100 ms in 4-ms increments
Speed cross check on value change	Determines if speed difference evaluation should take place continuously or only when there is a change in speed at **Motion In 2** or – at the latest – after the maximum evaluation pause	• Continuous speed cross check • Speed cross check on value change
Max. evaluation pause	Defines the minimum interval for evaluating the speed difference	4 … 1,000 ms in 4-ms increments
Speed output mode		
Speed output mode	Mode for calculation of the speed output at the **Motion Out** output	• Speed of Motion In 1 • Higher speed of Motion In 1 or Motion In 2 • Mean speed of Motion In 1 and Motion In 2

Speed cross check mode

The **Speed cross check mode** determines whether a speed cross check is performed and whether the sign is taken into account in this case when the speed values are cross checked.

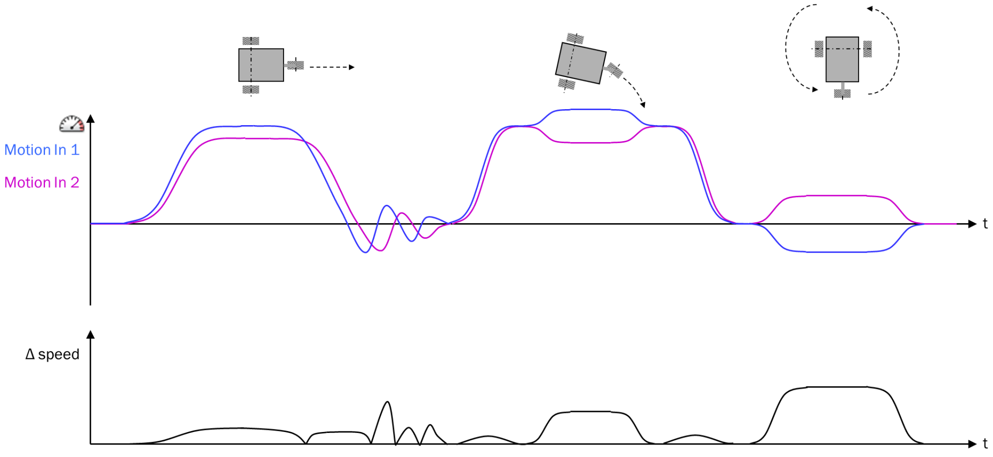


Figure 201: Speed cross check mode: Speed difference calculated with sign

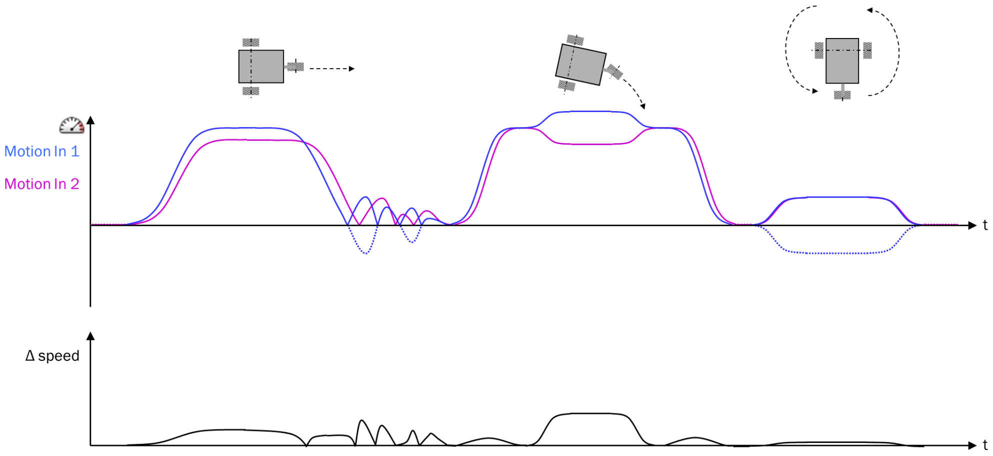


Figure 202: Speed cross check mode: Speed difference calculated without sign

Permanently tolerated speed difference

The **Absolute tolerance limit for speed difference** parameter can be used to define the permissible absolute speed difference. Speed differences that are less than the **Absolute tolerance limit for speed difference** are evaluated as 0, i.e. are ignored. The purpose of this is to avoid high speed ratios at low speeds. Otherwise, a low speed difference could lead to a high speed ratio because the speed being referenced is also low. Speed differences that exceed this tolerance limit are taken fully into account.

Permanently tolerated speed ratio

The **Speed ratio tolerance limit 1** parameter can be used to define the permissible speed ratio with reference to the higher of the speed values from **Motion In 1** and **Motion In 2**. The higher of the two values is taken to be 100% and the defined speed ratio is limited to 100%.

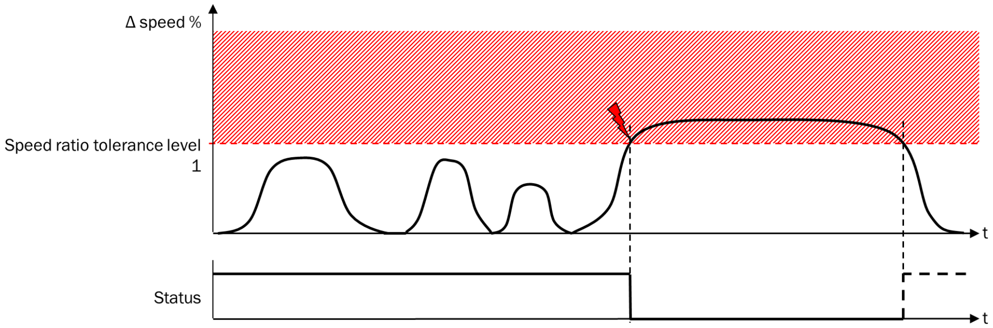


Figure 203: Permanently tolerated speed ratio

When the Flexi Soft system transitions to the Run status, the **Monitoring status** output is 1. As long as no errors occur, the value of the output does not change.

The **Monitoring status** output switches to 0 when the permissible speed ratio (relative speed difference in %) defined by the parameter **Speed ratio tolerance limit 1** is exceeded. This happens regardless of the status of the **Inhibit error reaction** input.

If the speed at the **Motion In 1** or **Motion In 2** input is invalid, then the speed of the affected input is set to 0 internally for the speed cross check. This means that an invalid speed will not have a direct influence on the **Monitoring status** output, but the possible resultant higher speed than the permitted speed difference will influence this.

NOTE
If the speed value of either the **Motion In 1** or the **Motion In 2** input is invalid, the **Motion Out** output always becomes invalid as well.

Increased tolerance limit for the speed ratio

It is possible to increase the tolerance limit for the permissible speed ratio. The **Speed Cross Check V1** function block supports two additional tolerance limits with a conditional increase. Each of these can be configured with its own maximum time period.

The **Enable tolerance limit 2** and **Enable tolerance limit 3** inputs must be activated in the configuration dialog of the function block. Unless these inputs are activated, **Speed ratio tolerance limit 2** and **Speed ratio tolerance limit 3** will not be available.

When the **Enable tolerance limit 2** input is set to 1, it is permitted to exceed the value of **Speed ratio tolerance limit 1**. In this case, the increased **Speed ratio tolerance limit 2** is active. The duration for the limit being exceeded can be limited by the **Max. time Tolerance limit 2** parameter. A value of 0 ms here means infinite, i.e. no time limit. If the **Speed ratio tolerance limit 1** is exceeded for longer than the configured **Max. time Tolerance limit 2**, the **Monitoring status** output switches to 0.

The same applies to the third possible limitation: If the **Enable tolerance limit 3** input is set to 1, it is permitted to exceed the value of **Speed ratio tolerance limit 2** and the increased **Speed ratio tolerance limit 3** is active. The duration for the limit being exceeded can be limited by the **Max. time Tolerance limit 3** parameter. A value of 0 ms here means infinite, i.e. no time limit. If the **Speed ratio tolerance limit 2** is exceeded for longer than the configured **Max. time Tolerance limit 3**, the **Monitoring status** output switches to 0.

Speed ratio tolerance limit 3 is the highest increased tolerance limit for the speed ratio and must never be exceeded.

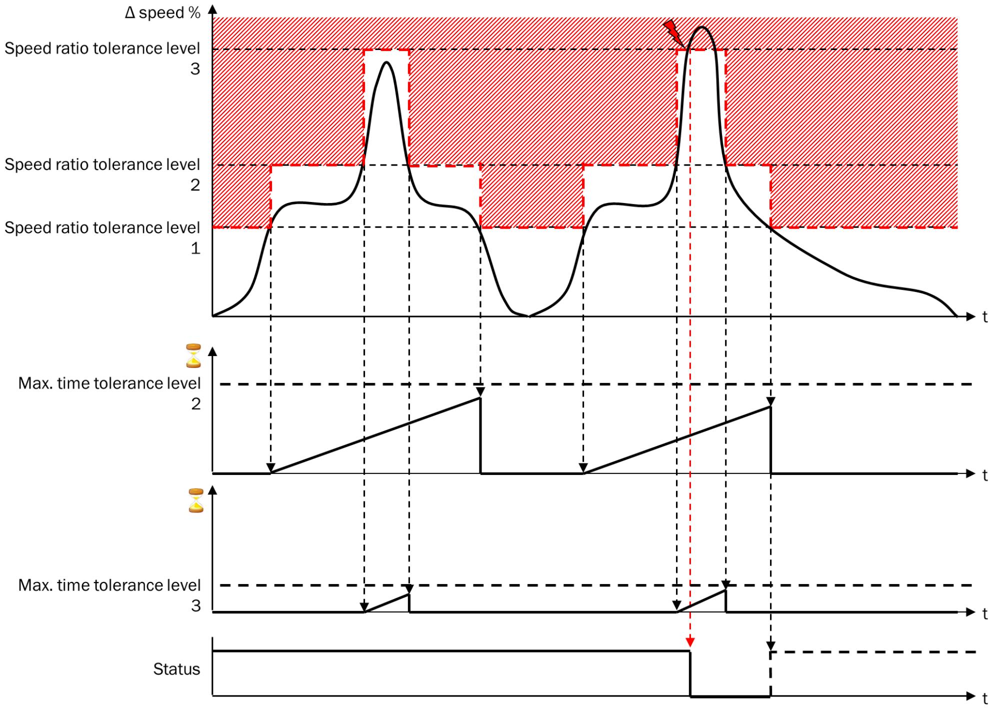

Δ speed %
Speed ratio tolerance level 3
Speed ratio tolerance level 2
Speed ratio tolerance level 1
t
Max. time tolerance level 2
t
Max. time tolerance level 3
t
Status
t

Figure 204: Conditional increased tolerance limit for the permitted speed ratio with exceeded tolerance limit

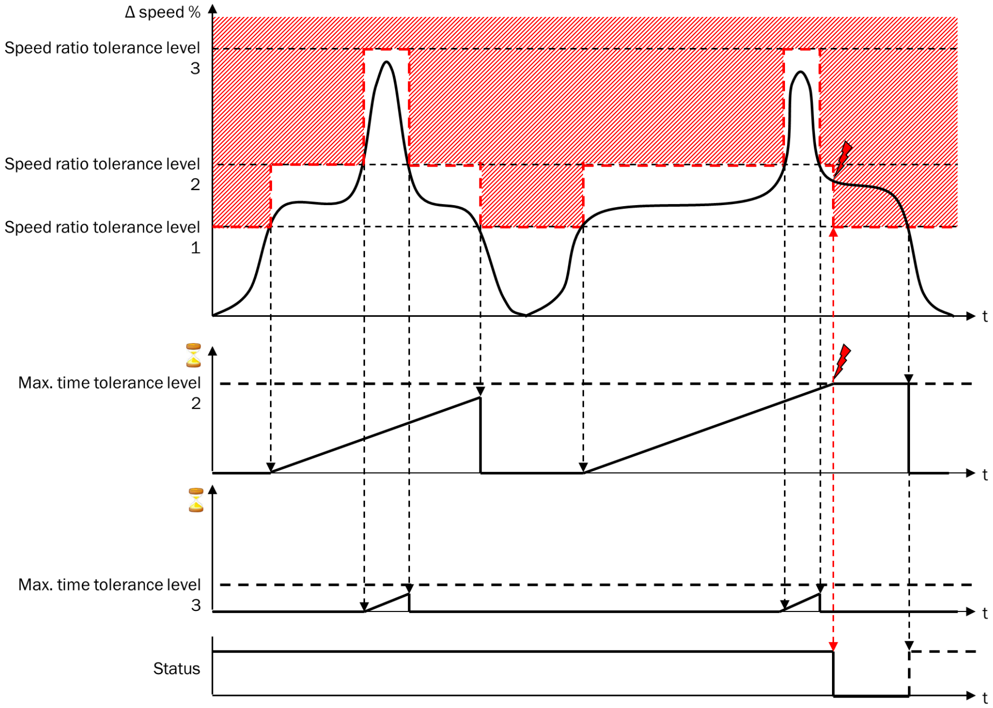


Figure 205: Conditional increased tolerance limit for the permissible speed ratio with exceeded time limit

For example, in the case of an AGV, the increased **Speed ratio tolerance limit 2** allows the difference in speed between two wheels to be tolerated during cornering. This means that the vehicle control system can enable the **Enable tolerance limit 2** input when the vehicle is turning a corner. The increased **Speed ratio tolerance limit 3** can be used to tolerate very short speed differences, e.g. when a wheel spins briefly.

Propagation delay compensation for Motion In 2

If one of the two speed signals is transmitted with a delay, the delayed signal can be connected to the **Motion In 2** input and the delay compensated using the **Signal propagation delay Motion In 2** parameter. This means that the speed values of the **Motion In 1** input are delayed by the specified period of time when calculating the speed difference. This makes it possible to reduce the speed differences that would otherwise be caused by delayed transmission, particularly in the case of rapid changes in speed.

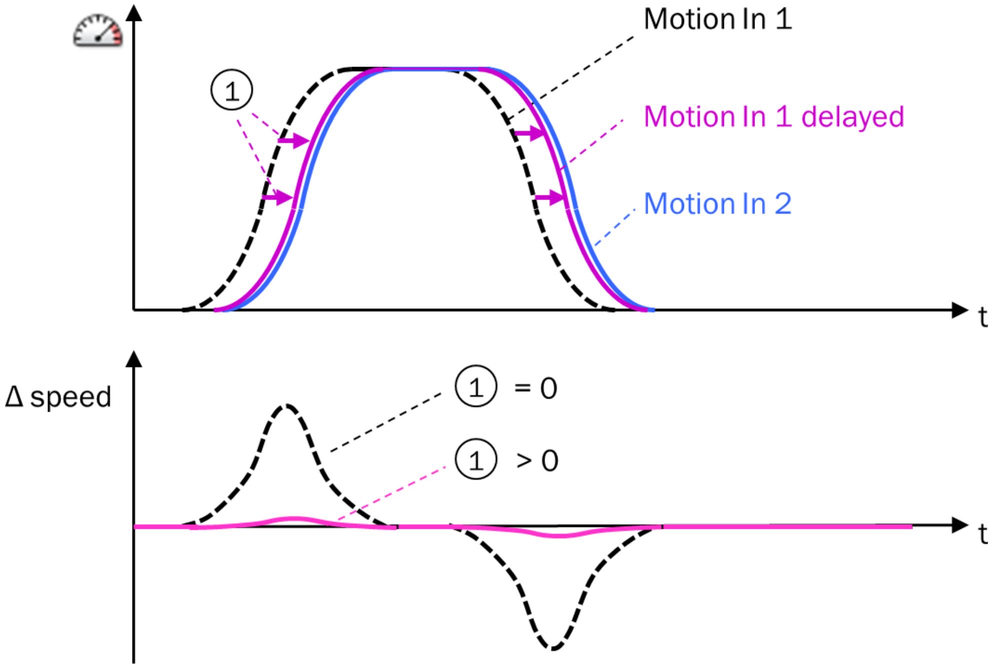


Figure 206: Compensation of transmission delays with Signal propagation delay Motion In 2

① Signal propagation delay Motion In 2 parameter

After a transition from the Stop status to the Run status, the first cross check is performed at the earliest after **Signal propagation delay Motion In 2** has elapsed.

Cross check only performed when there is a change to the speed values

If the update rate of either speed signal is lower than that of the other signal and is also less than the logic execution time, the cross check can be skipped between updates. This may be applicable, for example, if one of the encoders has a much lower resolution than the other one or if the second speed signal is received via the FLEXBUS+ bus.

The **Max. evaluation pause** serves to reduce the resultant speed differences. If this parameter is > 0, the cross check is only performed if the value of **Motion In 2** changes or – at the latest – after the time set by the parameter.

If you set the **Max. evaluation pause** to 0, the cross check is performed continuously.

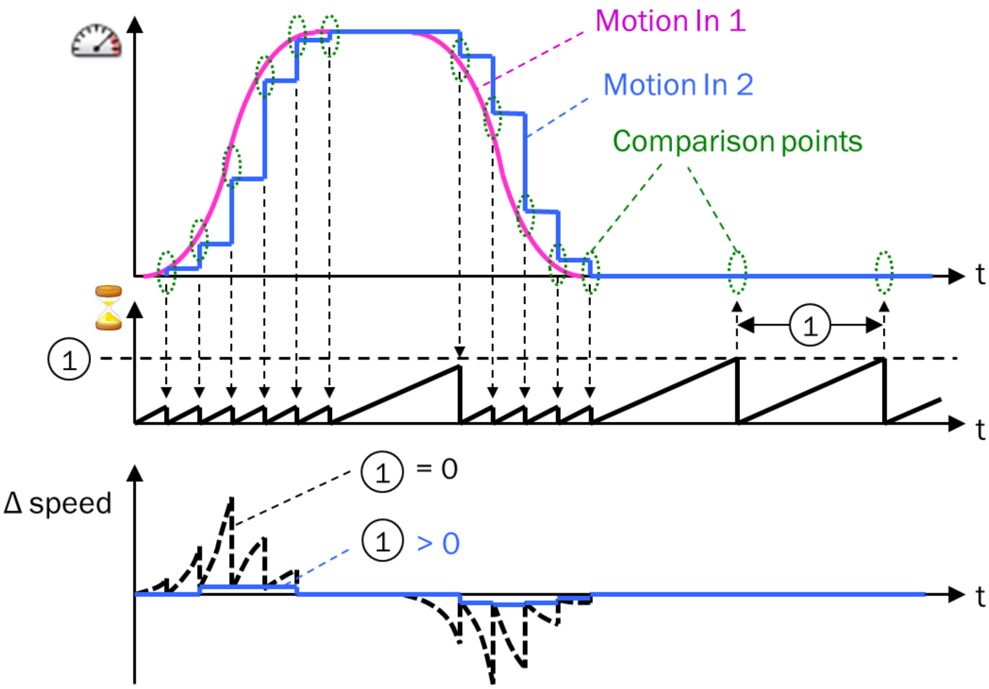


Figure 207: Speed comparison with max. evaluation pause

① Max. evaluation pause parameter

Speed output mode for output at the Motion Out output

The values of the Motion Out output are formed based on the Motion In 1 and Motion In 2 inputs and are dependent on the Speed output mode parameter in some cases. The following settings are possible:

- Speed of Motion In 1
- Higher speed of Motion In 1 or Motion In 2
- Mean speed of Motion In 1 and Motion In 2

NOTE

- If the speed value of either the **Motion In 1** or the **Motion In 2** input is invalid, the **Motion Out** output always becomes invalid as well.
- The **Motion Out** output also becomes invalid if the function block is in the error status, i.e., when one of the cross checks fails and the error status is not prevented by the **Inhibit error reaction** input.

Speed of Motion In 1

Speed at **Motion Out** output:

With this setting, the **Speed** value at the **Motion Out** output corresponds to the **Speed** value at the **Motion In 1** input.

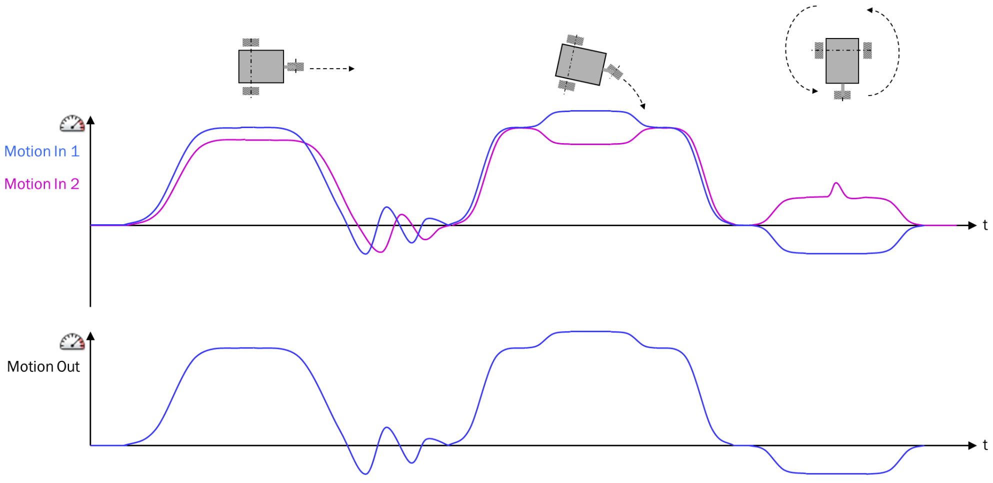


Figure 208: Speed output mode: Speed of Motion In 1

Speed of Motion In 1 is usually selected when there is a leading encoder with a higher resolution and also a second encoder that is used for plausibility checks. In this case, the values at the **Motion In 1** input are used for further evaluation.

Relative position at **Motion Out** output:

With this setting, the **Relative position** value at the **Motion Out** output corresponds to the **Relative position** value at the **Motion In 1** input.

Higher speed of Motion In 1 or Motion In 2

WARNING
Discontinuous speed curve and faulty standstill detection

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

- ► Take account of discontinuous speed curves.
- ► Evaluate the standstill condition for each of the two encoders (e.g., by using two separate **Speed monitoring V1** function blocks) and the results combined using a logical **AND** function block.

Speed at **Motion Out** output:

With this setting, the higher of the two speeds at the **Motion In 1** and **Motion In 2** inputs is output at the **Motion Out** output.

NOTE
The resulting speed curve may be non-continuous if two opposing directions of movement are involved.

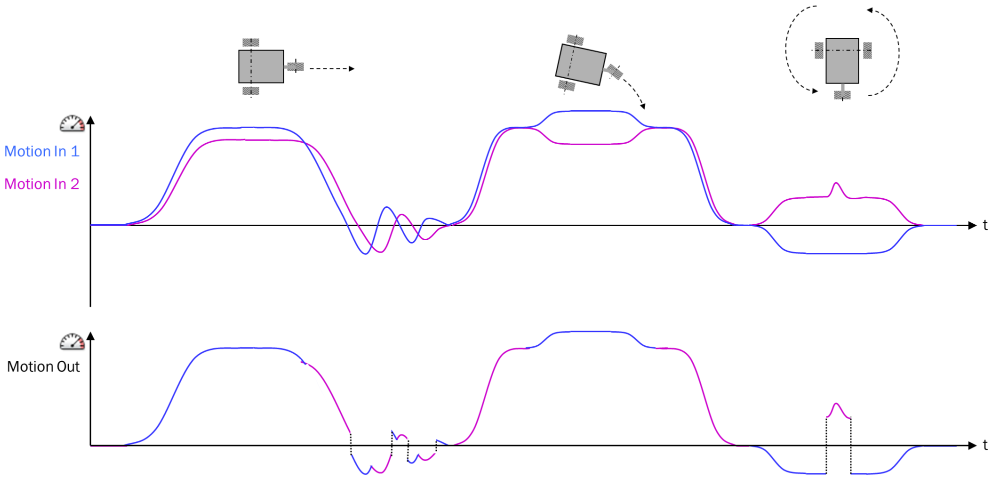


Figure 209: Speed output mode: Higher speed of Motion In 1 or Motion In 2

The **Higher speed of Motion In 1 or Motion In 2** setting should be selected if the higher value from two encoders is to be used for further evaluation, e.g. because the speed of the outer wheel of an AGV during cornering is of relevance (worst-case scenario).

Relative position at **Motion Out** output:

With this setting, the system first calculates the respective differences in **Relative position** at the **Motion In 1** and **Motion In 2** inputs compared with the previous cycle. The higher of the two differences is then added to the **Relative position** value at the **Motion Out** output from the previous cycle, which means that the greatest relative difference in position is applied.

NOTE

If the **Speed output mode** parameter is set to **Higher speed of Motion In 1 or Motion In 2** or to **Mean speed of Motion In 1 and Motion In 2** and the value at the **Motion Out** output is used for standstill monitoring with standstill position tolerance (e.g., using the **Speed monitoring V1** function block), the standstill condition may be met even though the relative positions at the **Motion In 1** input and **Motion In 2** input would not meet the standstill condition when considered on their own. This case can occur if the two encoders are moving in opposite directions and the resulting relative difference in position is less than the position difference of each individual encoder.

Mean speed of Motion In 1 and Motion In 2

WARNING

Incorrect standstill detection

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

► Evaluate the standstill condition for each of the two encoders (e.g., by using two separate **Speed monitoring V1** function blocks) and the results combined using a logical **AND** function block.

Speed at **Motion Out** output:

With this setting, it is the average speed from the **Motion In 1** and **Motion In 2** inputs that is output at the **Motion Out** output together with its sign.

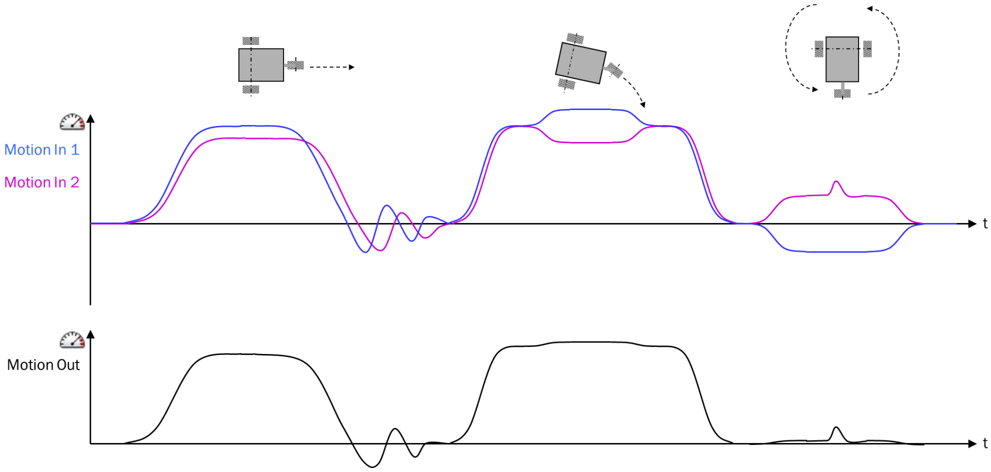


Figure 210: Speed output mode: Mean speed of Motion In 1 and Motion In 2

The **Mean speed of Motion In 1 and Motion In 2** setting is usually used when the average speed from two encoders is of relevance for further evaluation because, for example, it represents the speed at the central point of an AGV. With this setting, turning on the spot (when both wheels are traveling at the same speed in opposite directions) is evaluated as a standstill, for example.

Relative position at **Motion Out** output:

When the **Speed output mode** is set to **Mean speed of Motion In 1 and Motion In 2**, the system first calculates the respective differences in **Relative Position** at the **Motion In 1** and **Motion In 2** inputs compared with the previous cycle. The average of these two values is then added to the **Relative position** value at the **Motion Out** output from the previous cycle.

NOTE

If the **Speed output mode** parameter is set to **Higher speed of Motion In 1 or Motion In 2** or to **Mean speed of Motion In 1 and Motion In 2** and the value at the **Motion Out** output is used for standstill monitoring with standstill position tolerance (e.g., using the **Speed monitoring V1** function block), the standstill condition may be met even though the relative positions at the **Motion In 1** input and **Motion In 2** input would not meet the standstill condition when considered on their own. This case can occur if the two encoders are moving in opposite directions and the resulting relative difference in position is less than the position difference of each individual encoder.

Inhibit error reaction

You can use the optional **Inhibit error reaction** input to control whether an error (**Monitoring status** = 0) triggers an error response at the same time. In this case, the **Motion Out** output is set to invalid. If safety is assured independently of the monitored movement, e.g. by means of a closed protective door, this error response can be inhibited. If the **Inhibit error reaction** input is 1, an error (**Monitoring status** = 0) does not result in the **Motion Out** output being set to 0.

Resetting the function block

The **Monitoring status** output switches back to 1 once the error has been reset. This can be achieved in two ways:

- Manual reset: An error is reset by a rising signal edge at the optional **Reset** input if the previously averaged speed at the **Motion In 1** input has remained virtually zero for a period of approximately 1 s and the relative speed difference is below **Speed ratio tolerance limit 1**.

Virtually zero means...
  - 40 rpm for encoders with rotary movement and
  - 80 mm/s for encoders with linear movement
- Automated reset: An error is reset if the optional **Inhibit error reaction** input is 1 and the speed difference drops below the permissible relative speed difference before the **Inhibit error reaction** input switches from 1 to 0. The **Automated reset is dependent upon the absolute speed** option determines whether the absolute speed should be taken into account as part of this. If this option is activated, the absolute speed must exceed the speed limit at which the **Monitoring status** output previously switched to the 0 status, i.e. the point when the error was detected. Otherwise, the speed will not affect the error reset process.

NOTE

If neither the optional **Inhibit error reaction** input nor the **Reset** optional input is activated, there is no way to reset an error status during operation.

9.10.2 Speed monitor V1

Function block diagram

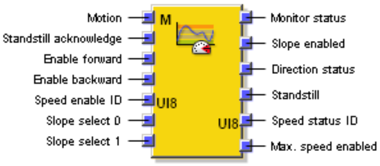


Figure 211: Inputs and outputs of the Speed monitoring V1 function block

General description

The **Speed monitor V1** function block is the central block for all speed and direction monitoring functions within an application. It can essentially perform the following functions:

- Safe speed monitor (SSM)
- Safely limited speed (SLS)
- Safe direction (SDI)
- Safe operating stop (SOS)
- Monitoring of up to four different speed ramps during the transition from a monitored speed to a lower speed

Speed monitoring

- Maximum speed monitoring
- Monitoring the speed limits selected via the inputs **Speed enable ID** and, where applicable, **Ramp selection 1** and **Ramp selection 2**
- Direction monitoring, selected via the inputs **Enable forward** and **Enable backward**

The speed monitoring status is displayed at the **Monitoring status** output.

Inputs of the function block

Table 123: Inputs of the Speed monitoring V1 function block

Input	Description	Signal value
Motion in	Data of type Motion V1 is expected, either directly from an encoder or from another function block (see "Data types used in the logic of the FX3-MOC0", page 253).	Data of type Motion V1
Standstill acknowledge	Optional input, activates internal standstill detection	0 = Standstill detection deactivated 1 = Standstill detection active
Enable forward and enable backward	Optional inputs to enable the relevant direction of movement. If either of these inputs is not used, it means that the associated direction of movement is permanently enabled.	0 = Direction not enabled 1 = Direction enabled Input not used = Direction permanently enabled
Speed enable ID	Activates the permitted speed limit	0 ... 255
Ramp selection 1 and Ramp selection 2	Selection of up to four speed ramps with different gradients	0 or 1

Outputs of the function block

Table 124: Outputs of the Speed monitoring V1 function block

Output	Description	Signal value
Monitoring status	At the **Monitoring status** output, the combined status of the various monitoring functions is displayed (AND link). The initial status when the Flexi Soft system transitions to the Run status is 1.	0 = Error detected 1 = OK (no error detected)
Ramp active	Indicates that a speed ramp is active	0 = No ramp active 1 = Speed ramp active
Direction status	Indicates the direction of movement. No change in the event of standstill. If the status is unknown, i.e., if the speed at the **Motion in** input is invalid, the output switches to 0. The initial status when the Flexi Soft system transitions to the Run status is 0.	0 = Forwards or status unknown 1 = Backwards
Standstill	Indicates whether the standstill condition is fulfilled (either standstill speed and/or standstill position, taking into account the filter and acceptance criteria such as the **Standstill approval** input and the **Standstill speed acceptance time** parameter). If the status is unknown, i.e., if the relevant data at the **Motion in** input is invalid, the output switches to 0. The initial status when the Flexi Soft system transitions to the Run status is 0.	0 = No standstill or status unknown 1 = Standstill
Speed status ID	Indicates which speed range the current speed at the **Motion in** input corresponds to. This is not affected by the speed limit currently active for monitoring.	0 ... 10 0 = Invalid speed 1 = Standstill 2 ... 10 = speed range 2 ... 10
Max. speed enabled	Indicates whether the highest configured speed limit is active (selected via the **Speed enable ID**) input. This is not affected by the current speed at the **Motion in** input.	0 = Maximum speed not enabled 1 = Maximum speed enabled

Function block parameters

Table 125: Parameters of the Speed monitoring V1 function block

Parameter	Description	Possible values
Max. speed		
Max. speed	Defines the maximum speed that is allowed to occur within the system	**1 … 32,767 digits =** • 0.5 … 16,383 rpm • 1 … 32,767 mm/s
Standstill monitoring		
Standstill Speed	Defines which speed still counts as a standstill	**0 … 32,767 digits =** • 0 … 16,383 rpm • 0 … 32,767 mm/s
Standstill speed acceptance time:	Defines the uninterrupted period of time for which the **Standstill speed** may not be exceeded	0 … 248 ms in 4 ms increments
Standstill Position Window Tolerance	Defines which relative position change still counts as a standstill during standstill monitoring	**0 … 500,000,000 digits =** • max. 16,666 rev. • max. 2,000,000 mm • 0 = Deactivated
Standstill acknowledge	Activates the optional **Standstill approval** input	• Deactivated • Active
Speed limits		
Speed limit 1	Speed limit 1 always corresponds to the **Standstill speed**	**0 … 32,767 digits =** • 0 … 16,383 rpm • 0 … 32,767 mm/s
Speed limit 2 … 9	Up to 9 speed limits, including standstill speed	
Speed enable ID	Activates the optional **Speed enable ID** input	• Deactivated • Active
Ramps for speed transitions		
Delay time until start of ramp	How long the function block expects there to be no response from the system, i.e., the length of time for which it does not expect a delay ramp	0 … 248 ms in 4 ms increments
Ramp configuration (ramp steepness, speed transitions 1 … 4)	Increment for the reduction in speed when changing from a higher active speed enable ID to a lower one, selected via the **Speed enable ID** input. You can define up to four different ramps. The **Speed reduction** and the **Duration of the speed reduction** are entered.	**0 … 2,147,418,112 digits =** • 0.5 … 16,383 (rpm)/ms • 1 … 65,535 (mm/s)/ms • 0 = no ramp
Direction monitoring		
Enable forward	Activates the optional input **Enable forward**	• Deactivated • Active
Enable backward	Activates the optional input **Enable backward**	• Deactivated • Active

Safe speed monitor (SSM)

The parameters **Standstill speed** and **Speed limits 2–9** can be used to configure 9 speed limits (including the standstill speed) for up to 10 speed ranges. The lowest speed limit is always the standstill speed.

The **Speed status ID** output indicates which speed range corresponds to the current speed at the **Motion in** input. This is not affected by the speed limit that is currently active. The data is output as a UI8 value. To connect this value to Boolean signals, use the **UI8 to Bool V1** function block.

*Table 126: **Speed status ID** output with 9 configured speed limits*

Speed at Motion in input	Meaning	Speed status ID
The speed is not valid.	Invalid	0
Standstill condition met: • The speed remains lower than the standstill speed for at least as long as the standstill speed acceptance time. **Or:** • The standstill position tolerance has been determined and will not be exceeded.	Standstill	1
No standstill and speed > standstill speed Speed ≤ speed limit 2	Speed range 2	2
Speed > speed limit 2 Speed ≤ speed limit 3	Speed range 3	3
Speed > speed limit n−1 Speed ≤ speed limit n	Speed range n	n
Speed > speed limit 8 Speed ≤ speed limit 9	Speed range 9	9
Speed > speed limit 9	Speed range 10	10

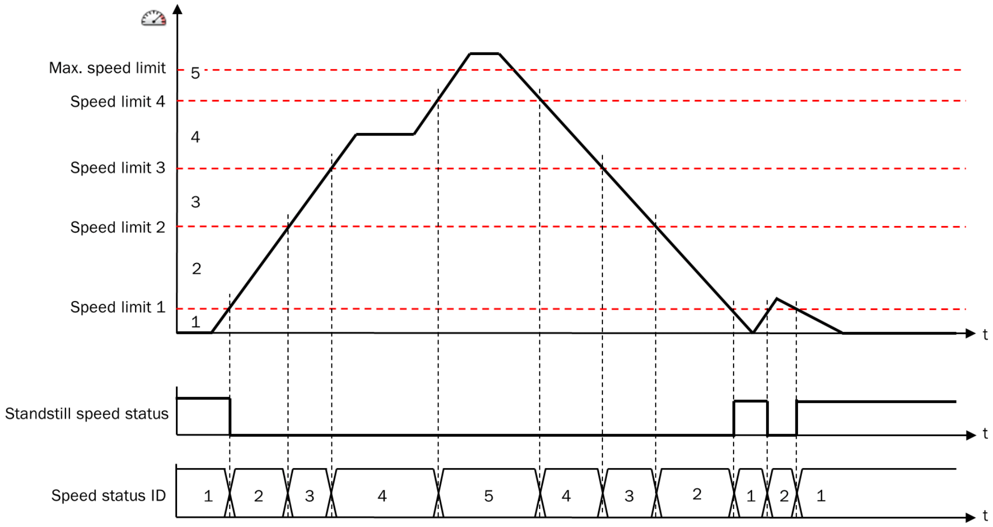


Figure 212: Sequence/timing diagram for the Speed status ID output (example involving four configured speed limits, i.e., five speed ranges)

NOTE
The configured value for the parameter **Max. speed** does not affect the speed status ID, i.e., the **Speed status ID** does not change even if the **max. speed** is exceeded.

Monitoring functions

When the Flexi Soft system transitions to the Run status, the **Monitoring status** output is 1. As long as no errors occur, the value of the output does not change.

The **Monitoring status** output is 0 if one of the following conditions is met:

- Maximum speed monitoring produces a result of 0.
- Speed limit monitoring (selected via the **Speed enable ID** input) produces a result of 0.
- Direction monitoring produces a result of 0.

The Monitoring status output switches back to 1 if at least one of the following conditions is met:

- All associated monitoring processes produce a result of 1.
- The speed at the **Motion in** input is invalid.

The **Monitoring status** output is usually connected to the **Safe stop 2A** input of the **Safe stop V1** function block. This means that an impermissible speed or direction will result in a stop.

Speed monitoring functions

The functions for monitoring the **Max. speed** and the **Speed limits** can be used to implement the safely-limited speed (SLS) function. For transitions from a higher to a lower speed limit, **Speed ramps** can be configured.

Maximum speed monitoring

Max. speed monitoring is always active. If the current speed is greater than the **Max. speed** that has been configured, the **Monitoring status** output switches to 0. When used in conjunction with a **Safe stop V1** function block, this allows the maximum travel/maximum time for a Safe stop to be determined reliably.

The **Max. speed enabled** output is set to 1 when the highest configured speed limit is activated via the **Speed enable ID** input. This output can be used as a reset condition at the **Reset** input of a subsequent **Safe stop V1** function block. If, for example, the **Max. speed** is enabled as long as a safety door remains closed, you can reset a stop ramp that has been triggered by closing the safety door.

Monitoring of the speed limits

The optional **Speed enable ID** input activates the permissible speed limit. The input will accept a UI8 value (0 … 255). To connect the input to Boolean signals, use the **Bool to UI8 V1** function block.

If the current speed at the **Motion in** input is greater than the active speed limit, then the **Monitor status** output is set to 0.

NOTE

- The values 0 and 1 at the **Speed enable ID** input activate standstill monitoring. If neither standstill speed monitoring nor standstill position monitoring is activated, then the **Monitoring status** output always remains at 0 (error).
- Any value above the number of configured speed limits activates the maximum permitted speed.

Speed ramps

The **Ramp speed transitions** parameters can be used to define up to four speed ramps. This allows the current speed limit to be reduced evenly from a higher to a lower speed limit in accordance with the configured increment instead of switching to the lower speed limit immediately. This happens regardless of the actual current speed, i.e., even if the actual speed is already below the new lower speed limit.

Up to four speed ramps with different increments can be defined. A ramp can be selected using the **Ramp selection 1** and **Ramp selection 0** inputs.

Table 127: Speed ramp selection

Input values		Selected ramp
Ramp selection 1	Ramp selection 0	
0	0	Ramp speed transition 1 (fastest ramp)

Input values		Selected ramp
Ramp selection 1	Ramp selection 0	
0	1	Ramp speed transition 2
1	0	Ramp speed transition 3
1	1	Ramp speed transition 4 (slowest ramp)

NOTE
Any change to the input values also affects any speed ramp that happens to be active when this change occurs.

The **Delay time until start of ramp** parameter specifies the delay time that must elapse before the speed ramp commences. This makes it possible, for example, to tolerate a delay in the drive response that is caused by communication and processing cycle issues.

While a speed ramp is active, the **Ramp active** output remains set to 1.

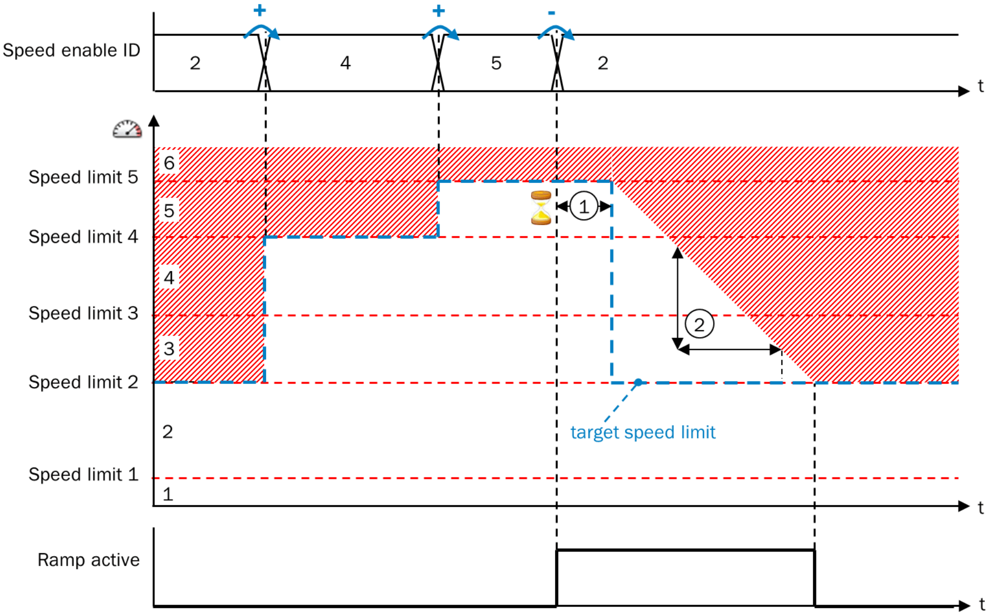


Figure 213: Example 1 for monitoring the speed limit

① Delay time until start of ramp

② Speed transition ramps

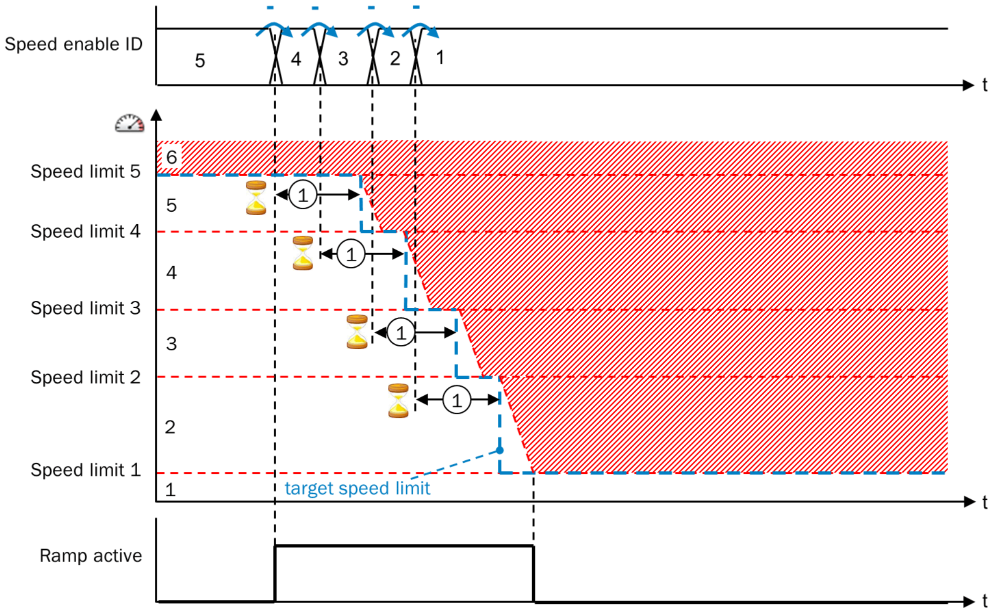


Figure 214: Example 2 for monitoring the speed limit

① Delay time until start of ramp

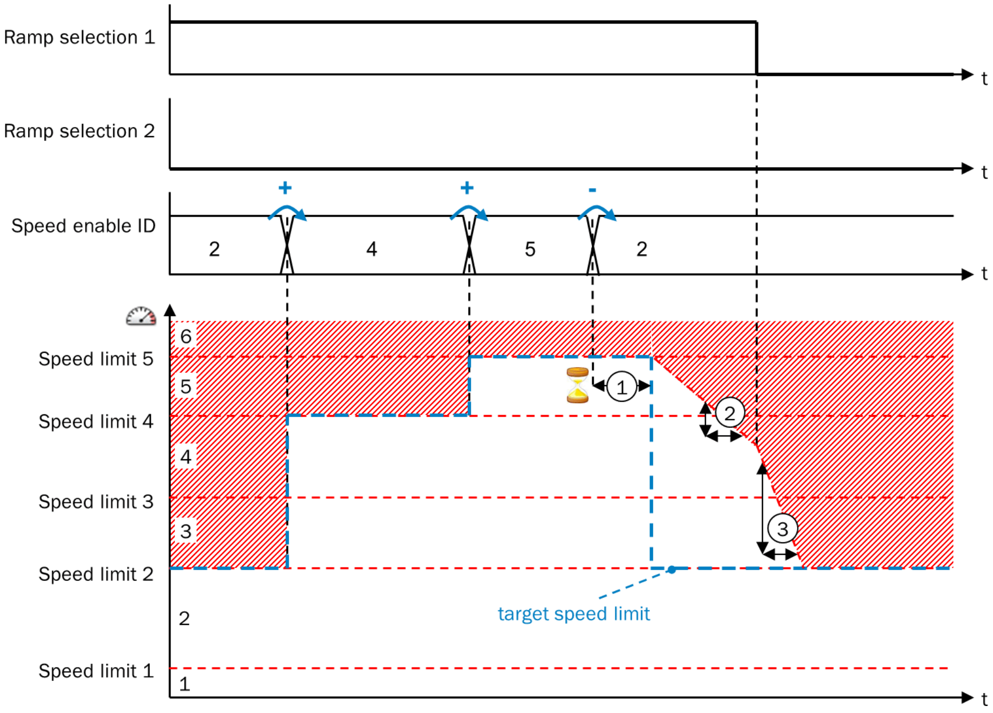


Figure 215: Speed ramp selection

① Delay time until start of ramp

② Ramps speed transition 2

③ Ramps speed transition 1

NOTE

The internal value of the current speed limit can be tracked in the online monitor of the logic editor in the FX3-MOCx and recorded in the data recorder.

Standstill detection

Standstill detection can be used to implement the safe operating stop (SOS) function.

The **Standstill** output switches to 1 and the **Speed status ID** is set to 1 if the conditions for either standstill detection with standstill speed or standstill position are fulfilled.

Standstill detection with standstill speed

Standstill detection with standstill speed takes effect when the following conditions are fulfilled:

- The **Standstill approval** input is 1 or deactivated.
- The speed at the **Motion in** input has the status 1 (valid) and is uninterrupted and remains lower than the standstill speed for at least the duration of the **Standstill speed acceptance time.**

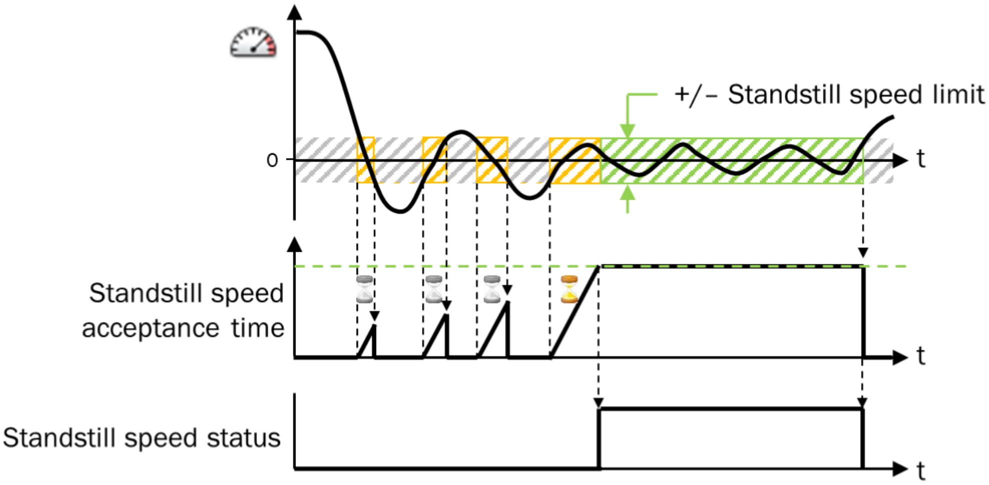


Figure 216: Standstill detection with standstill speed acceptance time

NOTE
The **Standstill speed acceptance time** is also taken into account at each change of the speed value status bit from 0 (invalid) to 1 (valid) at the **Motion in** input. This also applies when the Flexi Soft system transitions to the Run status.

Standstill detection with standstill position tolerance

A **Standstill position tolerance** can be defined. This requires a signal to be present at the **Motion in** input that contains a valid relative position (e.g., from an A/B incremental, Sin/Cos, or SSI encoder).

Standstill detection with standstill position tolerance takes effect when the following conditions are fulfilled:

- The **Standstill approval** input is 1 or deactivated.
- The speed at the **Motion in** input has the status 1 (valid) and has reached a value of 0 three times or has changed sign (i.e., has exceeded the zero line).
- The corresponding relative position values at the **Motion in** input have the status 1 (valid) and are within the two potential **Standstill position tolerance** values.

Once this happens, the current speed is no longer taken into account even if it is greater than the **Standstill speed.** This remains the case until the **Standstill position tolerance** is exceeded or the **Relative position status** at the **Motion in** input switches to 0 (invalid).

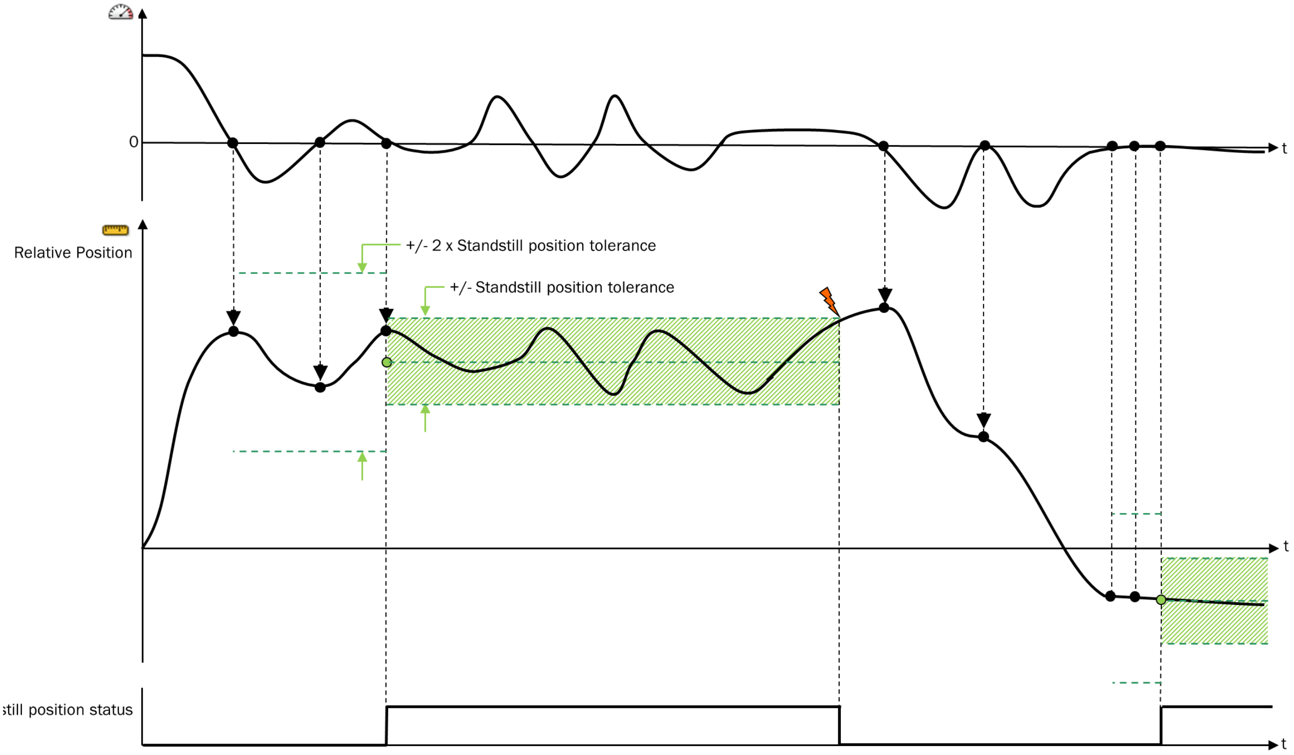

0
t
Relative Position
+/- 2 x Standstill position tolerance
+/- Standstill position tolerance
t
still position status
t

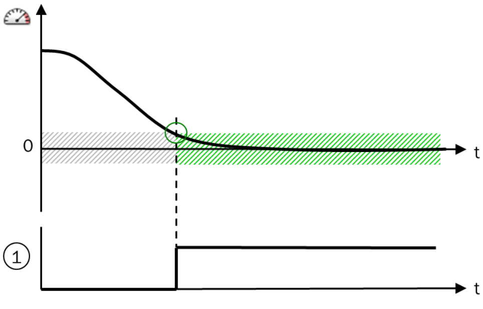
Figure 217: Standstill detection with standstill position tolerance

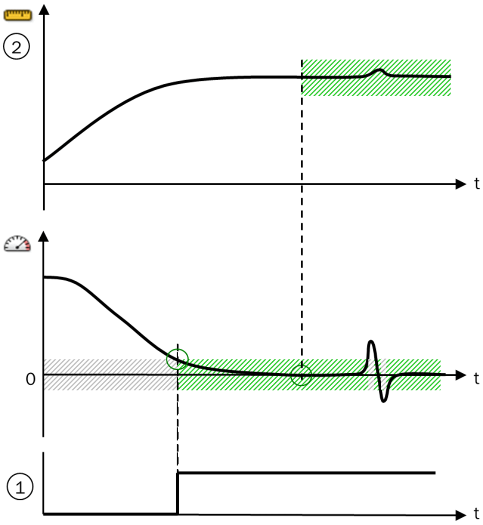
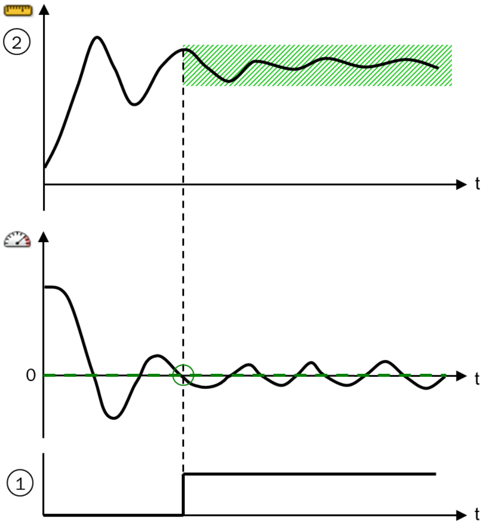
NOTE

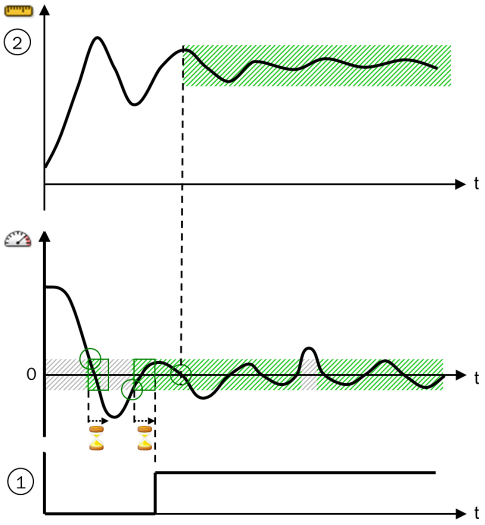
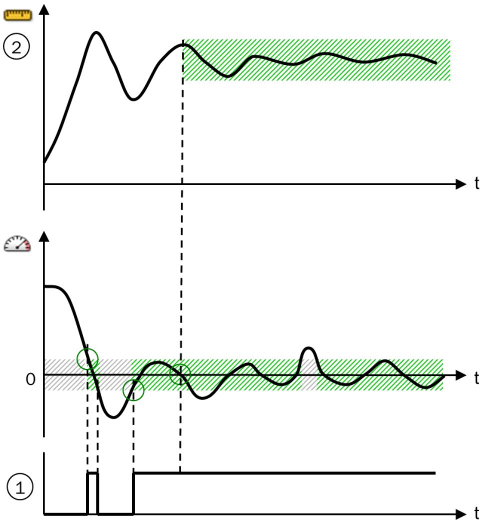
If you are using a **Standstill position tolerance** with the **Standstill speed** deactivated, the position must be maintained and the outputs **Standstill** and **Speed status ID** must remain set to 1 only for as long as the position remains within the **Standstill position tolerance.** Even if the speed moves away from the **Standstill position tolerance** value at a very slow speed, the standstill condition is no longer met.

Configuration examples for standstill

Table 128: Configuration examples for standstill

Speed curve	Description and configuration
 ① Standstill	Asymptotic approximation of the speed to zero. A safe standstill speed > 0 has been selected so that the standstill status can be achieved as quickly as possible. **Configuration** Standstill speed > 0 Standstill acceptance time = 0 Standstill position tolerance = 0

Speed curve	Description and configuration
 ① Standstill ② Relative position	Asymptotic reduction in speed to zero with possible subsequent peak in speed; e.g., due to mechanical shock. A safe standstill speed > 0 and a safe standstill position tolerance > 0 have been selected so that the standstill status can be achieved as quickly as possible while simultaneously tolerating peaks in speed. **Configuration** Standstill speed > 0 Standstill acceptance time = 0 Standstill position tolerance > 0
 ① Standstill ② Relative position	Reduction in speed with oscillation around zero; e.g., due to positioning control. A standstill position tolerance > 0 has been selected that is regarded as safe so that the standstill status can be achieved as quickly as possible but not until the standstill position tolerance requirement has been met (and not simply as soon as a speed limit is undershot). **Configuration** Standstill speed = 0 Standstill acceptance time = 0 Standstill position tolerance > 0

Speed curve	Description and configuration
 ① Standstill ② Relative position	Reduction in speed with oscillation around zero, e.g., due to positioning control. The safe standstill speed > 0 and safe standstill position tolerance > 0 have been selected so that the standstill status can be achieved as quickly as possible while also tolerating peaks in speed. To prevent the standstill status from being triggered as soon as the speed drops to zero for the first time, a standstill speed acceptance time > 0 has also been selected. **Configuration** Standstill speed > 0 Standstill speed acceptance time > 0 Standstill position tolerance > 0
 ① Standstill ② Relative position	Reduction in speed with oscillation around zero; e.g., due to positioning control. A safe standstill speed > 0 and a safe standstill position tolerance > 0 have been selected so that the standstill status can be achieved as quickly as possible but also while tolerating peaks in speed. In this case, the standstill status may be activated and subsequently deactivated again at the beginning. For this reason, this configuration is **not recommended**. **Configuration** Standstill speed > 0 Standstill acceptance time = 0 Standstill position tolerance > 0

Standstill acknowledge

The optional **Standstill approval** input can be used to deactivate internal standstill detection. If this input is used, the **Standstill** and **Speed status ID** outputs can only switch to 1 if both the standstill condition is met and the **Standstill approval** input is 1. This applies to standstill detection with **Standstill speed** and to standstill detection with **Standstill position tolerance.**

The internal standstill detection with **Standstill speed** and, if applicable, **Standstill speed acceptance time** and **Standstill position tolerance** functions independently of the **Standstill approval** input.

Direction detection

The **Direction status** output indicates the direction of movement:

0 = forward (positive speed) or status unknown (speed at **Motion in** input is invalid)

1 = backward (negative speed)

When the Flexi Soft system transitions to the Run status, the **Direction status** output is 0.

The direction status does not change in the event of a standstill. This means that when a movement is performed in one direction with intermediate stops, the indicated direction does not change.

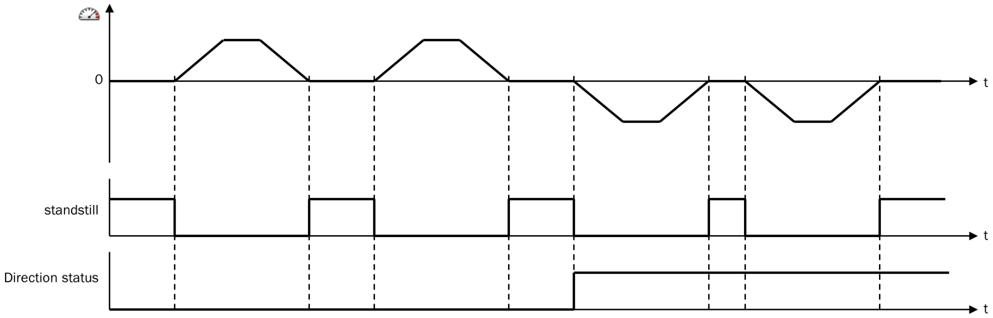


Figure 218: Sequence/timing diagram for direction status

Safe direction (SDI)

The optional **Enable forward** and **Enable backward** inputs can be used to enable the permissible direction of movement. If the current status is not standstill (**Standstill** output is 0) and if the current direction of movement is not enabled, the **Monitor status** output switches to 0.

If either of these inputs is not used, it means that the associated direction of movement is permanently enabled.

9.10.3 Safe Stop V1

Function block diagram

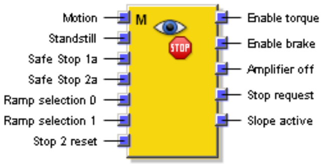


Figure 219: Inputs and outputs of the Safe stop V1 function block

General description

The **Safe Stop V1** function block is used to trigger and monitor the safe stop of a drive system. The drive has to be shut down in a controlled manner. The braking torque of the drive can be used to bring the drive to a standstill in a shorter period of time than an uncontrolled stop could.

The stop ramp of a drive system is not usually safe. For this reason, the **Safe stop V1** function block monitors the actual reduction in speed until the drive comes to a standstill.

Inputs of the Safe stop V1 function block

Table 129: Inputs of the Safe stop V1 function block

Input	Description	Signal value
Motion in	Data of type Motion V1 is expected, either directly from an encoder or from another function block (see "Data types used in the logic of the FX3-MOC0", page 253).	Data of type Motion V1
Standstill	Indicates whether the standstill condition is fulfilled. Generally connected to the **Standstill** output of the Speed monitoring V1 function block	0 = No stop 1 = Standstill

Input	Description	Signal value
Safe stop 1A and Safe stop 1B	To trigger the Safe stop 1 function	Falling signal edge (1–0)
Safe stop 2A and Safe stop 2B	To trigger the Safe stop 2 function	Falling signal edge (1–0)
Ramp selection 0 and Ramp selection 1	Selection of up to four stop ramps with different gradients	0 or 1
Stop 2 reset	Optional input for resetting the function block after a Safe stop 2	Rising signal edge (0–1)

Outputs of the Safe Stop V1 function block

The drive system can be controlled via the **Enable torque**, **Enable brake**, **Amplifier off**, and **Stop request** outputs of the function block.

Table 130: Outputs of the Safe Stop V1 function block

Output	Description	Signal value
Enable torque	Switches off the torque of the drive system, e.g., via the **External device monitoring** function block or (if available) via the safety capable inputs on the drive system for switching off the torque.	0 = Switch off 1 = Enable
Enable brake	Switches off the voltage supply for the mechanical brake (where applicable), e.g., via the **External device monitoring** function block.	0 = Switch off 1 = Enable
Amplifier off	Triggers switch-off of the amplifier and the drive torque plus – where applicable – engagement of the brake.	0 = Switch off 1 = Enable
Stop request	Triggers the stop ramp of the drive.	0 = Stop requested 1 = No stop
Slope active	Indicates whether a stop ramp is active.	0 = No ramp 1 = Ramp active

The **Amplifier off** and **Stop request** outputs allow, for example, the drive system to be informed of the imminent safety response, giving it a chance to respond itself in a controlled manner before being switched off by the safety path.

Function block parameters

Table 131: Parameters of the Safe Stop V1 function block

Parameter	Description	Possible values
Stop ramps		
Delay time until start of ramp	Amount of time by which the beginning of the stop ramp should be delayed to take into account the drive system response time in the event of a stop request	0 … 248 ms in 4 ms increments
Stop ramp speed offset	Optional speed addition to the start value of the stop ramp. Prevents the stop ramp from being exceeded accidentally, e.g., due to mechanical vibrations.	**0 … 32,767 digits =** • 0.5 to 16,383 U/min • 1 to 32,767 mm/s • 0 = Deactivated
Steepness of stop ramp 1 to 4	Increments of speed reduction You can define up to four different ramps. The **Speed reduction** and the **Duration of the speed reduction** are entered.	**0 … 2,147,418,112 digits =** • 0.5 … 16,383 (rpm)/ms • 1 … 65,535 (mm/s)/ms • 0 = No ramp
Off delays for Safe stop 1		
Off delay for Enable brake	Off delay for deactivating Enable brake based on when the **Amplifier off** output is switched off	0 … 248 ms in 4 ms increments
Off delay for Enable torque	Off delay for deactivating Enable torque based on when the **Amplifier off** output is switched off	0 … 248 ms in 4 ms increments

Description of operation

Drive systems usually have various “escalation levels”. The **Safe Stop V1** function block is used to implement the higher escalation levels.

Table 132: Typical escalation levels of a drive system

Level	Possible trigger	Control actions (not safe)	Safety functions
1	• Access to hazardous area required (e.g., warning field of an electro-sensitive protective device interrupted)	PLC reduces the control value for the speed of the drive, e.g., via fieldbus	Monitoring of the speed ramp by the **Speed monitor V1** function block
2	• Speed exceeds speed ramp • Protective field of an electro-sensitive protective device interrupted • Emergency stop pushbutton pressed	Drive system travels along stop ramp, e.g., using a digital input	Monitoring of the stop ramp (Safe Stop 1 or Safe Stop 2) by the **Safe Stop V1**. function block. The stop ramp is typically faster than the speed ramp associated with the **Speed monitor V1** function block.
3	• Speed exceeds stop ramp	Brake engages, drive system amplifier switched off	Deactivation of braking force energy and drive energy (**Torque off**), either via a cable or by using the inputs for switching off the torque on the drive

Activation of the outputs on system start

When the Flexi Soft system transitions to the Run status, all outputs except Ramp active switch to 1 if the following conditions are fulfilled:

- The speed at the **Motion In** input is valid.
- All of the used **Safe stop** inputs are 1.

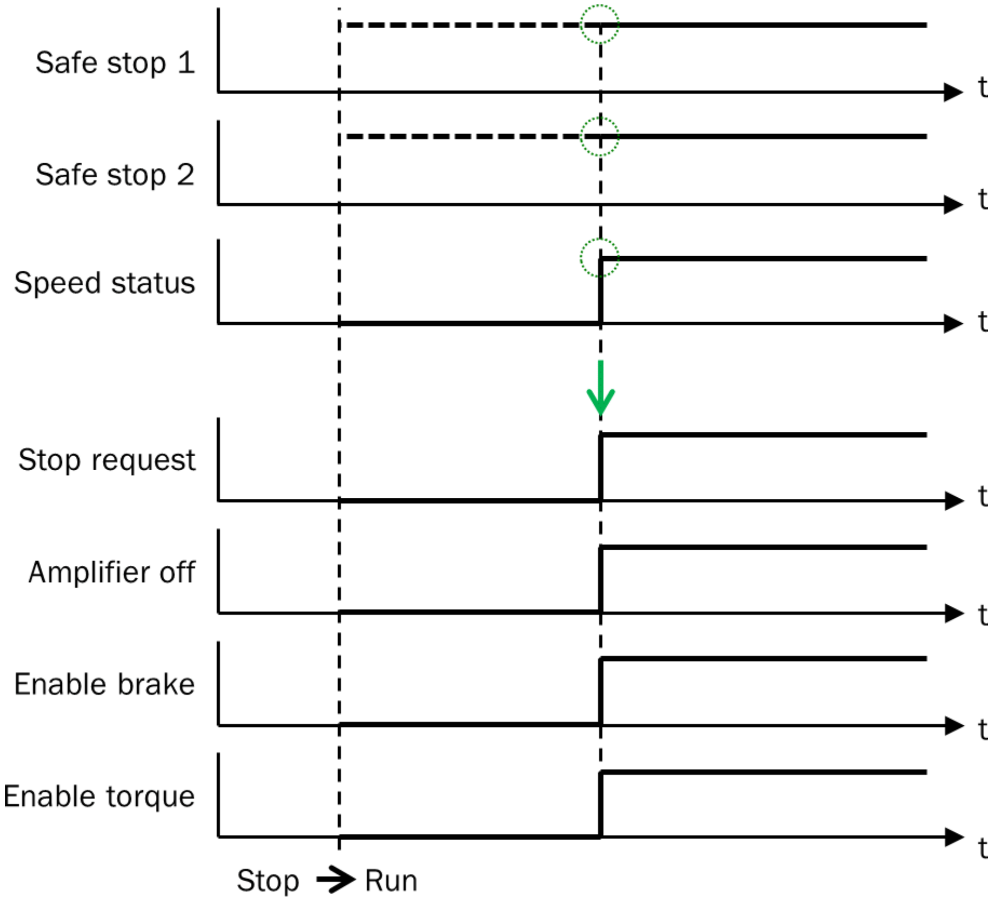


Figure 220: Conditions for activating the outputs

Safe stop 0, Safe stop 1, and Safe stop 2

The **Safe stop V1** function block supports three stop categories as defined in IEC 61800-5-2 and IEC 60204-1.

In the case of the Safe stop 0 (SS0) function, the drive system torque is switched off immediately. The function block executes a Safe stop 0 when the stop ramp conditions are not fulfilled or when the stop ramp cannot be monitored because the speed at the **Motion In** input is invalid.

Stop category 1 and stop category 2 differ in terms of how the stop ramp ends. In the case of the Safe stop 1 (SS1) function, the drive system torque is switched off after the drive has come to a standstill.

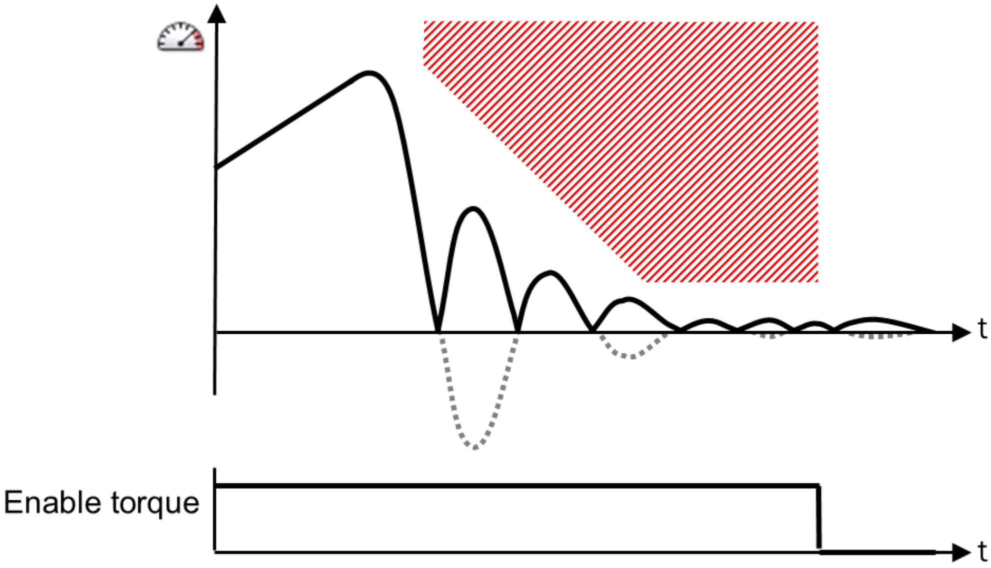


Figure 221: Principle of operation for Safe Stop 1

NOTE
The Safe stop 1 function corresponds to a controlled stop in accordance with IEC 60204-1 stop category 1.

In contrast to this, the Safe Stop 2 (SS2) function keeps the torque enabled, although the standstill condition is monitored. This enables the drive to perform holding control.

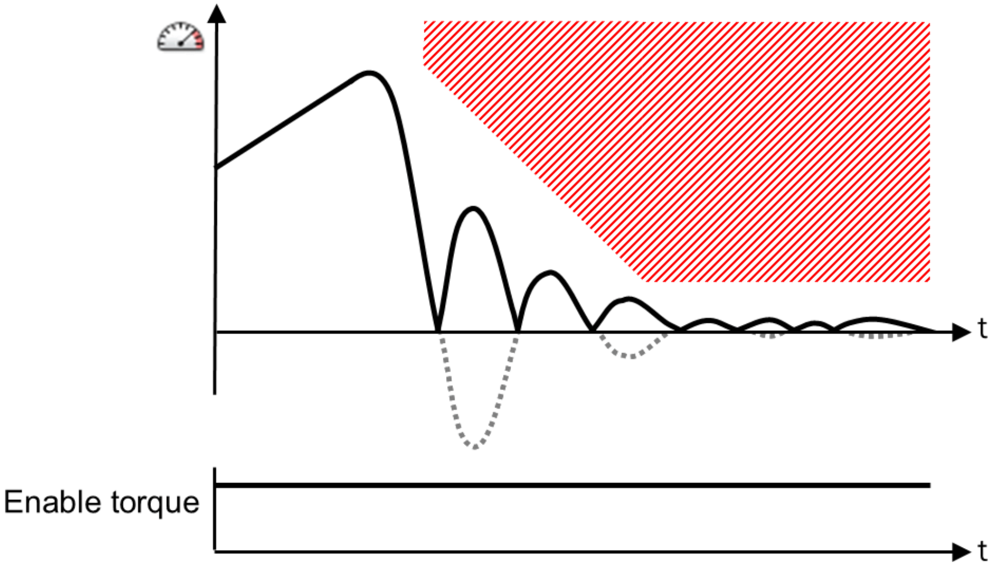


Figure 222: Principle of operation for Safe Stop 2

NOTE
The Safe stop 2 function corresponds to a controlled stop in accordance with IEC 60204-1 stop category 2.

The two stop categories are divided into the following phases:

Table 133: Phases of Safe Stop 1 and Safe Stop 2

Phase	Safe Stop 1	Safe Stop 2
1	Wait for stop request	
2	Delay time for beginning of stop ramp	
3	Monitoring of the stop ramp	
4	Temporary standstill monitoring after Safe Stop 1	Permanent standstill monitoring after Safe Stop 2
5	Switch off torque	

9.10.3.1 Safe stop 1

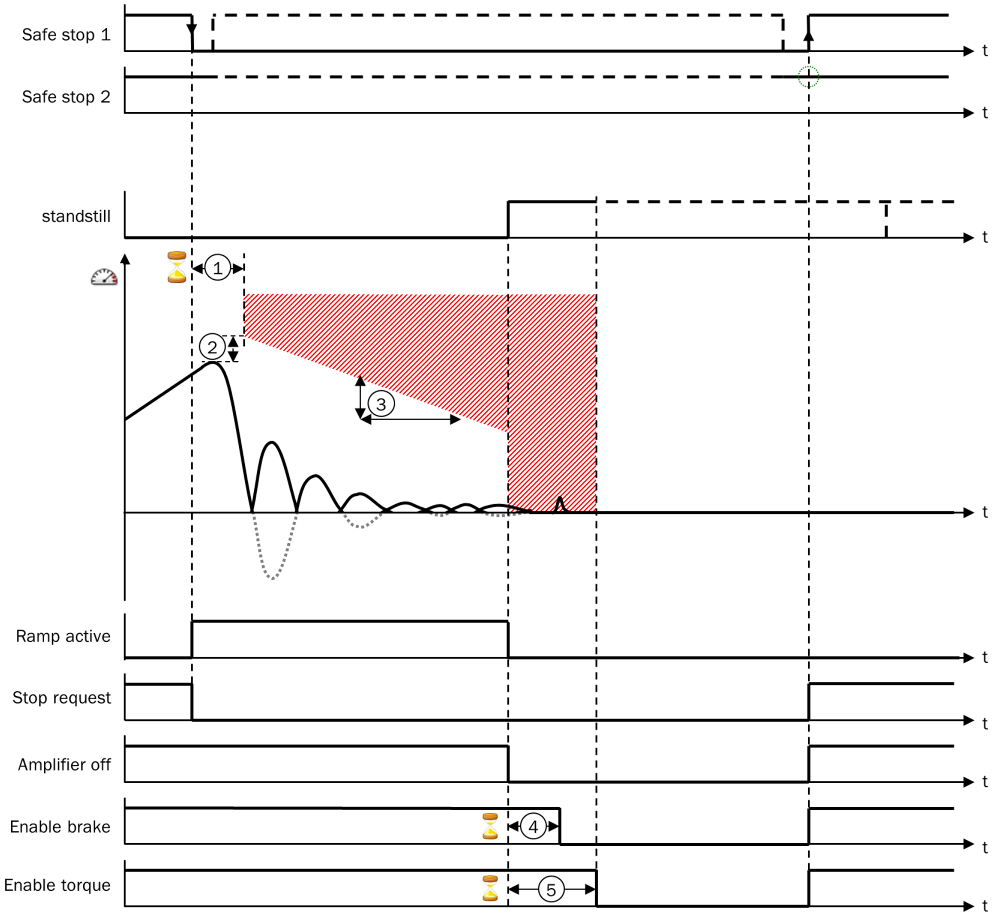


Figure 223: Monitoring function of Safe stop 1

① Delay time until start of ramp

② Stop ramp speed offset

③ Steepness of stop ramp 1 … 4

④ Off delay for Enable brake

⑤ Off delay for Enable torque

For detailed information about the configurable parameters of the **Safe stop V1** function block: see table 131, page 281.

Phase 1: Wait for stop request

The **Safe stop V1** function block offers two optional inputs for each stop mode. A falling signal edge at any of these inputs triggers the corresponding stop mode, i.e., the delay time for the stop ramp starts running. If a Safe stop 2 occurs first and then a Safe stop

1 is also triggered during one of the subsequent phases, the Safe stop 1 function has priority. This means that phase 5 of the Safe stop 1 function (switch off torque) will definitely be triggered.

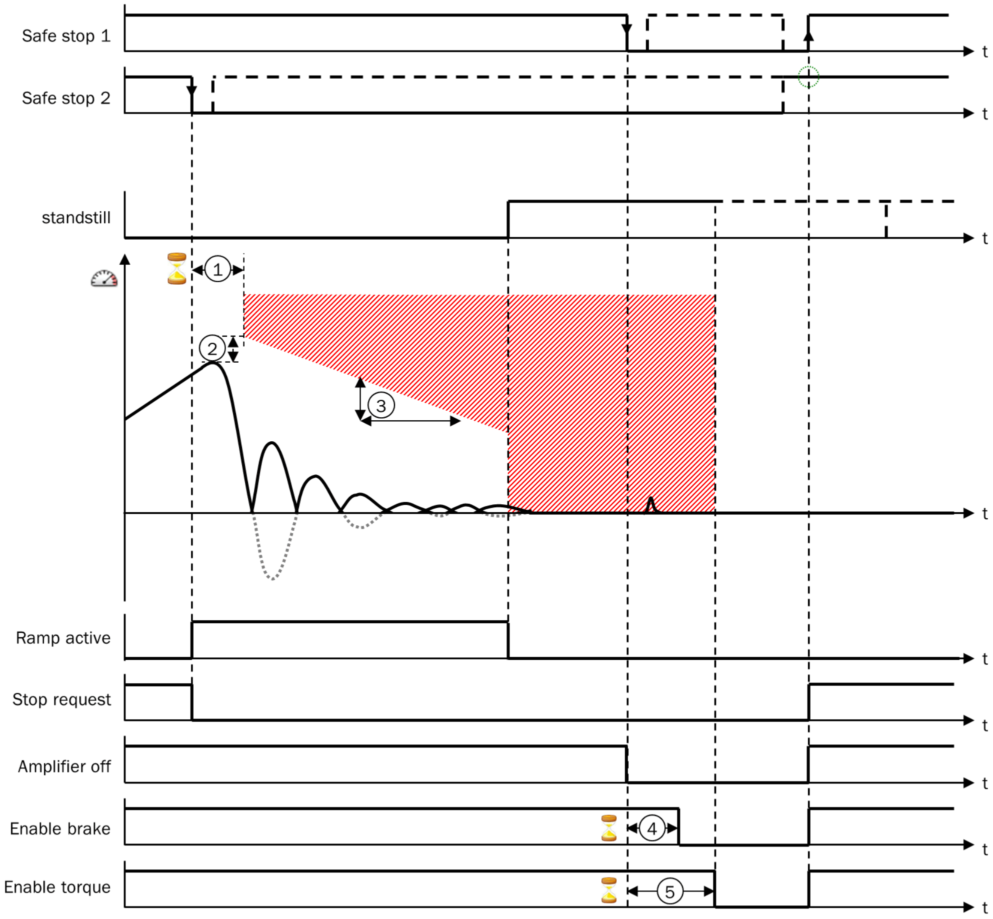


Figure 224: Safe stop 1 after Safe stop 2

① Delay time until start of ramp

② Stop ramp speed offset

③ Steepness of stop ramp 1 ... 4

④ Off delay for Enable brake

⑤ Off delay for Enable torque

For detailed information about the configurable parameters of the **Safe stop V1** function block: see table 131, page 281.

As soon as a stop is triggered, the **Stop request** output is set to 0. This output should be used to trigger the stop ramp of the drive system. This is generally a non-safe signal.

Phase 2: Delay time for beginning of stop ramp

The **Delay time until start of ramp** parameter can be used to configure a delay time for the beginning of the stop ramp. This should correspond to the amount of time it takes the drive system to respond to a stop request.

During this phase, the highest absolute speed is measured and used as the basis for the stop ramp start value. If no delay time has been configured (**Delay time until start of ramp** = 0), the current speed at the time of triggering is used as the start value instead.

At the start of this phase, the **Ramp active** output is switched to 1.

Phase 3: Monitoring of the stop ramp

The value of the **Stop ramp speed offset** parameter is added to the highest absolute speed (i.e., without a sign) that was measured during phase 2. The total is used as the start value for the stop ramp. In this way, the stop ramp is adapted to the current speed.

Monitoring of the stop ramp means that speed limiting begins with the start value and is then constantly reduced in accordance with the **Ramp steepness** parameter.

Once the drive has come to a standstill (standstill condition met, **standstill** input switches to 1), the stop ramp is terminated. This means that the system does not have to wait until the end of the stop ramp if the drive comes to a standstill before the end of the maximum permissible time. The standstill condition is usually monitored using the **Speed monitor V1** function block.

Up to four stop ramps with different increments can be defined. A ramp can be selected using the **Ramp selection 1** and **Ramp selection 0** inputs.

Table 134: Selection of the stop ramp

Input values		Selected ramp
Ramp selection 1	Ramp selection 0	
0	0	Ramp steepness 1 (fastest ramp)
0	1	Ramp steepness 2
1	0	Ramp steepness 3
1	1	Ramp steepness 4 (slowest ramp)

NOTE
Any change to the input values also affects any stop ramp that happens to be active when this change occurs.

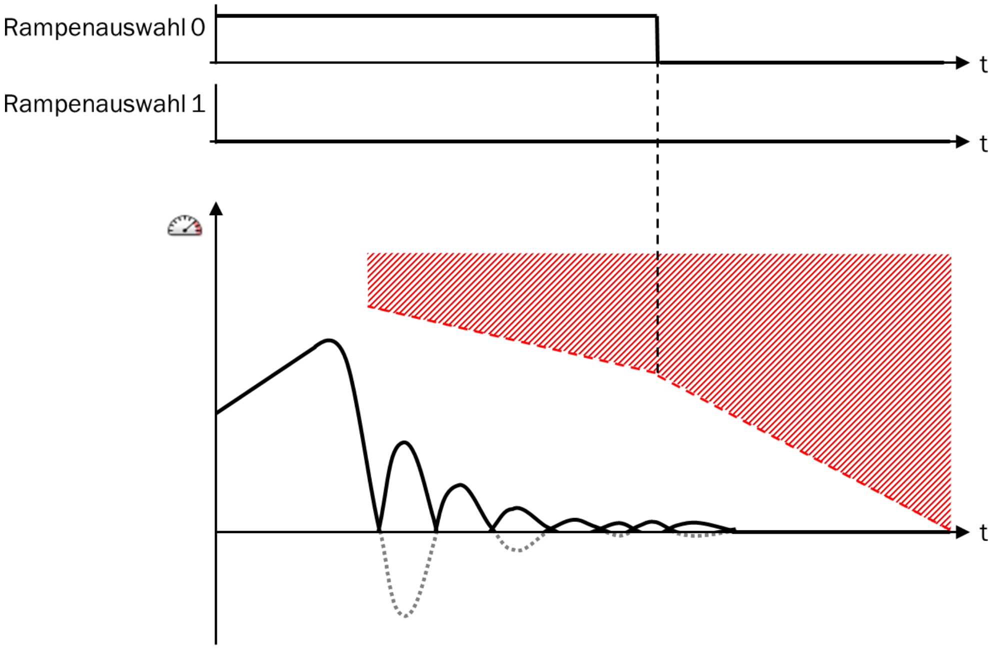


Figure 225: Selection of the stop ramp

For detailed information about the configurable parameters of the **Safe stop V1** function block: see table 131, page 281.

The **Ramp active** output is set to 1 during the delay time for the beginning of the stop ramp and also while stop ramp monitoring is being performed.

Phase 4 of Safe stop 1: Temporary standstill monitoring after Safe stop 1

Phase 4 begins when the drive reaches a standstill, i.e., when the standstill condition is met. In this status, the **Amplifier off**, **Enable brake**, and **Enable torque** outputs are all set to 0. An optional delay can be applied for the purpose of deactivating the **Enable brake** and **Enable torque** outputs.

- The **Amplifier off** output is deactivated immediately.
- Deactivation of the **Enable brake** output is delayed by the amount of time defined by the **Off delay for enable brake** parameter.
- Deactivation of the **Enable torque** output is delayed by the amount of time defined by the **Off delay for enable torque** parameter.

If the drive system is equipped with a brake, the **Off delay for enable torque** parameter is usually set to a higher value than the **Off delay for enable brake** parameter, i.e., the torque is only switched off once the brake has been triggered. This is particularly useful for applications that involve heavy loads and where the torque is required to maintain the position so that the weight of the load does not cause the axis to move. In this case, the drive must be blocked by the brake before the torque is switched off.

Once the **Off delay for enable brake** and **Off delay for enable torque** have elapsed, phase 5 (**Switch off torque**) commences.

During phase 4, the standstill condition at the **Standstill** input and the speed at the **Motion In** input are both monitored. If the **Standstill** input is 0 and the speed at the **Motion In** input is not 0, phase 5 (**Switch off torque**) is triggered immediately. The same applies if the speed at the **Motion In** input is or becomes invalid.

Phase 5: Switch off torque

In phase 5, the **Enable torque**, **Enable brake**, and **Amplifier off** outputs are always deactivated without any further delay.

Resetting of Safe stop 1 during phase 5

During phase 5, the outputs can be reactivated by a rising signal edge at either the Safe stop 1A or Safe stop 1B input, subject to the following conditions:

- All of the **Safe stop X** inputs are set to 1.
- The speed at the **Motion In** input is valid.

The current speed is not taken into account. Consequently, a reset is possible even if the drive is still moving. This also applies to stops that have been triggered because the speed at the **Motion In** input was invalid.

9.10.3.2 Safe stop 2

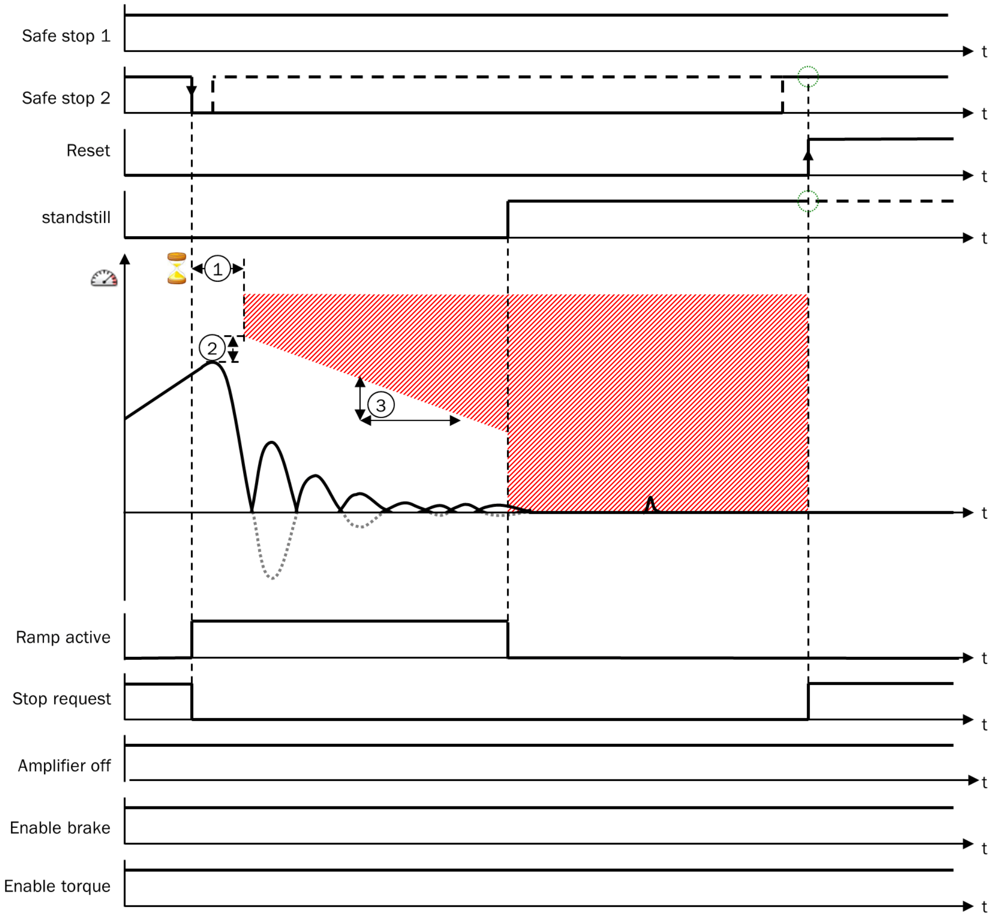


Figure 226: Monitoring function of Safe stop 2

① Delay time until start of ramp

② Stop ramp speed offset

③ Steepness of stop ramp 1 ... 4

For detailed information about the configurable parameters of the **Safe stop V1** function block: see table 131, page 281.

The first three phases of Safe stop 2 are the same as the first three phases of Safe stop 1:

- Phase 1: Wait for stop request (see "Phase 1: Wait for stop request", page 285)
- Phase 2: Delay time for beginning of stop ramp (see "Phase 2: Delay time for beginning of stop ramp", page 286)
- Phase 3: Monitoring of the stop ramp (see "Phase 3: Monitoring of the stop ramp", page 287)

Phase 4 of Safe stop 2: Permanent standstill monitoring after Safe stop 2

Phase 4 begins when the drive reaches a standstill, i.e., when the standstill condition is met. With a Safe stop 2, the **Amplifier off**, **Enable brake**, and **Enable torque** output remain set to 1.

During phase 4, the standstill condition at the **Standstill** input and the speed at the **Motion In** input are both monitored. If the **Standstill** input is 1 and the speed at the **Motion In** input is not 0, phase 5 (**Switch off torque**) is triggered immediately. The same applies if the speed at the **Motion In** input is or becomes invalid.

If a falling signal edge occurs at any point before or during phase 4 at either of the **Safe stop 1** inputs, phase 4 of Safe stop 1 (Temporary standstill monitoring after Safe stop 1) is triggered. This means that a Safe stop 1 always takes priority over a Safe stop 2.

Resetting of Safe stop 2 during phase 4

When the optional Stop 2 reset input is used, a Safe stop 2 can be reset during phase 4 by a rising signal edge at the Stop 2 reset input, if the following conditions are fulfilled:

- All of the used **Safe stop** inputs are 1.
- The **Standstill** input is 1.
- The speed at the **Motion In** input is valid.

If the optional **Stop 2 reset** input is not used, a Safe stop 2 can only be reset by first triggering phase 5 and then ensuring that the conditions for resetting phase 5 are fulfilled.

Exceptions

If the normal sequence is not adhered to, the following exceptional cases may occur:

- If the speed exceeds the speed limit for the stop ramp, the **Amplifier off**, **Enable brake**, and **Enable torque** outputs are deactivated immediately. This is a Safe stop 0 or phase 5 with a Safe stop 1.

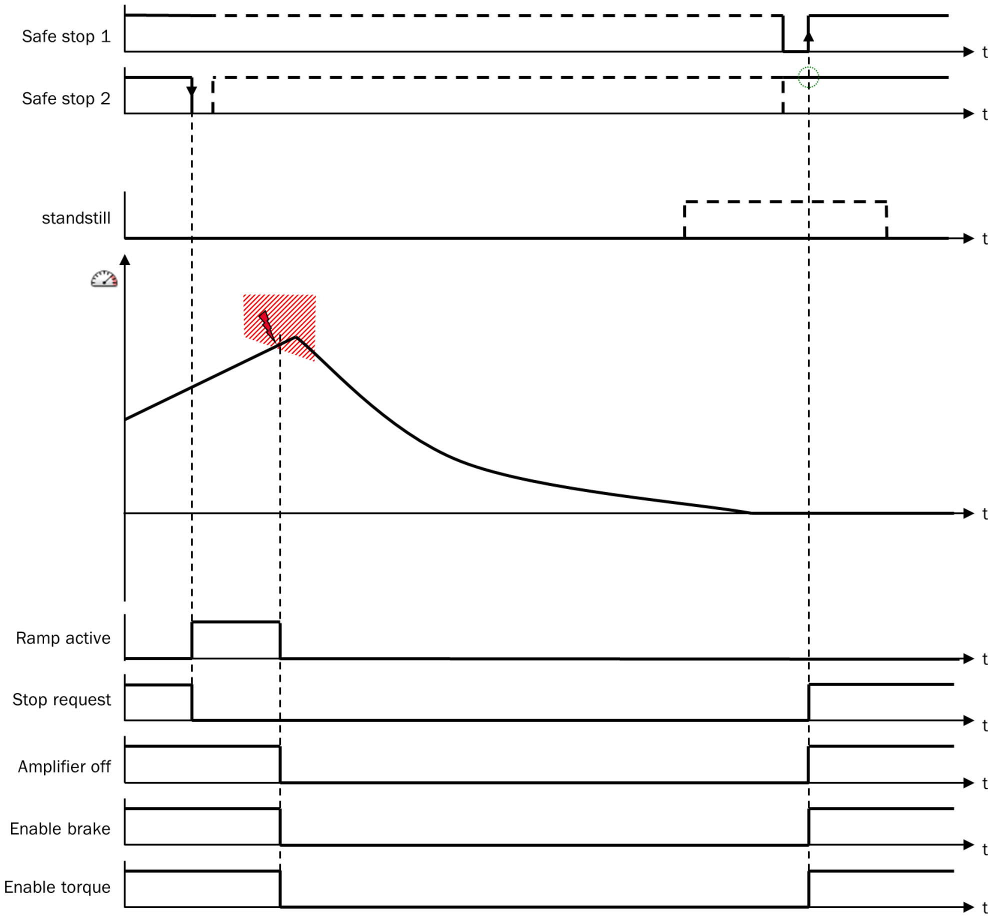


Figure 227: Exception – Stop ramp exceeded

For detailed information about the configurable parameters of the **Safe stop V1** function block: see table 131, page 281.

- If the **Standstill** input switches to 0 at any time during standstill monitoring once a Safe stop 1 or a Safe stop 2 has occurred (i.e., if the standstill condition is not met or stops being met), the **Amplifier off** output is deactivated immediately, and the **Enable brake** and **Enable torque** outputs are deactivated after the configured delay.

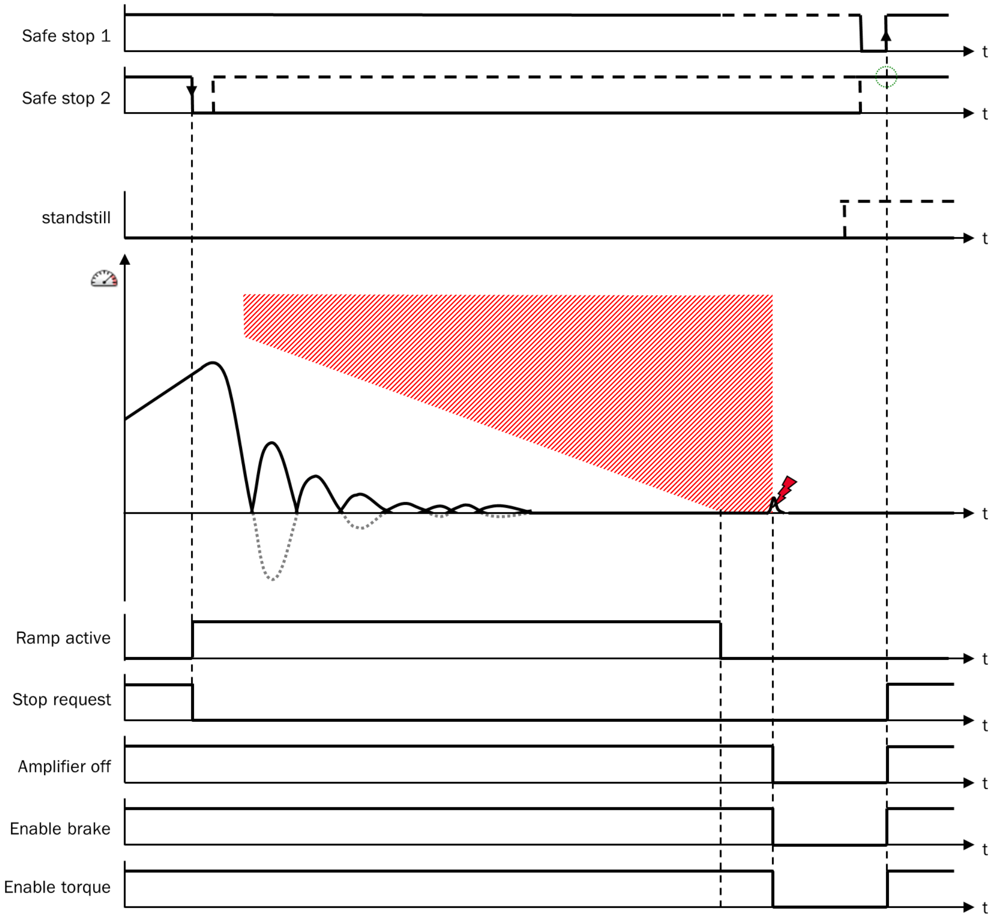


Figure 228: Example 1 for exception – Standstill condition not met during standstill monitoring

For detailed information about the configurable parameters of the **Safe stop V1** function block: see table 131, page 281.

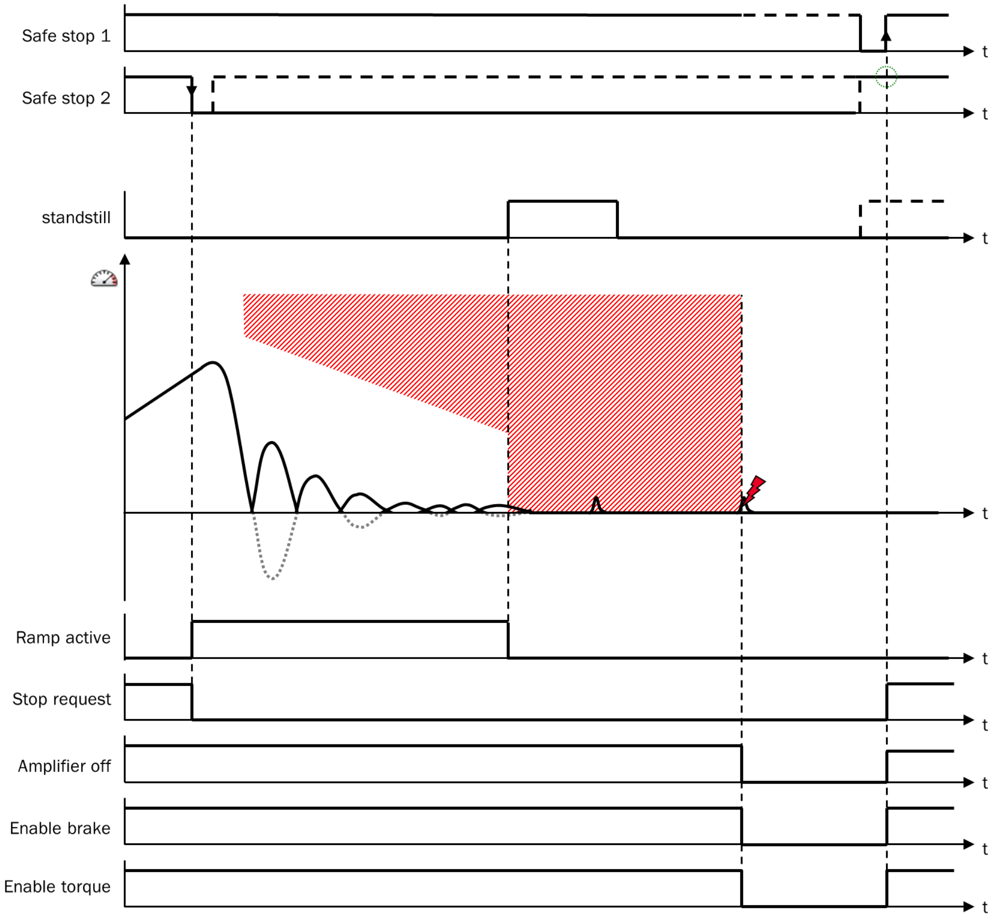


Figure 229: Example 2 for exception – Standstill condition not met during standstill monitoring

For detailed information about the configurable parameters of the **Safe stop V1** function block: see table 131, page 281.

For detailed information about the configurable parameters of the **Safe stop V1** function block: see table 131, page 281.

- If the speed at the **Motion In** input becomes invalid, e.g. due to a monitoring function within a function block located further along the signal path, a Safe stop 1 is triggered. Ramp monitoring is based on the last speed as usual to ensure that the same delay is achieved as for a normal scenario. However, the stop ramp is not terminated prematurely because the standstill condition is ignored. At the end of the stop ramp, all outputs are deactivated immediately, i.e. no additional delay is applied to the **Enable torque** and **Enable brake** outputs.

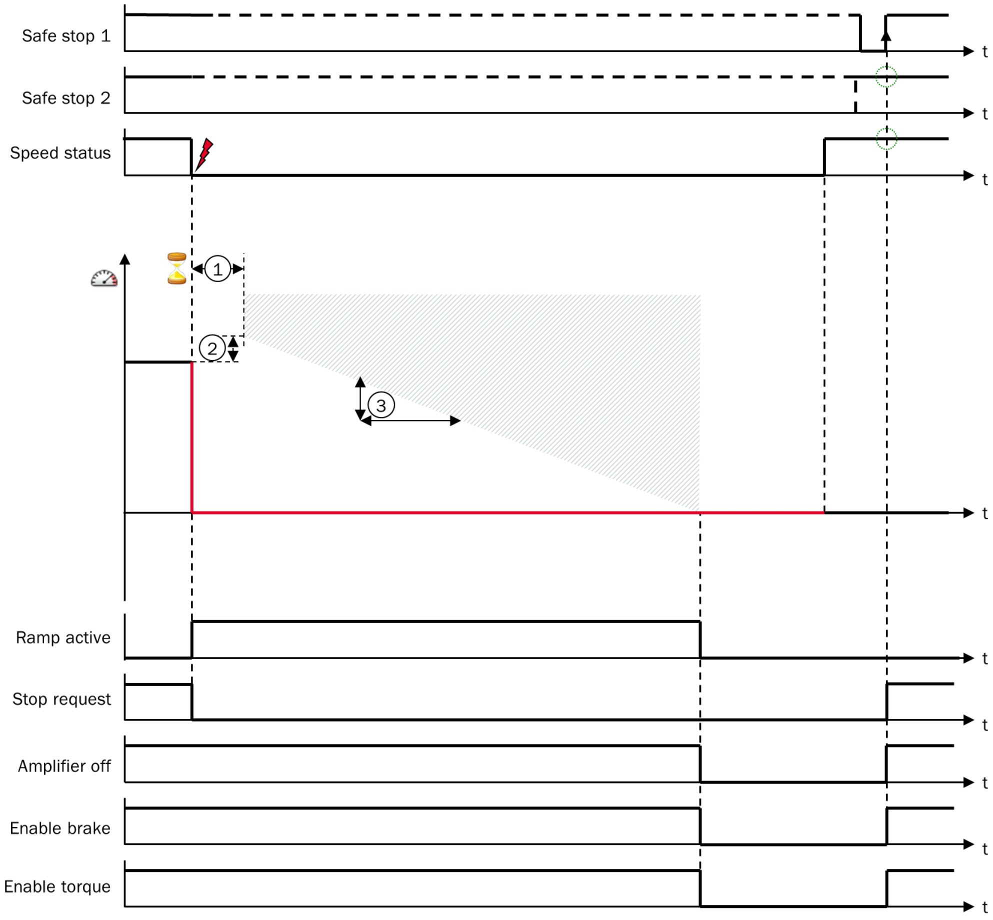


Figure 230: Exception – Invalid speed

For detailed information about the configurable parameters of the **Safe stop V1** function block: see table 131, page 281.

9.11 Function blocks for data conversion

9.11.1 UI8 to Bool V1

Function block diagram

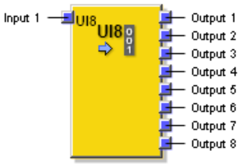


Figure 231: Inputs and outputs of the UI8 to Bool V1 function block

General description

The **UI8 to Bool V1** function block converts an 8-bit integer value (UINT8) at **Input 1** into Boolean. **Output 1** through **Output 8** provide the converted value in Boolean format. All this function does is convert the data type so that the data can be connected to Boolean signals.

Truth table for the UI8 to Bool V1 function block

Table 135: Truth table for the UI8 to Bool V1 function block

Input 1	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	1	1
4	0	0	0	0	0	1	0	0
…	…	…	…	…	…	…	…	…
253	1	1	1	1	1	1	0	1
254	1	1	1	1	1	1	1	0
255	1	1	1	1	1	1	1	1

9.11.2 Bool to UI8 V1

Function block diagram

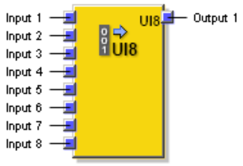


Figure 232: Inputs and outputs of the Bool to UI8 V1 function block

General description

The **Bool to UI8 V1** function block converts an 8-bit Boolean value at **Input 1** through **Input 8** into an integer value (UINT8). **Output 1** provides the converted value as an integer. All this function does is convert the data type so that the data can be connected to Boolean signals.

Truth table for the Bool to UI8 V1 function block

Table 136: Truth table for the Bool to UI8 V1 function block

Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1	Output 1
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	1	1	3
0	0	0	0	0	1	0	0	4
…	…	…	…	…	…	…	…	…
1	1	1	1	1	1	0	1	253
1	1	1	1	1	1	1	0	254
1	1	1	1	1	1	1	1	255

9.11.3 Motion status to Bool V1

Function block diagram

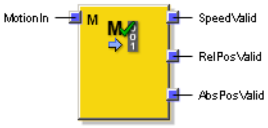


Figure 233: Inputs and outputs of the Motion Status to Bool V1 function block

General description

The **Motion Status to Bool V1** function block converts the **Speed status**, the **Relative position status** and the **Absolute position status** at the **Motion In** input to Boolean values. All this function does is convert the data type so that the data can be connected to Boolean signals.

NOTE

- This function block is available with FX3-MOCx modules featuring a firmware version ≥ V1.10.0.
- Absolute position data is not supported by FX3-MOC0 modules. Consequently, the **Absolute position valid** output is always 0 = invalid for these modules.

Outputs of the function block

Table 137: Outputs of the Motion status to Bool V1 function block

Output	Value	Meaning
SpeedValid	0	Invalid speed
	1	Valid speed
RelPosValid	0	Invalid relative position
	1	Valid relative position
AbsPosValid	0	Invalid absolute position
	1	Valid absolute position

9.11.4 Speed to Bool V1

Function block diagram

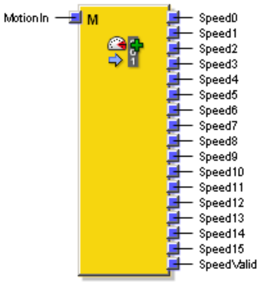


Figure 234: Inputs and outputs of the Speed to Bool V1 function block

General description

The **Speed to Bool V1** function block converts the speed value and the speed status at the **Motion in** input into Boolean values. All this function does is convert the data type so that the data can be connected to Boolean signals, e.g., for the purpose of transmitting the speed value to the main module. The speed status is also output at the **Speed status** output.

NOTE
This function block is available with FX3-MOC0 modules featuring a firmware version ≥ V1.10.0.

Speed bit x outputs

The **Speed bit 15** to **Speed bit 0** outputs correspond to the bits of the speed value at the **Motion in** input, in digits in the internal display and with a sign in binary format (see table 116).

Table 138: Speed bit x output of Speed to Bool V1 function block

Speed value in digits	Binary speed value (Speed bit 15 … 0 outputs)	Resolution for rotary movement type	Resolution for linear movement type
–32,768	1000 0000 0000 0000	1 digit = 0.5 rpm	1 digit = 1 mm/s
…	…		
–1	1111 1111 1111 1111		
0	0000 0000 0000 0000		
1	0000 0000 0000 0001		
…	…		
32,767	0111 1111 1111 1111		

Speed status output

The value of the **Speed status** output corresponds to the speed status at the **Motion in** input (see table 116).

Table 139: Speed status output in the Speed to Bool V1 function block

Value	Meaning
0	Invalid speed
1	Valid speed

9.11.5 Speed to laser scanner V1

Function block diagram

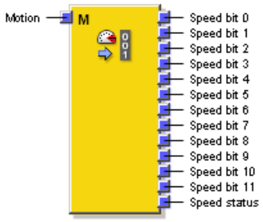


Figure 235: Inputs and outputs of the Speed to laser scanner V1 function block

General description

The **Speed to laser scanner V1** function block converts the speed at the **Motion in** input into a Boolean value with cm/s scaling. The 12 outputs from **Speed bit 11** to **Speed bit 0**, along with the **Speed status** output, are available for this purpose. Each of the **Speed bit x** outputs contains the corresponding bit of the output value that has been calculated for the speed.

This value can, for example, be output to a SICK laser scanner connected via EFI so that the scanner can use it to switch over the monitoring case based on the speed.

ℹ **NOTE**
An encoder with linear movement must be connected to the **Motion in** input. Encoders with rotary movement cannot be used.

Plausibility check

The **Speed to laser scanner V1** function block carries out a plausibility check. If the speed status at the **Motion in** input is 0 (invalid) and the speed does not exceed the range –2,048 to +2,047 cm/s, the **Speed status** output is set to 1 (valid). If either or both of these conditions is/are not met, the **Speed status** output switches to 0 (invalid) and remains in this status until both conditions are met once again.

Speed bit x outputs

The **Speed bit 11** to **Speed bit 0** outputs correspond to the speed value at the **Motion in** input, converted into cm/s and in signed binary format (see table 116).

Table 140: Speed bit x output of Speed to laser scanner V1 function block

Speed in cm/s	Binary speed value (Speed bit 11 … 0 outputs)
–2,048	1000 0000 0000
…	…
–1	1111 1111 1111
0	0000 0000 0000
1	0000 0000 0001
…	…
2,047	0111 1111 1111

9.12 Easy Applications

General description

With Flexi Soft Designer Version ≥ V1.7.1, Easy Applications are available in the logic editor of the FX3-MOCx modules. These are special customized function blocks that have been prepared by SICK ready for immediate use. They have their own logic page and their own configuration dialog. In addition, they feature automatic calculation processes for configuration parameters that are independent of one another and for additional plausibility checks.

Easy Applications make it much easier to configure standard applications. They also make it easier to understand the parameters of the individual function blocks involved in the Easy Application because they rely on familiar terminology and because it is easy to see how changes to the application parameters affect the configuration of the function blocks used.

Different levels can be permitted for configuring and adapting Easy Applications:

- **Protected mode** (default): Only the basic parameters (**Machine parameters** and **Application parameters**) of the application can be configured. On the basis of this information, the parameters of the individual blocks used (**Function block parameters**) are then calculated and applied automatically.
- **Expert Mode**: The **Function block parameters** can be configured by the user. Changes to calculated values are not applied automatically but are displayed as suggestion values. This makes it clear what consequences will ensue if changes are made to the basic parameters for the **Function block parameters.** The suggestion values can either be applied or adapted individually.
- Easy Applications can be converted into grouped function blocks, the logic of which can be freely adapted. However, in this case, the special configuration dialog and extended functionality of the Easy Application will be lost. On the other hand, the logic operations and current configuration of the underlying function blocks will be retained.

NOTE

- Easy Applications carry out extended plausibility checks. This involves checking the plausibility of the configuration across all the blocks. Depending on the result of this check, a warning can be output or the configuration can be marked as defective. In this case, it is not possible to transfer the configuration to the Flexi Soft system.
- The rules for some Easy Applications may depend on parameters that are configured outside of the Easy Application. If changes are made to these parameters, the Easy Application will indicate an error. In this case, the user must reopen the configuration dialog of the Easy Application and confirm the changes in order to reset the error.
- As part of the process of verifying the configuration, the configuration for the function blocks on which the Easy Application is based must be verified in the report. The Easy Application parameters (the **Machine parameters** and the **Application parameters** in the configuration dialog) are not safety-related parameters.

9.12.1 Using Easy Applications

Downloading an Easy Application

1. On the Flexi Soft Designer start screen, click the link on the bottom left for downloading Flexi Soft Designer example projects.
2. Select, download, and run the setup file for the required Easy Application.

✓ The example project is installed in a default folder on the computer along with the associated information ready for use. On the Flexi Soft Designer start screen, click **Load installed SICK example project** on the top left-hand side.

Creating the hardware configuration and connecting the inputs and outputs

Before you can use an Easy Application, you must first create the necessary hardware configuration in Flexi Soft Designer. The easiest way to do this is with the example projects that SICK has included with the software. There are two options:

► Open the example project for the relevant Easy Application that was installed together with the Flexi Soft Designer program. To do this, either click **Load SICK example project** on the start screen or use the **Project - Open** menu command. In the case of standard installation, the example projects are located inside the following folder: **Programs/SICK AG/Flexi Soft Designer/SampleProjects.**

► Open a new project, add a suitable main module, and then load the partial application for the relevant Easy Application that contains the remaining hardware and logic.

In both cases, the Easy Application is preconfigured and ready to use in the logic editor.

NOTE

- The Easy Applications can be found in the logic editor of the Motion Control FX3-MOCx.

Configuring the Easy Application in Protected Mode

To open the configuration dialog, double-click on the Easy Application in the logic editor workspace. In **Protected Mode**, the configuration dialog is exactly the same as for a standard function block except that there are some extra buttons in the footer of the dialog window.

In **Protected Mode**, the various tabs of the configuration dialog only allow you to configure the basic parameters (**Machine parameters** and **Application parameters**) of the application.

Changes made in the **Machine parameters** area may result in changes to parameters in the **Application parameters** area unless these have already been adjusted manually. In this case, the relevant application parameter is highlighted accordingly and the newly calculated value is suggested in a blue speech bubble .

If a tab contains invalid values, it is highlighted by a warning triangle.

If you click the **OK** button while you are in **Protected Mode**, the parameters that have been entered are applied, and the parameters of the underlying blocks are calculated and adapted automatically on the basis of internal rules.

Configuring the Easy Application in Expert mode

Expert Mode is activated in the footer of the configuration dialog.

In **Expert Mode**, the basic parameters behave in exactly the same way as in **Protected Mode**.

In **Expert Mode**, the navigation tree also shows how the **Function blocks** have been grouped. It lists the function blocks used in the Easy Application together with their complete configuration dialogs. The parameters of the function blocks can be edited directly in **Expert Mode**.

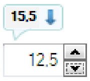
If new parameter values are calculated for the function blocks used, they are not applied automatically in **Expert Mode**. If the suggestion value for a parameter has changed since the configuration dialog was last opened, the parameter is marked with a lightning symbol. This indicates a change in the decision-making basis for the current configuration of the parameter concerned.

If the suggestion value deviates from the currently configured parameter value, it is displayed inside a blue speech bubble above the relevant parameter.

Parameters can be locked, either by entering a value manually or by clicking on the lock symbol next to the parameter.

If you click the **Adopt all values** button in the footer of the configuration dialog, the suggestion values are applied to all the parameters that have not been locked.

Table 141: Identification of function block parameters in Expert Mode

	Speech bubble containing the deviating suggestion value. The suggestion value can be applied by clicking the blue arrow. If the parameter is locked, the process of applying the suggestion value unlocks it. Once the new value has been applied, you can click the lock symbol so that the parameter becomes locked again – this time with the new value. You can hide the suggestion values temporarily by activating the **Hide Suggestion Values** option in the footer of the configuration dialog.
🔓	Not locked. The suggestion value is identical to the current value. You can lock the parameter by clicking the lock symbol or by changing the value manually.

⚡🔓	Not locked. The suggestion value deviates from the current value. The suggestion value is displayed in a speech bubble above the current value. The suggestion value can be applied by clicking the blue arrow inside the speech bubble. You can lock the parameter by clicking the lock symbol.
🔒	Locked. The value has either been changed manually or the current value has been locked. There has been no change in the basis for deciding that the value should be locked since the Easy Application configuration dialog was last opened. If the suggestion value deviates from the current value, the suggestion value is displayed in a speech bubble above the current value. In this case, the suggestion value can be applied by clicking the blue arrow. The process of applying the suggestion value unlocks the parameter. You can lock the parameter again by clicking the lock symbol.
⚡🔒	Locked. The suggestion value deviates from the current value. There has been a change in the basis for deciding that the value should be locked since the Easy Application configuration dialog was last opened. If the suggestion value deviates from the current value, it is displayed above it inside a speech bubble and can be applied by clicking the blue arrow. The process of applying the suggestion value unlocks the parameter. You can lock the parameter again by clicking the lock symbol.

NOTE

The identification symbols and suggestion values can refer to groups of associated function block parameters as well as to individual function block parameters.

All tabs that contain deviating/new suggestion values or locked parameters, or tabs that contain subordinate tabs with deviating/new suggestion values or locked parameters, are marked with the corresponding symbols in the navigation tree of the configuration dialog.

To close the configuration dialog, click the **OK** button. If there are no locked parameters with deviating suggestion values, a user notification will be provided accordingly.

NOTE

It is possible to switch back to **Protected Mode** from **Expert Mode**. However, if parameters have been changed individually in **Expert Mode**, these changes will be lost. The settings will be replaced by the automatically calculated values of the Easy Application.

How to convert the Easy Application into a grouped function block and freely adapt the logic

If you click the **Unlock Logic** button in the footer of the configuration dialog, the Easy Application is converted into a grouped function block that can be freely edited. For example, you can change the wiring and delete existing/add new function blocks.

However, the configuration that applied within the Easy Application for the individual underlying function blocks at the time of conversion will be retained.

The configuration dialog and the extended functionality of the Easy Application (plausibility checks, automatic calculation of suggestion values, etc.) will be lost as a result of the conversion process.

Once an Easy Application has been converted into a grouped function block, this cannot be undone.

9.12.2 Typical procedure for working with Easy Applications

- ► Open the example project that contains the Easy Application or add the Easy Application to an existing project as a subproject.
- ► In **Protected Mode**, adapt the **Machine parameters** and **Application parameters** as required.
- ► If the desired configuration cannot be achieved in **Protected Mode**, switch to **Expert Mode.**
- ► In **Expert Mode**, modify the **Machine parameters** and **Application parameters** as required to ascertain which parameters of the underlying function blocks need to be changed manually. Once the parameters have been changed manually, lock them.
- ► Perform an **Adopt all values** operation for all unlocked parameters. Then, check the values of the locked parameters. In the case of inappropriate values, apply the value suggested by the Easy Application. This unlocks the parameter concerned.
- ► Press **OK** to confirm the configuration.
- ► If further adjustments have to be made to the logic (by adapting the wiring, inserting additional function blocks), use the **Unlock Logic** button to convert the Easy Application into a customized function block and then continue editing it.

10 Encoder in Motion Control FX3-MOC0

To configure an encoder that is connected to the Motion Control FX3-MOC0, select **Edit…** from the encoder's context menu or double-click on the encoder using the left mouse button. The **Element settings** window opens.

For additional information about connecting and configuring encoders, please see the operating instructions titled "Flexi Soft Modular Safety Controller Hardware".

10.1 Functions for all encoder types

The functions described here are available for all types of encoder.

10.1.1 General parameters of the encoder on the FX3-MOC0

Table 142: General parameters of the encoder on the FX3-MOC0

Parameter	Description
Scaling of the measurement system	see "Scaling of the measurement system", page 302
Counting direction	see "Encoder count direction", page 303
Encoder splitter box	see "Encoder connection type and ID code monitoring", page 303
Encoder voltage supply	see "Encoder connection type and ID code monitoring", page 303
Max. step change in speed	see "Monitoring of the maximum step change in speed", page 304

10.1.2 Scaling of the measurement system

NOTE
The information in this section applies to all FX3-MOCx modules.

The scaling of the measurement system defines the ratio between the information from the encoder and the mechanically driven part (number of increments per revolution or per millimeter, depending on type of movement).

On the basis of this scaling, the information supplied by the encoder is converted so that the internal motion signal always has uniform mapping. This means the use in the logic independently of the measurement system scaling is possible.

The resolution of the calculated speed depends on the scaling of the measurement system; i.e., the resulting speed value is always a multiple of the speed resolution. The lower the resolution of the encoder system, the lower the speed resolution; i.e., the coarser the divisions. The calculated speed resolution should always be significantly less than the speeds configured in the function blocks.

NOTE
The scaling can be calculated directly in the configuration window taking into account a gear factor and a mechanical factor.

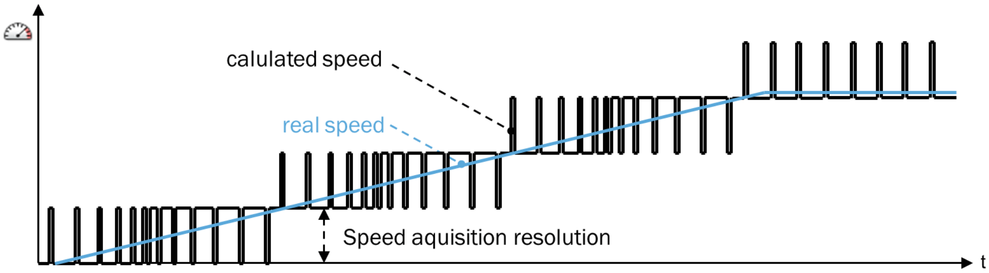


Figure 236: Resolution of the calculated speed as a function of the measurement system scaling

10.1.3 Encoder count direction

NOTE
The information in this section applies to all FX3-MOCx modules.

The count direction determines if the identified change in position is evaluated as positive (normal) or negative (inverted). This parameter can be used to adjust the count direction for encoders that count in the opposite direction due to their mounting position.

The definition of the signal sequence that applies for a normal count direction with A/B incremental encoders and with Sin/Cos encoders is provided in the technical data section for the Motion Control module in the operating instructions entitled “Flexi Soft Modular Safety Controller Hardware”.

10.1.4 Encoder connection type and ID code monitoring

NOTE
The information in this section applies to all FX3-MOCx modules.

Encoder voltage supply

The choice of voltage supply (**FX3-MOCx** or **External**) does not affect how the device works. Depending on what is selected, only the wiring example is adapted accordingly in the report.

Encoder connection type

The encoder connection type determines whether an encoder connection box is used for the encoder. Depending on what is selected here, ID code monitoring is either activated or deactivated for the encoder connection box. The wiring diagram is also adapted in the report.

ID code monitoring

Each encoder connection box contains an ID code, along with the outputs for the encoder voltage supply that comes from the FX3-MOCx module (ENC1_24V or ENC2_24V). If a connection type involving at least one encoder connection box (e.g., FX3-EBX1, FX3-EBX3 or FX3-EBX4) is selected in the configuration, the FX3-MOCx module checks this ID code on a cyclical basis.

To do this, the FX3-MOCx module switches the supply on and off at ENC1_24V and ENC2_24V alternately at 4 msintervals. The encoder remains oblivious to this because the supply voltages are combined via diodes. The ID code of the encoder connection box is then measured using the switched off supply. If the ID code measurement function detects an invalid value, the status bits are set to Invalid in the Motion data of the associated encoder. This happens when ENC1_24V / ENC2_24V or the shared 0 V voltage supply (ENC_0V) is interrupted between the FX3-MOCx module and the encoder connection box.

The status bits become valid again when the following conditions are met for at least the duration of the **Error recovery time** [6] without interruption:

- The ID code monitoring function detects valid values.
- Any other possible checks likewise provide a positive result.

With the help of the ID code monitoring function, breaks in the shared 0 V ENC_0V voltage supply or in the shared connecting cable that runs between the FX3-MOCx module and the encoder connection box can be detected.

10.1.5 Monitoring of the maximum step change in speed

NOTE
This function is only available for the FX3-MOC0 V1.

This parameter determines the maximum step change in speed that is allowed to occur in a configuration/application. If higher step changes in speed are detected by the FX3-MOC0 module (e.g., due to errors such as an electrical connection or a mechanical coupling being broken), the status bits are set to invalid in the Motion data for the corresponding encoder.

Description of the **Motion V1** data type: see "Data types used in the logic of the FX3-MOC0", page 253.

The status bits become valid again when the following conditions are met without interruption for at least the duration of 1 s (**Error recovery time**):

- The difference between the currently detected speed and the last valid speed detected returns to within the configured **Max. step change in speed.**
- Any other possible checks likewise deliver positive results.

The value is configured as the change in speed per millisecond [1]. The FX3-MOC0 module checks the step change in speed at 4 ms intervals [2], i.e. for four times the selected value. This corresponds to the logic execution time of the FX3-MOC0.

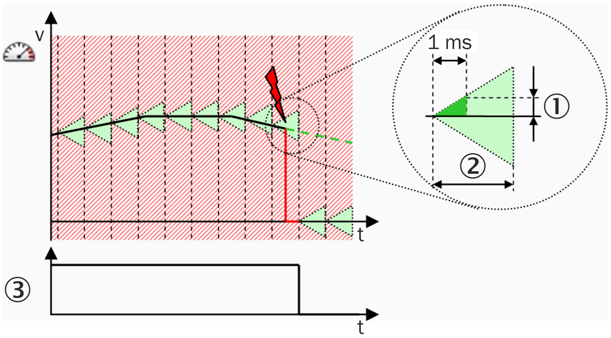


Figure 237: Monitoring of the maximum step change in speed for the FX3-MOC0

① Max. step change in speed
② FX3-MOCx logic execution time
③ Speed status

[6] The **Error recovery time** is 1 s in the FX3-MOC0, and 0.14 s or 1 s in the FX3-MOC1, depending on the configuration.

10.2 A/B incremental encoder

With this type of encoder, there are no specific parameters and monitoring functions. To achieve the desired level of safety function blocks can be included in the FX3-MOCx logic for the purpose of checking the information provided by the encoder (motion data) (see "Logic programming in Motion Control FX3-MOC0", page 250).

10.3 Sin/Cos encoders

10.3.1 Special parameters for Sin/Cos encoders

Table 143: Special parameters for Sin/Cos encoders

Parameter	Description
Sin/Cos analog voltage monitoring	see "Sin/Cos analog voltage monitoring", page 305
Resolution enhancement	see "Sin/Cos resolution enhancement", page 313

10.3.2 Sin/Cos analog voltage monitoring

This function is used to identify errors in the encoder system. This can be particularly helpful in the case of applications where an axis is to be monitored using just one Sin/Cos encoder. When Sin/Cos analog voltage monitoring is enabled, the system checks whether the relationship between the sine and cosine voltage is as required.

If the Sin/Cos analog voltage monitoring function detects invalid voltage conditions, the reliability bits are set to unreliable in the Motion data of the associated encoder.

The status bits become valid again when the following conditions are met without interruption for at least the duration of the Error recovery time [7]:

- The Sin/Cos analog voltage monitoring function detects valid voltage conditions.
- Any other possible checks likewise deliver positive results.

WARNING
Using unsuitable encoders

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

- ► Only use suitable encoders.
- ► Make sure that the encoder characteristics specified by the manufacturer will continue to apply to further deliveries or that you will be kept informed of any changes.
- ► Make sure that all errors to be considered are detected or can be prevented.

NOTE
IEC 61800-5-2 offers possible support for the errors to be considered.

The following must typically be obtained from the encoder manufacturer for this purpose:

- An implementation manual containing specific usage requirements for achieving a particular level of safety

or

- Information on the encoder design and the effects of errors on the Sin/Cos signals

During Sin/Cos analog voltage monitoring, the relationship between the sine and cosine voltages is checked based on two criteria:

- Vector length
- Signal deviation

[7] The **Error recovery time** is 1 s for the FX3-MOC0 and 0.14 s or 1 s for the FX3-MOC1 depending on configuration.

Monitoring of the vector length

If the ideal sine and cosine voltage values are transferred to an XY coordinate system, they describe a circle. Mathematically, the radius of the circle (vector length) is calculated from √ (sine^2 + cosine^2).

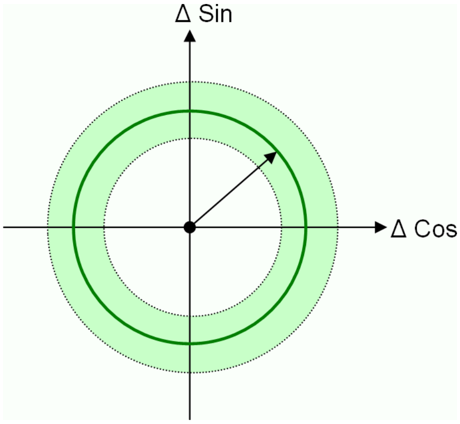


Figure 238: Monitoring of the vector length

This monitoring criterion is used to check whether the vector length is within the anticipated tolerance band. The specific limits that apply to this monitoring function are provided in the technical data section for the Motion Control module in the operating instructions entitled “Flexi Soft Modular Safety Controller Hardware”.

Monitoring of the signal deviation

This monitoring criterion is used to check whether the sine signal exhibits the anticipated signal deviation when the cosine signal has changed by at least the minimum vector length that is expected. In the same way, the signal deviation of the cosine signal is checked when the sine signal has changed.

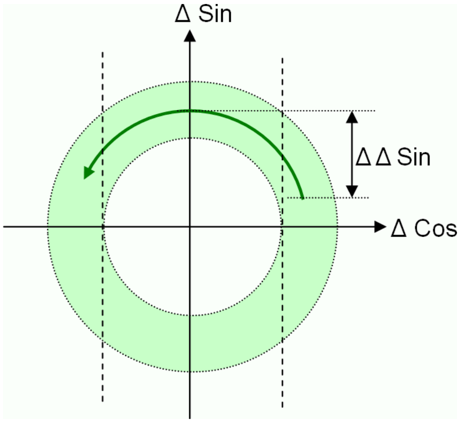


Figure 239: Monitoring of the signal deviation

This monitoring criterion even enables fault patterns to be detected in cases where either the sine signal or the cosine signal is affected by a stuck-at fault, but the resulting signal is still within the tolerance band (green circle) meaning that the fault cannot be detected by monitoring the vector length (see the second example in the list of possible fault patterns).

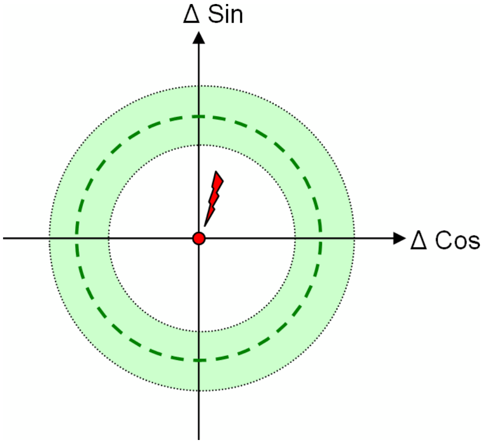
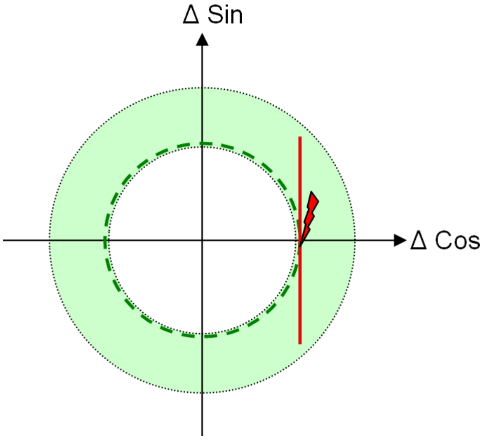
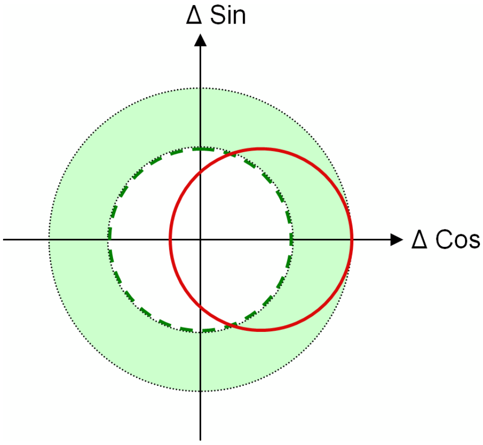
Example fault patterns

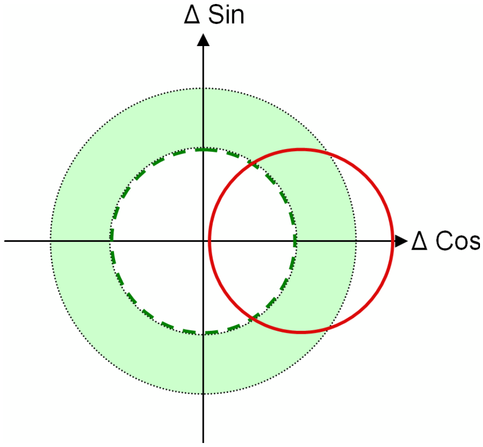
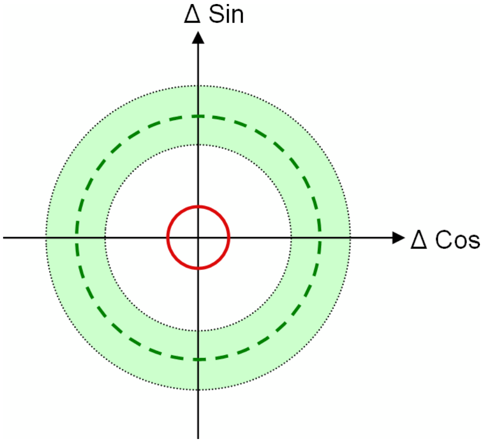
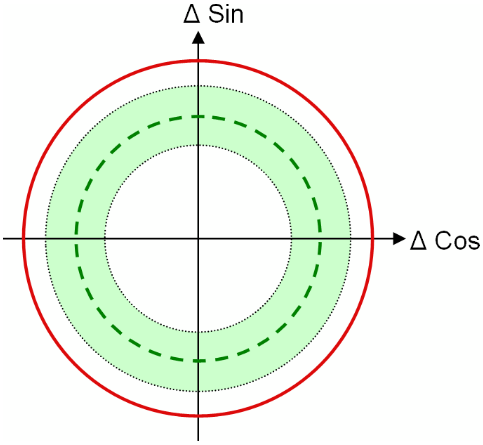
The following table shows some example fault patterns where the relationship between the sine and cosine voltage is not as required. Here ...

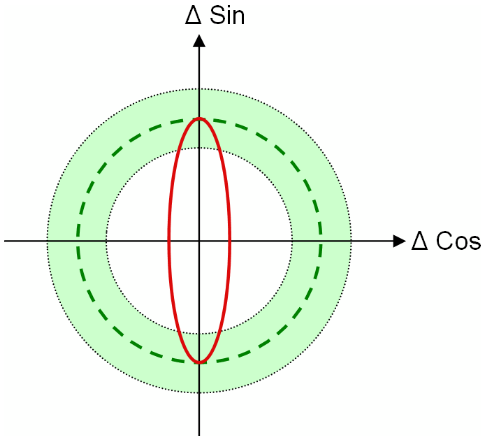
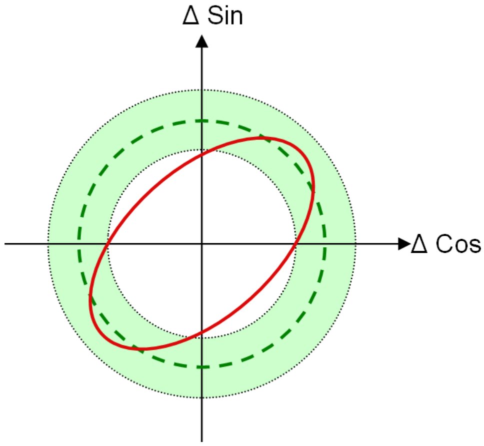
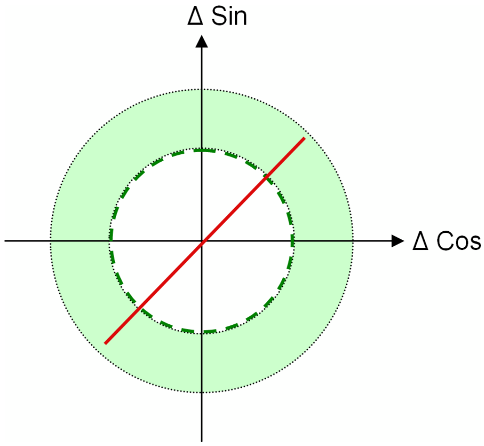
ΔSin = differential voltage between Sin+ and Sin– on Motion Control module

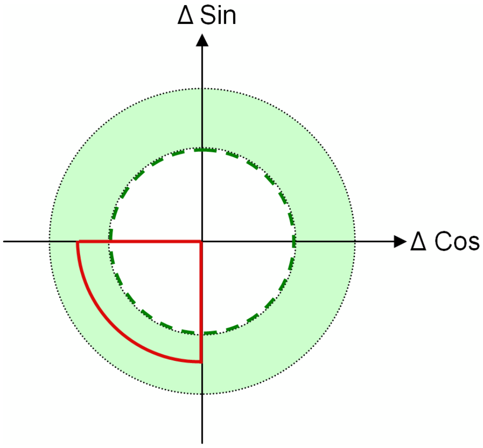
ΔCos = differential voltage between Cos+ and Cos– on Motion Control module

Table 144: Possible fault patterns during Sin/Cos analog voltage monitoring

Fault pattern	Possible causes of faults
	- Break in the encoder connection - No light emitted by transmitting diode - Internal encoder voltage supply is faulty
	- Stuck-at fault affecting sine or cosine signal
	- Interruption of or change in the Sin_Ref or Cos_Ref voltage

Fault pattern	Possible causes of faults
	- Interruption of or change in the Sin_Ref or Cos_Ref voltage
	- Internal encoder voltage supply is too low - Not enough light emitted by transmitting diode
	- Too much light emitted by transmitting diode

Fault pattern	Possible causes of faults
	- Sine or cosine gain factor is too low, e.g. due to change in resistance
	- Increased filter time due to increase in resistance - Cross-circuit between Sin+ and Cos+ - Cross-circuit between Sin– and Cos–
	- Cross-circuit between Sin and Cos in the case of encoders with Sin_Ref and Cos_Ref

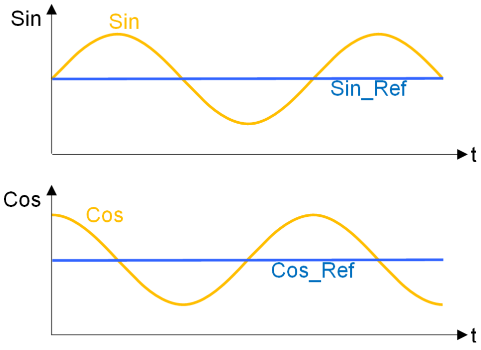
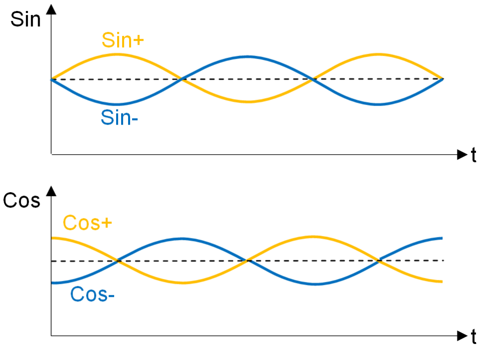
Fault pattern	Possible causes of faults
	- Change in the internal encoder reference voltage source for Sin_Ref and Cos_Ref with the result that the analog output stage of the encoder approaches the saturation limit and half-waves are partially or fully clipped.

10.3.3 Limits of Sin/Cos analog voltage monitoring

This section covers all applications with Sin/Cos encoders when the following conditions apply:

- A separate encoder is used to monitor each axis.
  and
- Sin/Cos encoders with Sin_Ref and Cos_Ref output signals are used.

Table 145: Examples of Sin/Cos encoder signals

Sin/Cos encoder signals	Examples for encoder
 Sin_Ref and Cos_Ref are DC voltages, typically 2.5 V DC	- SKS36S - SKM36S **Note:** If just one encoder of this kind is going to be used to monitor an axis, supplementary error control measures are required; e.g., shared use of the encoder signals for electronic commutation of the drive system.
	- DFS60S Pro **Note:** Encoders of this kind do not require any of the supplementary fault control measures described here.

Supplementary fault control measures

If the final two examples from the list of possible fault patterns shown in table 144 cannot be completely ruled out for the encoders that are being used, supplementary measures must be implemented to control these faults.

This is necessary because the values might only leave the tolerance band (green circle) briefly in the case of certain faults and the FX3-MOCx module might not be able to detect this in the event of high signal frequencies. If this happens, there is no guarantee that the FX3-MOCx module will be able to determine the speed or relative position correctly.

The following options are available for supplementary fault control:

- Fault detection by means of additional plausibility checks
- Shared encoder signals for electronic commutation of the drive system and fault detection based on safe status within the process

Error detection by means of additional plausibility checks

Another signal from the process can be evaluated in combination with the logic of the Motion Control module and main module in order to check the plausibility of the encoder motion signal. For example, a signal that evaluates the status of the drive can be used for this purpose (drive moving/drive not moving).

Shared use of encoder signals for electronic commutation

WARNING
Changes in the drive system

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

- ► Have the manufacturer confirm the relevant properties of the drive system.
- ► Check if changes in the drive system (e.g. due to product updates or reconfiguration) affect the common use of the encoder signals for electronic commutation.

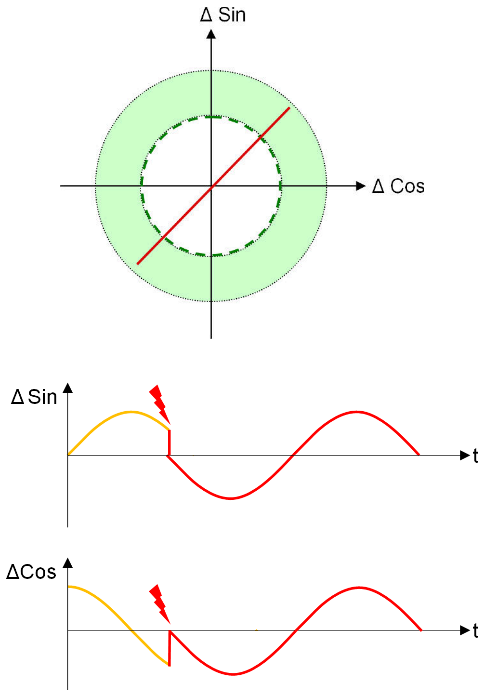
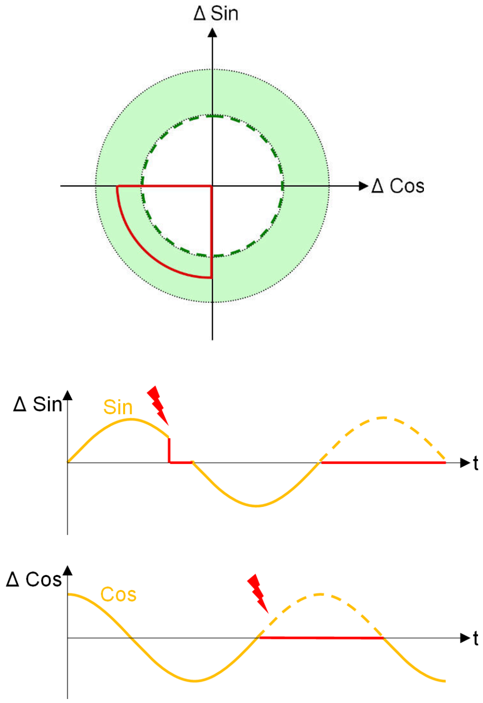
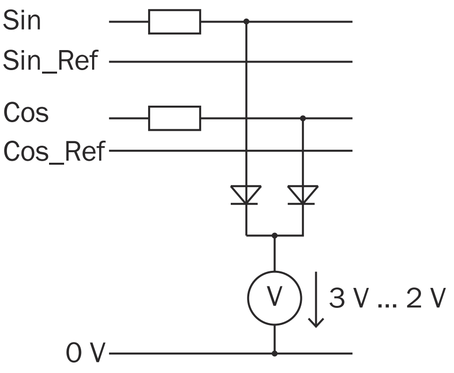
If the encoder is used for the Motion Control module and also for drive control, you can verify that the specified fault patterns reliably lead to a safe drive status (e.g., standstill or reduced torque). This is possible if one of the basic functional requirements of the drive system is that the encoder must determine the pole positions correctly in order for the rotary field to be generated, and if stationary commutation also results in a drive system standstill (synchronous drive).

In the case of encoders with Sin/Sin_Ref and Cos/Cos_Ref (Sin_Ref and Cos_Ref are DC voltages, typically 2.5 V DC), the encoder signals for electronic commutation of the drive system have to be shared. In this case, the polarity position is coupled directly and electronically with the current vector requirement for the three-phase rotary field. It is therefore assumed that if the commutation is stationary, the drive system will also be at a standstill.

NOTE
In encoders with Sin+/Sin– and Cos+/Cos– (Sin– and Cos– are inverted voltages of Sin+ and Cos+), there is **no** requirement to share encoder signals for the electronic commutation of the drive system.

The following table shows how the relevant error patterns can be simulated in order to check the effect on the drive system.

Table 146: Simulating error patterns for Sin/Cos encoder signals

Error pattern	Error simulation
	► Insert a series resistance of approximately 100 Ω in the sine signal line and in the cosine signal line running from the encoder to the drive system. The purpose of this is to prevent damage to the encoder. ► To activate simulation of the error, establish a connection (cross-circuit) between the sine and cosine signals.
	► Insert a series resistance of approximately 100 Ω in the sine signal line and in the cosine signal line running from the encoder to the drive system. The purpose of this is to prevent damage to the encoder. ► Connect diodes and a voltage regulator. Adjust the voltage regulator in line with the peak sine and cosine output voltage (typically 3 V).  ► To activate simulation of the error, reduce the voltage of the voltage regulator until the anticipated error pattern emerges (typically at around 2 V).

For this check, we recommend the following procedure:

- Install the circuit components for simulating the error but do not activate them.
- Check that the drive system is functioning correctly. The purpose of this is to verify whether simply installing the circuit components for error simulation without activating them is sufficient to bring about a safe status.

- ► Activate error simulation.
- ► Check the anticipated error pattern (by measuring with an oscilloscope).
- ► Check the anticipated effect on the drive system (safe status).

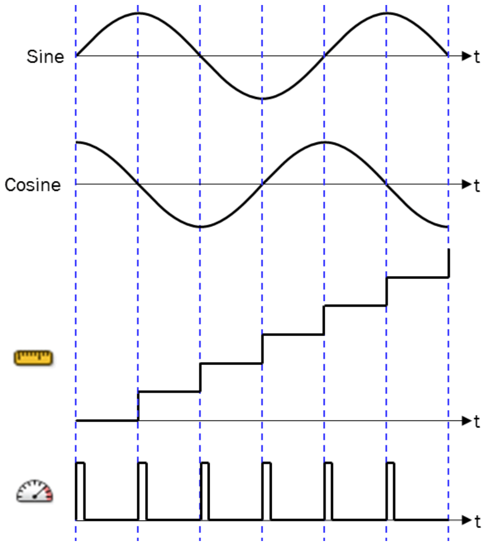
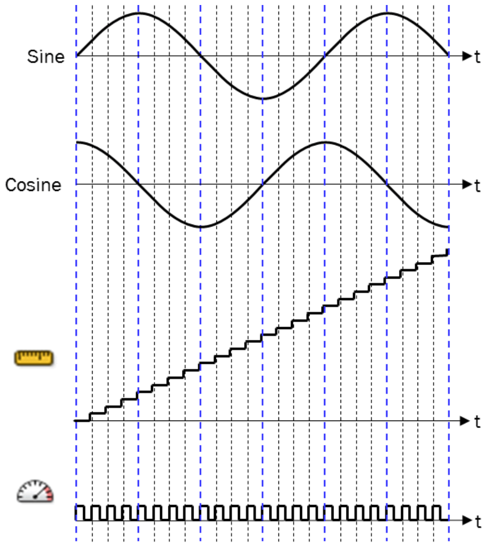
10.3.4 Sin/Cos resolution enhancement

NOTE
The information in this section applies to all FX3-MOCx modules.

This function is available for Sin/Cos encoders and is relevant for Sin/Cos encoder systems with a low resolution that can produce wider result stepping for speed detection. When resolution enhancement is activated, the number of counting points is increased by a factor of four, thereby improving the resolution of speed detection.

NOTE
This option has no effect on position formation (relative position value). The position shown in the diagram is an internal value used solely for calculating the speed.

Table 147: Sin/Cos resolution enhancement

Without resolution enhancement (deactivated)	With resolution enhancement (activated)
	

If – before Sin/Cos resolution enhancement is applied – the speed detection resolution is already less than or equal to the speed value mapped internally in the **Motion** data type (1 digit = 0.5 rpm or 1 mm/s), this option has no effect even when activated.

10.4 SSI encoder

The functions described here are available for SSI encoders (SSI master, SSI listener).

10.4.1 Special parameters for SSI encoders

Table 148: Special parameters for SSI encoders

Parameter	Description	Possible values
Data transmission rate	Data transmission rate for the clock output as SSI master	• 0 = listener • 100 ... 1,000 kBaud
Number of bits for the entire SSI protocol frame	Number of clock cycles for a single transmission	8 ... 62 [1)]

Parameter	Description	Possible values
Number of leading bits	Number of leading bits that do not contain position data	0 … 54 [2]
Number of position data bits	Number of bits containing the relevant position data bits	8 … 32 [3]
Transmit position data twice	For selecting whether the position value should be transmitted once or twice using an SSI protocol frame	• Transmission of a single position value • Duplicate transmission of the position value
Number of bits between the position data bits	Only available with duplicate transmission of the position value	0 … 30
Data encoding	Data encoding for the position data bits	• Binary • Gray
Error bit evaluation	Monitoring of error bits supplied by the encoder in the SSI protocol frame. For each individual bit, you can specify whether the error status should be represented by a 1 or a 0.	**For each bit that is not a position data bit** • 1 is error • 0 is error
Max. data reception interval	Maximum time within which valid position data is expected to arrive	4 … 100 ms

[1] With firmware version ≥ V1.10.0. With earlier firmware versions 16 … 62 bits.
[2] With firmware version ≥ V1.10.0. With earlier firmware versions 0 … 46 bits.
[3] With firmware version ≥ V1.10.0. With earlier firmware versions 16 … 32 bits.

10.4.2 Double data transmission

Overview

The information in this section applies to all FX3-MOCx modules.

Double data transmission

Certain SSI encoders support multiple transmission of the position data. This means that the same encoder data is output again, provided that the clock gap between the data packages (monoflop time) is not exceeded. This makes it possible to detect data that has been corrupted by transmission faults, for example.

The FX3-MOCx module supports duplicate transmission of the position data. When “Transmit position data twice” is activated, the FX3-MOCx module checks whether the two position data values within the received SSI protocol frames are identical. If they are not identical, the position data within this SSI protocol frame is ignored. All other SSI telegrams arriving in the same 4 ms logic cycle of the FX3-MOCx are also ignored.

You can find Information on how motion data from the associated encoder is affected in the “Maximum data reception interval” section.

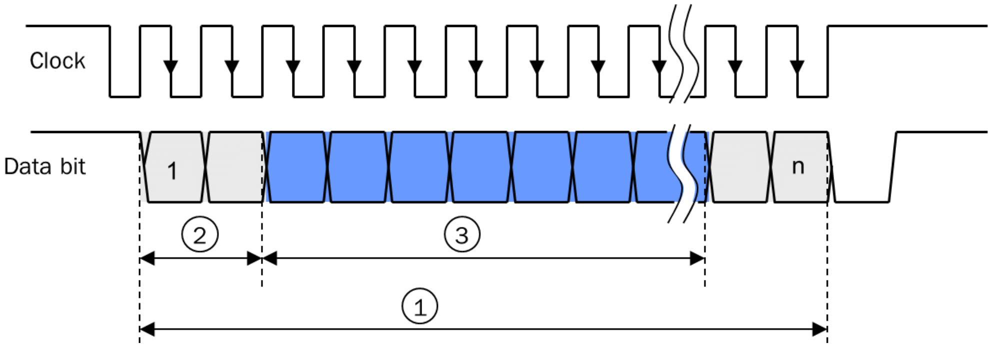


Figure 240: Transmission of a single position value

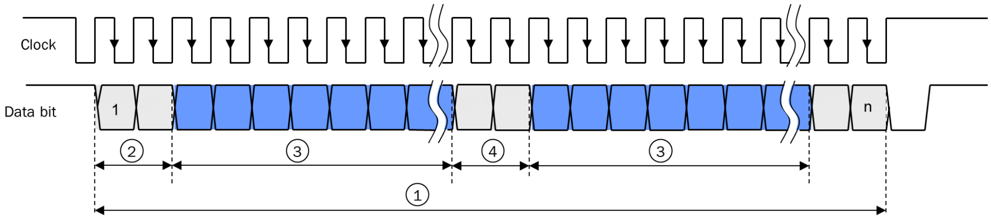


Figure 241: Duplicate transmission of the position value

① Number of bits for the entire SSI protocol frame
② Number of leading bits
③ Number of position data bits
④ Number of bits between the position data bits

10.4.3 Error bit evaluation

NOTE
The information in this section applies to all FX3-MOCx modules.

As well as including the position data bits in the SSI protocol frame, certain SSI encoders also transmit error bits that represent the results of internal monitoring functions performed by the encoder. Such error bits can be evaluated with the FX3-MOCx. For each individual bit, you can specify whether the error status should be represented by a 1 or a 0. If an error status is detected for one or more of the selected error bits, the position data within this SSI protocol frame is ignored.

10.4.4 Maximum data reception interval

This function enables the system to tolerate invalid position data temporarily by relying on the most recently valid position data in the meantime. The results of all relevant monitoring functions must remain valid for longer than the **Max. data reception interval** at least once. If not, the status bits are set to invalid in the motion data of the associated encoder.

Description of the **Motion V1** data type: see "Data types used in the logic of the FX3-MOCO", page 253.

In the case of the SSI encoder, the following monitoring functions affect the **Max. data reception interval**:

- SSI protocol frame not received or not received in full (only applies to SSI listener)
- Transmit position data twice
- Error bit evaluation
- Max. step change in speed

The status bits become valid again when all monitoring functions are met without interruption for at least the duration of 1 s (**Error recovery time**):

NOTE
In SSI listener mode, only ever one SSI protocol frame is evaluated within the 4 ms cycle. Further SSI protocol frames transmitted within the same cycle are not evaluated.

11 Logic programming in Motion Control FX3-MOC1

11.1 General description

The Motion Control module FX3-MOC1 module is a module for drive monitoring. It can be used for the safe monitoring of different drives (electric, pneumatic, hydraulic, etc.), provided that a suitable sensor system is installed.

The FX3-MOC1 has its own logic editor. If a project includes one or more FX3-MOC1, you can open the associated logic editors by double-clicking on the relevant module or by using the **Logic editor** menu.

The functions blocks described in this chapter are only available in the logic editor of an FX3-MOC1. They have been specifically tailored for drive monitoring applications. Firstly, there are the actual monitoring function blocks that can be used to monitor the speed, position, or stopping and braking functions. Secondly, there are the data conversion function blocks. These are required because the Motion Control modules differ from the rest of the Flexi Soft system in that they are also capable of processing integer data types.

Function blocks for logic operations (AND, NOT, OR) are also available.

NOTE
The encoder elements are configured outside of the logic editor. Refer to "Encoder in Motion Control FX3-MOC1", page 419.

Differences between FX3-MOC0 and FX3-MOC1

In the FX3-MOC0, two motion signals can be plausibility-checked using a speed comparison. The relative positions of both signal sources may continuously diverge (e.g., in a vehicle with a sensor on both a left wheel and a right wheel).

In the FX3-MOC1, two motion signals can be plausibility-checked either by means of a position comparison or, as from FX3-MOC1 V3, also be means of a speed cross check. The position comparison is less sensitive to short-term differences in speed. Both detected positions must change at an even rate (allowing for slight drift).

The Motion Control FX3-MOC1 also features additional functions for monitoring absolute positions. It also includes the logic functions AND, NOT and OR to facilitate signal processing within the FX3-MOC1 logic.

16 additional customized monitor bits in the FX3-MOC1 allow diagnostics data to be transferred to the main module and to the gateways (data set 3) using the module status bits. As a result, the 16 process data bits that can be sent from the FX3-MOC1 to the main module do not have to be used for diagnostics functions.

The FX3-MOC1 uses the expanded data type Motion V2, which allows for a more robust and intelligent evaluation (see "Data types used in the logic of the FX3-MOC1", page 319).

11.2 Safety notes for logic programming

Standards and safety regulations

All safety-related parts of the system (wiring, connected sensors and control devices, configuration) must conform to the relevant standards (e.g., IEC 62061 or ISO 13849) and safety regulations.

WARNING
Incorrect configuration of the safety application

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

► Observe all applicable standards and safety regulations.
► Check that the operating principle of the Flexi Soft hardware and the logic program react in accordance with the risk avoidance strategy.
► Only use safety-related signals for safety-related applications.
► Always use the correct signal sources for the function blocks.

Safe value

The safe value of process data and outputs is 0 or Low and this is set when an error is identified.

WARNING
Inadequate safety measures

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

If the safe value (signal = Low) could lead to a dangerous state in the application, additional measures must be implemented. This applies in particular to inputs with signal edge detection.

► Analyze the status of the process data.
► Switch off the affected outputs if the status analysis detects an error.

Unexpected rising or falling signal edges

An error at an input can result in unexpected rising or falling signal edges (e.g. an interruption in network communication, a cable break at a digital input, a short-circuit at a digital input that is connected to a test output). The safe value remains set until the conditions for resetting the error have been met. For this reason, the affected signal may behave as follows:

- It temporarily switches to 1 instead of remaining set to 0 as it normally would in the fault-free status (rising signal edge and falling signal edge, i.e. 0–1–0).
- It temporarily switches to 0 instead of remaining set to 1 as it normally would in the fault-free status (falling signal edge and rising signal edge, i.e. 1–0–1).
- It remains set to 0 instead of switching to 1 as it normally would in the fault-free status.

WARNING
Unexpected rising or falling signal edges

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

► Take into account unexpected rising or falling signal edges.

Delays due to jump addresses

Jump addresses can extend the logic execution time and thus the response time if a logical loopback occurs through them.

A logical loopback occurs when a function block input is connected to a destination jump address but the associated source jump address is linked to an output of the same function block or to an output of a function block with a higher function block index number. [8] In this case, the input comprises not the output values of the current logic cycle, but rather the output value of the previous logic cycle. This must be taken into account in terms of the functionality and, in particular, when calculating the response time.

If there is a logical loopback due to a jump address, then this effects a delay of one logic cycle. In this case, the input of the jump address is displayed with a clock symbol (with Flexi Soft Designer ≥ V1.3.0).

WARNING
Extension of the response time

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.

► The delays caused by the logical loopbacks must be factored in when calculating the response time and functionality.

11.3 Parameterization of function blocks

The function blocks have configurable parameters. Double-clicking on a function block opens the configuration window of the function block. The configurable parameters are distributed on different tabs here.

In the case of function blocks for which speeds or positions have to be configured, you can set the units that are to be used on the **Units** tab; e.g., for the purpose of calculating speeds (mm/s, km/h, rpm, etc.).

The **I/O Comment** tab allows you to replace the default designations of the function block inputs and outputs with your own. It also enables you to add a name or descriptive text for the function block, which is displayed under the function block in the logic editor.

The remaining configurable parameters for the function block can be found on the other tabs in accordance with the type of function block.

The **Report** tab shows a summary of the function block configuration. This includes all the links between the inputs and outputs, and the configured parameters.

11.4 Inputs and outputs in the logic editor

Inputs

The following input data is available in the logic editor of the FX3-MOC1:

- Data routed from the main module of the Flexi Soft system, see "Exchange of process data between main module and FX3-MOC1", page 321
- Motion data of the connected encoders, see "Data types used in the logic of the FX3-MOC1", page 319 and "Encoder in Motion Control FX3-MOC1", page 419
- General data sources of the FX3-MOC1, see "General data sources", page 253

NOTE
The inputs and outputs are color-coded according to their function:

- Gray: Non-safe
- Yellow: Safe
- Blue: Diagnostics

[8] The function block index number is displayed at the top of each function block and indicates the position occupied by the function block within the execution sequence.

Outputs

The following output data is available in the logic editor of the FX3-MOC1:

- Data routed to the main module of the Flexi Soft system, see "Exchange of process data between main module and FX3-MOC1", page 321
- 4 customized MOC status bits and 16 customized MOC monitor bits, see "Module status bits of the FX3-MOC1", page 322

11.4.1 General data sources

Static 0 and Static 1

The inputs **Static 0** and **Static 1** are available in the **Inputs** selection window.

The **Static 0** input can be used to set a function block input permanently to 0. Similarly, the **Static 1** input can be used to set a function block input permanently to 1. This might be necessary, for example, to achieve a valid logic configuration if the relevant function block contains inputs that are not required but cannot be deactivated.

First logic cycle

This input has the value 1 in the first cycle in which the logic of the Motion Control is executed, otherwise it has the value 0.

NOTE

The value of the **First logic cycle** input refers to the logic of the Motion Control. This starts before the logic of the main module. As a result, this input is set to 1 before the corresponding input in the logic editor of the main module.

11.5 Time values and logic execution time

The logic execution time of the Motion Control is 4 ms.

the accuracy of which is ±100 ppm (parts per million).

Table 149: Precision of times (parameters and invariable values) in accordance with increment and absolute value

Configuration increment	Value range for the function block	Precision
4 ms	≤ 5,000 ms	±0.5 ms
	> 5,000 ms	±100 ppm of the configured time
1 s	≤ 40 s	±4 ms
	> 40 s	±100 ppm of the configured time

11.6 Data types used in the logic of the FX3-MOC1

The function blocks in the FX3-MOC1-MOCx are capable of processing various data types. This makes them different from the function blocks in the main module, which can only process Boolean values. The type of data that is expected or output depends on which function block input or output is used in each case.

Boolean

Boolean data is binary. It can only be 1 or 0.

Motion V2

NOTE
Type **Motion V2** data is supported from FX3-MOCx V2. Type **Motion V2** data is an extended version of the **Motion V1** data type used by the FX3-MOC0 V1.

Type **Motion V2** data encompasses all the information provided by an encoder. It consists of the following elements:

Table 150: Composition of data of type Motion V2

Element	Size	Internal value range (number of digits)	Internal resolution for rotary movement type	Internal resolution for linear movement type
Speed	16 bits with sign	–32,767 ... +32,767	1 digit = 0.5 rpm	1 digit = 1 mm/s
Speed status	1 bit	0 = invalid 1 = valid	–	–
Speed reliability	1 bit	0 = unreliable 1 = reliable	–	–
Relative position [1]	32 bits with sign	–2,147,483,648 ... +2,147,483,647	1 digit = 1/30,000 rev .	1 digit = 1/250 mm
Relative position status	1 bit	0 = invalid 1 = valid	–	–
Relative position reliability	1 bit	0 = unreliable 1 = reliable	–	–
Absolute position [2]	32 bits with sign	–2,147,483,648 ... +2,147,483,647	1 digit = 1/30,000 rev .	1 digit = 1/250 mm
Absolute position status	1 bit	0 = invalid 1 = valid	–	–
Absolute position reliability	1 bit	0 = unreliable 1 = reliable	–	–
Update status	1 bit	0 = not current 1 = current	–	–

[1] A relative position means that the traveled path can be reproduced, but the position in relation to the mechanical position is not clear. This primarily occurs because the start value for the relative position in the encoder’s Motion V2 data always starts with a 0, regardless of the mechanical position.

[2] An absolute position means that the position value is clear for any possible mechanical position in the application. This also applies following a restart of the measurement system.

NOTE

The following rules apply in relation to the status and reliability:

- When a status bit is 0 = invalid, then the corresponding value is 0.
- A value can be reliable only if it is also valid.
- The **Relative position** can only be valid when the **Speed** is also valid.
- The **Absolute position** can only be valid when the **Relative position** is also valid.
- If the encoder system used is unable to deliver an absolute position value due to its design, then the **Absolute position** is permanently invalid.
- When the status of a value is set to invalid, it can become valid again only after 1 s has elapsed (**Error recovery time**) as soon as valid data is available again.
- After switching on the FX3-MOC1, the **Speed** and therefore the **Relative position** and the **Absolute position** are invalid for at least 0.5 s. As a result, the system only starts to evaluate the Motion V2 values in the function blocks at the earliest once this period has elapsed. Until this point is reached, the logic is executed at least once in the main module and the results of this process are transferred to the FX3-MOC1 so that current, valid values are available at the start of the evaluation in the FX3-MOC1. These values affect factors such as the selection of the permitted speed limit and the permitted direction of movement.
- The **Update status** is set to 1 = current when the Motion V2 values are updated based on the encoder signals recorded in the current processing cycle. When the **Update status** is 0, the previous values remain unchanged and remain value if the relevant **Validity status** is 1.

The **Motion V2** data type is used by the **Motion In** inputs and the **Motion Out** output of various FX3-MOC1 function blocks. The individual elements of the data type are evaluated automatically in the respective FX3-MOC1 function block.

Internal resolution of the speed and position information

The smallest unit for the detected speed and position information is determined by the internal resolution of this data (see table 150). It may be further restricted by the resolution of the encoder system.

UI8

Data of the UI8 type can, for example, be used to select or display a speed or position range.

Table 151: Possible values for UI8 data

Element	Size	Values for speed ID
UI8	8 bit	0 = invalid 1 ... 31 = range index

NOTE

Inputs and outputs that expect or that output data types other than Boolean are marked accordingly on the function block icons. Within this context, **M** stands for **Motion V2** and **UI8** stands for **Unsigned Integer 8-Bit**.

11.7 Exchange of process data between main module and FX3-MOC1

Given that the main modules and the FX3-MOCx modules are capable of processing different types of data and that more complex signal preprocessing and logic can be programmed in the FX3-MOCx, data exchange between the modules must be properly organized. 18 bits can be sent from the main module to the FX3-MOCx and 16 bits can be sent from the FX3-MOCx to the main module. These bits must be linked in the logic editor.

The bits from the FX3-MOCx to the main module appear ...

- In the Logic editor of the FX3-MOCx under **Outputs** and
- In the Logic editor of the main module under **Inputs** for the relevant FX3-MOCx module.

The bits from the main module to the FX3-MOCx appear ...

- In the Logic editor of the FX3-MOCx under **Inputs** and
- In the logic editor of the main module under **Outputs** for the relevant FX3-MOCx.

The tag name for these bits is pre-populated with the name of the input + block + module (default name). This name can be changed if required.

The data is exchanged via the internal FLEXBUS+ bus.

Requirements for signals

WARNING
Unrecognized signals from the FX3-MOCx module

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

► Observe the time requirements for signals that are sent from the FX3-MOCx to the main module.

The signals that are sent from an FX3-MOCx to the main module must meet the same time requirements as all other signals. If the logic execution time of the main module is more than 4 ms, a signal sent from the FX3-MOCx to the main module must continue to have the same status for at least as long as the logic execution time of the main module. This is necessary to ensure that this status can always be detected in the main module logic.

Switch-on delay in the main module logic

Following a transition to the Run status, there is a delay of up to 80 ms before logic processing by the main module commences. This ensures that the logic of the main module always has the latest valid signals of the expansion modules to work with. As a result of this, all the data sent from the main module to the FX3-MOCx remains set to 0 once the system has transitioned to the Run status for up to 80 ms plus the logic execution time of the main module. This applies in particular to signals that are transmitted by inputs to expansion modules and whose status is sent to the FX3-MOCx via the main module.

After switching on the FX3-MOC1, all motion values of the encoder are invalid for at least 0.5 s. All the motion values of the encoder are then set to valid as soon as valid data is available. As a result, the system only starts to evaluate the motion values in the function blocks at the earliest once this period has elapsed. Until this point is reached, the logic is executed at least once in the main module and the results of this process are transferred to the FX3-MOC1 so that current, valid values are available at the start of the evaluation in the FX3-MOC1. These values affect factors such as the selection of the permitted speed limit and the permitted direction of movement.

11.8 Module status bits of the FX3-MOC1

Overview

The module status bits for the expansion modules contain diagnostics data. This data is refreshed approximately every 200 ms. Due to the longer refresh interval, this data may not be consistent with the latest process data for the module.

Important information

WARNING
Non-secure or inconsistent data

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

- ► Only use secure data for safety-related applications.
- ► Only use the module status bits of the expansion modules for diagnostic purposes.

Module status bits of the FX3-MOC1

The module status bits of the FX3-MOC1 are available at the following locations:

- In the logic editor of the main module under **Diagnostics**, for use as inputs in the logic program of the main module.
- In the Flexi Soft gateways in data set 3.
- In the RS-232 routing of the main module.

Table 152: Module status bits of the FX3-MOC1

Module status bit	Description
Configuration is valid	0 = Configuration invalid 1 = Configuration valid
Encoder 1 is OK	0 = Error 1 = No error or not used
Encoder 2 is OK	0 = Error 1 = No error or not used
Teach position for encoder 1 is OK	0 = Error 1 = No error or not used
Teach position for encoder 2 is OK	0 = Error 1 = No error or not used
User-defined MOC status bit 1 … 4	• User-defined module status bits • Alarm generation
User-defined MOC monitor bit 1 … 16	• User-defined module monitor bits • No alarm generation

The user-defined MOC monitor bits can be used, for example, to visualize the speed values. The MOC monitor bits are transferred to a controller via a gateway without needing to use process data bits from the FX3-MOC1 to the main module.

Further topics

- "Input data status and output data status", page 93

11.9 Overview of the function blocks in the FX3-MOC1

The Logic editor of the FX3-MOC1 uses function blocks to define the safety-related logic. A configuration can include up to 25 function blocks.

Depending on the type of function blocks used, even a small number can result in the available computing time in the logic execution cycle or the available memory capacity being exceeded. These values are displayed in the Logic editor of the FX3-MOC1, on the **FB group info** tab.

There are some function blocks for monitoring and others for logic functions and data conversion. The following table lists all the function blocks that are available for the FX3-MOC1:

Table 153: Overview of the function blocks in the FX3-MOC1

Function block name	Description
Function blocks for monitoring	
Position by reference V1 and Position by reference (with restore) V1	Used to generate an absolute position based on a relative position (e.g. from an A/B incremental encoder) and a reference signal (e.g. from a reference cam). The **Position by reference (with restore) V1** function block enables you to save the absolute position value and to restore it when resuming operation.
Position cross check V1	Compares position values from two different signal sources. This makes it possible to achieve a higher level of safety. For this purpose, the relative position values from both signal sources are cross checked so that it is possible to evaluate and monitor the standstill, direction of movement, and speed. If absolute position values are available, these are also cross checked so that it is possible to evaluate and monitor the absolute position.
Speed monitoring V2	Enables speed and direction monitoring. The basic functions are: • Safe speed monitor (SSM) • Safely limited speed (SLS) • Safe direction (SDI) • Safe operating stop (SOS) • Monitoring of up to four different speed ramps during the transition from a monitored speed to a lower speed
Position monitoring V1	The **Position monitoring V1** function block is the central block for all position, speed and direction monitoring functions within an application. It contains the functions of the **Speed monitoring V2** function block plus additional functions for position evaluation and position monitoring. • Safely-limited position (SLP) • Safe cam (SCA) • Safe speed monitor (SSM) • Safely limited speed (SLS) • Safe direction (SDI) • Safe operating stop (SOS) • Monitoring of up to four different speed ramps during the transition from a monitored speed to a lower speed
Safe Stop V2	This is used to trigger and monitor a Safe Stop for a drive system. The drive has to be shut down in a controlled manner. The stop ramp of a drive system is not usually safe. For this reason, the Safe stop V2 function block monitors the actual reduction in speed until the drive comes to a standstill. Functions: • Safe Stop 1 (SS1) • Safe Stop 2 (SS2) Typical application: • Monitoring the shutdown and stopping behavior of a machine
Function blocks for data conversion	
UI8 to Bool V1	Converts an 8-bit integer value into a Boolean value. Possible application: For connecting the **Speed status ID** output of the **Position monitoring V1** function block to Boolean signals so that they can be forwarded to the main module.

Function block name	Description
Bool to UI8 V1	Converts a Boolean value into an 8-bit integer value. Possible application: For connecting the **Speed enable ID** input of the **Position monitoring V1** function block to Boolean signals from the main module.
Motion status to Bool V2	Converts the **Speed status**, the **Relative position status**, and the **Absolute position status** from the data type Motion V2 to Boolean.
Speed to Bool V2	Converts the **Speed** and the **Speed status** from the data type Motion V2 to Boolean.
Speed to laser scanner V2	Converts the **Speed** from the data type Motion V2 into a format with cm/s scaling that is suitable for a SICK laser scanner.
Function blocks for logic functions	
NOT V1	Negation
AND8 V1	ANDing of up to 8 inputs, invertible
OR8 V1	ORing of up to 8 inputs, invertible

NOTE
The suffix **V1** or **V2** is used to differentiate between the different versions of the function blocks. The version of the function block that is used depends on the module type used (FX3-MOC0 or FX3-MOC1) and the module version.

11.10 Function blocks for monitoring

11.10.1 General principles

Reliability

Reliable Motion V2 signals are essential to safe monitoring. This is particularly true for the position monitoring, speed monitoring and Safe stop functions. The required level of reliability is achieved through the performance of plausibility checks on encoder signals.

It is possible to do this in the following ways (individually or in combination):

- One Sin/Cos encoder with Sin/Cos analog voltage monitoring in the FX3-MOC1
- Two non-safe encoders with downstream speed cross check or position comparison in the FX3-MOC1

Effect chain using the Motion V2 signal

The various function blocks are linked together via the Motion V2 signal. This contains the values for speed, relative position and absolute position, as well as the relevant validity status and the reliability status of the respective values.

The Speed monitoring and Position monitoring function blocks do not have a Motion output. For this reason, the error reaction must be explicitly polled for these function blocks via the relevant Monitoring status output and connected with a Safe stop function block, for example. The **Safe stop 2** is normally used for this. In the event of an impermissible speed or direction of travel, this allows the drive to be stopped with subsequent standstill monitoring without switching off the voltage supply.

In terms of the validity and reliability of the status bits, the following applies:

- 0 = invalid/unreliable/unknown
- 1 = valid/reliable
- Initial status when the Flexi Soft system transitions to the Run status = 0

The following explanations apply to the function block monitoring outputs:

- 0 = error detected
- 1 = OK (no error detected or status because relevant data is unreliable or invalid at one of the inputs for **Motion In**)
- Initial status when the Flexi Soft system transitions to the Run status = 1

This means that, even if an error is detected at any point in the chain, the subsequent function blocks will not indicate an error at their status outputs. This enables the user to clearly identify the location of the error.

However, it also means that it is not possible to query the output status of the last block in a chain to verify that no errors are present, i.e. that the data at the **Motion In** inputs is valid and reliable.

Instead, it is necessary to query the reliability status of the Motion V2 signal at the end of the chain. You can do this implicitly by using the Motion V2 signal at a safe stop function block, for example, or explicitly using one of the following conversion function blocks:

- Motion status to Bool V2
- Speed to laser scanner V2
- Speed to Bool V2

Alternatively, you can implicitly query the status data indicating invalid Motion V2 values, such as:

- Speed status ID output (0 = invalid) of the speed monitor or position monitor function blocks
- Position status ID output (0 = invalid) of the position monitor function block

Inhibiting error messages and error responses

In some operating situations in which errors are expected, error messages and, if necessary, the responses to those errors can be repressed.

Repress error message

Using the **Repress error signal** input on the **Position comparison V1** function block and with the optional input of the same name on the **SSI encoder** and on the **Sin/Cos encoder** in the FX3-MOC1 logic, messages relating to the plausibility functions for the function block and the encoder can be repressed. The following messages can be repressed:

- Entry in the diagnostics history (encoder and function block)
- Error messages in the module status bits (data set 3 of the gateways) (encoders only)
- Display of the LED MS on the FX3-MOC1 (encoders only)

NOTE

- The **Inhibit error indication** input for the encoders appears in the logic editor of the FX3-MOC1 under **Outputs** for the corresponding FX3-MOC1.
- If the **Inhibit error indication** input of the encoder is connected to an output of a function block in the FX3-MOC1 module (rather than to a bit that originates from the main module), the input will be delayed in its response by one logic cycle, because it must evaluate the result of the function block from the previous cycle.

The error response, particularly the change of the status bits in the **Motion out** output, is not affected by the **Repress error signal** input.

Inhibiting the error response

If the system is rendered safe by other means (e.g. protective door closed), it may be expedient to inhibit the error response at the end of the signal chain, e.g. in production phases with a very high potential for faults to occur. This can be achieved using the **Inhibit motion bits reaction** input of the **Safe stop V2** function block.

11.10.2 Programming examples

Evaluation of speed and standstill

Programming example 1: Evaluation of speed using two A/B incremental encoders

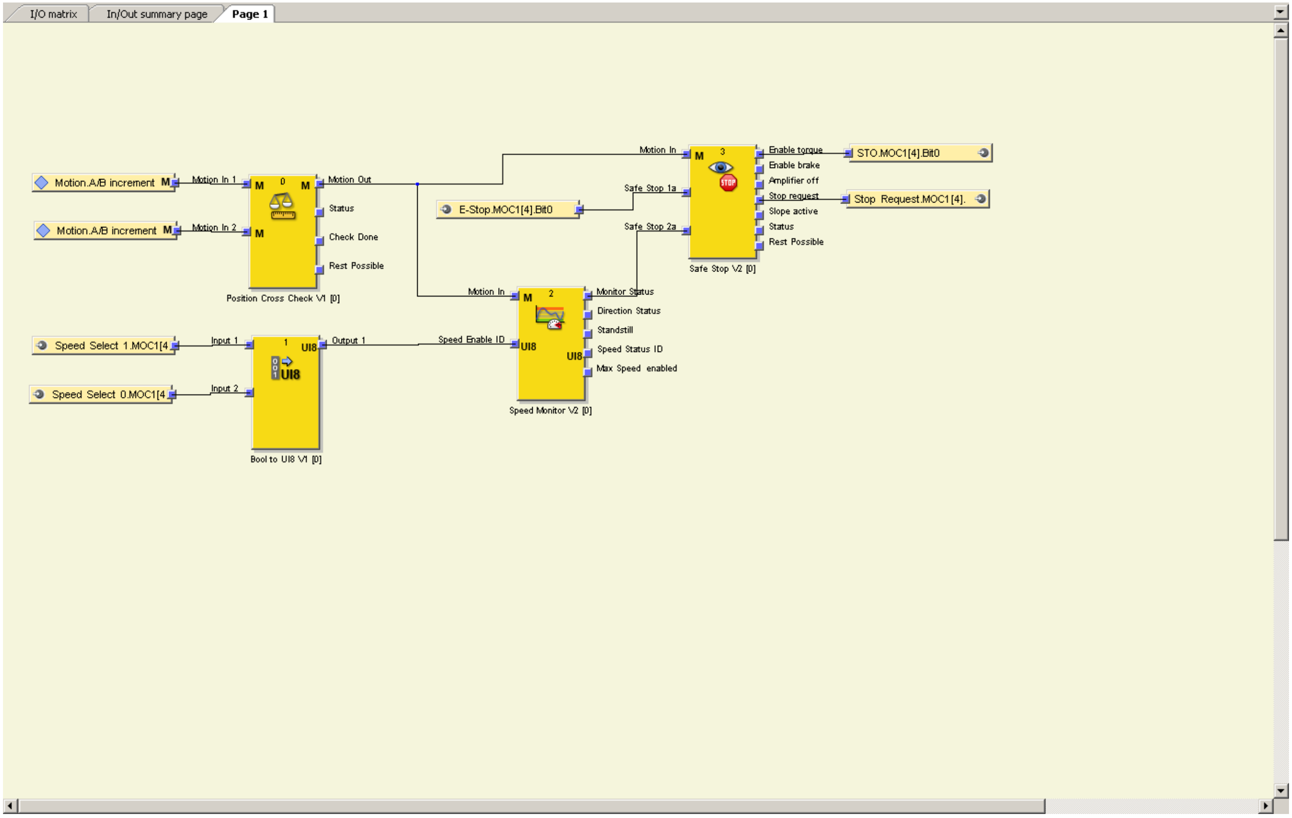


Figure 242: Evaluation of speed using two A/B incremental encoders

Programming example 2: Evaluation of speed using a safe Sin/Cos encoder (e.g., DFS60S Pro)

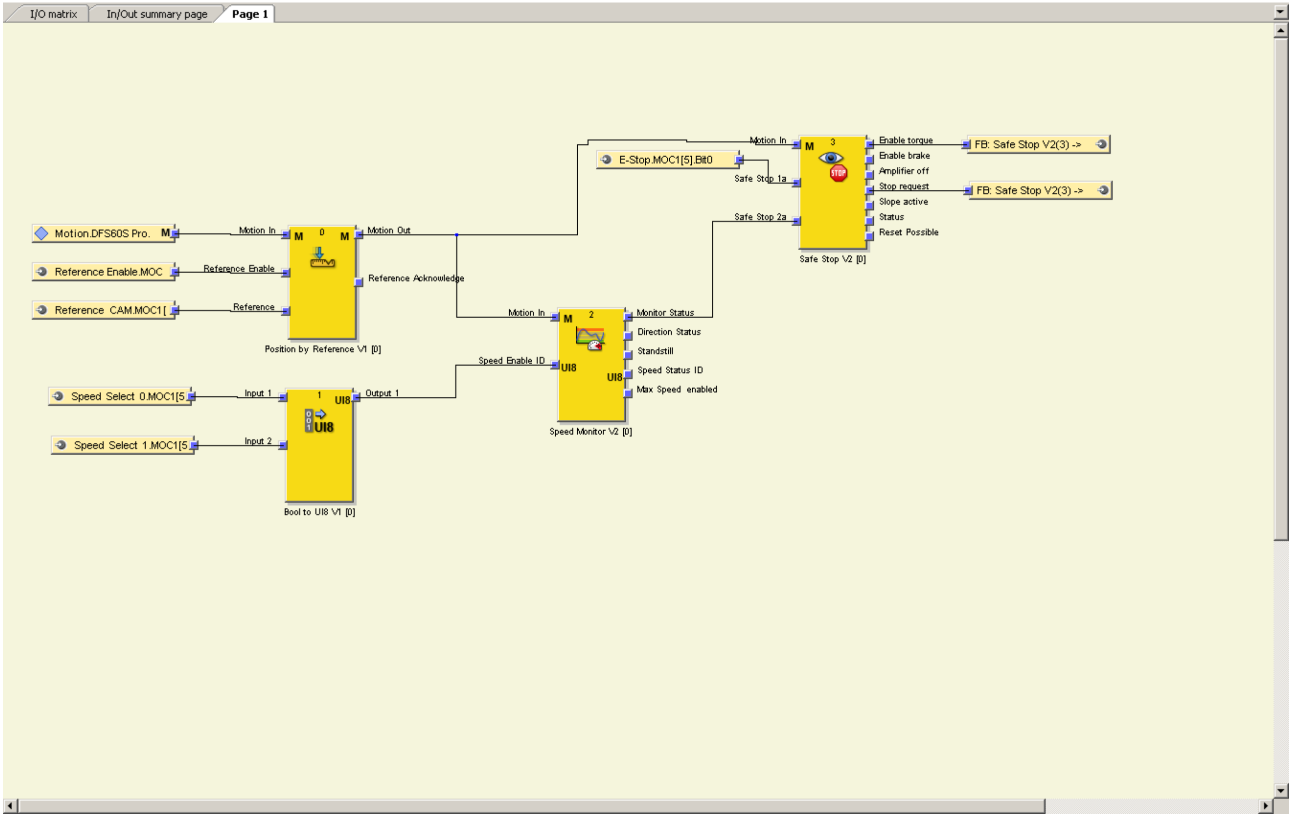


Figure 243: Evaluation of speed using a safe Sin/Cos encoder (e.g., DFS60S Pro)

Evaluation of speed, standstill and position

Programming example 3: Evaluation of position using two A/B incremental encoders, each with one reference signal (cam) and reference run on each system start

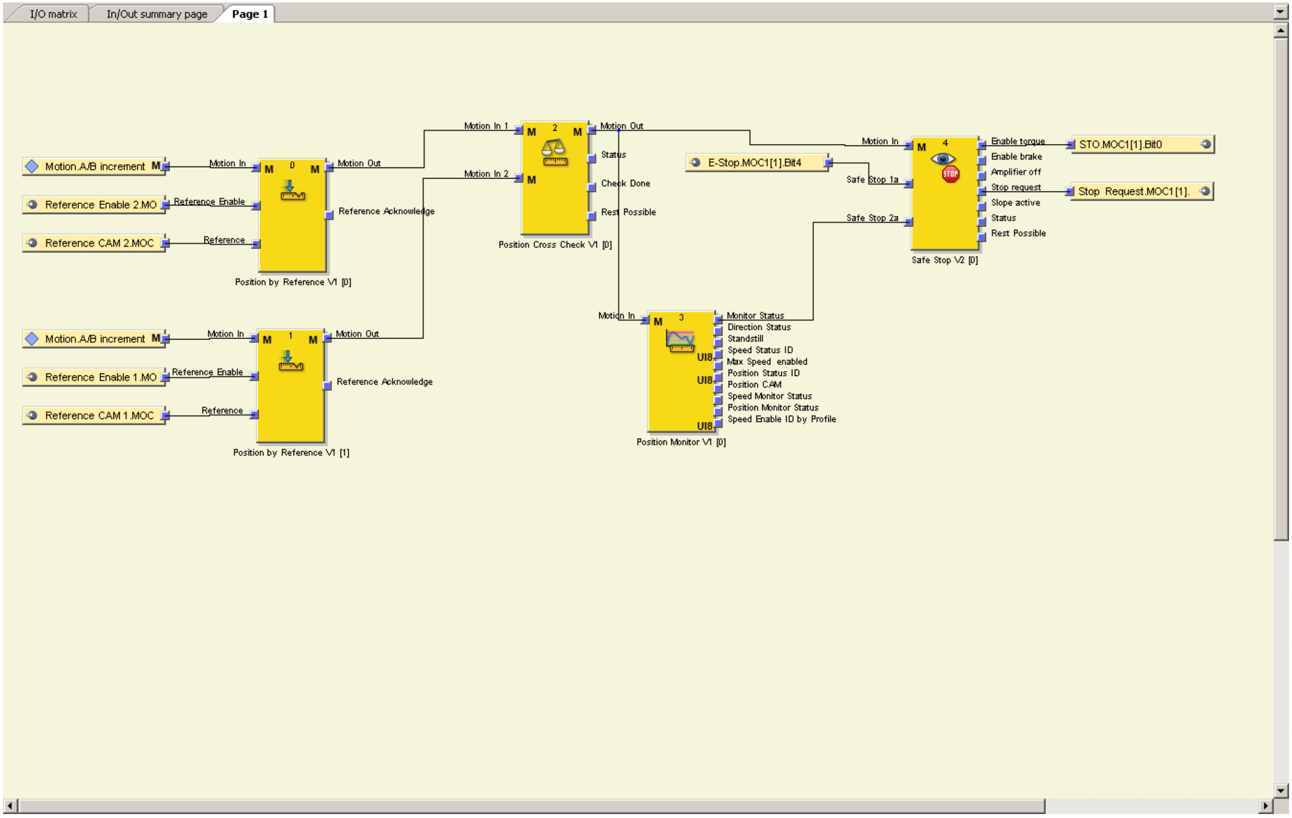


Figure 244: Evaluation of position using two A/B incremental encoders, each with one reference signal (cam) and reference run on each system start

Programming example 4: Evaluation of position using a safe Sin/Cos encoder (e.g., DFS60S Pro), a safe reference signal (cam) and a reference run on each system start

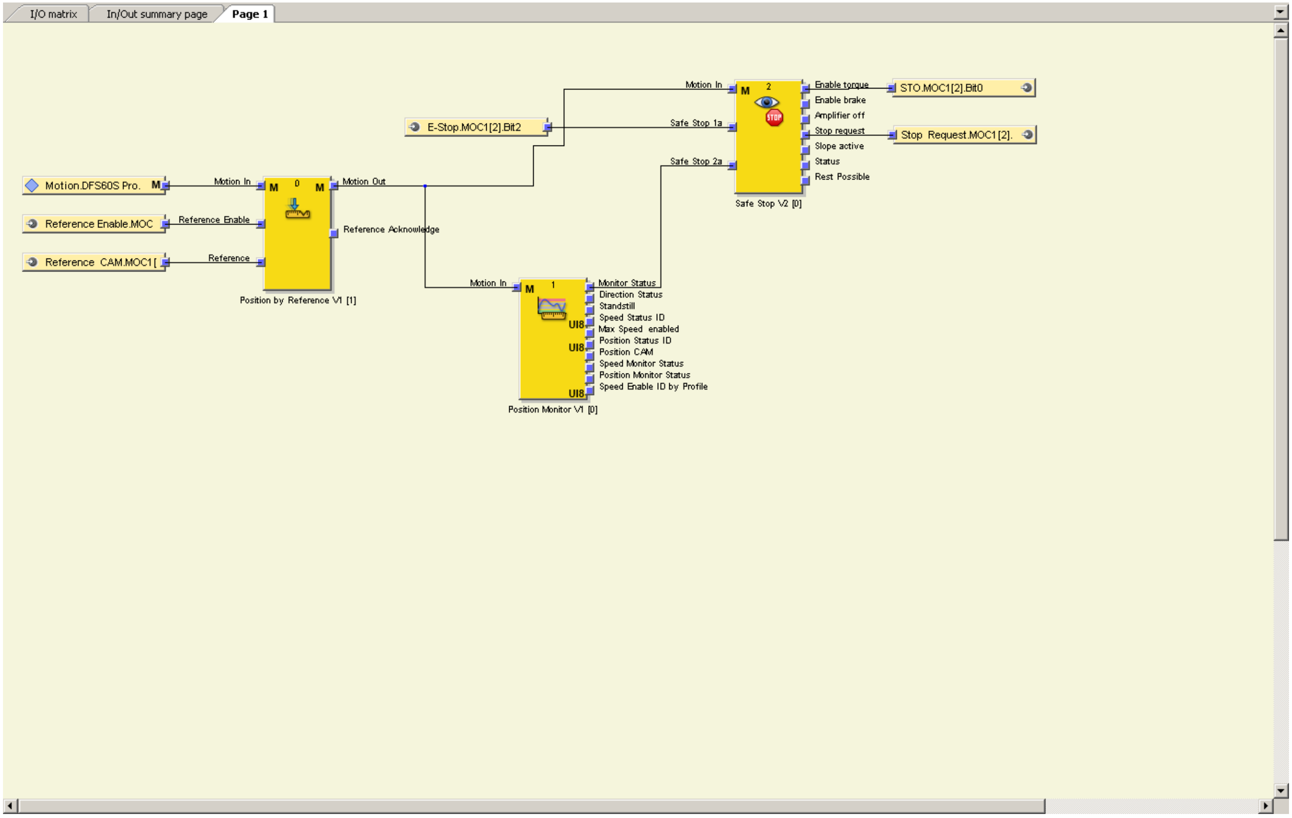


Figure 245: Evaluation of position using a safe Sin/Cos encoder (e.g., DFS60S Pro), a safe reference signal (cam) and a reference run on each system start

Programming example 5: Evaluation of position using a safety encoder with absolute position (SSI+Sin/Cos encoder) with initial reference run on commissioning of the machine

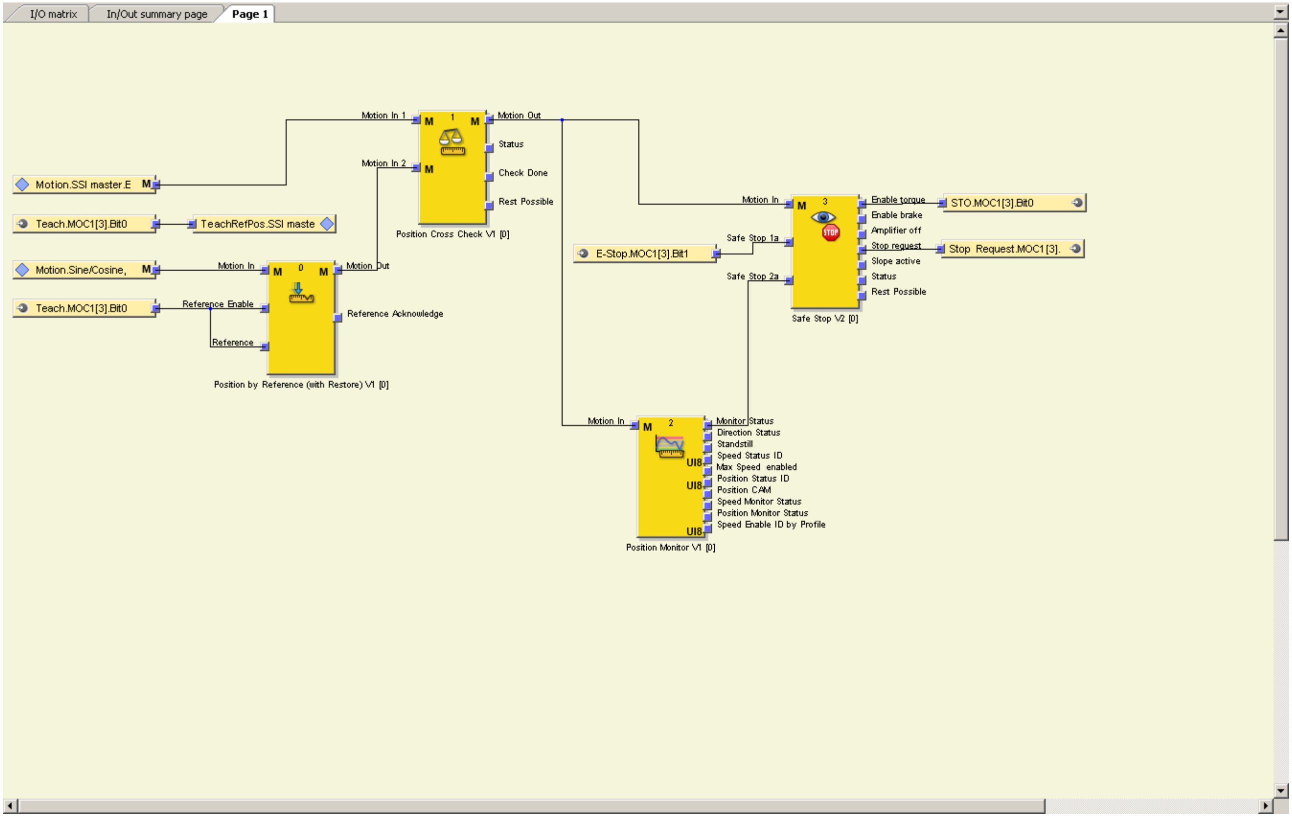


Figure 246: Evaluation of position using a safety encoder with absolute position (SSI+Sin/Cos encoder) with initial reference run on commissioning of the machine

Repressing error messages and error responses

Programming example 6: Repress error response using a safety encoder with absolute position (SSI+Sin/Cos encoder)

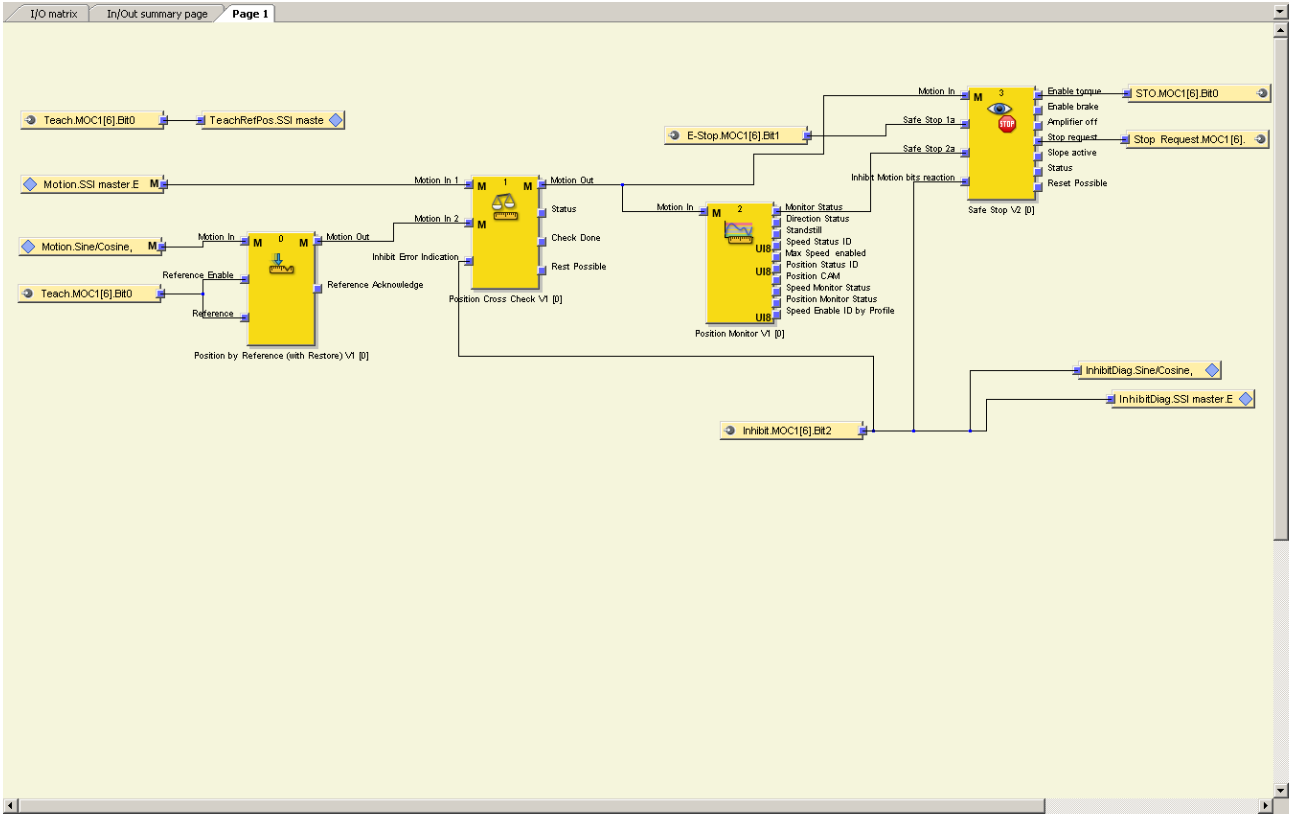


Figure 247: Repress error response using a safety encoder with absolute position (SSI+Sin/Cos encoder)

11.10.3 Position by reference V1

Function block diagram

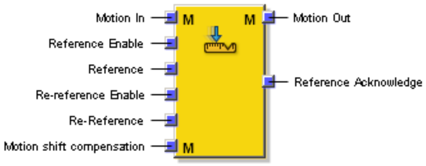


Figure 248: Inputs and outputs of the Position by reference function block (with and without restore function) V1

General description

The **Position by reference V1** function block is used for generating an absolute position based on the following factors:

- A relative position (e.g., from a Sin/Cos encoder or an A/B incremental encoder)
- A reference signal (e.g., from a reference switch in combination with a reference cam or an actuator operated by the user)

The absolute position is calculated by accumulating the relative position at the **Motion In** input. The start value is always determined using a reference signal, which must be clearly identifiable throughout the entire range of motion.

NOTE

The absolute position must not exceed the permitted data range, i.e., the maximum possible value for the absolute position at the **Motion out** output (-2^{31} … $2^{31}-1$). Otherwise, the absolute position becomes invalid and referencing must be performed again.

The **Position by reference V1** function block is available both with and without a restore function. The restore function allows the user to restore the absolute position after an operational interruption without performing referencing again (see "Restoring the absolute position", page 338).

Inputs of the function block

Table 154: Inputs of the Position by reference V1 function block

Input	Description	Signal value
Motion In	To record the relative position of the monitored vehicle or machine on an ongoing basis	Motion V2 data, either directly from an encoder or from another function block; e.g., **Position cross check V1**.
Reference enable	Starts and ends the reference process	1 = enable
Reference	Sets the absolute position to the configured reference position. For a detailed description, see see "Referencing", page 334.	Rising signal edge
Re-reference enable	Starts and ends the re-reference process	1 = enable
Re-reference	Sets the absolute position to one of the configured Re-reference positions. For a detailed description, see see "Re-reference", page 337.	Rising signal edge
Motion shift compensation	Compensating a possible position shift when restoring the absolute position after restarting the Flexi Soft system.	Motion V2 data, directly from an encoder

Outputs of the function block

Table 155: Outputs of the Position by reference V1 function block

Output	Description	Signal value
Motion Out	Motion data including the absolute position for use in subsequent function blocks	Output of data from the Motion In input, supplemented by the value and the status bits for the absolute position
Reference acknowledge	Indicates whether the start value for the absolute position has been successfully set.	0 = No start value set 1 = start value set

Function block parameters

Table 156: Parameters of the Position by reference V1 function block

Parameter	Description	Possible values
Referencing		
Reference position	Start value for calculating the absolute position	**–2,147,483,648 ... +2,147,483,647 digits =** • +/– 71,583 rpm • +/– 8,590 m
Re-referencing		
Re-reference position	Values for re-referencing the absolute position	**–2,147,483,648 ... +2,147,483,647 digits =** • +/– 71,583 rpm • +/– 8,590 m
Re-reference position tolerance	Tolerance range for valid re-referencing	**0 ... 500,000,000 digits =** • 0 ... 16,666 rev. • 0 ... 2,000,000 mm
Restoring the absolute position		
Configuration CRC	Internal parameter, not configurable, represents the configuration components required to restore the absolute position after an operational interruption.	0000 to FFFF
Shift compensation		
Tolerance Shift compensation	Tolerance range for valid shift compensation	**0 ... 500,000,000 digits =** • max. 16,666 rev. • max. 2,000,000 mm • 0 = Deactivated

Referencing

The referencing process sets the absolute position of the function block to the start value. This is determined by the **Reference position** parameter.

The referencing process is required in the following cases as a minimum:

When using the Position by reference function block (without restore function):

- Every time the Flexi Soft system has been restarted

When using the Position by reference function block (with restore function):

- During the initial commissioning of the machine
- Every time the FX3-MOC1 is replaced

The referencing procedure is started by a rising signal edge (0–1) at the **Enable referencing** input and ended by a falling signal edge (1–0) at the **Enable referencing** input. A 1 in the first execution cycle is evaluated as a rising edge.

At the beginning of the reference process, the absolute position at the **Motion Out** output of the function block is set to invalid and unreliable.

A rising signal edge at the **Reference** input during an active reference process sets the absolute position value to the starting value specified by the **Reference position** parameter. The rising signal edge at the **Reference** input is accepted even if the **Reference enable** input switches to 1 in the same cycle.

The **Reference acknowledge** output is then set to 1 and the function block starts calculating the absolute position internally by using the relative position data at the **Motion In** input. If several rising signal edges occur at the **Reference** input before the reference process ends, the previously calculated values are discarded and the function block restarts its calculations from the reference position.

As a result, it is always the last rising signal edge that takes effect at the **Reference** input. This may be the case if the reference cam is overrun multiple times during the referencing procedure.

A falling signal edge (1–0) at the **Enable referencing** input ends the referencing procedure. Additional rising signal edges at the **Reference** input have no further effect. It is only after this point that the calculated absolute position is output at the **Motion out** output.

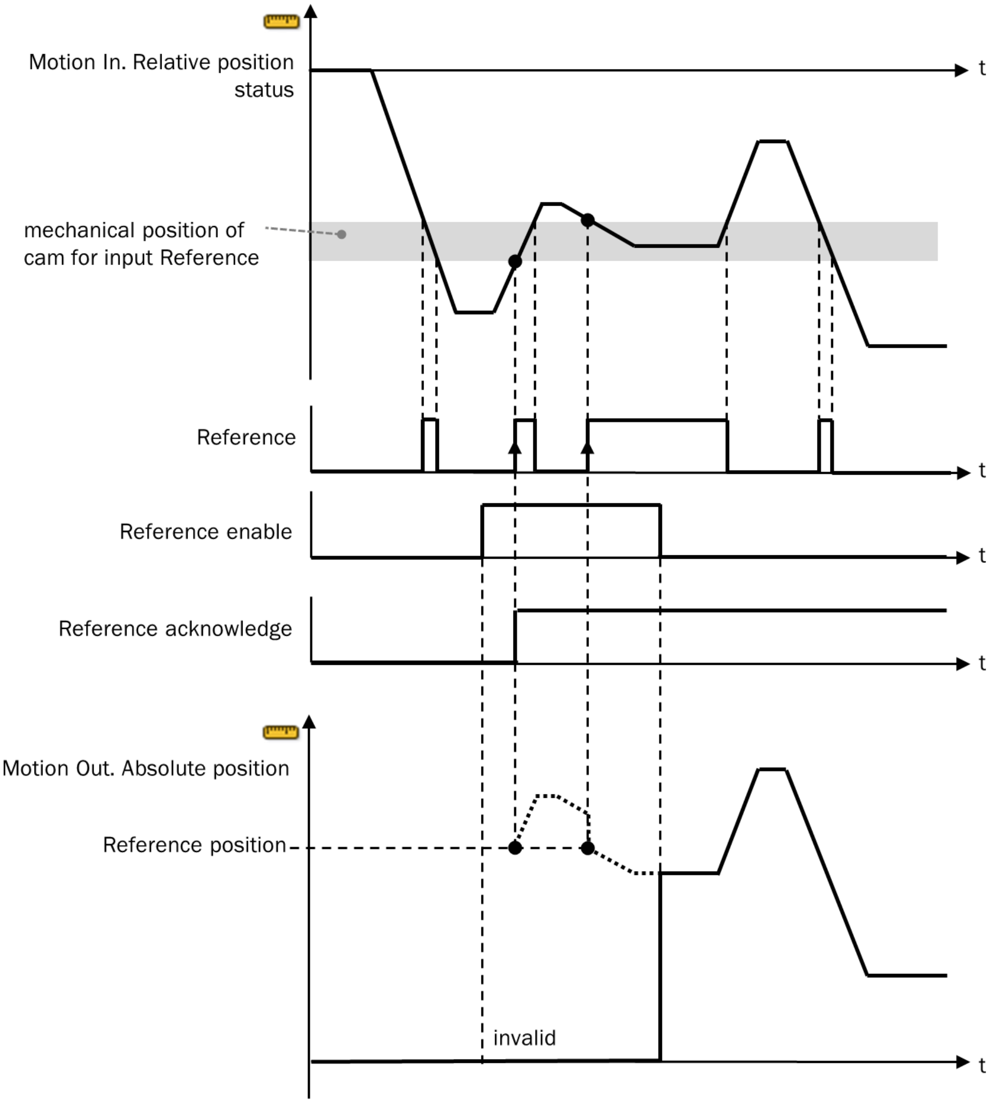

Motion In. Relative position status
t
mechanical position of cam for input Reference
Reference
t
Reference enable
t
Reference acknowledge
t
Motion Out. Absolute position
Reference position
invalid
t

Figure 249: Referencing procedure

If a reference process is ended without a rising signal edge at the **Reference** input, the absolute position value at the **Motion Out** output remains invalid and an entry is generated for the diagnostics history.

NOTE

The **Reliability reference signal** parameter can be used to configure whether the signal source for the **Reference** input is a reliable signal. If this is the case, and the relative position data at the **Motion In** input is reliable, the absolute position at the **Motion Out** output is also reliable. Otherwise, the reliability of the absolute position must be checked separately; e.g., via a **Position cross check V1** function block.

Duration of the reference signal

In order to be detected reliably, the signal at the **Reference** input must be at least as long as the logic execution time or the input updating time – whichever is higher. This determines the maximum speed for the reference process, depending on the mechanical width of the reference signal:

Maximum speed = width of the reference signal/input update time

Table 157: Maximum speed examples for the reference process

Type of reference signal	Width of reference signal	Logic execution time or input updating time	Maximum speed
Zero pulse with 90° electrical period (1 period = 360°), with 1,024 periods per revolution	1/4,096 revolutions	4 ms	0.061 rev/s = 3.6 rpm
		20 ms	0.012 rev/s = 0.7 rpm
Proximity switch	1 cm	4 ms	2.5 m/s = 9 km/h
		20 ms	0.5 m/s = 1.8 km/h

Referencing accuracy

Depending on the direction of movement, either the left or the right edge of the reference cam can trigger a rising signal edge at the **Reference** input. If the direction of movement during the reference process is unknown, the resulting absolute position may feature an additional inaccuracy in the distance between the two switching points of the reference cam. The **Reference enable** input can be used to control the reference process in such a way that referencing is only carried out on the required side of the reference cam.

The **Direction status** output of a **Speed monitoring V2** or **Position monitoring V1** function block, for example, may be used for this purpose.

In order to ensure that the referencing process is accurate, the various signal propagation delays at the **Motion in** and **Referencing** inputs must be taken into account. If the speed is not 0, the mechanical position may change as a result of the different signal propagation delays before the rising signal edge at the **Reference** input becomes effective.

If a Position cross check function block is used to cross check the absolute position of this function block against the position signal of another source, the position shift caused by inaccurate referencing may lead to an error, depending on the selected position tolerance.

WARNING
Unintentional referencing

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

- Prevent unintentional referencing, e.g. via one of the following measures:
  - Carry out a check using a second signal source (encoder) with the aid of a **Position comparison V1** function block.
  - Limit the ability to set the **Reference enable** input to 1, e.g. release by a PLC only during specific time windows.
  - Pre-evaluate the signal for the **Reference** input, e.g. using a filter.

Re-reference

A re-referencing run can be executed during operation to correct the absolute position of the function block within a specific tolerance and to set the position to a pre-defined **Re-reference position**. The **Re-reference position tolerance** parameter indicates the maximum range around the re-reference position within which the re-referencing process can correct the position. Up to eight different re-reference positions can be configured.

NOTE

If multiple re-reference positions are configured, the **Re-reference position tolerance** may not exceed a quarter of the distance between the two closest reference positions.

Re-referencing can be performed only under the following conditions:

- The **Enable re-referencing** input is 1.
- The **Enable referencing** input is 0.

If these conditions are met and a rising signal edge occurs at the **Re-reference** input, the function block checks that the current absolute position is within a configured re-referencing range, i.e., that the position deviates from the nearest **Re-reference position** by no more than the **Re-reference position tolerance.** If this is the case, the absolute position is set to this **Re-reference position**. Otherwise, the re-reference signal is ignored.

No check is performed to determine whether a signal edge has occurred at the **Re-reference** input of each configured re-reference range.

Re-referencing can be performed several times as long as the **Re-reference enable** input is 1.

Unlike during referencing, the **Motion Out** output is not explicitly set to invalid during re-referencing while the **Enable re-reference** input is set to 1. As a result, the **Enable re-reference** input may remain static at 1 if re-referencing is to be performed on an ongoing basis.

NOTE

The **Reliability re-reference signal** parameter can be used to configure whether the signal source for the **Re-reference** input is a reliable signal. If this is the case, and the relative position data at the **Motion In** input is reliable, the absolute position at the **Motion Out** output is also reliable after re-referencing. Otherwise, the reliability of the absolute position must be checked separately; e.g., via a **Position cross check V1** function block.

This requires the **Reliability reference signal** parameter to be enabled as well.

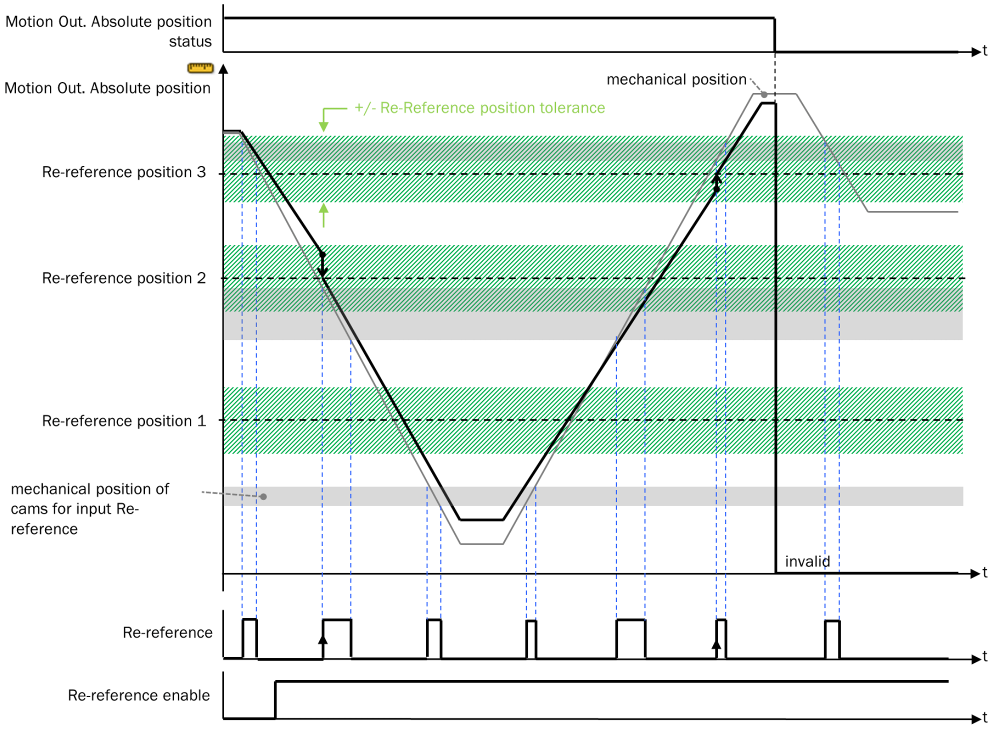

Motion Out. Absolute position status
t
Motion Out. Absolute position
mechanical position
+/- Re-Reference position tolerance
Re-reference position 3
Re-reference position 2
Re-reference position 1
mechanical position of cams for input Re-reference
invalid
t
Re-reference
t
Re-reference enable
t

Figure 250: Re-referencing procedure

Re-referencing accuracy

Depending on the direction of movement, either the left or the right edge of the re-reference cam can trigger a rising signal edge at the **Re-reference** input. If the direction of movement during the reference process is unknown, the resulting absolute position may feature an additional inaccuracy in the distance between the two switching points of the re-reference cam. The **Re-reference enable** input can be used to control the re-reference process in such a way that re-referencing is only carried out on the required side of the re-reference cam.

In order to ensure that the re-referencing process is accurate, the various signal propagation delays at the **Motion in** and **Re-referencing** inputs must be taken into account. If the speed is not 0, the mechanical position may change as a result of the different signal propagation delays before the rising signal edge at the **Re-reference** input becomes effective.

Restoring the absolute position

NOTE

- The function for restoring the absolute position is only available with the **Position by reference (with restore) V1** function block.
- The **Position by reference (with restore function) V1** function block can only be used once for each FX3-MOC1 module.

In the event of an operational interruption, the **Position restore** function enables you to continue working with the same absolute position that was valid before the interruption, without the need to perform a new reference process.

The most recently valid absolute position is recorded for this purpose. If the Flexi Soft system is stopped, the recorded value is stored in the non-volatile memory of the FX3-MOC1.

NOTE

The system always saves the last **valid** position. In some cases, this means that the absolute position can be restored even if the position value was invalid before the Flexi Soft system was stopped (e.g., because the minimum supply voltage for the encoder system fell below the required level faster than the supply voltage for the Flexi Soft system).

The next time you start the Flexi Soft system, the stored value is restored and used as the start value for the absolute position, provided that the following conditions are met:

- The stored absolute position is valid.
- The relative position at the **Motion In** input is valid.

If you have successfully restored the absolute position, the **Reference acknowledge** output is then set to 1 and the function block starts calculating the absolute position by using the relative position data at the **Motion In** input.

NOTE

The restored absolute position is marked as unreliable in any case.

WARNING

Unreliable absolute position

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

One of the following measures must be implemented for a reliable absolute position:

► Carry out a plausibility check (e.g. using a second signal source for the absolute position and a **Position comparison V1** function block)

► Referencing with a reliable reference signal

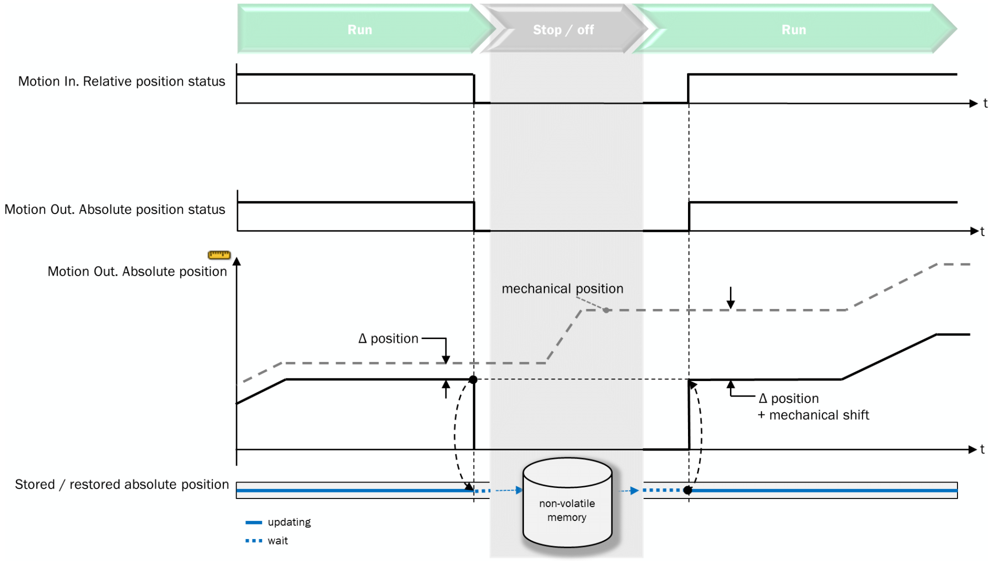

Run
Stop / off
Run
Motion In. Relative position status
t
Motion Out. Absolute position status
t
Motion Out. Absolute position
mechanical position
Δ position
Δ position
+ mechanical shift
t
Stored / restored absolute position
non-volatile memory
updating
wait

Figure 251: Restoring the absolute position without shift compensation

Restoring the absolute position after changing the configuration

It is possible to restore the absolute position even if other parts of the Flexi Soft configuration have been changed. This enables you to adapt the configuration during commissioning, for example, without the need to perform a new referencing process after each change.

A **configuration CRC** is calculated for this purpose. This represents the configuration components required to restore the position. If the value of the **configuration CRC** has changed compared to a previous configuration, the stored absolute position is set to invalid and referencing must be performed again:

The **Configuration checksum** is displayed as a parameter for this function block in the report.

In the following cases, the stored absolute position is set to invalid and referencing must be performed again:

- The configuration of this function block has been changed.
- The configuration of the encoder or the function blocks acting as sources for the **Motion in** input or the **Motion shift compensation** input was modified.

NOTE

When a referencing procedure is started (**Enable referencing** input switches to 1), the stored absolute position is set to invalid. An absolute position is only regenerated when the referencing procedure is completed successfully.

Shift compensation

The optional **Shift compensation** function can be used to compensate for any minor electrical counting inaccuracies at the **Motion in** input when the encoder is switched on or off, or any minor mechanical shifts in position that occur during an operational interruption. In addition to the source for the relative position (e.g., an A/B incremental encoder), a source that also indicates the mechanical position – even if this position changed during the operational interruption – is also required for the absolute position. This generally takes the form of a second source that is installed for plausibility checks (e.g., an SSI encoder or the SSI components of a safe SSI+Sin/Cos encoder).

To use the **Shift compensation** function, the second source must be connected to the **Motion shift compensation** input. If this is the case, the absolute position at this input is also stored in the non-volatile memory of the FX3-MOC1, and taken into account when the Flexi Soft system is restarted or if the absolute position is restored.

The following conditions must be fulfilled when saving the values:

- The **Shift compensation** function is active.
- The absolute position at the **Motion Out** output is valid.

The following conditions must be fulfilled for restoring the values:

- The **Shift compensation** function is active.
- The stored absolute position is valid.
- The relative position at the **Motion In** input is valid.

If these conditions are not fulfilled for more than 60 seconds before saving, or after the Flexi Soft system is restarted, then the stored shift compensation position is set to invalid. In this case, the shift compensation function has failed. This means that the absolute position which is issued at the **Motion Out** output is restored; if necessary, without shift compensation.

If a valid absolute position was restored for shift compensation following a restart of the Flexi Soft system, the function block compares this restored value with the current valid position at the **Motion shift compensation** input. If the difference between these two positions does not exceed the configured **Shift compensation tolerance**, then the absolute position output at the **Motion out** output is corrected by this difference. If the configured **Shift compensation tolerance** is exceeded, then the shift compensation system fails; the absolute position output at the **Motion out** output is restored, where necessary, without any shift compensation being applied.

NOTE

Regardless of whether shift compensation has been successful, the restored absolute position is always marked as unreliable.

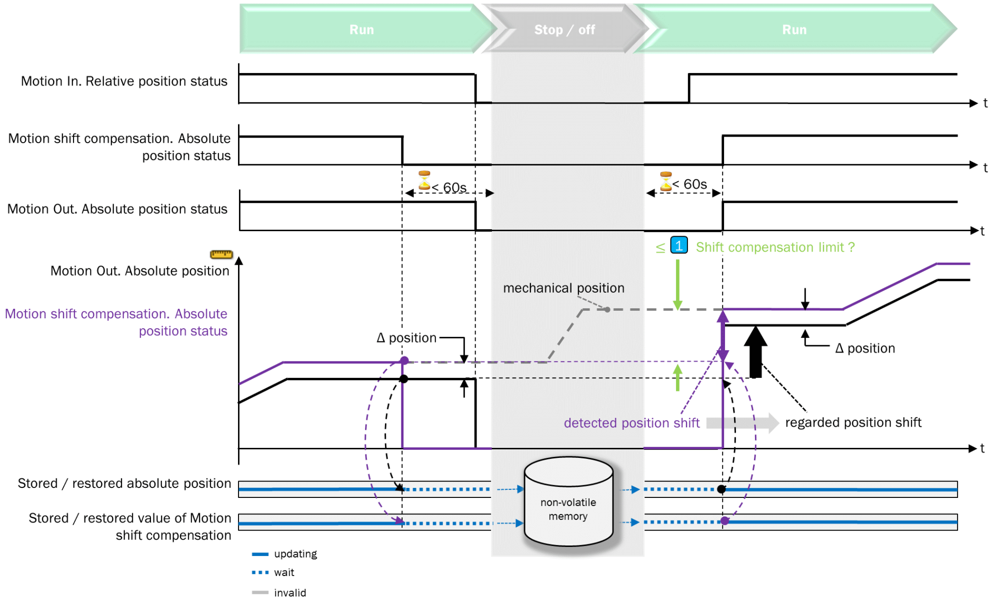

Run
Stop / off
Run
Motion In. Relative position status
Motion shift compensation. Absolute position status
Motion Out. Absolute position status
< 60s
< 60s
t
t
t
t
≤ 1 Shift compensation limit ?
Motion Out. Absolute position
Motion shift compensation. Absolute position status
mechanical position
Δ position
Δ position
detected position shift
regarded position shift
Stored / restored absolute position
Stored / restored value of Motion shift compensation
non-volatile memory
updating
wait
invalid

Figure 252: Restoring the absolute position with shift compensation

NOTE

If the source for the **Motion shift compensation** input is also used to perform a position comparison with the absolute position (calculated using the **Position by reference V1** function block), this source acts as the sole source for the absolute position during the operational interruption.

WARNING

Malfunction due to faulty position detection

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

- ► Potential errors during the phase between the last valid position value before the interruption and the first valid position value after the restart must therefore be considered separately or excluded.
  - o Potential electrical errors can be excluded if the encoder for the **Motion shift compensation** input is not supplied with power. For this reason, invalid position values – both before the operational interruption and after the restart – are only tolerated for a maximum of 60 seconds.
  - o Potential errors during operation can be detected using a **Position comparison V1** function block.

11.10.4 Position cross check V1

Function block diagram

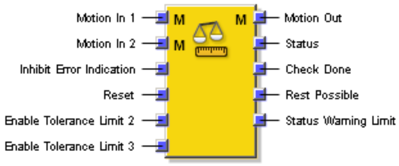


Figure 253: Inputs and outputs of the Position comparison V1 function block

General description

The **Position cross check V1** function block compares position values from two different signal sources. The associated performance checks are used to achieve a higher level of safety, particularly when working with non-safe encoders.

This function block compares the absolute positions from both signal sources, as long as both sources have a validity status of 1 (valid) and the monitoring mode for the absolute position is activated (comparison mode = same detection direction or opposite detection direction). If a positive result is obtained, the reliability status for the absolute position, the relative position and the speed is set to 1 (reliable).

Otherwise, the relative positions are compared, as long as both positions have a validity status of 1 (valid) and the function block is not deactivated (comparison mode ≠ deactivated). If a positive result is obtained, the reliability status for the relative position and the speed is set to 1 (reliable). This means, for example, that the system can execute a reference run at a safely limited speed even if the absolute position is invalid.

NOTE
The positions of both signal sources must always have a fixed ratio, with a small drift tolerated.

Inputs of the function block

Table 158: Inputs of the Position cross check V1 function block

Input	Description	Signal value
Motion In 1 Motion In 2	For connection of two encoders, with a downstream **Position by reference** function block if applicable.	Motion V2 data
Inhibit error indication	Optional input; inhibits a diagnostics history entry if a position cross check fails	0 = no inhibiting 1 = error indication inhibited
Reset	Optional input; resets the monitoring functions of the function block	Rising signal edge (0–1)
Enable tolerance limit 2 and Enable tolerance limit 3	Optional inputs for selecting increased tolerances for the position cross check if needed.	0 = no enable 1 = enable

Outputs of the function block

Table 159: Outputs of the Position cross check V1 function block

Output	Description	Signal value
Motion Out	Outputs the checked Motion V2 data for use in another function block; e.g., **Position monitor V1**. Depending on the function block configuration, it may be possible to delay the output of the values.	Values from **Motion In 1** if they are valid, otherwise values from **Motion In 2**. The validity and reliability bits are set depending on the results of the position cross check.

Output	Description	Signal value
Status	Indicates whether the position cross check function has failed; for example, in locating errors. The initial status when the Flexi Soft system transitions to the Run status is 1. The output switches to 0 if the deviation determined during a position cross check exceeds the selected position tolerance. The output switches back to 1 when the determined deviation is again lower than or equal to the selected tolerance, but only after the **Error recovery time** of 1 s has elapsed. If the relative position at the **Motion in 1** input or the **Motion in 2** input becomes invalid, then the **Status** output immediately switches to 1, as no evaluation can be performed in this state and the error is further ahead in the signal path.	0 = error detected 1 = OK (no error detected or status unknown)
Check done	The output switches to 1 when the position comparison is performed for the first time. The output switches back to 0 when the relative position at the **Motion in 1** input or the **Motion in 2** input is invalid, meaning that position comparison cannot be carried out.	0 = Position comparison not performed 1 = position cross check executed
Reset possible	Indicates whether resetting via the **Reset** input is possible.	0 = reset not possible 1 = reset possible
Warning limit status	The output switches to 0 if the deviation determined during the position cross check exceeds the **Warning limit** parameter. The output switches back to 1 if the deviation determined is lower than or equal to the **Warning limit** parameter. The initial status when the Flexi Soft system transitions to the Run status is 1.	0 = warning 1 = OK

Function block parameters

Table 160: Parameters of the Position cross check V1 function block

Parameter	Description	Possible values
Cross check mode		
Cross check mode	Defines the type of position cross check or deactivates the function	• Deactivated • Same direction detection • Opposite direction detection • Relative position only
Motion In 1 position – Motion In 2 position	Constant position difference between **Motion in 1** and **Motion in 2** with position comparison running in the same direction (comparison mode = same detection direction)	**–2,147,483,648 ... +2,147,483,647 digits =** • +/– 71,583 rev. • +/– 8,590 m
Motion In 1 position + Motion In 2 position	Constant position total of **Motion in 1** and **Motion in 2** with position comparison running in opposite directions (comparison mode = opposite detection direction)	**–2,147,483,648 ... +2,147,483,647 digits =** • +/– 71,583 rev. • +/– 8,590 m
Interpolation mode	Activation of interpolation mode for position cross check	• Disabled • Active
Motion In 1 delay	Displays the internal delay for **Motion In 1** and **Motion In 2** which is active for internal evaluation and for producing the output at the **Motion Out** output	0 ... 4 ms
Motion In 2 delay		
Position tolerances		
Position tolerance limit 1 ... 3	Permitted deviation during position comparison. If multiple position tolerances are configured, these tolerances can be selected using the optional **Enable tolerance limit 2 ... 3** inputs.	**0 ... 1,073,741,823 digits =** • +/– 35,791 rev. • +/– 4,295 m

Parameter	Description	Possible values
Warning limit	If the warning limit is exceeded, the **Warning limit status** output switches to 0	**0 ... 1,073,741,823 digits =** • +/– 35,791 rev. • +/– 4,295 m
Relative position tolerance		
Drift time	Enables you to compensate for a slow change in distance (drift) between the positions of two encoders during a relative position cross check	1 ... 60 s 0 = deactivated

Description of operation

The **Position comparison V1** function block compares the position values at the **Motion in 1** and **Motion in 2** inputs. The function block can also take into account the tolerance limits that can be configured by the user. Depending on the result of the comparison, the function block sets the reliability bit at the **Motion out** output and switches the **Status** output.

The Position cross check V1 function block compares the following position values:

- Two absolute position values (absolute position cross check):
  This is useful for encoder systems that supply absolute position values, either automatically (e.g., via an SSI interface) or based on a combination of A/B-incremental encoders and a referencing procedure using the **Position by reference V1** function block.
  The absolute position comparison can be configured to check whether the difference between or the total of the two position values is within the expected tolerance range. Deviations due to slippage or friction in the application must be within the tolerance limits. For absolute position comparison, this can be achieved, for example, using the **Re-referencing** function of the **Position by reference V1** function block.
- Two relative position values(relative position cross check):
  The relative position cross check is only relevant if it is not possible to execute an absolute position cross check. This can occur, for example, if there are no two absolute position values that are valid, or if the **Relative position only** monitoring mode is selected. A relative position cross check is therefore usually executed with A/B incremental encoders.
  Deviations due to slippage or friction in the application must be within the tolerance limits. For relative position comparison, this can be achieved, for example, using the **Drift time** or **Reset** functions of this function block.
  Relative position comparison can also be used in applications that generate an absolute position from a relative position and an initial referencing procedure. This approach allows the user to control the relative position and therefore, implicitly, the speed of the system too.

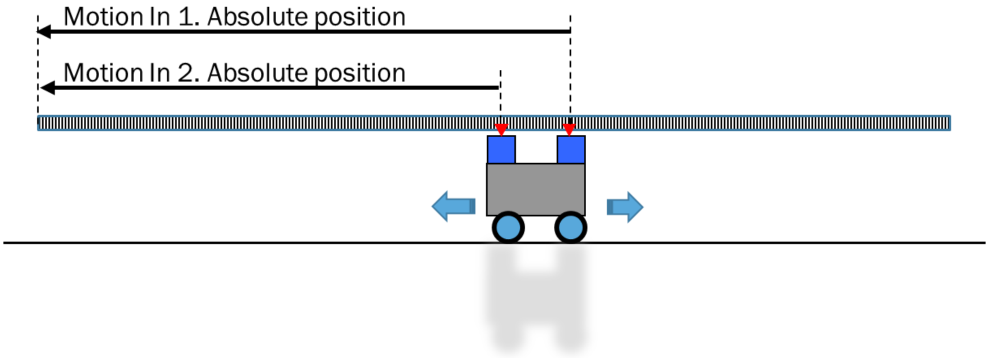

Motion In 1. Absolute position
Motion In 2. Absolute position

Figure 254: Position comparison with the same detection direction

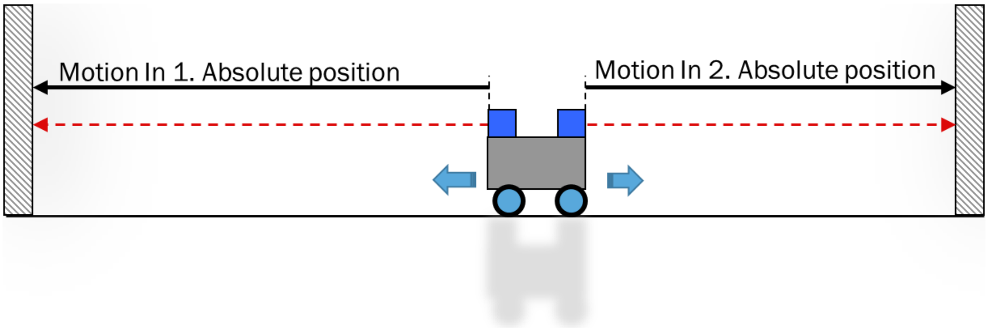

Motion In 1. Absolute position
Motion In 2. Absolute position

Figure 255: Position comparison with opposite detection directions

The absolute position must be valid at both inputs for an absolute position cross check; the relative position must be valid at both inputs as a minimum for a relative position cross check.

Position tolerances

Up to three different process tolerance limits can be configured:

- Position tolerance limit 1: Permanent position tolerance
- Position tolerance limit 2: First increased position tolerance
- Position tolerance limit 3: Second increased position tolerance

The increased position tolerances can be selected using the optional **Enable tolerance limit 2** and **Enable tolerance limit 3** inputs.

Table 161: Selecting the position tolerance

Enable tolerance limit 3 input	Enable tolerance limit 2 input	Selected position tolerance
0 or input not activated	0 or input not activated	Position tolerance 1
0 or input not activated	1	Position tolerance 2
1	Any (1 or 0)	Position tolerance 3

The selected position tolerance limit is used for both the absolute position cross check and the relative position cross check. However, the calculation method is different in each case.

Absolute position cross check

In absolute position comparison, depending on the comparison mode selected, the system calculates the deviation based on the difference between the values, or the total of the two absolute position values. The resulting value is then compared with the current position tolerance:

- Comparison in same direction: Deviation = (absolute position **Motion in 2** – absolute position **Motion in 1**) – parameter **Motion in 2 position – Motion in 1 position**
- Comparison in opposite directions: Deviation = (absolute position **Motion in 2** + absolute position **Motion in 1**) – parameter **Motion in 2 position + Motion in 1 position**

If interpolation is active, the internal delay and interpolation values are taken into account in the calculation.

The comparison is deemed successful if the absolute value of the relevant result does not exceed the current position tolerance.

Both comparison modes are also compatible with periodic positions, e.g., for rotating tables or eccentric presses.

NOTE

The calculated deviation in the absolute position can be read in the online monitoring system of the FX3-MOCx Logic editor and in the Flexi Soft data recorder (under **Absolute position deviation.**

Relative position cross check

With relative position comparison, the difference between the relative position values of both encoders is calculated on an ongoing basis. The calculation takes into account whether the encoders are configured for the same or for opposite directions. This means that the value of **Motion 2** is reversed when the **Opposite detection directions** comparison mode is configured.

The system checks whether the fluctuation in the difference is lower than twice the position tolerance. For this purpose, it stores the highest and the lowest position difference values. Half the difference between these two valuesis used as a cross check value for the relative position cross check. The cross check is successful as long as this half-difference is lower than the currently selected position tolerance.

Constant deviation between the two relative positions does not affect this calculation. Something like this may occur if, for example, both sources have been valid at different times.

NOTE

- In the case of the relative position cross check, the difference is halved due to the deviating calculation method, so that the relative position cross check satisfies the same criteria as the absolute position cross check.
- The calculated deviation in the relative position (half difference) can be read in the online monitoring system of the FX3-MOCx Logic editor and in the Flexi Soft data recorder (under **Relative position deviation.**

Warning limit

In addition to the position tolerance limits, you can configure a **Warning limit.**

The **Warning threshold status** output switches to 0 if the calculated deviation exceeds the **Warning threshold** parameter during a position comparison. The output switches back to 1 when the calculated deviation once again drops below or is equal to the **Warning threshold.**

The **Warning threshold status** output can be used for applications in which a measured position differs from the actual position due to slippage or friction and in which this issue can be corrected via re-referencing (e.g., using the **Re-referencing** function of the **Position by reference V1** function block). This output indicates that re-referencing is necessary before the position comparison fails.

Drift time

The **Drift time** allows the user to compensate for slow changes in distance (drift) between the positions of the two encoders in relative position comparisons. Drift may be caused by a number of factors, including slippage or friction in the application.

When the **Drift time** parameter is activated (**Drift time** > 0), the saved highest and lowest values are continually adjusted to the actual measured value during the relative position comparison. If the highest measured position difference is lower than the highest saved position difference, the saved value is slowly reduced. The same applies in reverse to the lowest saved position difference.

The speed at which the value changes depends on the value of the **Drift time** parameter. The higher this value, the slower the speed at which the function works.

The process of adjusting the value follows a filter function. When the difference in the relative position remains constant, then the previously calculated deviation will have almost completely evened out by the time three **Drift times** have elapsed.

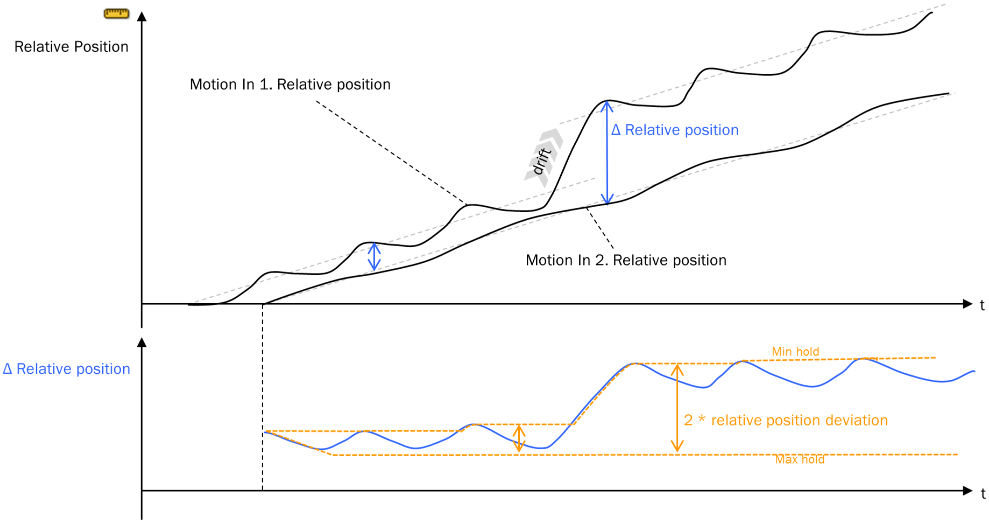


Figure 256: Relative position comparison without drift time

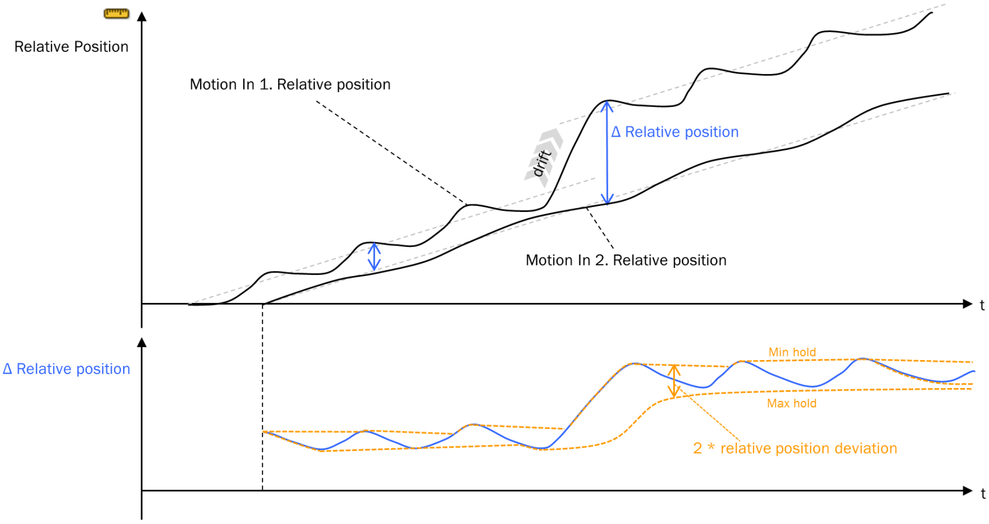


Figure 257: Relative position comparison with drift time

As speed is calculated based on the relative position, by comparing the relative position, the system is, implicitly, also checking the speed.

When the **Drift time** parameter is used, the speed tolerance increases during the relative position comparison. This means that the speeds of both sources for **Motion in 1** and **Motion in 2** around the **Drift time** may differ without causing any error in the relative position comparison. However, this only applies when no absolute position comparison is performed (e.g., because this function is deactivated or cannot be executed).

The additional speed tolerance can be calculated using the following formula:

Additional speed tolerance = position tolerance/drift time

Table 162: Additional speed tolerance depends on drift time

Position tolerance	Drift time	Additional speed tolerance
1 mm	1 s	1 mm/s = 0.001 m/s
10 mm	1 s	10 mm/s = 0.01 m/s

Position tolerance	Drift time	Additional speed tolerance
100 mm	1 s	100 mm/s = 0.1 m/s
90° = 1/4 rev	1 s	1/4 rev/s= 15 rpm
90° = 1/4 rev	10 s	1/40 rev/s= 1.5 rpm

WARNING
Increased speed tolerance through drift time

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

► Take into account the additional speed tolerance.

NOTE
The **Drift time** parameter has no influence on the absolute position comparison.

Interpolation mode

Depending on the encoders used, the detection time, the refresh interval, and the signal propagation delay may differ at the two inputs. This point is of particular relevance when different encoder types are used. For example, due to the transmission time for SSI telegrams, an SSI encoder has a higher latency than an A/B incremental encoder. In particular, an SSI encoder also may have different detection times in listener mode.

The Interpolation option can be used to minimize these influences:

- Influences due to different signal propagation delays are minimized on the basis of the configured encoder type, by means of an internal delay affecting the path that is faster for the cross check in each case.
- These variations in detection time are compensated for by the system retrospectively calculating an interpolated position value for each encoder in turn, based on the two most recent position values; this calculated value is then compared with the last position value of the other encoder. For this to take place, the system must have previously recorded values that permit interpolation in combination with the current position value. The **Comparison done** output indicates when this condition is met for the first time following the Flexi Soft system's transition to run status.

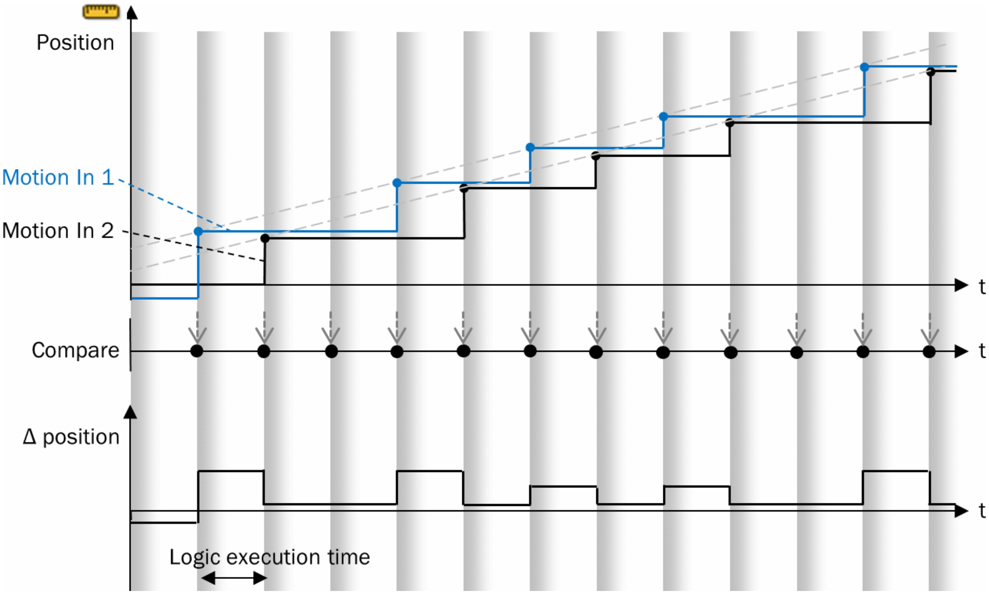


Figure 258: Position comparison without interpolation

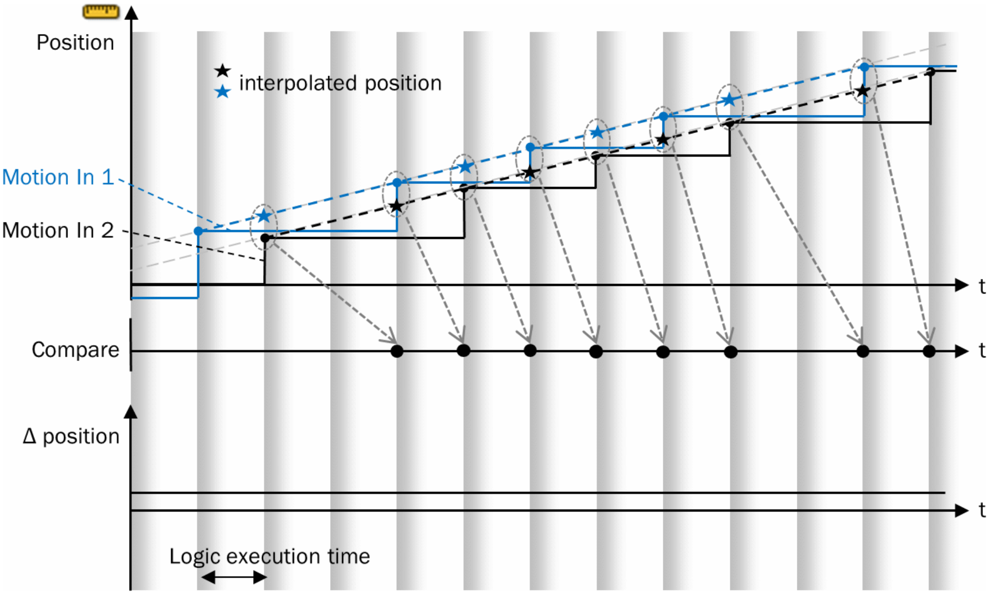


Figure 259: Position comparison with interpolation

Output of the values for **Motion In 1** and **Motion In 2** at the **Motion Out** output may be delayed by the interpolation. The effective delays for **Motion In 1** and **Motion In 2** are shown in the report as the internal parameters **Motion In 1 delay** and **Motion In 2 delay.**

To execute a position comparison with the interpolation mode active, there must be sufficient position values with regular refresh intervals available at both the **Motion in 1** and **Motion in 2** inputs.

Error detection time

The error detection time refers to the time it takes until an error which occurs at the function block inputs is indicated at the following outputs:

- Motion Out (reliability bits of the relative or absolute position)
- Monitoring status

Among other things, the error detection time depends on whether the connected encoders are evaluated with or without interpolation. When interpolation is activated, the fault detection time increases until an error is indicated at the **Motion Out** (reliability bit) and **Status** outputs.

WARNING

Extended error detection time

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

► Take into account the extended error detection time.

Interpolation activated: Fault detection time is the larger value of ...

a) Response time for signal path at **Motion In 1** input + **Motion In 1 delay** parameter + refresh interval of **Motion In 2** input

b) Response time for signal path at **Motion In 2** input + **Motion In 2 delay** parameter + refresh interval of **Motion In 1** input

NOTE

- The refresh interval is the longest possible time until the **Update status** returns to 1 (valid) while the **Absolute position status**/**Relative position status** remains 1 (valid) without changing. In SSI encoders, this is always the **Max. data reception interval** parameter; in A/B incremental encoders and Sin/Cos encoders, it is always 4 ms.
- The refresh interval of the other input is relevant (**Motion In 1**/**Motion In 2**).

Interpolation deactivated: Fault detection time is the larger value of...

a) Response time of signal path for **Motion In 1** input

b) Response time of signal path for **Motion In 2** input

NOTE

- The refresh interval is not relevant in this case.
- **Motion In 1 delay** and **Motion In 2 delay** are always 0 in this case.

Response time for Motion Out

WARNING

Extended response time for Motion Out

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

► Take into account the extended response time.

If the **Motion In 1 delay** or **Motion In 2 delay** is not equal to 0, this delay must be taken into account in the calculation of the response time of the signal path via **Motion Out.**

Generally, only the **Motion in 1 delay** will be relevant because, upon successful completion of the position comparison, the value of **Motion in 1** (which may be a delayed value) is output to **Motion out.**

If only **Motion Out** is used for the further evaluation of **Motion In 1** and **Motion In 2**, the error detection time also determines the response time. In such cases, you must bear in mind that, in the event of an error in the path for **Motion in 1**, an error response will only be triggered at the **Motion out** (reliability bits) or **Status** output if the position comparison fails. During this time, the system will continue to evaluate based on the false values from **Motion In 1** and **Motion In 2** will not be taken into account.

Reset

If a position comparison fails, the **Status** output switches to 0 and the corresponding reliability bits in **Motion Out** switch to 0 (unreliable). This status remains unchanged for at least the duration of 1 s (**Error recovery time**) even if the position comparison is positive again before this.

The **Error recovery time** is intended to ensure that detected faults can also be recognized by slower evaluations, e.g., using the **custom MOC status bits** of the FX3-MOC1 via a gateway (data set 3).

The time can be interrupted using the **Reset** input. A rising signal edge (0–1) at the **Reset** input resets the monitoring functions of the function block. For this to take place, the **Reset possible** output must be set to 1.

The **Reset possible** output is 1 if the following conditions are met:

1. The relative positions at the **Motion in 1** and **Motion in 2** inputs are valid.
2. **And**
   a) The absolute positions at the **Motion in 1** and **Motion in 2** inputs are valid.
   Or
   b) The **comparison mode** = **relative position only.**

Or

c) The calculated **Relative position deviation** is less than or equal to the selected **Position tolerance**.

After resetting:

- The **Error Recovery Time** is fulfilled or expired.
- The internal values for the relative position comparison are reinitialized, i.e., they are both set to the current relative position difference value. Consequently, the calculated **Relative position deviation** is zero and the relative position comparison is automatically positive.

The **Error recovery time** is also interrupted if either of the inputs **Motion in 1** or **Motion in 2** becomes invalid.

As a reset can only take place (the **Reset possible** output is 1) if it is not possible to perform an absolute position comparison, the **Status** output switches back to 1 without delay in the event of a reset.

Inhibit error indication

The **Inhibit error indication** input can be used to inhibit a diagnostics history entry if a position cross check fails. This may be useful in certain operating situations where faults or detection gaps are expected in order to minimize the impact of the fault.

As long as the **Inhibit error indication** input is set to 1, an error will not result in a diagnostics history entry. If the **Inhibit error indication** input switches to 0 while an error is still present, a corresponding error message is subsequently entered in the diagnostics history.

The error response, particularly the change of the status bits in the **Motion out** output, is not affected by the **Repress error signal** input.

11.10.5 Speed cross check V2

Function block diagram

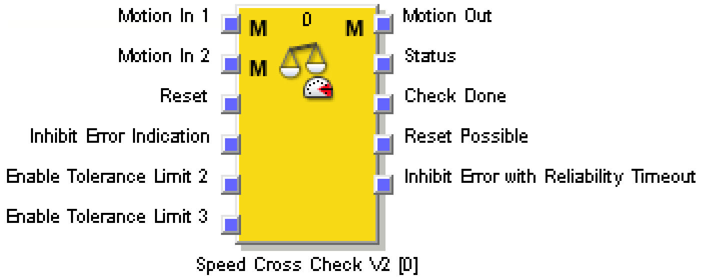


Figure 260: Inputs and outputs of the speed cross check V2 function block

General description

The **Speed cross check V2** function block compares speed values from two different signal sources. The checks performed are used to achieve a higher level of safety, particularly when working with non-safe encoders.

Deviations in both measured values may occur continuously or temporarily due to phenomena such as slip, abrasion, or mechanical coupling behavior. Consequently, this function block offers various parameters that can be used to tolerate deviations of this kind. In this way, unintended shutdowns can be avoided and machine availability can be ensured.

The following factors can be taken into account during evaluation:

- Permanently tolerated absolute speed difference or permanently tolerated speed ratio (relative speed difference in %), e.g. caused by different levels of component wear)
- Temporarily increased tolerance limits for the speed ratio, e.g. due to automation process requirements such as cornering by an AGV
- Signs of the speed values when calculating the speed difference

Fault detection

WARNING
Incorrect configuration

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

► As part of the safety assessment, consider the parameters used to configure the **Speed cross check V2** function block.

In accordance with generally recognized testing principles, test authorities usually stipulate that the application must ensure the monitored unit performs a movement at least once within the space of 24 hours. This movement must generate a signal change on the encoder system so that the speed cross check function can use it as a basis for detecting the relevant faults.

Main differences compared with Speed cross check V1 function module (FX3-MOCO V1.xx)

- The **Motion In 1** and **Motion In 2** inputs and the **Motion Out** output are of the type Motion V2.
- If an impermissible speed ratio is detected, the **Motion Out** output is no longer set to invalid. Instead, the bits **Speed reliability** and **Relative position reliability** in the **Motion Out** output are influenced. This allows the stop ramp to be monitored with the **Safe stop V2** function block even if one of the two sources for **Motion In 1** or **Motion In 2** supplies an invalid signal.
- The speed values can be interpolated if necessary. For this purpose, the values for **Motion In 1** or **Motion In 2** are internally delayed for processing and output at the **Motion Out** output. This makes it possible to reduce undesirable effects of different detection times, refresh intervals or signal propagation delays of the sources. The parameters **Signal propagation delay Motion In 2**, **Speed cross check on value change** and **Max. evaluation pause** are omitted.
- The output status is set to 1 = OK (no error detected or status unknown) if the speed becomes invalid at the **Motion In 1** or **Motion In 2** input because no evaluation is possible in this case and the error is located further forward in the signal path.
- The relative position for **Motion Out** is calculated based on the speed at **Motion Out** instead of the relative position of **Motion In 1** and **Motion In 2**.
- The new output **Check done** has been added.
- A manual reset no longer requires the mean speed to be approximately zero for a period of approx 1 s.
- The **Automated reset dependent on the absolute speed** function is omitted.
- The **Inhibit error reaction** input was replaced by the **Inhibit error indication** input. With this input, only addition of messages to the diagnostics history is suppressed, while the error reaction is no longer influenced.
- A new function for encoder reliability monitoring was added.

Inputs of the function block

Table 163: Inputs of the function block Speed cross check V2

Input	Description	Possible values
Motion In 1	For connecting two encoders.	Data of type Motion V2
Motion In 2		
Reset	Optional input for error reset by an external signal	- Inactive - Active
Inhibit error indication	Optional input, allows entry in the diagnostics history to be suppressed if a speed cross check fails	0 = no suppression 1 = error indication inhibited
Enable tolerance limit 2	Optional inputs that can be used to select increased tolerances for the speed cross check if required	- Inactive - Active
Enable tolerance limit 3		

Outputs of the function block

Table 164: Outputs of the function block Speed cross check V2

Output	Description	Possible values
Motion Out	Output of the checked Motion V2 data for use in another function block, e.g. **Speed monitoring V2**. Output of the values may take place after a delay depending on the configuration of the function block.	Depending on the **Speed output mode** parameter. The bits for validity and reliability are set depending on the results of the speed cross check.
Status	Indicates whether a speed cross check has failed. The initial status when the Flexi Soft system transitions to the Run status is 1. The output switches to 0 if the determined deviation exceeds the selected speed tolerance in a speed cross check. The output switches back to 1 when the determined deviation is again lower than or equal to the selected tolerance, but only after the **Error recovery time** of 1 s has elapsed. If the speed at the **Motion In 1** input or the **Motion In 2** input becomes invalid, then the **Status** output immediately switches to 1, as no evaluation can be performed in this state and the error is further ahead in the signal path.	0 = Error detected 1 = OK (no error detected or status unknown)
Check done	The output switches to 1 when the speed cross check is performed for the first time. The output switches back to 0 when the speed at the **Motion In 1** input or the **Motion In 2** input is invalid, meaning that a speed cross check cannot be carried out.	0 = speed cross check not performed 1 = speed cross check performed
Reset possible	Indicate whether a reset is possible by the **Reset** input	0 = reset not possible 1 = reset possible
Inhibit error with time monitoring	The output switches to 0 if the **Max. time without encoder reliability monitoring** is 0. The output otherwise corresponds to the **Inhibit error indication** input. This output can be connected to the **Inhibit motion bits reaction** input of a downstream **Safe stop V2** function block to suppress the error reaction there only for as long as the **Max. time without encoder reliability monitoring** is not exceeded.	0 = **Inhibit motion bits reaction** input is 0 or **Max. time without encoder reliability monitoring** has been exceeded 1 = **Inhibit motion bits reaction** input is 1 and **Max. time without encoder reliability monitoring** has been exceeded

Function block parameters

Table 165: Parameters of the function block Speed cross check V2

Parameter	Description	Possible values
Speed cross check mode		
Cross check mode	Specifies whether a tolerance speed is to be calculated and whether the sign should be taken into account for the calculation.	• No speed cross check • With sign • Without sign
Interpolation mode	Activates interpolation for the speed cross check	• Inactive • Active
Motion In 1 delay	Displays the internal delay for **Motion In 1** and **Motion In 2** which is active for internal evaluation and for producing the output at the **Motion Out** output	0 … 4 ms
Motion In 2 delay		
Limits for speed cross check		
Absolute tolerance limit for speed difference	Permanently permissible absolute speed difference between **Motion In 1** and **Motion In 2**. Speed differences that exceed this limit are taken fully into account.	**0 … 32,767 digits =** • 0 … 16,383 rpm • 0 … 32,767 mm/s
Speed ratio Tolerance limit 1	Permanent tolerance for the speed ratio (permissible relative speed difference in %) between **Motion In 1** and **Motion In 2** based on the higher of the two values	0 … 100%
Speed ratio Tolerance limit 2	Conditional increased tolerance for the speed ratio (permissible relative speed difference in %) between **Motion In 1** and **Motion In 2** based on the higher of the two values. • Enable via **Enable tolerance limit 2** and **Enable tolerance limit 3** input • Optional time limitation	0 … 100%
Speed ratio Tolerance limit 3		
Max. time for tolerance limit 2	Maximum length of time for which **Speed ratio tolerance limit 1** may be exceeded while tolerance limit 2 is valid	0 = infinite 4 … 60,000 ms in 4 ms increments
Max. time for tolerance limit 3	Maximum length of time for which **Speed ratio tolerance limit 2** may be exceeded while tolerance limit 3 is valid	0 = infinite 4 … 60,000 ms in 4 ms increments
Speed output mode		
Speed output mode	Mode for calculation of the speed output at the **Motion Out** output	• Speed of Motion In 1 • Higher speed of Motion In 1 or Motion In 2 • Mean speed of Motion In 1 and Motion In 2
Reset		
Reset input	Activates the optional Reset input which permits errors to be reset by means of an external signal	• Inactive • Active
Encoder reliability monitoring		
Encoder reliability monitoring	Activates a check to establish whether the signals of the encoders were reliable (e.g. damage during standstill)	• Inactive • Active
Max. time without encoder reliability monitoring	Maximum permitted time without exceeding the parameterized speed threshold	• 1 … 168 h
Speed threshold	Minimum speed for detection of encoder activity for encoder reliability monitoring	• 1 … 32,767 mm/s • 0 = infinite

Speed cross check mode

The **Speed cross check mode** determines whether a speed cross check is performed and whether the sign is taken into account in this case when the speed values are cross checked.

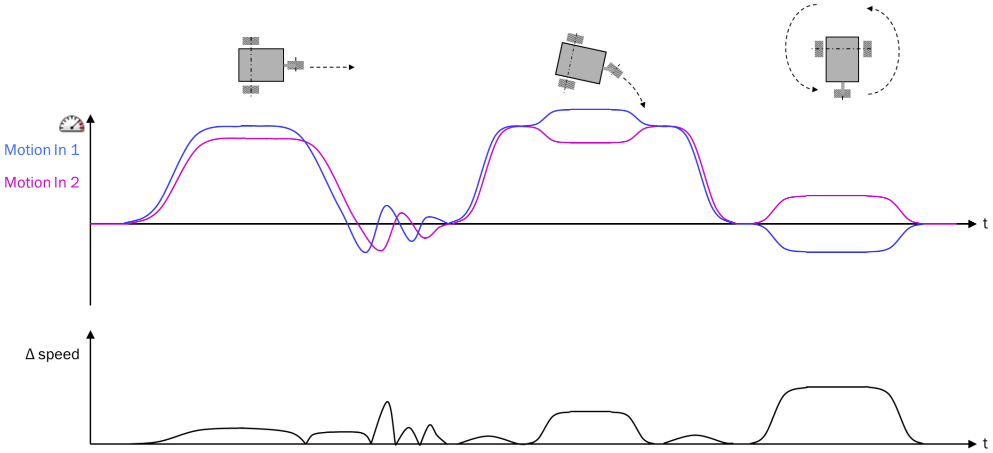


Figure 261: Speed cross check mode: Speed difference calculated with sign

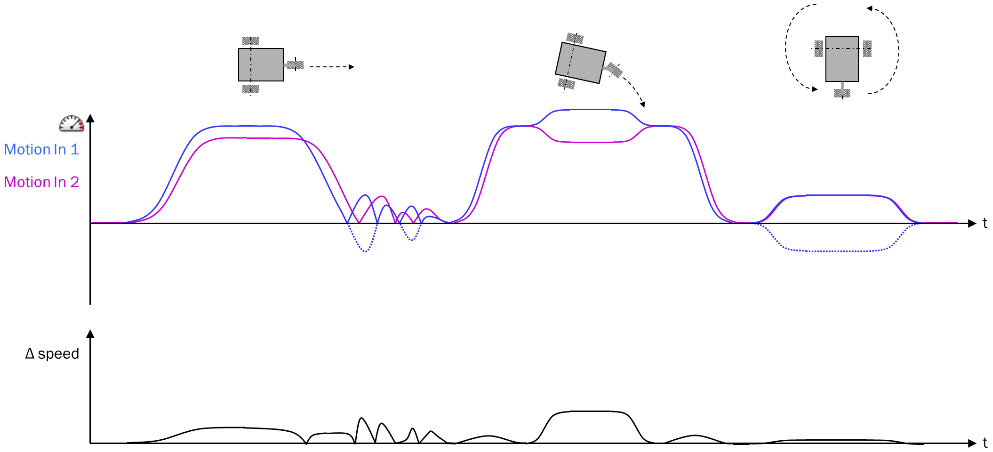


Figure 262: Speed cross check mode: Speed difference calculated without sign

Interpolation mode

Depending on the encoders used, the detection time, the refresh interval, and the signal propagation delay may differ at the two inputs. This point is of particular relevance when different encoder types are used. For example, due to the transmission time for SSI telegrams, an SSI encoder has a higher latency than an A/B incremental encoder. In particular, an SSI encoder also may have different detection times in listener mode.

These influences can be minimized with the Interpolation option:

- Influences due to different signal propagation times are minimized depending on the configured encoder type by internally delaying the faster path each case for the check.
- These variations in detection time are compensated for by the system retrospectively calculating an interpolated speed value for each encoder in turn, based on the two most recent speed values received; this calculated value is then compared with the last speed value of the other encoder. For this to take place, the system must have previously recorded speed values that permit interpolation in combination with the current speed value. The **Check done** output indicates when this condition is met for the first time following the Flexi Soft system's transition to run status.

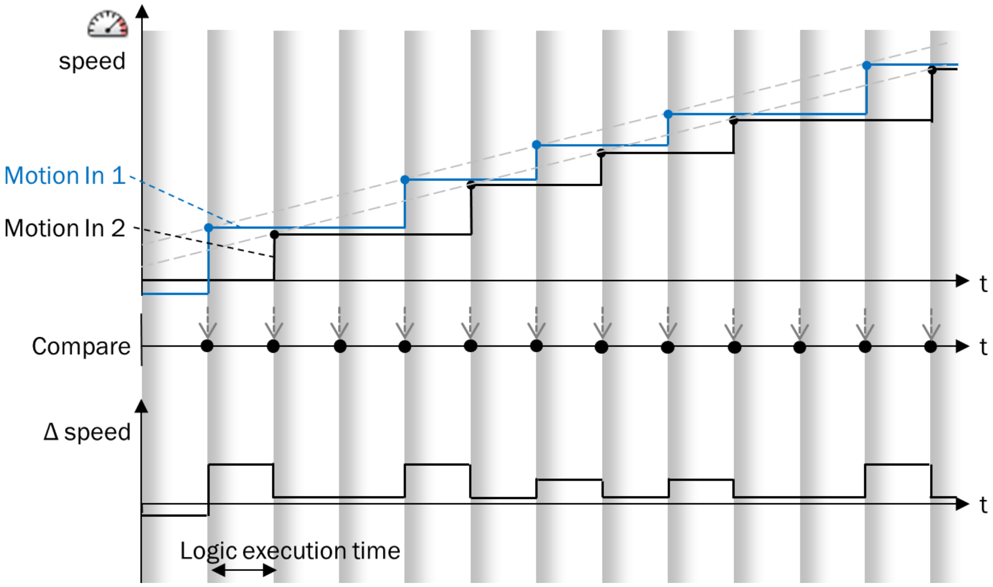


Figure 263: Speed cross check without interpolation

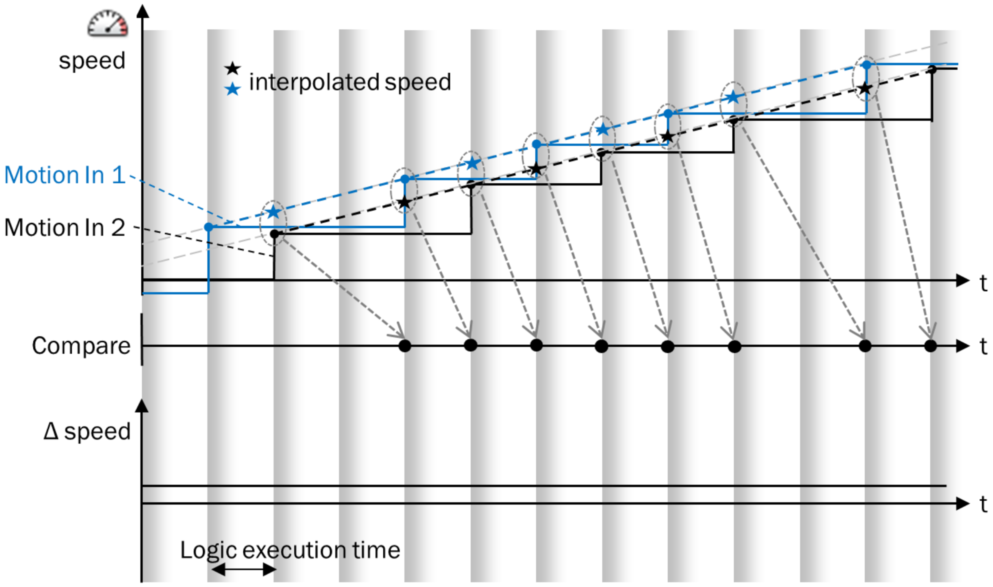


Figure 264: Speed cross check with interpolation

Output of the values for **Motion In 1** and **Motion In 2** at the **Motion Out** output may be delayed by the interpolation. The effective delays for **Motion In 1** and **Motion In 2** are shown in the report as the internal parameters **Motion In 1 delay** and **Motion In 2 delay.**

To execute a speed cross check with the interpolation mode active, there must be sufficient speed values with regular refresh intervals available at both the **Motion In 1** and **Motion In 2** inputs.

Error detection time

The error detection time refers to the time that is required until a fault at the inputs of the function block is displayed at the following outputs:

- **Motion Out** (bits for reliability of speed and relative position)
- **Status**

Among other things, the error detection time depends on whether the connected encoders are evaluated with or without interpolation. When interpolation is activated, the fault detection time increases until an error is indicated at the **Motion Out** (reliability bit) and **Status** outputs.

WARNING
Extended error detection time

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

► Take into account the extended error detection time.

Interpolation activated: Fault detection time is the larger value of ...

a) Response time for signal path at **Motion In 1** input + **Motion In 1 delay** parameter + refresh interval of **Motion In 2** input
b) Response time for signal path at **Motion In 2** input + **Motion In 2 delay** parameter + refresh interval of **Motion In 1** input

NOTE

- The refresh interval is the longest possible time until the **Update status** returns to 1 (valid) while the **Absolute position status**/**Relative position status** remains 1 (valid) without changing. In SSI encoders, this is always the **Max. data reception interval** parameter; in A/B incremental encoders and Sin/Cos encoders, it is always 4 ms.
- The refresh interval of the other input is relevant (**Motion In 1**/**Motion In 2**).

Interpolation deactivated: Fault detection time is the larger value of...

a) Response time of signal path for **Motion In 1** input
b) Response time of signal path for **Motion In 2** input

NOTE

- The refresh interval is not relevant in this case.
- **Motion In 1 delay** and **Motion In 2 delay** are always 0 in this case.

Response time for Motion Out

WARNING
Extended response time for Motion Out

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

► Take into account the extended response time.

If the **Motion In 1 delay** or **Motion In 2 delay** is not equal to 0, this delay must be taken into account in the calculation of the response time of the signal path via **Motion Out.**

- If the **Speed output mode** parameter is set to **Speed of Motion In 1**, the response time is calculated as follows:
  Response time for signal path at **Motion In 1** input + **Motion In 1 delay** parameter
- If the **Speed output mode** parameter is set to **Higher speed of Motion In 1 or Motion In 2** or to **Mean speed of Motion In 1 and Motion In 2**, the response time will correspond to the greater of the two following values:
  - Response time for signal path at **Motion In 1** input + **Motion In 1 delay** parameter
  - Response time for signal path at **Motion In 2** input + **Motion In 2 delay** parameter

If only **Motion Out** is used for the further evaluation of **Motion In 1** and **Motion In 2**, the error detection time also determines the response time. In such cases, you must bear in mind that, in the event of an error in the path for **Motion In 1**, an error response will only be triggered at the **Motion Out** (reliability bits) or **Status** outputs if the speed cross check fails. During this time, the system will continue to evaluate based on the false values from **Motion In 1** and **Motion In 2** will not be taken into account.

Permanently tolerated speed difference

The **Absolute tolerance limit for speed difference** parameter can be used to define the permissible absolute speed difference with reference to the higher of the speed values from **Motion In 1** and **Motion In 2**. Speed differences that are less than the **Absolute tolerance limit for speed difference** are evaluated as 0, i.e. are ignored. The purpose of this is to avoid high speed ratios at low speeds. Otherwise, a low absolute speed difference could lead to a high speed ratio because the speed being referenced is also low. Speed differences that exceed this tolerance limit are taken fully into account.

Permanently tolerated speed ratio

The **Speed ratio tolerance limit 1** parameter can be used to define the permissible speed ratio with reference to the higher of the speed values from **Motion In 1** and **Motion In 2**. The higher of the two values is taken to be 100% and the defined speed ratio is limited to 100%.

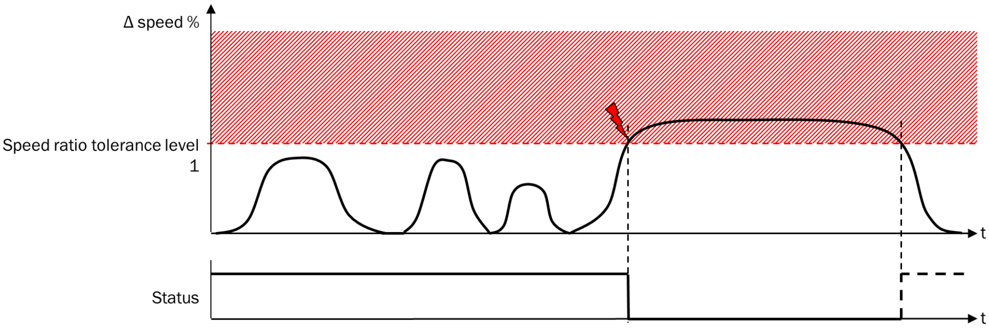


Figure 265: Permanently tolerated speed ratio

When the Flexi Soft system transitions to the Run status, the **Status** output is 1. As long as no errors occur, the value of the output does not change.

The **Status** output switches to 0 when the permissible speed ratio (relative speed difference in %) defined by the parameter **Speed ratio tolerance limit 1** is exceeded. This happens independently of the status of the **Inhibit error indication** input.

If the speed at the **Motion In 1** input or the **Motion In 2** input becomes invalid, then the **Status** output immediately switches to 1, as no evaluation can be performed in this state and the error is further ahead in the signal path.

Increased tolerance limit for the speed ratio

It is possible to increase the tolerance limit for the permissible speed ratio. The **Speed cross check V2** function block supports two additional tolerance limits with a conditional increase. Each of these can be configured with its own maximum time period.

The **Enable tolerance limit 2** and **Enable tolerance limit 3** inputs must be activated in the configuration dialog of the function block. **Speed ratio tolerance limit 2** and **Speed ratio tolerance limit 3** are available only if these inputs are activated.

When the **Enable tolerance limit 2** input is set to 1, it is permitted to exceed the value of **Speed ratio tolerance limit 1**. In this case, the increased **Speed ratio tolerance limit 2** is active. The duration for the limit being exceeded can be limited by the **Max. time Tolerance limit 2** parameter. A value of 0 ms here means infinite, i.e. no time limit. If the **Speed ratio tolerance limit 1** is exceeded for longer than the configured **Max. time Tolerance limit 2**, the **Status** output switches to 0.

The same applies to the third possible limitation: If the **Enable tolerance limit 3** input is set to 1, it is permitted to exceed the value of **Speed ratio tolerance limit 2** and the increased **Speed ratio tolerance limit 3** is active. The duration for the limit being exceeded can be limited by the **Max. time Tolerance limit 3** parameter. A value of 0 ms here means infinite, i.e. no time limit. If the **Speed ratio tolerance limit 2** is exceeded for longer than the configured **Max. time Tolerance limit 3**, the **Status** output switches to 0.

Speed ratio tolerance limit 3 is the highest increased tolerance limit for the speed ratio and must never be exceeded.

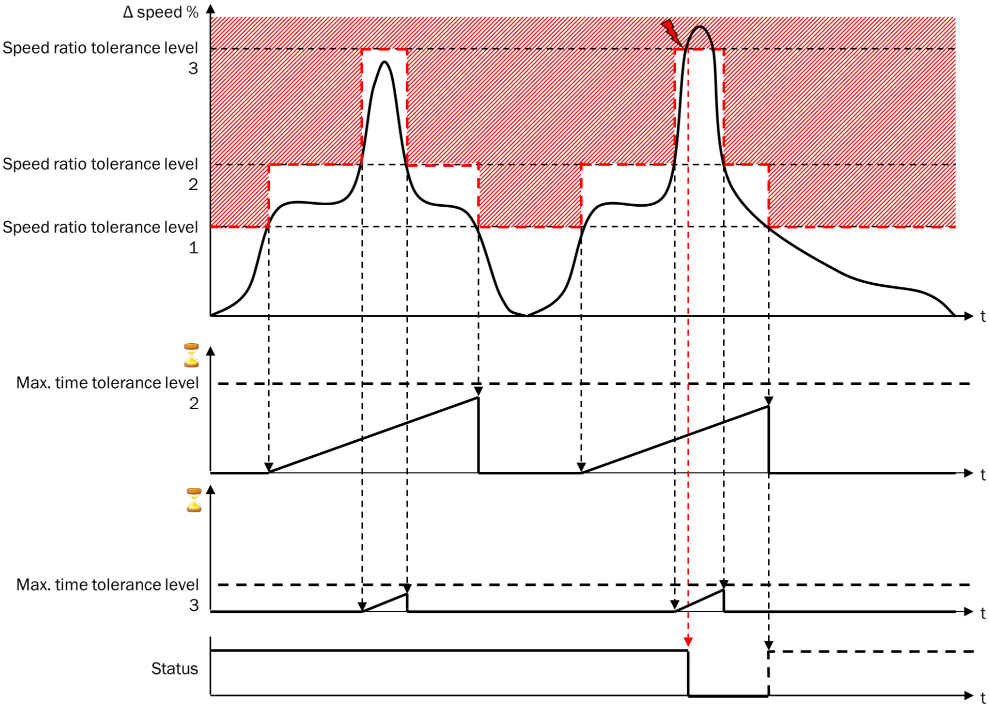

Δ speed %
Speed ratio tolerance level 3
Speed ratio tolerance level 2
Speed ratio tolerance level 1
t
Max. time tolerance level 2
t
Max. time tolerance level 3
t
Status
t

Figure 266: Conditional increased tolerance limit for the permitted speed ratio with exceeded tolerance limit

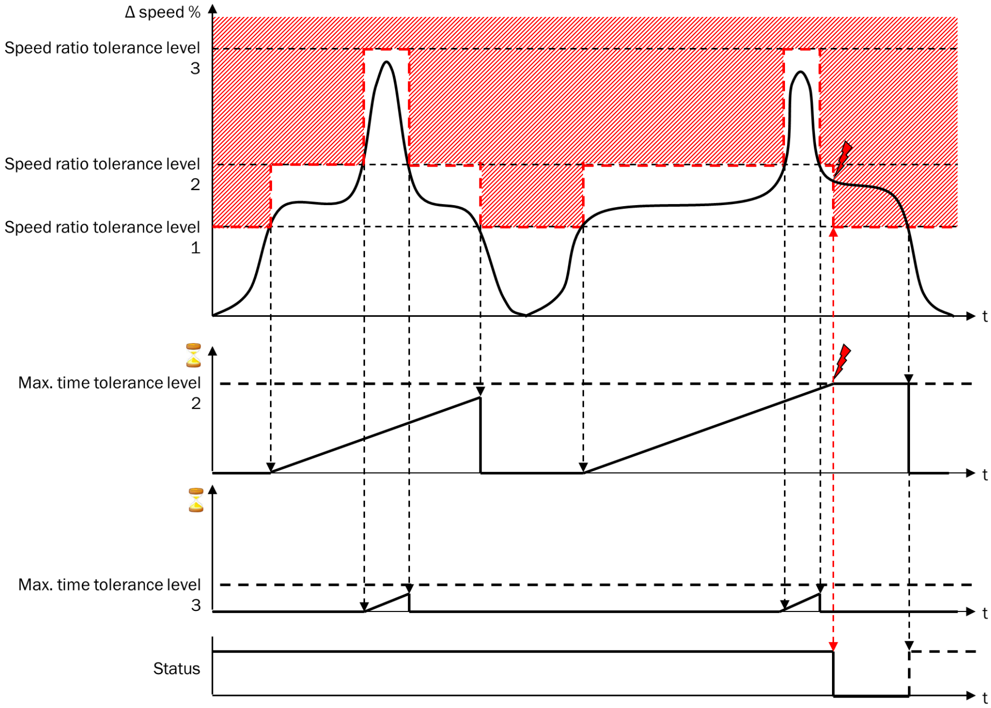


Figure 267: Conditional increased tolerance limit for the permissible speed ratio with exceeded time limit

For example, in the case of an AGV, the increased **Speed ratio tolerance limit 2** allows the difference in speed between two wheels to be tolerated during cornering. This means that the vehicle control system can enable the **Enable tolerance limit 2** input when the vehicle is turning a corner. The increased **Speed ratio tolerance limit 3** can be used to tolerate very short speed differences, e.g. when a wheel spins briefly.

Speed output mode for output at the Motion Out output

The values of the **Motion Out** output are formed based on the **Motion In 1** and **Motion In 2** inputs and are dependent on the **Speed output mode** parameter in some cases.

NOTE
For the purposes of internal and evaluation and production of the output at **Motion Out**, the two inputs **Motion In 1** and **Motion In 2** are delayed internally corresponding to the parameters **Motion In 1 delay** and **Motion In 2 delay**. If the designation **Motion In 1** input or **Motion In 2** input is used in the description, this refers to the possibly delayed value.

The following three cases are distinguished for output at the Motion Out output:
a) Both **Motion In 1 speed status** and **Motion In 2 speed status** are invalid (0).
b) Either **Motion In 1 speed status** or **Motion In 2 speed status** is invalid (0).
c) Both **Motion In 1 speed status** and **Motion In 2 speed status** are valid (1).

a. If both **Motion In 1 speed status** and **Motion In 2 speed status** are invalid (0), all values at the **Motion Out** output are set to 0.

b. If either Motion In 1 speed status or Motion In 2 speed status is valid (0), the valid speed value is output in each case independently of the Speed output mode parameter. The following applies in this case:

- **Motion Out speed status** = 1 (valid)
- **Motion Out speed reliability** = 0 (unreliable)
- **Motion Out speed** = either **Motion In 1 speed** or **Motion In 2 speed**, depending on the input at which the bit for **Speed status** is set to 1 (valid).

ℹ **NOTE**

No cross check can be performed if only one signal is valid. For this reason, **Motion Out speed reliability** and **Motion Out relative position reliability** are set to 0 (unreliable) in this case.

Even if the information is deemed to be unreliable because it is only from one source, the Safe Stop V2 function block still evaluates the speed when monitoring the stop ramp, for example, in order to detect a situation where the ramp is exceeded and to switch off the drive as early as possible in this error case.

The Update status at the Motion Out output is 1 (current) in this case if one of the following cases applies:

- The **Update status** at at least one of the inputs **Motion In 1** or **Motion In 2** is 1 (current).
- Either **Motion In 1 speed status** or **Motion In 2 speed status** has just changed to invalid (0) while the other input is set to 1 (valid). In this case, switchover to the speed value of the still valid input takes place for output. Since the speed value at the **Motion Out** output can change as a result, the **Update status** at the **Motion Out** output is set to 1 (current) in the event of this switchover.

c. If both **Motion In 1 speed status** and **Motion In 2 speed status** are valid (1), the values at the **Motion Out** output are formed as follows:

*Table 166: Formation of Motion V2 data at the **Motion Out** output when both **Motion In 1 speed status** and **Motion In 2 speed status** are valid (1)*

Element	Description
Speed	Depending on **Speed output mode** parameter **The following settings are possible:** • Speed of **Motion In 1** • Higher speed of **Motion In 1** or **Motion In 2** • Mean speed of **Motion In 1** and **Motion In 2**
Speed status	Always 1 (valid)
Speed reliability	**Depending on Cross check mode parameter:** • If the **Cross check mode** is set to **With sign** or **Without sign**, then this bit corresponds to the **Status** output. • If the **Cross check mode** is set to **No cross check**, then **Speed reliability** at the **Motion Out** output is 1 (reliable), if **Speed reliability** is also 1 (reliable) at both the inputs **Motion In 1** and **Motion In 2**.
Relative position	Depending on **Speed output mode** parameter (see below)
Relative position status	1 (valid) if both **Motion In 1 relative position status** and **Motion In 2 relative position status** are 1 (valid)
Relative position reliability	If the **Relative position status** is 0 (invalid) at at least one of the inputs **Motion In 1** or **Motion In 2** 0 (invalid), then this bit is 0 (unreliable). **If both Motion In 1 relative position status and Motion In 2 relative position status are 1 (valid), then the bit is dependent on the Cross check mode parameter:** • If the **Cross check mode** is set to **With sign** or **Without sign**, then this bit corresponds to the **Status** output. • If the **Cross check mode** is set to **No cross check**, then **Relative position reliability** at the **Motion Out** output is 1 (reliable), if **Relative position reliability** is also 1 (reliable) at both the inputs **Motion In 1** and **Motion In 2**.
Absolute position	Always 0
Absolute position status	Always 0 (invalid)
Absolute position reliability	Always 0 (unreliable)
Update status	Depending on **Speed output mode** parameter (see below)

Speed of Motion In 1

Speed at **Motion Out** output:

With this setting, the **Speed** value at the **Motion Out** output corresponds to the **Speed** value at the **Motion In 1** input.

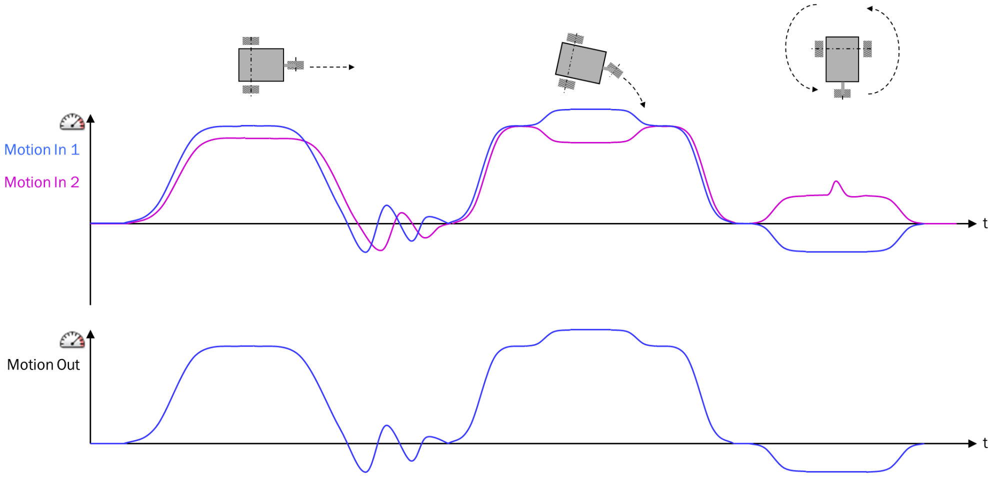


Figure 268: Speed output mode: Speed of Motion In 1

Speed of Motion In 1 is usually selected when there is a leading encoder with a higher resolution and also a second encoder that is used for plausibility checks. In this case, the values at the **Motion In 1** input are used for further evaluation.

Update status at **Motion Out** output:

With this setting, the **Update status** at the **Motion Out** output corresponds to the **Update status** at the **Motion In 1** input.

Relative position at **Motion Out** output:

With this setting, the **Relative position** value at the **Motion Out** output corresponds to the **Relative position** value at the **Motion In 1** input.

Higher speed of Motion In 1 or Motion In 2

Speed at **Motion Out** output:

With this setting, the higher of the two speeds at the **Motion In 1** and **Motion In 2** inputs is output at the **Motion Out** output.

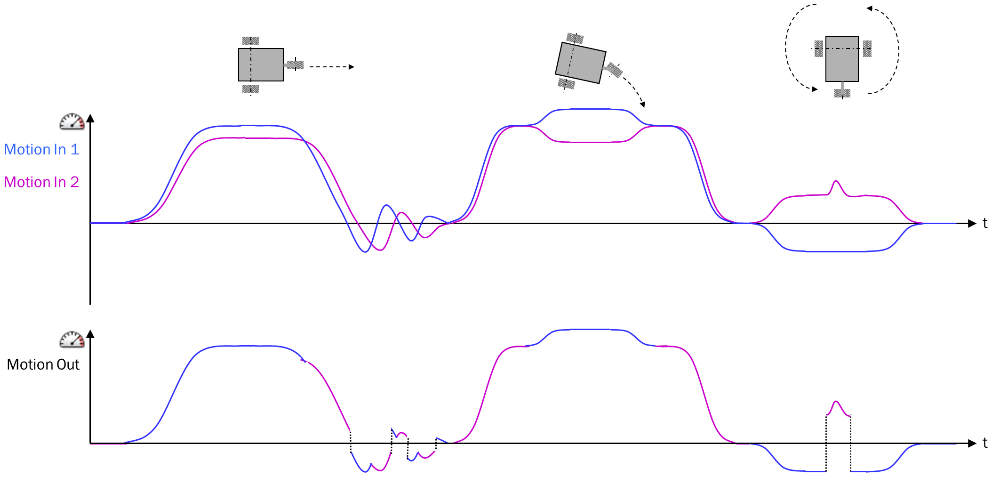


Figure 269: Speed output mode: Higher speed of Motion In 1 or Motion In 2

The **Higher speed of Motion In 1 or Motion In 2** setting should be selected if the higher value from two encoders is to be used for further evaluation, e.g. because the speed of the outer wheel of an AGV during cornering is of relevance (worst-case scenario).

Update status at **Motion Out** output:

With this setting, the **Update status** at the **Motion Out** output is 1 (current) if the **Update status** at at least one of the inputs **Motion In 1** or **Motion In 2** is 1 (current).

Relative position at **Motion Out** output:

If the parameter **Speed output mode** is set to **Higher speed of Motion In 1 or Motion In 2**, then the relative position for the **Motion Out** output is calculated based on the speed value for the **Motion Out** output. This means that the values for the relative position at the inputs **Motion In 1** and **Motion In 2** are not used for calculation of the relative position at the **Motion Out** output and only their respective validity status is evaluated. The **Relative position status** at the **Motion Out** output can become 1 (valid) only if **Relative position status** is also 1 (valid) at both the **Motion In 1** input and the **Motion In 2** input.

NOTE

- The resulting speed curve may be non-continuous if two opposing directions of movement are involved.
- If the **Speed output mode** parameter is set to **Higher speed of Motion In 1 or Motion In 2** or to **Mean speed of Motion In 1 and Motion In 2** and the value at the **Motion Out** output is used for standstill monitoring with standstill position tolerance (e.g. using the **Speed monitoring V2** function block), the standstill condition may be met even though the relative positions at the **Motion In 1** input and **Motion In 2** input would not meet the standstill condition when considered on their own. This case can occur if the two encoders are moving in opposite directions and the resulting average speed is considerably less than the speed of each individual encoder.

Mean speed of Motion In 1 and Motion In 2

Speed at **Motion Out** output:

With this setting, it is the average speed from the **Motion In 1** and **Motion In 2** inputs that is output at the **Motion Out** output together with its sign.

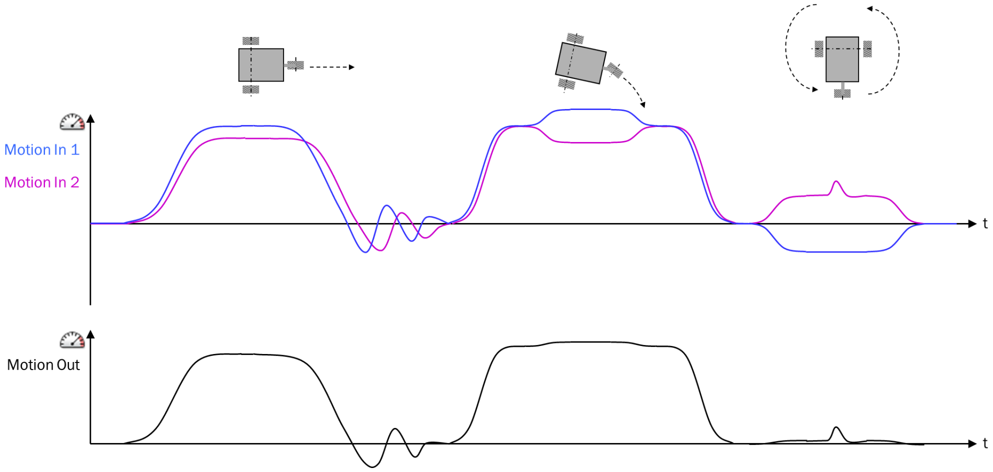


Figure 270: Speed output mode: Mean speed of Motion In 1 and Motion In 2

The **Mean speed of Motion In 1 and Motion In 2** setting is usually used when the average speed from two encoders is of relevance for further evaluation because, for example, it represents the speed at the central point of an AGV. With this setting, turning on the spot (when both wheels are traveling at the same speed in opposite directions) is evaluated as a standstill, for example.

Update status at **Motion Out** output:

With this setting, the **Update status** at the **Motion Out** output is 1 (current) if the **Update status** at at least one of the inputs **Motion In 1** or **Motion In 2** is 1 (current).

Relative position at **Motion Out** output:

If the parameter **Speed output mode** is set to **Mean speed of Motion In 1 and Motion In 2**, then the relative position for the **Motion Out** output is calculated based on the speed value for the **Motion Out** output. This means that the values for the relative position at the inputs **Motion In 1** and **Motion In 2** are not used for calculation of the relative position at the **Motion Out** output and only their respective validity status is evaluated. The **Relative position status** at the **Motion Out** output can become 1 (valid) only if **Relative position status** is also 1 (valid) at both the **Motion In 1** input and the **Motion In 2** input.

NOTE

- If the **Speed output mode** parameter is set to **Higher speed of Motion In 1 or Motion In 2** or to **Mean speed of Motion In 1 and Motion In 2** and the value at the **Motion Out** output is used for standstill monitoring with standstill position tolerance (e.g. using the **Speed monitoring V2** function block), the standstill condition may be met even though the relative positions at the **Motion In 1** input and **Motion In 2** input would not meet the standstill condition when considered on their own. This case can occur if the two encoders are moving in opposite directions and the resulting average speed is considerably less than the speed of each individual encoder.

Resetting the function block

If a speed cross check fails, the **Status** output switches to 0 (error detected) and the corresponding reliability bits in **Motion Out** switch to 0 (unreliable). This status remains unchanged for at least the duration of 1 s (**Error recovery time**) even if the speed cross check already produces a valid result again before this.

The **Error recovery time** serves to ensure that the speed cross check must first continuously supply a positive result for a minimum period before the associated reliability bits in **Motion Out** are set to 1 (reliable) again. This also permits error detection by slower evaluations, e.g. using the **customized MOC status bits** of the FX3-MOC1 via a gateway (data set 3).

The optional **Reset** input can be used to cancel the **Error recovery time.** A rising signal edge (0–1) at the **Reset** input resets the monitoring functions of the function block. For this to take place, the **Reset possible** output must be set to 1.

The **Error recovery time** is also canceled if either of the inputs **Motion In 1** or **Motion In 2** becomes invalid.

The function block can be reset in two ways:

- Manual reset: If the **Reset possible** output is 1, then an error can be reset by a rising signal edge at the optional **Reset** input. The **Reset possible** output is set to 1 if the speed cross check is activated, the values at the **Motion In 1** input and **Motion In 2** input are valid and the speed ratio (relative speed difference in %) is lower than the currently valid **Speed ratio tolerance limit x.**
- Automated reset: An error is reset if the speed cross check has continuously supplied a positive result for at least a period of 1 s (**Error recovery time**).

NOTE

If the optional **Reset** input is not activated, it is not possible to manually reset an error condition during operation.

Inhibit error indication

Using the **Inhibit error indication** input, it is possible to prevent an entry being made in the diagnostics history when a speed cross check fails (**Status** output = 0). This can be expedient in certain operating situations where faults or detection gaps are expected in order to minimize the impact of a fault in a targeted way.

If the **Inhibit error indication** input is 1, an error does not lead to an entry in the diagnostics history. If the **Inhibit error indication** switches to 0 while an error is still present, a corresponding error message is subsequently entered in the diagnostics history.

The **Inhibit error indication** input does not have any influence on the **Status** output or the reliability bits in the **Motion Out** output.

Encoder reliability monitoring

In accordance with generally recognized testing principles, test authorities usually stipulate that the application must ensure the monitored unit performs a movement at least once within the space of 24 hours. This movement must generate a signal change on the encoder system so that the speed cross check function can use it as a basis for detecting the relevant faults. The encoder reliability monitoring function allows monitoring of whether the required movement has taken place, i.e. whether the required minimum speed was reached during the required time interval.

Whether **encoder reliability monitoring** is required or not depends on the risk analysis.

The **Max. time without encoder reliability monitoring** can be set to a value of 1 hour to 168 hours (1 week) for different applications.

The required minimum speed for detection of a movement is set with the **Speed threshold** parameter. If the speed at the **Motion Out** output falls below the **Speed threshold**, the timer for the **Max. time without encoder reliability monitoring** starts to count down. The timer is reset again when the speed at the **Motion Out** output exceeds the **Speed threshold** again.

The timer is also reset from the Stop status to the Run status after every transition, i.e. after every time the supply voltage to the Flexi Soft main module is switched off and back on. As a result, the total time during which the speed at the **Motion Out** output lies below the speed threshold can be longer than the parameterized **maximum time without encoder reliability monitoring**.

⚠ WARNING
Unrecognized encoder malfunction

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.

► Select a suitable value for the **maximum time without encoder reliability monitoring.** When doing so, take into account that the timer is reset by switching the supply voltage to the Flexi Soft main module off and back on.

Or:

► Ensure via the application that the required movement is performed in the required time interval (without monitoring this via the encoder reliability monitoring function).

If the timer reaches 0 before the speed at the **Motion Out** output exceeds the **Speed threshold** again, the **Status** output switches to 0. The **Status** output is set to 1 again if the speed at the **Motion Out** output subsequently exceeds the **Speed threshold** again and no error occurs.

11.10.6 Speed monitor V2

Function block diagram

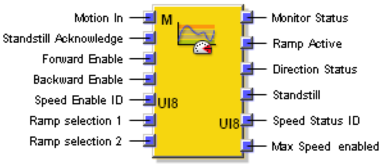


Figure 271: Inputs and outputs of the Speed monitoring V2 function block

General description

The **Speed monitoring V2** function block can be used for speed and direction monitoring in an application. It can essentially perform the following functions:

- Safe speed monitor (SSM)
- Safely limited speed (SLS)
- Safe direction (SDI)
- Safe operating stop (SOS)
- Monitoring of up to four different speed ramps during the transition from a monitored speed to a lower speed.

Speed monitor

- Maximum speed monitoring
- Monitoring the speed limits selected via the inputs **Speed enable ID** and, where applicable, **Ramp selection 1** and **Ramp selection 2**
- Direction monitoring, selected via the inputs **Enable forward** and **Enable backward**

The speed monitoring status is displayed at the **Monitoring status** output.

NOTE

Main differences compared with the Speed monitoring V1 function block (FX3-MOC0 V1.xx)

- Additional function **Maximum speed filter distance**
- The standstill speed can be deactivated so that only standstill position monitoring can be activated.
- Changed standstill position monitoring:
  - Evaluation of the optional **Standstill approval** input; this input also affects **Standstill position monitoring** rather than just **Standstill speed monitoring**.
  - The **Standstill position tolerance** is set when the required three detected positions are within a tolerance band of ± **Standstill position tolerance** (previously: ± 2× **Standstill position tolerance**).
- **Motion In** input for the extended Motion data type **Motion V2**.

Inputs of the function block

Table 167: Inputs of the Speed monitoring V2 function block

Input	Description	Signal value
Motion In	Data of type Motion V2 is expected, either directly from an encoder or from another function block, e.g., **Position Cross Check V1**.	Motion V2 data
Standstill acknowledge	Optional input, activates internal standstill detection	0 = standstill detection deactivated 1 = standstill detection active

Input	Description	Signal value
Enable forward and Enable backward	Optional inputs which enable the relevant direction of movement. If either of these inputs is not used, it means that the associated direction of movement is permanently enabled.	0 = direction not enabled 1 = direction enabled Input not used = Direction permanently enabled
Speed enable ID	Activates the permitted speed limit	0 ... 255
Ramp selection 1 and Ramp selection 2	Selection of up to four speed ramps with different gradients	0 or 1

NOTE

The **Speed monitoring V2** function block requires data with a reliability status of 1 at the **Motion In** input. The data must therefore be plausibility-checked, e.g., using a Sin/Cos encoder with analog voltage monitoring or through the use of two encoders connected to a **Position comparison** function block.

Outputs of the function block

Table 168: Outputs of the Speed monitor V2 function block

Output	Description	Signal value
Monitoring status	The **Monitoring status** output indicates the combined status of the various monitoring functions (AND connection). If the status is unknown; i.e., if the relevant data at the **Motion In** input is unreliable or invalid, the output switches to 1. The initial status when the Flexi Soft system transitions to the Run status is 1.	0 = error detected 1 = OK (no error detected or status unknown)
Ramp active	Indicates whether a speed ramp is active	0 = No ramp active 1 = speed ramp active
Direction status	Indicates the direction of movement. No change during standstill. If the status is unknown; i.e., if the speed at the **Motion In** input is unreliable or invalid, the output switches to 0. The initial status when the Flexi Soft system transitions to the Run status is 0.	0 = Forwards or status unknown 1 = Backwards
Standstill	Indicates whether the standstill condition is fulfilled (either standstill speed and/or standstill position, taking into account the filter and acceptance criteria such as the **Maximum speed filter distance** parameter, the **Standstill approval** input, and the **Standstill speed acceptance time** parameter). If the status is unknown; i.e., if the relevant data at the **Motion In** input is unreliable or invalid, the output switches to 0. The initial status when the Flexi Soft system transitions to the Run status is 0.	0 = No standstill or status unknown 1 = standstill
Speed status ID	Indicates which speed range the current speed at the **Motion in** input corresponds to. This is not affected by the speed limit currently active for monitoring.	0 ... 10 0 = speed invalid or unreliable 1 = standstill 2 ... 10 = speed range 2 ... 10
Max. speed enabled	Indicates whether the highest configured speed limit is active (selected via the **Speed enable ID**) input. This is not affected by the current speed at the **Motion in** input.	0 = Maximum speed not enabled 1 = Maximum speed enabled

Function block parameters

Table 169: Parameters of the Speed monitor V2 function block

Parameter	Description	Possible values
Standstill monitoring		

Parameter	Description	Possible values
Standstill speed monitoring	Activates the **Standstill speed monitor** function	• Deactivated • Active
Standstill speed	Defines which speed still counts as a standstill	**0 … 32,766 digit =** • 0 … 16,383 rpm • 0 … 32,766 mm/s
Standstill speed acceptance time	Defines the uninterrupted period of time for which the **Standstill speed** may not be exceeded	0 … 248 ms in 4 ms increments
Standstill position monitoring	Activates the **Standstill position monitoring** function	• Deactivated • Active
Standstill position tolerance	Defines which relative position change still counts as a standstill during standstill monitoring. While the **Standstill position tolerance** is not exceeded, the speed is not taken into account even if it is greater than the **Standstill speed.**	**0 … 500,000,000 digit =** • 0 … 16,666 rev. • 0 … 2,000,000 mm
Standstill acknowledge	Activates the optional **Standstill acknowledge** input	• Deactivated • Active
Speed ranges		
Max speed limit	Defines the maximum speed that is allowed to occur within the system	**1 … 32,767 digit =** • 0.5 … 16,383 rpm • 1 … 32,767 mm/s
Speed limit 1	Speed limit 1 always corresponds to the **standstill speed**	**0 … 32,766 digit =** • 0 … 16,383 rpm • 0 … 32,766 mm/s
Speed limit 2 … 9	Up to 9 speed limits, including standstill speed	
Maximum distance for speed filter	Defines the distance that the drive is allowed to travel despite exceeding the current speed limit before the excess speed causes the system to switch off.	**0 … 65,534 digits (position) =** • max. 2.18 rev. • max. 262 mm • 0 = Deactivated
Ramp for speed limit transitions		
Delay time until start of ramp	How long the function block expects there to be no response from the system; i.e., the length of time for which it does not expect a delay ramp	0 … 248 ms in 4 ms increments
Ramp configuration (ramp steepness, speed transitions 1 … 4)	Increment for the reduction in speed when changing from a higher active speed enable ID to a lower one, selected via the **Speed enable ID** input. You can define up to four different ramps. The **Speed reduction** and the **Duration of the speed reduction** are entered.	**0 … 2,147,418,112 digit =** • 0.5 … 16,383 (rpm)/ms • 1 … 65,535 (mm/s)/ms • 0 = No ramp
Optional inputs		
Speed enable ID	Activates the optional **Speed enable ID** input	• Disabled • Active
Enable forward	Activates the optional **Enable forward** input	• Disabled • Active
Backward enable	Activates the optional **Enable backward** input	• Disabled • Active

Safe speed monitor (SSM)

Up to 9 speed limits (including standstill speed) can be configured for up to 10 speed ranges. Speed limit 1 is always standstill speed.

The **Speed status ID** output indicates which speed range corresponds to the current speed at the **Motion in** input. This is not affected by the speed limit that is currently active. The data is output as a UI8 value. To connect this value to Boolean signals, use the **UI8 to Bool V1** function block.

*Table 170: **Speed status ID** output for 9 configured speed limits*

Speed at Motion In input	Meaning	Speed status ID
The speed is invalid or unreliable.	Invalid	0
Standstill condition [1] met: • The speed is lower than the standstill speed for at least the duration of the standstill speed acceptance time. **or** • The standstill position tolerance is determined and not exceeded.	Standstill	1
No standstill and speed > standstill speed Speed ≤ speed limit 2	Speed range 2	2
Speed > speed limit 2 Speed ≤ speed limit 3	Speed range 3	3
Speed > speed limit n–1 Speed ≤ speed limit n	Speed range n	n
Speed > speed limit 8 Speed ≤ speed limit 9	Speed range 9	9
Speed > speed limit 9	Speed range 10	10

[1] If neither standstill speed monitoring nor standstill position monitoring is activated, then the Speed status ID output never switches to 1.

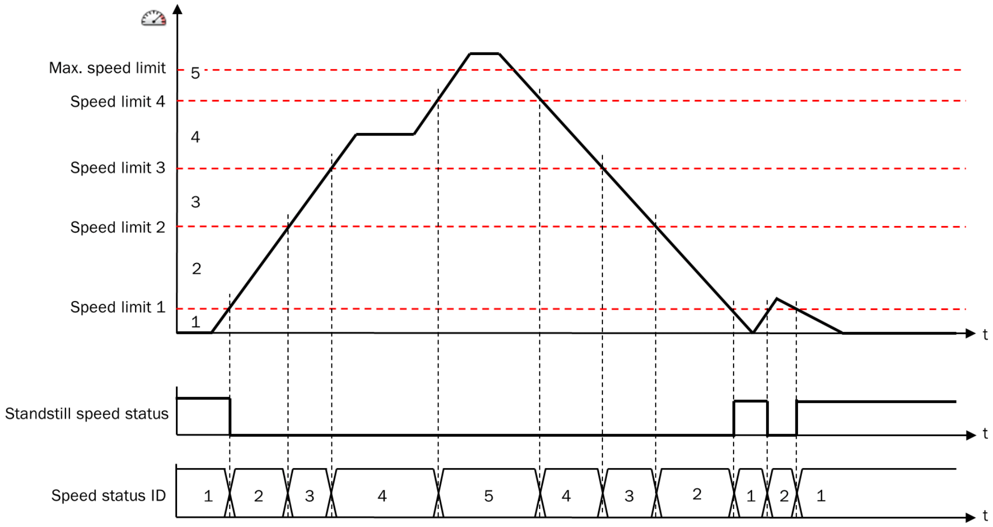


Figure 272: Sequence/timing diagram for the Speed status ID output (example involving four configured speed limits, i.e., five speed ranges)

NOTE

The configured value for the parameter **Max. speed** does not affect the speed status ID, i.e., the **Speed status ID** does not change even if the **max. speed** is exceeded.

Monitoring functions

The Monitoring status output is set to 1 if one of the following conditions is met:

- No error detected
- Status unknown, because the speed at the **Motion in** input is unreliable or invalid

When the Flexi Soft system transitions to the Run status, the **Monitoring status** output is 1.

The Monitoring status output switches to 0 whenever any of the following monitoring functions produces a result of 0:

- Maximum speed monitoring
- Monitoring of speed limits, selected via the **Speed enable ID** input
- Direction monitoring

The Monitoring status output switches back to 1, if one of the following conditions is met:

a) All associated monitoring processes are fulfilled.
b) The speed at the **Motion in** input is unreliable or invalid.

The **Monitoring status** output is usually connected to the **Safe stop 2A** input of the **Safe stop V2** function block. This means that an impermissible speed or direction will result in a stop.

Speed monitoring functions

The functions for monitoring the **Max. speed** and the **Speed limits** can be used to implement the safely-limited speed (SLS) function. For transitions from a higher to a lower speed limit, **Speed ramps** can be configured.

Maximum speed monitoring

Max. speed monitoring is always active. If the current speed is greater than the **Max. speed** that has been configured, the **Monitoring status** output switches to 0. When used in conjunction with a **Safe stop V2** function block, this allows the maximum travel/maximum time for a Safe stop to be determined reliably.

The **Max. speed enabled** output is set to 1 when the highest configured speed limit is activated via the **Speed enable ID** input. This output can be used as a reset condition at the **Reset** input of a subsequent **Safe stop V2** function block. If, for example, the **Max. speed** is enabled as long as a safety door remains closed, you can reset a stop ramp that has been triggered by closing the safety door.

Monitoring of the speed limits

The optional **Speed enable ID** input activates the permissible speed limit. The input will accept a UI8 value (0 … 255). To connect the input to Boolean signals, use the **Bool to UI8 V1** function block.

If the current speed at the **Motion in** input is greater than the active speed limit, then the **Monitor status** output is set to 0.

NOTE

- The values 0 and 1 at the **Speed enable ID** input activate standstill monitoring. If neither standstill speed monitoring nor standstill position monitoring is activated, then the **Monitoring status** output always remains at 0 (error), unless the speed at the **Motion in** input is unreliable or invalid.
- Any value above the number of configured speed limits activates the maximum permitted speed.

Speed ramps

The **Ramp speed transitions** parameters can be used to define up to four speed ramps. This allows the current speed limit to be reduced evenly from a higher to a lower speed limit in accordance with the configured increment instead of switching to the lower speed limit immediately. This happens regardless of the actual current speed, i.e., even if the actual speed is already below the new lower speed limit.

Up to four speed ramps with different increments can be defined. A ramp can be selected using the **Ramp selection 2** and **Ramp selection 1** inputs.

Table 171: Speed ramp selection

Input values		Selected ramp
Ramp selection 2	Ramp selection 1	
0	0	Ramp speed transition 1 (fastest ramp)
0	1	Ramp speed transition 2
1	0	Ramp speed transition 3
1	1	Ramp speed transition 4 (slowest ramp)

NOTE
Any change to the input values also affects any speed ramp that happens to be active when this change occurs.

The **Delay time until start of ramp** parameter specifies the delay time that must elapse before the speed ramp commences. This makes it possible, for example, to tolerate a delay in the drive response caused by communication and processing cycle issues.

While a speed ramp is active, the **Ramp active** output remains set to 1.

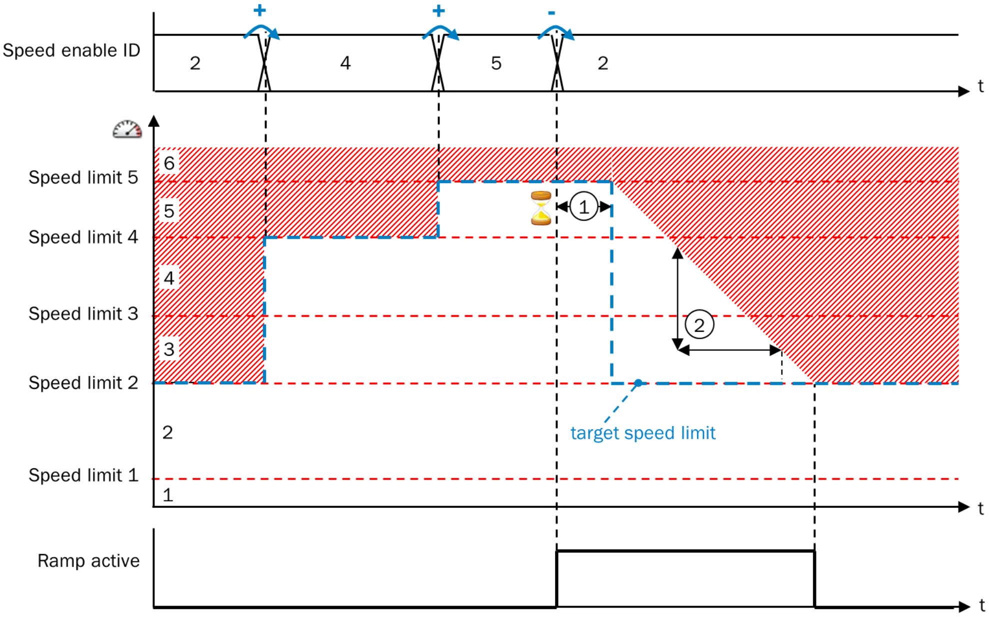


Figure 273: Example 1 for monitoring the speed limit

① Delay time until start of ramp

② Speed transition ramps

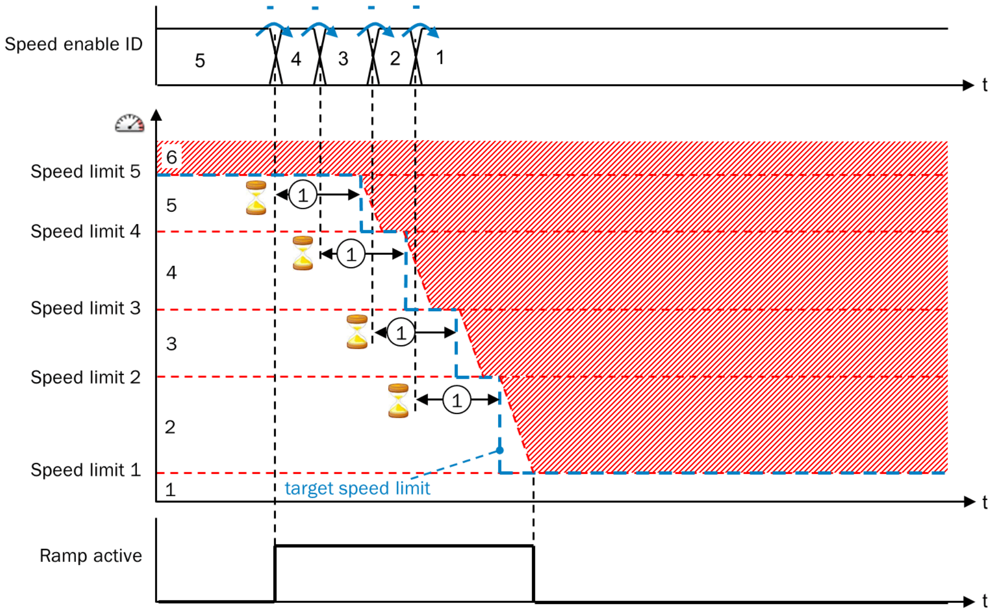


Figure 274: Example 2 for monitoring the speed limit

① Delay time until start of ramp

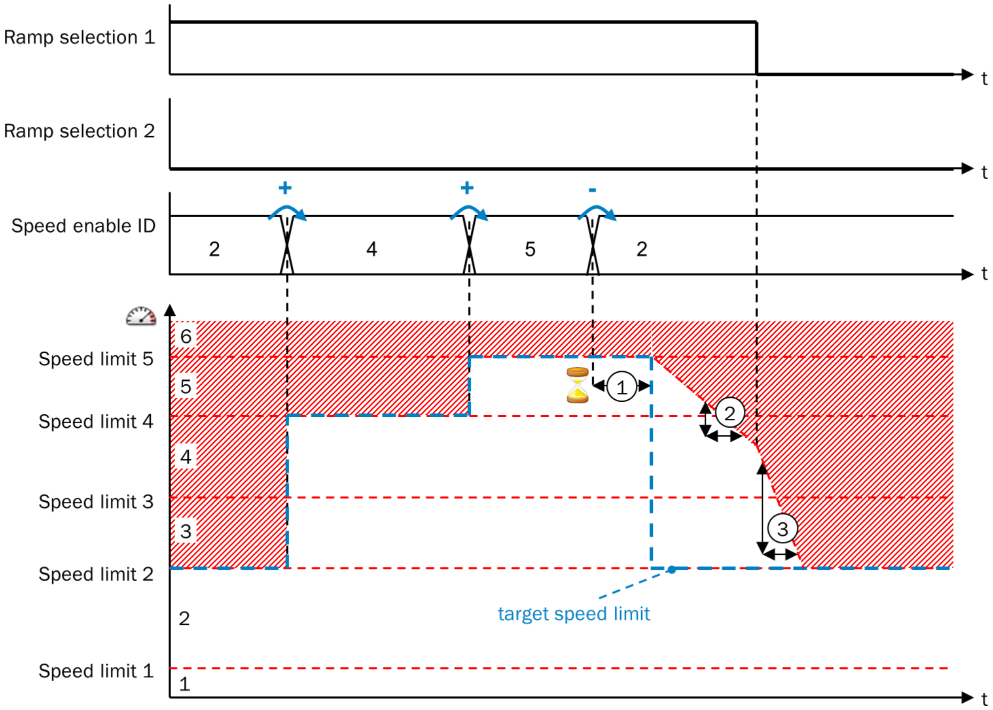


Figure 275: Speed ramp selection

① Delay time until start of ramp

② Ramps speed transition 2

③ Ramps speed transition 1

NOTE

The internal value of the current speed limit can be tracked in the online monitor of the logic editor and recorded in the data recorder.

Standstill detection

Standstill detection can be used to implement the safe operating stop (SOS) function.

The **Standstill** output and the **Speed status ID** are set to 1 if either the standstill detection with standstill speed or the standstill detection with standstill position tolerance is met.

Standstill detection with standstill speed

Standstill detection with standstill speed becomes valid when the following conditions are met:

- The **Standstill acknowledge** input is 1 or deactivated.
- The speed at the **Motion In** input is set to reliability status 1 (reliable), is continuous, and remains lower than the standstill speed for at least as long as the **standstill speed acceptance time**.

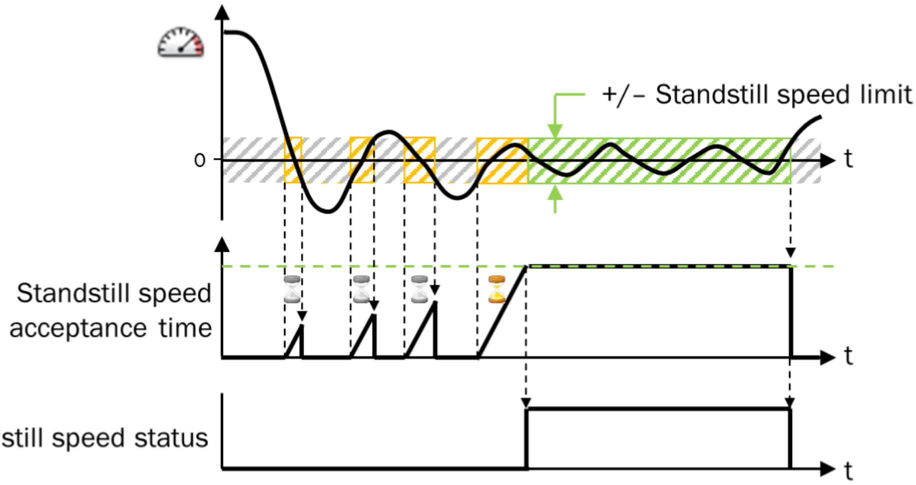


Figure 276: Standstill detection with standstill speed acceptance time

NOTE
The **standstill speed acceptance time** is also taken into account whenever the reliability bit for the speed value at the **Motion In** input switches from 0 (unreliable) to 1 (reliable). This also applies when the Flexi Soft system transitions to the Run status.

Standstill detection with standstill position tolerance

A **standstill position tolerance** can be defined. For this, there must be a signal present at the **Motion In** input that contains a valid relative position (e.g., from an A/B incremental, Sin/Cos, or SSI encoder).

Standstill detection with standstill position tolerance takes effect when the following conditions are fulfilled:

- The **Standstill approval** input is 1 or deactivated.
- The speed at the **Motion In** input is set to reliability status 1 (reliable), has reached the 0 value three times, or has changed signs (i.e., has exceeded the zero line)
- The corresponding relative position values at the **Motion in** input have the reliability status 1 (valid) and are within the potential **Standstill position tolerance** values.

Once this happens, the current speed is no longer taken into account even if it is greater than the **Standstill speed**. This remains the case until the **Standstill position tolerance** is exceeded or the relative position becomes unreliable.

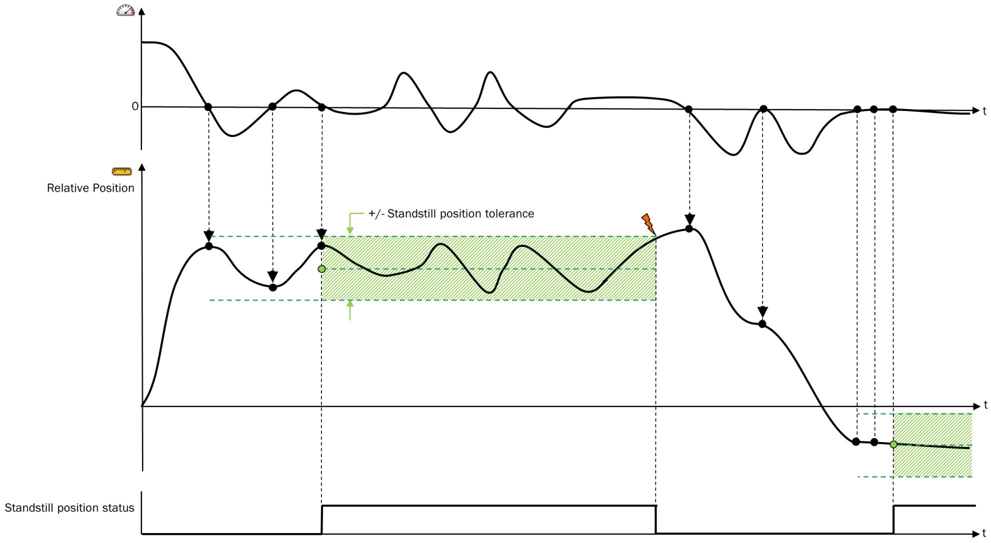


Figure 277: Standstill detection with standstill position tolerance

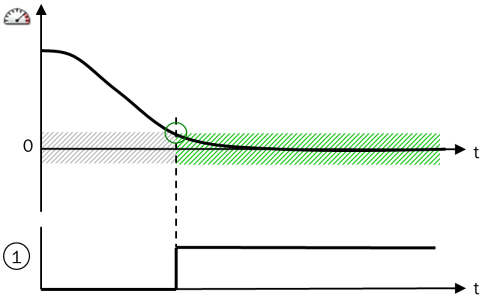
NOTE

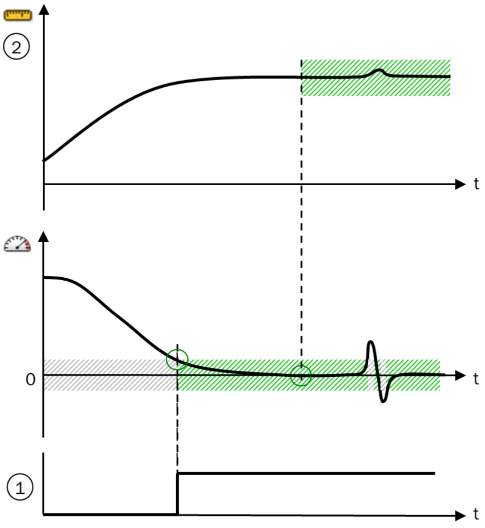
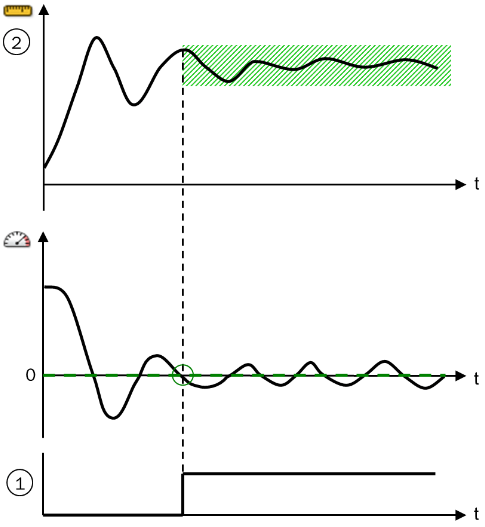
If you are using a **Standstill position tolerance** with the **Standstill speed** deactivated, the position must be maintained and the outputs **Standstill** and **Speed status ID** must remain set to 1 only for as long as the position remains within the **Standstill position tolerance.** Even if the speed moves away from the **Standstill position tolerance** value at a very slow speed, the standstill condition is no longer met.

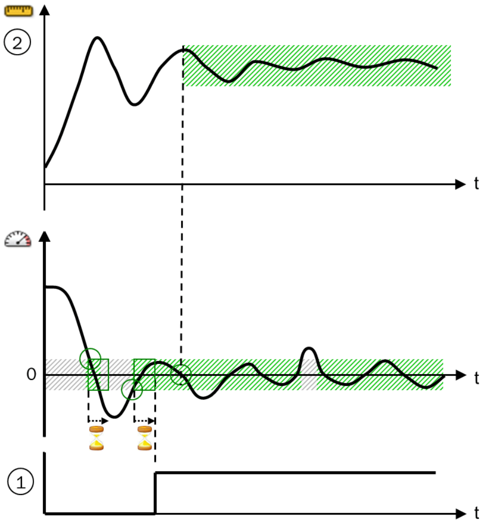
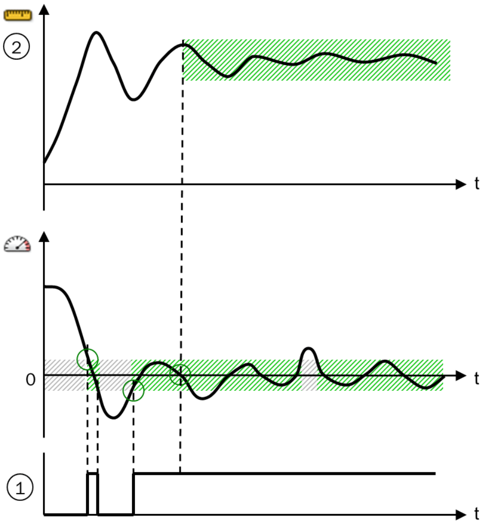
If you are using the **Standstill acknowledge** input, the **standstill position tolerance** is not determined again until this input has switched to 0 and back to 1.

Configuration examples for standstill

Table 172: Configuration examples for standstill

Speed curve	Description and configuration
 ① Standstill	Asymptotic approximation of the speed to zero. A safe standstill speed > 0 has been selected so that the standstill status can be achieved as quickly as possible. **Configuration** Standstill speed > 0 Standstill speed acceptance time = 0 Standstill position tolerance = deactivated

Speed curve	Description and configuration
 ① Standstill ② Relative position	Asymptotic reduction in speed to zero with possible subsequent peak in speed; e.g., due to mechanical shock. A safe standstill speed > 0 and a safe standstill position tolerance > 0 have been selected so that the standstill status can be achieved as quickly as possible while simultaneously tolerating peaks in speed. **Configuration** Standstill speed > 0 Standstill speed acceptance time = 0 Standstill position tolerance > 0
 ① Standstill ② Relative position	Reduction in speed with oscillation around zero; e.g., due to positioning control. A standstill position tolerance > 0 has been selected that is regarded as safe so that the standstill status can be achieved as quickly as possible but not until the standstill position tolerance requirement has been met (and not simply as soon as a speed limit is undershot). **Configuration** Standstill speed = deactivated Standstill speed acceptance time = 0 Standstill position tolerance > 0

Speed curve	Description and configuration
 ① Standstill ② Relative position	Reduction in speed with oscillation around zero; e.g., due to positioning control. A safe standstill speed > 0 and a safe standstill position tolerance > 0 have been selected so that the standstill status can be achieved as quickly as possible but also while tolerating peaks in speed. To prevent the standstill status from being triggered as soon as the speed drops to zero for the first time, a standstill speed acceptance time > 0 has also been selected. **Configuration** Standstill speed > 0 Standstill speed acceptance time > 0 Standstill position tolerance > 0
 ① Standstill ② Relative position	Reduction in speed with oscillation around zero; e.g., due to positioning control. A safe standstill speed > 0 and a safe standstill position tolerance > 0 have been selected so that the standstill status can be achieved as quickly as possible but also while tolerating peaks in speed. In this case, the standstill status may be activated and subsequently deactivated again at the beginning. For this reason, this configuration is **not recommended**. The **Maximum distance for speed filter** function can be used as a remedy for the speed peak. **Configuration** Standstill speed > 0 Standstill speed acceptance time = 0 Standstill position tolerance > 0

Standstill acknowledge

The optional **Standstill approval** input can be used to deactivate internal standstill detection. If this input is used, the **Standstill** and **Speed status ID** outputs can only switch to 1 if both the standstill condition is met and the **Standstill approval** input is 1. This applies to standstill detection with **Standstill speed** and to standstill detection with **Standstill position tolerance.**

The internal standstill detection with **Standstill speed** and, if applicable, **Standstill speed acceptance time** and **Standstill position tolerance** functions independently of the **Standstill approval** input.

Direction detection

The **Direction status** output indicates the direction of movement:

0 = forward (positive speed) or status unknown (speed at **Motion in** input is invalid)

1 = backward (negative speed)

When the Flexi Soft system transitions to the Run status, the **Direction status** output is 0.

The direction status does not change in the event of a standstill. This means that when a movement is performed in one direction with intermediate stops, the indicated direction does not change.

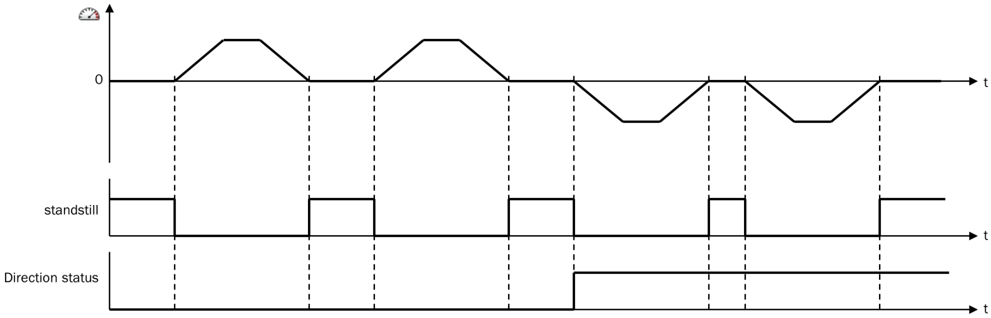


Figure 278: Sequence/timing diagram for direction status

Safe direction (SDI)

The optional **Enable forward** and **Enable backward** inputs can be used to enable the permissible direction of movement. If the current status is not Standstill (**Standstill** output is 0) and if the current direction of movement is not enabled, the **Monitoring status** output switches to 0.

If either of these inputs is not used, it means that the associated direction of movement is permanently enabled.

Maximum distance for speed filter

This function can be used to tolerate short-term increases or decreases in speed. The **Maximum distance for speed filter** parameter determines the extent to which exceeding the relevant speed limit is tolerated (filtered). This involves configuring a maximum distance that the drive may additionally travel despite exceeding the relevant speed limit.

As this is not a time-based filter, this function does not increase the response time. Instead, the current speed is used to calculate the additional distance that will be covered before the next logic execution cycle, assuming that the speed remains the same (anticipated additional distance); this value is then added to the additional distance that has already been traveled. If the total is larger than the **Maximum speed filter distance** parameter, the breach of the speed limit will no longer be tolerated. This means that if the speed limit has been exceeded to the extent that the **Maximum speed filter distance** will be exceeded in the next cycle, the breach of the speed limit becomes active immediately.

The calculated additional distance is reduced by falling below the speed limit. This also applies if the speed corresponds exactly to the speed limit. In this case, the calculated additional distance is diminished at the latest after 32 logic execution cycles, and the full tolerance is available again.

This function affects the following speed limits:

- Standstill speed monitor
- Speed monitor, selected via the **Speed enable ID** input

The function affects the following outputs:

- **Standstill** output
- **Speed status ID** output

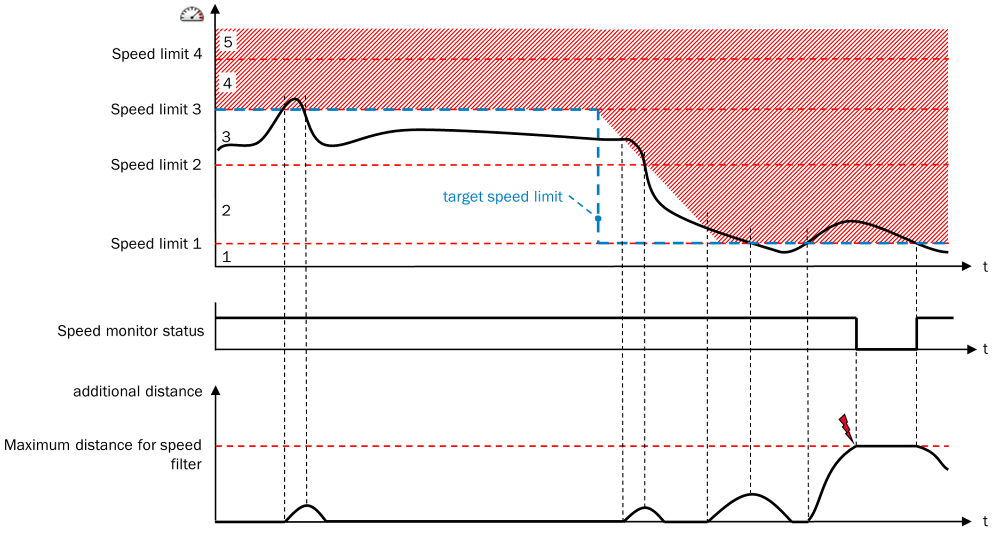


Figure 279: Maximum speed filter distance

NOTE

At the **Speed status ID** output, the function also takes effect if the lower speed limit for the current speed range is not met. The output only changes to another value after the tolerance specified by this filter function has been exceeded.

Exception: If the speed at the **Motion In** input becomes unreliable, the **Speed status ID** output immediately switches to 0 (invalid).

11.10.7 Position monitor V1

Function block diagram

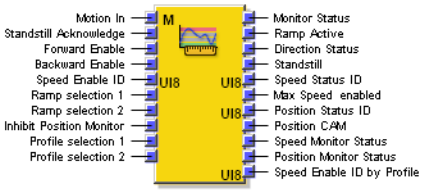


Figure 280: Inputs and outputs of the Position monitoring V1 function block

General description

The **Position monitor V1** function block is the central block for all position, speed, and direction monitoring functions within an application. It contains the functions of the **Speed monitor V2** function block and additional functions for position evaluating and monitoring.

The **Position monitor V1** function block can essentially perform the following functions:

- Safely-limited position (SLP)
- Safe cam (SCA)
- Safe speed monitor (SSM)
- Safely-limited speed (SLS)
- Safe direction (SDI)
- Safe operating stop (SOS)
- Monitoring of up to four different speed ramps during the transition from a monitored speed to a lower speed.

Speed monitor

- Maximum speed monitoring
- Monitoring the speed limits selected via the inputs **Speed enable ID** and, where applicable, **Ramp selection 1** and **Ramp selection 2**
- Direction monitoring, selected via the inputs **Enable forward** and **Enable backward**

The speed monitoring status is indicated at the **Speed monitor status** output.

Position monitoring

Selected via the relevant active **Speed position profile**:

- Speed limit monitoring including standstill monitoring
- Direction monitoring
- Value of the **Position cam** output for each position range

The position monitoring status is indicated at the **Position monitor status** output.

The combined status of the monitoring function is displayed at the **Monitoring status** output (AND operation). This means that the effective speed limit is always the lowest speed limit selected (via the **Speed enable ID** or the speed enable ID based on the active **Speed position profile**).

Inputs of the function block

Table 173: Inputs of the Position monitor V1 function block

Input	Description	Signal value
Motion In	Data of the type Motion V2 is expected, either directly from an encoder or from another function block, e.g. **Position comparison V1**.	Motion V2 data
Standstill acknowledge	Optional input, activates internal standstill detection	0 = standstill detection deactivated 1 = standstill detection active
Enable forward and Enable backward	Optional inputs which enable the relevant direction of movement. If either of these inputs is not used, it means that the associated direction of movement is permanently enabled.	0 = direction not enabled 1 = direction enabled Input not used = direction permanently enabled
Speed enable ID	Activates the permitted speed limit	0 … 255
Ramp selection 1 and Ramp selection 2	Selection of up to four speed ramps with different gradients	0 or 1
Inhibit position monitor	Optional input, for temporarily inhibiting the position monitoring functions; e.g., if the absolute position is unreliable or invalid.	0 = position monitor active 1 = position monitor inhibited
Profile selection 1 and Profile selection 2	Switching between two different speed-position profiles	0 or 1

NOTE

The **Position monitoring V1** function block requires data with a reliability status of 1 at the **Motion In** input. The data must therefore be plausibility-checked, e.g., using a Sin/Cos encoder with analog voltage monitoring or through the use of two encoders connected to a **Position comparison** function block.

Outputs of the function block

Table 174: Outputs of the Position monitor V1 function block

Output	Description	Signal value
Monitoring status	The **Monitoring status** output indicates the combined status of the **Speed monitor status** and **Position monitor status** outputs (AND connection). If the status is unknown; i.e., if the relevant data at the **Motion In** input is unreliable or invalid, the output switches to 1. The initial status when the Flexi Soft system transitions to the Run status is 1.	0 = error detected 1 = OK (no error detected or status unknown)
Ramp active	Indicates whether a speed ramp is active	0 = No ramp active 1 = speed ramp active
Direction status	Indicates the direction of movement. No change during standstill. If the status is unknown; i.e., if the speed at the **Motion In** input is unreliable or invalid, the output switches to 0. The initial status when the Flexi Soft system transitions to the Run status is 0.	0 = Forwards or status unknown 1 = Backwards
Standstill	Indicates whether the standstill condition is met (standstill speed and/or standstill position, taking into account the filter and acceptance criteria such as the **Maximum distance for speed filter**parameter, the **Standstill acknowledge** input, **and the Standstill speed acceptance time** parameter) If the status is unknown; i.e., if the relevant data at the **Motion In** input is unreliable or invalid, the output switches to 0. The initial status when the Flexi Soft system transitions to the Run status is 0.	0 = No standstill or status unknown 1 = standstill
Speed status ID	Indicates which speed range the current speed at the **Motion in** input corresponds to. This is not affected by the speed limit currently active for monitoring.	0 … 10 0 = speed invalid or unreliable 1 = standstill 2 … 10 = speed range 2 … 10
Max. speed enabled	Indicates whether the highest configured speed limit is active (selected via the **Speed enable ID**) input. This is not affected by the current speed at the **Motion in** input.	0 = Maximum speed not enabled 1 = Maximum speed enabled
Position status ID	Indicates the current position range depending on the absolute position at the **Motion in** input.	0 … 63 0 = position invalid or unreliable 1 … 63 = Current position range
Position CAM	Can be configured for each position range in each speed-position profile. For implementing an electronic cam circuit	0 or 1
Speed monitor status	• Maximum speed monitoring • Speed limit and standstill monitoring, selected via the inputs **Speed enable ID** and, where applicable, **Ramp selection 1** and **Ramp selection 2** • Direction monitoring, selected via the inputs **Enable forward** and **Enable backward** If the status is unknown; i.e., if the relevant data at the **Motion In** input is unreliable or invalid, the output switches to 1. The initial status when the Flexi Soft system transitions to the Run status is 1.	0 = error detected 1 = OK (no error detected or status unknown)

Output	Description	Signal value
Position monitor status	Selected via the active speed-position profile: • Speed limit monitoring including standstill monitoring • Direction monitoring If the status is unknown, i.e., if the relevant data at the **Motion in** input is unreliable or invalid, or if the **Repress position monitoring** input is 1, then the **Position monitoring status** output switches to 1. The initial status when the Flexi Soft system transitions to the Run status is 1.	0 = error detected 1 = OK (no error detected, status unknown, or position monitor inhibited)
Speed enable ID according to profile	Indicates the speed limit that applies to the current position range, dependent on the active **Speed position profile.** If no speed limit has been selected for the current position range, the maximum speed is permitted according to the active **Speed position profile**. In this case, the output value is 31.	0 = invalid 1 = standstill 2 … 9 = Speed limit 2 … 9 31 = preset 255 = position monitor inhibited or no configured profile

Function block parameters

Table 175: Parameters of the Position monitor V1 function block

Parameter	Description	Possible values
Standstill monitoring		
Standstill monitoring	Activates the **Standstill speed monitor** function	• Deactivated • Active
Standstill speed	Defines which speed still counts as a standstill	0 … 32,766 digit = • 0 … 16,383 rpm • 0 … 32,766 mm/s
Standstill speed acceptance time	Defines the uninterrupted period of time for which the **Standstill speed** may not be exceeded	0 … 248 ms in 4 ms increments
Standstill position monitoring	Activates the **Standstill position monitoring** function	• Deactivated • Active
Standstill position tolerance	Defines which relative position change still counts as a standstill during standstill monitoring	0 … 500,000,000 digit = • 0 … 16,666 rev. • 0 … 2,000,000 mm
Standstill acknowledge	Activates the optional **Standstill acknowledge** input	• Deactivated • Active
Speed ranges		
Max speed limit	Defines the maximum speed that is allowed to occur within the system.	1 … 32,767 digit = • 0.5 … 16,383 rpm • 1 … 32,767 mm/s
Speed limit 1	Speed limit 1 always corresponds to the **standstill speed**	0 … 32,766 digit = • 0 … 16,383 rpm • 0 … 32,766 mm/s
Speed limit 2 … 9	Up to 9 speed limits, including standstill speed	
Maximum distance for speed filter	Defines the distance that the drive is allowed to travel despite exceeding the current speed limit before the excess speed causes the system to switch off.	0 … 65,534 digits (position) = • max. 2.18 rev. • max. 262 mm • 0 = Deactivated
Ramp for speed limit transitions		
Delay time until start of ramp	How long the function block expects there to be no response from the system; i.e., the length of time for which it does not expect a delay ramp	0 … 248 ms in 4 ms increments

Parameter	Description	Possible values
Configuration of ramps (ramp steepness, speed transitions 1 … 4)	Increment for the reduction in speed when changing from a higher active speed enable ID to a lower one, selected via the **Speed enable ID** input. You can define up to four different ramps. The **Speed reduction** and the **Duration of the speed reduction** are entered.	**0 … 2,147,418,112 digit =** - 0.5 … 16,383 (rpm)/ms - 1 … 65,535 (mm/s)/ms - 0 = No ramp
Position ranges		
Position limit 1 … 62	Up to 62 different position limits can be defined. The range that corresponds to the current absolute position at the **Motion in** input is output at the **Position status ID** output.	**–2,147,483,648 … +2,147,483,647 digits =** - +/– 71,583 rpm - +/– 8,590 m
Speed-position profiles		
Profile mode	Defines whether the speed-position profiles are activated depending on direction or independently of the direction.	- Direction-independent - Direction-dependent
Speed-position profile 1 … 2	**The following can be defined for each position range:** - Permitted speed (**Speed enable ID according to profile**) 1 = Standstill 2 … 9 = Speed limit 2 … 9 31 = preset - Status of the **Position cam** output - Permitted direction of movement	–
Optional inputs		
Speed enable ID	Activates the optional **Speed enable ID** input	- Disabled - Active
Enable forward	Activates the optional **Enable forward** input	- Disabled - Active
Backward enable	Activates the optional **Enable backward** input	- Disabled - Active
Repress position monitoring	Activates the optional **Inhibit position monitor** input	- Disabled - Active

Safe speed monitor (SSM)

Up to 9 speed limits (including standstill speed) can be configured for up to 10 speed ranges. Speed limit 1 is always standstill speed.

The **Speed status ID** output indicates which speed range corresponds to the current speed at the **Motion in** input. This is not affected by the speed limit that is currently active. The data is output as a UI8 value. To connect this value to Boolean signals, use the **UI8 to Bool V1** function block.

*Table 176: **Speed status ID** output for 9 configured speed limits*

Speed at Motion In input	Meaning	Speed status ID
The speed is invalid or unreliable.	Invalid	0
Standstill condition [1] met: - The speed is lower than the standstill speed for at least the duration of the standstill speed acceptance time. **or** - The standstill position tolerance is determined and not exceeded.	Standstill	1

Speed at Motion In input	Meaning	Speed status ID
No standstill and speed > standstill speed Speed ≤ speed limit 2	Speed range 2	2
Speed > speed limit 2 Speed ≤ speed limit 3	Speed range 3	3
Speed > speed limit n−1 Speed ≤ speed limit n	Speed range n	n
Speed > speed limit 8 Speed ≤ speed limit 9	Speed range 9	9
Speed > speed limit 9	Speed range 10	10

[1] If neither standstill speed monitoring nor standstill position monitoring is activated, then the Speed status ID output never switches to 1.

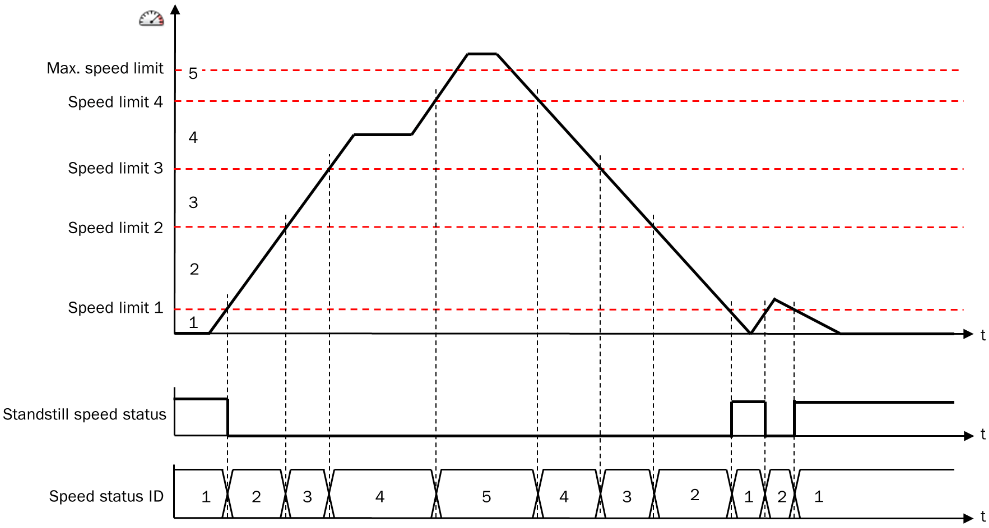


Figure 281: Sequence/timing diagram for the Speed status ID output (example involving four configured speed limits, i.e., five speed ranges)

NOTE

The configured value for the parameter **Max. speed** does not affect the speed status ID, i.e., the **Speed status ID** does not change even if the **max. speed** is exceeded.

Monitoring functions

The **Monitoring status** output indicates the combined status of the **Speed monitor status** and **Position monitor status** outputs (AND connection).

The **Monitoring status** output is usually connected to the **Safe stop 2A** input of the **Safe stop V2** function block. This means that an impermissible speed or direction will result in a stop.

The **Speed monitor status** output is the result of

- Maximum speed monitoring
- Speed limit and standstill monitoring, selected via the inputs **Speed enable ID** and, where applicable, **Ramp selection 1** and **Ramp selection 2**
- Direction monitoring, selected via the inputs **Enable forward** and **Enable backward**

The **Speed monitoring status** output is set to 1 (OK) upon transitioning to the Run status. The value switches to 0 (error detected) when one or more of the associated monitoring processes fail. The value switches back to 1 if one of the following conditions is met:

a) All the associated monitoring functions have been performed.
b) The speed at the **Motion In** input is unreliable or invalid.

The **Position monitor status** output is the result of

- Monitoring of the speed limits including standstill monitoring, selected via the active speed position profile
- Direction monitoring, selected via the active speed position profile

The **Status position monitoring** output is 1 (OK) by default. The value switches to 0 (error detected) when one or more of the associated monitoring processes fail. The value switches back to 1 if one of the following conditions is met:

a) All the associated monitoring functions have been performed.
b) The position at the **Motion In** input is unreliable or invalid.
c) The **Repress position monitoring** input is 1.

Speed monitoring functions

The functions for monitoring the **Max. speed** and the **Speed limits** can be used to implement the safely-limited speed (SLS) function. For transitions from a higher to a lower speed limit, **Speed ramps** can be configured.

Maximum speed monitoring

Max. speed monitoring is always active. If the current speed is greater than the **Max. speed** that has been configured, the **Monitoring status** output switches to 0. When used in conjunction with a **Safe stop V2** function block, this allows the maximum travel/maximum time for a Safe stop to be determined reliably.

The **Max. speed enabled** output is set to 1 when the highest configured speed limit is activated via the **Speed enable ID** input. This output can be used as a reset condition at the **Reset** input of a subsequent **Safe stop V2** function block. If, for example, the **Max. speed** is enabled as long as a safety door remains closed, you can reset a stop ramp that has been triggered by closing the safety door.

Monitoring of the speed limits

The optional **Speed enable ID** input activates the permissible speed limit. The input will accept a UI8 value (0 ... 255). To connect the input to Boolean signals, use the **Bool to UI8 V1** function block.

If the current speed at the **Motion in** input is greater than the active speed limit, then the **Monitor status** output is set to 0.

NOTE

- The values 0 and 1 at the **Speed enable ID** input activate standstill monitoring. If neither standstill speed monitoring nor standstill position monitoring is activated, then the **Monitoring status** output always remains at 0 (error), unless the speed at the **Motion in** input is unreliable or invalid.
- Any value above the number of configured speed limits activates the maximum permitted speed.

Speed ramps

The **Ramp speed transitions** parameters can be used to define up to four speed ramps. This allows the current speed limit to be reduced evenly from a higher to a lower speed limit in accordance with the configured increment instead of switching to the lower speed limit immediately. This happens regardless of the actual current speed, i.e., even if the actual speed is already below the new lower speed limit.

Up to four speed ramps with different increments can be defined. A ramp can be selected using the **Ramp selection 2** and **Ramp selection 1** inputs.

Table 177: Speed ramp selection

Input values		Selected ramp
Ramp selection 2	Ramp selection 1	
0	0	Ramp speed transition 1 (fastest ramp)
0	1	Ramp speed transition 2
1	0	Ramp speed transition 3
1	1	Ramp speed transition 4 (slowest ramp)

NOTE

Any change to the input values also affects any speed ramp that happens to be active when this change occurs.

The **Delay time until start of ramp** parameter specifies the delay time that must elapse before the speed ramp commences. This makes it possible, for example, to tolerate a delay in the drive response caused by communication and processing cycle issues.

While a speed ramp is active, the **Ramp active** output remains set to 1.

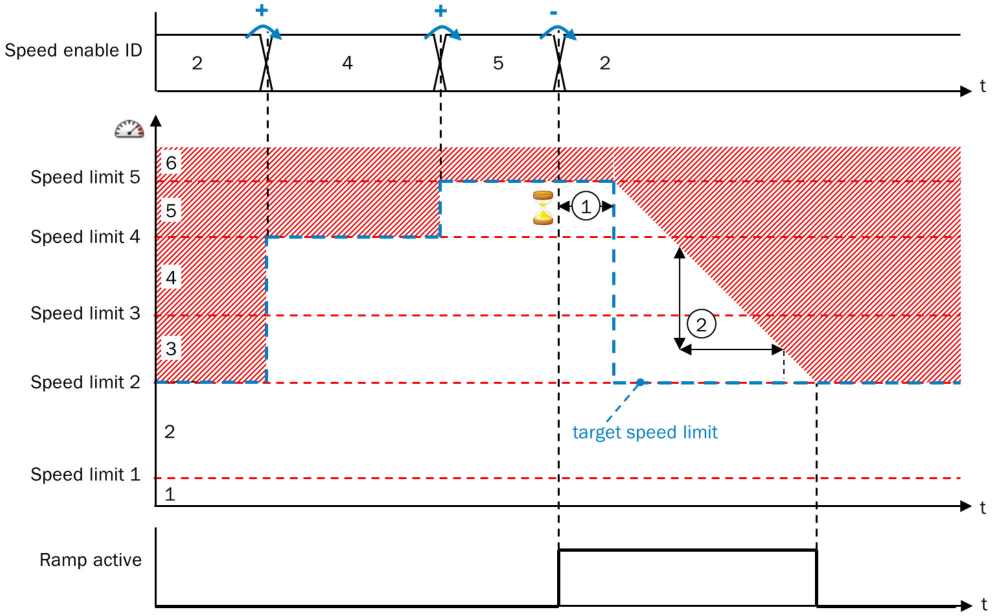


Figure 282: Example 1 for monitoring the speed limit

① Delay time until start of ramp

② Speed transition ramps

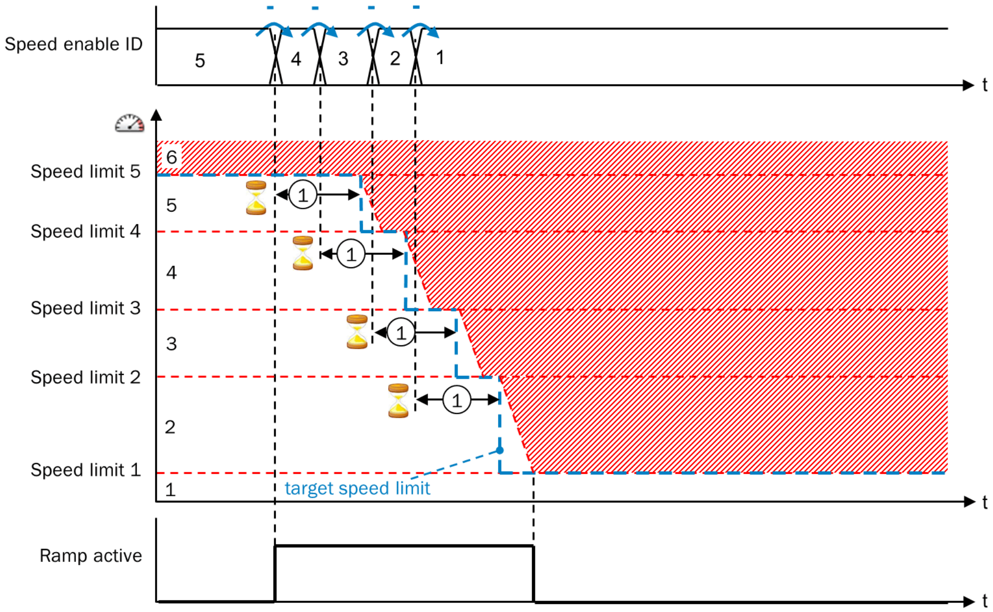


Figure 283: Example 2 for monitoring the speed limit

① Delay time until start of ramp

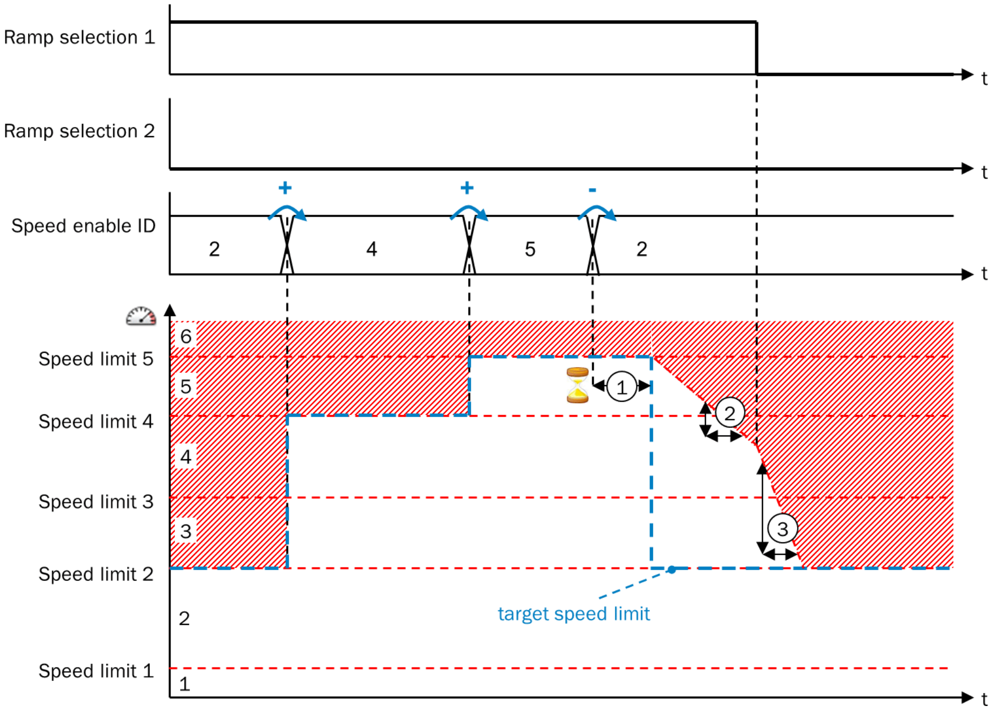


Figure 284: Speed ramp selection

① Delay time until start of ramp

② Ramps speed transition 2

③ Ramps speed transition 1

NOTE

The internal value of the current speed limit can be tracked in the online monitor of the logic editor and recorded in the data recorder.

Standstill detection

Standstill detection can be used to implement the safe operating stop (SOS) function.

The **Standstill** output and the **Speed status ID** are set to 1 if either the standstill detection with standstill speed or the standstill detection with standstill position tolerance is met.

Standstill detection with standstill speed

Standstill detection with standstill speed becomes valid when the following conditions are met:

- The **Standstill acknowledge** input is 1 or deactivated.
- The speed at the **Motion In** input is set to reliability status 1 (reliable), is continuous, and remains lower than the standstill speed for at least as long as the **standstill speed acceptance time**.

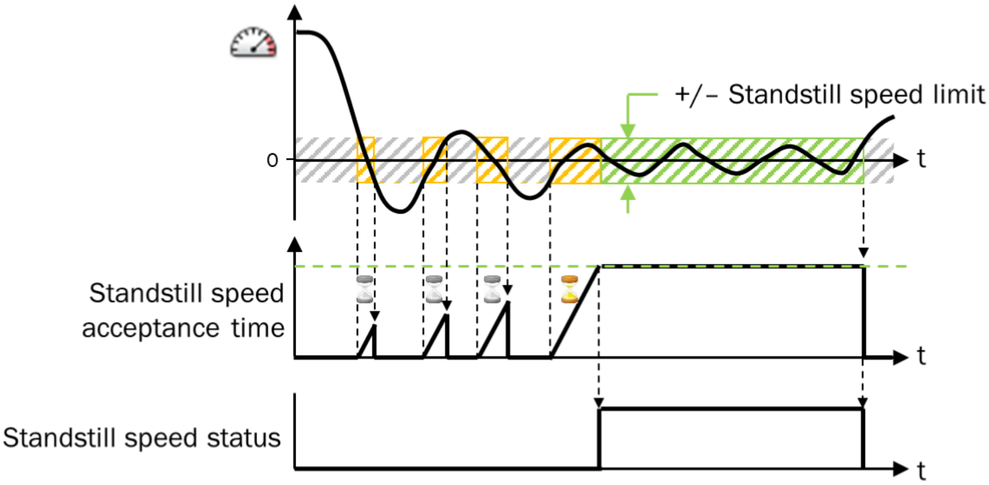


Figure 285: Standstill detection with standstill speed acceptance time

NOTE
The **standstill speed acceptance time** is also taken into account whenever the reliability bit for the speed value at the **Motion In** input switches from 0 (unreliable) to 1 (reliable). This also applies when the Flexi Soft system transitions to the Run status.

Standstill detection with standstill position tolerance

A **standstill position tolerance** can be defined. For this, there must be a signal present at the **Motion In** input that contains a valid relative position (e.g., from an A/B incremental, Sin/Cos, or SSI encoder).

Standstill detection with standstill position tolerance takes effect when the following conditions are fulfilled:

- The **Standstill approval** input is 1 or deactivated.
- The speed at the **Motion In** input is set to reliability status 1 (reliable), has reached the 0 value three times, or has changed signs (i.e., has exceeded the zero line)
- The corresponding relative position values at the **Motion in** input have the reliability status 1 (valid) and are within the potential **Standstill position tolerance** values.

Once this happens, the current speed is no longer taken into account even if it is greater than the **Standstill speed.** This remains the case until the **Standstill position tolerance** is exceeded or the relative position becomes unreliable.

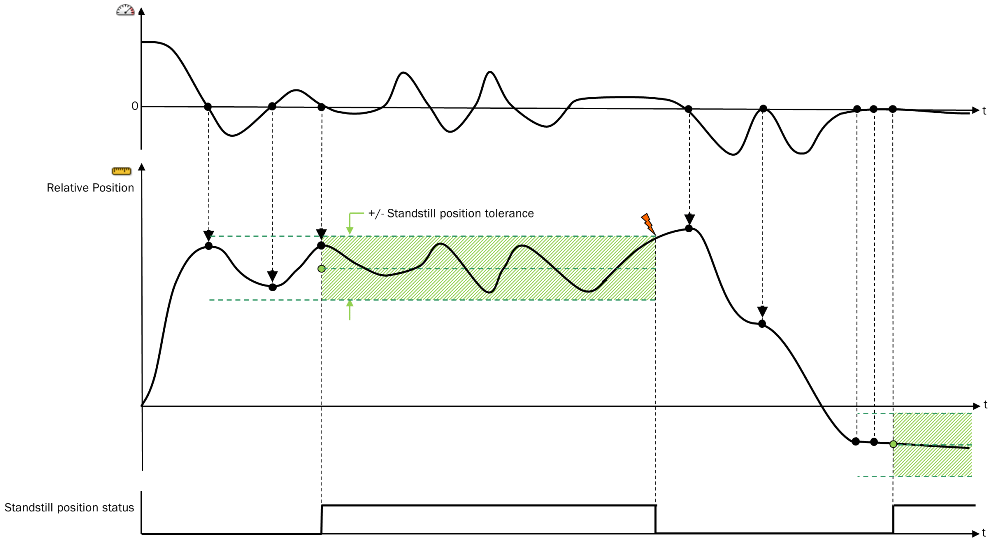


Figure 286: Standstill detection with standstill position tolerance

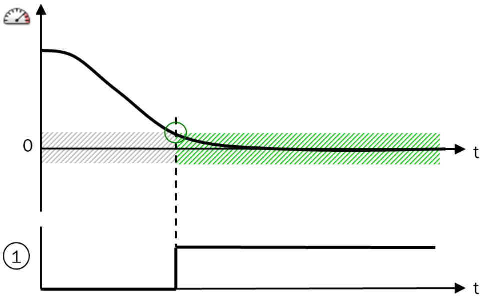
NOTE

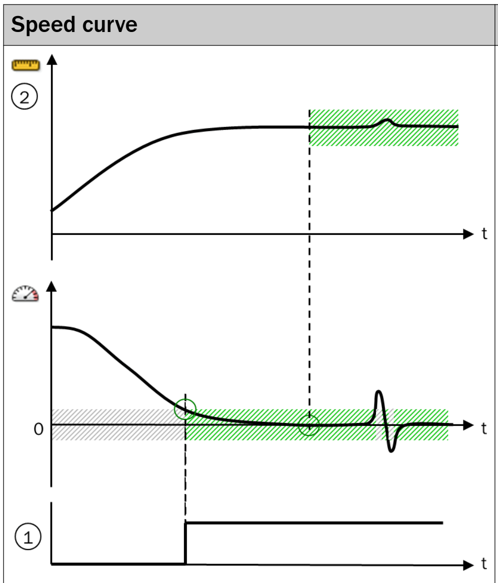
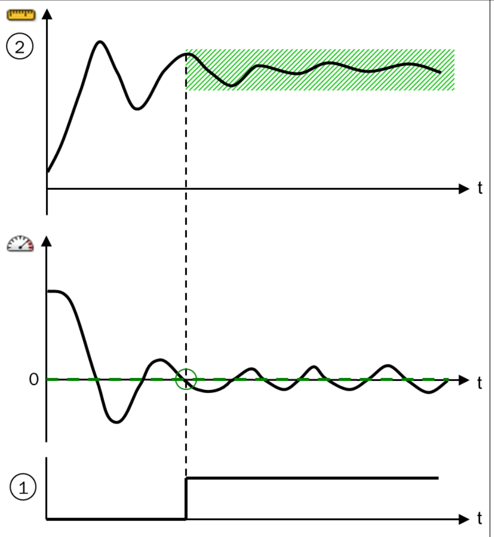
If you are using a **Standstill position tolerance** with the **Standstill speed** deactivated, the position must be maintained and the outputs **Standstill** and **Speed status ID** must remain set to 1 only for as long as the position remains within the **Standstill position tolerance.** Even if the speed moves away from the **Standstill position tolerance** value at a very slow speed, the standstill condition is no longer met.

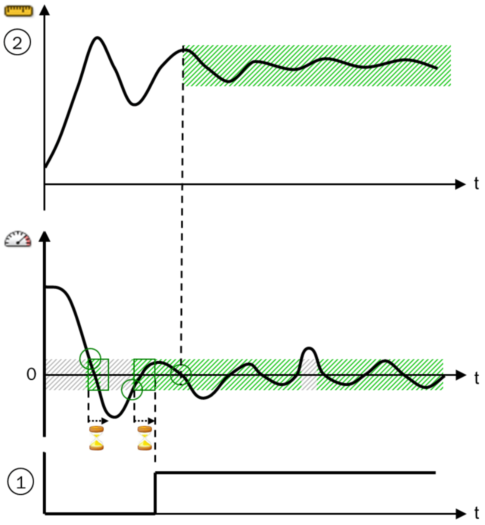
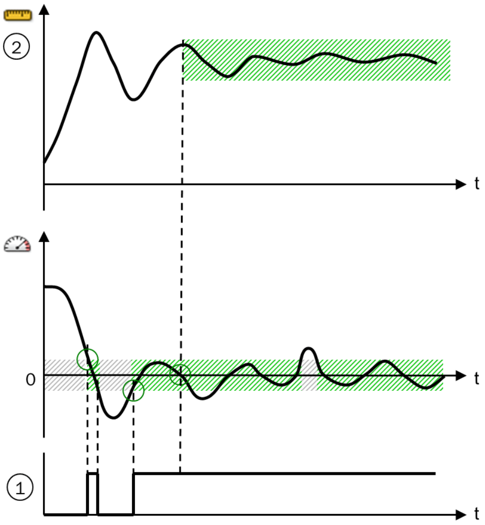
If you are using the **Standstill acknowledge** input, the **standstill position tolerance** is not determined again until this input has switched to 0 and back to 1.

Configuration examples for standstill

Table 178: Configuration examples for standstill

Speed curve	Description and configuration
 ① Standstill	Asymptotic approximation of the speed to zero. A safe standstill speed > 0 has been selected so that the standstill status can be achieved as quickly as possible. **Configuration** Standstill speed > 0 Standstill speed acceptance time = 0 Standstill position tolerance = deactivated

Speed curve	Description and configuration
 ① Standstill ② Relative position	Asymptotic reduction in speed to zero with possible subsequent peak in speed; e.g., due to mechanical shock. A safe standstill speed > 0 and a safe standstill position tolerance > 0 have been selected so that the standstill status can be achieved as quickly as possible while simultaneously tolerating peaks in speed. **Configuration** Standstill speed > 0 Standstill speed acceptance time = 0 Standstill position tolerance > 0
 ① Standstill ② Relative position	Reduction in speed with oscillation around zero; e.g., due to positioning control. A standstill position tolerance > 0 has been selected that is regarded as safe so that the standstill status can be achieved as quickly as possible but not until the standstill position tolerance requirement has been met (and not simply as soon as a speed limit is undershot). **Configuration** Standstill speed = deactivated Standstill speed acceptance time = 0 Standstill position tolerance > 0

Speed curve	Description and configuration
 ① Standstill ② Relative position	Reduction in speed with oscillation around zero; e.g., due to positioning control. A safe standstill speed > 0 and a safe standstill position tolerance > 0 have been selected so that the standstill status can be achieved as quickly as possible but also while tolerating peaks in speed. To prevent the standstill status from being triggered as soon as the speed drops to zero for the first time, a standstill speed acceptance time > 0 has also been selected. **Configuration** Standstill speed > 0 Standstill speed acceptance time > 0 Standstill position tolerance > 0
 ① Standstill ② Relative position	Reduction in speed with oscillation around zero; e.g., due to positioning control. A safe standstill speed > 0 and a safe standstill position tolerance > 0 have been selected so that the standstill status can be achieved as quickly as possible but also while tolerating peaks in speed. In this case, the standstill status may be activated and subsequently deactivated again at the beginning. For this reason, this configuration is **not recommended**. The **Maximum distance for speed filter** function can be used as a remedy for the speed peak. **Configuration** Standstill speed > 0 Standstill speed acceptance time = 0 Standstill position tolerance > 0

Standstill approval

The optional **Standstill approval** input can be used to deactivate internal standstill detection. If this input is used, the **Standstill** and **Speed status ID** outputs can only switch to 1 if both the standstill condition is met and the **Standstill approval** input is 1. This applies to standstill detection with **Standstill speed** and to standstill detection with **Standstill position tolerance.**

The internal standstill detection with **Standstill speed** and, if applicable, **Standstill speed acceptance time** and **Standstill position tolerance** functions independently of the **Standstill approval** input.

Direction detection

The **Direction status** output indicates the direction of movement:

0 = forward (positive speed) or status unknown (speed at **Motion in** input is invalid)

1 = backward (negative speed)

When the Flexi Soft system transitions to the Run status, the **Direction status** output is 0.

The direction status does not change in the event of a standstill. This means that when a movement is performed in one direction with intermediate stops, the indicated direction does not change.

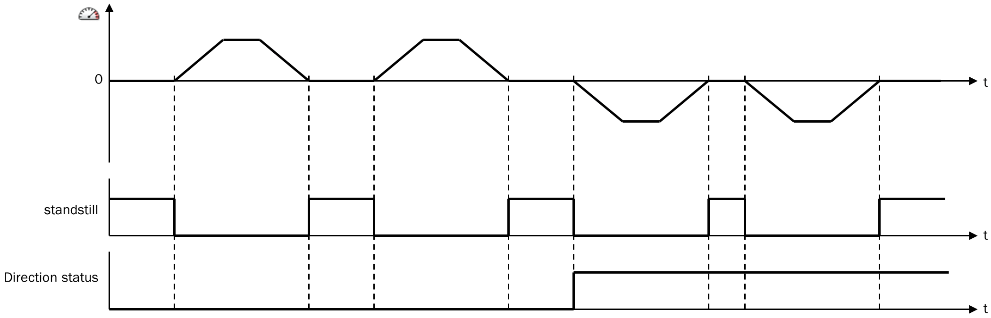


Figure 287: Sequence/timing diagram for direction status

Safe direction (SDI)

The optional **Enable forward** and **Enable backward** inputs can be used to enable the permissible direction of movement. If the current status is not Standstill (**Standstill** output is 0) and if the current direction of movement is not enabled, the **Monitoring status** output switches to 0.

If either of these inputs is not used, it means that the associated direction of movement is permanently enabled.

Position monitoring

Position monitoring can be used to implement the safely-limited position (SLP) and safe cam (SCA) functions.

You can define up to 62 position limits, allowing a maximum of 63 position ranges. The ID of the current position range is output at the **Position status ID** output based on the current absolute position at the **Motion In** input.

Table 179: Position ranges in the FX3-MOC1

Condition	Position status ID
The absolute position at the **Motion In** input is invalid or unreliable	0
No position limits are configured. **Or:** The absolute position at the **Motion In** input is ≤ Position limit 1	1
Position limit 1 < absolute position at the **Motion In** input ≤ Position limit 2	2
…	…
Position limit 61 < absolute position at the **Motion In** input ≤ Position limit 62	62
Absolute position at the **Motion In** input > Position limit 62	63

Position-dependent monitoring using speed-position profiles

You can configure two **Speed-position profiles** that allow different speeds or directions of movement depending on the current position.

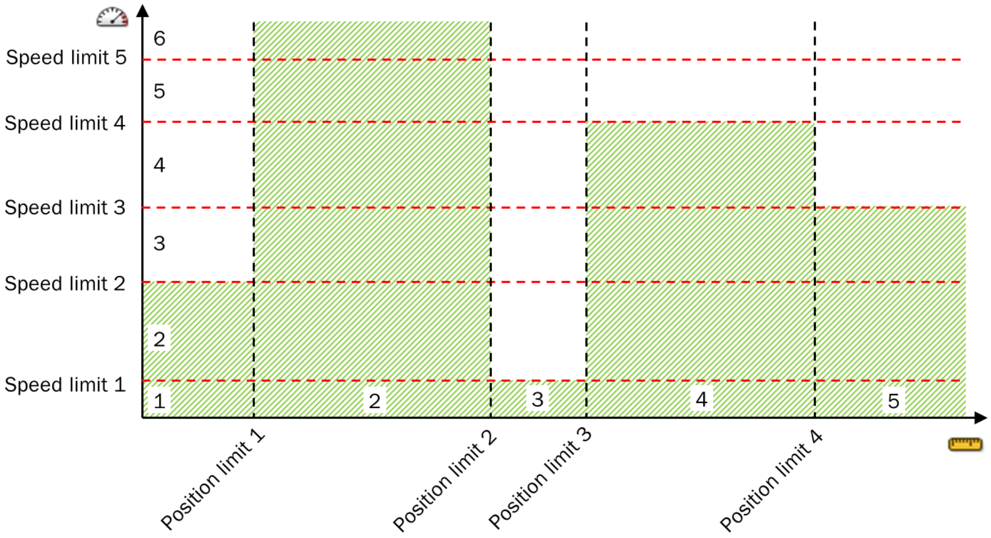


Figure 288: Speed position profile

Configuring speed-position profiles

- ► First, configure the required speed limits and position limits.
- ► Activate the desired **Profile mode** on the **Speed-position profiles** tab.
- ► Using the **Edit** selection button, activate editing mode for the position profile that you wish to edit and select the highest permissible speed for each position range in the matrix. Permissible speed ranges below the speed limit are shown in green.

NOTE

The profile you are editing is always shown on the display. Using the visibility selection field, the second profile can be shown on the display. The permissible speed ranges for the second profile are shown in gray.

NOTE

- The **Monitoring status** output is the combined status of the **Speed monitoring status** and **Position monitoring status** outputs (AND operation). This means that the effective speed limit is always the lowest speed limit selected (via the **Speed enable ID** or the speed enable ID based on the active **Speed position profile**).
  For example, the profile can be used to set the permitted speed for normal operation for the various position ranges and, if necessary, to switch to a lower speed or standstill monitoring via the **Speed enable ID** input (e.g., for troubleshooting or in set-up operation).
- The speed ramps are taken into account when the speed is specified via the **Speed enable ID** input, as the system must react if the speed limit is reduced due to external influences. On the other hand, monitoring by means of the **Speed enable ID** takes place without the speed ramps taken into account, because the system can predict the transition to a lower speed limit in another position range.
- Position-dependent direction monitoring: In the **Speed position profile**, the permitted direction of movement can be enabled depending on the position. If the current status is not standstill (standstill output is 0) and if the current direction of movement is not enabled, the **Position monitoring status** output switches to 0.
  Since the **Monitoring status** output is the combined status of the **Speed monitor status** and **Position monitor status** outputs (AND connection), this means that a direction of movement is only permitted if it is enabled by one of the **Enable forward** or **Enable backward** inputs, and is specified by the speed-position profile.

Safe cam (SCA)

In each speed position profile, the value of the **Position cam** output can be individually defined for each position range (0 or 1). This allows this output to be used to implement electronic cams.

► On the **Speed-position profiles** tab, activate editing mode for the required profile and select or do not select the **Position cam** option for each position range. Selected position cams are displayed in green.

If the relevant profile is active and the vehicle or the machine enters a position range that has an active position cam, the **Position cam** output switches to 1. If the vehicle or the machine enters a position range that has no active position cam, the **Position cam** output switches to 0.

Maximum distance for speed filter

This function can be used to tolerate short-term increases or decreases in speed. The **Maximum distance for speed filter** parameter determines the extent to which exceeding the relevant speed limit is tolerated (filtered). This involves configuring a maximum distance that the drive may additionally travel despite exceeding the relevant speed limit.

As this is not a time-based filter, this function does not increase the response time. Instead, the current speed is used to calculate the additional distance that will be covered before the next logic execution cycle, assuming that the speed remains the same (anticipated additional distance); this value is then added to the additional distance that has already been traveled. If the total is larger than the **Maximum speed filter distance** parameter, the breach of the speed limit will no longer be tolerated. This means that if the speed limit has been exceeded to the extent that the **Maximum speed filter distance** will be exceeded in the next cycle, the breach of the speed limit becomes active immediately.

The calculated additional distance is reduced by falling below the speed limit. This also applies if the speed corresponds exactly to the speed limit. In this case, the calculated additional distance is diminished at the latest after 32 logic execution cycles, and the full tolerance is available again.

This function affects the following speed limits:

- Standstill speed monitor and speed monitor, selected via the **Speed enable ID** input
- Standstill speed monitor and speed monitor, selected via a **Speed-position profile** input

The function affects the following outputs:

- **Standstill** output
- **Speed status ID** output

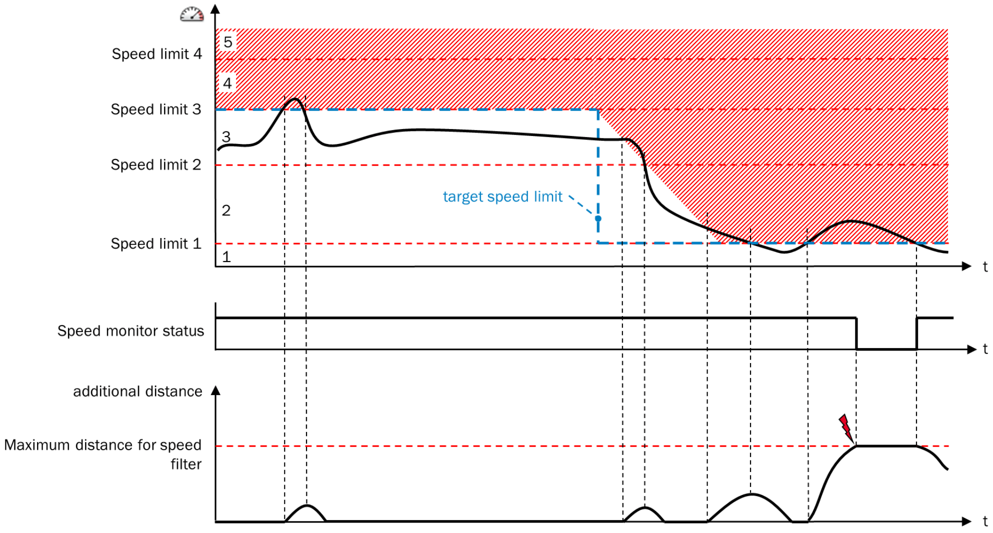


Figure 289: Maximum speed filter distance

NOTE

At the **Speed status ID** output, the function also takes effect if the lower speed limit for the current speed range is not met. The output only changes to another value after the tolerance specified by this filter function has been exceeded.

Exception: If the speed at the **Motion In** input becomes unreliable, the **Speed status ID** output immediately switches to 0 (invalid).

11.10.8 Safe Stop V2

Function block diagram

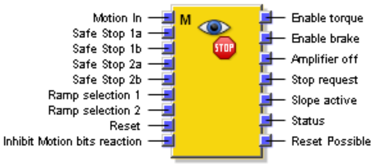


Figure 290: Inputs and outputs of the Safe stop V2 function block

General description

The **Safe Stop V2** function block is used to trigger and monitor the safe stop of a drive system. The drive has to be shut down in a controlled manner. The braking torque of the drive can be used to bring the drive to a standstill in a shorter period of time than an uncontrolled stop could.

The stop ramp of a drive system is not usually safe. For this reason, the **Safe stop V2** function block monitors the actual reduction in speed until the drive comes to a standstill.

Differences compared with the Safe stop V1 function block

- New **Maximum speed filter distance** function for filtering out spikes in speed while monitoring a stop ramp
- The standstill condition check integrated into the function block via the additional **Standstill speed** and **Standstill position tolerance** parameters replaces the **Standstill** input.
- A stop ramp is continued to its end, rather than being terminated prematurely, even if the standstill condition is fulfilled (i.e., direct transition to standstill monitoring).

- The **Torque off** status can be ended not only by a rising signal edge on one of the **Safe stop 1x** inputs but also by a rising signal edge on the **Reset** input. This is particularly important if the **Torque off** status was triggered by an error response (e.g., maximum permitted speed exceeded).
- The **Temporary standstill monitoring after Safe stop 1** phase can be ended before the **Switch-off delay for enable torque** time has expired if the reset conditions are fulfilled.
- Data type Motion V2 at the **Motion In** input rather than data type Motion V1. Unreliable speed values (speed value reliability = 0) will trigger a Safe stop 1.
- The **Repress motion bit response** input can be used to prevent a Safe stop 1 being triggered by invalid or unreliable speed values.
- If the speed is unreliable but valid (e.g., because the position cross check is failing), the ramp is monitored based on the unreliable speed and switched off if the maximum speed is exceeded.

Inputs of the Safe Stop V2 function block

Table 180: Inputs of the Safe stop V2 function block

Input	Description	Signal value
Motion In	Data of type Motion V2 is expected, either directly from an encoder or from another function block, e.g., **Position Cross Check V1**.	Motion V2 data
Safe stop 1A and Safe stop 1B	For triggering the Safe Stop 1 function	Falling signal edge (1–0)
Safe stop 2A and Safe stop 2B	For triggering the Safe Stop 2 function	Falling signal edge (1–0)
Ramp selection 1 and Ramp selection 2	Selection of up to four stop ramps with different gradients	0 or 1
Reset	Optional input for resetting the function block after a Safe stop	Rising signal edge (0–1)
Repress motion bit response	Prevents a Safe stop 0 being triggered if the speed at the **Motion in** input is unreliable or invalid.	0 = no inhibiting 1 = error response inhibited

Outputs of the Safe Stop V2 function block

Table 181: Outputs of the Safe Stop V2 function block

Output	Description	Signal value
Enable torque	Switches off the torque of the drive system, e.g., via the **External device monitoring** function block or (if available) via the safety capable inputs on the drive system for switching off the torque.	0 = switch off 1 = enable
Enable brake	Switches off the voltage supply for the mechanical brake (where applicable), e.g., via the **External device monitoring** function block.	0 = switch off 1 = enable
Amplifier off	Triggers switch-off of the amplifier and the drive torque plus – where applicable – engagement of the brake.	0 = switch off 1 = enable
Stop request	Triggers the stop ramp of the drive.	0 = stop requested 1 = No stop
Ramp active	Indicates whether a stop ramp is active.	0 = No ramp 1 = ramp active
Status	Indicates whether the torque of the drive has been switched off due to an impermissible speed or movement. If the status is unknown; i.e., if the relevant data at the **Motion In** input is unreliable or invalid, the output switches to 1. The initial status when the Flexi Soft system transitions to the Run status is 1.	0 = error detected 1 = OK (no error detected or status unknown)
Reset possible	Indicates whether a reset can be performed via the **Reset** input or either the **Safe Stop 1A** or **Safe Stop 1B** inputs.	0 = reset not possible 1 = reset possible

NOTE

The drive system can be controlled via the **Enable torque**, **Enable brake**, **Amplifier off**, and **Stop request** outputs of the function block.

The **Amplifier off** and **Stop request** outputs allow, for example, the drive system to be informed of the imminent safety response, giving it a chance to respond itself in a controlled manner before being switched off by the safety path.

Function block parameters

Table 182: Parameters of the Safe Stop V2 function block

Parameter	Description	Possible values
Stop ramps		
Delay time until start of ramp	Amount of time by which the beginning of the stop ramp should be delayed to take into account the drive system response time in the event of a stop request	0 … 248 ms in 4 ms increments
Stop ramp speed offset	Optional speed addition to the start value of the stop ramp. Prevents the stop ramp from being exceeded accidentally; e.g., due to mechanical vibrations.	**0 … 32,766 digits =** • 0.5 … 16,383 rpm • 1 … 32,766 mm/s • 0 = Deactivated
Steepness of stop ramp 1 … 4	Increments of speed reduction You can define up to four different ramps. The **Speed reduction** and the **Duration of the speed reduction** are entered.	**0 … 2,147,418,112 digit =** • 0.5 … 16,383 (rpm)/ms • 1 … 65,535 (mm/s)/ms • 0 = No ramp
Standstill monitoring		
Standstill speed	Defines which speed still counts as a standstill	**0 … 32,766 digit =** • 0 … 16,383 rpm • 0 … 32,766 mm/s
Standstill position tolerance	Defines which relative position change still counts as a standstill during standstill monitoring. While the **Standstill position tolerance** is not exceeded, the speed is not taken into account even if it is greater than the **Standstill speed**.	**0 … 500,000,000 digit =** • max. 16,666 rev. • max. 2,000,000 mm • 0 = Deactivated
Maximum distance for speed filter	Specifies the distance that the drive can travel, despite the maximum permitted speed being exceeded, before the exceeding of the maximum speeds leads to a switch off.	**0 … 65,534 digit (position) =** • max. 2.18 rev. • max. 262 mm • 0 = Deactivated
Off delays for Safe stop 1		
Off delay for Enable brake	Off delay for deactivating Enable brake based on when the **Amplifier off** output is switched off	0 … 248 ms in 4 ms increments
Off delay for Enable torque	Off delay for deactivating Enable torque based on when the **Amplifier off** output is switched off	0 … 248 ms in 4 ms increments

Description of operation

Drive systems usually have various “escalation levels”. The **Safe Stop V2** function block is used to implement the higher escalation levels.

Table 183: Typical escalation levels of a drive system

Level	Possible trigger	Control actions (not safe)	Safety functions
1	• Access to hazardous area required (e.g., warning field of an electro-sensitive protective device interrupted)	PLC reduces the control value for the speed of the drive; e.g., via fieldbus	Monitoring of the speed ramp by the **Speed monitor V2** or **Position monitor V1** function block
2	• Speed exceeds speed ramp • Protective field of an electro-sensitive protective device interrupted • Emergency stop pushbutton pressed	Drive system travels along stop ramp; e.g., using a digital input	Monitoring of the stop ramp (Safe Stop 1 or Safe Stop 2) by the **Safe Stop V2**. function block. The stop ramp is typically faster than the speed ramp associated with the **Speed monitor V2** and **Position monitor V1** function blocks.
3	• Speed exceeds stop ramp	Brake engages, drive system amplifier switched off	Deactivation of braking force energy and drive energy (**Torque off**), either via a cable or by using the inputs for switching off the torque on the drive

Activation of the outputs on system start

When the Flexi Soft system transitions to the Run status, all outputs except Ramp active and Reset possible switch to 1 if the following conditions are fulfilled:

- The speed at the **Motion in** input is reliable or the optional **Repress motion bit response** input is 1.
- All of the used **Safe stop X** inputs are 1.

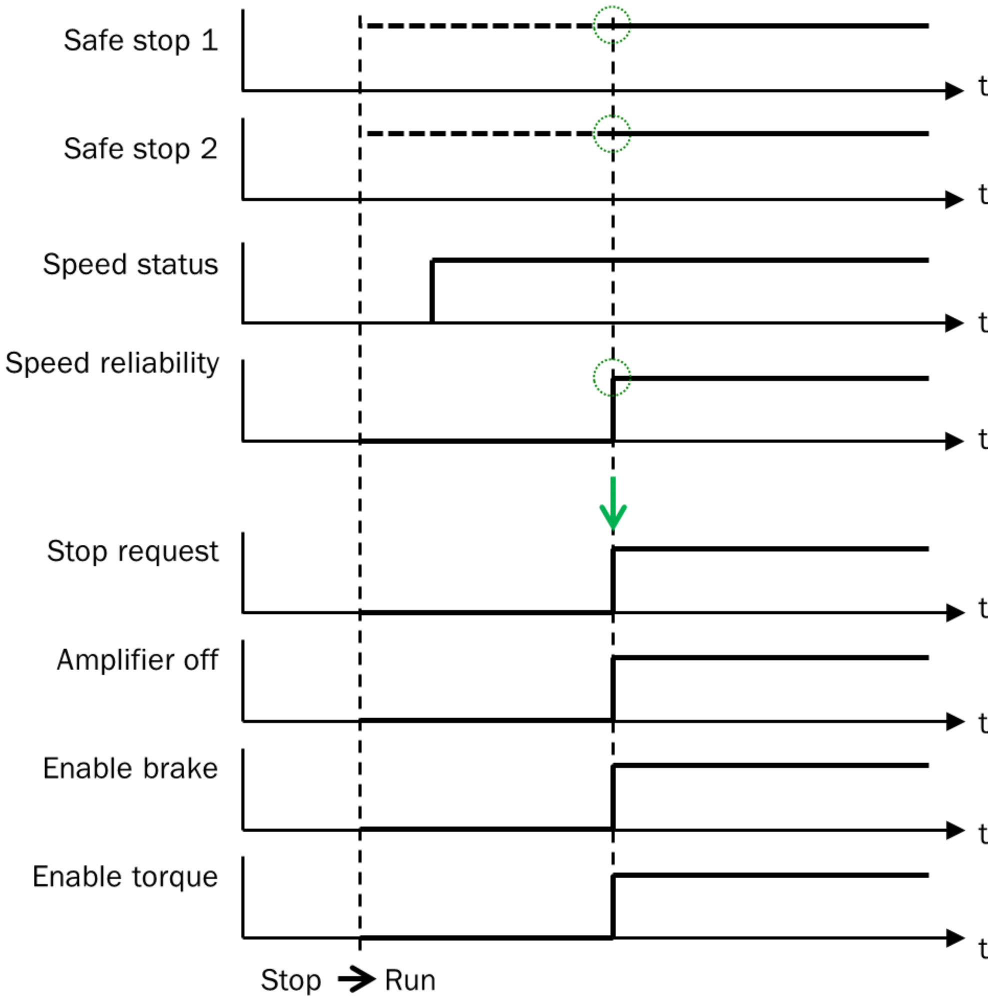


Figure 291: Conditions for activating the outputs

Maximum distance for speed filter

This function can be used to enable the system to tolerate short-term increases in speed. The **Maximum speed filter distance** parameter defines the extent to which any breach of the relevant speed limit will be tolerated (filtered). The parameter specifies the maximum length of the route that the drive may continue to travel in spite of the fact that the applicable speed limit has been exceeded.

As this is not a time-based filter, this function does not increase the response time. Instead, the current speed is used to calculate the additional distance that will be covered before the next logic execution cycle, assuming that the speed remains the same (anticipated additional distance); this value is then added to the distance that has already been traveled. If the total is larger than the **Maximum speed filter distance** parameter, the breach of the speed limit will no longer be tolerated. This means that if the speed limit has been exceeded to the extent that the **Maximum speed filter distance** will be exceeded in the next cycle, the breach of the speed limit becomes active immediately.

The calculated additional distance is reduced by falling below the speed limit. This also applies if the speed corresponds exactly to the speed limit. In this case, the calculated additional distance is diminished at the latest after 32 logic execution cycles, and the full tolerance is available again.

The Maximum distance for speed filter function is active during the following phases:

- Monitoring of the stop ramp
- Temporary standstill monitoring after Safe Stop 1
- Permanent standstill monitoring after Safe Stop 2

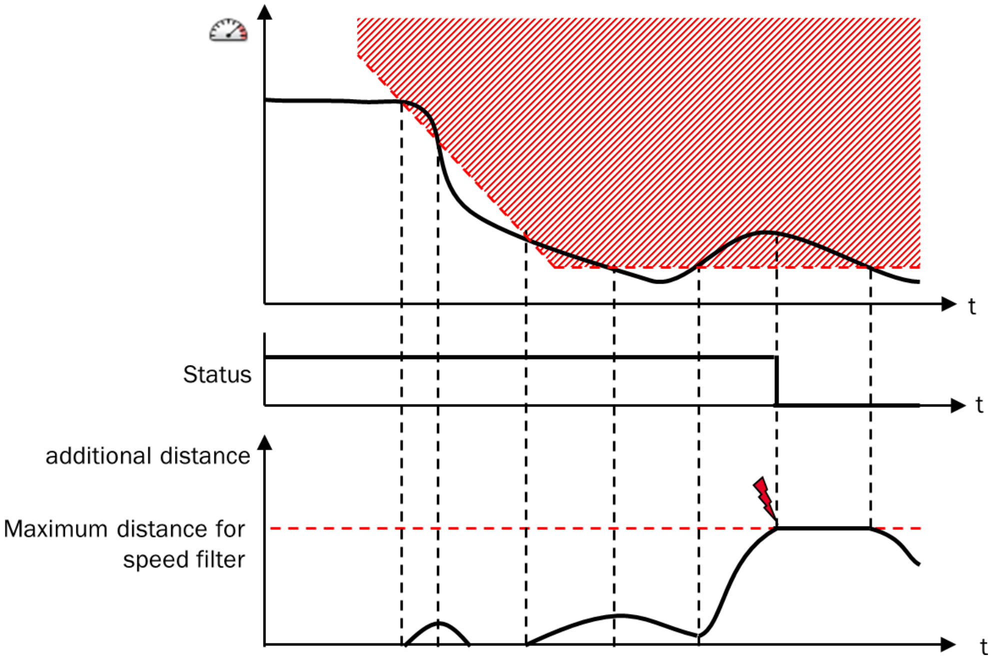


Figure 292: Maximum distance for speed filter function

Status output

The Status output indicates whether the drive torque has been switched off due to an impermissible speed or movement during a stop sequence. The Status output is 0 if one of the following conditions is met:

- The permissible speed is exceeded during stop ramp monitoring.
- The standstill condition is not met during standstill monitoring.

Reset possible output

The output Reset possible indicates whether a reset can be performed via the Reset input or either the Safe Stop 1A or Safe Stop 1B inputs. The Reset possible output is set to 1 if all the following conditions are met:

- The function block is in the **Standstill monitoring** or **Torque off** status.
- All **Safe Stop 2x** inputs in use are set to 1.
- At most, one of the **Safe Stop 1x** inputs in use is set to 0.
- The **Speed reliability** value at the **Motion in** input is 1 (reliable) or the **Repress motion bit response** input is in use and is 1.

In all other cases, the **Reset possible** output is set to 0.

Repress motion bit response

If the speed at the **Motion in** input becomes unreliable (**Speed reliability** = 0), e.g., due to a monitoring function within a function block located further along the signal path, a Safe stop 1 is triggered. However, the speed information remains in use until monitoring of the stop process commences. The speed continues to be evaluated if the speed value at the **Motion in** input is unreliable, but valid. If the speed at the **Motion in** input becomes invalid (**Speed status** = 0), then a Safe stop 0 is triggered, i.e., the system immediately switches to **Torque off** status.

This response can be repressed using the optional **Repress motion bit response** input. If the **Repress motion bit response** input is 1, the function block continues to respond as though the speed values at the **Motion in** input are valid and reliable.

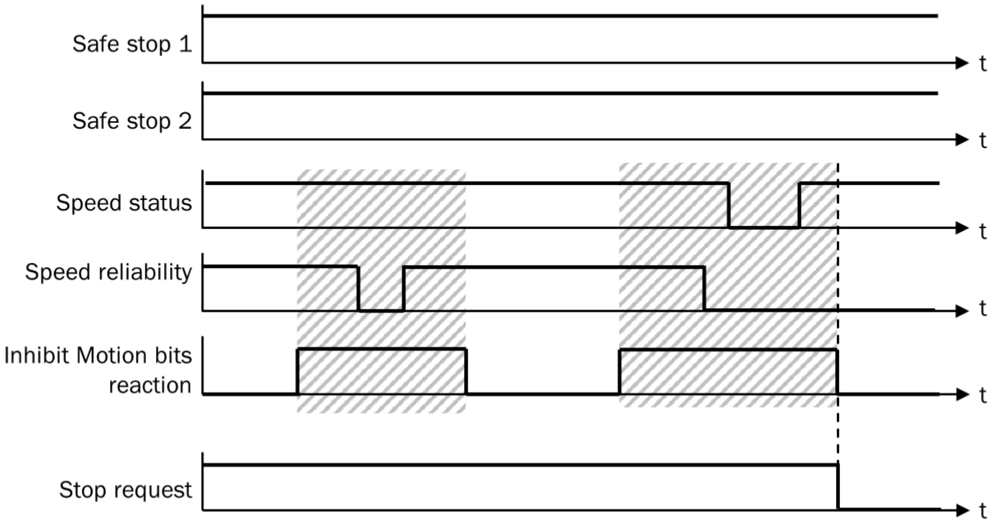


Figure 293: Repress motion bit response

WARNING
Restricted error detection

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

- ► Note that the use of the optional **Repress motion bit response** input in combination with a speed cross check function block or a position comparison function block leads to restricted error detection.
- ► Secure the plant using additional measures, e.g. a protective door.
- ► Make sure that all errors to be considered are detected.

In accordance with generally recognized testing principles, test authorities usually stipulate that the application must ensure the monitored unit performs a movement at least once within the space of 24 hours. This movement must generate a signal change on the encoder system so that the speed cross check or position comparison function can use it as a basis for detecting the relevant faults.

Safe stop 0, Safe stop 1, and Safe stop 2

The **Safe stop V2** function block supports three stop categories as defined in IEC 61800-5-2 and IEC 60204-1.

In the case of the Safe stop 0 (SS0) function, the drive system torque is switched off immediately. The function block executes a Safe stop 0 when the stop ramp conditions are not fulfilled or when the stop ramp cannot be monitored because the speed at the **Motion In** input is invalid.

Stop category 1 and stop category 2 differ in terms of how the stop ramp ends. In the case of the Safe stop 1 (SS1) function, the drive system torque is switched off after the drive has come to a standstill.

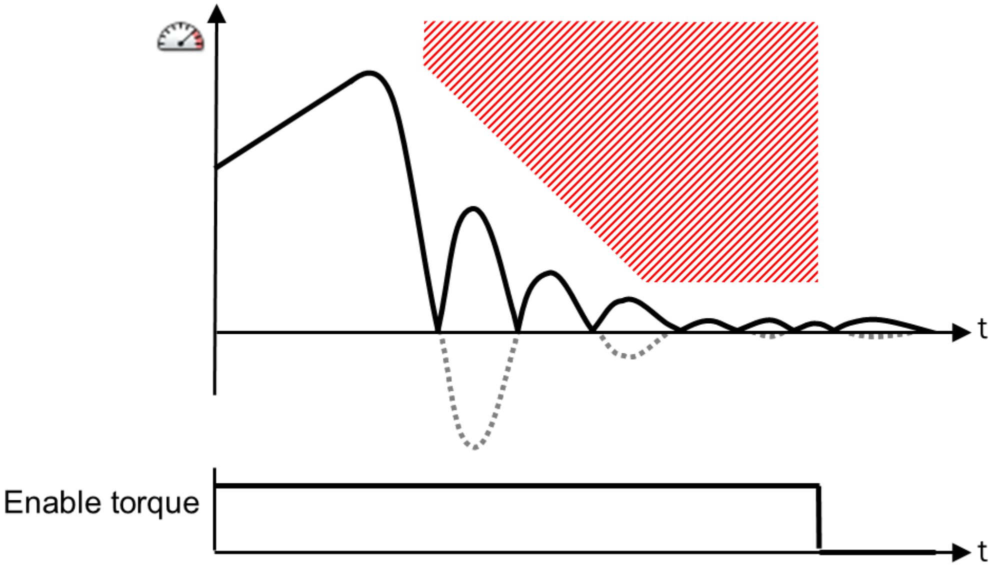


Figure 294: Principle of operation for Safe Stop 1

NOTE
The safe stop 1 function corresponds to a controlled stop in accordance with IEC 60204-1, stop category 1.

In contrast to this, the Safe Stop 2 (SS2) function keeps the torque enabled, although the standstill condition is monitored. This enables the drive to perform holding control.

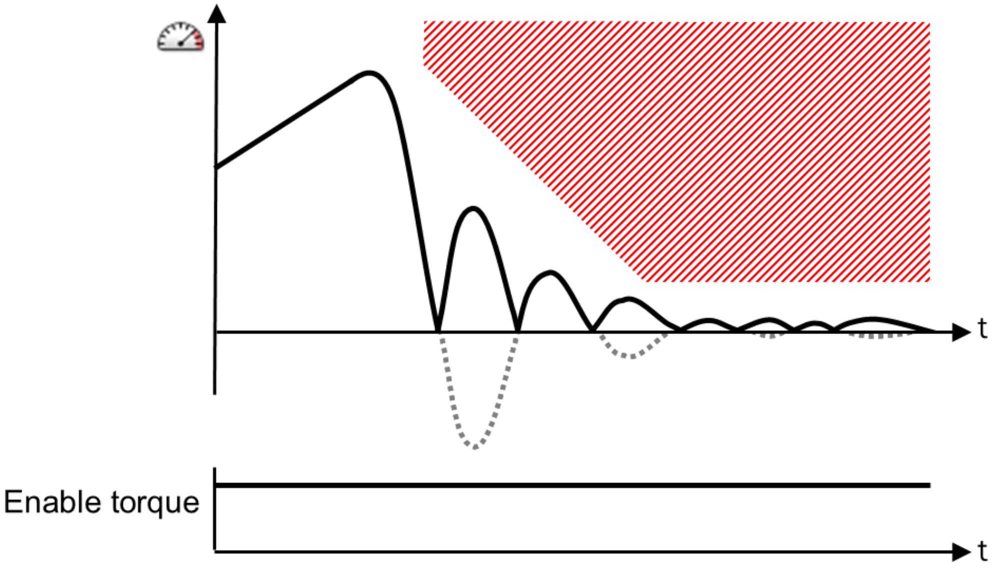


Figure 295: Principle of operation for Safe Stop 2

NOTE
The Safe Stop 2 function corresponds to a controlled stop in accordance with IEC 60204-1, stop category 2.

The two stop categories are divided into the following phases:

Table 184: Phases of Safe Stop 1 and Safe Stop 2

Phase	Safe Stop 1	Safe Stop 2
1	Wait for stop request	
2	Delay time for beginning of stop ramp	
3	Monitoring of the stop ramp	
4	Temporary standstill monitoring after Safe Stop 1	Permanent standstill monitoring after Safe Stop 2
5	Switch off torque	

11.10.8.1 Safe stop 1

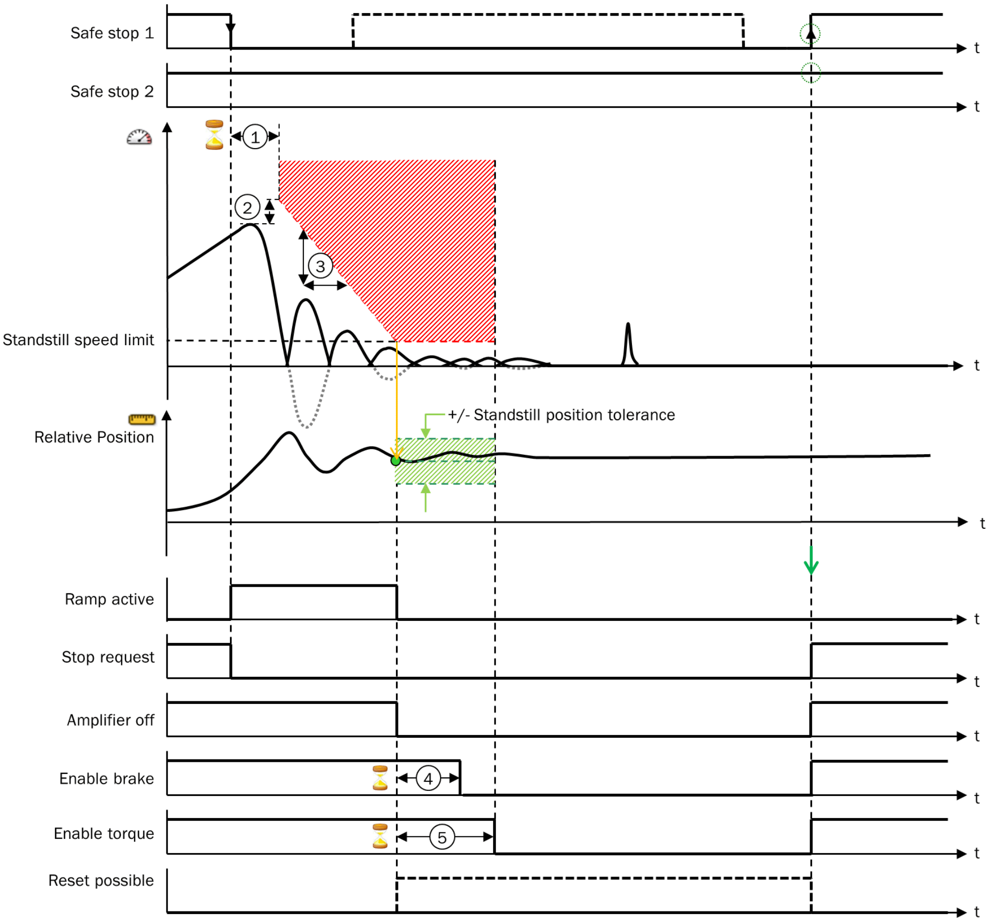


Figure 296: Monitoring function of Safe stop 1

① Delay time until start of ramp
② Stop ramp speed offset
③ Steepness of stop ramp 1 … 4
④ Off delay for Enable brake
⑤ Off delay for Enable torque

Phase 1: Wait for stop request

The **Safe stop V2** function block offers two optional inputs for each stop mode. A falling signal edge at any of these inputs triggers the corresponding stop mode, i.e., the delay time for the stop ramp starts running and the **Ramp active** output is switched to 1.

If a Safe stop 2 occurs first and then a Safe stop 1 is also triggered during one of the subsequent phases, the Safe stop 1 function has priority. This means that phase 5 of the Safe stop 1 function (switch off torque) will definitely be triggered.

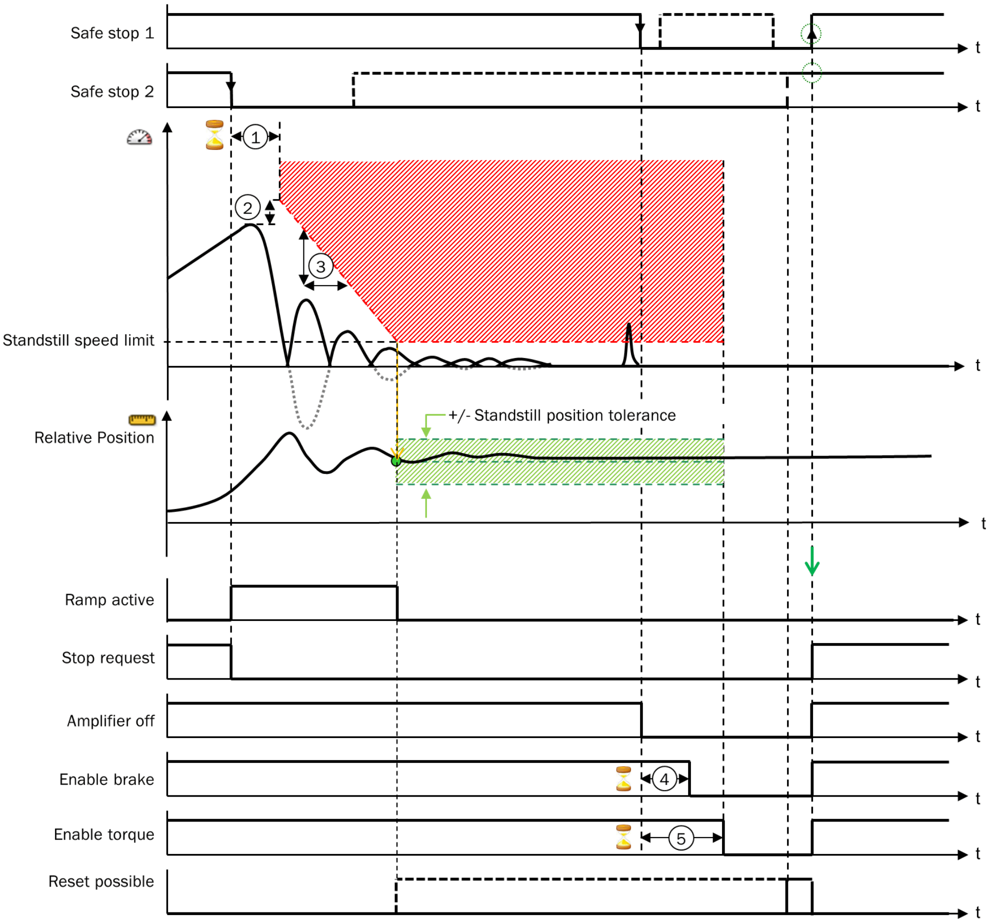


Figure 297: Safe stop 1 after Safe stop 2

① Delay time until start of ramp
② Stop ramp speed offset
③ Steepness of stop ramp 1 … 4
④ Off delay for Enable brake
⑤ Off delay for Enable torque

As soon as a stop is triggered, the **Stop request** output is set to 0. This output should be used to trigger the stop ramp of the drive system. This is generally a non-safe signal.

Phase 2: Delay time for beginning of stop ramp

The **Delay time until start of ramp** parameter can be used to configure a delay time for the beginning of the stop ramp. This should correspond to the amount of time it takes the drive system to respond to a stop request.

During this phase, the highest absolute speed is measured and used as the basis for the stop ramp start value. If no delay time has been configured (**Delay time until start of ramp** = 0), the current speed at the time of triggering is used as the start value instead.

At the start of this phase, the **Ramp active** output is switched to 1.

Phase 3: Monitoring of the stop ramp

The value of the **Stop ramp speed offset** parameter is added to the highest absolute speed (i.e., without a sign) that was measured during phase 2. The total is used as the start value for the stop ramp. In this way, the stop ramp is adapted to the current speed.

Monitoring of the stop ramp means that speed limiting begins with the start value and is then constantly reduced in accordance with the **Ramp steepness** parameter. The speed of the drive continuously compared to the current speed limiting. The optional configurable **Maximum speed filter distance** is taken into account.

If the speed of the drive exceeds the maximum permitted speed during stop ramp monitoring, stop ramp monitoring is immediately terminated and the function block switches to **Torque off** status. The **Enable torque**, **Enable brake**, **Amplifier off**, and **Status** outputs are immediately set to 0.

While the stop ramp is being monitored, the **Ramp active** output remains set to 1.

Up to four stop ramps with different increments can be defined. A ramp can be selected using the **Ramp selection 2** and **Ramp selection 1** inputs.

NOTE
The internal value of the stop ramp can be recorded in the data recorder.

Table 185: Selection of the stop ramp

Input values		Selected ramp
Ramp selection 2	Ramp selection 1	
0	0	Ramp steepness 1 (fastest ramp)
0	1	Ramp steepness 2
1	0	Ramp steepness 3
1	1	Ramp steepness 4 (slowest ramp)

NOTE
Any change to the input values also affects any stop ramp that happens to be active when this change occurs.

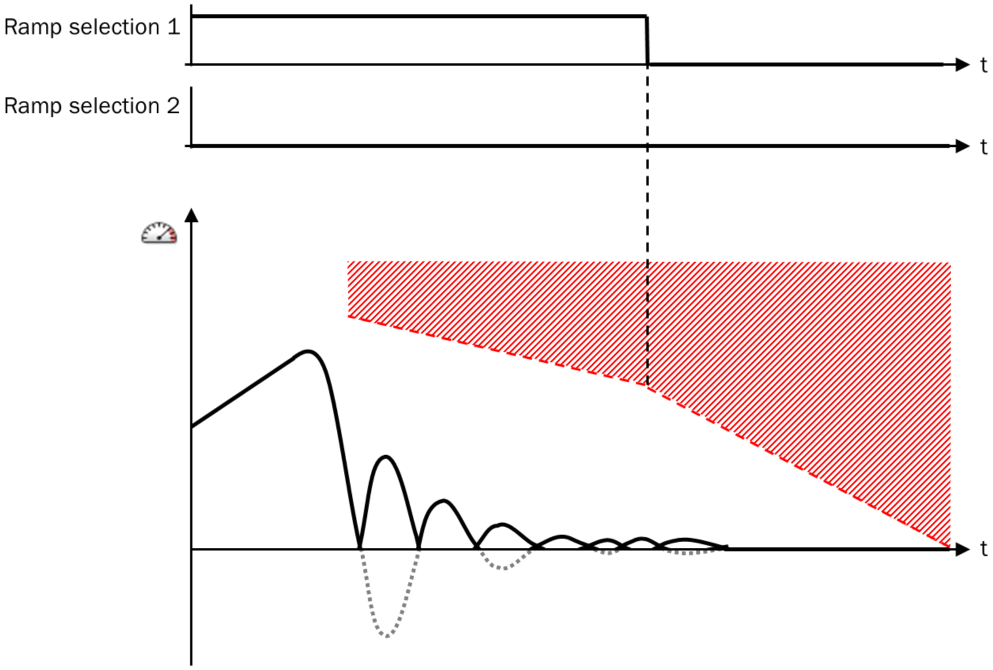


Figure 298: Selection of the stop ramp

NOTE

If a non-configured ramp is selected, then a ramp with infinite steepness is effected, i.e. the monitoring of the stop ramp ends (taking into account any configured delay) and the **Ramp active** input is set to 0.

This case can occur if only three ramps are configured and ramp 4 is selected (both inputs **ramp selection x** = 1) which is not configured.

If the speed limit is not exceeded, the monitoring of the stop ramp finishes as soon as the ramp ends. This is the case if the speed limit is the same as the configured standstill speed. This also applies if the drive previously reached the standstill speed, i.e. the monitoring of the stop ramp is not ended prematurely in this case either.

Once the stop ramp ends, the **Ramp active** output is set to 0. Furthermore, the current position is recorded as a reference for the standstill position monitoring. If there is no valid relative position value available at this point in time, then standstill position monitoring is not possible even if the relative position value becomes valid during the subsequent standstill monitoring. In this case, only the standstill speed monitoring is in effect.

Phase 4 of Safe stop 1: Temporary standstill monitoring after Safe stop 1

Phase 4 begins when the stop ramp has ended and the drive has reached the standstill speed. During a Safe stop 1, the **Amplifier off**, **Enable brake** and **Enable torque** outputs are all set to 0. An optional delay can be applied for the purpose of deactivating the **Enable brake** and **Enable torque** outputs.

- The **Amplifier off** output is deactivated immediately.
- Deactivation of the **Enable brake** output is delayed by the amount of time defined by the **Switch-off delay for enable brake** parameter.
- Deactivation of the **Enable torque** output is delayed by the amount of time defined by the **Switch-off delay delay for enable torque** parameter.

If the drive system is equipped with a brake, the **Switch-off delay for enable torque** parameter is usually set to a higher value than the **Off delay for enable brake** parameter, i.e., the torque is only switched off once the brake has been triggered. This is particularly useful for applications that involve heavy loads and where the torque is required to maintain the position so that the weight of the load does not cause the axis to move. In this case, the drive must be blocked by the brake before the torque is switched off.

During a Safe stop 1, phase 5 **Switch off torque** begins after the **Off delay for enable torque** ends.

During phase 4, the speed and, if necessary, the relative position are monitored at the **Motion in** input. If neither the standstill monitoring with **Standstill speed** or the standstill monitoring with **Standstill position tolerance** are fulfilled or deactivated at any point during this process, phase 5 **Switch off torque** is triggered immediately. The optional configurable **Maximum speed filter distance** is taken into account.

If **Standstill position monitoring** is active, the current speed is no longer taken into account even if it is greater than the **Standstill speed**. This remains the case until the **Standstill position tolerance** is exceeded or the relative position becomes unreliable.

Phase 5 **Switch off torque** (Stop 0) is also triggered if the speed at the **Motion in** input becomes invalid (**Speed status** = 0).

Resetting of Safe stop 1 during phase 4

During phase 4, the outputs can be reactivated by a rising signal edge at either the Safe Stop 1A or Safe Stop 1B input, or by a rising signal edge at the Reset input, subject to the following conditions:

- All of the **Safe stop X** inputs are set to 1.
- The speed at the **Motion in** input is valid and reliable or the optional **Repress motion bit response** input is 1.

The current speed is not taken into account. Consequently, a reset is possible even if the drive is still moving. This also applies to stops that have been triggered because the speed at the **Motion In** input was invalid or unreliable.

Phase 5: Switch off torque

In phase 5, the **Enable torque**, **Enable brake,** and **Amplifier off** outputs are always deactivated without any further delay.

11.10.8.2 Safe stop 2

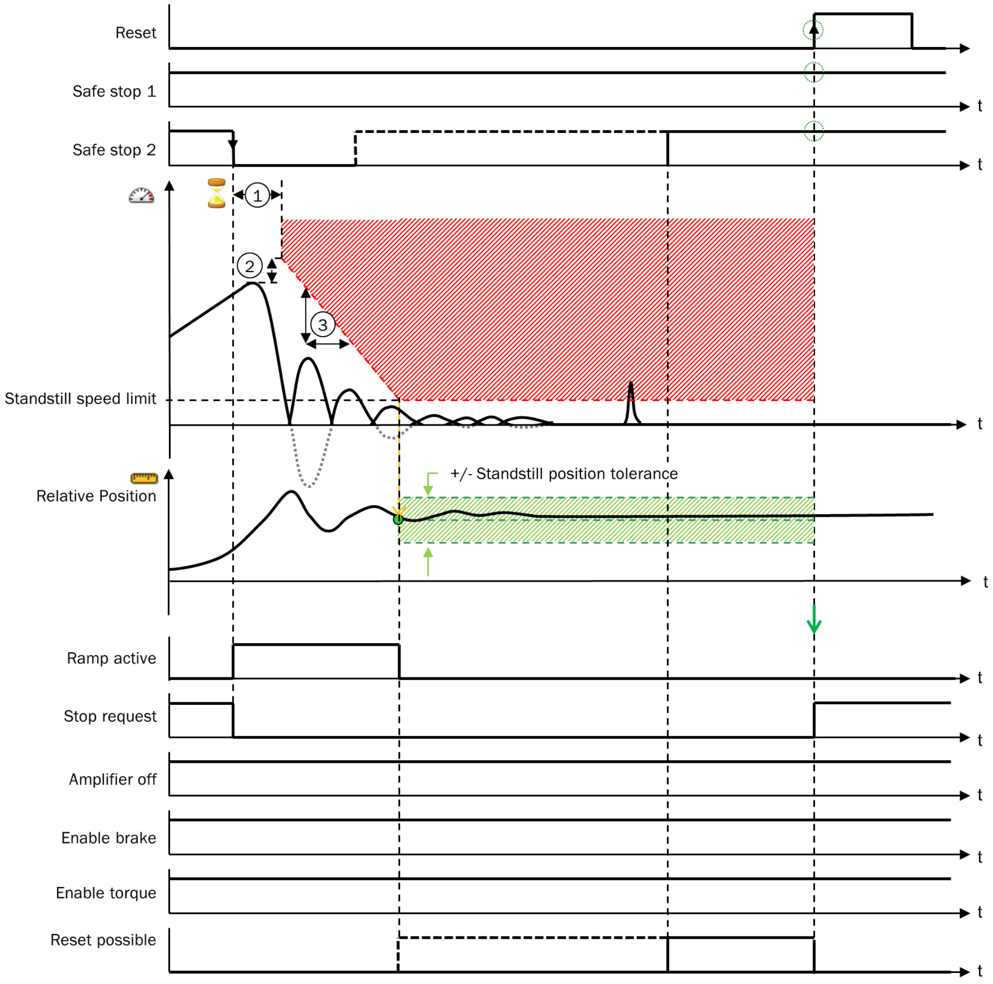


Figure 299: Monitoring function of Safe stop 2

① Delay time until start of ramp

② Stop ramp speed offset

③ Steepness of stop ramp 1 ... 4

The first three phases of Safe stop 2 are the same as the first three phases of Safe stop 1:

- "Phase 1: Wait for stop request", page 403
- "Phase 2: Delay time for beginning of stop ramp", page 404
- "Phase 3: Monitoring of the stop ramp", page 405

Phase 4 of Safe stop 2: Permanent standstill monitoring after Safe stop 2

Phase 4 begins when the stop ramp has ended and the drive has reached the standstill speed. With a Safe stop 2, the **Amplifier off**, **Enable brake**, and **Enable torque** output remain set to 1.

During phase 4, the speed and, if necessary, the relative position are monitored at the **Motion in** input. If neither the standstill monitoring with **Standstill speed** or the standstill monitoring with **Standstill position tolerance** are fulfilled or deactivated at any point during this process, phase 5 **Switch off torque** is triggered immediately. The optional configurable **Maximum speed filter distance** is taken into account.

If **Standstill position monitoring** is active, the current speed is no longer taken into account even if it is greater than the **Standstill speed**. This remains the case until the **Standstill position tolerance** is exceeded or the relative position becomes unreliable.

Phase 5 **Switch off torque** (Stop 0) is also triggered if the speed at the **Motion in** input becomes invalid (**Speed status** = 0).

If a falling signal edge occurs at any point before or during phase 4 of Safe stop 2 at either of the **Safe stop 1** inputs, phase 4 of Safe stop 1 (Temporary standstill monitoring after Safe stop 1) is triggered. This means that a Safe stop 1 always takes priority over a Safe stop 2.

Resetting of Safe stop 2 during phase 4

If the optional Reset input is used, a Safe stop 2 can be reset during phase 4 by a rising signal edge at the Reset input, if the following conditions are fulfilled:

- All of the used **Safe stop** inputs are 1.
- The speed at the **Motion In** input is valid and reliable.

If the optional **Reset** input is not used, a Safe stop 2 can only be reset by first triggering phase 5 and then ensuring that the conditions for resetting phase 5 are fulfilled.

Exceptions

If the normal sequence is not adhered to, the following exceptional cases may occur:

- If the speed exceeds the speed limit for the stop ramp, the **Amplifier off**, **Enable brake**, and **Enable torque** outputs are deactivated immediately. This is a Safe stop 0 or phase 5 with a Safe stop 1.

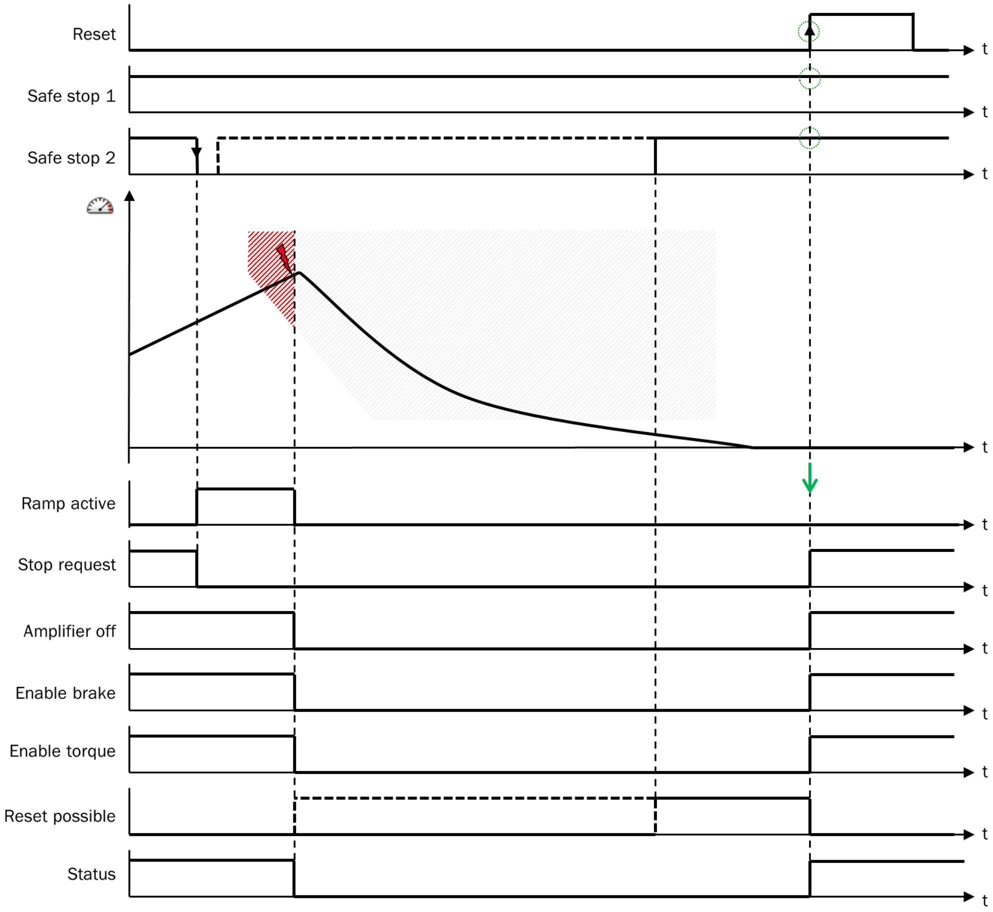


Figure 300: Exception – Stop ramp exceeded

- If the standstill condition is not or is no longer fulfilled at any point during standstill monitoring following a Safe stop 1 or Safe stop 2, the **Amplifier off**, **Enable brake**, and **Enable torque** outputs are immediately deactivated.

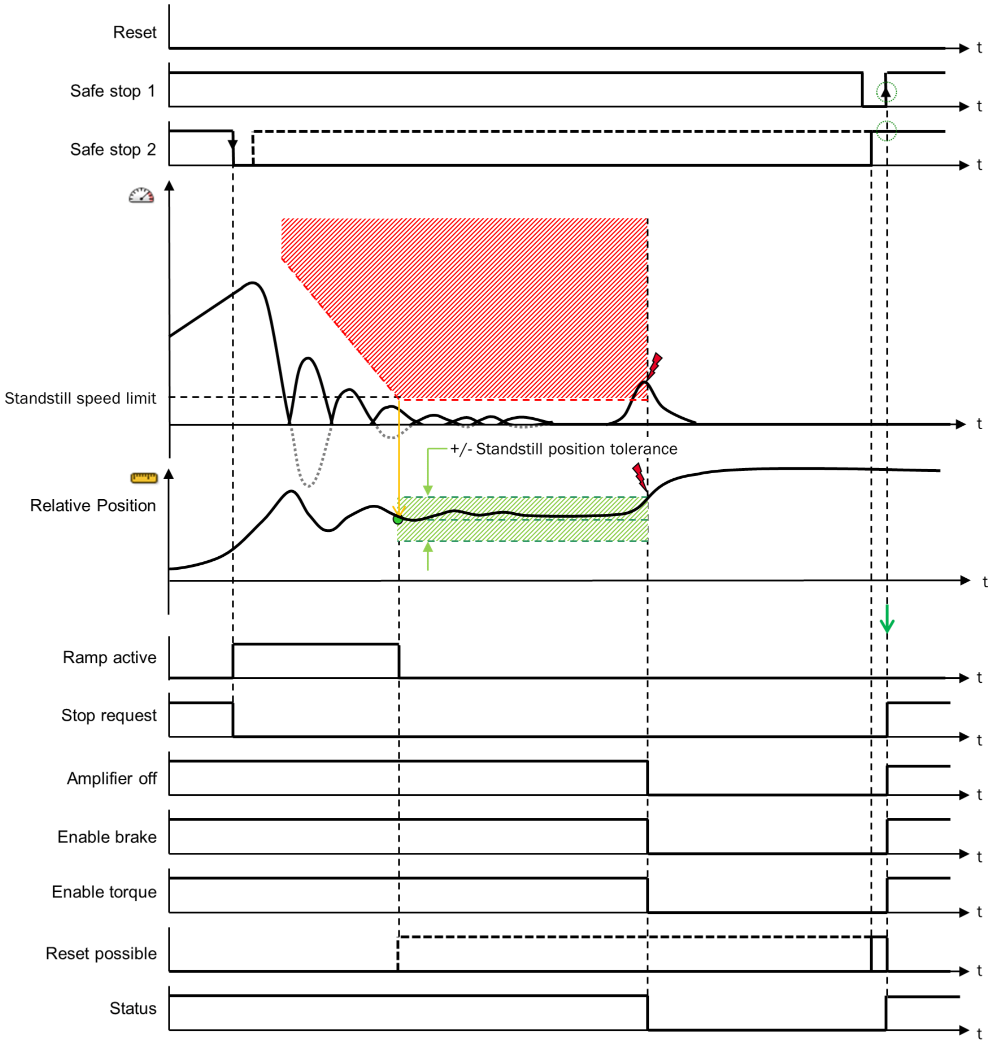


Figure 301: Exception – Standstill condition not met during standstill monitoring

11.11 Function blocks for data conversion

11.11.1 UI8 to Bool V1

Function block diagram

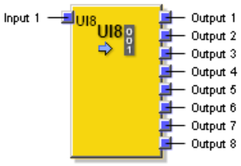


Figure 302: Inputs and outputs of the UI8 to Bool V1 function block

General description

The **UI8 to Bool V1** function block converts an 8-bit integer value (UINT8) at **Input 1** into Boolean. **Output 1** through **Output 8** provide the converted value in Boolean format. All this function does is convert the data type so that the data can be connected to Boolean signals.

Truth table for the UI8 to Bool V1 function block

Table 186: Truth table for the UI8 to Bool V1 function block

Input 1	Output 8	Output 7	Output 6	Output 5	Output 4	Output 3	Output 2	Output 1
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0
3	0	0	0	0	0	0	1	1
4	0	0	0	0	0	1	0	0
…	…	…	…	…	…	…	…	…
253	1	1	1	1	1	1	0	1
254	1	1	1	1	1	1	1	0
255	1	1	1	1	1	1	1	1

11.11.2 Bool to UI8 V1

Function block diagram

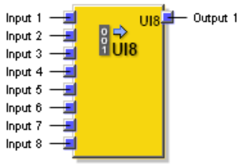


Figure 303: Inputs and outputs of the Bool to UI8 V1 function block

General description

The **Bool to UI8 V1** function block converts an 8-bit Boolean value at **Input 1** through **Input 8** into an integer value (UINT8). **Output 1** provides the converted value as an integer. All this function does is convert the data type so that the data can be connected to Boolean signals.

Truth table for the Bool to UI8 V1 function block

Table 187: Truth table for the Bool to UI8 V1 function block

Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1	Output 1
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	1	1	3
0	0	0	0	0	1	0	0	4
…	…	…	…	…	…	…	…	…
1	1	1	1	1	1	0	1	253
1	1	1	1	1	1	1	0	254
1	1	1	1	1	1	1	1	255

11.11.3 Motion status to Bool V2

Function block diagram

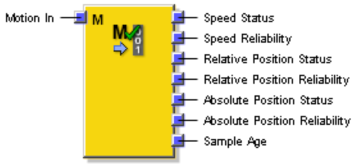


Figure 304: Inputs and outputs of the Motion Status to Bool V2 function block

General description

The **Motion Status to Bool V2** function block converts the relevant status (validity and reliability) for the speed, the relative position, and the absolute position and the update status at the **Motion in** input to Boolean values. All this function does is convert the data type so that the data can be connected to Boolean signals.

Outputs of the function block

Table 188: Outputs of the Motion status to Bool V2 function block

Output	Value	Meaning
Speed status	0	Invalid speed
	1	Valid speed
Speed reliability	0	Unreliable speed
	1	Reliable speed
Relative position status	0	Invalid relative position
	1	Valid relative position
Relative position reliability	0	Unreliable relative position
	1	Reliable relative position
Absolute position status	0	Invalid absolute position
	1	Valid absolute position
Absolute position reliability	0	Unreliable absolute position
	1	Reliable absolute position
Refresh status	0	Refresh status not current
	1	Refresh status current

11.11.4 Speed to Bool V2

Function block diagram

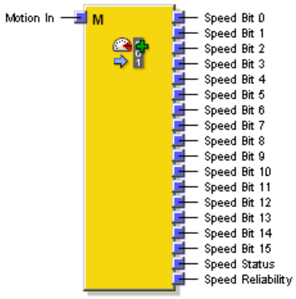


Figure 305: Inputs and outputs of the Speed to Bool V2 function block

General description

The **Speed to Bool V2** function block converts the speed value and the speed status at the **Motion in** input from data type Motion V2 into Boolean values. All this function does is convert the data type so that the data can be linked to Boolean signals, e.g., for the purpose of transmitting the speed value to the main module. The speed status is also output at the **Speed status** output, and the speed reliability is output at the **Speed reliability** output.

Function block parameters

Table 189: Speed to Bool V2 function block parameter

Parameter	Description	Value range
Number of speed bits	Number of Boolean outputs used for the speed	8 … 12

Speed bit x outputs

The **Speed bit 15** to **Speed bit 0** outputs correspond to the bits of the speed value at the **Motion In** input, in digits in the internal representation and with a sign in binary format.

Table 190: Speed bit x output of Speed to Bool V2 function block

Speed value in digits	Binary speed value (Speed bit 15 … 0 outputs)	Resolution for rotary movement type	Resolution for linear movement type
–32,767	1000 0000 0000 0001	1 digit = 0.5 rpm	1 digit = 1 mm/s
etc.	etc.		
–1	1111 1111 1111 1111		
0	0000 0000 0000 0000		
1	0000 0000 0000 0001		
etc.	etc.		
32,767	0111 1111 1111 1111		

Number of speed bits

If not all speed bits are required, the number of outputs used can be reduced using the **Number of speed bits** parameter. The function block checks whether the speed can be indicated by the used speed bits. If the speed exceeds the value that can be indicated, all outputs are switched to 0.

Table 191: Permitted speed values depending on the number of speed bits

Number of speed bits	Maximum possible output value (+/–) [digits]	Maximum permitted speed (+/–) at the Motion in input with rotary movement [rpm]	Maximum permitted speed (+/–) at the Motion in input with linear movement [mm/s]
16	32,767	16,383	32,767
15	16,383	8,191	16,383
14	8,191	4,095	8,191
13	4,095	2,047	4,095
12	2,047	1,023	2,047
11	1,023	511	1,023
10	511	255	511
9	255	127	255
8	127	63	127

NOTE
The **Speed bit 15** output represents the sign and is required to display negative values. This means that the outputs **Speed bit 14** to **Speed bit 7** are optional, depending on the **Number of speed bits**.

Speed status output

The value of the **Speed status** output corresponds to the speed status at the **Motion In** input.

Table 192: Speed status output in the Speed to Bool V2 function block

Value	Meaning
0	Invalid speed. **Or:** The speed at the **Motion In** input exceeds the value that can be indicated via the configured **Number of speed bits**.
1	Valid speed

Speed reliability output

The value of the **Speed reliability** output corresponds to the speed reliability at the **Motion In** input.

Table 193: Speed reliability output in the Speed to Bool V2 function block

Value	Meaning
0	Unreliable speed. **Or:** The speed at the **Motion in** input exceeds the value that can be indicated via the configured **Number of speed bits**.
1	Reliable speed

11.11.5 Speed to laser scanner V2

Function block diagram

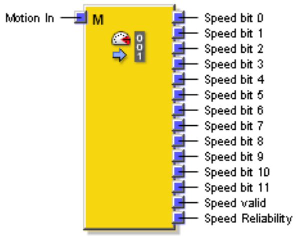


Figure 306: Inputs and outputs of the Speed to laser scanner V2 function block

General description

The **Speed to laser scanner V2** function block converts the speed at the **Motion in** input into a Boolean value with cm/s scaling. The 12 outputs from **Speed bit 11** to **Speed bit 0** are available for this purpose. The speed status is also output at the **Speed status** output, and the speed reliability is output at the **Speed reliability** output.

The converted speed value can, for example, be output to a SICK laser scanner connected via EFI so that the scanner can use it to switch over the monitoring case based on the speed.

ℹ **NOTE**
An encoder with linear movement must be connected to the **Motion in** input. Encoders with rotary movement cannot be used.

Function block parameters

Table 194: Speed to laser scanner V2 function block parameter

Parameter	Description	Value range
Number of speed bits	Number of Boolean speed outputs used	8 to 12

Speed bit x outputs

The **Speed bit 11** to **Speed bit 0** outputs correspond to the speed value at the **Motion In** input, converted into cm/s and in signed binary format.

Table 195: Speed bit x output of Speed to laser scanner V2 function block

Speed in cm/s	Binary speed value (Speed bit 11 … 0 outputs)
–2,048	1000 0000 0000
…	…
–1	1111 1111 1111
0	0000 0000 0000
1	0000 0000 0001
…	…
2,047	0111 1111 1111

Number of speed bits

If not all speed bits are required, the number of outputs used can be reduced using the **Number of speed bits** parameter. The function block checks whether the speed can be indicated by the used speed bits. If the speed exceeds the value that can be indicated, all outputs are switched to 0.

Table 196: Permitted speed values depending on the number of speed bits

Number of speed bits	Maximum possible output value (+/–) [cm/s]	Maximum permitted speed (+/–) at the Motion in input [digits]
12	2,047	20,470
11	1,023	10,230
10	511	5,110
9	255	2,550
8	127	1,270

ℹ **NOTE**
The **Speed bit 11** output represents the sign and is required to display negative values. This means that the outputs **Speed bit 10** to **Speed bit 7** are optional, depending on the **Number of speed bits** parameter.

Speed status output

The value of the **Speed status** output corresponds to the speed status at the **Motion In** input.

Table 197: Speed status output in the Speed to laser scanner V2 function block

Value	Meaning
0	Invalid speed. **Or:** The speed at the **Motion in** input exceeds the value that can be indicated via the configured **Number of speed bits.**
1	Valid speed

Speed reliability output

The value of the **Speed reliability** output corresponds to the speed reliability at the **Motion In** input.

Table 198: Speed reliability output in the Speed to laser scanner V2 function block

Value	Meaning
0	Unreliable speed. **Or:** The speed at the **Motion in** input exceeds the value that can be indicated via the configured **Number of speed bits.**
1	Reliable speed

11.12 Logical function blocks

11.12.1 NOT V1

Function block diagram

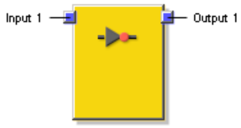


Figure 307: Inputs and outputs of the NOT V1 function block

General description

The value at the output is the inverted value of the input. If, for example, the input is set to 1, the output is set to 0.

Truth table

Table 199: Truth table for the NOT V1 function block

Input	Output
0	1
1	0

11.12.2 AND8 V1

Function block diagram

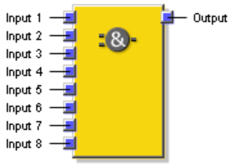


Figure 308: Inputs and outputs of the AND8 V1 function block

General description

The output is set to 1 when all the evaluated inputs are 1. Up to eight inputs are evaluated.

Each input can be individually inverted. At an inverted input, a 0 works in the same way as a 1 and vice versa.

Function block parameters

Table 200: Parameters of the AND8 V1 function block

Parameter	Possible values
Number of inputs	2 to 8
Invert input x	Every input of this function block can be inverted.

Truth tables

The following explanations apply to the truth tables in this section:

- “x” signifies “any” (0 or 1).

Table 201: Truth table for AND evaluation with two inputs without inversion

Input 1	Input 2	Output
0	x	0
x	0	0
1	1	1

Table 202: Truth table for AND evaluation with eight inputs without inversion

Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	Output
0	x	x	x	x	x	x	x	0
x	0	x	x	x	x	x	x	0
x	x	0	x	x	x	x	x	0
x	x	x	0	x	x	x	x	0
x	x	x	x	0	x	x	x	0
x	x	x	x	x	0	x	x	0
x	x	x	x	x	x	0	x	0
x	x	x	x	x	x	x	0	0
1	1	1	1	1	1	1	1	1

11.12.3 OR8 V1

Function block diagram

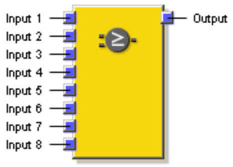


Figure 309: Inputs and outputs of the OR8 V1 function block

General description

The output is set to 1 when any of the evaluated inputs are 1. Up to eight inputs are evaluated.

Each input can be individually inverted. At an inverted input, a 0 works in the same way as a 1 and vice versa.

Function block parameters

Table 203: Parameters of the OR8 V1 function block

Parameter	Possible values
Number of inputs	2 to 8
Invert input x	Every input of this function block can be inverted.

Truth tables

The following explanations apply to the truth tables in this section:

- “x” signifies “any” (0 or 1).

Table 204: Truth table for OR evaluation with two inputs without inversion

Input 1	Input 2	Output
0	0	0
1	x	1
x	1	1

Table 205: Truth table for OR evaluation with eight inputs without inversion

Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	Output
0	0	0	0	0	0	0	0	0
1	x	x	x	x	x	x	x	1
x	1	x	x	x	x	x	x	1
x	x	1	x	x	x	x	x	1
x	x	x	1	x	x	x	x	1
x	x	x	x	1	x	x	x	1
x	x	x	x	x	1	x	x	1
x	x	x	x	x	x	1	x	1
x	x	x	x	x	x	x	1	1

12 Encoder in Motion Control FX3-MOC1

To configure an encoder that is connected to the Motion Control FX3-MOC1, select **Edit...** from the encoder's context menu or double-click on the encoder using the left mouse button. The **Element settings** window opens.

For additional information about connecting and configuring encoders, please see the operating instructions titled "Flexi Soft Modular Safety Controller Hardware".

12.1 Functions for all encoder types

The functions described here are available for all types of encoder.

12.1.1 General parameters of the encoder on the FX3-MOC1

Table 206: General parameters of the encoder on the FX3-MOC1

Parameter	Description
Scaling of the measurement system	see "Scaling of the measurement system", page 419
Counting direction	see "Encoder count direction", page 420
Encoder splitter box	see "Encoder connection type and ID code monitoring", page 420
Encoder voltage supply	see "Encoder connection type and ID code monitoring", page 420
Repress error message	see "Inhibit Error Message", page 421

12.1.2 Scaling of the measurement system

NOTE
The information in this section applies to all FX3-MOCx modules.

The scaling of the measurement system defines the ratio between the information from the encoder and the mechanically driven part (number of increments per revolution or per millimeter, depending on type of movement).

On the basis of this scaling, the information supplied by the encoder is converted so that the internal motion signal always has uniform mapping. This means the use in the logic independently of the measurement system scaling is possible.

The resolution of the calculated speed depends on the scaling of the measurement system; i.e., the resulting speed value is always a multiple of the speed resolution. The lower the resolution of the encoder system, the lower the speed resolution; i.e., the coarser the divisions. The calculated speed resolution should always be significantly less than the speeds configured in the function blocks.

NOTE
The scaling can be calculated directly in the configuration window taking into account a gear factor and a mechanical factor.

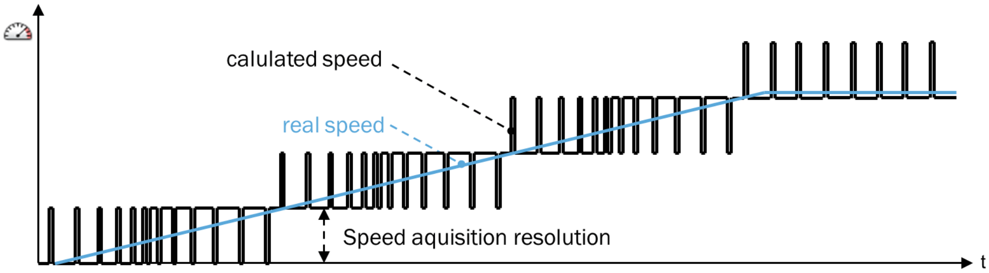


Figure 310: Resolution of the calculated speed as a function of the measurement system scaling

12.1.3 Encoder count direction

NOTE
The information in this section applies to all FX3-MOCx modules.

The count direction determines if the identified change in position is evaluated as positive (normal) or negative (inverted). This parameter can be used to adjust the count direction for encoders that count in the opposite direction due to their mounting position.

The definition of the signal sequence that applies for a normal count direction with A/B incremental encoders and with Sin/Cos encoders is provided in the technical data section for the Motion Control module in the operating instructions entitled “Flexi Soft Modular Safety Controller Hardware”.

12.1.4 Encoder connection type and ID code monitoring

NOTE
The information in this section applies to all FX3-MOCx modules.

Encoder voltage supply

The choice of voltage supply (**FX3-MOCx** or **External**) does not affect how the device works. Depending on what is selected, only the wiring example is adapted accordingly in the report.

Encoder connection type

The encoder connection type determines whether an encoder connection box is used for the encoder. Depending on what is selected here, ID code monitoring is either activated or deactivated for the encoder connection box. The wiring diagram is also adapted in the report.

ID code monitoring

Each encoder connection box contains an ID code, along with the outputs for the encoder voltage supply that comes from the FX3-MOCx module (ENC1_24V or ENC2_24V). If a connection type involving at least one encoder connection box (e.g., FX3-EBX1, FX3-EBX3 or FX3-EBX4) is selected in the configuration, the FX3-MOCx module checks this ID code on a cyclical basis.

To do this, the FX3-MOCx module switches the supply on and off at ENC1_24V and ENC2_24V alternately at 4 msintervals. The encoder remains oblivious to this because the supply voltages are combined via diodes. The ID code of the encoder connection box is then measured using the switched off supply. If the ID code measurement function detects an invalid value, the status bits are set to Invalid in the Motion data of the associated encoder. This happens when ENC1_24V / ENC2_24V or the shared 0 V voltage supply (ENC_0V) is interrupted between the FX3-MOCx module and the encoder connection box.

The status bits become valid again when the following conditions are met for at least the duration of the **Error recovery time** [9] without interruption:

- The ID code monitoring function detects valid values.
- Any other possible checks likewise provide a positive result.

With the help of the ID code monitoring function, breaks in the shared 0 V ENC_0V voltage supply or in the shared connecting cable that runs between the FX3-MOCx module and the encoder connection box can be detected.

12.1.5 Inhibit Error Message

The error reset time for the status bits is set to 1 second by default. Using the **Repress error signal** input, the error reset time can be reduced to 0.14 seconds.

The error reset time is accurate to +/– 20 ms.

The input is always evaluated after 0.14 s has elapsed. If the input is set to 1 at this time, the error reset time has expired. If it is set to 0 at this time, the normal error reset time is 1 s.

The **Repress error signal** input can also be used to repress error messages from the encoder:

- Entry in the diagnostics history
- Error messages in the module status bits (data set 3 of the gateways)
- Display of the LED MS on the FX3-MOC1

This may be useful in certain operating situations where faults or detection gaps are expected in order to minimize the impact of the fault.

The error response, particularly the change of the status bits in the **Motion out** output, is not affected by the **Repress error signal** input.

Example applications:

- Overhead conveyors with code band for position detection: At locations where a code gap is expected (e.g., at a diverter), activate the **Repress error signal** input.
- Production phases with very high potential for faults (e.g., welding processes)

If safety is guaranteed through other measures (e.g., a closed safety door), then the **Repress motion bits response** input in the Safe stop V2 function block can be used to repress the error response at the end of the signal chain.

NOTE

- The **Repress error signal** input for the encoder appears in the Logic editor of the FX3-MOC1-under “Outputs” for the corresponding FX3-MOC1.
- If the **Repress error signal** input of an encoder is connected to an output of a function block in the FX3-MOC1 module (rather than to a bit that originates from the main module), the input will be delayed in its response by one logic cycle, because it must evaluate the result of the function block from the previous cycle.

12.2 A/B incremental encoder

With this type of encoder, there are no specific parameters and monitoring functions. To achieve the desired level of safety function blocks can be included in the FX3-MOCx logic for the purpose of checking the information provided by the encoder (motion data) (see "Logic programming in Motion Control FX3-MOC1", page 316).

[9] The **Error recovery time** is 1 s in the FX3-MOC0, and 0.14 s or 1 s in the FX3-MOC1, depending on the configuration.

12.3 Sin/Cos encoders

12.3.1 Special parameters for Sin/Cos encoders

Table 207: Special parameters for Sin/Cos encoders

Parameter	Description
Sin/Cos analog voltage monitoring	see "Sin/Cos analog voltage monitoring", page 422
Increased resolution	see "Sin/Cos resolution enhancement", page 430

12.3.2 Sin/Cos analog voltage monitoring

This function is used to identify errors in the encoder system. This can be particularly helpful in the case of applications where an axis is to be monitored using just one Sin/Cos encoder. When Sin/Cos analog voltage monitoring is enabled, the system checks whether the relationship between the sine and cosine voltage is as required.

If the Sin/Cos analog voltage monitoring function detects invalid voltage conditions, the reliability bits are set to unreliable in the Motion data of the associated encoder.

The status bits become valid again when the following conditions are met without interruption for at least the duration of the Error recovery time [10]:

- The Sin/Cos analog voltage monitoring function detects valid voltage conditions.
- Any other possible checks likewise deliver positive results.

WARNING
Using unsuitable encoders

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

- ► Only use suitable encoders.
- ► Make sure that the encoder characteristics specified by the manufacturer will continue to apply to further deliveries or that you will be kept informed of any changes.
- ► Make sure that all errors to be considered are detected or can be prevented.

NOTE
IEC 61800-5-2 offers possible support for the errors to be considered.

The following must typically be obtained from the encoder manufacturer for this purpose:

- An implementation manual containing specific usage requirements for achieving a particular level of safety

or

- Information on the encoder design and the effects of errors on the Sin/Cos signals

During Sin/Cos analog voltage monitoring, the relationship between the sine and cosine voltages is checked based on two criteria:

- Vector length
- Signal deviation

Monitoring of the vector length

If the ideal sine and cosine voltage values are transferred to an XY coordinate system, they describe a circle. Mathematically, the radius of the circle (vector length) is calculated from √ (sine^2 + cosine^2).

[10] The **Error recovery time** is 1 s for the FX3-MOC0 and 0.14 s or 1 s for the FX3-MOC1 depending on configuration.

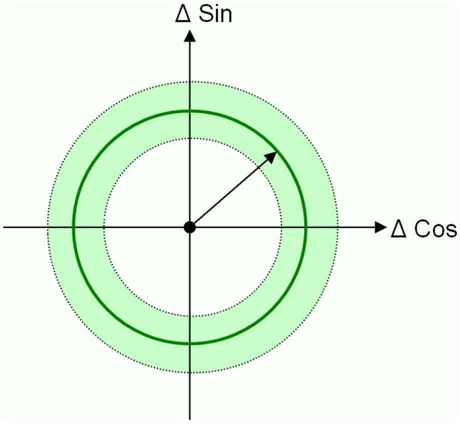


Figure 311: Monitoring of the vector length

This monitoring criterion is used to check whether the vector length is within the anticipated tolerance band. The specific limits that apply to this monitoring function are provided in the technical data section for the Motion Control module in the operating instructions entitled “Flexi Soft Modular Safety Controller Hardware”.

Monitoring of the signal deviation

This monitoring criterion is used to check whether the sine signal exhibits the anticipated signal deviation when the cosine signal has changed by at least the minimum vector length that is expected. In the same way, the signal deviation of the cosine signal is checked when the sine signal has changed.

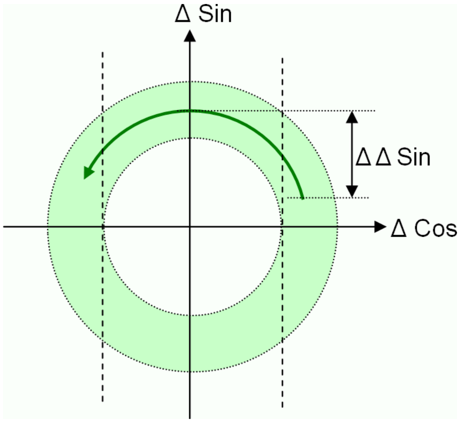


Figure 312: Monitoring of the signal deviation

This monitoring criterion even enables fault patterns to be detected in cases where either the sine signal or the cosine signal is affected by a stuck-at fault, but the resulting signal is still within the tolerance band (green circle) meaning that the fault cannot be detected by monitoring the vector length (see the second example in the list of possible fault patterns).

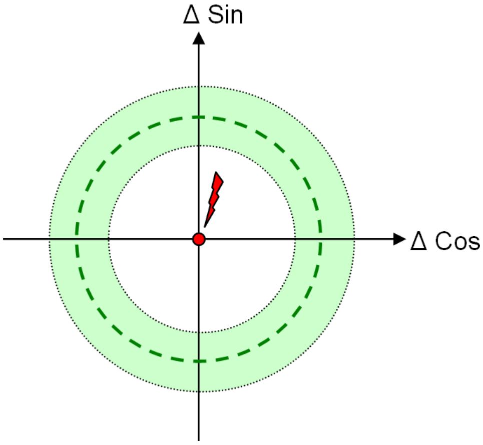
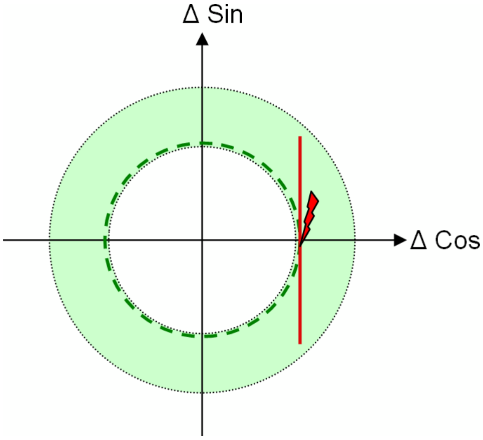
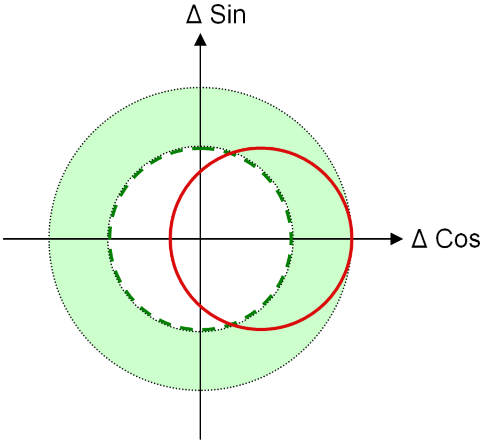
Example fault patterns

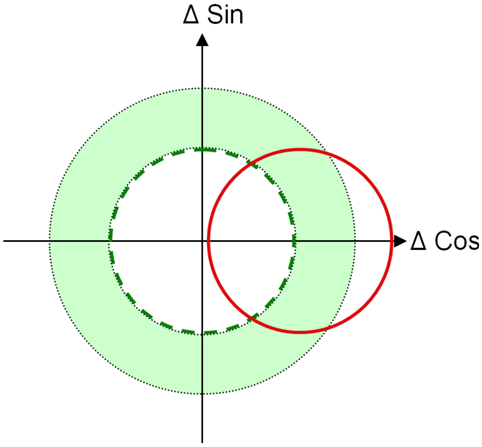
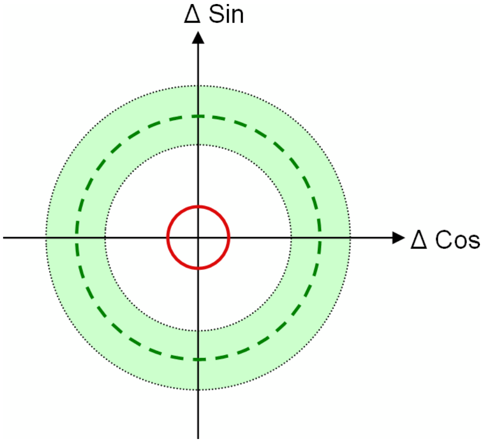
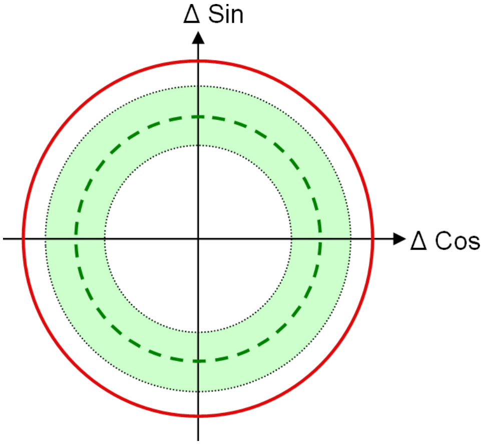
The following table shows some example fault patterns where the relationship between the sine and cosine voltage is not as required. Here ...

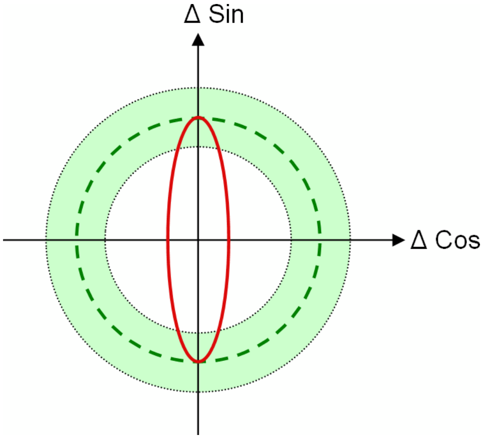
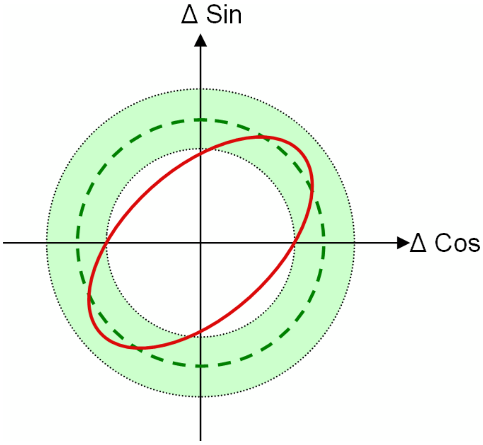
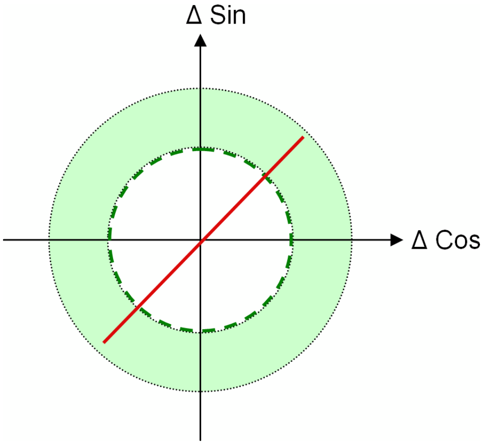
ΔSin = differential voltage between Sin+ and Sin– on Motion Control module

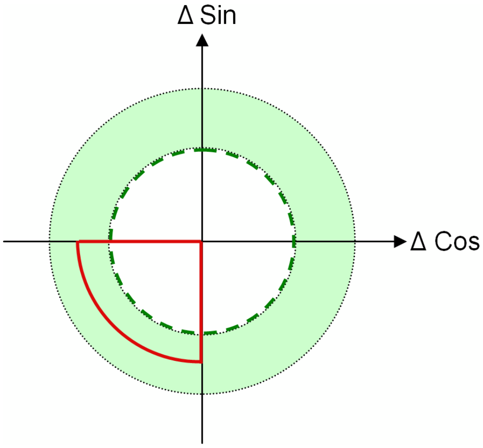
ΔCos = differential voltage between Cos+ and Cos– on Motion Control module

Table 208: Possible fault patterns during Sin/Cos analog voltage monitoring

Fault pattern	Possible causes of faults
	- Break in the encoder connection - No light emitted by transmitting diode - Internal encoder voltage supply is faulty
	- Stuck-at fault affecting sine or cosine signal
	- Interruption of or change in the Sin_Ref or Cos_Ref voltage

Fault pattern	Possible causes of faults
	- Interruption of or change in the Sin_Ref or Cos_Ref voltage
	- Internal encoder voltage supply is too low - Not enough light emitted by transmitting diode
	- Too much light emitted by transmitting diode

Fault pattern	Possible causes of faults
	- Sine or cosine gain factor is too low, e.g. due to change in resistance
	- Increased filter time due to increase in resistance - Cross-circuit between Sin+ and Cos+ - Cross-circuit between Sin– and Cos–
	- Cross-circuit between Sin and Cos in the case of encoders with Sin_Ref and Cos_Ref

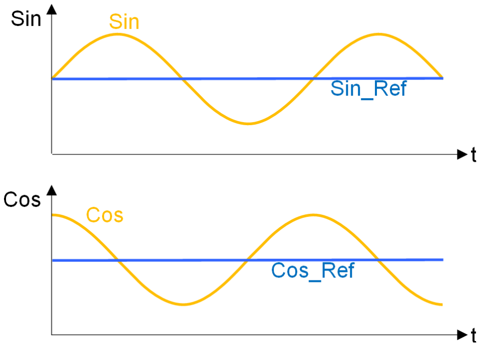
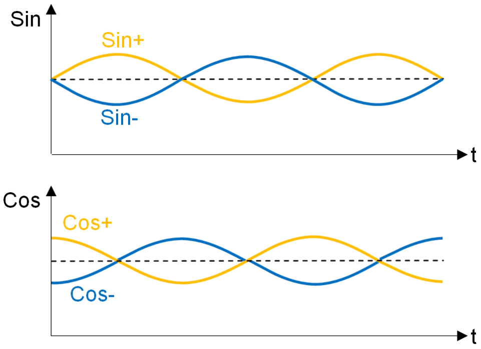
Fault pattern	Possible causes of faults
	- Change in the internal encoder reference voltage source for Sin_Ref and Cos_Ref with the result that the analog output stage of the encoder approaches the saturation limit and half-waves are partially or fully clipped.

12.3.3 Limits of Sin/Cos analog voltage monitoring

This section covers all applications with Sin/Cos encoders when the following conditions apply:

- A separate encoder is used to monitor each axis.
and
- Sin/Cos encoders with Sin_Ref and Cos_Ref output signals are used.

Table 209: Examples of Sin/Cos encoder signals

Sin/Cos encoder signals	Examples for encoder
 Sin_Ref and Cos_Ref are DC voltages, typically 2.5 V DC	- SKS36S - SKM36S **Note:** If just one encoder of this kind is going to be used to monitor an axis, supplementary error control measures are required; e.g., shared use of the encoder signals for electronic commutation of the drive system.
	- DFS60S Pro **Note:** Encoders of this kind do not require any of the supplementary fault control measures described here.

Supplementary fault control measures

If the final two examples from the list of possible fault patterns shown in table 144 cannot be completely ruled out for the encoders that are being used, supplementary measures must be implemented to control these faults.

This is necessary because the values might only leave the tolerance band (green circle) briefly in the case of certain faults and the FX3-MOCx module might not be able to detect this in the event of high signal frequencies. If this happens, there is no guarantee that the FX3-MOCx module will be able to determine the speed or relative position correctly.

The following options are available for supplementary fault control:

- Fault detection by means of additional plausibility checks
- Shared encoder signals for electronic commutation of the drive system and fault detection based on safe status within the process

Error detection by means of additional plausibility checks

Another signal from the process can be evaluated in combination with the logic of the Motion Control module and main module in order to check the plausibility of the encoder motion signal. For example, a signal that evaluates the status of the drive can be used for this purpose (drive moving/drive not moving).

Shared use of encoder signals for electronic commutation

⚠ **WARNING**

Changes in the drive system

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

► Have the manufacturer confirm the relevant properties of the drive system.

► Check if changes in the drive system (e.g. due to product updates or reconfiguration) affect the common use of the encoder signals for electronic commutation.

If the encoder is used for the Motion Control module and also for drive control, you can verify that the specified fault patterns reliably lead to a safe drive status (e.g., standstill or reduced torque). This is possible if one of the basic functional requirements of the drive system is that the encoder must determine the pole positions correctly in order for the rotary field to be generated, and if stationary commutation also results in a drive system standstill (synchronous drive).

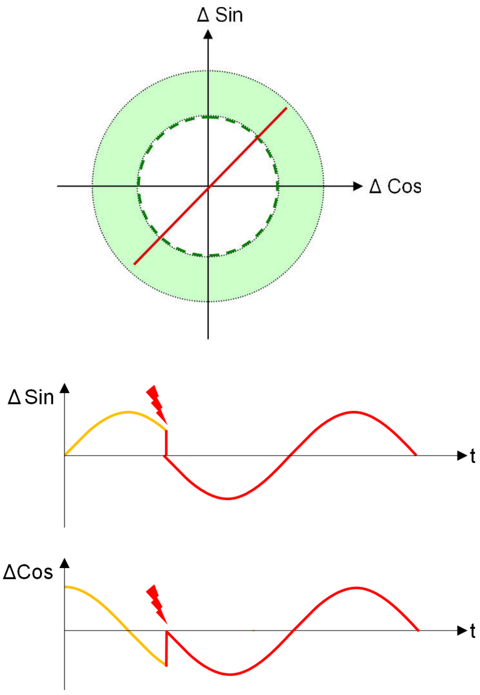
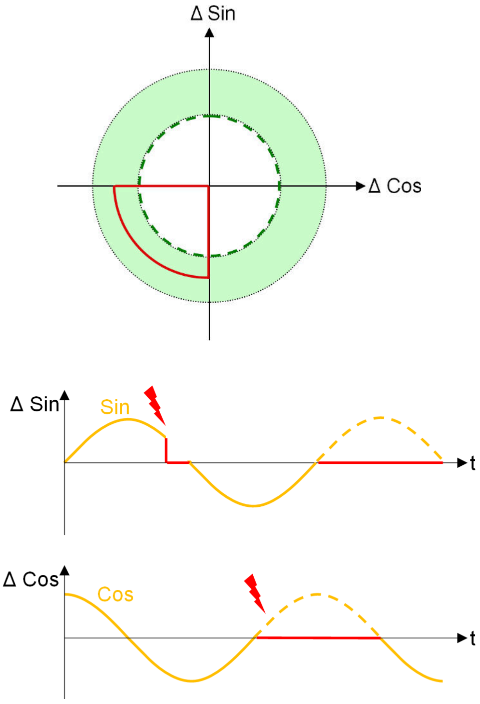
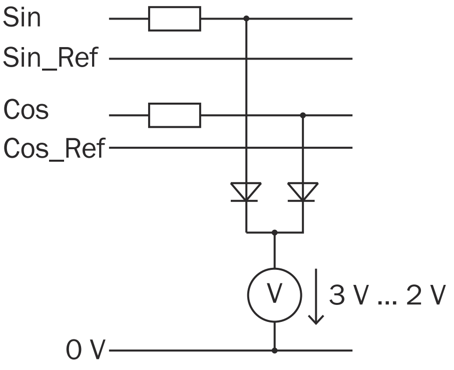
In the case of encoders with Sin/Sin_Ref and Cos/Cos_Ref (Sin_Ref and Cos_Ref are DC voltages, typically 2.5 V DC), the encoder signals for electronic commutation of the drive system have to be shared. In this case, the polarity position is coupled directly and electronically with the current vector requirement for the three-phase rotary field. It is therefore assumed that if the commutation is stationary, the drive system will also be at a standstill.

ℹ **NOTE**

In encoders with Sin+/Sin– and Cos+/Cos– (Sin– and Cos– are inverted voltages of Sin+ and Cos+), there is **no** requirement to share encoder signals for the electronic commutation of the drive system.

The following table shows how the relevant error patterns can be simulated in order to check the effect on the drive system.

Table 210: Simulating error patterns for Sin/Cos encoder signals

Error pattern	Error simulation
 	► Insert a series resistance of approximately 100 Ω in the sine signal line and in the cosine signal line running from the encoder to the drive system. The purpose of this is to prevent damage to the encoder. ► To activate simulation of the error, establish a connection (cross-circuit) between the sine and cosine signals.
 	► Insert a series resistance of approximately 100 Ω in the sine signal line and in the cosine signal line running from the encoder to the drive system. The purpose of this is to prevent damage to the encoder. ► Connect diodes and a voltage regulator. Adjust the voltage regulator in line with the peak sine and cosine output voltage (typically 3 V).  ► To activate simulation of the error, reduce the voltage of the voltage regulator until the anticipated error pattern emerges (typically at around 2 V).

For this check, we recommend the following procedure:

- ► Install the circuit components for simulating the error but do not activate them.
- ► Check that the drive system is functioning correctly. The purpose of this is to verify whether simply installing the circuit components for error simulation without activating them is sufficient to bring about a safe status.

- ► Activate error simulation.
- ► Check the anticipated error pattern (by measuring with an oscilloscope).
- ► Check the anticipated effect on the drive system (safe status).

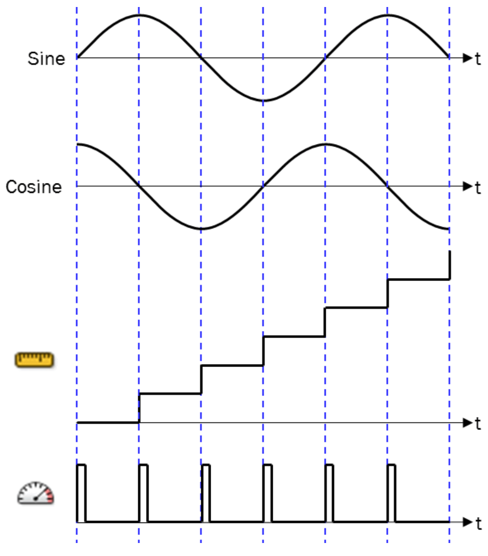
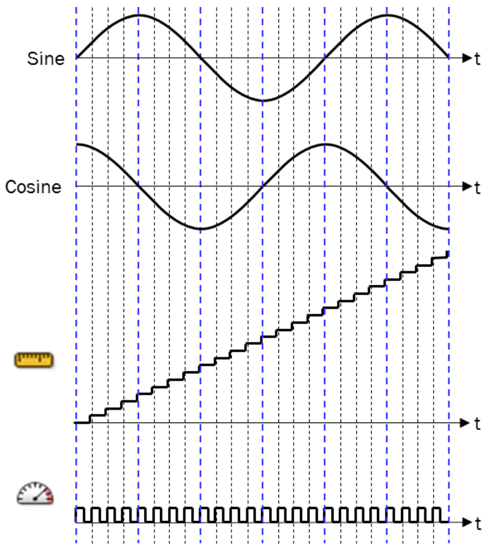
12.3.4 Sin/Cos resolution enhancement

NOTE
The information in this section applies to all FX3-MOCx modules.

This function is available for Sin/Cos encoders and is relevant for Sin/Cos encoder systems with a low resolution that can produce wider result stepping for speed detection. When resolution enhancement is activated, the number of counting points is increased by a factor of four, thereby improving the resolution of speed detection.

NOTE
This option has no effect on position formation (relative position value). The position shown in the diagram is an internal value used solely for calculating the speed.

Table 211: Sin/Cos resolution enhancement

Without resolution enhancement (deactivated)	With resolution enhancement (activated)
	

If – before Sin/Cos resolution enhancement is applied – the speed detection resolution is already less than or equal to the speed value mapped internally in the **Motion** data type (1 digit = 0.5 rpm or 1 mm/s), this option has no effect even when activated.

12.4 SSI encoder

The functions described here are available for SSI encoders (SSI master, SSI listener).

12.4.1 Special parameters for SSI encoders

Table 212: Special parameters for SSI encoders

Parameter	Description	Possible values
Data transmission rate	Data transmission rate for the clock output as SSI master	• 0 = listener • 100 ... 1,000 kBaud
Number of bits for the entire SSI protocol frame	Number of clock cycles for a single transmission	8 ... 62

Parameter	Description	Possible values
Number of leading bits	Number of leading bits that do not contain position data	0 … 54
Number of position data bits	Number of bits containing the relevant position data bits	8 … 32
Double data transmission	For selecting whether the position value should be transmitted once or twice using an SSI protocol frame	• Transmission of a single position value • Duplicate transmission of the position value
Number of bits between the position data bits	Only available with duplicate transmission of the position value	0 … 30
Data encoding	Data encoding for the position data bits	• Binary • Gray
Error bit evaluation	Monitoring of error bits supplied by the encoder in the SSI protocol frame. For each individual bit, you can specify whether the error status should be represented by a 1 or a 0.	**For each bit that is not a position data bit** • 1 = error • 0 = error
Max. data reception interval	Maximum time within which valid position data is expected to arrive	4 … 100 ms
Maximum speed jump	Enables the filtering of faulty SSI telegrams	• Disabled • 1 … 32,767 digit_speed o 0.5 … 16,383 rpm o 1 … 32,767 mm/s
Maximum position jump	For controlling system-related position jumps	• Disabled • 1 … 500,000 digit_position o max. 16.6 rpm o max. 2,000 mm
Value range for encoder increments	For adjusting to the encoder value range	• Full range • 10 … $2^{\text{Number of position data bits}} - 1$
Position type	Type of position in the SSI telegram	• Relative • Absolute
Periodic position length	Value for overflow with periodic position	• Disabled • 1 … 2^{30} digit_position (half the position range)
Position origin	Adapt the position origin value if the absolute position range extends into the negative range	• If **Periodic position** active (> 0): (1 – Length of periodic position) … 0 • Otherwise: Full position range
Teach position	For teaching an original position	Position value range of the encoder

12.4.2 Double data transmission

Overview

The information in this section applies to all FX3-MOCx modules.

Double data transmission

Certain SSI encoders support multiple transmission of the position data. This means that the same encoder data is output again, provided that the clock gap between the data packages (monoflop time) is not exceeded. This makes it possible to detect data that has been corrupted by transmission faults, for example.

The FX3-MOCx module supports duplicate transmission of the position data. When “Transmit position data twice” is activated, the FX3-MOCx module checks whether the two position data values within the received SSI protocol frames are identical. If they are not identical, the position data within this SSI protocol frame is ignored. All other SSI telegrams arriving in the same 4 ms logic cycle of the FX3-MOCx are also ignored.

You can find Information on how motion data from the associated encoder is affected in the “Maximum data reception interval” section.

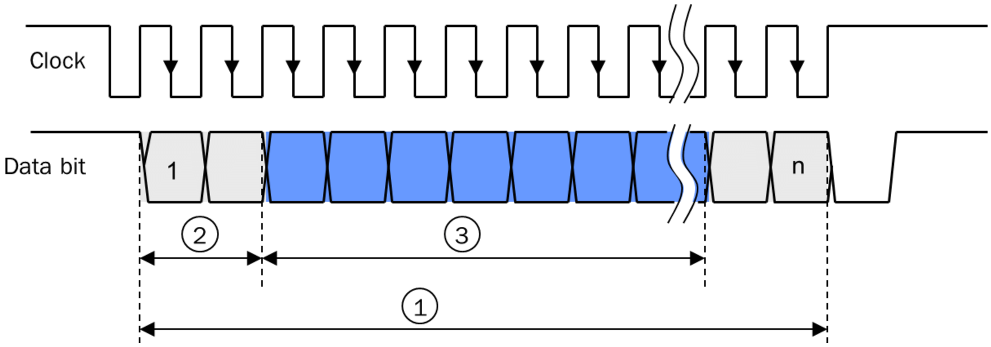


Figure 313: Transmission of a single position value

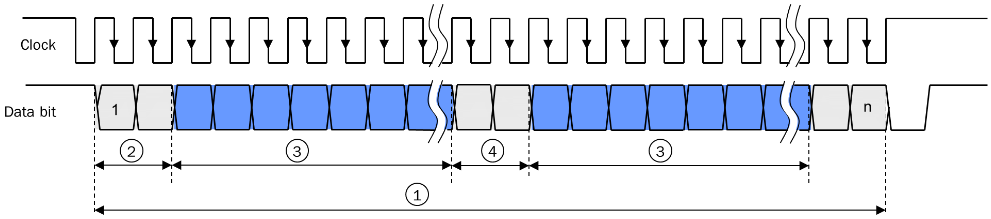


Figure 314: Duplicate transmission of the position value

① Number of bits for the entire SSI protocol frame
② Number of leading bits
③ Number of position data bits
④ Number of bits between the position data bits

12.4.3 Error bit evaluation

NOTE
The information in this section applies to all FX3-MOCx modules.

As well as including the position data bits in the SSI protocol frame, certain SSI encoders also transmit error bits that represent the results of internal monitoring functions performed by the encoder. Such error bits can be evaluated with the FX3-MOCx. For each individual bit, you can specify whether the error status should be represented by a 1 or a 0. If an error status is detected for one or more of the selected error bits, the position data within this SSI protocol frame is ignored.

12.4.4 Filtering of speed jumps

This function is used to filter invalid SSI telegrams that signal changes in speed which cannot occur in the application due to faults.

The value for the **Maximum speed jump** parameter must therefore be higher than the change in speed that is actually possible in the application within the **Max. data reception interval**.

The **Filter speed jumps** function should not be confused with the **Monitoring maximum speed jump** function in the FX3-MOC0 V1, which triggers an error response when the set value is exceeded. The **Filter speed jumps** function, on the other hand, is used to ignore invalid SSI telegrams. All other SSI telegrams arriving in the same 4 ms logic cycle of the FX3-MOCx are also ignored.

The **Max. data reception interval** parameter must be set to at least 8 ms so that SSI telegrams can initially be ignored without an error response.

12.4.5 Maximum position jump monitoring

This function is used to control system-related position jumps.

Example application: Overhead conveyors with a code band for position detection

This function can be used in locations where a code jump is expected (e.g. at a diverter or at the meeting point between the end and the start of the code band).

In these areas, the sensor normally reports the position without special actions via the SSI interface. In the FX3-MOC1, this leads to high speed values because the speed is calculated from the position difference per time interval. These increased speed values can lead in turn to an unwanted speed monitoring reaction.

This can be prevented by using the function **Maximum position jump monitoring** to selectively trigger an error response, thereby forcing a reinitialization of the SSI encoder evaluation in the new position range.

If the FX3-MOC1 detects higher position jumps, the SSI telegrams are ignored until the position change – relative to the last valid (not rejected) position – returns within the range defined by the **Maximum position jump** parameter.

Ignoring the SSI telegrams starts the timer for monitoring the **Max. data reception interval**. If the **Max. data reception interval** parameter is set to a value greater than 4 ms, the (desired) error response is delayed accordingly. After the **Max. data reception interval** has expired, the status bits are set to invalid in the Motion data of the associated encoder.

The status bits become valid again when the following conditions are met without interruption for at least the duration of the Error recovery time [11]:

- The position jump between the current and the previously received position value is lower than the **Maximum position jump** parameter.
- Any other checks also deliver positive results.

In this context, it makes sense to use the following functions:

- **Inhibit Error Message** input of the SSI encoder to shorten the duration of the error response.
- **Inhibit error reaction** input of the **Safe Stop V2** function block in order to inhibit the error response at the end of the signal chain if safety is ensured by other measures.

NOTE

The position jump test is valid in both the **Relative position** and **Absolute position** modes. In this case, overflows in the relative position value range for the Motion V2 data are irrelevant.

[11] The **Error recovery time** is 1 s for the FX3-MOC0 and 0.14 s or 1 s for the FX3-MOC1 depending on configuration.

12.4.6 Maximum data reception interval

This function enables the system to tolerate invalid position data temporarily by relying on the most recently valid position data in the meantime. The results of all relevant monitoring functions must remain valid for longer than the **Max. data reception interval** at least once. If not, the status bits are set to invalid in the motion data of the associated encoder.

In the case of the SSI encoder, the following monitoring functions affect the **Max. data reception interval:**

- SSI protocol frame not received or not received in full (only applies to SSI listener)
- Double data transmission
- Error bit evaluation
- Filtering of speed jumps
- Maximum position jump monitoring

The status bits become valid again if all monitoring functions are met without interruption for at least the duration of the **Error recovery time** [12]:

NOTE
In SSI listener mode, only ever one SSI protocol frame is evaluated within the 4 ms cycle. Further SSI protocol frames transmitted within the same cycle are not evaluated.

12.4.7 Position range

Position type

The Position type parameter specifies whether the position in the SSI telegram is a relative position or absolute position.

- A relative position means that the traveled path can be reproduced, but the mechanical position is not clear. This is mainly due to the fact that the relative position start value in the Motion V2 data of the encoder always starts with 0, regardless of the mechanical position.
  As long as there is no error status, only the relative position values and speed values in the SSI encoder's Motion data are valid. The absolute position values are always invalid.
- An absolute position means that the position value is clear for any possible mechanical position in the application. This also applies after the measuring system has been restarted.
  As long as there is no error status, the absolute position values including the relative position values, and the speed values in the SSI encoder's Motion data are valid.

If the **Absolute** setting is selected for the **Position type**, then the system checks whether the absolute position range has been exceeded; i.e., if there has been an overflow. An overflow occurs when the position has jumped by half the position range or more.

If an overflow of the absolute position range is detected, the SSI telegrams are ignored until the position change is not in an overflow state, based on the last valid (not disregarded) position.

Ignoring the SSI telegrams starts the timer for monitoring the **Max. data reception interval.** If the **Max. data reception interval** parameter is set to a value greater than 4 ms, the error response is delayed accordingly. After the **Max. data reception interval** has expired, the status bits are set to invalid in the Motion data of the associated encoder.

[12] The **Error recovery time** is 0.14 s or 1 s depending on configuration.

The status bits become valid again if the following conditions are met without interruption for at least the duration of the Error recovery time [12]:

- There was no overflow between the current and the previously received position value.
- Any other possible monitoring functions likewise deliver positive results.

If the **Absolute** setting is selected for the **Position type**, additional options are available to determine the absolute position range:

Value range for encoder increments

The **Value range for encoder increments** parameter defines the value range for the encoder's position if the possible value range is not being used fully according to the number of position data bits (**Position Data Width** parameter in the SSI settings).

For example, at 14 position data bits, the maximum possible value range is 16,384 increments. However, if the encoder has a value range of 10,000 increments, then this can be configured accordingly using the **Encoder increments value range** parameter, to ensure that the speed calculations are accurate in the event of an overflow of the value range.

Position origin

The **Position origin** parameter can be used to adapt the position origin value if the absolute position range also extends into the negative range.

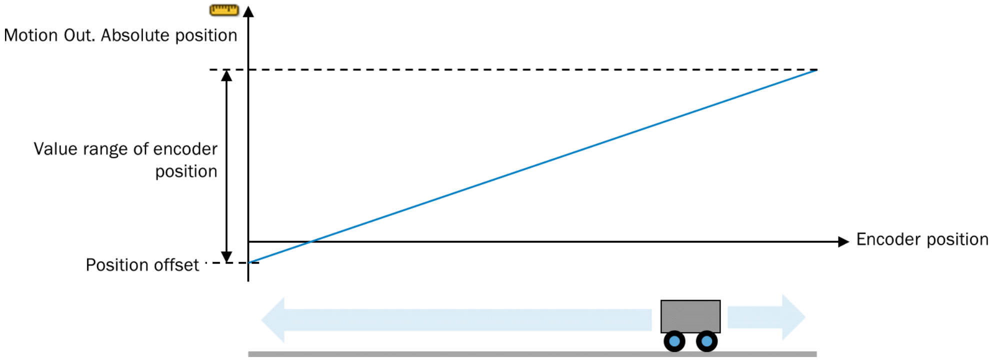

Motion Out. Absolute position
Value range of encoder position
Position offset
Encoder position

Figure 315: Position origin

Periodic position length

The **Periodic position length** parameter is intended for applications in which the mechanical position is a periodic position, i.e., where the start and end of the position range are located at the same point. This situation may occur, for example, with round table or mechanical presses, where the mechanical position repeats every 360°.

When the periodic position length is reached, the MOC continues counting using the origin value (position origin).

The periodic position length must be a whole number divisor of the encoder position range. The encoder position range is displayed in the **SSI settings** area.

Example:

Encoder position range: 819,200 mm

Periodic position length: 200 mm, 400 mm, 600 mm

819,200 mm / 200 mm = **4096** ✓ Valid
819,200 mm / 400 mm = **2048** ✓ Valid
819,200 mm / 600 mm = **1,365.34** Invalid

[12] The **Error recovery time** is 0.14 s or 1 s depending on configuration.

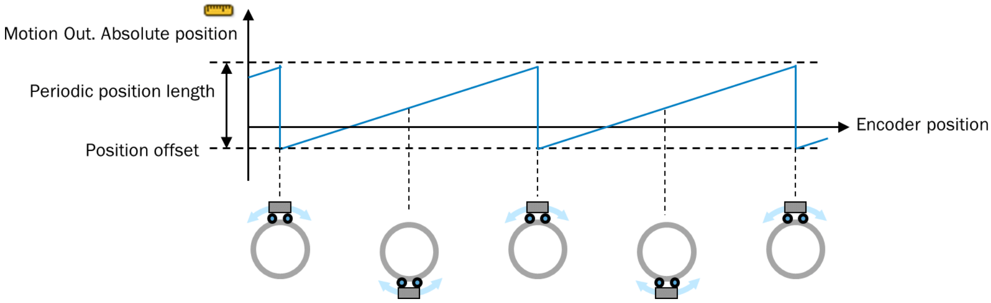


Figure 316: Periodic position length

12.4.8 Teach position

The **Teach position** function is used to electronically calibrate the encoder position. This is necessary, for example, in applications with rotary encoders; in such applications, the mechanical installation position of the encoder is not calibrated or cannot easily be calibrated to a high degree of precision.

Using the **Teach position** function, the required position offset can be taught in (Teach), to achieve the required position value at the **Motion Out** output in the installed position.

The **Reference position** parameter indicates the desired position value. In the event of a rising signal edge at the **Teach** input, the FX3-MOC1 calculates the required position offset and stores it in the FX3-MOC1 (EEPROM). In order for the teach-in process to be successful, the SSI encoder must receive valid data. This means that the validity status of the relative position at the **Motion Out** output of this encoder is 1 (valid).

Once the teach process has been completed successfully, the saved offset value is always applied, including at the next system start (transition of the Flexi Soft system to Run status).

After a rising signal edge at the **Teach** input, the absolute position at the **Motion Out** output becomes invalid. If the teach-in process was successful, it becomes valid again after 1.5 s at the latest.

If the **Teach** input is in use and the configuration of the SSI encoder is changed, then the teach process must be carried out again to ensure that a valid absolute position is maintained at the **Motion Out** output. This includes the first-time activation of the input.

Referencing accuracy

In order to ensure that the referencing process is accurate, the various signal propagation delays of the SSI telegrams and the **Teach** input must be taken into account. If the speed is not 0, the mechanical position may change as a result of the different signal propagation delays before the rising signal edge at the **Teach** input becomes effective.

13 Analog input module FX3-ANA0

Description

The FX3-ANA0 analog input module is used for monitoring an analog process variable with one or two sensors.

The valuesof the two input channels are continuously compared in order to check the plausibility. The two channels must not deviate from one another by more than one configurable discrepancy value. If the permissible difference is exceeded, a sensor error is generated and the **enable** bit is set to 0.

After the plausibility check, the FX3-ANA0 checks whether the value measured by the sensors is within a configurable process range. If this is the case, the FX3-ANA0 sets the **enable** bit that is sent to the main module to 1. If the sensors' measured value exceeds or falls below one of the configured limits, the FX3 ANA0 sets the **enable** bit to 0.

Up to 15 different process ranges can be configured. It is possible to switch from one process range to another process range during operation.

Furthermore, the total observable value range can be divided into up to 15 signal ranges. The FX3-ANA0 sends the number of the signal range in which the current measured value is located to the main module. This number can then be used for functions such as process control.

Sensors

WARNING
Ineffectiveness of the protective device due to selection of unsuitable sensors
The target safety-related level may not be achieved in the event of non-compliance.

- ► Select suitable sensors.
- ► Take suitable measures against the sensors' systematic errors and common causes of error.

Choosing the right sensors is crucial to achieving the desired safety integrity level (SIL) and performance level (PL). Systematic faults and common cause faults (CCF), in particular, need to be minimized in this case.

Sensors featuring diverse redundancy are supported for the safe measurement of a process variable. The characteristic lines of the sensors are standardized in the module for this purpose. The standardized measured values of the two sensors are compared with one another in order to check their plausibility.

Uniformly redundant sensors can also be used. In this case, the characteristic lines of both sensors must have identical configurations.

Depending on the process variable, a time delay can occur at sensors which are attached at a distance from one another within a local area, or which have different transceivers. This transit time difference can be taken into account during the plausibility check.

Instead of two redundant sensors, an individual single-channel or dual-channel safety sensor can be used. A single-channel safety sensor must be connected in series to both inputs.

Configuration steps

1. Add an FX3-ANA0 to the Flexi Soft project in the **hardware configuration.**
2. From the **elements catalog**, drag either two single-channel or one dual-channel **analog signal transmitter** to the FX3-ANA0 sensor inputs.

3. Double-click on the FX3-ANA0 to open the configuration dialog for the module.
4. Configure the Input signals of the connected sensors.
5. Configure one fixed process range or up to 15 variable Process ranges for the application.
6. Configure up to 15 different Signal ranges for the application (optional).

Fault detection

⚠ **WARNING**
Unclear fault detection by the sensors

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

► As part of the safety assessment, the parameters used to configure the FX3-ANA0 must be considered.

In accordance with generally recognized testing principles, test authorities usually stipulate that the application must ensure the monitored unit performs a signal change at least once within the space of 24 hours. This signal change must be conditioned so that the faults to be considered can be detected via a comparison of the analog values.

13.1 Input signals

The basic parameters for the plausibility check and the assessment of the data sent by the sensors are entered under the **input signal**.

The process variable to be monitored must be dynamic. If the physical values are virtually static, a short circuit or cross-circuit may not be detected at the sensor or at the input of the module.

⚠ **WARNING**
Impairment of the safety function due to short-circuit or cross-circuit

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

► If necessary, carry out daily tests to rule out short-circuits or cross-circuits at the sensor or at the input of the module.

Unit

The unit of the process variable to be monitored is entered here, e.g. **°C** or **Bar**.

The unit is the same for all the sensors connected to the module.

Characteristic curves

Two support points for calculating the characteristic curve must be entered for each connected sensor. The characteristic curve is calculated according to the following linear equation:

y = m × x + b

- m = gradient
- x = current measured value of the sensor
- b = offset

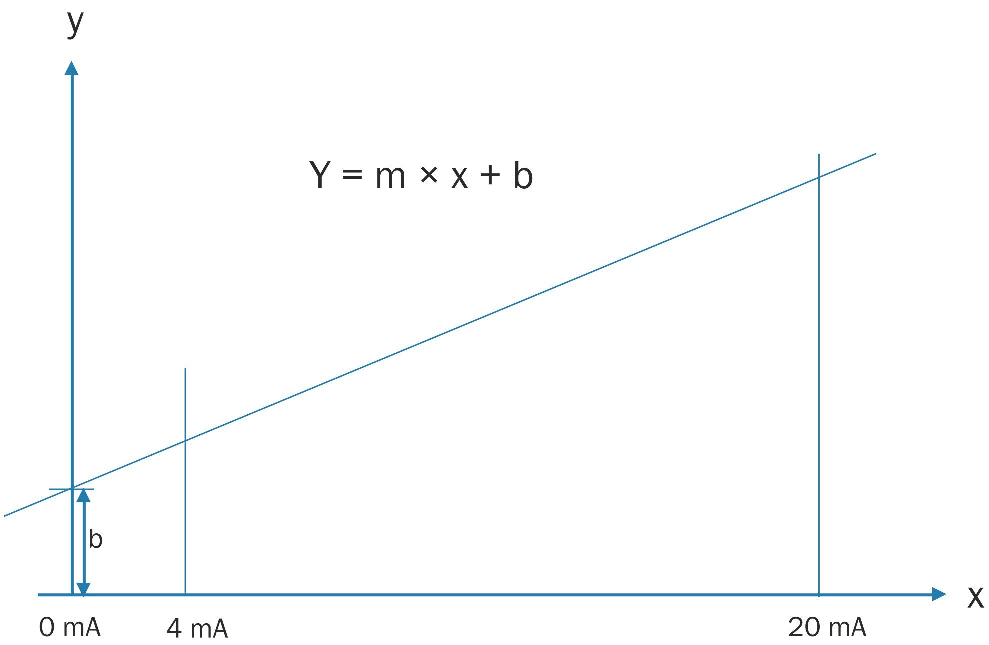


Figure 317: Calculation of the characteristic curve

The gradient and offset result from the support points entered. The gradient indicates the ratio of the measured value to the sensor signal in milliamperes. The offset corresponds to the measured value for an assumed sensor signal of 0 mA.

The **maximum monitorable range** is calculated based on the standardized characteristic curve of the sensors. The result of the calculation is shown at the bottom on the record card.

NOTE

- Only sensors with a standardized power interface in accordance with EN 61131-2 5.3.1 with a signal strength of 4 to 20 mA can be connected and evaluated. This range cannot be changed.
- Values outside of a signal strength range between 3.5 and 20.5 mA are interpreted as sensor errors.
- The input signal originating from the sensors must have a **linear** correlation with the process variable to be measured.
- Sensor AI1 must not have a falling characteristic curve.
- The **maximum monitorable range** is the value range which can be checked for plausibility by **both sensors** without either sensor exceeding or falling below a signal strength of 4 to 20 mA. The **maximum monitorable range** is not identical to the process range permissible for the application.
- If you are using uniform sensors, the characteristic curves must have identical configurations.
- Instead of two redundant sensors, one single-channel or dual-channel safety sensor can also be used. If only one sensor is used, then only the characteristic curve of that sensor can be configured. Further parameters described in this section (delay, discrepancy time monitoring, integration) are not required in this case.

Delay from sensor AI1 to sensor AI2

Using two sensors can lead to runtime differences in the sensor signals; this may be due to factors such as runtime differences in the sensor electronics or the two sensors being mounted in separate locations. Therefore, the evaluation of the signal from sensor AI1 may need to be delayed. In this case, the sensor with the shorter runtime must be connected to AI1 and the sensor with the longer runtime to AI2.

The delay affecting sensor AI1 can be adjusted in 4 ms increments from 0 to 252 ms.

If a delay is configured for sensor AI1, then the delayed value of sensor AI1 is used for both the consolidation of the measured values and the plausibility check. This can lead to an increase in the response time of the Flexi Soft system.

⚠ **WARNING**
Extension of the response time

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

► Take into account the delay for sensor AI1 when calculating the response time.

Discrepancy monitoring

For the plausibility check, the standardized measured values of both sensors (taking into account the delay for sensor AI1 if applicable) are compared to one another. The permissible difference between the two sensors serves to observe any discrepancies resulting from the measurement accuracy. The values must not differ by more than the **permanently permissible difference** which can be configured here.

A **limited permissible increased difference** and the **discrepancy time** – i.e., the permitted duration of the increased difference – can be configured as an option.

The **discrepancy time** can be adjusted from 0 to 65,532 ms in 4 ms increments.

⚠ **WARNING**
Impairment of the fault detection due to discrepancy monitoring

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

► Keep the values for the **permanently permissible difference** as well as, where applicable, for the **temporarily permissible increased difference** and the **discrepancy time** as low as possible.

The **discrepancy error reset time** depends on the configured **discrepancy time**. The **discrepancy error reset time** is five times the **discrepancy time**, but at least 1,000 ms and maximum 65,532 ms. If the **discrepancy time** is set to 0, the **discrepancy error reset time** is 65,532 ms.

Table 213: Discrepancy error reset time depending on the discrepancy time

Discrepancy time	Discrepancy error reset time
0 ms	65,532 ms
4 … 200 ms	1,000 ms
204 … 13,104 ms	5 × discrepancy time
13,108 … 65,532 ms	65,532 ms

Error status

If at least one of the two measured values is outside the signal strength range of 3.5 to 20.5 mA or a discrepancy error occurs, the FX3-ANA0 goes into the error state. In this case the module sets the bits for **Signal range**, **Enable**, and **Sensor status** to 0.

A discrepancy error occurs in the following cases:

- The measured values of sensor AI1 and sensor AI2 differ by more than the **unlimited permissible difference.** There is no **limited permissible increased difference** configured and/or the **discrepancy time** is set to 0 (not illustrated).
- The measured values from sensor AI1 and sensor AI2 deviate from one another for longer than the configured **Discrepancy time** and by more than the **Permanently permissible difference**, without the configured **Limited permissible increased difference** being exceeded (figure 318, case II).

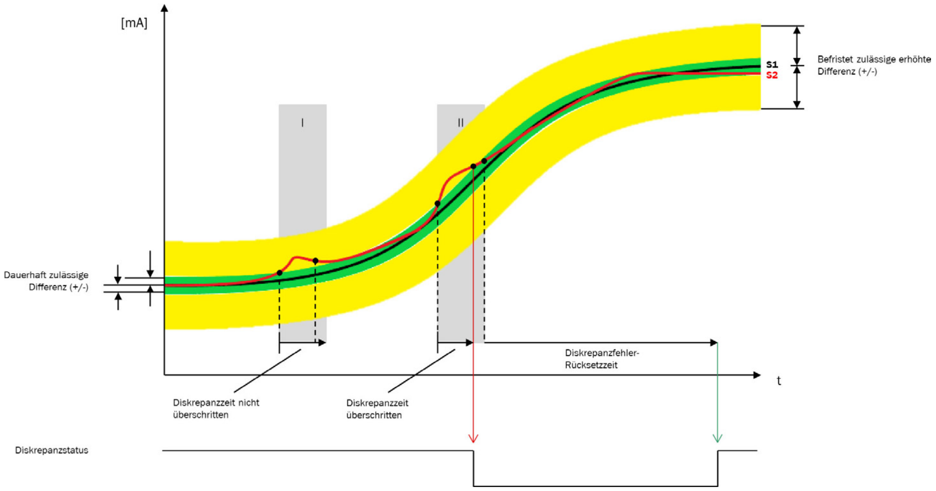

[mA]
I
II
S1
S2
Befristet zulässige erhöhte Differenz (+/-)
Dauerhaft zulässige Differenz (+/-)
Diskrepanzfehler-Rücksetzzeit
t
Diskrepanzzeit nicht überschritten
Diskrepanzzeit überschritten
Diskrepanzstatus

Figure 318: Exceeding the unlimited permissible difference and discrepancy time

- The measured values from sensor AI1 and sensor AI2 deviate from one another by **more** than the **Limited permissible increased difference** (figure 319).

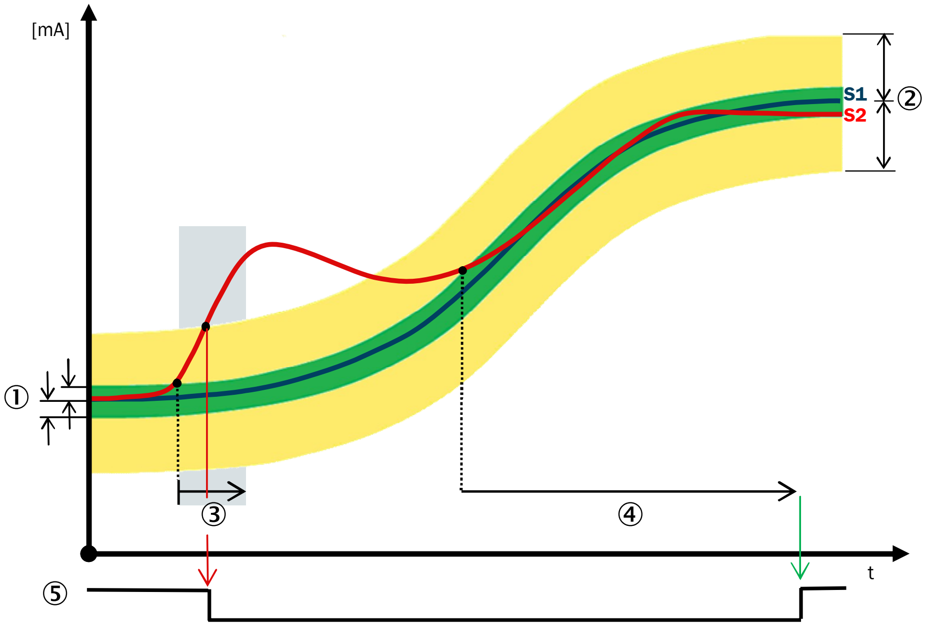


Figure 319: Exceeding the limited permissible increased difference

① Unlimited permissible difference (+/–)

② Limited permissible increased difference (+/–)

③ Discrepancy time

④ Discrepancy error reset time

⑤ Discrepancy status

Resetting an error status

A fault is reset if both signals are within the current process range again, differ from one another by less than the **permanently permissible difference**, and this status persists for at least the **discrepancy error reset time.** The FX3-ANA0 then begins to evaluate the measured values again according to the configuration, and sets the bits for **Signal range**, **Enable** and **Sensor status** to the appropriate values.

The **Discrepancy reset time** depends on the configured **Discrepancy time**, see "Discrepancy monitoring", page 440.

WARNING
Uncontrolled machine restart due to consecutive sensor errors

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

- ► Use the logic program to prevent the machine or plant from restarting automatically after a discrepancy error due to the re-issued approval from the FX3-ANA0.
- ► Asses the **Sensor status** input in the logic program and connect it with a corresponding reset function.
- ► Check the sensors and inputs after a discrepancy error occurs.

Value for process and signal range monitoring

This parameter determines how the process variable relevant for further evaluation is calculated using the measurement results from the sensors when a valid signal is received. The following options are available:

- Sensor AI1 (S1)
- Sensor AI2 (S2)
- Maximum value of S1 and S2
- Minimum value of S1 and S2
- Average value of S1 and S2

13.2 Process ranges

The process range check can be used to monitor whether the measured value is within adjustable process limits.

The FX3-ANA0 can operate either with a **fixed process range** or with up to 15 **variable process ranges.** A process range consists of two configurable values, the **minimum** value, and the **maximum** value of the permissible value range.

NOTE
The **minimum** and the **maximum** values of a process range must not fall outside the **maximum observable range.** The latter depends on the type of sensors used, and is calculated by the FX3-ANA0 based on the sensors' characteristic curves.

Fixed process range

If the **Fixed process range** option is activated, the FX3-ANA0 works with the values configured there as the **minimum** and **maximum** values.

If the tested and valid input signal of the connected sensors falls within the process range, the FX3-ANA0 sets the **enable** bit to 1. If the signal falls below the configured **minimum** or above the configured **maximum** values, the FX3 ANA0 sets the **enable** bit to 0.

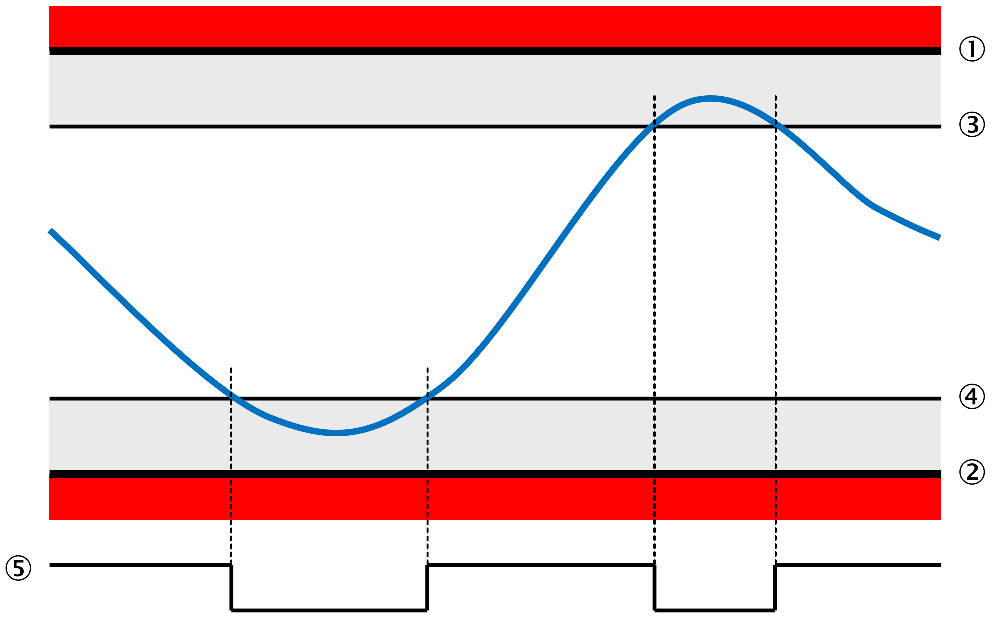


Figure 320: Behavior of the FX3-ANA0 with fixed process range

①	Upper limit of the current measuring range 20 mA
②	Lower limit of the current measuring range 4 mA
③	Upper process range limit (maximum, configurable)
④	Lower process range limit (minimum, configurable)
⑤	Enable

If the **Fixed process range** option is activated, any **process range** that is selected in the main module logic is ignored.

Variable process range

If the FX3-ANA0 is configured with the **Variable process range** option, it is possible to switch between up to 15 different process ranges during operation using the logic program in the main module.

Each process range that is to be used must be activated, and the **minimum** and **maximum** values of the process range must be configured.

One of the activated and configured **process ranges** can be selected using the four **process range bit x** outputs of the FX3-ANA0 in the main module logic editor. Here, **bit 3** represents the most significant bit and **bit 0** represents the least significant bit.

Table 214: Selecting the process range in the FX3-ANA0

Process range	FX3-ANA0 outputs in the logic editor			
	Process range bit 3	Process range bit 2	Process range bit 1	Process range bit 0
0001	0	0	0	1
0010	0	0	1	0
0100	0	1	0	0
0101	0	1	0	1
...	...	...	...	...

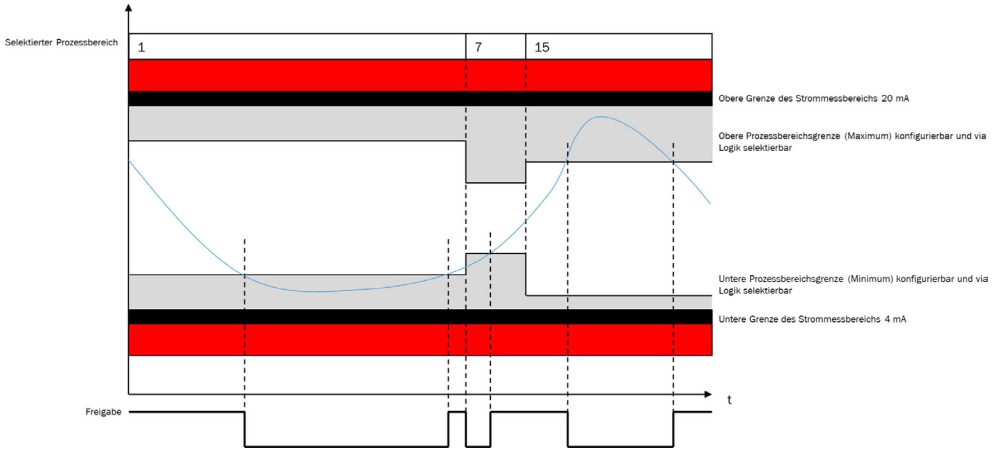

Selektierter Prozessbereich
1
7
15
Obere Grenze des Strommessbereichs 20 mA
Obere Prozessbereichsgrenze (Maximum) konfigurierbar und via Logik selektierbar
Untere Prozessbereichsgrenze (Minimum) konfigurierbar und via Logik selektierbar
Untere Grenze des Strommessbereichs 4 mA
t
Freigabe

Figure 321: Behavior of the FX3-ANA0 with variable process range

NOTE

If no process range is selected (all process range bits = 0), or if a process range that is not activated in the FX3-ANA0 configuration is selected, this results in a violation of the process range. In this case, the FX3-ANA0 sets the **enable** bit to 0.

13.3 Signal ranges

The **maximum observable range** can be divided into up to 15 **signal ranges.** In addition to the process range check, this enables you to determine more precisely in which value range the sensor value is located. For this purpose, the FX3-ANA0 sends the number of the current signal range to the main module.

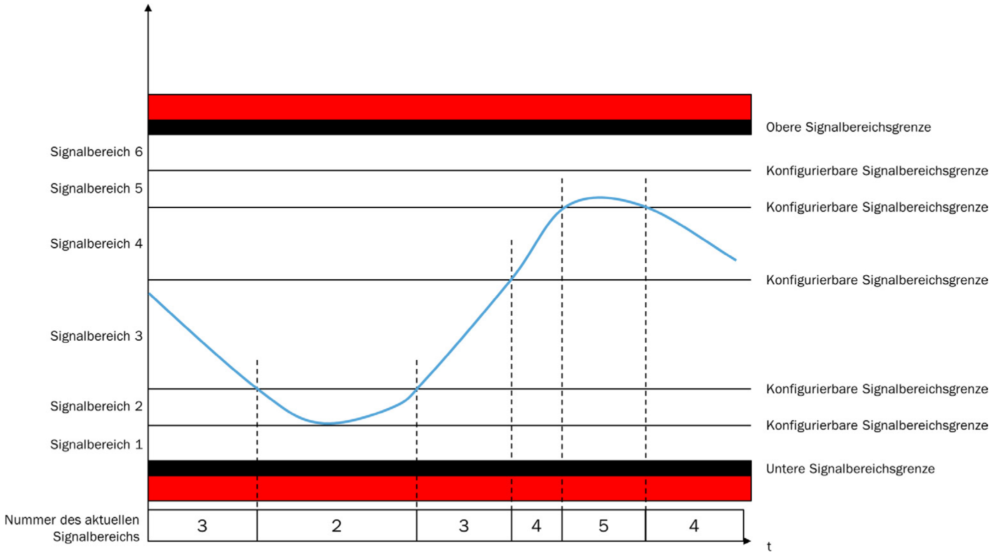

Signalbereich 6
Signalbereich 5
Signalbereich 4
Signalbereich 3
Signalbereich 2
Signalbereich 1
Obere Signalbereichsgrenze
Konfigurierbare Signalbereichsgrenze
Konfigurierbare Signalbereichsgrenze
Konfigurierbare Signalbereichsgrenze
Konfigurierbare Signalbereichsgrenze
Konfigurierbare Signalbereichsgrenze
Untere Signalbereichsgrenze
Nummer des aktuellen Signalbereichs
3
2
3
4
5
4
t

Figure 322: FX3-ANA0 signal range check with 6 configured signal ranges

NOTE

- The signal range check does not depend on the result of the process range check or on the **enable** bit value.
- The number of the current signal range is always transmitted as long as a valid signal is present.
- If the sensor value is invalid – in the event of a sensor or discrepancy error, for example – the FX3-ANA0 sets the number of the signal range to 0.
- If the sensor value is exactly on the limit between two signal ranges, the higher of the two signal ranges is active.

By default, only a single signal range is configured with number 1. The associated signal range limits are identical to the upper limit and the lower limit of the **maximum observable range**.

Configuring additional signal ranges

- ► Click to select an existing signal range.
- ► Click the up arrow to insert another signal range above the selected one.
- ► Click the down arrow to insert another signal range below the selected one.
- ► If necessary, change the values of the new signal range limit.
- ► If applicable, enter a name for the new signal range limit.

NOTE
The minimum and maximum signal range limits cannot be changed. They correspond to the limits of the **maximum observable range**.

Deleting signal ranges

- ► Click to select an existing signal range.
- ► Click the **Delete** button to delete the selected signal range.

NOTE
The minimum and maximum signal range limits cannot be deleted.

Coding of signal ranges

The coding of the signal ranges determines how the number of the current signal range is transmitted to the main module. The coding also determines how many bits the signal range number sent to the main module is comprised of, and how many signal ranges can be configured.

Table 215: Coding of signal ranges

Coding	Max. number of signal ranges	Number of bits
1-of-n (name)	7	7
Binary	15	4
Leveling	7	7
Inverse leveling	6	7

NOTE
To select the coding **1-of-n**, activate the **Name** option.

1-of-n (name)

With 1-of-n coding, a bit is assigned to each signal range. A maximum of seven signal ranges are available.

The advantage of this coding is that each bit can be assigned a tag name.

Table 216: 1-of-n coding of signal ranges

Number [1]	Bit pattern	Signal range						
		Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7	1000000	1	0	0	0	0	0	0
6	0100000	0	1	0	0	0	0	0
5	0010000	0	0	1	0	0	0	0
4	0001000	0	0	0	1	0	0	0
3	0000100	0	0	0	0	1	0	0
2	0000010	0	0	0	0	0	1	0
1	0000001	0	0	0	0	0	0	1

[1] Number 0 is invalid or represents an error.

Binary

With binary signal range coding, the number of the current signal range is transmitted to the main module as a binary value. Four bits are used; **Signal range bit 3** is the highest-value bit.

With this coding, the maximum number of up to 15 signal ranges are available. However, no tag names can be used.

Table 217: Binary coding of signal ranges

Number [1]	Bit pattern	Signal range			
		Bit 3	Bit 2	Bit 1	Bit 0
15	1111	1	1	1	1
14	1110	1	1	1	0
13	1101	1	1	0	1
...	...	...	...	...	...
2	0010	0	0	1	0
1	0001	0	0	0	1

[1] Number 0 is invalid or represents an error.

Leveling

With leveling coding, the seven used bits are activated in ascending order. A maximum of seven signal ranges are available.

Leveling coding is typically used in applications to measure fill levels. As the fill level rises, all the thresholds that are exceeded are flagged up as active.

Table 218: Leveling coding of signal ranges

Number [1]	Bit pattern	Signal range						
		Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7	1111111	1	1	1	1	1	1	1
6	0111111	0	1	1	1	1	1	1
5	0011111	0	0	1	1	1	1	1
4	0001111	0	0	0	1	1	1	1
3	0000111	0	0	0	0	1	1	1
2	0000011	0	0	0	0	0	1	1
1	0000001	0	0	0	0	0	0	1

[1] Number 0 is invalid or represents an error.

Inverse leveling

With inverse leveling coding, the seven used bits are deactivated in ascending order. A maximum of six signal ranges are available.

As is the case with leveling coding, inverse leveling coding is typically used in applications to measure fill levels. As the fill level rises, all the thresholds that are exceeded are signaled as deactivated.

Table 219: Inverse leveling coding of signal ranges

Number [1]	Bit pattern	Signal range						
		Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
6	1000000	1	0	0	0	0	0	0
5	1100000	1	1	0	0	0	0	0
4	1110000	1	1	1	0	0	0	0
3	1111000	1	1	1	1	0	0	0
2	1111100	1	1	1	1	1	0	0
1	1111110	1	1	1	1	1	1	0

[1] Number 0 is invalid or represents an error.

13.4 Additional configuration

Averaging

A smoothed average based on several samples is generated for the purpose of evaluating the sensor signals. This is used to compensate for interference impulses and to obtain more reliable values.

Automated average value formation

This is the default setting. The FX3-ANA0 averages 32 scans per processing cycle (4 ms).

Manual configuration of the averaging

There are two ways to adjust the averaging:

- The number of analyzed cycles can be increased to a maximum of 25.
  The more cycles are analyzed, the less sensitive the response of the FX3-ANA0 to changes in the sensor signal. This can compensate for signal fluctuations caused by interference.
  32 scans per cycle are always used for this setting.
  When averaging over several cycles, the response time of the FX3-ANA0 is increased by the cycle time (4 ms) per analyzed cycle.

WARNING
Extension of the response time

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

► Take into account the number of analyzed cycles when calculating the total response time of the Flexi Soft system.

For detailed information on calculating the Flexi Soft system response time, please refer to the operating instructions titled “Flexi Soft Modular Safety Controller Hardware”.

- The number of scans for averaging can be reduced. If this option is activated, then an average of the selected number of scans is generated per cycle. The maximum value corresponds to the default setting of 32 scans. The averaging over several cycles is not active in this case, i.e. only one cycle is analyzed at a time.
  If the number of scans is reduced, the FX3-ANA0 will be more sensitive to changes in the sensor signal. This may be particularly necessary if even slight changes in the signal must be registered quickly or if quick successive strong fluctuations of the signal are expected.

Hiding status bits 16 to 31

The status bits 16 to 31 of the FX3-ANA0 continuously provide the measured value, e.g. for forwarding via a gateway. EtherCAT gateways (FX0-GETC) interpret these bits as error bits. Every change to one of these bits is therefore seen as an error.

The status bits 16 to 31 of the FX3-ANA0 can be hidden to prevent error messages from the EtherCAT gateway.
► Select the option **Hide status bits 16 to 31**.
✓ The status bits 16 to 31 of the FX3-ANA0 are set to the static value 1.

NOTE
This function is available with firmware V2.00.0 or higher of the FX3-ANA0.

13.5 The FX3-ANA0 in the logic editor

Inputs

The FX3-ANA0 provides the following safety capable inputs:

Table 220: FX3-ANA0 inputs in the logic editor

Input	Description
Signal range bit 0 to 6	For more information on the transmission of the number of the current signal range, see "Signal ranges", page 444
Enable	1 = No error. All of the following conditions are met: • All sensor signals are valid. • There are no discrepancy errors. • The measured value falls within the current process range. 0 = error: At least one of the specified conditions has not been met.
Sensor status	1 = No error. All of the following conditions are met: • All sensor signals are valid. • There are no discrepancy errors. 0 = sensor error: At least one of the specified conditions has not been met. **Note**: If the **enable** bit is set to 0, the **sensor status** can be used to determine if the error was caused by a sensor error or a valid measured value outside the process range.

WARNING
Uncontrolled machine restart due to consecutive sensor errors

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

- ► Use the logic program to prevent the machine or plant from restarting automatically after a discrepancy error due to the re-issued approval from the FX3-ANA0.
- ► Asses the **Sensor status** input in the logic program and connect it with a corresponding reset function.
- ► Check the sensors and inputs after a discrepancy error occurs.

Outputs

The FX3-ANA0 provides the following safety outputs:

Table 221: FX3-ANA0 outputs in the logic editor

Output	Description
Process range bit 0 to 3	For selecting the process range, see "Process ranges", page 442

13.5.1 Module status bits of the FX3-ANA0

Overview

The module status bits for the expansion modules contain diagnostics data. This data is refreshed approximately every 200 ms. Due to the longer refresh interval, this data may not be consistent with the latest process data for the module.

Important information

WARNING
Non-secure or inconsistent data

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

- ► Only use secure data for safety-related applications.
- ► Only use the module status bits of the expansion modules for diagnostic purposes.

Module status bits of the FX3-ANA0

Table 222: Module status bits of the FX3-ANA0

Module status bit	Description
Configuration is valid	1 = Configuration valid 0 = Configuration invalid
Sensor AI1 lower input range OK	1 = No error 0 = Error: Measured value from sensor AI1 is below the permissible range (< 3.5 mA)
Sensor AI1 upper input range OK	1 = No error 0 = Error: Measured value from sensor AI1 exceeds the permissible range (> 20.5 mA)
Sensor AI2 lower input range OK	1 = No error 0 = Error: Measured value from sensor AI2 is below the permissible range (< 3.5 mA)
Sensor AI2 upper input range OK	1 = No error 0 = Error: Measured value from sensor AI2 exceeds the permissible range (> 20.5 mA)
Sensor AI1 lower process range OK	1 = No error 0 = Error: Measured value from sensor AI1 is below the minimum value of the current process range
Sensor AI1 upper process range OK	1 = No error 0 = Error: Measured value from sensor AI1 exceeds the maximum value of the current process range
Sensor AI2 lower process range OK	1 = No error 0 = Error: Measured value from sensor AI2 is below the minimum value of the current process range
Sensor AI2 upper process range OK	1 = No error 0 = Error: Measured value from sensor AI2 exceeds the maximum value of the current process range
Discrepancy status OK	1 = No error 0 = Discrepancy error
Lower process range limit OK	1 = No error 0 = Error: Consolidated signal from the sensors is below the minimum value of the current process range
Upper process range limit OK	1 = No error 0 = Error: Consolidated signal from the sensors exceeds the maximum value of the current process range

Module status bit	Description
Measured value bit 0 … 15 [1]	Analog value of the consolidated signal from the sensors **Note:** The consolidated, scaled value is transferred in digits. Bit 0 is the lowest-value bit and bit 15 is the highest-value bit. The value of the bits can be found in the report. The sensor value can also be calculated using the following formula: Sensor value = measured value × m / 2,500 + b • Measured value = The digital value of measured value bit 0 – 15. • m = gradient of sensor AI1 [2] • b = offset of sensor AI1 The values for the gradient and offset of sensor AI1 can be taken from the report.
Input data status	Corresponds to the safe input **sensor status** 1 = No error. All of the following conditions are met: • All sensor signals are valid. • No discrepancy error is present. 0 = Sensor error: At least one of the specified conditions is not met.

[1] If the Flexi Soft system contains gateways, the measured value is made permanently available to these gateways. EtherCAT gateways (FX0-GETC) may interpret the measured value as an error. For this reason, if an FX0-GETC is used, the alarm memory in the control system must be either read out on an ongoing basis or these bits must be hidden.

[2] Sensor AI1 is the leading sensor. For this reason, the gradient of sensor AI1 must be used to convert the measured value, irrespective of the **sensor merging** settings configured under **Input signals**.

13.6 The FX3-ANA0 in the data recorder

You can record one analog measured value in the data recorder per FX3-ANA0. The following data is available:

- Sensor AI1, filtered
- Sensor AI2, filtered
- Sensor AI1, delayed
- Consolidated measured value
- Sensor AI1, unfiltered (last sample)
- Sensor AI2, unfiltered (last sample)

NOTE

The analog measured values of the FX3-ANA0 can be selected on the **Diagnostics** tab.

In addition, all the FX3-ANA0 inputs, outputs, and diagnostic bits that are available in the logic editor can be recorded in the data recorder.

In total, up to 4 channels can be recorded.

14 Flexi Link

14.1 Overview of Flexi Link

Flexi Link allows up to four Flexi Soft stations to be combined via EFI so that data can be exchanged safely and securely. It is only possible to use FX3-CPU1 main modules or higher as part of a Flexi Link system. It is not possible to connect FX3-CPU0 main modules.

The process data of each station (inputs and outputs, logic results, etc.) can be made available to all the other stations in the Flexi Link system. The Teach function allows you to deactivate individual stations temporarily without impairing the function of the system as a whole.

Features

- Secure connection of up to four Flexi Soft stations via EFI
- Connection via EFI1 or EFI1+2
- Send/receive up to 52 bits of information per station (26 bits per EFI connection)
- A globally valid tag name can be assigned to each bit.
- The Teach function can be used to simulate the presence of temporarily suspended (deactivated) stations.
- Any station can be used as an access point for the purpose of addressing and configuring the entire system with Flexi Soft Designer.
- The configuration for the entire Flexi Link system is stored in a single project file.

14.1.1 System requirements and restrictions for Flexi Link

The following minimum system requirements must be met for Flexi Link:

Table 223: System requirements for Flexi Link

System component	Version
Hardware	FX3-CPU1, FX3-CPU2, or FX3-CPU3 with firmware version ≥ V2.00.0
Software	Flexi Soft Designer version ≥ V1.3.0

The Flexi Link system can either by connected via EFI1 only or via EFI1+2. The total number of status bits per station that can be made available to the other stations in the same Flexi Link system depends on which connection type is used.

Table 224: Available status bits depending on connection type

Type of connection	Status bits available per station
EFI1	26
EFI1+2	52

NOTE

- Flexi Link cannot be combined with EFI communication. That is, it is not possible to connect other EFI-enabled devices to EFI2 when using EFI1 for Flexi Link.
- The process data that is sent from any station is received by all the other stations at virtually the same time. However, as the stations are not synchronized, (logic) processing by the individual stations is not necessarily simultaneous.
- The data is consistent within EFI1 and within EFI2. However, the EFI1 and EFI2 data may be temporarily inconsistent because it is transmitted separately.

14.2 Principle of operation

A Flexi Link system is configured in two steps.

- The first step is to configure the network settings and Flexi Link address. As part of this process, the system automatically detects wiring errors or the presence of devices that are not suitable for Flexi Link projects.
- The second step is to configure the individual stations in the system: main module, expansion modules, connected elements, gateways, logic, and process image for the Flexi Link system.

14.2.1 Flexi Link address

Flexi Soft Designer needs the Flexi Link address so that each station within a Flexi Link system (up to a maximum of four) can be uniquely identified. This is the first major setting that has to be made when configuring a Flexi Link system.

The Flexi Link address is freely configurable from A to D.
For detailed information about assigning a Flexi Link address, see "Flexi Link system: Network settings", page 470.

14.2.2 Flexi Link ID

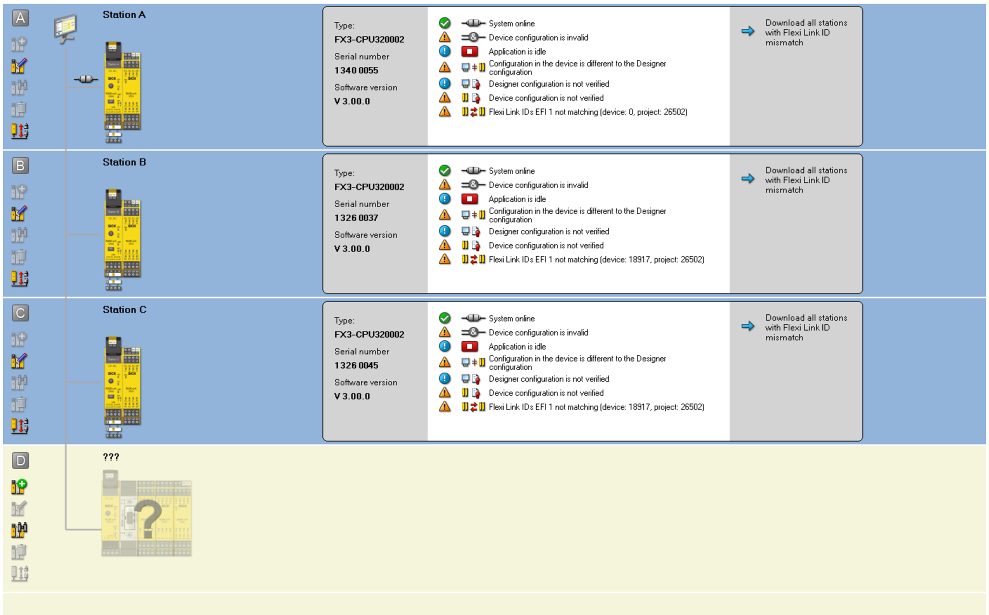
The Flexi Link ID is required so that the stations in a Flexi Link system can communicate with one another. All the stations in a Flexi Link system must have an identical Flex Link ID in order to be able to exchange their process images with one another. This ensures that stations can only communicate with each other if they belong to the same Flexi Link system. If an inconsistent Flexi Link ID is detected within a Flexi Link system, all connected stations switch to **Invalid configuration** mode (MS LED flashes ☀ red at 1 Hz).

The Flexi Link IDs are numerical values and are calculated on the basis of the default values for the process image. Consequently, if a change is made to the default values for the process image of any station, the Flexi Link ID of all the other stations changes as well. The process of adding or deleting a station likewise changes the Flexi Link ID of the system.

NOTE
If a change is made to the process image of any station, the new configuration must be transferred to all the stations (e.g., using the **Transfer** command in the Flexi Link **System overview**). This sets all the Flexi Link IDs to the same value at once. Otherwise, inconsistent Flexi Link IDs will be present within the system, causing the safety communication between the stations to be interrupted.

The Flexi Link IDs form part of the configuration, are transferred together with it, and are stored in the system plug of each connected main module.

The Flexi Link IDs of the current configuration in Flexi Soft Designer are always displayed in the Flexi Link toolbar. The Flexi Link IDs that are currently stored in the individual stations are displayed in the Flexi Link **System overview** and continue to be compared with the Flexi Link ID in the project file on the computer for as long as this computer remains connected to the station. If Flexi Soft Designer detects an inconsistent Flexi Link ID, a warning symbol appears and it recommends what to do next on the right-hand side of the screen.



A

Station A

Type:
FX3-CPU320002
Serial number
1340 0055
Software version
V 3.00.0

✓ System online
⚠ Device configuration is invalid
ⓘ Application is idle
⚠ Configuration in the device is different to the Designer configuration
ⓘ Designer configuration is not verified
⚠ Device configuration is not verified
⚠ Flexi Link IDs EFI 1 not matching (device: 0, project: 26502)

Download all stations with Flexi Link ID mismatch

B

Station B

Type:
FX3-CPU320002
Serial number
1326 0037
Software version
V 3.00.0

✓ System online
⚠ Device configuration is invalid
ⓘ Application is idle
⚠ Configuration in the device is different to the Designer configuration
ⓘ Designer configuration is not verified
⚠ Device configuration is not verified
⚠ Flexi Link IDs EFI 1 not matching (device: 18917, project: 26502)

Download all stations with Flexi Link ID mismatch

C

Station C

Type:
FX3-CPU320002
Serial number
1326 0045
Software version
V 3.00.0

✓ System online
⚠ Device configuration is invalid
ⓘ Application is idle
⚠ Configuration in the device is different to the Designer configuration
ⓘ Designer configuration is not verified
⚠ Device configuration is not verified
⚠ Flexi Link IDs EFI 1 not matching (device: 18917, project: 26502)

Download all stations with Flexi Link ID mismatch

D

???

Figure 323: Flexi Link IDs displayed in the Flexi Link system overview

The Flexi Link IDs are also displayed under the Flexi Link **Network settings**.

If the configuration of any station within the Flexi Link system is changed in such a way as to affect the system process image (e.g., if a station is added to the system or if the default value for one of the transmitted bits is changed), Flexi Soft Designer calculates a new Flexi Link ID on the basis of the modified process image. In this case, the configuration must be transfered to all stations within the system and not just to the station whose configuration has been changed. Otherwise, the new Flexi Link IDs will only be transferred to this station, with all the others retaining the old Flexi Link ID. The resulting inconsistency between the Flexi Link IDs of the various stations will then interrupt the exchange of process images within the system. If an inconsistent Flexi Link ID is detected, it is no longer possible for process images to be transferred between the stations and all the main modules within the system indicate a remediable error (MS LED flashes ☀ red at 1 Hz and the EFI1 and EFI2 LEDs light up ● red). In this case, station configuration and diagnostics are the only available options.

For additional information about correcting an inconsistent Flexi Link ID, see "Flexi Link troubleshooting", page 480.

14.3 Setting up a new Flexi Link system

This chapter describes how to set up a new Flexi Link system. Before doing this, the hardware for the project must be configured. This can be achieved in two ways:

- Start by assembling and wiring the hardware, then connect the computer to the system and use Flexi Soft Designer to load the hardware configuration.
- If the necessary hardware is not yet available, first set up the hardware configuration for the Flexi Link project in Flexi Soft Designer, then transfer the configuration at a later point once the hardware has been assembled.

As soon as the entire hardware configuration for the Flexi Link project has been fully created using either of the methods described, continue with the software configuration process. The final step is to transfer the finished configuration to the various stations of the Flexi Link project, then verify and start the system.

14.3.1 Connecting to an existing hardware system

Step 1: Assemble and wire the hardware

- ► Set up the hardware for the Flexi Link system (Flexi Soft main modules FX3-CPU1, FX3-CPU2, or FX3-CPU3, expansion modules, and connected devices such as sensors, switches, actuators).

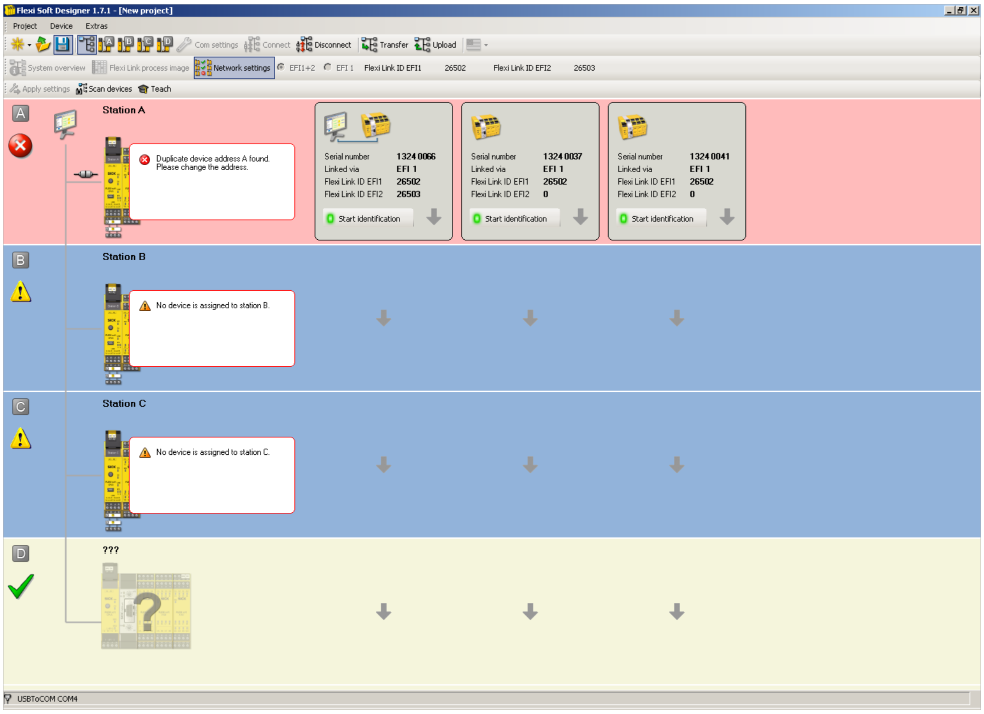
NOTE
For wiring information, refer to the “Flexi Soft Modular Safety Controller Hardware” operating instructions (SICK part number 8012999) or the Flexi Soft Designer report.

Step 2: Establish a connection to the hardware system

- ► Connect a computer to the RS-232 interface of the USB interface of any main module within the system.
- ► Switch on the hardware system.
- ► Start the Flexi Soft Designer configuration software that has been installed on the computer.
- ► If necessary, adjust the communication settings (see "Editing communication settings", page 28).
- ► Click on **Connect to physical device** or go to the **Device** menu and select the **Detect project** command. Flexi Soft Designer then searches the network for connected devices.

NOTE
If the error message “No valid Flexi Link system found” appears, check that all the main modules in the system have a firmware version of at least V2.00.0. You will find the firmware version on the module type label in the **Firmware version** field.

- ► The connected main modules will only have valid Flexi Link addresses if they have already been configured for Flexi Link. Otherwise, Flexi Soft Designer will now open the Flexi Link **Network settings** with a list of all the stations found:



Flexi Link ID EFI1 26502 Flexi Link ID EFI2 26503

Duplicate device address A found. Please change the address.

Serial number	1324 0066	Serial number	1324 0037	Serial number	1324 0041
Linked via	EFI 1	Linked via	EFI 1	Linked via	EFI 1
Flexi Link ID EFI1	26502	Flexi Link ID EFI1	26502	Flexi Link ID EFI1	26502
Flexi Link ID EFI2	26503	Flexi Link ID EFI2	0	Flexi Link ID EFI2	0

No device is assigned to station B.

No device is assigned to station C.

Figure 324: Flexi Link network settings without valid addresses assigned

- ► Using the Up and Down arrow keys or the mouse, move the stations into the various rows for stations A through D so that no two stations occupy the same address.
- ► There are two ways to identify a station:
  - ► On one of the displayed stations, click the **Start identification** button. The MS and EFI1 LEDs of the corresponding station start to flash alternately with the EFI2 LED (2 Hz). The Authorized client password is required for this. The default password is “SICKSAFE”. To stop the LEDs flashing, click the button again (which is now called **Stop identification**).
  - ► Check the serial number on the system plug and compare it with the serial number displayed in Flexi Soft Designer. The serial number displayed under the **Network settings** is the serial number of the system plug and not the one for the main module.

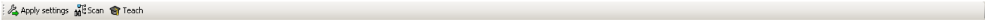
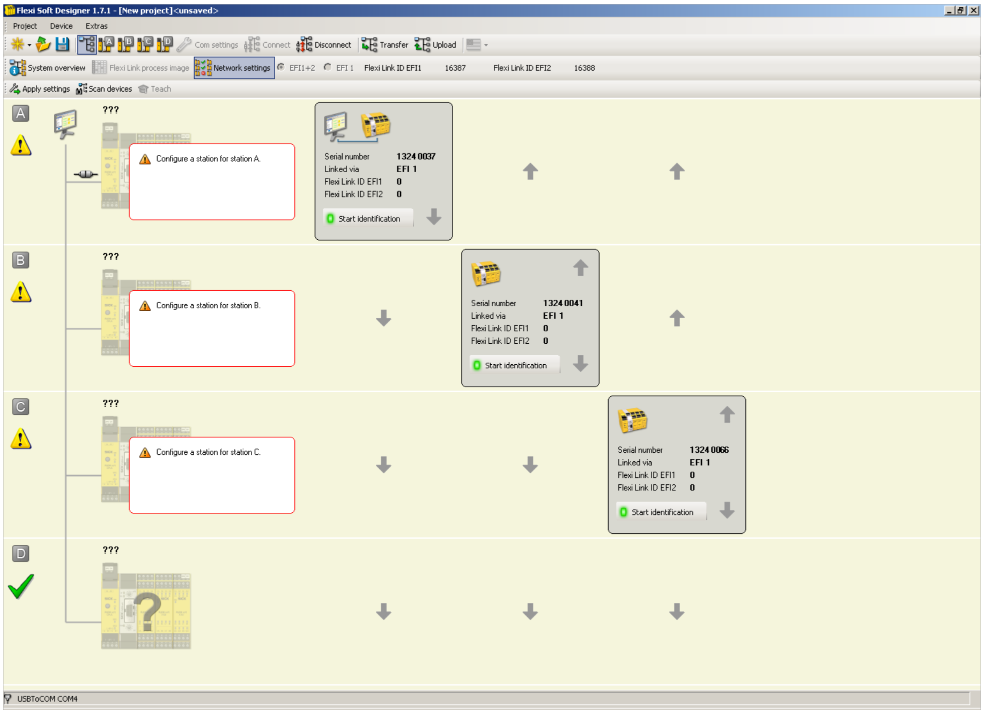


Figure 325: Apply settings button

- ► In the upper left corner of the screen, click on the **Apply settings** button. The Flexi Link addresses of the stations are changed.



Flexi Link ID EFI1 16387 Flexi Link ID EFI2 16388

Configure a station for station A.

Serial number 1324 0037
Linked via EFI 1
Flexi Link ID EFI1 0
Flexi Link ID EFI2 0

Configure a station for station B.

Serial number 1324 0041
Linked via EFI 1
Flexi Link ID EFI1 0
Flexi Link ID EFI2 0

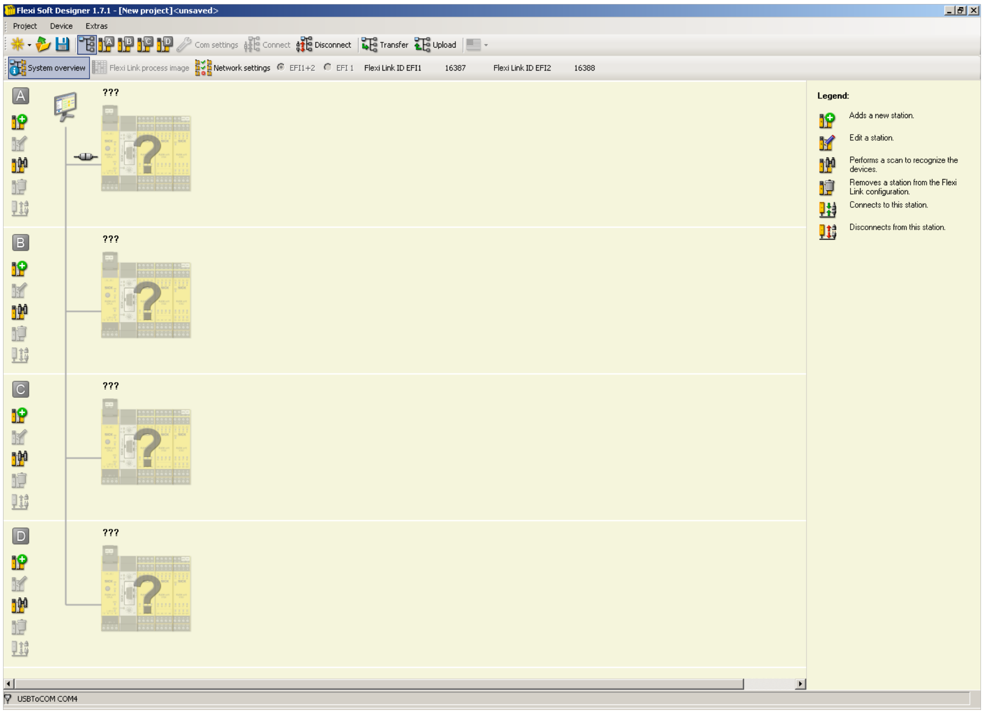
Configure a station for station C.

Serial number 1324 0066
Linked via EFI 1
Flexi Link ID EFI1 0
Flexi Link ID EFI2 0

Figure 326: Flexi Link network settings with valid addresses assigned

Step 3: Load the hardware settings

- ► Click the **System overview** button. The following view opens:



Legend:

Adds a new station.

Edit a station.

Performs a scan to recognize the devices.

Removes a station from the Flexi Link configuration.

Connects to this station.

Disconnects from this station.

Figure 327: Flexi Link system overview, connected status

- ► Click the binoculars symbol next to station A. Flexi Soft Designer loads the hardware and configuration settings for all the devices of this station. Repeat for each of the other stations.
- ► Once the hardware configuration is complete, click **Disconnect**. The project can now be configured (see "Flexi Link configuration", page 458).

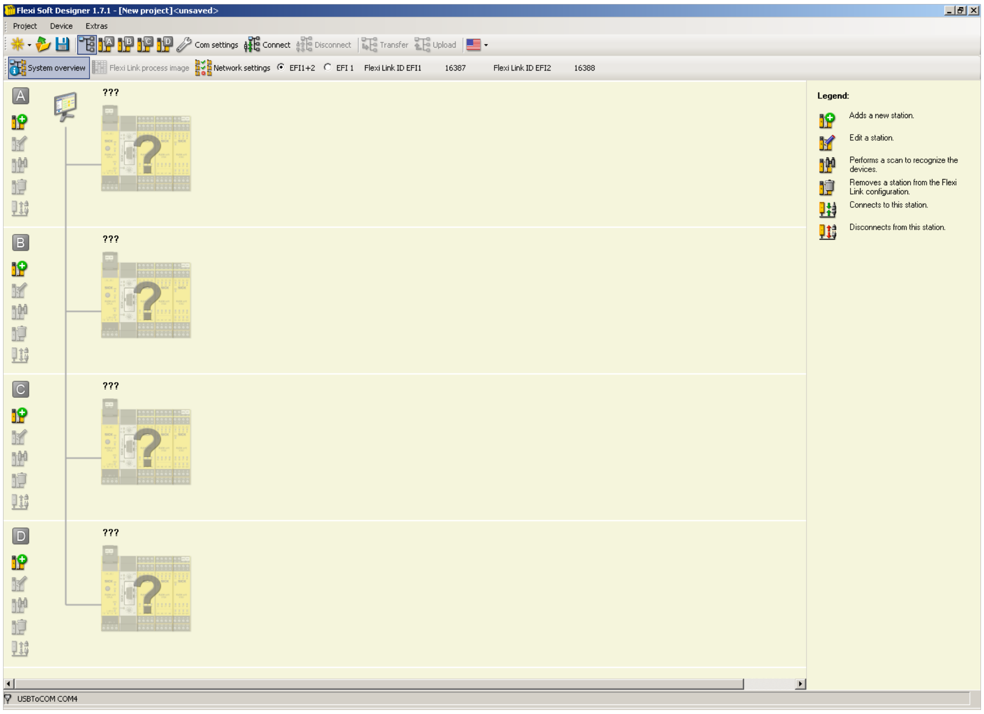
NOTE

The **Disconnect** command in the system overview simultaneously breaks the connections to all Flexi Link stations. The **Transfer** and **Upload** buttons work in a similar way. However, if the view for only one station is open, the buttons only affect this station.

14.3.2 Setting up a Flexi Link project when the hardware is not available

If the necessary hardware is not yet available, the hardware configuration for the Flexi Link project can be set up in advance in Flexi Soft Designer.

- ► Open the Flexi Soft Designer.
- ► In the start dialog, click on **Create new Flexi Link project** or go to the **Project** menu and select **New** followed by **Flexi Link project**. The Flexi Link **System overview** opens.



● EFI1+2 ○ EFI 1 Flexi Link ID EFI1 16387 Flexi Link ID EFI2 16388

Legend:

- Adds a new station.
- Edit a station.
- Performs a scan to recognize the devices.
- Removes a station from the Flexi Link configuration.
- Connects to this station.
- Disconnects from this station.

Figure 328: Flexi Link system overview, unconnected status

► Select whether to wire the Flexi Link system using EFI1 only, or using both EFI connections (**EFI1+2** option). With EFI1, up to 26 bits can be exchanged per station. If EFI1 and EFI2 are used together, each station can make up to 52 bits available to the other stations.

NOTE
This setting can still be changed at a later time.

► Add the first station to the project. Click one of the buttons for adding new stations that can be found on the left-hand side of the screen (green “+” symbol). This opens the view for the single station concerned. Alternatively, go to the top edge of the screen and use the station buttons on the toolbar to switch to the view for a single station.

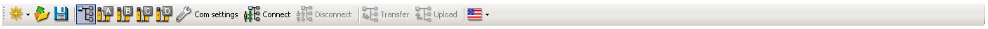


Figure 329: Buttons for the Flexi Link system overview and stations

► In the view for this single station, add the desired hardware, see "Configuring the Flexi Soft modules", page 42 and see "Connecting elements", page 44.
► Once the hardware configuration for the selected station is complete, click the **Flexi Link system overview** button on the toolbar to switch back to the Flexi Link system overview.
► Now add the other necessary stations to the Flexi Link project.
► Once the hardware configuration for the Flexi Link system is complete, the project can configured.

14.3.3 Flexi Link configuration

This section reveals in detail how data can be exchanged between the individual stations of a Flexi Link system.

Example: Straightforward Flexi Link project with two stations. An emergency stop pushbutton and a restart button on station A are used for the simultaneous control of two robots, which are connected to stations A and B.

Setting up the hardware

- ► Create a new Flexi Link project (see "Setting up a Flexi Link project when the hardware is not available", page 457).
- ► In the Flexi Link **System overview**, set the connection type to **EFI1+2**. Then click the **Insert new station** button for station A. The **Hardware configuration** for station A opens.
- ► Add an FX3-CPU1, FX3-CPU2, or FX3-CPU3 main module and an FX3-XTIO module to station A.
- ► Connect a single-channel emergency stop pushbutton to input I1 and a single-channel reset to input I2 of the FX3-XTIO module of station A.
- ► Connect a single-channel robot to output Q1 and a lamp to output Q2 of the FX3-XTIO module of station A.
- ► Go to the toolbar and click the button for station B. The **Hardware configuration** for station B opens.
- ► Add an FX3-CPU1, FX3-CPU2, or FX3-CPU3 main module and an FX3-XTIO module to station B.
- ► Connect a single-channel robot to output Q1 and a lamp to output Q2 of the FX3-XTIO module of station B.

Configuring the logic for station A

- ► Go to the toolbar and click the button for station A and switch to the **logic editor** for station A.
- ► With the help of the input and output elements connected to the FX3-XTIO module and a **Restart** function block, create the following logic configuration:

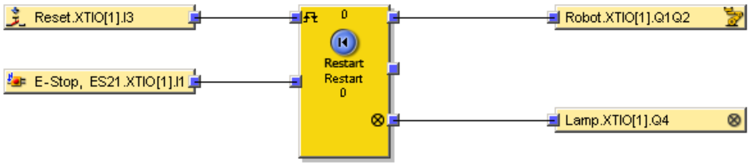
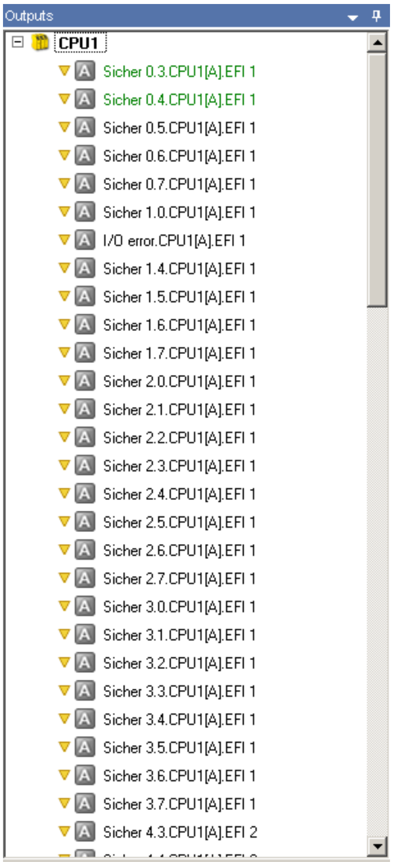


Figure 330: Example for logic configuration (station A)

Configuring the Flexi Link routings for station A

- ► In the Logic editor for station A, add an extra **Routing n:n** function block; configure it so that it has two inputs and outputs, and then connect its inputs to the inputs of the FX3-XTIO module for the reset button and emergency stop pushbutton.
- ► Drag two main module outputs of station A from the outputs selection window under **CPU1** or **CPU2** or **CPU3** to the workspace of the logic editor.



Outputs

CPU1

- Sicher 0.3.CPU1[A].EFI 1
- Sicher 0.4.CPU1[A].EFI 1
- Sicher 0.5.CPU1[A].EFI 1
- Sicher 0.6.CPU1[A].EFI 1
- Sicher 0.7.CPU1[A].EFI 1
- Sicher 1.0.CPU1[A].EFI 1
- I/O error.CPU1[A].EFI 1
- Sicher 1.4.CPU1[A].EFI 1
- Sicher 1.5.CPU1[A].EFI 1
- Sicher 1.6.CPU1[A].EFI 1
- Sicher 1.7.CPU1[A].EFI 1
- Sicher 2.0.CPU1[A].EFI 1
- Sicher 2.1.CPU1[A].EFI 1
- Sicher 2.2.CPU1[A].EFI 1
- Sicher 2.3.CPU1[A].EFI 1
- Sicher 2.4.CPU1[A].EFI 1
- Sicher 2.5.CPU1[A].EFI 1
- Sicher 2.6.CPU1[A].EFI 1
- Sicher 2.7.CPU1[A].EFI 1
- Sicher 3.0.CPU1[A].EFI 1
- Sicher 3.1.CPU1[A].EFI 1
- Sicher 3.2.CPU1[A].EFI 1
- Sicher 3.3.CPU1[A].EFI 1
- Sicher 3.4.CPU1[A].EFI 1
- Sicher 3.5.CPU1[A].EFI 1
- Sicher 3.6.CPU1[A].EFI 1
- Sicher 3.7.CPU1[A].EFI 1
- Sicher 4.3.CPU1[A].EFI 2

Figure 331: Outputs of the main module in the output selection window

NOTE

- A square containing a letter from A to D designates a bit within the Flexi Link process image.
- Each output can only be used once. Outputs that have already been used are displayed in green.

► Connect the outputs of the **Routing n:n** function block to the two CPU1[A] outputs (e.g., **Safe 0.3.CPU1[A].EFI1** and **Safe 0.4.CPU1[A].EFI1**, see figure 332, page 460).

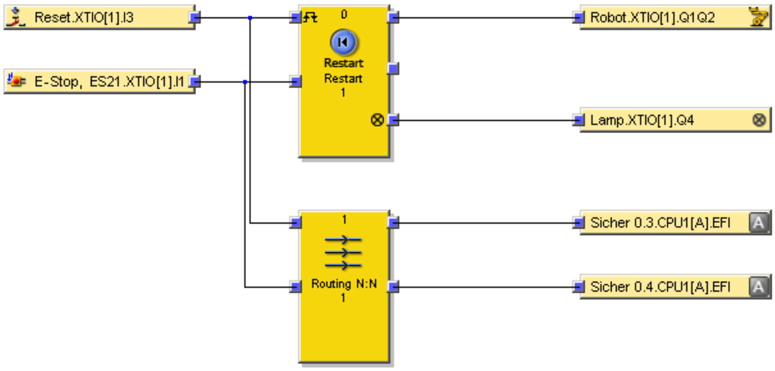


Figure 332: Logic example for Flexi Link routing

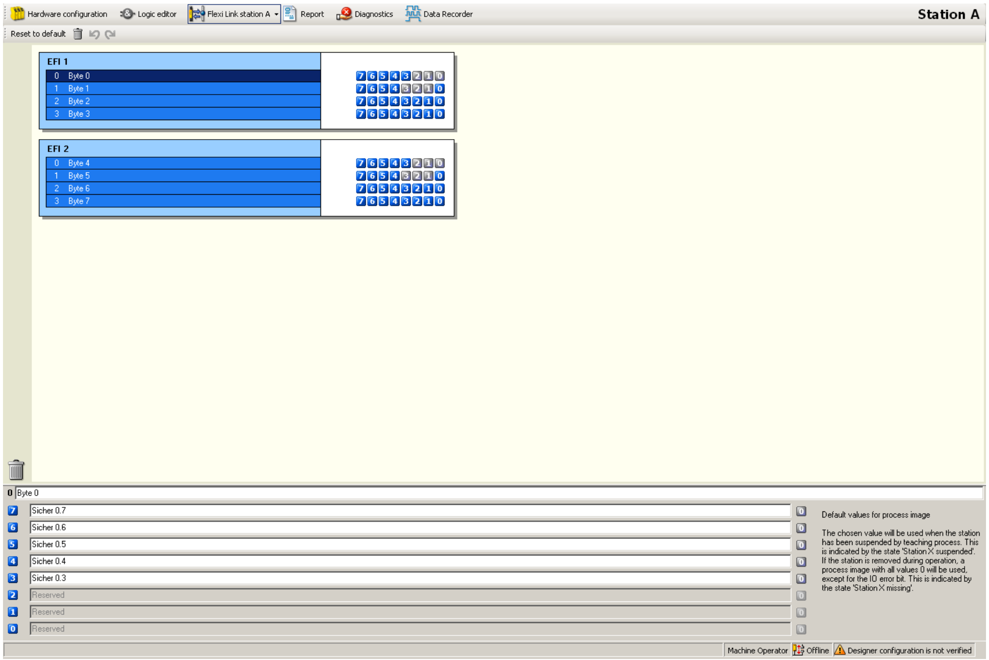
► Note which input is routed to which output. In the example shown, the reset on **Reset.XTIO[1].I2** is routed to **Safe 0.3.CPU1[A].EFI 1** and the emergency stop on **Not-Halt, ES21.XTIO[1].I1** is routed to **Safe 0.4.CPU1[A].EFI 1**.

NOTE

In complex projects, we recommend that routing connections be configured on a separate page of the logic editor.

Assigning tag names for Flexi Link routing

- ► In the view for station A, select the **Interfaces** button followed by **Flexi Link station A** to switch to the Flexi Link routing table.
- ► Click **Byte 0** in the **EFI1** area to display the tag names for byte 0 and the associated bits in the lower half of the window.



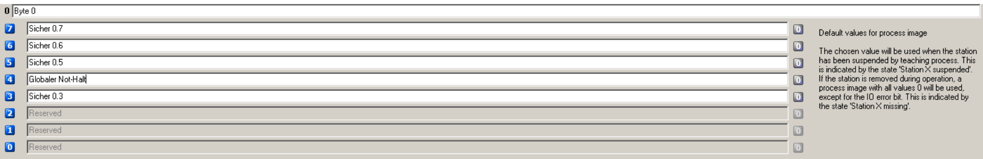
0	Byte 0	
7	Sicher 0.7	0
6	Sicher 0.6	0
5	Sicher 0.5	0
4	Sicher 0.4	0
3	Sicher 0.3	0
2	Reserved	0
1	Reserved	0
0	Reserved	0

Default values for process image

The chosen value will be used when the station has been suspended by teaching process. This is indicated by the state 'Station X suspended'. If the station is removed during operation, a process image with all values 0 will be used, except for the IO error bit. This is indicated by the state 'Station X missing'.

Figure 333: Flexi Link routing table and tag names

- ► Replace the default tag names (e.g., **Safe 0.3** and **Safe 0.4**) with meaningful tag names (e.g., **Global reset** and **Global emergency stop**). The newly assigned tag names are then displayed in the logic editor.



0	Byte 0	
7	Sicher 0.7	0
6	Sicher 0.6	0
5	Sicher 0.5	0
4	Globaler Not-Halt	0
3	Sicher 0.3	0
2	Reserved	0
1	Reserved	0
0	Reserved	0

Default values for process image

The chosen value will be used when the station has been suspended by teaching process. This is indicated by the state 'Station X suspended'. If the station is removed during operation, a process image with all values 0 will be used, except for the IO error bit. This is indicated by the state 'Station X missing'.

Figure 334: Assigned tag names in the routing table

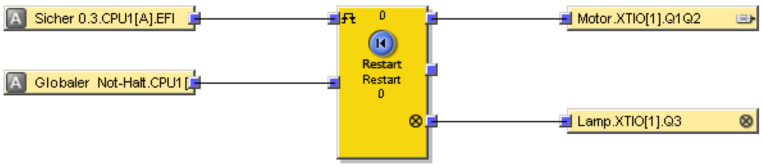
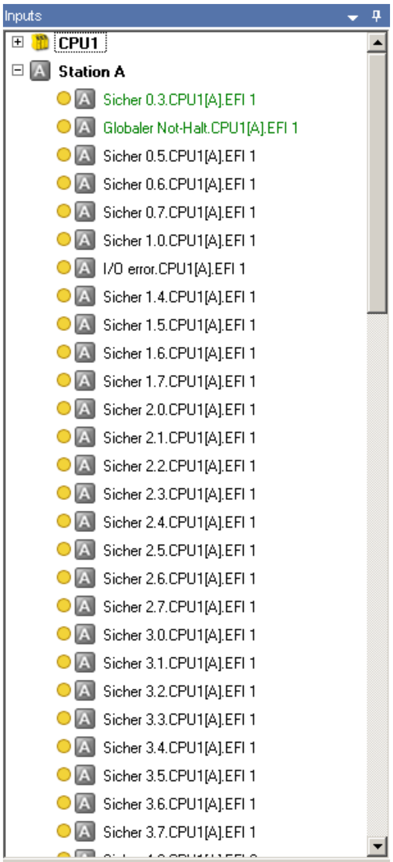


Figure 335: Assigned tag names in the logic editor

NOTE
Default tag names can also be used for the routing. However, you can maintain a clearer overview of your projects and avoid confusion by using meaningful tag names.

Configuring the logic for station B

- ► Go to the toolbar and click the button for station B and switch to the **logic editor** for station B.
- ► In the input selection window, look for the two inputs of the station A main module that are to be routed via Flexi Link. These can be recognized by their tag names:



Inputs

- CPU1
- Station A
  - Sicher 0.3.CPU1[A].EFI 1
  - Globaler Not-Halt.CPU1[A].EFI 1
  - Sicher 0.5.CPU1[A].EFI 1
  - Sicher 0.6.CPU1[A].EFI 1
  - Sicher 0.7.CPU1[A].EFI 1
  - Sicher 1.0.CPU1[A].EFI 1
  - I/O error.CPU1[A].EFI 1
  - Sicher 1.4.CPU1[A].EFI 1
  - Sicher 1.5.CPU1[A].EFI 1
  - Sicher 1.6.CPU1[A].EFI 1
  - Sicher 1.7.CPU1[A].EFI 1
  - Sicher 2.0.CPU1[A].EFI 1
  - Sicher 2.1.CPU1[A].EFI 1
  - Sicher 2.2.CPU1[A].EFI 1
  - Sicher 2.3.CPU1[A].EFI 1
  - Sicher 2.4.CPU1[A].EFI 1
  - Sicher 2.5.CPU1[A].EFI 1
  - Sicher 2.6.CPU1[A].EFI 1
  - Sicher 2.7.CPU1[A].EFI 1
  - Sicher 3.0.CPU1[A].EFI 1
  - Sicher 3.1.CPU1[A].EFI 1
  - Sicher 3.2.CPU1[A].EFI 1
  - Sicher 3.3.CPU1[A].EFI 1
  - Sicher 3.4.CPU1[A].EFI 1
  - Sicher 3.5.CPU1[A].EFI 1
  - Sicher 3.6.CPU1[A].EFI 1
  - Sicher 3.7.CPU1[A].EFI 1

Figure 336: Routed inputs of station A in the logic editor for station B

- ► With the help of these inputs, the output elements connected to the FX3-XTIO module of station B, and a **Restart** function block, create the following logic configuration:

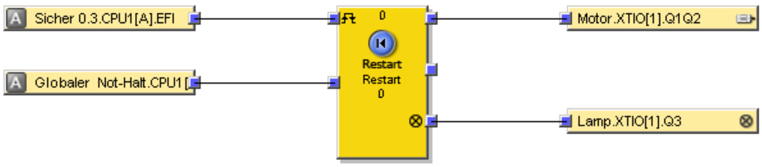


Figure 337: Example for logic configuration (station B)

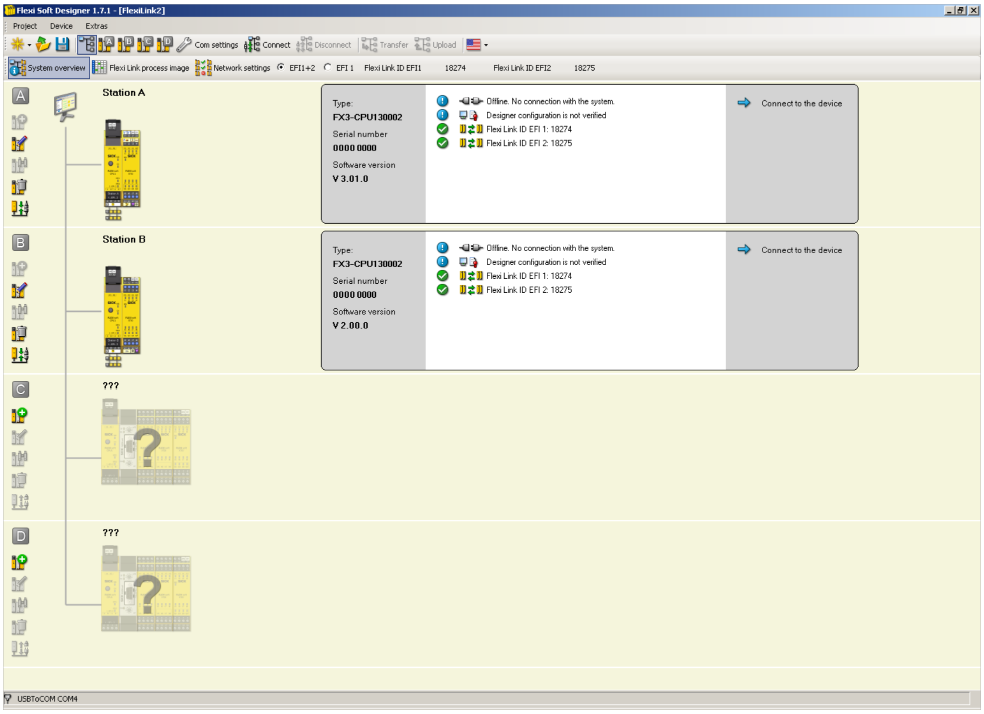
The example project is now complete. The input signals of the emergency stop pushbutton and the reset on station A are routed to station B via Flexi Link so that the robots that are connected to both stations can be controlled simultaneously.

14.3.4 Transferring and verifying the Flexi Link configuration

To start up the Flexi Link system, the final step is to connect the computer to the system, and transfer and verify the configuration. The stations can now be started. To do so, the configuration must be complete, and the necessary Flexi Soft modules and other hardware must have been set up and connected.

Establishing a connection to the Flexi Link system

- ► Connect a computer to the RS-232 interface or the USB interface of any main module within the system.
- ► Switch on the Flexi Link system.
- ► Start the Flexi Soft Designer configuration software that has been installed on the computer and load the project file that contains the configuration.
- ► If necessary, adjust the configuration settings (see "Editing communication settings", page 28).
- ► Switch to the Flexi Link **System overview**. The configured stations within the project are displayed against a light yellow background.



● EFI1+2 ○ EFI 1 Flexi Link ID EFI1 18274 Flexi Link ID EFI2 18275

Station A

Type: FX3-CPU130002
Serial number 0000 0000
Software version V 3.01.0

Offline. No connection with the system.
Designer configuration is not verified
Flexi Link ID EFI 1: 18274
Flexi Link ID EFI 2: 18275

Connect to the device

Station B

Type: FX3-CPU130002
Serial number 0000 0000
Software version V 2.00.0

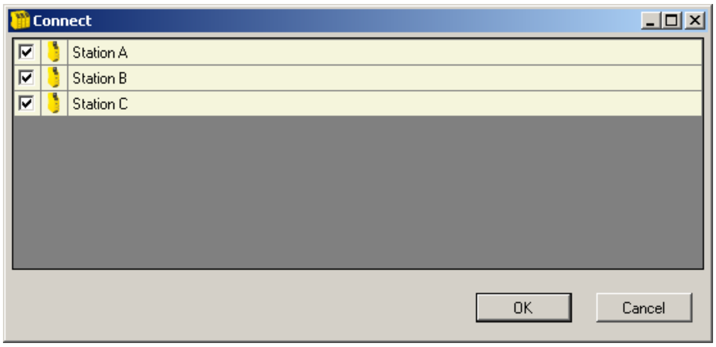
Offline. No connection with the system.
Designer configuration is not verified
Flexi Link ID EFI 1: 18274
Flexi Link ID EFI 2: 18275

Connect to the device

USBToCOM COM4

Figure 338: Flexi Link system overview, system not connected

- ► Click **Connect** and select the station to which a connection is to be established.
- ► Activate all stations then click **OK**.

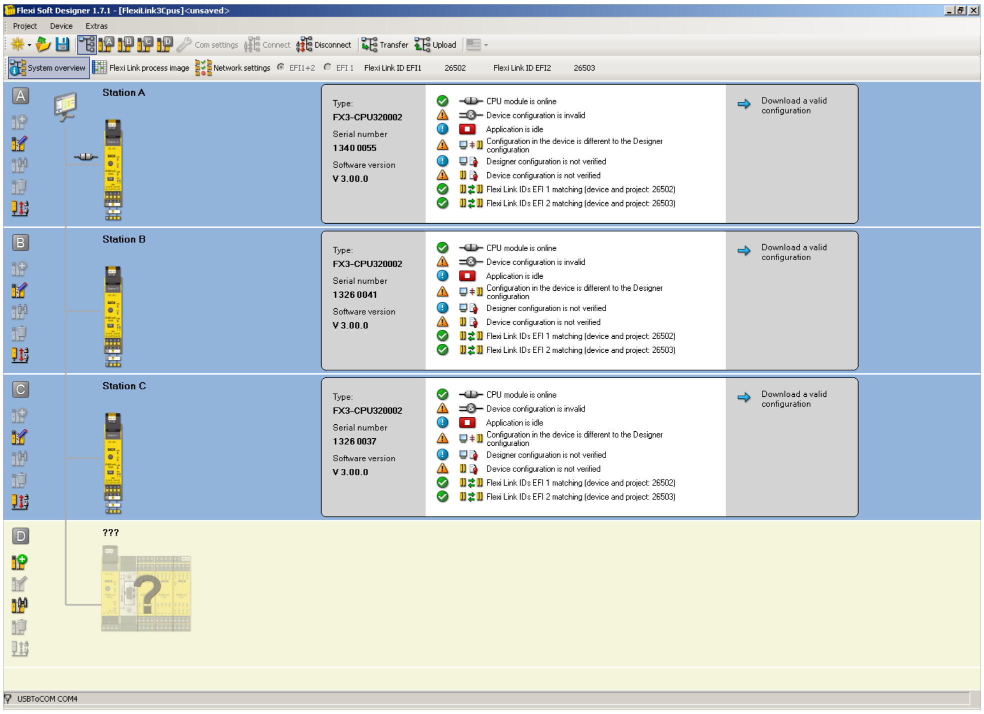


☑ Station A
☑ Station B
☑ Station C

Figure 339: Connect dialog box

- ► If the main modules of the connected stations have not yet been configured for Flexi Link, Flexi Soft Designer will now open the Flexi Link **Network settings** with a list of all the stations found. In this case, a unique Flexi Link address from A through D must be assigned to each individual station (see "Connecting to an existing hardware system", page 454, step 2).

Flexi Soft Designer connects to the Flexi Link system, compares the existing hardware and software configuration with the configuration that is currently stored within Flexi Soft Designer, and displays the results. If the configuration in Flexi Soft Designer is not identical to the configuration of the connected stations, these are displayed against a blue background.



Flexi Link ID EFI1 26502 Flexi Link ID EFI2 26503

Station A

Type: FX3-CPU320002
Serial number 1340 0055
Software version V 3.00.0

- CPU module is online
- Device configuration is invalid
- Application is idle
- Configuration in the device is different to the Designer configuration
- Designer configuration is not verified
- Device configuration is not verified
- Flexi Link IDs EFI 1 matching (device and project: 26502)
- Flexi Link IDs EFI 2 matching (device and project: 26503)

Download a valid configuration

Station B

Type: FX3-CPU320002
Serial number 1326 0041
Software version V 3.00.0

- CPU module is online
- Device configuration is invalid
- Application is idle
- Configuration in the device is different to the Designer configuration
- Designer configuration is not verified
- Device configuration is not verified
- Flexi Link IDs EFI 1 matching (device and project: 26502)
- Flexi Link IDs EFI 2 matching (device and project: 26503)

Download a valid configuration

Station C

Type: FX3-CPU320002
Serial number 1326 0037
Software version V 3.00.0

- CPU module is online
- Device configuration is invalid
- Application is idle
- Configuration in the device is different to the Designer configuration
- Designer configuration is not verified
- Device configuration is not verified
- Flexi Link IDs EFI 1 matching (device and project: 26502)
- Flexi Link IDs EFI 2 matching (device and project: 26503)

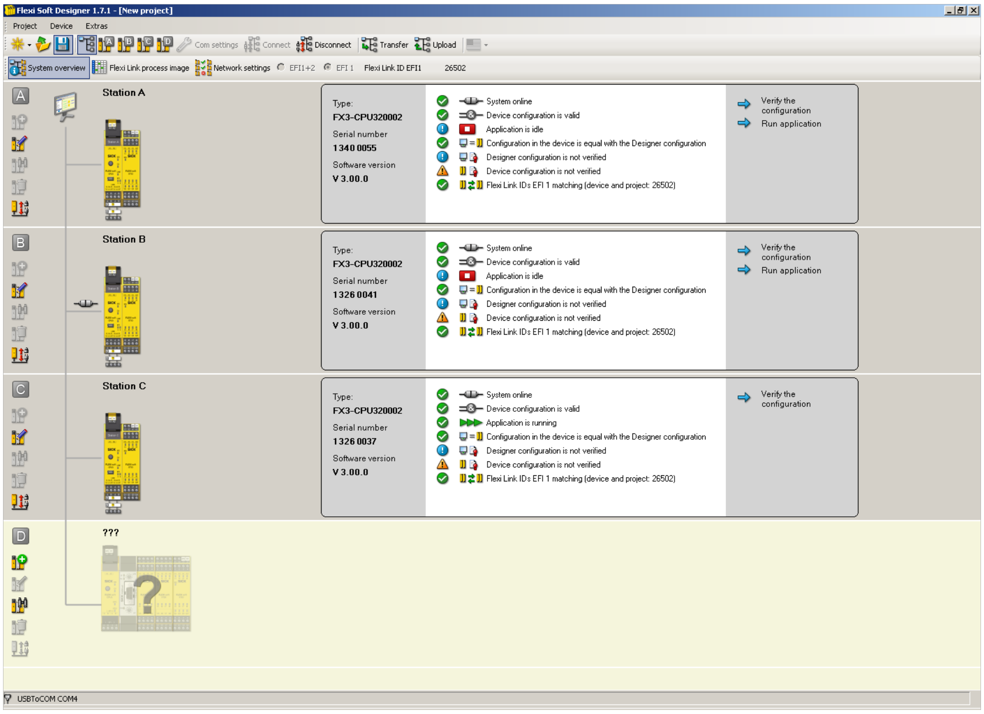
Download a valid configuration

Figure 340: Flexi Link system overview, system connected, non-matching configurations

Transferring the configuration

- ► Click **Transfer**.
- ► Select all stations and click **OK**. Flexi Soft Designer now transfers the configuration to all the stations. The Authorized client password is required for this. The default password is “SICKSAFE”.
- ► As soon as a valid configuration has been successfully transferred to a station, the program asks whether to start up the station concerned.

Regardless of your decision, the station is then displayed in the Flexi Link **System overview** against a gray background. This indicates that the configuration within the station is identical to the one in Flexi Soft Designer.



Station A

Type:
FX3-CPU320002
Serial number
1340 0055
Software version
V 3.00.0

- ✓ System online
- ✓ Device configuration is valid
- ! Application is idle
- ✓ Configuration in the device is equal with the Designer configuration
- ! Designer configuration is not verified
- ⚠ Device configuration is not verified
- ✓ Flexi Link IDs EFI 1 matching (device and project: 26502)

Verify the configuration
Run application

Station B

Type:
FX3-CPU320002
Serial number
1326 0041
Software version
V 3.00.0

- ✓ System online
- ✓ Device configuration is valid
- ! Application is idle
- ✓ Configuration in the device is equal with the Designer configuration
- ! Designer configuration is not verified
- ⚠ Device configuration is not verified
- ✓ Flexi Link IDs EFI 1 matching (device and project: 26502)

Verify the configuration
Run application

Station C

Type:
FX3-CPU320002
Serial number
1326 0037
Software version
V 3.00.0

- ✓ System online
- ✓ Device configuration is valid
- ✓ Application is running
- ✓ Configuration in the device is equal with the Designer configuration
- ! Designer configuration is not verified
- ⚠ Device configuration is not verified
- ✓ Flexi Link IDs EFI 1 matching (device and project: 26502)

Verify the configuration

Figure 341: Flexi Link system overview, system connected, configuration identical but not verified

NOTE

A station can be started using the **Start** button in the hardware configuration for the relevant station.

Verifying the configuration

- ► Switch to the hardware configuration for any of the stations. If the configuration for the devices is valid and identical to the one in Flexi Soft Designer but has not yet been verified, the CV LED will flash on the main module along with the **Receive and compare configuration** button on the left-hand side of the configuration area.
- ► Click the **Receive and compare configuration** button. Flexi Soft Designer now loads the configuration from the main module and compares it with the configuration in the software. If the two configurations are identical, the result is displayed as a report and the program asks whether to set the configuration in the device to **Verified**.
- ► Carefully check the report.

NOTE

The report can be saved as a PDF or printed.

- ► To verify the configuration in the device, click **Yes**. If the device is not running already, the program now asks whether to start it.
- ► Repeat this procedure for each of the stations in the Flexi Link system.

For detailed information about transferring and verifying a configuration, see "Transferring the configuration", page 493

14.4 Flexi Link functions

This chapter provides an overview of the Flexi Link functions that are available in Flexi Soft Designer. Some of these functions relate to the Flexi Link system as a whole while others concern the individual stations within a Flexi Link system. The buttons in the toolbar can be used to switch between the overall system and the individual stations.

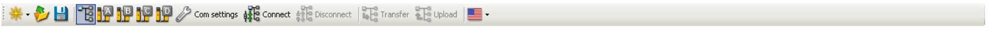
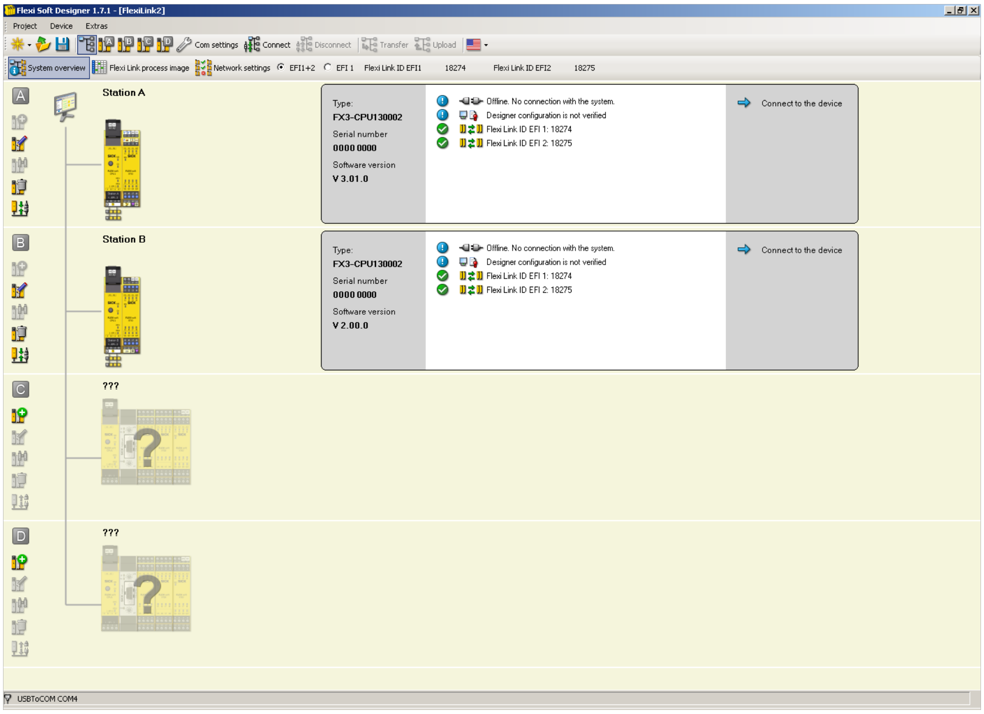


Figure 342: Buttons for the Flexi Link system overview and stations

Flexi Link system functions

In the Flexi Link system overview, you can switch between the views for the various Flexi Link system functions.



● EFI1+2 ○ EFI 1 Flexi Link ID EFI1 18274 Flexi Link ID EFI2 18275

Station A

Type: FX3-CPU130002
Serial number 0000 0000
Software version V 3.01.0

Offline. No connection with the system.
Designer configuration is not verified
Flexi Link ID EFI 1: 18274
Flexi Link ID EFI 2: 18275

Connect to the device

Station B

Type: FX3-CPU130002
Serial number 0000 0000
Software version V 2.00.0

Offline. No connection with the system.
Designer configuration is not verified
Flexi Link ID EFI 1: 18274
Flexi Link ID EFI 2: 18275

Connect to the device

Figure 343: Buttons for Flexi Link system functions

- The Flexi Link **System overview** provides information about the configured/connected stations and their status (see "Flexi Link system: System overview", page 468).
- The **Flexi Link process image** can be used to monitor the information that is exchanged between the connected Flexi Link stations (see "Flexi Link system: Process image", page 469).
- The Flexi-Link **Network settings** view can be used to search the Flexi Link system for connected stations. It is also possible to display and assign the **Flexi Link addresses** of the connected stations, and to allocate the stations to their individual positions (A to D) within the Flexi Link system (see "Flexi Link system: Network settings", page 470).
- The **EFI1+2** and **EFI1** options are used to specify the connection type, i.e. whether only one EFI connection or both EFI connections should be used.
- On the right, you will see the **Flexi Link IDs** of the current configuration within Flexi Soft Designer.

Flexi Link station functions

The Flexi Link functions of an individual station can be used if the view for the station concerned is open.

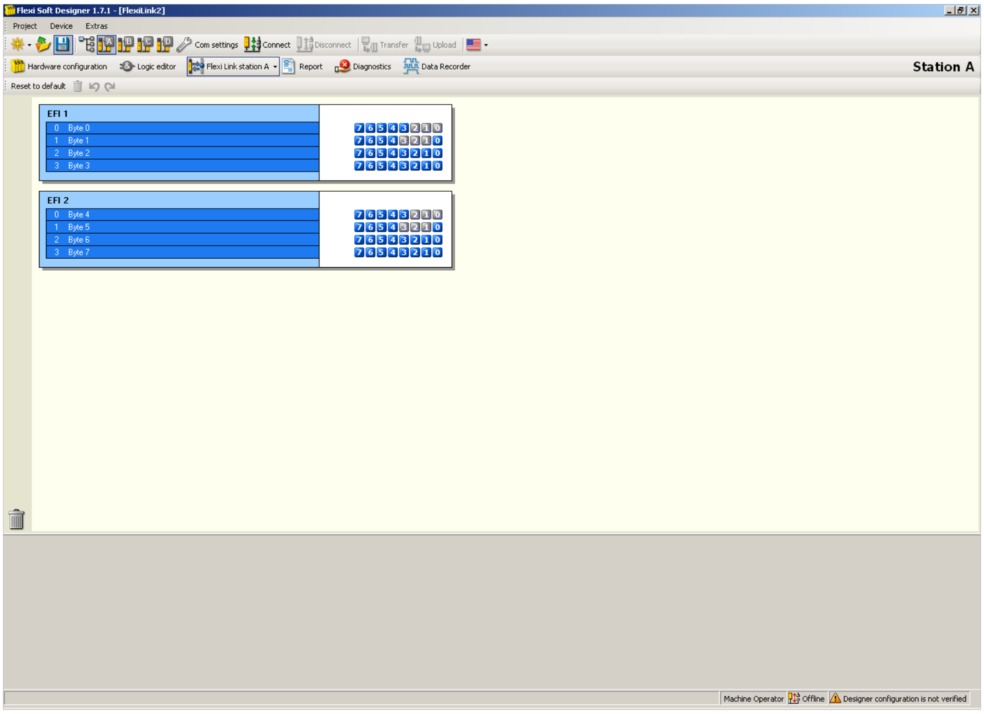


Figure 344: Buttons for Flexi Link station functions

- Most of these functions are the same as for a single-station project, e.g., **Hardware configuration**, **Report**, or **Diagnostics**. This chapter only describes the additional functions that are specific to Flexi Link.
- The **logic editor** can be used to configure what information each station should send to the other stations within the Flexi Link system. The information that the other Flexi Link stations provide on the network is also available here so that it can be used as inputs for the logic program (see "Flexi Link stations: Flexi Link data in the logic editor", page 472).
- The **Flexi Link station X** view can be used to assign tag names to the data that the respective station sends to the Flexi Link system, and to change the defaults for the values in the process image (1 or 0). These values are utilized when the **Teach function** is used to simulate the presence of the station concerned (see "Flexi Link stations: Station X view and process image", page 474 and see "Flexi Link stations: Teach function", page 476).

If the station features a gateway, the **Flexi Link station X** button is displayed in the **Interfaces** menu.

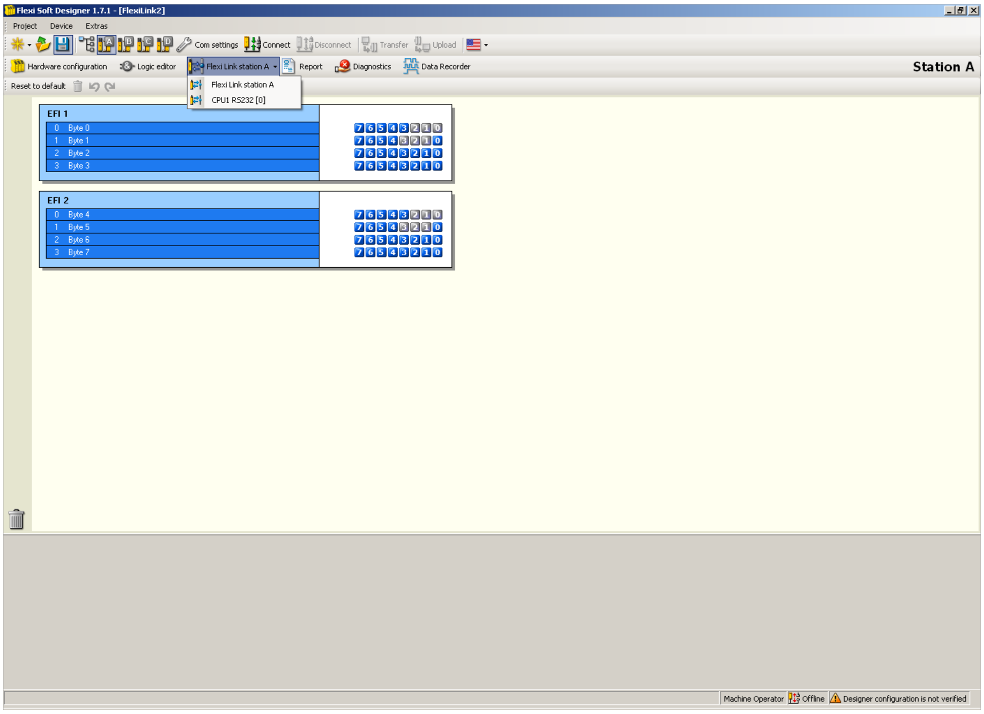
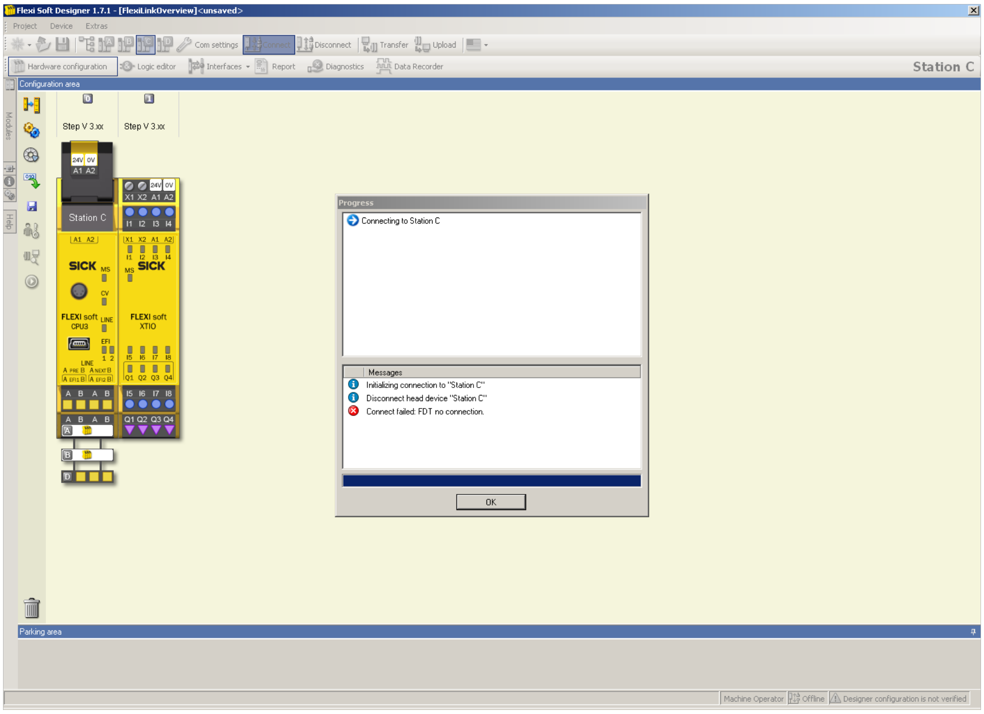


Figure 345: Buttons for a Flexi Link station featuring a gateway

14.4.1 Flexi Link system: System overview

The Flexi Link **System overview** provides information about the configured/connected stations and their status. To open the system overview, start by clicking the **Flexi Link system overview** button on the toolbar and then click the **System overview** button.



Progress

Connecting to Station C

Messages

Initializing connection to "Station C"
Disconnect head device "Station C"
Connect failed: FDT no connection.

OK

Figure 346: Flexi Link system overview

This view shows each station with its current hardware configuration, information about its main module, the status (online or offline), the status of the configuration, and the associated Flexi Link IDs. The background color of each station likewise indicates its status as well as that of its configuration (see figure 346, page 468).

In the example shown, station A is offline (light yellow background), station B is online with a valid configuration (gray background), station C is online with an invalid configuration (blue background), and no station D has been configured (light yellow background).

To the left of each station there are buttons for the following functions:

Table 225: Functions in the Flexi Link system overview

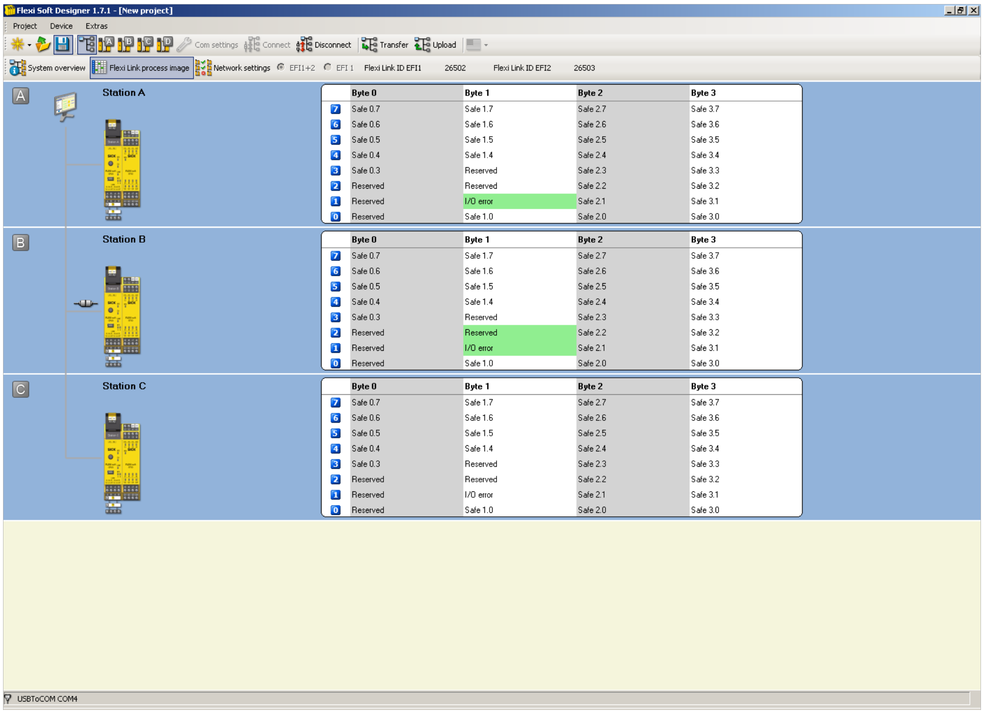
Button	Function	Description
	Add	Inserts a new station into the relevant row (A to D) and switches to the view for the new station. This function is only available if no station has yet been added to the particular row concerned.
	Edit	Switches to the view for the station concerned. This is where the station properties (including the name of the station) can be edit, the connected hardware configured, the logic programmed, the configuration verified, and the configuration protected so that it cannot be overwritten, etc.
	Detect	Connects Flexi Soft Designer to the station concerned, loads its hardware configuration, and then asks whether to load the software configuration for this station.
	Delete	Deletes the station concerned from the system. **Note:** This command is performed without any further prompts and cannot be undone. Any unsaved changes will be lost.
	Connect	Connects Flexi Soft Designer to the station concerned so that the configuration can be transferred, loaded or verified, and the application started or stopped, etc.
	Disconnect	Breaks the connection to the station concerned to enable, for example, the configuration to be edited.

NOTE

- If a function is not available, the relevant button is displayed in gray.
- Instead of using the **Add** or **Edit** buttons, it is also possible to switch to the view for an individual station by clicking the button for station A to D or by double-clicking on a station.
- The **Connect** and **Disconnect** buttons to the left of each station only affect the station concerned. In contrast, the **Connect** and **Disconnect** buttons in the menu of the Flexi Link system overview can be used to connect to/disconnect from all the stations in the Flexi Link system at once. After clicking on the **Connect** button in the menu, a dialog is displayed for selecting the station to which a connection is to be made.
- It is not possible to start or stop all the stations at the same time in the Flexi Link system overview. This must be done for each station individually.

14.4.2 Flexi Link system: Process image

The **Flexi Link process image** shows the information that is exchanged between the Flexi Link stations. The left-hand section of the screen shows the hardware of each station while the right-hand side contains the bits for EFI1 and (if used) EFI2 plus their respective tag names. Any bits that are set to 1 in the current process image are highlighted in green.



Flexi Soft Designer 1.7.1 - [New project]

● EFI1+2 ○ EFI 1 Flexi Link ID EFI1 26502 Flexi Link ID EFI2 26503

	Byte 0	Byte 1	Byte 2	Byte 3
7	Safe 0.7	Safe 1.7	Safe 2.7	Safe 3.7
6	Safe 0.6	Safe 1.6	Safe 2.6	Safe 3.6
5	Safe 0.5	Safe 1.5	Safe 2.5	Safe 3.5
4	Safe 0.4	Safe 1.4	Safe 2.4	Safe 3.4
3	Safe 0.3	Reserved	Safe 2.3	Safe 3.3
2	Reserved	Reserved	Safe 2.2	Safe 3.2
1	Reserved	I/O error	Safe 2.1	Safe 3.1
0	Reserved	Safe 1.0	Safe 2.0	Safe 3.0

	Byte 0	Byte 1	Byte 2	Byte 3
7	Safe 0.7	Safe 1.7	Safe 2.7	Safe 3.7
6	Safe 0.6	Safe 1.6	Safe 2.6	Safe 3.6
5	Safe 0.5	Safe 1.5	Safe 2.5	Safe 3.5
4	Safe 0.4	Safe 1.4	Safe 2.4	Safe 3.4
3	Safe 0.3	Reserved	Safe 2.3	Safe 3.3
2	Reserved	Reserved	Safe 2.2	Safe 3.2
1	Reserved	I/O error	Safe 2.1	Safe 3.1
0	Reserved	Safe 1.0	Safe 2.0	Safe 3.0

	Byte 0	Byte 1	Byte 2	Byte 3
7	Safe 0.7	Safe 1.7	Safe 2.7	Safe 3.7
6	Safe 0.6	Safe 1.6	Safe 2.6	Safe 3.6
5	Safe 0.5	Safe 1.5	Safe 2.5	Safe 3.5
4	Safe 0.4	Safe 1.4	Safe 2.4	Safe 3.4
3	Safe 0.3	Reserved	Safe 2.3	Safe 3.3
2	Reserved	Reserved	Safe 2.2	Safe 3.2
1	Reserved	I/O error	Safe 2.1	Safe 3.1
0	Reserved	Safe 1.0	Safe 2.0	Safe 3.0

USBToCOM COM4

Figure 347: Flexi Link process image

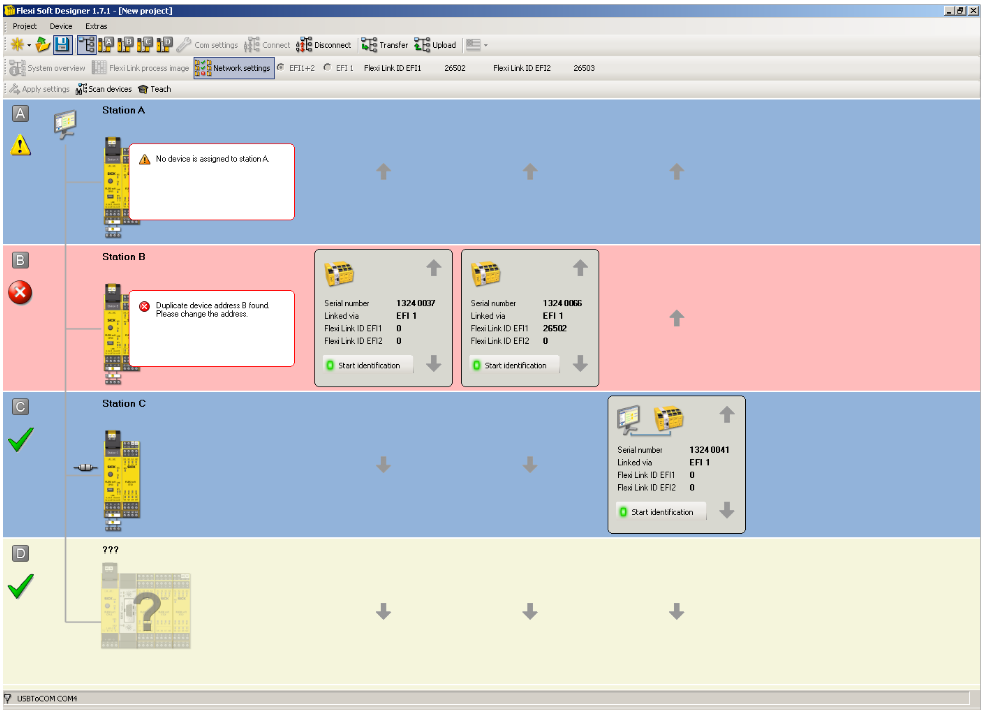
NOTE

- If a station is not running, its process image is set to 0 and its I/O error status bit is set to 1 (see "EFI I/O error status bits in the logic editor", page 96).
- By double-clicking the hardware symbol for a station, the routing view for the station concerned can be opened, and then the tag names of the bits and bytes that the station sends to the network edited (see "Flexi Link stations: Flexi Link data in the logic editor", page 472).

14.4.3 Flexi Link system: Network settings

Under **Network settings**, a Flexi Link address (A, B, C, or D) can be assigned to the individual stations within the Flexi Link system. This is vital from a configuration perspective because Flexi Soft Designer can only address the individual stations via the Flexi Link address, which is also its sole means of identifying the bits in the Flexi Link process image (e.g., station A, EFI1, byte 0, bit 0).

The **Network settings** open automatically when Flexi Soft Designer is connected to a Flexi Link system and the software detects an address assignment error, e.g., if two or more connected stations have the same Flexi Link address. This may happen, for example, if a Flexi Link system is created with new main modules, or if one or more of the main modules are replaced within an existing system.



No device is assigned to station A.

Duplicate device address B found.
Please change the address.

Serial number 1324 0037
Linked via EFI 1
Flexi Link ID EFI1 0
Flexi Link ID EFI2 0

Serial number 1324 0066
Linked via EFI 1
Flexi Link ID EFI1 26502
Flexi Link ID EFI2 0

Serial number 1324 0041
Linked via EFI 1
Flexi Link ID EFI1 0
Flexi Link ID EFI2 0

Figure 348: Flexi Link network settings

If at least one station within a Flexi Link system is online, all the connected stations are displayed together with their currently assigned addresses (addresses A to D). In addition, the system plug serial number and the current Flexi Link IDs for EFI1 and EFI2 are displayed for each station. To refresh this information, click the **Search for devices** button on the top left-hand side of the window. Error messages and warnings concerning the current system status are displayed as a pop-up for each station.

NOTE

- There are two ways to identify a station:
  - On one of the displayed stations, click the **Start identification** button. The MS and EFI1 LEDs of the corresponding station start to flash alternately with the EFI2 LED (2 Hz). The Authorized client password is required for this. The default password is “SICKSAFE”. To stop the LEDs flashing, click the button again (which is now called **Stop identification**).
  - Check the serial number on the system plug and compare it with the serial number displayed in Flexi Soft Designer. The serial number displayed under the **Network settings** is the serial number of the system plug and not the one for the main module.

Changing the assigned Flexi Link address (A to D)

► Move the station to the required position within the window either by using the Up and Down arrows, or by dragging the station using the mouse.

Figure 349: Apply settings button

► In the upper left corner of the screen, click on the **Apply settings** button. The Flexi Link addresses of the stations are changed.

ℹ **NOTE**

- The **Apply settings** button does not affect the Flexi Link IDs of the stations. The Flexi Link IDs are transferred to the stations as part of the configuration. Therefore, if there is a change in the Flexi Link IDs as a result of changes to the configuration of any station, the configuration must be transferred to all the stations again so that the new Flexi Link IDs are assigned.
- It does not matter which Flexi Link address is assigned to which station. However, for the sake of clarity, we recommend that you put them in the same order that they were installed in the control cabinet, working from left to right.
- If the addresses assigned within a Flexi Link system are changed at a later time, it may be necessary to reconfigure the process image as well as those parts of the programmed logic that use input bits from the Flexi Link process image. The Flexi Link address forms part of how the bits are assigned in the process image.

14.4.4 Flexi Link stations: Flexi Link data in the logic editor

In the logic editor, the information that is available within the Flexi Link system is processed centrally. Flexi Link stations are treated like EFI sensors:

- Each station can use the information from the other stations as input data.
- Each station can make its own data available as output data.

Signal duration

Signals must be present for at least the duration of the logic execution time of the Flexi Link system in order to be reliably detected and transmitted to the other Flexi Link stations.

⚠ **WARNING**

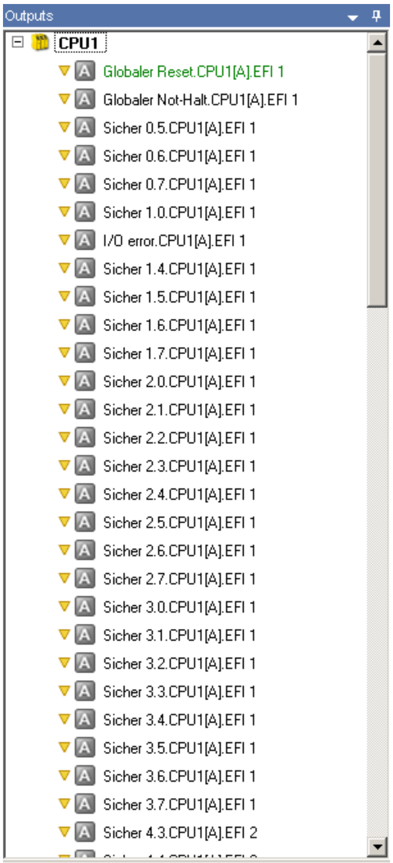
Malfunction due to too short signals

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

► Make sure that all signals remain present long enough to be detected within the Flexi Link system (e.g., by using a function block for delays within the logic program).

Routing data to the Flexi Link system

In order for data to be written to the Flexi Link system so that it can be used by the other stations, it is necessary to determine which bit within the Flexi Link process image needs to be set in each case. The bits that can be configured for each station are listed on the **Outputs** tab of the logic editor under the symbol for the FX3-CPUx main module that is being used:



Outputs

CPU1
- A Globaler Reset.CPU1[A].EFI 1
- A Globaler Not-Halt.CPU1[A].EFI 1
- A Sicher 0.5.CPU1[A].EFI 1
- A Sicher 0.6.CPU1[A].EFI 1
- A Sicher 0.7.CPU1[A].EFI 1
- A Sicher 1.0.CPU1[A].EFI 1
- A I/O error.CPU1[A].EFI 1
- A Sicher 1.4.CPU1[A].EFI 1
- A Sicher 1.5.CPU1[A].EFI 1
- A Sicher 1.6.CPU1[A].EFI 1
- A Sicher 1.7.CPU1[A].EFI 1
- A Sicher 2.0.CPU1[A].EFI 1
- A Sicher 2.1.CPU1[A].EFI 1
- A Sicher 2.2.CPU1[A].EFI 1
- A Sicher 2.3.CPU1[A].EFI 1
- A Sicher 2.4.CPU1[A].EFI 1
- A Sicher 2.5.CPU1[A].EFI 1
- A Sicher 2.6.CPU1[A].EFI 1
- A Sicher 2.7.CPU1[A].EFI 1
- A Sicher 3.0.CPU1[A].EFI 1
- A Sicher 3.1.CPU1[A].EFI 1
- A Sicher 3.2.CPU1[A].EFI 1
- A Sicher 3.3.CPU1[A].EFI 1
- A Sicher 3.4.CPU1[A].EFI 1
- A Sicher 3.5.CPU1[A].EFI 1
- A Sicher 3.6.CPU1[A].EFI 1
- A Sicher 3.7.CPU1[A].EFI 1
- A Sicher 4.3.CPU1[A].EFI 2

Figure 350: Flexi Link output bits of station A in the logic editor

NOTE

- A square containing a letter from A to D designates a bit within the Flexi Link process image.
- Each output bit can only be used once. Any outputs that are already in use are shown in green.
- The tag names of the output bits can be changed in the **Flexi Link station X** view (see "Flexi Link stations: Station X view and process image", page 474).

Sending information to the Flexi Link system

- ► Drag an output bit into the workspace and connect it to an output of a function block.
- ► To make the value of an input directly available to all the stations within a Flexi Link system, use a **Routing 1:n** or **Routing n:n** function block (see figure 351, page 473).

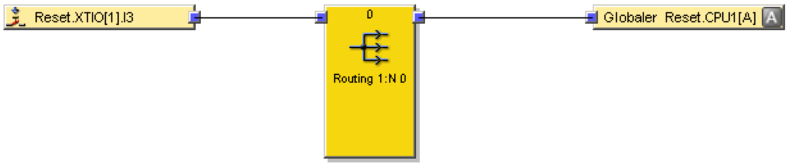
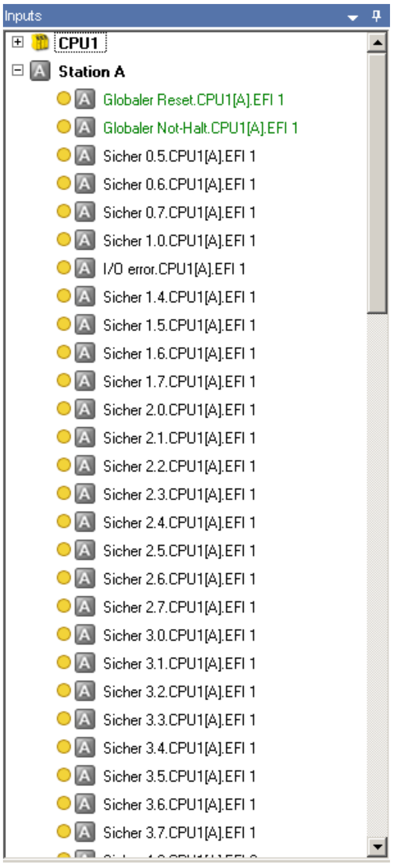


Figure 351: Routing an input to the Flexi Link system

Using data from the Flexi Link system

All the information that is made available by other stations within the Flexi Link system is listed on the **Inputs** tab of the logic editor under the symbol for the station concerned:



Inputs

- CPU1
- Station A
  - Globaler Reset.CPU1[A].EFI 1
  - Globaler Not-Halt.CPU1[A].EFI 1
  - Sicher 0.5.CPU1[A].EFI 1
  - Sicher 0.6.CPU1[A].EFI 1
  - Sicher 0.7.CPU1[A].EFI 1
  - Sicher 1.0.CPU1[A].EFI 1
  - I/O error.CPU1[A].EFI 1
  - Sicher 1.4.CPU1[A].EFI 1
  - Sicher 1.5.CPU1[A].EFI 1
  - Sicher 1.6.CPU1[A].EFI 1
  - Sicher 1.7.CPU1[A].EFI 1
  - Sicher 2.0.CPU1[A].EFI 1
  - Sicher 2.1.CPU1[A].EFI 1
  - Sicher 2.2.CPU1[A].EFI 1
  - Sicher 2.3.CPU1[A].EFI 1
  - Sicher 2.4.CPU1[A].EFI 1
  - Sicher 2.5.CPU1[A].EFI 1
  - Sicher 2.6.CPU1[A].EFI 1
  - Sicher 2.7.CPU1[A].EFI 1
  - Sicher 3.0.CPU1[A].EFI 1
  - Sicher 3.1.CPU1[A].EFI 1
  - Sicher 3.2.CPU1[A].EFI 1
  - Sicher 3.3.CPU1[A].EFI 1
  - Sicher 3.4.CPU1[A].EFI 1
  - Sicher 3.5.CPU1[A].EFI 1
  - Sicher 3.6.CPU1[A].EFI 1
  - Sicher 3.7.CPU1[A].EFI 1

Figure 352: Flexi Link output bits of station A displayed in the logic editor as input bits for another station

These inputs can be used like any other input.

NOTE

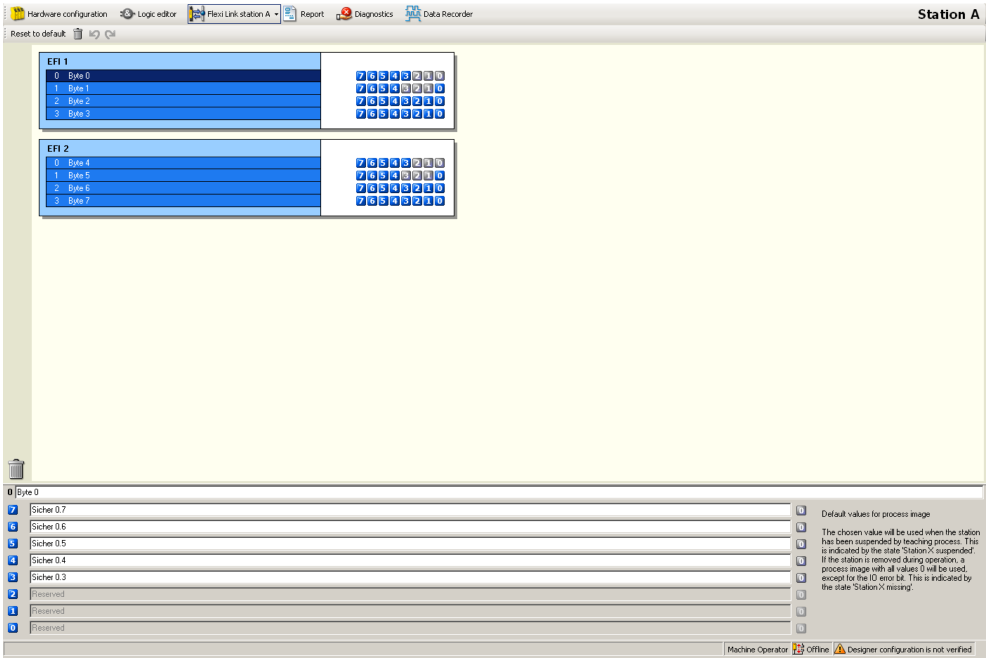
- Inputs can be used multiple times.
- Any inputs that have been used at least once within the logic for the station concerned are shown in green.
- The Flexi Link inputs are displayed with their respective tag names. The tag names can be changed in the **Flexi Link station X** view of the station concerned (see "Flexi Link stations: Station X view and process image", page 474).

14.4.5 Flexi Link stations: Station X view and process image

In the **Flexi Link station X** view, the following actions can be performed:

- Edit the tag names of the bits and bytes that this station sends to the Flexi Link system.
- Set the values for the bits in the process image of this station to 0 or 1 (see "Flexi Link stations: Teach function", page 476).

► Click the corresponding button on the toolbar to open the **Flexi-Link-Station X** view. If the station also features a gateway, this button can be found in the **Interfaces** menu.



EFI 1
0 Byte 0
1 Byte 1
2 Byte 2
3 Byte 3

EFI 2
0 Byte 4
1 Byte 5
2 Byte 6
3 Byte 7

0 Byte 0
7 Sicher 0.7 0
6 Sicher 0.6 0
5 Sicher 0.5 0
4 Sicher 0.4 0
3 Sicher 0.3 0
2 Reserved 0
1 Reserved 0
0 Reserved 0

Default values for process image

The chosen value will be used when the station has been suspended by teaching process. This is indicated by the state 'Station X suspended'. If the station is removed during operation, a process image with all values 0 will be used, except for the IO error bit. This is indicated by the state 'Station X missing'.

Figure 353: View for Flexi Link station A

The toolbar

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

- **Reset to default**: Resets the tag names and default values that have been configured for all bits and bytes to their original settings.
- **Delete selected byte**: Deletes all tag names for the selected byte and its bits. Sets the default value for all bits of the byte to 0.
- **Undo**
- **Redo**

Editing tag names

- ► Click on a byte in the EFI1 or EFI2 area to display its bits in the lower half of the window.
- ► Change the displayed tag names as required. The bits appear with their new tag names on the **Outputs** tab of the logic editor.

NOTE

- Some bits are reserved and cannot be used or edited. These bits are displayed in gray in the top half of the window.
- Tag names can also be deleted. It is not possible to use bits without a tag name. These are displayed in gray in the top half of the window but do not appear in the Flexi Link process image.

Changing the default values

- ► Click on a byte in the EFI1 or EFI2 area to display its bits in the lower half of the window.
- ► There is a button to the right of the tag name input field for each bit. Click this button once to change the default value for the bit concerned (0 or 1).

The set value is used in the process image of the Flexi Link system when the relevant station is suspended (see "Flexi Link stations: Teach function", page 476).

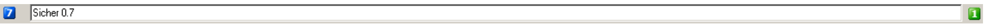

7 Sicher 0.7 1

Figure 354: Default value of a Flexi Link input bit set to 1

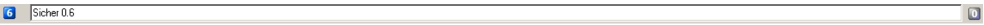


Figure 355: Default value of a Flexi Link input bit set to 0

NOTE
When the value of a bit is changed, this also changes the process image of the configuration and, in turn, the Flexi Link ID for the EFI connection associated with the modified bit. A warning is displayed that the modified configuration needs to be transferred to **all** stations so that the new Flexi Link ID gets properly assigned. Otherwise, communication within the Flexi Link system will be interrupted because the Flexi Link IDs are different (see "Flexi Link ID", page 453 and see "Flexi Link troubleshooting", page 480).

14.4.6 Flexi Link stations: Teach function

The Teach function enables a Flexi Link system to remain in operation even if one or more stations are missing from the system (i.e., if they are switched off). The Teach function suspends the missing stations and their presence is then simulated by the other stations. Each suspended station is treated as if it were online and running. In this case, the Flexi Link process image uses the default values that have been set for the station concerned (see "Flexi Link stations: Station X view and process image", page 474). This may prove useful when setting up a system or carrying out maintenance, for example.

When the Teach function is active on any station and this station is connected to the system and running, the entire system carries out a network scan and all missing stations are treated as suspended. In other words, the system continues working in the same way as if these stations were still online while relying on the default process images for the stations concerned.

If a station is suspended using the Teach function, the safety outputs on the stations that remain active may be set to High under certain circumstances.

WARNING
Limited safety due to Teach function

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

- ► Analyze the application and check whether additional safety measures are required if the Teach function is activated.
- ► The Teach function is to be regarded as a configuration procedure. Consequently, the Teach function must be secured in accordance with the relevant safety requirements, e.g., by using a key switch that has the Teach input connected to it within the logic and by using a Restart function block to monitor the timing conditions.
- ► Determine how the deactivated parts of the machine or plant/system have to be dealt with. Indicate that the control elements and sensors will not have any effect on those parts of the machine that were previously connected to them (for example, attach “Out of service” signs to the emergency stop pushbuttons).
- ► Only authorized and specially trained personnel are permitted to activate the Teach function.
- ► Before using the Teach function, make sure that there is no one inside the hazardous area and that no one can access this area while the Teach function is active.
- ► After using the Teach function, check the safety function of the entire system.

NOTE

- A station is regarded as “missing” and can be suspended if its voltage supply has been switched off or if its EFI connection to the Flexi Link system has been completely broken. It is not possible to suspend a station while it is still connected and if, for example, one of the following conditions applies:
  - The station is not in the Run status.
  - The station has generated an EFI error, e.g., due to an inconsistent Flexi Link ID.
- The Teach function always applies to all the activated stations within a Flexi Link system, not just a single station. Consequently, it may be sufficient to implement the Teach function on merely one of the connected stations. However, if only one station within the system features a Teach pushbutton and is configured accordingly, this station cannot be suspended because it is required to activate the Teach function.
- Each main module within a Flexi Link system signals its current status for the Teach function via status bits, which can be used as inputs in the Logic editor (see "Flexi Link status bits", page 478).

Configuring the Teach function

► Connect a Teach pushbutton to the inputs of each station within the Flexi Link system that you want to support the Teach function. The Teach pushbutton can, for example, take the form of a dual-channel key-operated switch.

► Within the logic for these stations, connect the input of the Teach pushbutton to the Teach output of the relevant station via a **Restart** function block (see figure 356, page 477).

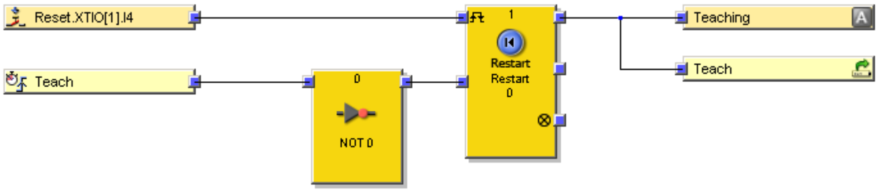


Figure 356: Configuration of the Teach function in the logic editor

When the Teach pushbutton is pressed, the Teach output switches to 1 for the duration of one logic cycle. The rising signal edge (0-1) at the Teach output triggers the Teach function.

NOTE

If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal may produce a pulse if the signal for **Teach** is reset as a result of short-circuit detection.

WARNING

Malfunction of the signals for teaching in the event of a short-circuit to high

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

► Make sure that the transitions of the signals for **Teach** meet the requirements of safety standards and regulations.
  ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines).
  ► No short-circuit detection, i.e., no referencing to test outputs.

Using the Teach function

- ► Switch off the voltage supply for one or more stations (e.g., station C) within a Flexi System that is currently in operation. The system detects that these stations are missing and sets their process image to safe values (0). The remaining stations indicate an EFI error (EFI LEDs flash ● red) and their EFI status bits for the deactivated stations (e.g., **Station C missing**) are set to 1 along with the higher-level **Station missing** status bit.
- ► Now press the Teach pushbutton on one of the remaining stations. The system will now carry on working as if the missing stations were still present. The process image of these stations is replaced, however, by the static values that were previously configured (see "Flexi Link stations: Station X view and process image", page 474). The EFI status bits of the remaining stations now indicate which stations have been suspended (e.g., **Station C missing** and **Station missing** return to 0 and **Station C suspended** switches to 1).
- ► To put a missing and subsequently suspended station back into operation, reconnect its voltage supply. As soon as the station has powered up, the other stations will detect its presence and indicate an EFI error. The EFI status bit (e.g., **Station C suspended** remains set to 1 and the **Suspended stations found** status switches to 1.
- ► Press the Teach pushbutton again. The system reintegrates the previously suspended station and resumes operation. The **Suspended stations found** status bit switches to 0 along with the respective EFI status bits (e.g., **Station C missing** remains set to 0 and **Station C suspended** likewise switches to 0).

NOTE

If a station is missing because its EFI connection has been broken and not because its voltage supply has been switched off, it is likely to be in the error status. In this case, this station must first be reset by interrupting its voltage supply for at least three seconds before it can be reintegrated into the system.

14.4.7 Flexi Link status bits

Each main module within a Flexi Link system uses status bits to signal whether the Teach function is required and which station is missing or has been suspended (= taught). These status bits are available in the **Diagnostics** window of the logic editor as inputs of the main module concerned.

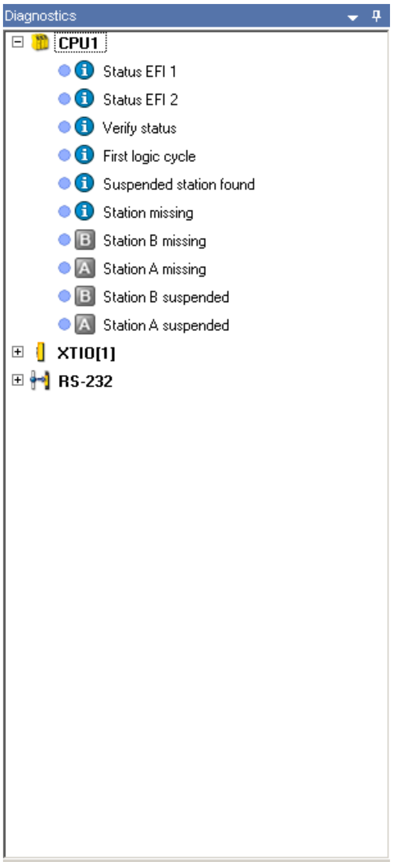


Figure 357: Status information for the Flexi Link system in the logic editor

Table 226: Meaning of the status bits for the Teach function

Status bit of main module	Meaning
Suspended stations found	A station that was previously suspended is now present within the system again. In this case, the process image of the stations that detect the recurring station is set to 0 and the EFI I/O error status bit is set to 1. The Teach function is required for operation to continue. The Teach function resets the I/O error status bit even if the station that was found has been suspended again in the meantime.
Station missing	At least one station is missing from the system. The Teach function is required for operation to continue. This means that at least one of the **Station X missing** status bits (see below) is also set to 1.
Station X missing	The station with Flexi Link address X (= A, B, C, or D) is missing. In this case, the process image of the associated Flexi Link station is set to 0 and the EFI I/O error status bit is set to 1. This means that the **Station missing** status bit is also set to 1.
Station X suspended	The station with Flexi Link address X (= A, B, C, or D) has been suspended. In this case, the default process image for the respective Flexi Link station is used.

These status bits can be used to set up your own diagnostics system, e.g., by connecting them to a **Message generator** function block or by switching on a warning light when the Teach function is required or active.

NOTE
After a transition from the Stop status to the Run status, a station is considered missing unless it is found within the space of three minutes.

Description of further diagnostic bits of the main module: see "Module status bits of the main module", page 92.

14.5 Flexi Link troubleshooting

This chapter covers the processes of diagnosing and correcting malfunctions of the Flexi Link system.

An overview of how the LEDs indicate faults is provided in the “Flexi Soft Modular Safety Controller Hardware” operating instructions (SICK part number 8012999).

14.5.1 Inconsistent Flexi Link ID

Error description

If no process images can be exchanged between the stations in the Flexi Link system and all the main modules are indicating a remediable error (MS LED flashes ☀ red at 1 Hz and the EFI1 and EFI2 LEDs light up ● red), an inconsistent Flexi Link ID may be the cause of the problem. This means that at least one station within the system has one or two Flexi Link IDs that differ from the Flexi Link IDs on the other stations.

Diagnostics

- ► Switch to the Flexi Link **System overview.**
- ► If Flexi Soft Designer is not connected to the system, establish a connection to all the stations.
- ► Check the status messages of the stations to see if they are indicating an inconsistent Flexi Link ID.

Rectification

If there are inconsistent Flexi Link IDs within a system, the current configuration must be transferred to all the stations again.

- ► Check whether the configuration in Flexi Soft Designer is correct.
- ► Establish a connection to all the stations.
- ► Transfer the configuration to all the stations.
- ► Switch to the view for each station and, if required, verifying the configuration.

15 Flexi Line

15.1 Overview of Flexi Line

Flexi Link enables up to 32 Flexi Soft stations to be networked securely. It is only possible to use FX3-CPU3 main modules as part of a Flexi Line system. It is not possible to connect a main module of any other type (FX3-CPU0, FX3-CPU1, FX3-CPU2).

A uniform process image is defined for the entire Flexi Line system. Each byte of this process image either applies globally (i.e., across the entire system) or locally (i.e., merely to the respective station and its neighboring stations). Each Flexi Line station uses this process image to communicate with its neighboring stations. The topology enables communication to take place without the need for addresses.

Features

- Secure connection of up to 32 Flexi Soft stations via the Flexi Line interface
- Addressless topology: If a change is made to the order of the stations, the new layout can be confirmed simply by pressing a Teach pushbutton.
- The EFI interface remains available without any restrictions:
  - EFI-enabled sensors can be connected.
  - A Flexi Link system can be connected.
- A global process image is defined for all stations.
- Globally or locally valid bytes can be defined within the process image.
- The process image can include up to 12 or 96 bits.
- The maximum length of the segment cable between two stations is 1,000 meters. Consequently, a system with 32 stations can have a total length of up to 31 kilometers.

15.1.1 System requirements and restrictions for Flexi Line

The following minimum system requirements must be met for Flexi Line:

Table 227: System requirements for Flexi Line

System component	Version
Hardware	FX3-CPU3
Software	Flexi Soft Designer version ≥ V1.6.0

NOTE

- Flexi Link or EFI communication can be used at the same time as Flexi Line, i.e., it is possible to connect either EFI-enabled devices or Flexi Link stations at the same time.
- The process image is transmitted from one station to another using a fixed send cycle time. However, as the stations are not synchronized with one another, (logic) processing by the individual stations is not necessarily simultaneous.
- The send cycle time of the Flexi Line system depends on the maximum length of the segment cable between two stations and on the size of the process image.

Table 228: Send cycle time of a Flexi Line system according to maximum length of segment cable and size of process image

Max. length of segment cable	32 bits	64 bits	96 bits
125 m	2 ms	2 ms	4 ms
250 m	2 ms	4 ms	8 ms
500 m	4 ms	8 ms	12 ms
1,000 m	8 ms	12 ms	20 ms

15.2 Flexi Line principle of operation

15.2.1 Topology

The individual stations within a Flexi Line system are not identified on the basis of addresses. Instead, each station is connected to its immediate neighbors and so it communicates with both the previous station and the subsequent station.

The positions of the stations within the Flexi Line system must be confirmed during commissioning by carrying out a Teach procedure. This arrangement is then monitored. If a station is replaced, disconnected from the system, or added, the station arrangement must be reconfirmed (see "Teach", page 487).

15.2.2 Flexi Line configuration

The process image lies at the very heart of every Flexi Line system. This process image defines what data gets communicated from one station to another as well as the volume of data, the send cycle time, the range (routing), and the default value to be used (1 or 0). The routing and the default value can be defined separately for each byte.

The process image is usually defined when the first station in the Flexi Line system is configured and is subsequently transferred to the remaining stations.

The **Flexi Line** view can be accessed at any time by clicking the **Interfaces** button on the toolbar. It opens automatically when a Flexi Line element is added to an FX3-CPU3 main module.

The **Flexi Line** view consists of the following elements:

- Flexi Line configuration toolbar with the following functions:
  - If Flexi Soft Designer is not connected to the main module: **Import** a Flexi Line definition and **Export** a Flexi Line definition
  - If Flexi Soft Designer is connected to the main module: **Teach** and **Restart Flexi Line system**
  - **Lock for protection**
- Toolbar for switching between the **General settings**, **Byte configuration**, and **Diagnostics** views
- Configuration areas for the views:
  - **General settings** view: **General Flexi Line information** and **Specification**
  - **Byte configuration** view: **Byte configuration** and **Details and bit configuration**
  - **Diagnostics** view: **Byte overview** and **Details overview**

The Flexi Line configuration view consists of two parts.

- Default settings can be specified under **General settings**. Primarily, this involves creating the required combination of the following: size of process image (number of bits to be transmitted), maximum length of segment cable between two neighboring stations, and send cycle time. The process image can also be assigned a name and revision number here.
- The **Byte configuration** area is used to define the data of the process image. Each byte is assigned a routing direction, a default value, and a name. Similarly, all the bits used can be given a tag name. By deactivating any unused bits, you can prevent them from being shown in the logic and in the Diagnostics view.

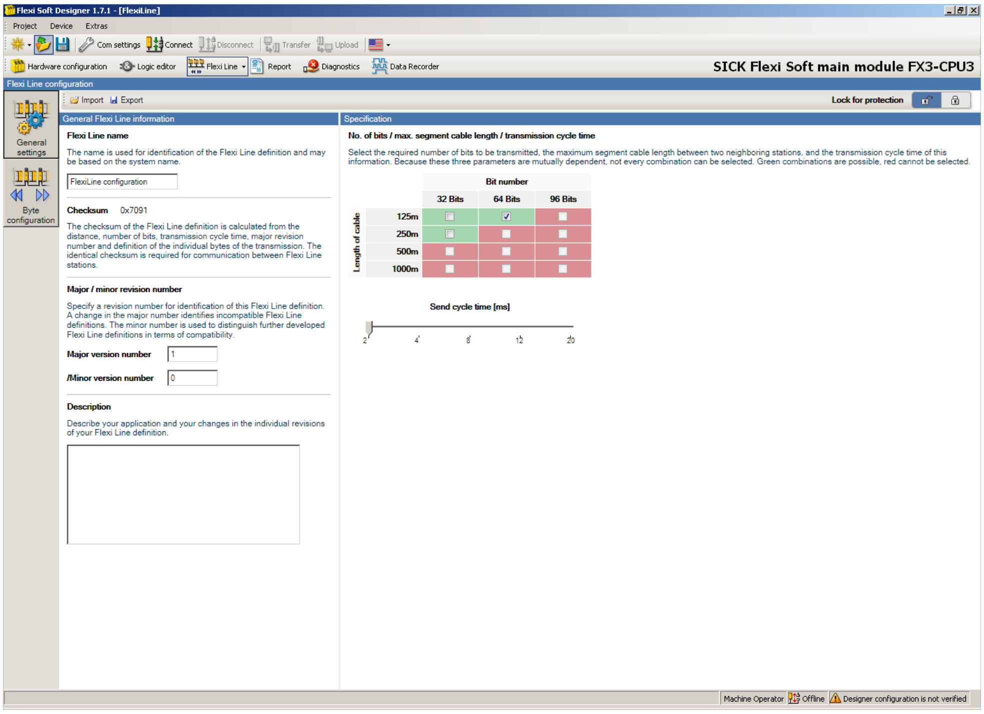
Flexi Line toolbar

The Flexi Line toolbar contains buttons for the following functions:

- Import a previously saved Flexi Line definition
- Export a Flexi Line definition

- Teach: Confirm the topology of the Flexi Line system during commissioning or following changes to the topology
- Restart Flexi Line system: Triggers a restart operation for all Flexi Soft stations in a Flexi Line system
- Lock the configuration: The Flexi Line configuration can be protected against accidental changes by moving the slider.

General settings



Flexi Line name

The name is used for identification of the Flexi Line definition and may be based on the system name.

FlexiLine configuration

Checksum 0x7091

The checksum of the Flexi Line definition is calculated from the distance, number of bits, transmission cycle time, major revision number and definition of the individual bytes of the transmission. The identical checksum is required for communication between Flexi Line stations.

Major / minor revision number

Specify a revision number for identification of this Flexi Line definition. A change in the major number identifies incompatible Flexi Line definitions. The minor number is used to distinguish further developed Flexi Line definitions in terms of compatibility.

Major version number 1

/Minor version number 0

Description

Describe your application and your changes in the individual revisions of your Flexi Line definition.

No. of bits / max. segment cable length / transmission cycle time

Select the required number of bits to be transmitted, the maximum segment cable length between two neighboring stations, and the transmission cycle time of this information. Because these three parameters are mutually dependent, not every combination can be selected. Green combinations are possible, red cannot be selected.

Length of cable	Bit number		
	32 Bits	64 Bits	96 Bits
125m	🟩 □	🟩 ☑	🟥 □
250m	🟩 □	🟥 □	🟥 □
500m	🟥 □	🟥 □	🟥 □
1000m	🟥 □	🟥 □	🟥 □

Send cycle time [ms]

2 4 8 12 20

Figure 358: Flexi Line view, General settings

On the left-hand side of this view, the following can be entered for the process image: a name, a description, and a revision number (consisting of a major version number plus a minor version number). The checksum for the process image is also shown here (see "Flexi Line checksum", page 485).

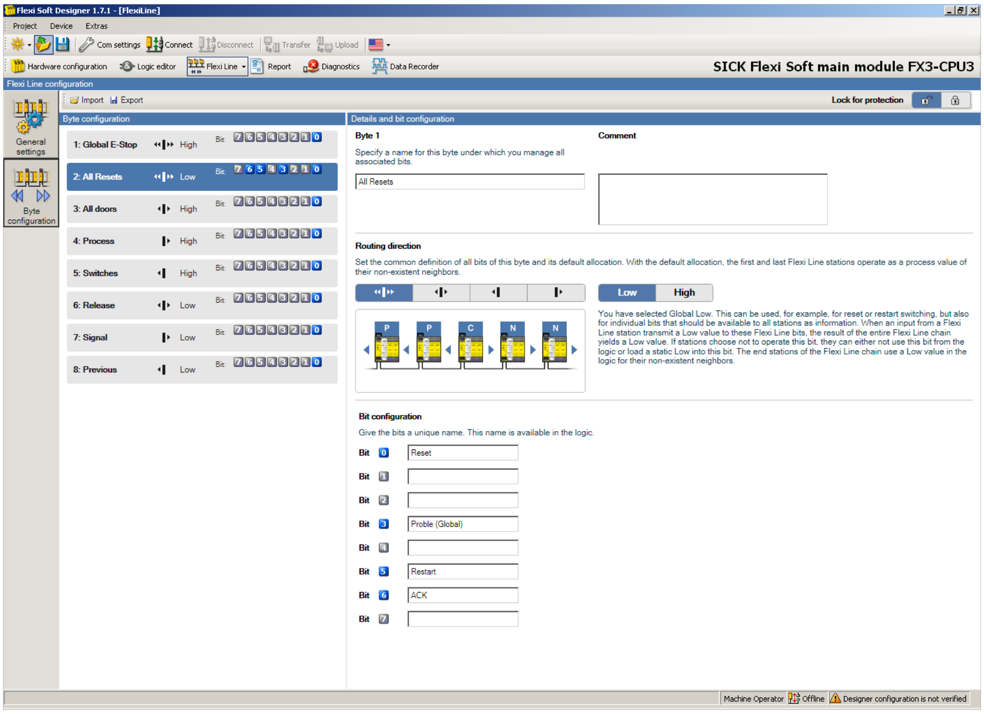
The size of the process image, and the send cycle time can be specified on the right-hand side. The values that are possible depend on the maximum length of the segment cable for the Flexi Line system as a whole (see table 228, page 481).

If a table cell is shown in red, it means that the combination you have selected for the maximum segment cable length and data size is not compatible with the set send cycle time. To be able to select this combination, the send cycle time must first be set to a higher value.

NOTE

The send cycle time is identical for all stations and so is not synchronized with the logic cycle, which can vary from one station to another.

Byte configuration



Byte configuration

Byte	Default value
1: Global E-Stop	High
2: All Resets	Low
3: All doors	High
4: Process	High
5: Switches	High
6: Release	Low
7: Signal	Low
8: Previous	Low

Details and bit configuration

Byte 1

Specify a name for this byte under which you manage all associated bits.

All Resets

Comment

Routing direction

Set the common definition of all bits of this byte and its default allocation. With the default allocation, the first and last Flexi Line stations operate as a process value of their non-existent neighbors.

You have selected Global Low. This can be used, for example, for reset or restart switching, but also for individual bits that should be available to all stations as information. When an input from a Flexi Line station transmit a Low value to these Flexi Line bits, the result of the entire Flexi Line chain yields a Low value. If stations choose not to operate this bit, they can either not use this bit from the logic or load a static Low into this bit. The end stations of the Flexi Line chain use a Low value in the logic for their non-existent neighbors.

Bit configuration

Give the bits a unique name. This name is available in the logic.

Bit 0	Reset
Bit 1	
Bit 2	
Bit 3	Proble (Global)
Bit 4	
Bit 5	Restart
Bit 6	ACK
Bit 7	

Machine Operator | Offline | Designer configuration is not verified

Figure 359: Flexi Line view, Byte configuration

The left-hand side of this view shows an overview of the bytes within the Flexi Line process image. After selecting a byte, the following settings for the byte concerned can be edited on the right-hand side:

- Name of the byte
- Comment
- Routing direction: The data of a byte can either be shared globally across the entire system or merely shared locally with one or both of the neighboring stations.
- Default value 1 or 0
- Tag names of the individual bits
- Activation/deactivation of the individual bits

Routing

A byte can either be valid locally (i.e., within one or both of the directly neighboring stations) or globally across the entire Flexi Line system.

A globally valid byte is communicated throughout the entire Flexi Line system. All stations are able to read and change every bit of this byte. If a station changes a bit, this change becomes effective on all other stations as well.

A locally valid byte is either shared with just one of the directly neighboring stations or with both of them. A station that receives a local byte from a neighboring station evaluates the information of this byte and, in turn, creates its own local byte, which it sends to one or both of its neighboring stations. The data that is received and the data that is sent are independent of one another.

Default value

The default value always applies to all the bits of a byte jointly. It defines how a bit is affected by a station:

- A bit with a default value of **High** has a value of 1 if all the stations signal a value of 1 for the bit concerned (logical AND operation). As soon as even one of the stations sets this bit to 0, the bit is set to 0. In conjunction with a globally valid bit, this setting can be used, for example, for emergency switching off circuits.
- A bit with a default value of **Low** has a value of 0 if all the stations signal a value of 0 for the bit concerned (logical OR operation). As soon as even one of the stations sets this bit to 1, the bit is set to 1. In conjunction with a globally valid bit, this setting can be used, for example, for a reset or restart function.

Activating and deactivating individual bits

Any bits that are not required can be deactivated by not assigning a tag name to them. Once deactivated, bits are no longer available/visible in the logic editor or in the Diagnostics view. The size of the process image is not affected by this.

15.2.3 Flexi Line checksum

The Flexi Line checksum is required so that the stations in a Flexi Line system can communicate with one another. All the stations in a Flexi Line system must have an identical Flexi Line checksum. This ensures that stations can only communicate with each other if they belong to the same Flexi Line system. If an inconsistent Flexi Line checksum is detected within a Flexi Line system, all connected stations switch to **Error on Flexi Line bus** mode (Line LED flashes ☀ red/green at 2 Hz).

The Flexi Line checksum is calculated from the following settings:

- Size of process image and maximum segment cable length
- Send cycle time
- Range of each byte
- Default value of each byte
- First part of revision number

The sub revision number and the customized names assigned to the bits, bytes, and process image do not affect the Flexi Line checksum.

NOTE

- If a change is made to the process image of any station and this alters the Flexi Line checksum, this new image must be transmitted to all the other stations as well. This sets the Flexi Line checksum back to the same value on all the stations.
- Otherwise, inconsistent Flexi Line checksums will be present within the Flexi Line system and it will not be possible for safety communication to be established between the stations.
- The Flexi Line checksum forms part of the configuration that is stored in the system plug of each connected FX3-CPU3 main module.

15.2.4 Flexi Line data in the logic editor

Each Flexi Line station automatically creates a local instance of the process image using the data received from its neighboring stations. If the local information of a station affects global bits, these values are immediately taken into account in the local instance of the process image as well.

The output process image is created with the help of routing function blocks. For this purpose, the signals of the local inputs must each be routed to a Flexi Line output.

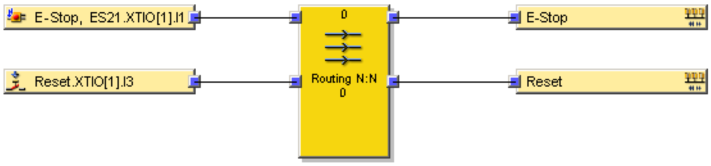


Figure 360: Routing of local signals to the Flexi Line process image

After this, the values of these local inputs are made available as Flexi Line inputs across the entire Flexi Line system via the Flexi Line process image.

From a logic programming perspective, the Flexi Line inputs are no different from any other type of safety capable input.

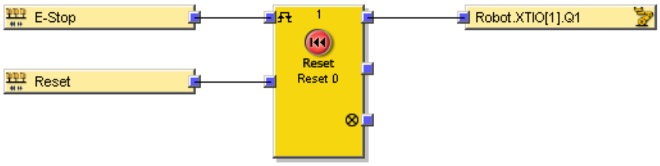
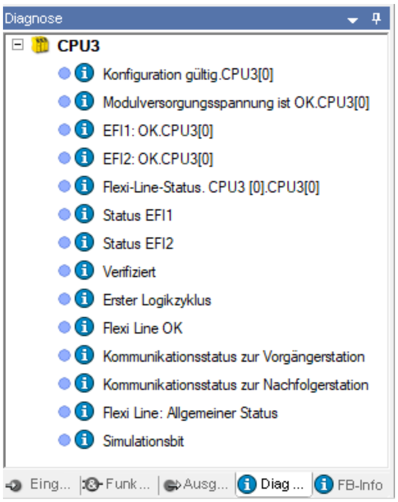


Figure 361: Using signals from the Flexi Line process image in the logic

15.2.5 Flexi Line status bits

Each main module in a Flexi Line system signals the current status using status bits. These status bits are available in the **Diagnostics** window of the Logic editor as inputs of the main module concerned.


Diagnose

CPU3
- Konfiguration gültig.CPU3[0]
- Modulversorgungsspannung ist OK.CPU3[0]
- EFI1: OK.CPU3[0]
- EFI2: OK.CPU3[0]
- Flexi-Line-Status. CPU3 [0].CPU3[0]
- Status EFI1
- Status EFI2
- Verifiziert
- Erster Logikzyklus
- Flexi Line OK
- Kommunikationsstatus zur Vorgängerstation
- Kommunikationsstatus zur Nachfolgerstation
- Flexi Line: Allgemeiner Status
- Simulationsbit

Figure 362: Status information for the Flexi Line system in the Logic editor

Table 229: Meaning of Flexi Line status bits

Status bit of main module	Meaning
Flexi Line system status	This diagnostic bit is 1 when the entire Flexi Line system is in operation and all Flexi Line participants are successfully communicating with one another.
Flexi Line OK	This diagnostic bit is 1 when the Flexi Line system is in operation and the main module is successfully communicating with the previous and next stations, or when no such stations exist. The Flexi Line status of the main module is **Teached**.

Status bit of main module	Meaning
Flexi Line teach required	This diagnostic bit is 1 when the Flexi Line status of the main module is **Not teached** and teaching of the Flexi Line station is required.
Communication status with the previous station	This diagnostic bit is 1 when the Flexi Soft station is successfully communicating with the previous station. If there is no previous station, this bit is always 1.
Communication status with the next station	This diagnostic bit is 1 when the Flexi Soft station is successfully communicating with the next station. If there is no next station, this bit is always 1.
Diagnostic bit for Flexi Line bit configuration	Each bit configured in the Flexi Line bit configuration has its own associated Flexi Line input process data and a Flexi Line output status message.

These status bits can be used to set up your own diagnostics system, e.g., by connecting them to a **Message generator** function block or by switching on a warning light when the Teach function is required or active.

Description of further diagnostic bits of the main module: see "Module status bits of the main module", page 92.

15.2.6 Teach

To activate the topology of the Flexi Line system, it must be confirmed by means of a Teach procedure. This can be carried out using Flexi Soft Designer. To allow for subsequent changes to the system topology, there is also an integrated Teach function available within the logic.

NOTE
Whenever a change is made to the topology of a Flexi Line system, Flexi Line communication stops immediately. Communication is only reinitialized and restarted once the Teach function has been performed.

If a station is suspended using the Teach function, the safety outputs on the stations that remain active may be set to High under certain circumstances.

WARNING
Limited safety due to Teach function

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

- ► Analyze the application and check whether additional safety measures are required if the Teach function is activated.
- ► The Teach function is to be regarded as a configuration procedure. Consequently, the Teach function must be secured in accordance with the relevant safety requirements, e.g., by using a key switch that has the Teach input connected to it within the logic and by using a Restart function block to monitor the timing conditions.
- ► Determine how the deactivated parts of the machine or plant/system have to be dealt with. Indicate that the control elements and sensors will not have any effect on those parts of the machine that were previously connected to them (for example, attach “Out of service” signs to the emergency stop pushbuttons).
- ► Only authorized and specially trained personnel are permitted to activate the Teach function.
- ► Before using the Teach function, make sure that there is no one inside the hazardous area and that no one can access this area while the Teach function is active.
- ► After using the Teach function, check the safety function of the entire system.

Performing a Teach process with Flexi Soft Designer

In the **Flexi Line** view of Flexi Soft Designer, there is a **Teach** button on the toolbar.

► During commissioning, click the **Teach** button once all stations are switched on and set to the **Teaching required** status. The topology of the Flexi Line system is checked and confirmed, and the system starts.

Performing a Teach process with a pushbutton

NOTE
If a short-circuit to High (to 24 V DC) occurs at a physical input, the evaluated signal may produce a pulse if the signal for **Teach** is reset as a result of short-circuit detection.

WARNING
Malfunction of the signals for teaching in the event of a short-circuit to high

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

► Make sure that the transitions of the signals for **Teach** meet the requirements of safety standards and regulations.
  ► Make sure the signal line is laid with protection (to prevent a cross-circuit with other signal lines).
  ► No short-circuit detection, i.e., no referencing to test outputs.

If it is necessary to remove, add, or replace individual stations during actual operation, the Teach function can also be implemented as a pushbutton.

Configuring a pushbutton for the Teach function

► Connect a Teach pushbutton to the inputs of each station within the Flexi Line system for which the Teach function needs to be available. The Teach pushbutton can, for example, take the form of a dual-channel key-operated switch.

► Within the logic for this station, connect the input of the Teach pushbutton to the Teach output of the relevant station via a **Restart** function block.

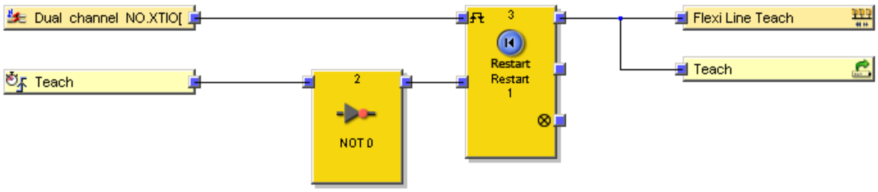


Figure 363: Configuration of the Teach function in the logic editor

When the Teach pushbutton is pressed, the Teach output switches to 1 for the duration of one logic cycle. The rising signal edge (0-1) at the Teach output triggers the Teach function.

15.2.7 Status and diagnostics

The **Diagnostics** view shows which data is received, used, and forwarded.

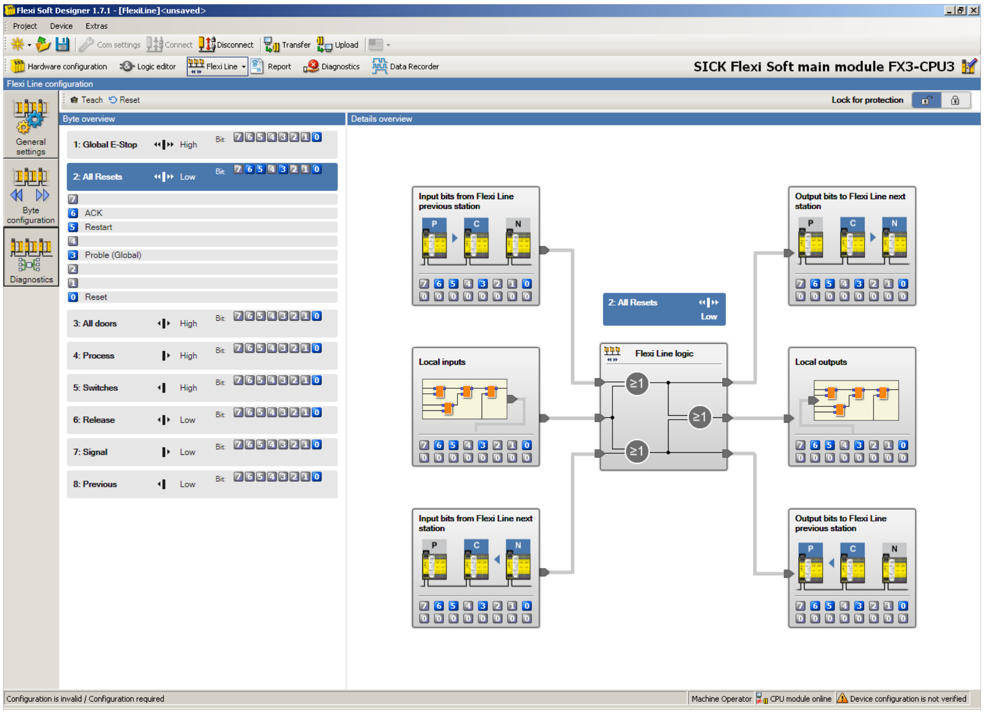


Figure 364: Flexi Line diagnostics

The **Byte overview** area on the left shows the bytes of the process image for the current station. After selecting one of the bytes, the associated bits along with their tag names appear underneath.

The **Details overview** area on the right shows how the selected byte is processed:

- The input bits that are received from one or both of the neighboring stations, as well as the status of the local inputs, are shown on the left-hand side.
- The output bits that are sent to one or both of the neighboring stations, as well as the status of the local outputs, are shown on the right-hand side.

When the Flexi Line system is online, the active bits are displayed in color and the deactivated bits in gray.

15.3 Setting up a new Flexi Line system

15.3.1 Configuring and commissioning a Flexi Line system

This section describes how to set up and commission a new Flexi Line system.

A Flexi Line system is configured in two steps.

- The first step is to configure the first station and define the process image.
- The second step is to configure the other stations. This involves transferring the process image to them.

NOTE

Each station within a Flexi Line system must be configured and commissioned as a single station in Flexi Soft Designer.

Configuring the first station

- ► Open the Flexi Soft Designer.
- ► In the start dialog, click **Create new project** or go to the **Project** menu and select **New** followed by **Single-station project.** The **Hardware configuration** window opens.

- Add an FX3-CPU3 main module.
- Now add the desired hardware, see "Configuring the Flexi Soft modules", page 42 and see "Connecting elements", page 44.
- Once the hardware configuration for the selected station is complete, go to the element selection list and drag the **Flexi Line** element over to the main module. The **Flexi Line configuration** dialog box opens.
- In the **Flexi Line configuration** dialog box, click **New Flexi Line definition**. The **Flexi Line** view opens.
- Configure the Flexi Line process image, see "Flexi Line configuration", page 482.

NOTE
Plan the Flexi Line process image carefully. If the process image is changed at a later time, it needs to be transferred again to each individual station within the Flexi Line system.

- Go to the Flexi Line toolbar and click the **Export Flexi Line definition** button to export the Flexi Line definition.
- Configure the logic for the station, see "Logic editor", page 61 and see "Configuring the Flexi Line logic", page 491.

Setting up the other stations

- Configure the hardware of the remaining Flexi Line stations in exactly the same way as for the first one.
- Once the hardware configuration for a station is complete, go to the element selection list and drag the **Flexi Line** element over to the main module. The **Flexi Line configuration** dialog box opens.
- In the **Flexi Line configuration** window, go to the **Found Flexi Line definitions** area and click the name of the file that contains the previously saved Flexi Line definition to import it.

Or:

- Click **Use existing Flexi Line definition.** A file selection window opens. Select the required file and click **Open**.
- Now configure the logic for the station.

Commissioning the Flexi Line system

- Connect the individual Flexi Line stations as described in the corresponding section of the “Flexi Soft Modular Safety Controller Hardware“ operating instructions.
- Commission each individual station just like a single system. The stations switch to the **Teaching required** status and the LINE LED starts flashing ✹ green at 2 Hz.
- Once the status of all the stations is set to **Teaching required**, switch to the **Flexi Line** view with Flexi Soft Designer connected to any of the stations.
- Click the **Teach** button on the toolbar to commission the Flexi Line system. The topology of the system undergoes checking and confirmation, and the Flexi Line system starts.

15.3.2 Modifying a Flexi Line system

New stations can be added to an existing Flexi Line system if these stations have a Flexi Line definition that matches the existing system. They can even be added while the existing system is in operation. As soon as the stations within the system detect the new addition, they switch to the **Teaching required** status and the LINE LED starts flashing ✹ green at 2 Hz.

If one or more stations are removed from an operable Flexi Line system while it is switched off, this system switches to the **Teaching required** status as soon as it is switched back on and the LINE LED starts flashing ✹ green at 1 Hz or 2 Hz.

If one or more stations are removed from a Flexi Line system while the system is in operation, the neighboring stations signal a Flexi Line error status, i.e., the LINE LED starts flashing ☀ red at 1 Hz. In this case, the error status can be reset by means of a Teach operation.

If a station is no longer required and is bypassed (muted) while the system is in operation, a Flexi Line error occurs (LINE LED starts flashing ☀ red at 1 Hz). In this case, the system cannot be reset by carrying out a Teach operation. Instead, it must be switched off and then back on again. When it is switched back on, the system switches to the **Teaching required** status and the LINE LED starts flashing ☀ green at 2 Hz.

15.3.3 Configuring the Flexi Line logic

The logic of a Flexi Line station is programmed in two steps.

- Integrate the local data of the station into the Flexi Line process image: All the local information that is of relevance to the Flexi Line process image must be integrated into the image using a routing function block.
- Create the local logic with the help of the process image data.

Example of straightforward Flexi Line logic

The following example shows a station with an emergency stop pushbutton and a reset. This station is responsible for switching a machine via a single-channel safety output. The following figure shows the hardware configuration:

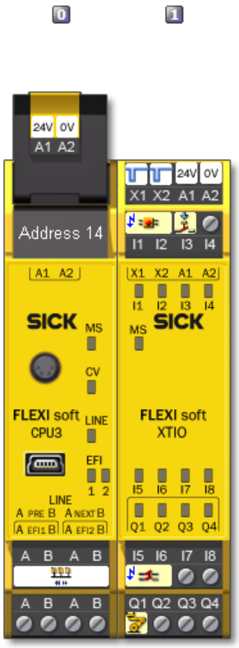


Figure 365: Example for the hardware configuration of a Flexi Line system

The station is connected via Flexi Line to other stations with an identical or similar configuration, where the statuses of both pushbuttons are likewise to be made available. Two bits within the process image are used for this purpose:

- **Bit 1**: global bit, default value: 1, name: E-Stop
  This bit brings together all the emergency stop commands for all stations: If an emergency stop pushbutton is pressed on any station, this bit is set to 0 (logical AND).
- **Bit 9**: global bit, default value: 0, name: Reset
  This bit brings together all the reset commands for all stations: If a reset is pressed on any station, this bit is set to 1 (logical OR).

The signals of both connected pushbuttons are now routed to the Flexi Line process image:

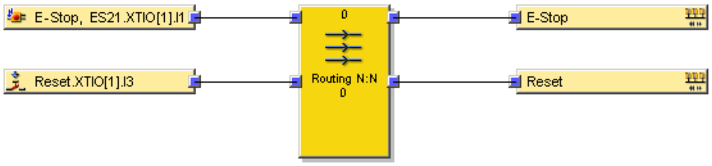


Figure 366: Routing of local signals to the Flexi Line process image

These signals can then be used as follows within the logic of each station included in this Flexi Line system:

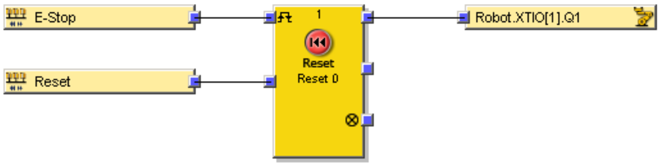


Figure 367: Using signals from the Flexi Line process image in the logic

16 Transferring the configuration

Overview

At first, the safety controller configuration only exists as a project, namely as a Flexi Soft configuration file. The configuration must be transferred to the system plug via the main module.

Important information

NOTE
The system plug and the main modules communicate via a internal interface. Directly connecting a computer to the system plug is not possible. Data can only be transferred to/read from the system plug using a compatible main module.

Transferring the configuration

During transfer to the system plug, the configuration data is checked for compatibility and can be subsequently verified. In addition, optional write protection can be applied to the data.

With Flexi Soft Designer, the configuration data can be transferred to any number of Flexi Soft system plugs.

The configuration data, including the verification and, if applicable, write protection information that was set when the first safety controller was configured, is copied exactly.

16.1 Transferring project data to the safety controller

Once it has been transferred, the configuration data is read back from the system plug if verification has been activated in Flexi Soft Designer (see "Verifying configuration", page 494).

NOTE
It takes a little while for the configuration data to be read back from the system plug; the system plug must not be disconnected while this is happening. Flexi Soft Designer displays a corresponding warning while the operation is in progress.

16.2 Compatibility check

For each module that is to be configured, the configuration data contains an electronic type code and a version code. During transfer, each module checks whether it is compatible with the configuration data. The compatibility check relates purely to the functional part of the module concerned, not to the type of hardware (the terminal design is not taken into account, for example).

If the compatibility check produces a negative result, a corresponding error message is generated within the module concerned and in the main module.

NOTE
In Flexi Soft Designer, some modules are stored with different version numbers and this means that a compatible module can be selected from a list below the relevant module.

16.3 Verifying configuration

Overview

Once the configuration has been successfully transferred to the controller, the Flexi Soft system can be verified. This involves reloading the transferred configuration from the Flexi Soft system and comparing it with the project data. If the data matches, it is displayed in a report. If the user confirms that the data is correct, the system is considered to be verified.

If the configuration is verified, the Flexi Soft system automatically switches to the Run status when the voltage supply is switched on. If the configuration is not verified, the system remains set to the Stop status after switch-on (CV LED on main module flashes) and must be started using Flexi Soft Designer.

Important information

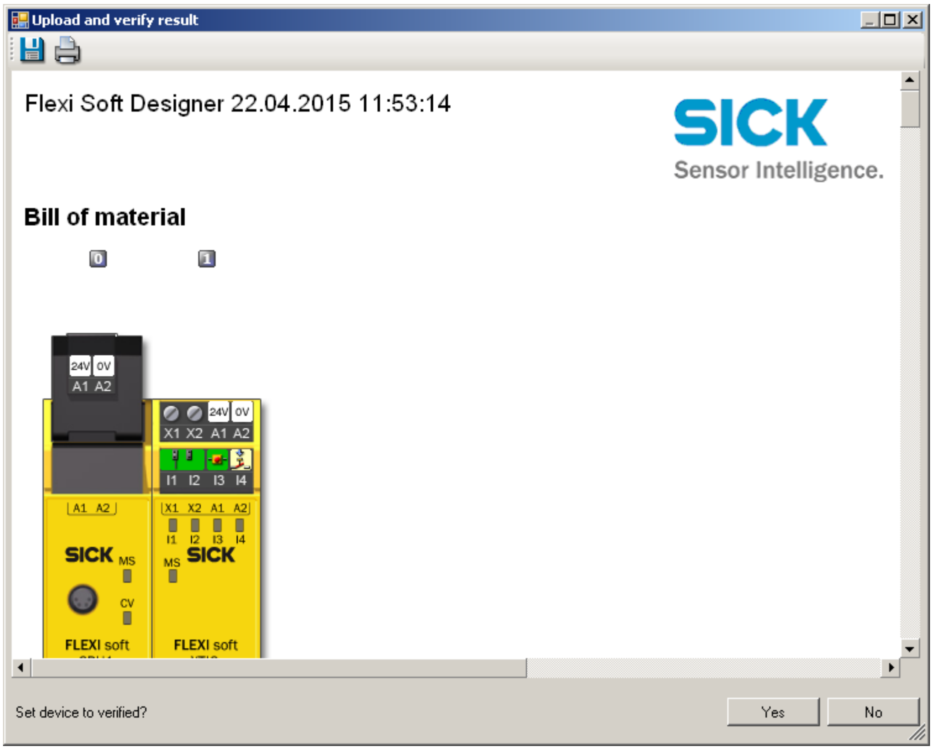
NOTE
The configuration of connected elements, e.g., EFI sensors, is not affected by verification of the Flexi Soft system. This configuration must be verified separately using the serial interface of the respective device.

Prerequisites

- To verify the configuration, you must log in as an authorized client.

Procedure

1. In the **Hardware configuration** view, click the **Receive and compare configuration** button. A report containing the current configuration is created and displayed in the **Load and compare** window. This report can be saved or printed out.


Upload and verify result

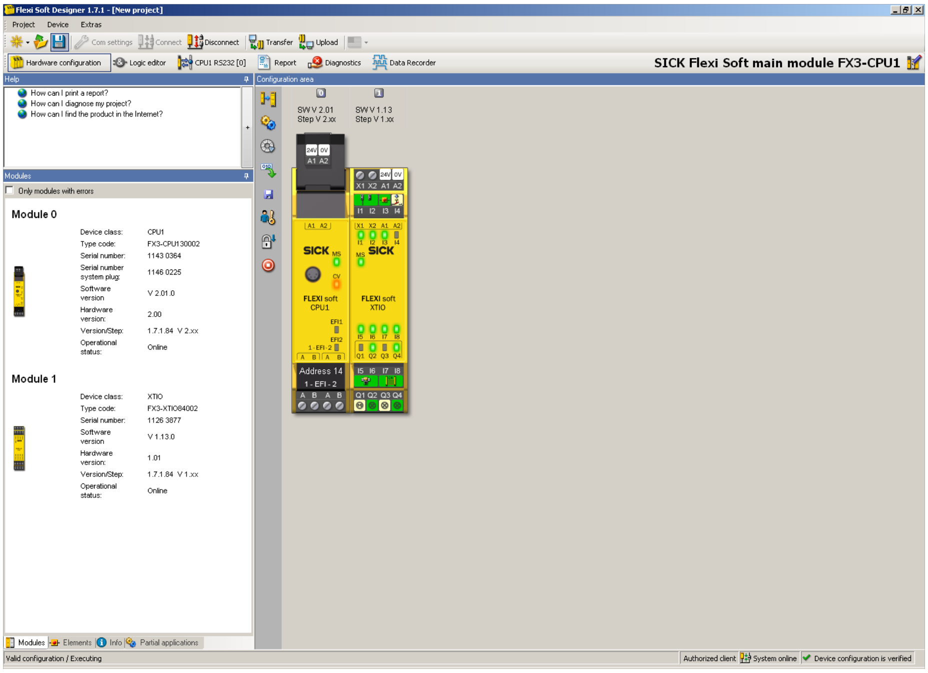
Flexi Soft Designer 22.04.2015 11:53:14

Bill of material

Set device to verified?

Figure 368: Setting the device to verified

2. If the configuration shown is exactly what you were expecting it to be, close the window by clicking **Yes**. The system is then classed as verified.



Module 0

Device class: CPU1
Type code: FX3-CPU130002
Serial number: 1143 0364
Serial number system plug: 1146 0225
Software version V 2.01.0
Hardware version: 2.00
Version/Step: 1.7.1.84 V 2.xx
Operational status: Online

Module 1

Device class: XTIO
Type code: FX3-XTIO84002
Serial number: 1126 3877
Software version V 1.13.0
Hardware version: 1.01
Version/Step: 1.7.1.84 V 1.xx
Operational status: Online

Figure 369: Verification successful

Complementary information

- The CPU logic pages of the configuration are verified and listed in the report when the Flexi Soft system is verified.
- The verification status is indicated in the Flexi Soft Designer status bar on the bottom right-hand side. It is also indicated by the CV LED on the Flexi Soft main module, which lights up.
- The verification flag is copied when the data is read back into the system plug and is automatically transferred to every safety controller where this configuration is to be duplicated.
- If the configuration that is read back differs from the one in the Flexi Soft Designer project, a corresponding error message is displayed. In this case, it is not possible to verify the configuration. The error message contains notes for further action.
- If you change a configuration after it has been verified, the status is reset and must be re-verified.

16.4 Activating write protection for the configuration in the controller

A verified configuration can be protected against accidental changes by applying write protection. Write protection can be set/canceled by using the lock symbol that is located to the left of the main module in the Hardware configuration view of Flexi Soft Designer.

Write protection is automatically copied across when the data is transferred to the system plug and is automatically applied to every safety controller where this configuration data is to be duplicated.

16.5 Configuration checksums

Overview

In the Flexi Soft Designer report, various checksums are displayed on the hardware configuration information page and on the configuration pages for Flexi Line and ACR.

Important information

NOTE

Each checksum exists both within the project and in the system plug of the respective Flexi Soft station. The checksums in the system plugs correspond to those checksums in the project that applied when the configuration was last transferred from the project to the system plug of the station concerned.

Each checksum is 4 bytes long.

Configuration checksums

- Flexi Soft checksum:
  This checksum covers the configuration for the relevant Flexi Soft system, i.e., the configuration of all Flexi Soft modules including the logic program. The device firmware and hardware version used has no effect on the checksum.
  The configuration for any EFI-enabled devices connected via EFI is not included in the Flexi Soft checksum.
  Within a Flexi Line system, the Flexi Soft checksum does not refer to the entire Flexi Line system, but only to the single station concerned.
- Flexi Soft checksum (verified):
  The Flexi Soft checksum (verified) is the Flexi Soft checksum that applied at the time of the most recent verification. If the Flexi Soft checksum and the Flexi Soft checksum (verified) are identical, the configuration of the Flexi Soft system is classed as verified.
- ACR checksum:
  If ACR is activated, this checksum covers the ACR configuration for the EFI-enabled devices that are connected via EFI.
- Flexi Line checksum:
  The Flexi Line checksum is generated from the configuration for the Flexi Line process image. All the stations in a Flexi Line system must have an identical Flexi Line process image and, in turn, an identical Flexi Line checksum. This ensures that stations can only communicate with each other if they belong to the same Flexi Line system.
- Total checksum:
  If ACR is deactivated: The same value as the Flexi Soft checksum
  If ACR is activated: Checksum from Flexi Soft checksum and ACR checksum
- Checksum of grouped function blocks:
  This checksum covers the logic program within a grouped function block in the CPU logic. The relative execution sequence of the function blocks is taken into account.
- Checksum of customized function blocks:
  This checksum covers the logic program within a customized function block in the CPU logic. The relative execution sequence of the function blocks is taken into account.
- Checksum of the logic page:
  This checksum covers the logic program within a logic page of the CPU logic. The relative execution sequence of the function blocks is taken into account.

16.6 Duplicating system plugs

With Flexi Soft Designer Version ≥ 1.7.1, users in the **Maintenance personnel** user group can transmit a verified configuration containing a new password for the **Authorized customers** user group. This feature can be used to transfer the same configuration to multiple system plugs.

Proceed as follows:

1. Create a configuration using Flexi Soft Designer version ≥ 1.7.1.
2. Transfer the configuration.
3. Change the password for **Authorized customers** and/or **Maintenance personnel** in the main module.
4. Verify the configuration.
5. Save the verified project.

✓ The verified project can now be transferred to other CPUs. During subsequent verification of the project, the user can select which password should be transferred.

ℹ **NOTE**
During verification, the passwords are read out of the main module and saved in the Flexi Soft Designer project. For this reason, the configuration must not be verified before the password is changed.

16.7 Automatic configuration recovery (ACR)

When EFI-enabled devices are replaced, the ACR function sets the configuration of the newly connected EFI-enabled devices to the same configuration as before. The ACR function is available on all main modules as of FX3-CPU2.

Table 230: System requirements for automatic configuration recovery (ACR)

System component	Version
Hardware	FX3-CPU2 or higher with firmware version ≥ V2.50.0
Software	Flexi Soft Designer version ≥ V1.5.0

ℹ **NOTE**

- The ACR function must be set separately for each EFI connection (EFI1 or EFI2). It is independent of the configuration for the logic and for the station setup.
- ACR always affects all of the EFI-enabled devices that are operated on the EFI connection concerned. It is not possible to control a single device on its own.
- ACR can only be used with replacement devices of the same type.

16.7.1 Supported EFI-enabled devices

The ACR function can be used to save and recover the configurations of the following device families:

- S3000 with firmware version ≥ V2.41, **not** in Compatibility mode. For details, please refer to the “Compatibility mode” chapter of the operating instructions for the “S3000 Safety Laser Scanner” (SICK part number 8009791).
- S300 with firmware version ≥ V2.10, **not** in Compatibility mode. For details, please refer to the “Compatibility mode” chapter of the operating instructions for the “S300 Safety Laser Scanner” (SICK part number 8010946).
- S300 Mini
- M4000
- C4000

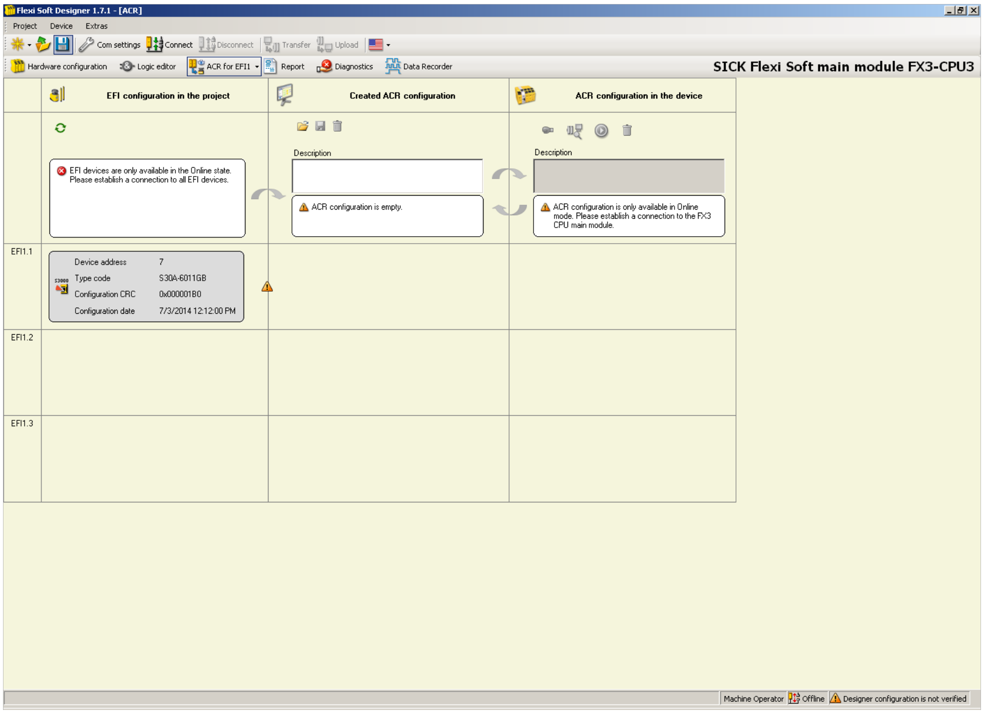
The ACR function can **not** be used to save and recover the configurations of the following device families:

- UE family (UE402/UE403, UE44xx, UE41xx, UExx40)
- Devices from the Flexi Soft range (FX3-CPUx)

Additional information on the ACR function can be found in the relevant operating instructions for the EFI-enabled devices that are used.

16.7.2 Structure of the ACR dialog box

► To open the ACR dialog box, click the **Interfaces** button and then select the desired EFI connection from the menu (**ACR for EFI1** or **ACR for EFI2**). The ACR dialog box opens.



EFI devices are only available in the Online state. Please establish a connection to all EFI devices.

ACR configuration is empty.

ACR configuration is only available in Online mode. Please establish a connection to the FX3 CPU main module.

Device address 7
Type code S30A-6011GB
Configuration CRC 0x000001B0
Configuration date 7/3/2014 12:12:00 PM

Designer configuration is not verified

Figure 370: ACR dialog box

The left-hand area of the dialog box contains the current configuration for the connected EFI-enabled devices. This configuration must have been verified. Otherwise, a corresponding message appears.

The central area contains the ACR configuration within the project. The verified EFI configuration can be transferred across to this area from the area on the left.

The area on the right contains the data that is present within the system plug of the Flexi Soft main module.

Table 231: Buttons in the ACR dialog box

Button	Meaning
📂	Import a saved ACR configuration
💾	Export the current ACR configuration
🔄	Refresh display
	Running (forcing) the ACR function

Button	Meaning
	Set ACR configuration to verified
▶	Activate ACR configuration
■	Deactivate ACR configuration
🗑	Delete data set from project or main module
↷	Transfer data for further processing
↶	Transfer verified ACR configuration back to the project

Table 232: Status indicators in the ACR dialog box

Icon	Meaning
⚠	Structural differences or configuration data discrepancies have been detected between the two neighboring elements.
✓	The neighboring data is identical.
ℹ	Information to be aware of

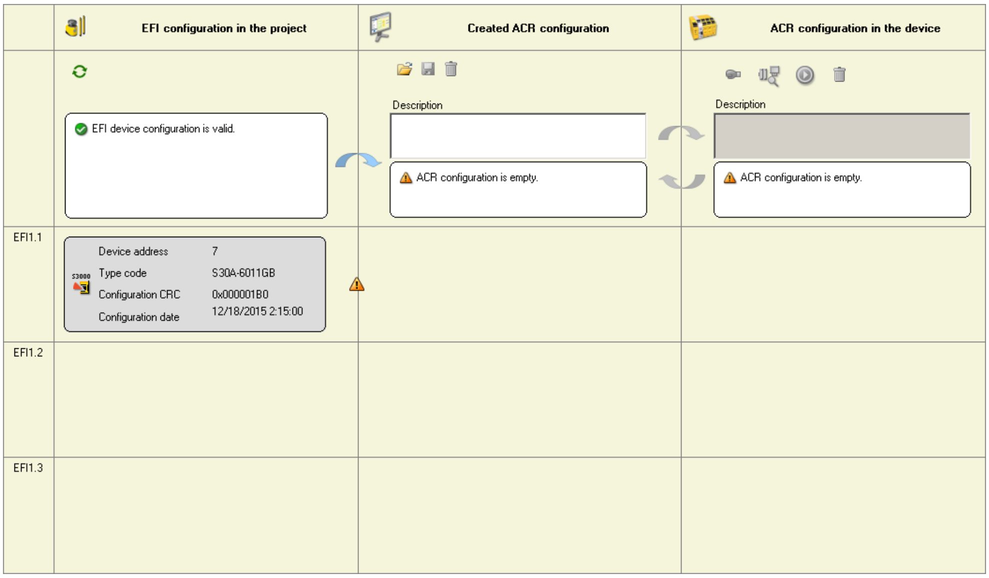
16.7.3 Setting up ACR

⚠ **WARNING**
Malfunction due to faulty configuration

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

- ► After setting up and activating ACR and after each device replacement, check the safety function of the devices.
- ► Before recommissioning the system, check the safety functions of the application (see "Thorough technical check and commissioning", page 519).

- ► Configure the connected EFI-enabled devices and verify the configuration. The following will then appear in the left-hand column of the ACR dialog box: the type code of the connected device, the configuration date, and the associated configuration checksum.


EFI configuration in the project
Created ACR configuration
ACR configuration in the device
Description
Description
EFI1.1
EFI1.2
EFI1.3
S3000

EFI device configuration is valid.

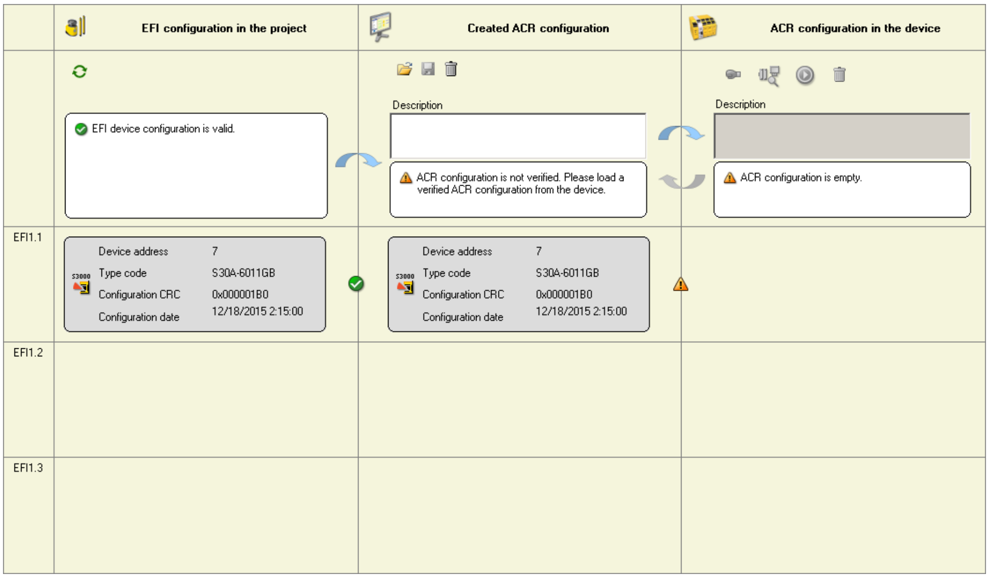
ACR configuration is empty.

ACR configuration is empty.

Device address	7
Type code	S30A-6011GB
Configuration CRC	0x000001B0
Configuration date	12/18/2015 2:15:00

Figure 371: Verified EFI configuration displayed in the ACR dialog box

► Now click the blue arrow between the left-hand and central areas to transfer this EFI configuration to the Flexi Soft project for use as an ACR configuration.


EFI configuration in the project
Created ACR configuration
ACR configuration in the device
Description
Description
EFI1.1
EFI1.2
EFI1.3
S3000
S3000

EFI device configuration is valid.

ACR configuration is not verified. Please load a verified ACR configuration from the device.

ACR configuration is empty.

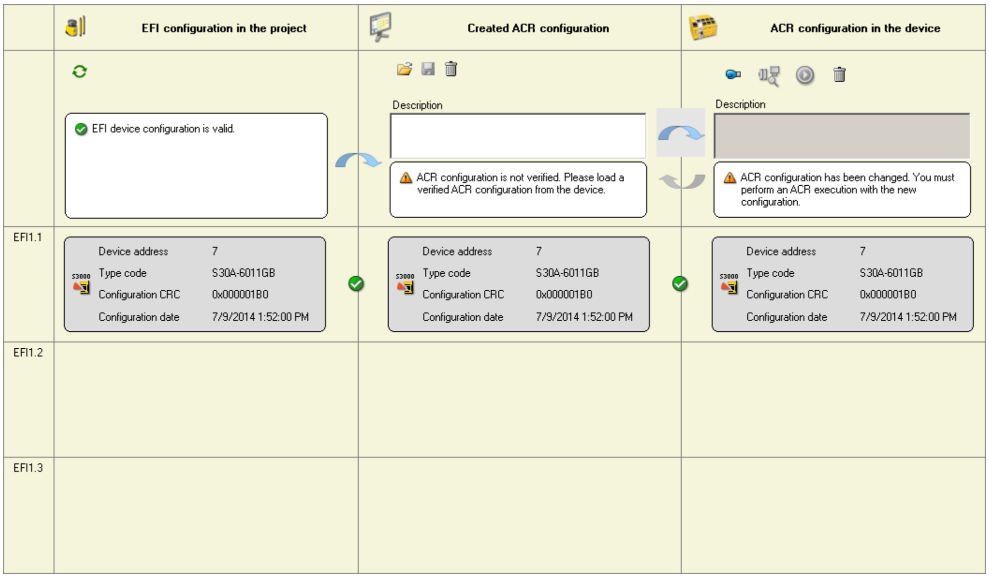
Device address	7
Type code	S30A-6011GB
Configuration CRC	0x000001B0
Configuration date	12/18/2015 2:15:00

Device address	7
Type code	S30A-6011GB
Configuration CRC	0x000001B0
Configuration date	12/18/2015 2:15:00

Figure 372: Transferring the EFI configuration for use as an ACR configuration

A description of the application can be entered in the **Description** field.

► Click the upper blue arrow between the central and right-hand areas to transfer the ACR configuration to the main module. A progress bar is displayed during this process.



EFI configuration in the project

EFI device configuration is valid.

Created ACR configuration

Description

ACR configuration is not verified. Please load a verified ACR configuration from the device.

ACR configuration in the device

Description

ACR configuration has been changed. You must perform an ACR execution with the new configuration.

EFI1.1

	EFI configuration in the project	Created ACR configuration	ACR configuration in the device
Device address	7	7	7
Type code	S30A-6011GB	S30A-6011GB	S30A-6011GB
Configuration CRC	0x000001B0	0x000001B0	0x000001B0
Configuration date	7/9/2014 1:52:00 PM	7/9/2014 1:52:00 PM	7/9/2014 1:52:00 PM

EFI1.2

EFI1.3

Figure 373: Transferring the ACR configuration to the main module

- ► Now click the **Run ACR** button to perform the function. A progress bar is displayed during this process.

NOTE

Once the **Run ACR** function has finished, the controller performs an automatic system reset.

- ► Click **Verify ACR configuration** to mark the ACR configuration in the device as verified.

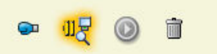


Figure 374: Verify ACR configuration

- ► Click **Activate ACR**. Another progress bar is displayed during this process. As soon as the activation process is complete, the ACR function starts monitoring the configuration of the EFI-enabled devices that are being operated on the relevant EFI connection (EFI1 or EFI2). If a change is detected, it recovers the configuration. The **Deactivate ACR** button can be used to deactivate the function again (e.g., so that you can change the configuration for the EFI-enabled devices that are connected).
- ► After a configuration has been changed, you must repeat the steps described above before you can use it:
  - o Transfer verified EFI configuration to the project for use as an ACR configuration
  - o Transfer ACR configuration to main module
  - o Run ACR function
  - o Verify ACR configuration
  - o Activate ACR configuration
- ► The ACR configuration can also be reactivated without performing the preceding steps. In this case, the ACR configuration that is stored in the main module will be transferred to the devices again. This will undo all of the changes that were made to the configuration for the EFI-enabled devices since ACR was deactivated.

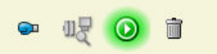


Figure 375: Activate ACR

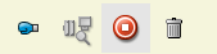
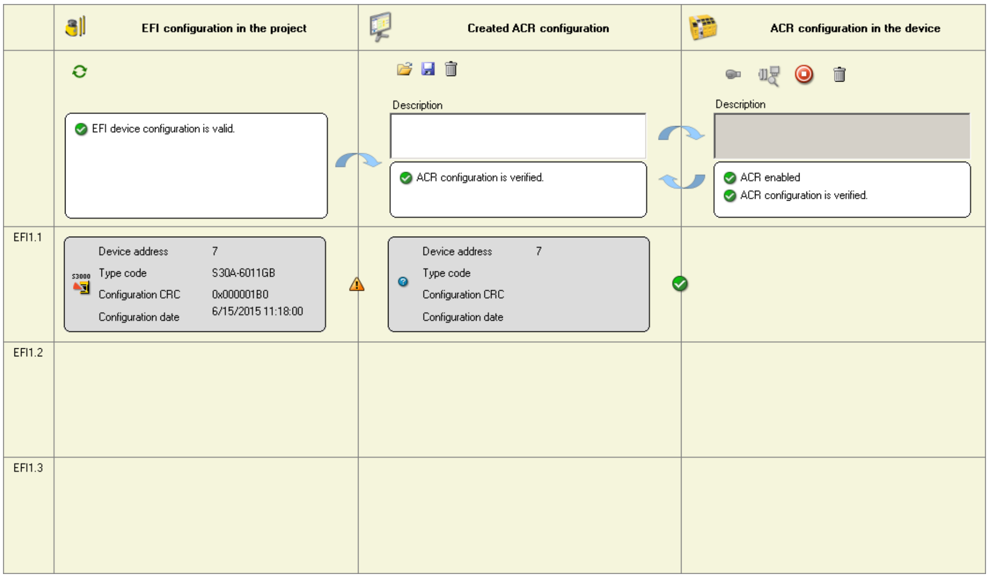


Figure 376: Deactivate ACR

- ► Transfer the verified ACR configuration back to the project. To do this, click the **Transfer verified data back to the project** button. This applies the verification status to the project. As of this point, the verification procedure never has to be repeated again when making further use of the project (e.g., in the context of series-produced machines).



EFI configuration in the project

EFI device configuration is valid.

Created ACR configuration

Description

ACR configuration is verified.

ACR configuration in the device

Description

ACR enabled
ACR configuration is verified.

EFI1.1

S3000

Device address	7
Type code	S30A-6011GB
Configuration CRC	0x000001B0
Configuration date	6/15/2015 11:18:00

Device address	7
Type code	
Configuration CRC	
Configuration date	

EFI1.2

EFI1.3

Figure 377: Verified ACR configuration within the project

This ACR configuration can now both be saved in the Flexi Soft project and exported again so that it can be used in other projects.

16.7.4 Replacing devices

WARNING
Malfunction due to faulty configuration

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

- ► After setting up and activating ACR and after each device replacement, check the safety function of the devices.
- ► Before recommissioning the system, check the safety functions of the application (see "Thorough technical check and commissioning", page 519).

When one or more of the connected EFI-enabled devices is replaced, the ACR function compares the newly connected devices with the stored data to make sure the type is identical. If the data matches, the configuration stored within the project is transferred to the newly connected devices. All the devices on the relevant EFI connection are configured jointly regardless of how many devices have actually been replaced.

This process starts approximately 15 seconds after an EFI-enabled device is started or restarted. Depending on the device and the size of the configuration, it can take anywhere between a few seconds and several minutes. To determine whether the ACR function is in progress or has been completed, look at the relevant EFI LED on the main module.

Table 233: Statuses indicated by the EFI LEDs when the ACR function is performed

EFI LED (EFI1 or EFI2)		Meaning
○		OK
●	Red	ACR function in progress
☀	Red (1 Hz)	Error while running ACR (ACR integration check failed, ACR transmission error)

16.7.5 Troubleshooting

Errors are indicated by the EFI LED for the relevant EFI connection. In addition, error and status messages are saved in the main module, e.g., information about whether the ACR function has been performed successfully, whether it has failed, or whether it has been deactivated.

Recognizing possible causes of error

► Compare the type code of the original device with that of the replacement device.

► Note the firmware version requirements that apply to the listed EFI-enabled devices: see "Supported EFI-enabled devices", page 497.

NOTE

- If the Flexi Soft main module detects a configuration that has been created using the ACR function, a warning is displayed. In this case, the ACR function must be deactivated on the relevant EFI connection before the configuration can be changed.
- The same error message appears when a device that was previously configured with ACR is to be used in a new environment or is to be reconfigured (Flexi Soft Designer ≥ V1.5.0 or CDS ≥ V3.6.8).

WARNING

Malfunction due to faulty configuration

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

► After setting up and activating ACR and after each device replacement, check the safety function of the devices.

► Before recommissioning the system, check the safety functions of the application (see "Thorough technical check and commissioning", page 519).

NOTE

Follow the additional instructions for using the ACR function that can be found in the operating instructions for the connected EFI-enabled devices.

16.8 Automated download of a configuration

Overview

With automated download, you can download verified configurations without user interaction, e.g., to configure several Flexi Soft stations in an identical manner. This supports applications such as series production.

Important information

WARNING
Malfunction due to faulty configuration

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

► After performing a configuration download and before commissioning, check the safety functions of the application (see "Thorough technical check and commissioning", page 519).

Prerequisites

- A license is required to use the automated configuration download function, see "Licensing and activating functions requiring a license", page 23.
- The configuration of all devices must be verified.
- The configuration does not contain any SICK safety systems requiring a license.

Principle of operation

You can generate download scripts using Flexi Soft Designer.

The scripts can then be run as follows using the Flexi Soft Download tool:

- In Windows with a batch file
- In Linux on a x64 architecture or an ARM architecture (ARM64, ARMHF (32 bit))

The following elements can be configured using the download scripts:

- Flexi Soft main modules FX3-CPUx
- EFI-enabled devices that support ACR, using ACR configuration

Complementary information

The download tool can be downloaded from the SICK website:

Table 234: Download tool

FSD download tool	SICK website
Windows 10	www.sick.com/676845
Linux ARM	www.sick.com/676846
Linux ARM64	www.sick.com/676847
Linux x64	www.sick.com/676848

16.8.1 Creating a download script

Important information

NOTICE
The exported configuration data may contain confidential information and may need to be protected from unauthorized access.

Procedure

1. Generate and activate a connection profile in Flexi Soft Designer.
   This connection profile is used during configuration using Flexi Soft Designer and the subsequently generated download script. All available connection profiles can be used (serial, TCP/IP, USB).
2. Connect Flexi Soft Designer to the Flexi Soft system.
3. Read in the configuration of all devices of the Flexi Soft system (main module and devices at the EFI connection, if applicable) or create a configuration in Flexi Soft Designer and transfer it to the Flexi Soft system.

4. Verify the configuration of the Flexi Soft system, if applicable.
5. In the **Extras** menu, select the **Generate configuration script ...** command. The **Generate configuration script** dialog box opens.
6. In the **Generate configuration script** dialog box, first enter the storage location and then the **Basic name for script files.** The files to be generated later are listed under **Generated files.**

NOTE
The base name must not contain any accented or Unicode characters.

NOTE
In the examples in this section, the **Basic name for script files**`Script` is used.

7. In the **Main module** tab, make the desired settings:
   - Enter a **Description** of the application.
   - Select the **Functions to be generated.** The selected functions are integrated into the script and executed when downloaded to a Flexi Soft system.
     - **Check device identify before transfer:** Checks whether the target device corresponds to the device used in the configuration before transferring the configuration.
     - **Check module identity before transfer:** Before transferring the configuration, checks whether the number and sequence of the physical modules for the download match the number and sequences of the modules in the configuration. For example, this option must be deactivated if only one main module is used to program several memory plugs.
     - **Verify device:** The configuration is marked as verified after transferring to the new device.
     - **Start device:** The new device automatically switches to the Run state after switching on.
     - **Limit to specified serial numbers:** When generating the script file, a dialog window opens. You can save a file with serial numbers in the dialog window. The serial numbers have to be from a main module or a system plug. The automatic download is only run for devices with these serial numbers.

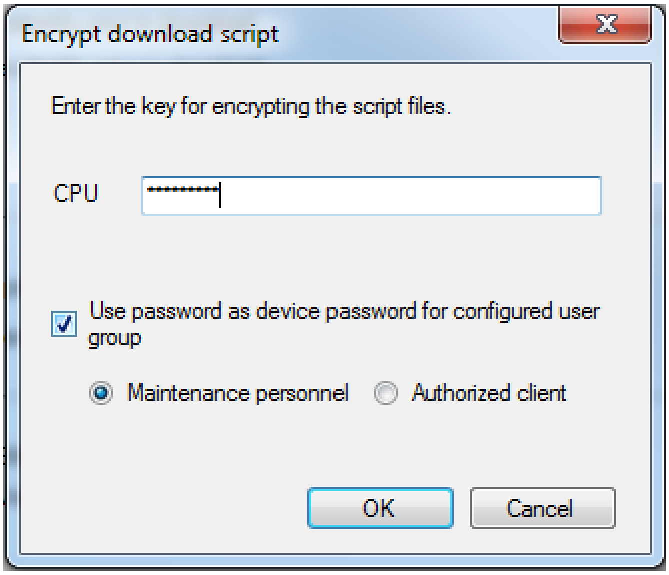
       Requirements for the file:
       - Serial numbers are in the SN list
       - Text file or CSV
   - Set **Passwords** for the **Maintenance engineer** and **Authorized customer** user groups. These passwords are transferred as device passwords to the respective Flexi Soft system during download.
8. Make the desired settings on the **EFI1** and **EFI2** tabs for any connected EFI-enabled devices:
   - If the devices are to be configured via ACR at the relevant EFI connection of the Flexi Soft main module, activate the **Use ACR for sensor configuration option instead of the script** option.
     The other options on this tab are deactivated in this case.

   NOTE
   After transferring a configuration generated with this option to a Flexi Soft system, ACR is **activated** for the EFI-enabled devices at the corresponding EFI connection.

- If the devices are to be configured at the relevant EFI connection with the CDS configuration software, deactivate the **Use ACR for sensor configuration instead of the script** option and select **Functions to be generated**.

NOTE
After transferring a configuration generated with this option to a Flexi Soft system, ACR is **deactivated** for the EFI-enabled devices at the corresponding EFI connection.

9. Click on the **Generate script files** button. The **Encrypt configuration script** dialog box appears.



Encrypt download script

Enter the key for encrypting the script files.

CPU **********

☑ Use password as device password for configured user group

● Maintenance personnel ○ Authorized client

10. Enter the desired key for the encryption of the script files and, if this key is also used in the current Flexi Soft system as the device password for the set user group (**Maintenance engineer** or **Authorized customer**), activate the corresponding option, if necessary.

NOTE
The key for encrypting the script files is only used to generate the script files. The device passwords for the Flexi Soft systems configured using the download tool must first be entered in the **Generate configuration script** dialog box.

11. Click on **OK**.
Depending on the configuration, the following files will be generated:
   - `Script.bat` is the batch file to perform the automated download.
   - `Script_CPU.dsc` contains the configuration data for the Flexi Soft main module.
   - `Script_EFI1.dsc` or `Script_ACR1.dsc` contains the configuration data for the devices at the **EFI1** connection of the Flexi Soft main module, if applicable.
   - `Script_EFI2.dsc` or `Script_ACR2.dsc` contains the configuration data for the devices at the **EFI2** connection of the Flexi Soft main module, if applicable.
   - `Script_COMM.dsc` contains the communication profile to be used for the download.
   - `Script.key` contains the encrypted password for the Maintenance engineer or Authorized customer user to login to the Flexi Soft main module.

16.8.2 Automated download in Windows

16.8.2.1 Running the download tool using the batch file

Procedure

1. Connect the Flexi Soft station to be configured to the computer in the same way as the station used to generate the configuration script.
2. Double-click the batch file.

✓ The generated configuration scripts are transferred one after the other to the main module and, if necessary, to the devices on the EFI connections of the main module.

✓ All download tool messages are written to a log file in .txt format. This is stored in the same folder and under the same name as the script file.

Complementary information

- If no other password is specified, the default password “SICKSAFE” is used for execution.
- If a different password is to be used, the batch file must be modified accordingly using a text editor.
- The batch file accesses the download tool via default path `C:\Program Files\SICK\FlexiSoft\FSDownloadTool\FSDownloadTool.exe` . The configuration files must be located in the same folder as the batch file. If different paths are to be used, the batch file must be modified accordingly.

16.8.2.2 Directly calling up download tool in Windows

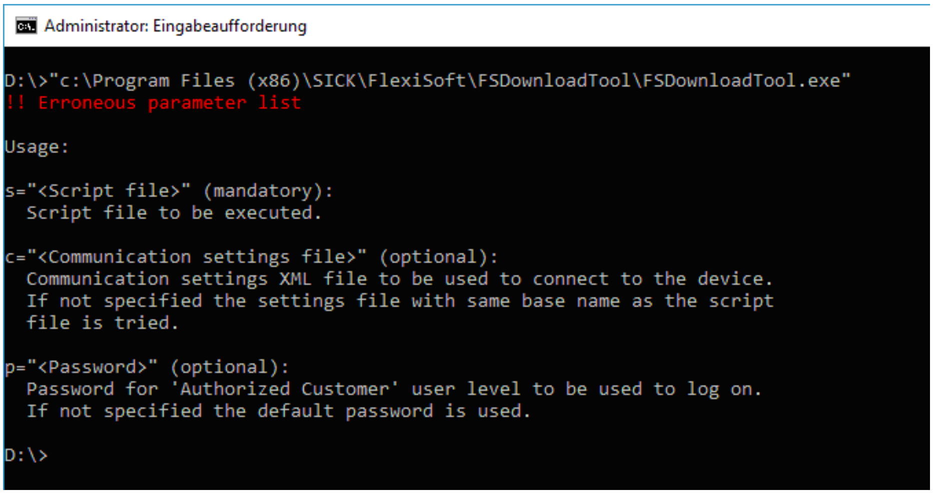
Procedure

1. Open the computer’s console window (e.g., the Windows command prompt).
2. Access the download tool `FSDownloadTool.exe` using the desired parameters, including the required file paths.

Table 235: Parameters of the download tool

Parameter	Meaning	Description
s	Script	The script to be executed with the configuration to be transferred, e.g., `Script_CPU.dsc`. This parameter must be specified.
c	Communication profile	The communication profile to be used, e.g., `Script_COMM.dsc`. Specifying this parameter is optional. If no communication profile is specified, the download tool may search and use a communication profile with the same path and basic name as the specified script.
p	Password	The password for the **Authorized customer** user group. Specifying this parameter is optional. If no password or an incorrect password is specified, the download tool uses the default password “SICKSAFE”.

✓ If no or incorrect parameters are specified, an explanation appears in the console window.

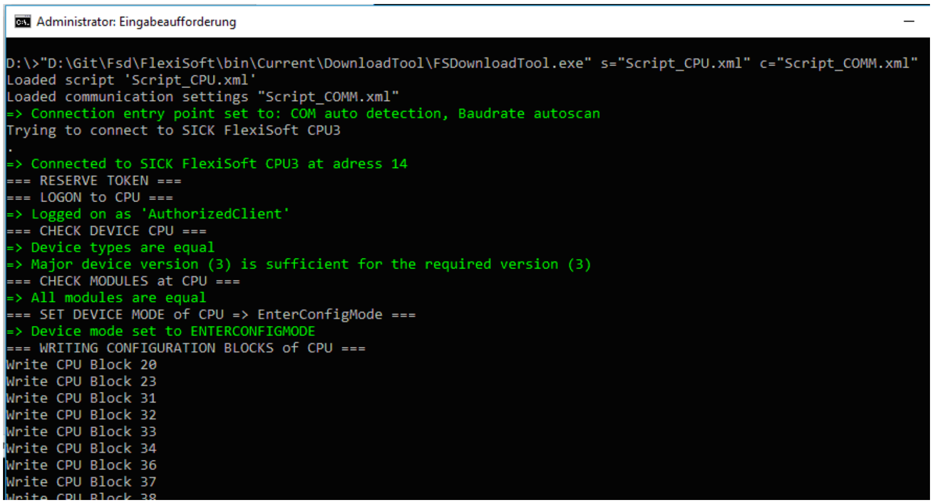


Administrator: Eingabeaufforderung

```
D:\>"c:\Program Files (x86)\SICK\FlexiSoft\FSDownloadTool\FSDownloadTool.exe"
!! Erroneous parameter list

Usage:

s="<Script file>" (mandatory):
  Script file to be executed.

c="<Communication settings file>" (optional):
  Communication settings XML file to be used to connect to the device.
  If not specified the settings file with same base name as the script
  file is tried.

p="<Password>" (optional):
  Password for 'Authorized Customer' user level to be used to log on.
  If not specified the default password is used.

D:\>
```

Figure 378: Explanation of the download tool in the console window

✓ During execution, the individual actions of the download tool are logged so that the success of the execution or any error messages can be checked.

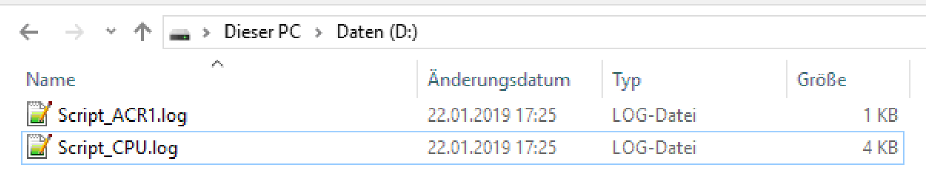


Administrator: Eingabeaufforderung

```
D:\>"D:\Git\Fsd\FlexiSoft\bin\Current\DownloadTool\FSDownloadTool.exe" s="Script_CPU.xml" c="Script_COMM.xml"
Loaded script 'Script_CPU.xml'
Loaded communication settings "Script_COMM.xml"
=> Connection entry point set to: COM auto detection, Baudrate autoscan
Trying to connect to SICK FlexiSoft CPU3
.
=> Connected to SICK FlexiSoft CPU3 at adress 14
=== RESERVE TOKEN ===
=== LOGON to CPU ===
=> Logged on as 'AuthorizedClient'
=== CHECK DEVICE CPU ===
=> Device types are equal
=> Major device version (3) is sufficient for the required version (3)
=== CHECK MODULES at CPU ===
=> All modules are equal
=== SET DEVICE MODE of CPU => EnterConfigMode ===
=> Device mode set to ENTERCONFIGMODE
=== WRITING CONFIGURATION BLOCKS of CPU ===
Write CPU Block 20
Write CPU Block 23
Write CPU Block 31
Write CPU Block 32
Write CPU Block 33
Write CPU Block 34
Write CPU Block 36
Write CPU Block 37
Write CPU Block 38
```

Figure 379: Messages from the download tool during execution

✓ All messages of the download tool are written to a log file in text format. This is saved in the same folder and under the same name as the script file with the `.log` file name extension.



Dieser PC > Daten (D:)

Name	Änderungsdatum	Typ	Größe
Script_ACR1.log	22.01.2019 17:25	LOG-Datei	1 KB
Script_CPU.log	22.01.2019 17:25	LOG-Datei	4 KB

Figure 380: Log files after execution of the download tool

Example

Access to download the configuration for the Flexi Soft main module:

▶
```
"C:\Program
Files\SICK\FlexiSoft\FSDownloadTool\FSDownloadTool.exe"
s="D:\Script_CPU.dsc" c="D:\Script_COMM.dsc" p="meinpasswort"
```

Access to download the configuration for ACR1:

►
```
“C:\Program
Files\SICK\FlexiSoft\FSDownloadTool\FSDownloadTool.exe”
s=”D:\Script_ACR1.dsc” c=”D:\Script_COMM.dsc” p=”meinpasswort”
```

Complementary information

- If no other password is specified, the default password “SICKSAFE” is used for execution.
- The batch file accesses the download tool via default path `C:\Program Files\SICK\FlexiSoft\FSDownloadTool\FSDownloadTool.exe` . The configuration files must be located in the same folder as the batch file. If different paths are to be used, the batch file must be modified accordingly.

16.8.2.3 Automatically detecting the communication interface

Overview

The download script can be run on other computers. The serial COM and the USB communication interface can automatically detect the Flexi Soft download tool. To do so, the communication profile needs to be modified.

Procedure

1. Modify the communication profile for the desired communication interface.
   - Communication profile for automatically detecting the USB interface on the main module (FX3-CPU3):
     ```
     <?xml version="1.0" encoding="utf-16"?>
     <ConnectionConfiguration xmlns:xsi="http://www.w3.org/
     2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/
     XMLSchema" AllowActivateEntryPoint="false">
     <EntryPoint active="true" saveToProjectFile="false"
     activateOnProjectLoad="false" entryName="USB"
     lastModification="2020-09-24T16:17:44.0200081+02:00">
     <COM active="false" lastBaudRate="0" fixBaudRate="0"
     autoscan="true" comAutoDetection="true" />
     <USB active="true" usbDeviceName="FX3-CPU3" usbVID="403" usbPID="6001"
     usbFtdiSerialNumber="" pathChain="" usbDeviceSerialNumber="" />
     </EntryPoint>
     </ConnectionConfiguration>
     ```
   - Communication profile for automatically detecting the serial (COM) interface on the main module:
     ```
     <?xml version="1.0" encoding="utf-16"?>
     <ConnectionConfiguration xmlns:xsi="http://www.w3.org/
     2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/
     XMLSchema" AllowActivateEntryPoint="false">
     <EntryPoint active="true" saveToProjectFile="false"
     activateOnProjectLoad="false" entryName="USB"
     lastModification="2020-09-24T16:17:44.0200081+02:00">
     <COM active="true" portName="0" lastBaudRate="0" fixBaudRate="0"
     autoscan="true" comAutoDetection="true" />
     <USB active="false" usbDeviceName="" usbVID="403"
     usbPID="6001" usbFtdiSerialNumber="" pathChain=""
     usbDeviceSerialNumber="" />
     </EntryPoint>
     </ConnectionConfiguration>
     ```

Complementary information

If several communication interfaces are connected to a main module or several main modules, the Flexi Soft download tool uses the interface that is found first.

16.8.2.4 Return values of the download tool

Overview

The Flexi Soft Download tool provides a return value that depends on the success of the download and can be evaluated automatically. This makes it possible to fully automate the download process, including error checking.

Prerequisites

- The query of the return value must be sent immediately after accessing the download tool.

Return values of the download tool

The return value is written to the `ERRORLEVEL` system variable in Windows and can be read using a batch file, for example:

```
ECHO OFF
:: #################################################################
:: This batch file executes the mentioned download script
:: files sequentially.
:: #################################################################
ECHO =========== Executing download script ============
set OCD=%cd%
cd /D %~dp0
"C:\Program Files\SICK\FlexiSoft\DownloadTool\FSDownloadTool.exe"
s="Script_CPU.dsc" c="Script_COMM.dsc"
IF %ERRORLEVEL% EQU 0 @ECHO Download OK
IF %ERRORLEVEL% NEQ 0 @ECHO Download FAILED! (ERRORLEVEL is
%ERRORLEVEL%)
cd /D %OCD%
```

If successful (i.e., if `ERRORLEVEL` = 0), this batch file issues the following message: `Download OK`

Otherwise, the batch file displays an error message (Errorlevel) and the value of the `ERRORLEVEL` system variable.

Table 236: Meaning of the error messages during automated download

Error message	Description
0	Not an error. Download successfully completed.
03	Invalid password
255	Timeout internal error
1000	RK512 error
1001	Connection error
1002	Loading of the script failed
1003	Device identify test (main module type) failed
1004	Module identity test (number and sequence of expansion modules) failed
1005	Verification unsuccessful
1006	Checking of serial number failed
1010	Internal error

Error message	Description
1011	Invalid parameters when accessing the script
1012	Flexi Soft checksum login does not match

16.8.2.5 Changing the gateway IP address

Overview

You can update the IP address of the Flexi Soft gateways FX0-GENT, FX0-GPNT and FX0-GMOD using the Flexi Soft download tool. The new settings, for example IP address and subnet mask, are passed as parameters when calling the Flexi Soft download tool. The parameters are optional and only required when changing the settings.

Procedure

1. Open the computer's console window (e.g., the Windows command prompt).
2. Call the download tool `FSDownloadTool.exe` with the download tool parameters (see "Directly calling up download tool in Windows", page 507) and also the gateway parameters.

Table 237: Gateway parameters of the download tool for gateway 1

Parameter	Description	Example
Gw1IpAddress:<IP address>	New IP address of gateway 1	`-param=Gw1IpAddress:192.168.78.101`
Gw1SubnetMask:<subnet mask>	New subnet mask of gateway 1	`-param=Gw1SubnetMask:255.255.255.0`
Gw1DefaultGateway:<IP address>	New standard gateway IP address for gateway 1	`-param=Gw1DefaultGateway:192.168.78.0`
Gw1DHCP:<True or False>	Activate DHCP for gateway 1	`-param=Gw1DHCP:True`

Table 238: Gateway parameters of the download tool for gateway 2

Parameter	Description	Example
Gw2IpAddress:<IP address>	New IP address of gateway 2	`-param=Gw2IpAddress:192.168.78.101`
Gw2SubnetMask:<subnet mask>	New subnet mask of gateway 2	`-param=Gw2SubnetMask:255.255.255.0`
Gw2DefaultGateway:<IP address>	New standard gateway IP address for gateway 2	`-param=Gw2DefaultGateway:192.168.78.0`
Gw2DHCP:<True or False>	Activates DHCP for gateway 2	`-param=Gw2DHCP:True`

Example

Set IP address 192.168.78.101 and subnet mask 255.255.255.0 for gateway 1:

►
```
FSDownloadTool.exe -s=Script_CPU.dsc
-c=Script_COMM.dsc -param=Gw1IpAddress:192.168.78.101
-param=Gw1SubnetMask:255.255.255.0
```

Activate DHCP for gateway 2:

►
```
FSDownloadTool.exe -s=Script_CPU.dsc -c=Script_COMM.dsc
-param=Gw2DHCP:True
```

16.8.3 Automated download in Linux

16.8.3.1 Running the download tool using a Linux shell script

Overview

You can run the Flexi Soft download tool in Linux on a x64 architecture or ARM architecture.

Procedure

1. The download script generated by the Flexi Soft Designer can be transmitted from the Window system to the Linux system.
2. Connect the Flexi Soft station to be configured to the Linux system.
3. Adjust the communication profile in the `<script>_COMM.dsc` file of the download script in line with the configuration interface used in the Flexi Soft main module and Linux system.
4. Generate and run Linux shell script.

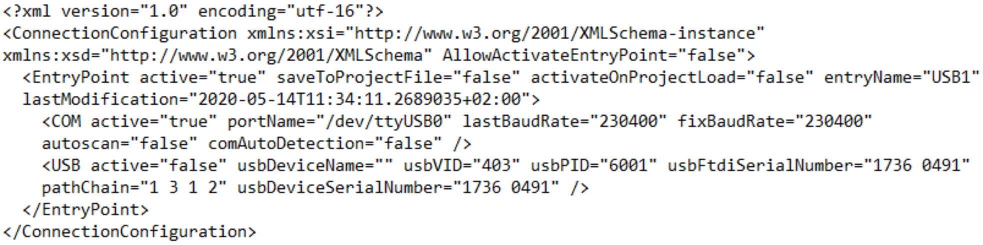
✓ The generated configuration scripts are transferred one after the other to the main module and, if necessary, to the devices on the EFI connections of the main module.

✓ All download tool messages are written to a log file in .txt format. This is stored in the same folder and under the same name as the script file.

Example

Communication profile for USB interface on the Flexi Soft main module (FX3-CPU3):

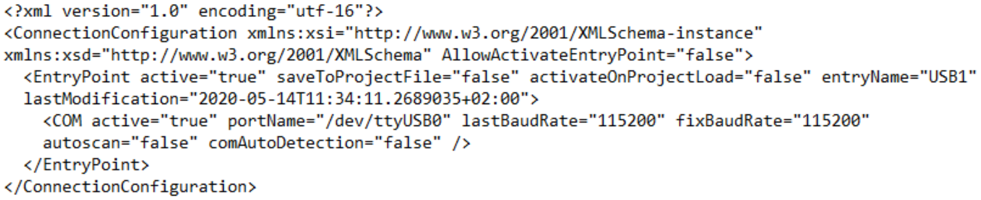
```
<COM active="true" portName="/dev/ttyUSB0" lastBaudRate="230400"
fixBaudRate="230400" autoscan="false" comAutoDetection="false" />
```



```
<?xml version="1.0" encoding="utf-16"?>
<ConnectionConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" AllowActivateEntryPoint="false">
  <EntryPoint active="true" saveToProjectFile="false" activateOnProjectLoad="false" entryName="USB1"
  lastModification="2020-05-14T11:34:11.2689035+02:00">
    <COM active="true" portName="/dev/ttyUSB0" lastBaudRate="230400" fixBaudRate="230400"
    autoscan="false" comAutoDetection="false" />
    <USB active="false" usbDeviceName="" usbVID="403" usbPID="6001" usbFtdiSerialNumber="1736 0491"
    pathChain="1 3 1 2" usbDeviceSerialNumber="1736 0491" />
  </EntryPoint>
</ConnectionConfiguration>
```

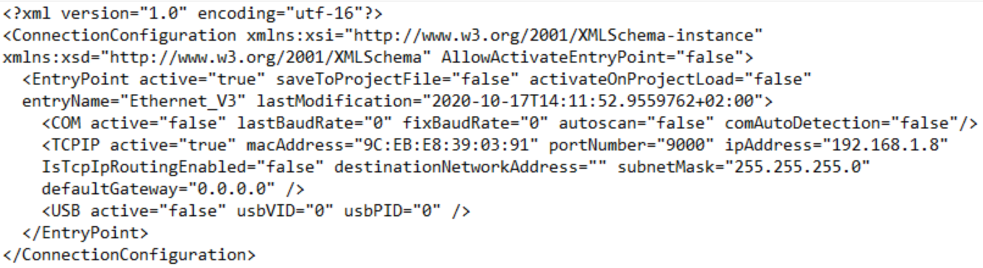
Communication profile for serial interface on the Flexi Soft main module and USB interface on the Linux system:

```
<COM active="true" portName="/dev/ttyUSB0" lastBaudRate="115200"
fixBaudRate="115200" autoscan="false" comAutoDetection="false" />
```



```
<?xml version="1.0" encoding="utf-16"?>
<ConnectionConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" AllowActivateEntryPoint="false">
  <EntryPoint active="true" saveToProjectFile="false" activateOnProjectLoad="false" entryName="USB1"
  lastModification="2020-05-14T11:34:11.2689035+02:00">
    <COM active="true" portName="/dev/ttyUSB0" lastBaudRate="115200" fixBaudRate="115200"
    autoscan="false" comAutoDetection="false" />
  </EntryPoint>
</ConnectionConfiguration>
```

Communication profile for the Ethernet interface on the Flexi Soft gateway:



```
<?xml version="1.0" encoding="utf-16"?>
<ConnectionConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" AllowActivateEntryPoint="false">
  <EntryPoint active="true" saveToProjectFile="false" activateOnProjectLoad="false"
  entryName="Ethernet_V3" lastModification="2020-10-17T14:11:52.9559762+02:00">
    <COM active="false" lastBaudRate="0" fixBaudRate="0" autoscan="false" comAutoDetection="false"/>
    <TCPIP active="true" macAddress="9C:EB:E8:39:03:91" portNumber="9000" ipAddress="192.168.1.8"
    IsTcpIpRoutingEnabled="false" destinationNetworkAddress="" subnetMask="255.255.255.0"
    defaultGateway="0.0.0.0" />
    <USB active="false" usbVID="0" usbPID="0" />
  </EntryPoint>
</ConnectionConfiguration>
```

Generating Linux shell script:

```
#!/bin/bash
```

```
./DownloadToolP s="script_CPU.dsc" c="script_COMM.dsc"
```

16.8.3.2 Directly calling up download tool in Linux

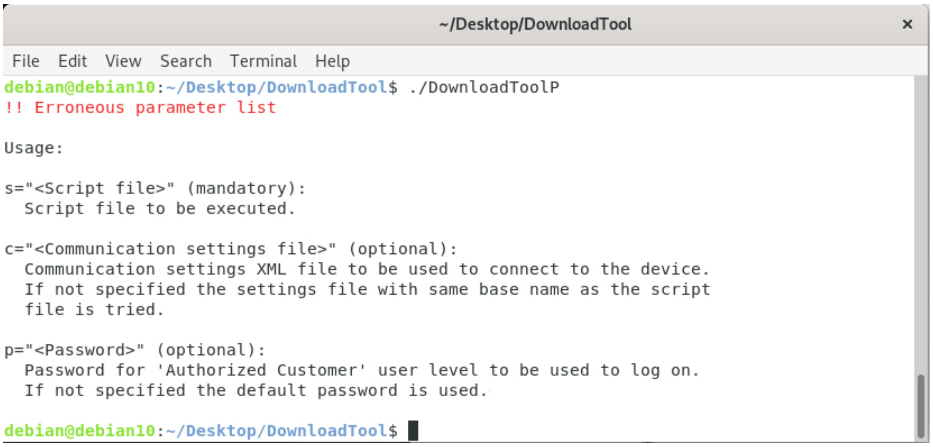
Procedure

1. Open the computer’s console window (e.g., a Linux terminal).
2. Access the `DownloadToolP` download tool using the desired parameters, including the required file paths.

Table 239: Parameters of the download tool

Parameter	Meaning	Description
s	Script	The script to be executed with the configuration to be transferred, e.g., `Script_CPU.dsc`. This parameter must be specified.
c	Communication profile	The communication profile to be used, e.g., `Script_COMM.dsc`. Specifying this parameter is optional. If no communication profile is specified, the download tool may search and use a communication profile with the same path and basic name as the specified script.
p	Password	The password for the **Authorized customer** user group. Specifying this parameter is optional. If no password or an incorrect password is specified, the download tool uses the default password “SICKSAFE”.

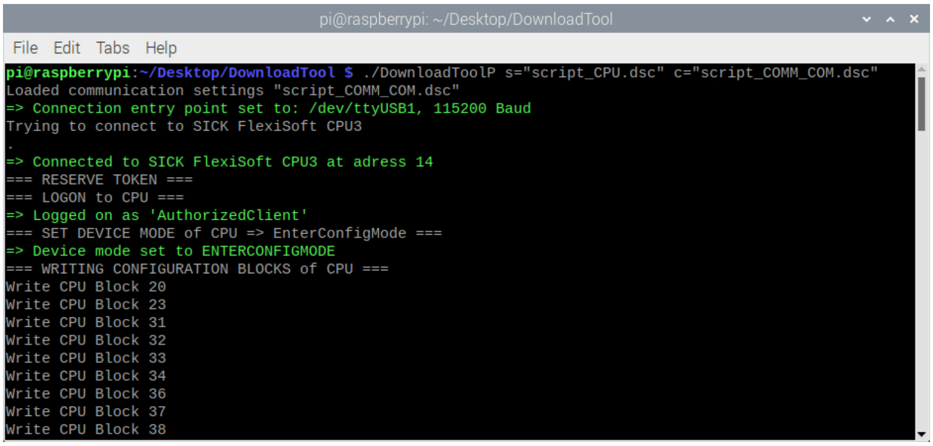
✓ If no or incorrect parameters are specified, an explanation appears in the console window.



```
debian@debian10:~/Desktop/DownloadTool$ ./DownloadToolP
!! Erroneous parameter list

Usage:

s="<Script file>" (mandatory):
  Script file to be executed.

c="<Communication settings file>" (optional):
  Communication settings XML file to be used to connect to the device.
  If not specified the settings file with same base name as the script
  file is tried.

p="<Password>" (optional):
  Password for 'Authorized Customer' user level to be used to log on.
  If not specified the default password is used.

debian@debian10:~/Desktop/DownloadTool$ 
```

Figure 381: Explanation of the download tool in the console window

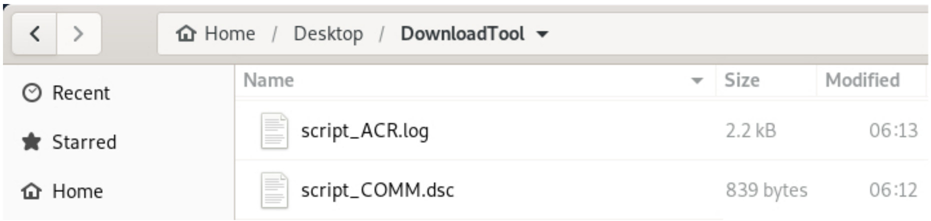
✓ During execution, the individual actions of the download tool are logged so that the success of the execution or any error messages can be checked.



```
pi@raspberrypi:~/Desktop/DownloadTool $ ./DownloadToolP s="script_CPU.dsc" c="script_COMM_COM.dsc"
Loaded communication settings "script_COMM_COM.dsc"
=> Connection entry point set to: /dev/ttyUSB1, 115200 Baud
Trying to connect to SICK FlexiSoft CPU3
.
=> Connected to SICK FlexiSoft CPU3 at adress 14
=== RESERVE TOKEN ===
=== LOGON to CPU ===
=> Logged on as 'AuthorizedClient'
=== SET DEVICE MODE of CPU => EnterConfigMode ===
=> Device mode set to ENTERCONFIGMODE
=== WRITING CONFIGURATION BLOCKS of CPU ===
Write CPU Block 20
Write CPU Block 23
Write CPU Block 31
Write CPU Block 32
Write CPU Block 33
Write CPU Block 34
Write CPU Block 36
Write CPU Block 37
Write CPU Block 38
```

Figure 382: Messages from the download tool during execution

✓ All messages of the download tool are written to a log file in text format. This is saved in the same folder and under the same name as the script file with the `.log` file name extension.



Name	Size	Modified
script_ACR.log	2.2 kB	06:13
script_COMM.dsc	839 bytes	06:12

Figure 383: Log files after execution of the download tool

Example

Access to download the configuration for the Flexi Soft main module:

▶ `./DownloadToolP s="script_CPU.dsc" c="script_COMM.dsc"`

Access to download the configuration for ACR1:

▶ `./DownloadToolP s="script_ACR1.dsc" c="script_COMM.dsc" p="mein-passwort"`

Complementary information

- If no other password is specified, the default password “SICKSAFE” is used for execution.

16.8.3.3 Return values of the download tool

Overview

The Flexi Soft Download tool provides a return value that depends on the success of the download and can be evaluated automatically. This makes it possible to fully automate the download process, including error checking.

Prerequisites

- The query of the return value must be sent immediately after accessing the download tool.

Return values of the download tool

The return value is written to the `$!` system variable in Linux and can be read using a shell script, for example:

```
#!/bin/bash
./DownloadToolP s="script_CPU.dsc" c="script_COMM.dsc"

exit_code=$?
if [ $exit_code -eq 0 ]
then
 echo
 echo Download OK
else
 echo
 echo Download FAILED! Return code is $exit_code
fi
```

If successful (i.e., if `$!` = 0), this shell script issues the following message: `Download OK`

Otherwise, the shell script displays an error message (exit code) and the value of the `$!` variable.

Table 240: Meaning of the error messages during automated download

Error message	Description
0	Not an error. Download successfully completed.
03	Invalid password
255	Timeout internal error
1000	RK512 error
1001	Connection error
1002	Loading of the script failed
1003	Device identify test (main module type) failed
1004	Module identity test (number and sequence of expansion modules) failed
1005	Verification unsuccessful
1006	Checking of serial number failed
1010	Internal error
1011	Invalid parameters when accessing the script
1012	Flexi Soft checksum login does not match

17 Device statuses in the Flexi Soft system

The Flexi Soft system recognizes various device statuses during operation. Some device statuses require intervention by the user, e.g., changing from the **Stop** status to the **Run** status with the help of Flexi Soft Designer. Other statuses are based on the internal self-test performed by the Flexi Soft system, e.g., **Internal error**. The following table provides a summary of the device statuses within the Flexi Soft system.

Table 241: Device statuses and LED indicators on the main module

	Meaning	Notes
MS LED		
○	Supply voltage is out of range	Switch on voltage supply and check at terminals A1 and A2.
☀ Red/ green (1 Hz)	Self-test in progress or system initializing	Please wait...
☀ Green (1 Hz)	System is in Stop status.	Start the application in Flexi Soft Designer.
☀ Green (2 Hz)	Identification in progress (e.g., for Flexi Link)	–
● Green	System is in Run status.	–
☀ Red (1 Hz)	Invalid configuration	Check the module type and version of all modules (main module and the expansion modules) whose MS LED ☀ is flashing red/green. Adjust the configuration if necessary. For more detailed diagnostic information, use Flexi Soft Designer.
☀ Red (2 Hz)	Serious error in the system, probably in this module. The application was stopped. All outputs have been switched off.	Switch the voltage supply off and then on again. If the error persists after repeating the off-on process several times, replace the module. For more detailed diagnostic information, use Flexi Soft Designer.
● Red	Serious error in the system, probably in a different module. The application was stopped. All outputs have been switched off.	Switch the voltage supply off and then on again. If the error persists after repeating the off-on process several times, replace the module that is showing the ☀ red (2 Hz) signal. Otherwise, use the diagnostic functions in Flexi Soft Designer to pinpoint the module concerned.
CV LED		
○	Configuration in progress	Please wait...
☀ Yellow (2 Hz)	Saving configuration data in system plug (non-volatile memory)	Do not interrupt the voltage supply until the save process has been completed.
☀ Yellow (1 Hz)	Unverified configuration	Verify the configuration.
● Yellow	Verified configuration	–

17.1 Changing the device status

Certain status changes within the Flexi Soft system need to be triggered manually in Flexi Soft Designer. These changes to the device status are as follows:

- A switch from the **Stop** to the **Run** status
- A switch from the **Run** to the **Stop** status

To change the device status, click the **Stop application** or **Start application** button in the hardware configuration. The button is located on the left-hand side next to the picture of the modules.

Table 242: The Start and Stop buttons

Button	Function	Description
▶	Start	Sets the Flexi Soft system to the Run status.
■	Stop	Sets the Flexi Soft system to the Stop status.

NOTE
If the configuration is verified, the Flexi Soft system automatically switches to the Run status when the voltage supply is switched on. If the configuration is not verified, the system must be manually switched to the Run status using Flexi Soft Designer.

17.2 Behavior on system startup

When the Flexi Soft safety controller transitions from the Stop status to the Run status, the system behaves as follows:

- The **First logic cycle** status bit of the main module remains set to 1 for the duration of the logic execution time. This status bit is available in the Logic editor as an input element of the main module.
- All timers and statuses – including error statuses of function blocks – are reset.

17.3 Software-controlled reset of the main module

It is possible to reset the main module via the software (i.e., without interrupting the voltage supply) if Flexi Soft Designer is connected to the main module.

Carrying out a software-controlled reset

► Switch to **Hardware configuration**.
► In the context menu of the main module, select the **Software reset** command.
► If necessary, enter the password to log in as an authorized client.
► A confirmation prompt appears. Click on **Yes** to reset the main module.

NOTE
When you reset the main module, the status of the Flexi Soft station outputs may change.

WARNING
Output status change during reset

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

► Before resetting the main module, check whether the status of the system is safe.
► Before resetting the main module, check whether the reset could lead to a dangerous state.
► The **Software reset** command may only be used if the hazardous area has been visually inspected, there is no one within the hazardous area, and nobody will be able to access it while the main module is being reset.

NOTE
If the configuration is verified, the main module automatically returns to the Run status once the reset has been completed. If the configuration is not verified, the main module must be manually restarted using the configuration software.

18 Commissioning

Before proceeding with technical commissioning, the process of configuring the Flexi Soft system must have been completed.

18.1 Wiring and voltage supply

! **NOTICE**
Incorrect connection

The device may be damaged in the event of non-compliance.

- ► When connecting the Flexi Soft system, remember to observe the technical data in the operating instructions titled “Flexi Soft Modular Safety Controller Hardware”.

- ► Connect the individual field devices to the corresponding connections.
- ► Switch on the voltage supply. As soon as the supply voltage is present at connections A1 and A2 of the FX3-CPUx main modules or the FX3-XTIO modules, the Flexi Soft system performs the following steps automatically:
  - o Internal self-test
  - o Load saved configuration
  - o Test validity of loaded configuration
- ► Check each safety capable input, test/signal output, and safety output to determine whether they are behaving as required for the application. The diagnostic information provided by the Flexi Soft LEDs will help you to validate the individual field signals.
- ► Check that the external circuitry, the wiring, plus the choice of control devices and the manner in which they have been arranged on the machine meet the required safety level.
- ► Remedy any faults (e.g., incorrect wiring or crossed signals) at each safety capable input, test/signal output, or safety output.

If these steps cannot be completed successfully, the system does not go into operation. In the event of an error, an LED indicator outputs a corresponding signal (see operating instructions titled “Flexi Soft Modular Safety Controller Hardware”) and the Flexi Soft system sets all the transmitted values to 0 or Low.

18.2 Transferring the configuration

Once the hardware and logic have been configured within the Flexi Soft system and checked for correctness, the configuration from Flexi Soft Designer can be transferred to the Flexi Soft system, see "Transferring the configuration", page 493

18.3 Thorough technical check and commissioning

The machine or system that is being protected by a Flexi Soft safety controller may only be commissioned once all safety functions have undergone a technical check with successful results. Only qualified safety personnel are allowed to perform the technical check.

The checklist for the thorough technical check includes the following points:

- ► Clearly mark all connecting cables and plug connectors on the Flexi Soft system to avoid mix-ups. The Flexi Soft system features several connections of the same design. Therefore, you must make sure that no unplugged connecting cables are connected to the wrong connection point.
- ► Verify the configuration of the Flexi Soft system.

- Check the signal paths and make sure that they have been correctly integrated into higher-level controllers.
- Check that data is transmitted correctly from and to the Flexi Soft safety controller.
- Check the logic program of the safety controller.
- Fully document the configuration for the entire system and individual devices, plus the results of the safety inspection.
- Carry out a full check of the safety functions on the machine or plant/system and make sure that the safety functions are working perfectly.
- Activate write protection for the Flexi Soft system configuration to prevent the configuration from being overwritten accidentally. This stops any further changes from being made until write protection is explicitly canceled.

19 Troubleshooting

19.1 Possible faults

Important information

NOTE
If a fault can result in an undesired valid status, you should evaluate the associated status bits in the logic to initiate suitable measures.

Possible faults and their causes

Table 243: Faults in the logic

Fault	Responses	Causes
Voltage supply A1 / A2 of the main module is lower than the operating range	• Safety controller switches to the **Supply voltage is out of range** status • **Module supply voltage is OK** module status bit = 0 • Status indicators are off	• Fault in the voltage supply • Line break • Interruption due to a fuse
Voltage supply of the main module is higher than the operating range	• Safety controller switches to the serious error status • **Internal tests** module status bit = 0	• Fault in the voltage supply • Short-circuit to other voltage-carrying line

Table 244: Faults in the safety capable inputs (I)

Fault	Responses	Causes
Electro-mechanical switch/safety switch (EMSS), non-contact safety sensors, testable single-beam photoelectric safety switches, Flexi Loop: safety capable input is Low instead of High	• Process data bit of the affected input = 0	• Short-circuit to another line e.g., GND • Error in the sensor • Line break
Electro-mechanical switch/safety switch (EMSS), non-contact safety sensors, testable single-beam photoelectric safety switches, Flexi Loop: test pulses from the associated test output X are not detected correctly (short-circuit detection)	• Process data bit of the affected input = 0 • **Input data status** diagnostic bit = 0 • Module status bit of the affected **Input Ix OK** input = 0	• Short-circuit to other voltage-carrying line • Error in the sensor (for externally tested sensors)
Dual-channel electro-mechanical safety switch (EMSS), ESPE: equivalent/complementary safety capable inputs exhibit different/the same values.	• Process data bit of the affected input = 0 • **Input data status** diagnostic bit = 0 • Module status bit of the affected **Inputs Ix, Iy dual-channel evaluation is OK** input = 0	• Short-circuit to another line e.g., GND • Error in the sensor • Line break • Discrepancy time • Sequence error

Fault	Responses	Causes
Safety pressure mats: safety capable input is Low instead of test pulse signal from the associated test output X	• Process data bit of the affected input pair = 0 • **Input data status** diagnostic bit = 0 • Module status bit of the affected **Input Ix OK** input = 0	• Line break Test output → sensor • Line break Sensor → safety capable input
Safety pressure mats: only one of the two safety capable inputs is High instead of test pulse signal from the associated test output X	• Process data bit of the affected input = 0 • **Input data status** diagnostic bit = 0 • Module status bit of the affected **Inputs Ix, Iy dual-channel evaluation is OK** input = 0	• Short-circuit to other voltage-carrying line
Internal error detected in input evaluation	• Safety controller switches to the serious error status • **Internal tests** module status bit = 0	• Internal device error

Table 245: Faults in the safety outputs (Q)

Fault	Responses	Causes
Auxiliary voltage supply A1 / A2 for the safety outputs is lower than the operating range	• All safety outputs of the module are switched off • **Output data status** diagnostic bit = 0 • **Module supply voltage is OK** module status bit = 0	• Fault in the voltage supply • Line break • Interruption due to a fuse
Auxiliary voltage supply A1 / A2 for the safety outputs is higher than the operating range	• **Internal tests** module status bit = 0 • **Module supply voltage is OK** module status bit = 0	• Fault in the voltage supply • Short-circuit to other voltage-carrying line
Output voltage level is Low instead of High when the output is “On”, overload/overcurrent[1] [2]	• Associated output/associated output pair is switched off • Depending on the load, the affected output may pulsate temporarily until the final switch-off • **Output data status** diagnostic bit = 0 • **Output X short-circuit to Low** module status bit = 0	• Short-circuit or resistance drop under load • Short-circuit to another line e.g., GND
Output voltage level is High instead of Low when the output is “Off”[3]	• All safety outputs of the module are switched off • **Output data status** diagnostic bit = 0[4] • **Output X short-circuit to High** module status bit = 0	• Short-circuit to other voltage-carrying line
Test pulses are not detected correctly when the output is “On”[3] [5]		• Short-circuit to other voltage-carrying line • Capacitive load too high

Fault	Responses	Causes
Internal error detected	• Safety controller switches to the serious error status • **Internal tests** module status bit = 0	• Internal device error • Capacitive load too high

[1] Depending on the voltage supply used, an overcurrent can also result in the voltage supply dropping out.

[2] Reset of the error: set the process data bit of the safety output to 0.

[3] Reset of the error: the process data bits for all safety outputs of the module are simultaneously 0 and the output level is Low.

[4] In the case of a High instead of a Low on a safety output, the supply to all safety outputs is switched off internally. If the cause of a short-circuit is in the wiring to 24 V, then the affected signal remains High and all other signals switch to Low.
Check if this is an undesired but valid signal value for the receiver, e.g., for the switching of the monitoring case of a SICK safety laser scanner by means of a complementary signal.

[5] Depending on the size of the capacitive load, this may lead in certain cases to an incorrect interpretation as an internal error since the effect on the output voltage is only temporary.

Table 246: Faults in test outputs (X), non-safe outputs (Y)

Fault	Responses	Causes
Voltage supply A1 / A2 for the test outputs is lower or higher than the operating range	See voltage supply A1 / A2 of the main module. The test outputs are supplied from the voltage supply of the main module.	
Output voltage level is Low instead of High when “On”, overload/overcurrent[1]	• Associated output/associated output pair are switched off temporarily (thermal overload limiting). • No status indicator or diagnostic message available for this fault.	• Short-circuit or resistance drop under load • Short-circuit to another line e.g., GND
Output voltage level is High instead of Low when “Off”	• No status indicator or diagnostic message available for this fault.	• Short-circuit to other voltage-carrying line

[1] Depending on the voltage supply used, an overcurrent can also result in the voltage supply dropping out.

Serious error status

Consequences of the serious error status:

- All applications are stopped.
- All safety outputs are switched off.
- All process data are 0
- Evaluation of the diagnostic bits and module status bits in the logic is no longer possible.
- Only limited diagnostics can be performed in the serious error status.

Alternatives to resetting the serious error status:

- Restart by switching the voltage supply off and on again
- Software reset using the configuration software

Complementary information

The status indicators and diagnostic messages may provide additional information.

Further topics

- "Module status bits of the main module", page 92
- "Input data status and output data status", page 93
- "Module status bits of the expansion modules", page 94
- "Device statuses in the Flexi Soft system", page 516

19.2 Troubleshooting

⚠ **WARNING**

Malfunction of the protective device

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

- ► Immediately put the plant/machine out of operation if it is not possible to clearly allocate the fault and safely remedy it.
- ► After remedying a fault, carry out an effects analysis and check all affected safety functions.

Current error messages and error codes are displayed under **Diagnostics** if you have established a connection to the Flexi Soft system. For additional information, see "Diagnostics", page 76.

The “Flexi Soft Modular Safety Controller Hardware” operating instructions contain a list of the LED error indicators and the associated error codes, possible causes of errors, and troubleshooting measures.

19.3 Creating diagnostics package

Overview

You can create a diagnostics package for service purposes.

Procedure

1. In the menu bar under **Extras**, select the **Generating diagnostics package** function.
2. Using the **Current project**, **Report** and **Complete event protocol** buttons, select which file groups should contain the diagnostics package.

✓ The button is blue when the file group has been added to the diagnostics package.

3. If necessary, add other files with the **Add files...** button.
4. If necessary, in the **Contained files** area, deactivate individual files using the check-boxes which should not be included in the diagnostics package.
5. If the diagnostics package must not contain any sensitive safety information, deactivate the **Allow sensitive safety information** checkbox.
6. Select **Save diagnostics package.**
7. Select the directory and save the diagnostics package.
8. Send diagnostics package (.zip) and a description of the error to SICK Service.

20 Deinstallation

To remove Flexi Soft Designer, proceed as follows:

- ► In the Windows Start menu in the **SICK/Flexi Soft Designer** program folder, select the **Remove Flexi Soft Designer** command and follow the instructions.

Or:

- ► Uninstall Flexi Soft Designer in the Control Panel.

21 Ordering information

21.1 Ordering information for Flexi Soft Designer functions requiring a license

Table 247: Ordering information for Flexi Soft Designer functions requiring a license

Part	Part number
Automated download of company license	1613943
Automated download of group license	1613944
Automated download of test license	1615553

22 Glossary

AGV	Automated guided vehicle
CCF	Common cause failure. Failures of various units due to a single event, whereby these failures do not cause each other.
Control input	A control input receives signals, e.g. from the machine or from the control. Use of control inputs is how the protective device receives information about the conditions at the machine, e.g., if there is a change of operating mode. If the protective device is configured appropriately, it will activate a different monitoring case after receiving a new control input. The control input information must be transmitted reliably. Generally, at least 2 separate channels are used to do this. Depending on the device, a control input can be realized as a static control input or a dynamic control input.
Dangerous state	A dangerous state is a status of the machine or facility, where people may be injured. Protective devices prevent this risk if the machine is operated within its intended use. The figures in this document always show the dangerous state of the machine as movement of a machine part. In practice, there are different dangerous states, such as: • Machine movements • Electrical parts • Visible and invisible beam • A combination of multiple hazards
EDM	External device monitoring
EFI-pro	EFI-pro [13] is an Ethernet-based network for general and safety-related data communication. EFI-pro allows for easy device identification, addressing, configuration, and diagnostics. Devices can exchange data via EFI-pro, such as control signals, safety-related shut-off signals, and diagnostics data. An EFI-pro network can have various structures (topologies), e.g., with cables running from one central device to all the others (star topology) or with cables running from one device to the next (line topology). Different topologies can be combined within one EFI-pro network to form a hybrid topology.
Electro-sensitive protective device	An electro-sensitive protective device is a device or system of devices for safety-related detection of people or parts of the body. It is used to protect people from machines and facilities that pose a risk of injury. It triggers the machine or facility to adopt a safe state before a person is exposed to a hazardous situation. Examples: Safety light curtain, safety laser scanner.
ESPE	Electro-sensitive protective device
EtherCAT	EtherCAT® (Ethernet for Control Automation Technology) is an Ethernet-based network used in industrial automation. With the Safety over EtherCAT® (FSoE, FailSafe over EtherCAT) protocol extension, EtherCAT® is also suitable for safety-related data communication.

[13] Enhanced Function Interface-pro based on EtherNet/IP™ – CIP Safety™.

EtherNet/IP	EtherNet/IP™ (EtherNet Industrial Protocol) is an Ethernet-based network used in industrial automation. EtherNet/IP™ implements the CIP™ (Common Industrial Protocol) based on the Ethernet and TCP/IP protocol family. EtherNet/IP™ with the CIP Safety™ protocol extension is also suitable for safety-related data communication.
External device monitoring	The external device monitoring (EDM) monitors the status of downstream contactors. In order to use external device monitoring, positively guided contactors must be used to switch off the machine. If the auxiliary contacts of the positively guided contactors are connected to the external device monitoring, the external device monitoring checks whether the contactors switch correctly when the OSSDs are switched off.
Hazardous area	Hazardous area is any space within and/or around machinery in which a person can be exposed to a hazard. (ISO 12100)
Incremental encoder	An incremental encoder generates electrical pulses proportional to a movement. Various physical quantities can be derived from these pulses, e.g. speed and distance covered.
Monitoring case	A monitoring case indicates the machine status to the sensor. Generally, one field set is assigned to each monitoring case. The sensor receives a defined signal for the current machine status. When a signal change occurs, the sensor activates the monitoring case and thereby the field set that is associated with the new machine status.
PL	Performance level (ISO 13849)
PROFINET	PROFINET (Process Field Protocol) is an Ethernet-based network used in industrial automation. With PROFIsafe , PROFINET is also suitable for safety-oriented data communication.
Protective field	The protective field is the area in which the test object specified by the manufacturer is detected by the electro-sensitive protective equipment (ESPE). As soon as the electro-sensitive protective device detects an object in the protective field, it switches the associated safety outputs to the OFF state. This signal can be passed to controllers resulting in the dangerous state coming to an end, e.g. to stop the machine or the vehicle.

Reset	When a protective device has sent a stop command, the stopped state must be maintained until a reset device is activated and the machine can be restarted in a second step. The reset brings the protective device back to the monitoring state after it has sent a stop command. The reset also quits the start-up or restart interlock of a protective device, so that the machine can be restarted in a second step. The reset must only be possible, when all safety functions and protective devices are functional. The reset of the protective device must not introduce any movement or dangerous situations itself. The machine is only permitted to start after the reset once a separate start command has been sent. • Manual resets are performed using a separate, manually operated device, such as a reset pushbutton. • Automatic resets by the protective device are only permitted in special cases, if one of the following conditions is met: o It must not be possible for people to be in the hazardous area without triggering the protective device. o It must be ensured that no people are in the hazardous area during or after the reset.
Resolution	The resolution of an active opto-electronic protective device (also known as the sensor detection capability) is the minimum size of an object for it to be reliably detected.
Response time	The protective device's response time is the maximum time between the occurrence of the event leading to the sensor's response and supply of the switch-off signal to the protective device's interface (for example OFF state of the OSSD pair).
Restart interlock	The restart interlock prevents the machine from automatically starting up, for example after a protective device has responded while the machine is operating or after changing the machine's operating mode. The restart interlock can be implemented in the protective device or in the safety controller. A command to reset the protective device must be given, for example using a reset pushbutton, before the machine can be restarted.
Safety function	Function of a machine whose failure can result in an immediate increase of the risk(s). (ISO 12100)
Safety output	A safety output provides safety-related information. Safety outputs are OSSDs, for example, or safety-related information on a safety-related network.
SIL	Safety integrity level
Stop Category 1	Robot motion is stopped with power available to the robot to achieve the stop and then removal of power when the stop is achieved. It is a controlled stop, where the robot will continue along the programmed path. Power is removed as soon as the robot stands still.
Stop Category 2	A controlled stop with power left available to the robot. The safetyrelated control system monitors that the robot stays at the stop position.

Warning field	The warning field monitors larger areas than the protective field. Simple switching functions can be triggered with the warning field, e.g. a warning light or an acoustic signal can be triggered if a person approaches, even before the person enters the protective field. The warning field must not be used for safety applications.

23 List of figures

24 List of tables

8012480/1KOF/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.