

samos[®]PLAN

samos[®]PRO configuration software

Operating instructions

Doc.-No. BA000518

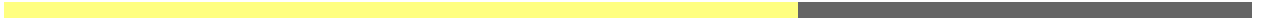
Updated: 04/2013 (Rev. F)

© 2011 Wieland Electric GmbH

This document is protected by the law of copyright, whereby all rights established therein remain with the company Wieland Electric GmbH. Reproduction of this document or parts of this document is only permissible within the limits of the legal determination of Copyright Law. Alteration or abridgement of the document is not permitted without the explicit written approval of the company Wieland Electric GmbH.

Microsoft, Windows 98, Windows NT, Windows 2000, Windows XP and .NET Framework are registered trademarks of Microsoft Corporation. Other product names and trademarks mentioned in these operating instructions are trademarks or registered trademarks of the respective owners.





Contents

1	About This Document	8
1.1	Function of this document	8
1.2	Scope and version	8
1.3	Target group	8
1.4	Function and structure of these software operating instructions	9
1.4.1	Recommendations for familiarising yourself with the software	9
1.4.2	Recommendations for experienced users	9
1.5	Symbols and notations used	9
2	On Safety	10
2.1	Qualified safety personnel	10
2.2	Correct use	10
3	Version, Compatibility and Features	11
4	Installation and Removal	12
4.1	System requirements	12
4.2	Installation	12
4.3	Update	12
4.4	Removal	12
4.5	Troubleshooting	13
5	The Graphical User Interface	14
5.1	Start view	14
5.2	Setting the desired language	14
5.3	Standard views	15
5.4	Positioning windows	16
5.5	“Hardware configuration” view	16
5.5.1	Exercise for configuring the samos [®] PRO modules	18
5.5.2	Module status bits in the Hardware configuration view	19
5.5.3	Exercise for configuring the connected devices	20
5.5.4	Expanding elements	21
5.5.5	Parametrization of connected elements	21
5.5.6	Customized elements	24
5.5.7	Export and import of a configuration	28
5.6	Logic editor view	30
5.6.1	Exercise for using the logic editor	31
5.6.2	Validation of the configuration	32
5.6.3	CPU status bits in the logic editor	32
5.6.4	EFI I/O error status bits in the logic editor	33
5.6.5	Module input and output status bits in the logic editor	33
5.6.6	CPU markers	34
5.6.7	Jump addresses	34
5.6.8	I/O matrix	36
5.6.9	Tag name editor	37
5.7	Report view	38
5.8	Diagnostics view	39
6	Connecting to the samos[®]PRO system	41
6.1	First steps for establishing a connection	41
6.1.1	Connecting the PC to the samos [®] PRO system via RS-232	41

6.1.2	Online status and background color	42
6.2	Editing the communication settings	43
6.3	Establishing a connection with the samos®PRO system	46
6.4	User levels in the samos®PLAN	47
6.5	Identify project	48
7	samos®NET	49
7.1	samos®NET overview	49
7.1.1	System requirements and restrictions for samos®NET	49
7.2	Function principle	49
7.2.1	samos®NET address	50
7.2.2	samos®NET ID	50
7.3	Getting started	51
7.3.1	Connecting to an existing samos®NET system	51
7.3.2	Setting up a samos®NET project in the samos®PLAN	54
7.3.3	samos®NET configuration	55
7.3.4	Transferring and verifying the samos®NET configuration	57
7.4	samos®NET functions	60
7.4.1	samos®NET system: System overview	62
7.4.2	samos®NET system: Process image	63
7.4.3	samos®NET system: Network settings	63
7.4.4	samos®NET stations: samos®NET data in the logic editor	65
7.4.5	samos®NET stations: Station X view and process image	66
7.4.6	samos®NET stations: Teach function	67
7.4.7	samos®NET teaching status and diagnostics	69
7.5	samos®NET troubleshooting	70
7.5.1	samos®NET ID mismatch	70
8	Logic programming – Function blocks	71
8.1	General safety notes for the logic programming	71
8.2	Function block overview	72
8.3	Function block properties	73
8.4	Input and output signal connections of function blocks	73
8.4.1	Function block input connections	73
8.4.2	Inversion of input connections	73
8.4.3	Function block output connections	74
8.5	Parametrization of function blocks	75
8.5.1	Timer values and logic execution time	75
8.5.2	Fault present output	75
8.6	Logic function blocks	76
8.6.1	NOT	76
8.6.2	AND	77
8.6.3	OR	78
8.6.4	XOR (exclusive OR)	79
8.6.5	XNOR (exclusive NOR)	79
8.6.6	RS Flip-Flop	80
8.6.7	JK Flip-Flop	80
8.6.8	Clock generator	81
8.6.9	Event counter (Up, Down and Up and down)	82
8.6.10	Fast shut off and Fast shut off with bypass	84
8.6.11	Edge detection	89
8.6.12	Binary encoder	90
8.6.13	Binary decoder	92

Contents

8.6.14	Message generator.....	95
8.6.15	Routing 1:N.....	97
8.6.16	Routing N:N.....	98
8.7	Application-specific function blocks	98
8.7.1	Reset.....	98
8.7.2	Restart	99
8.7.3	Off-delay timer.....	100
8.7.4	Adjustable off-delay timer	101
8.7.5	On-delay timer.....	102
8.7.6	Adjustable on-delay timer.....	103
8.7.7	EDM (External device monitoring)	104
8.7.8	Valve monitoring	105
8.7.9	User mode switch	108
8.7.10	Switch synchronization	110
8.7.11	Error output combination	113
8.7.12	Ramp down detection	114
8.7.13	Start warning	119
8.8	Function blocks for dual channel evaluation.....	123
8.8.1	Single-channel evaluation	124
8.8.2	Dual-channel evaluation (1 pair) and discrepancy time	124
8.8.3	Double dual-channel evaluation (2 pair synchronization evaluation) and synchronization time	126
8.8.4	Emergency stop.....	128
8.8.5	Magnetic switch	129
8.8.6	Light curtain monitoring.....	129
8.8.7	Safety gate monitoring.....	130
8.8.8	Two-hand control type IIIA.....	132
8.8.9	Two-hand control type IIIC.....	132
8.8.10	Multi operator (multiple two-hand control).....	134
8.9	Function blocks for Parallel muting, Sequential muting and Cross muting.....	135
8.9.1	Overview and general description.....	135
8.9.2	Parameters of the function blocks	138
8.9.3	Information on wiring.....	144
8.9.4	State transition from Stop to Run	144
8.9.5	Error states and information on resetting	145
8.9.6	Parallel muting.....	145
8.9.7	Sequential muting	147
8.9.8	Cross muting – direction of movement only forwards or backwards.....	149
8.9.9	Cross muting – material transport in both directions.....	150
8.10	Function blocks for press contact monitoring	152
8.10.1	Overview and general description.....	152
8.10.2	Eccentric press contact	153
8.10.3	Universal press contact	158
8.11	Function blocks for press cycle control	165
8.11.1	Press setup	165
8.11.2	Press single stroke.....	167
8.11.3	Press automatic.....	171
8.11.4	N-break (press with N-PSDI mode)	173
8.12	User defined function blocks.....	180
8.12.1	Grouped function block.....	180
8.12.2	Customized function block.....	184
8.13	Simulation of the configuration.....	186

8.14	Force mode	188
9	I/O modules	193
9.1	Dual channel evaluation and discrepancy time	193
10	Transferring the System Configuration	195
10.1	Transferring project data to the safety controller	195
10.2	Compatibility check.....	195
10.3	Verification of the configuration	195
10.4	Activating the write protection of the configuration in the control system	199
11	Device States of the samos®PRO System.....	200
11.1	Changing the device state	200
11.2	Behaviour on startup.....	201
11.3	Software reset of the controller module	201
12	Technical Commissioning.....	202
12.1	Wiring and voltage supply	202
12.2	Transferring the configuration	202
12.3	Technical test and commissioning	202
13	Troubleshooting	204
14	Annex	205
14.1	Example application report: Wood processing machine.....	205
14.2	List of tables.....	216
14.3	List of illustrations.....	218

1 About This Document

Please read this chapter carefully before working with these software operating instructions and the samos[®]PRO system.

1.1 Function of this document

For the samos[®]PRO system there are three operating instructions with clearly distinguished fields of application as well as mounting instructions for each module.

- The mounting instructions are enclosed with each samos[®]PRO module. They inform on the basic technical specifications of the modules and contain simple mounting instructions. Use the mounting instructions when mounting samos[®]PRO safety controllers.
- The samos[®]PRO hardware operating instructions describe all samos[®]PRO modules and their functions in detail. Use the Hardware operating instructions in particular to configure samos[®]PRO safety controllers.
- The samos[®]PRO gateways operating instructions describe all samos[®]PRO gateways and their functions in detail.
- The samos[®]PRO software operating instructions describe the software-supported configuration and parametrization of the samos[®]PRO safety controllers. In addition, the software operating instructions contain the description of the diagnostics functions that are important for operation and detailed information for the identification and elimination of errors. Use the Software operating instructions in particular for the configuration, commissioning and operation of samos[®]PRO safety controllers.

1.2 Scope and version

These software operating instructions apply for the samos[®]PLAN software version V1.0.0 or higher and SP-SCON and SP-SCON-NET firmware version V1.00.0 or higher.

This version of the operating instructions describes version V1.3.0 of the samos[®]PLAN software.

These operating instructions are original operating instructions.

Tab. 1: Overview of the samos[®]PRO documentation

Document	Title	Document number
Operating instructions	samos [®] PLAN Software	BA000518
Operating instructions	samos [®] PRO Hardware	BA000497
Operating instructions	samos [®] PRO-Gateways	BA000587
Mounting instructions	SP-SCON/SP-SCON-NET (controller modules of the samos [®] PRO modular safety control system)	BA000513
Mounting instructions	SP-SDI/SP-SDIO (extension modules of the samos [®] PRO modular safety control system)	BA000515
Mounting instructions	SP-EN-IP, SP-EN-MOD, and SP-EN-PN (samos [®] PRO Ethernet gateways)	BA000583
Mounting instructions	SP-PROFIBUS-DP (samos [®] PRO PROFIBUS DP gateway)	BA000572
Mounting instructions	SP-CANopen (CANopen gateway)	BA000683

1.3 Target group

The samos[®]PRO software operating instructions are addressed to planning engineers, designers and operators of systems into which a samos[®]PRO modular safety controller is integrated. It also addresses persons who carry out initial commissioning or who are in charge of servicing or maintaining a samos[®]PRO system.

These software operating instructions do not provide instructions for operating the machine or system in which a samos®PRO safety controller is integrated. Information of this kind will be found in the operating instructions for the machine or system.

1.4 Function and structure of these software operating instructions

These software operating instructions instruct the technical personnel of the machine manufacturer or machine operator in the software configuration, operation and diagnostics of a samos®PRO system using the samos®PLAN software. It only applies in combination with the "samos®PRO Hardware Operating Instructions".

Chapter 2 "On Safety" on page 10 contains fundamental safety instructions. These instructions must be read.

We also refer you to our homepage on the Internet at http://www.wieland-electric.com/public/we_en_products_safety_samos_PRO.AxCMS.

NOTE

There, you can download the following files:

- samos®PLAN
- Hardware and software operating instructions
- EDS and GSD files

1.4.1 Recommendations for familiarising yourself with the software

We recommend the following procedure for users who want to familiarise themselves with samos®PLAN for the first time:

- ➔ Read chapter 5 "The Graphical User Interface" on page 14 to familiarise yourself with the graphical user interface and do the exercises for the configuration of example applications.

1.4.2 Recommendations for experienced users

We recommend the following procedure for experienced users who have already worked with samos®PLAN:

- ➔ Familiarise yourself with the most recent version of the software by reading chapter 1.2 "Scope and version" on page 8.
- ➔ The table of contents lists all functions provided by the samos®PLAN. Use the table of contents to obtain information about the basic functions.

1.5 Symbols and notations used

Notes provide special information on a device or a software function.

NOTE

Warning!

A warning indicates concrete or potential dangers. Warnings are intended to protect you from harm and help avoid damage to devices and systems.

Read warnings carefully and follow them!

Otherwise the safety function may be impaired and a dangerous state may occur.



WARNING

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

Menus and commands

2 On Safety

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

➔ Please read this chapter carefully before working with a samos®PRO system.

2.1 Qualified safety personnel

The samos®PRO system must be installed, configured, commissioned and serviced only by qualified safety personnel. Qualified safety personnel are defined as persons who

- have undergone the appropriate technical training and
- have been instructed by the responsible machine operator in the operation of the machine and the current valid safety guidelines and
- have access to the samos®PRO operating instructions and have read and familiarised themselves with them.

2.2 Correct use

The samos®PLAN software is used to configure a safety controller consisting of modules of the samos®PRO system.

The samos®PRO system may only be used by qualified safety personnel and only at the machine at which it was mounted and initially commissioned by qualified safety personnel in accordance with the samos®PRO hardware and software operating instructions.



Wieland Electric GmbH accepts no claims for liability if the software or the devices are used in any other way or if modifications are made to the software or the devices — even in the context of mounting and installation.

➔ Observe the safety instructions and protective measures of the samos®PRO hardware and software operating instructions!

When implementing a safety-relevant functional logic, ensure that the regulations of the national and international rules and standards are observed, in particular the controlling strategies and the measures for risk minimisation that are mandatory for your application.

NOTES

- When mounting, installing and using the samos®PRO system, observe the standards and directives applicable in your country.
- The national and international rules and regulations apply to the installation and use as well as commissioning and periodic technical inspection of the samos®PRO safety controller, in particular:
 - Machinery Directive 2006/42/EC,
 - EMC Directive 2004/108/EC,
 - Provision and Use of Work Equipment Directive 2009/104/EC and the supplementary Directive 35/63/EC,
 - Low-Voltage Directive 2006/95/EC,
 - Work safety regulations and safety rules.
- The samos®PRO hardware and software operating instructions must be made available to the operator of the machine where a samos®PRO system is used. The machine operator is to be instructed in the use of the device by qualified safety personnel and must be instructed to read the operating instructions.

3 Version, Compatibility and Features

For the samos®PRO product family several firmware versions and function packages exist that allow different functions. This section gives an overview which firmware version, which function package and/or which version of the samos®PLAN software is required to use a certain function or device.

Feature	Minimum required version		
	CPU	SP-SDIO/SP-SDI	samos®PLAN
Logic offline simulation	– ¹⁾	–	V1.2.0
Logic import/export	–	–	V1.3.0
Automatic wiring diagrams	–	–	V1.3.0
Central tag name editor	–	–	V1.3.0
samos®NET (only with SP-SCON-NET)	V2.00 (Step 2.xx)	–	V1.3.0
Logic block documentation within samos®PLAN	–	–	V1.3.0
Input/output relation matrix	–	–	V1.3.0
Invertable inputs for the AND, OR, RS Flip-Flop and Routing N:N function blocks	V2.00 (Step 2.xx)	–	V1.3.0
New function blocks (e.g. Ramp down, Fast Shut Off with Bypass, Adjustable delays)	V2.00 (Step 2.xx)	–	V1.3.0
Verification without identical hardware possible	V2.00 (Step 2.xx)	–	V1.0.0
Test pulses on SP-SDIO can be switched off	–	V2.00 (Step 2.xx)	V1.3.0
Status input data and Status output data in logic	V1.11 (Step 1.xx)	V2.00 (Step 2.xx)	V1.3.0
Several safety mats per SP-SDIO/SP-SDI	–	V1.13	V1.3.0
Device	CPU	SP-SDIO/SP-SDI	samos®PLAN
ProfiNET, Modbus TCP, Ethernet/IP gateways	V1.11 (Step 1.xx)	–	V1.2.0
CANopen, DeviceNet, gateways	V1.11 (Step 1.xx)	–	V1.3.0
0-Speed Monitor SNS 40x4K	–	–	V1.3.0
ROHS conformity SP-SDIO	–	V1.01 ²⁾	–

Tab. 2: Required firmware and software versions

- You can find the firmware version on the type label of the samos®PRO modules in the field "Software version".
- In order to use modules with a newer firmware version, a new samos®PLAN version is required.
- The version of the samos®PLAN can be found in the **Extras** menu under **Info**.
- The newest version of the samos®PLAN can be obtained on the Internet under www.wieland-electric.com (goto "Support > Download Center > Safety Technology > samos®PRO").
- The function package (Step 1.xx or Step 2.xx) must be selected in the samos®PLAN hardware configuration. Function package Step 2xx is available with samos®PLAN 1.3.0 and higher.
- In order to use function package Step 2.xx, the respective module must have at least firmware version V2.00. Otherwise you will receive an error message when you try to upload a configuration using Step 2.xx to a module with a lower firmware version.
- Newer modules are downward compatible so that any module can be replaced by a module with a higher firmware version.
- You will find the device's date of manufacture on the type label in the format ww/yyyy (yyyy = year, ww = calendar week).

NOTES

¹⁾ "–" means "any" or "not applicable".

²⁾ All other modules from product launch onwards.

4 Installation and Removal

4.1 System requirements

Recommended system configuration:

- Windows 2000, Windows NT, Windows XXXP (32 Bit/64 Bit) or Windows Vista (32 Bit/64 Bit)
- 1 GHz processor
- 1 GB work memory
- 1024 × 768 pixel screen resolution
- 300 MB free hard disk memory

samos®PLAN is a .NET Framework application. It requires .NET Framework Version 2.0 (included on the samos®PLAN CD-ROM) or higher. Information on the current .NET Framework versions and supported operating systems is available on the Internet at <http://www.microsoft.com/>

Microsoft .NET Framework Version 2.0 or higher and any other components that may be needed can also be downloaded from <http://www.microsoft.com/downloads/>.

4.2 Installation

Insert the CD-ROM of the samos®PLAN into the corresponding drive of your computer in order to begin with the installation. If the AutoRun function for CDs is activated on your PC, the start screen is displayed after the CD has been inserted. Click **Install samos®PLAN** and follow the further instructions.

If the AutoRun function for CDs is not activated on your PC, start the installation manually by running the **setup.exe** file on the CD-ROM.

4.3 Update

The most recent version of the samos®PLAN is available on the Internet at www.wieland-electric.com (goto "Support > Download Center > Safety Technology > samos®PRO"). New software versions may contain new functions and support new samos®PRO modules.

Remove the old software version before installing a new one. The working directory in which the project data are stored is not overwritten during the new installation and is retained.

4.4 Removal

The software can be removed as follows:

- ➔ In the Windows Start menu, start **Remove samos®PLAN** in the samos®PLAN program folder.

4.5 Troubleshooting

Error/Error message	Cause	Recification
<p>When samos®PLAN is started, the following or a similar error message is displayed: "DLL not found – the Dynamic Link Library mscoree.dll was not found in the specified path. Specify the registration key HKLM\Software\Microsoft\NETFramework\InstallRoot so that it refers to the installation location of the .NET Framework."</p>	<p>Microsoft .NET Framework is not installed on the PC.</p>	<p>Install a suitable version of Microsoft .NET Framework. Ask your system administrator if appropriate. .NET Framework is available for downloading on the internet pages of Microsoft.</p> <p>Note: Install .NET Framework 2.0</p>

Tab. 3: Errors and error elimination

5 The Graphical User Interface

NOTE

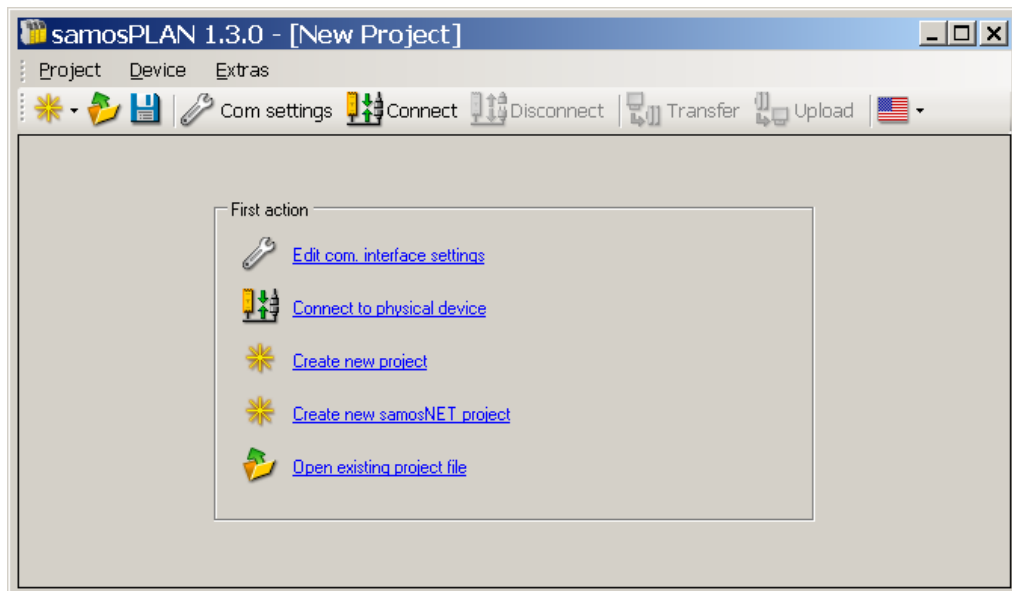
This chapter familiarises you with the basic elements of the graphical user interface as an introduction. This chapter does not give any information on the configuration of samos®PRO modules nor any instructions for logic programming. This chapter is only intended to explain the fundamental functioning of the samos®PLAN on the basis of a small section of the functions. Experienced users of samos®PLAN can skip this chapter.

5.1 Start view

After the software has been started, the start view is displayed. The user can specify here with which of the following actions he wants to start:

- adapting the parameters of the serial interface
- establishing the connection to a physically connected device
- creating a new project
- creating a new samos®NET project
- opening an existing project file

Fig. 1: Start view with selection of the action



5.2 Setting the desired language

- ➔ Click the flag icon in the menu bar at the far right and select the desired language version.

5.3 Standard views

The samos®PLAN has the following views that can be accessed via buttons below the menu bar.

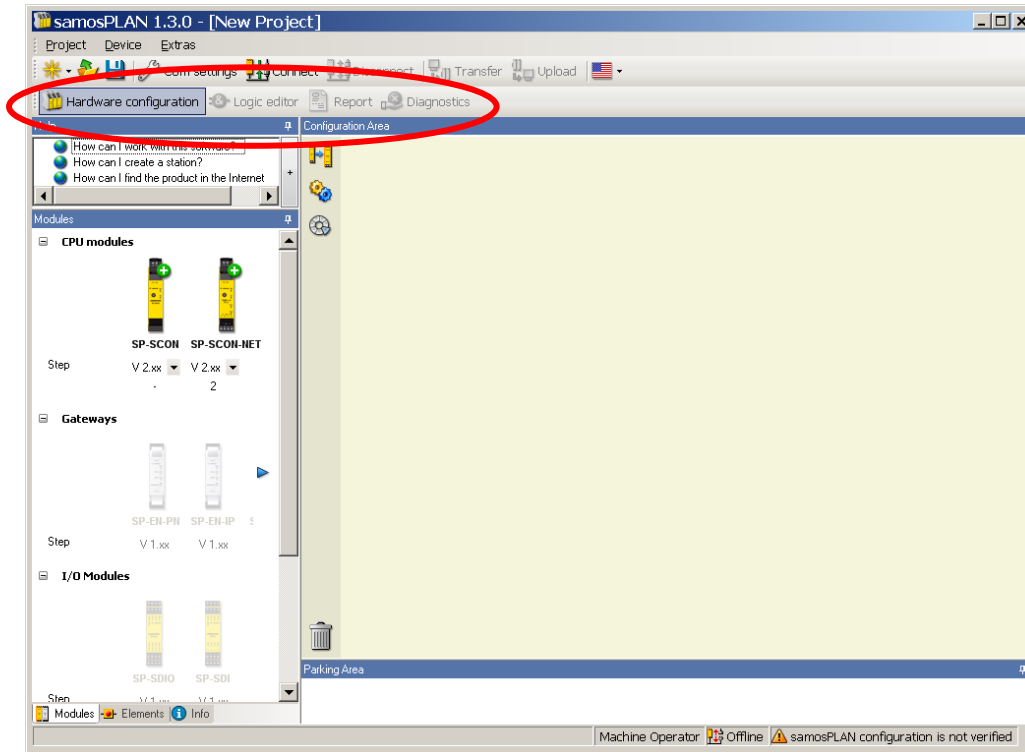


Fig. 2: The view can be selected below the menu bar

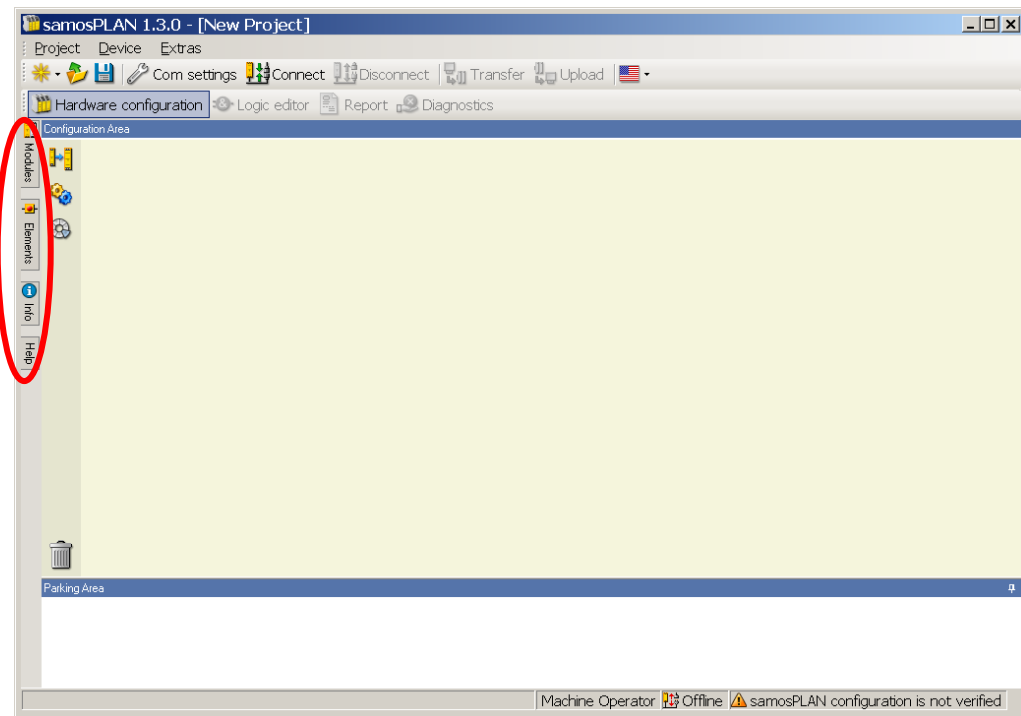
- The structure of a samos®PRO system consisting of various hardware modules as well as the configuration of the inputs and outputs and the connected elements are specified in the **Hardware configuration** view.
- The function logic can be configured by means of logic function blocks and application-specific function blocks in the **Logic editor** view. This view is not available unless a controller module has been selected beforehand in the hardware configuration.
- If the project contains at least one gateway or if RS-232 communication is enabled, the **Gateways** view is available. Here you can configure the gateways and the data that are transferred to and from the network.
- Complete information on the currently loaded project and all settings including the logic programming and wiring diagrams is available in the **Report** view. Furthermore, additional information on the project can be entered here. All information can be saved in standard file formats and printed out. The scope of the report can be compiled individually depending on the selection.
- The stored error messages are displayed as a history of a connected samos®PRO system in the **Diagnostics** view.

5.4 Positioning windows

Every view consists of several sub-windows that can be positioned freely. You can

- ➔ change the height, width and position of each sub-window by using the mouse to move the frame or title bar of the sub-window,
- ➔ convert a sub-window into a flyout window by clicking the “Hide” button (drawing pin symbol) on the right in the title bar. The flyout is then positioned on the left-hand margin of the samos®PLAN window,
- ➔ move flyout windows back to their normal position by clicking the drawing pin icon in the flyout window again.

Fig. 3: Sub-windows can be converted to flyout menus



5.5 “Hardware configuration” view

The **Hardware configuration** window consists of the following sub-windows:

- Tabs for switching between the **Hardware configuration**, **Logic editor**, **Report** and **Diagnostics** view.
- Menu bar with the menus **Project**, **Device**, **Extras**
- Toolbar with icons for rapid access to menus that are often used
- **Elements** selection window: All devices (e.g. sensors, actuators etc.) that can be connected to a samos®PRO safety controller are listed here. The devices can be parameterized and renamed. In addition, user-defined devices can be created and stored. In addition to the elements, EFI elements can also be connected. They are dragged to the two EFI interfaces of the controller module, provided that the controller module (e.g. SP-SCON-NET) provides EFI interfaces.
- **Parking area**: The user can compile a selection of devices for a concrete application and store them temporarily here.
- **Modules** selection window: All samos®PRO hardware modules that can be combined into a samos®PRO safety controller are listed here. The modules that cannot be selected at the current configuration are greyed out. Modules that can be added to the current configuration are identified by a green “+” symbol. The number of inputs, outputs and EFI connections is displayed for each module.

The Graphical User Interface

From a drop down list under the module, the STEP (or function package) for the respective module can be selected. The function package chosen defines the minimum firmware version that must be used: STEP 2.XX requires at least firmware version 2.00. See also chapter 3 "Version, Compatibility and Features" on page 11.

- **Configuration area:** The entire hardware configuration of the samos[®]PRO safety controller and of the connected devices is created here and represented graphically. The individual modules and connected devices can be named, have a tag name assigned and can be parametrized using the context menu of the devices. Additionally, it is possible to export or import a configuration (hardware configuration and logic) and – if the samos[®]PLAN is connected to the system – to change or reset the password or to perform a software reset of the system via the context menu of the controller module.
- Icons for the following functions are located on the left next to the positioned modules. From top: **Switch view**, **Settings** and **Edit tag names**. When a connection to a samos[®]PRO station is established, further functions are also available: **Log in** (change the user group), **Verify** (read in and compare the configuration) and **Run** or **Stop** the controller module.

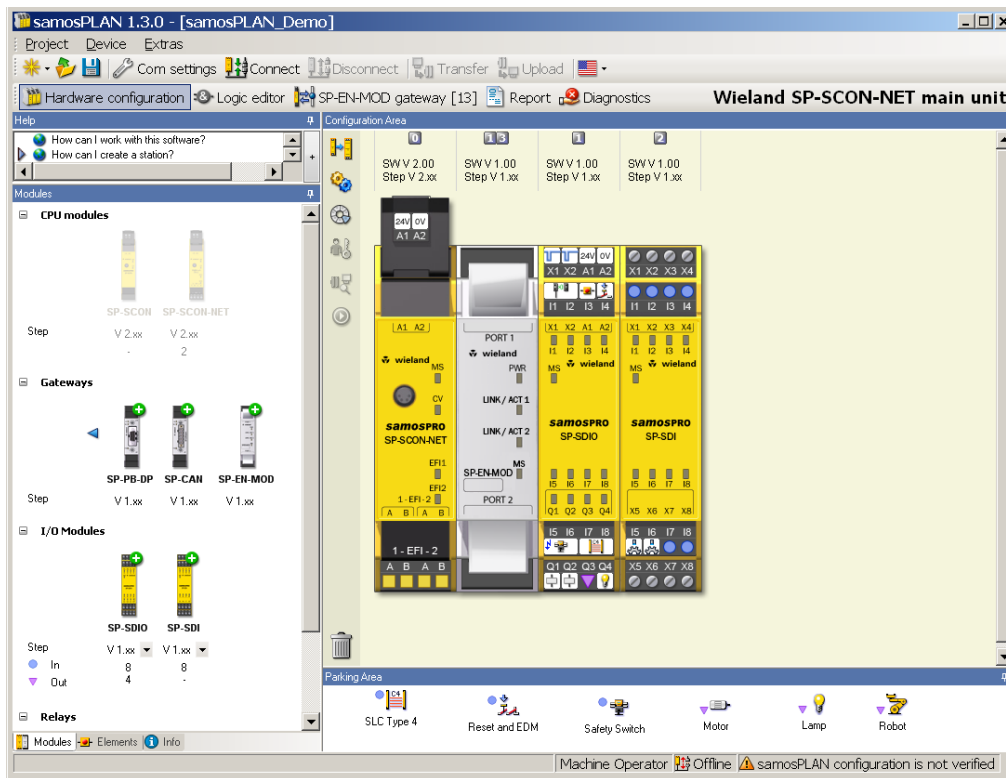


Fig. 4: The "Hardware configuration" view

- A double click on the controller module in the configuration area will open the logic editor.
- A double click on any gateway in the configuration area will open the gateway configuration view for the respective gateway.

NOTES

Switch view



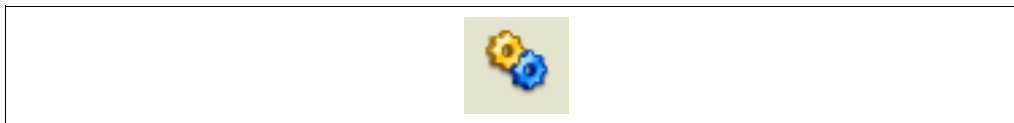
Fig. 5: Switch view button

The **Switch view** button toggles between an enlarged and a reduced view of the configuration area.

The Graphical User Interface

Settings

Fig. 6: Settings button



The **Settings** button opens a dialog where you can adjust the settings for your project. Here you can ...

- Create your own tag name format;
- Enable or disable customized elements (see section 5.5.6 “Customized elements” on page 24);
- Enable or disable the import of customized elements;
- Enable or disable RS-232 routing for the CPU;
- Save the current view and/or activate a saved view.

Edit tag names

Fig. 7: Edit tag names button



The **Edit tag names** button opens the central Tag name editor (see section 5.6.9 “Tag name editor” on page 37).

Edit button

If you need to change the configuration while the samos®PLAN is connected to the CPU, you can use the **Edit** button in the upper right corner of the screen over the configuration area to switch into the edit mode. This way it is possible to edit the configuration without disconnecting from the system first.

Fig. 8: Edit button in the hardware view



5.5.1 Exercise for configuring the samos®PRO modules

Exercise

- ➔ Create a new standalone project using the **New Project** button. All available samos®PRO modules are displayed in the **Modules** selection window. All modules are greyed out with the exception of the CPUx controller modules.
- ➔ Select the function package from the dropdown list under the desired controller module (SP-SCON or SP-SCON-NET). Function package Step V 2.xx requires CPU firmware version 2.00 or higher (see chapter 3 “Version, Compatibility and Features” on page 11).
- ➔ Use the mouse to drag the controller module into the **Configuration area**. The controller module is displayed magnified there. The inputs/outputs and terminals are visible. The controller modules are now greyed out and the other modules (gateways, I/O modules) can be selected in the **Modules** selection window.
- ➔ Create further samos®PRO I/O modules in the **Configuration area**. Green arrows indicate where the new module will be positioned. Grey arrows indicate possible other positions. The controller module is always located at the left. Up to two gateways follow directly to the right of the controller module. Then the expansion modules follow. The relay modules have to be positioned at the far right.
- ➔ Right-click the individual modules and select **Edit...** in the pop-up context menu. Enter a new tag name (module name) for the respective module and close the window by clicking **OK**.
- ➔ Change the positions of the modules subsequently by using the mouse to drag them to a different position.
- ➔ Remove modules from the configuration area by right-clicking the module and clicking **Remove module...** in the pop-up menu. Alternatively, use the mouse to drag the module to the recycle bin icon at the bottom left of the **Configuration area**.

5.5.2 Module status bits in the Hardware configuration view

When the samos®PRO system is online (i.e. the samos®PLAN is connected to the system), you can display the status bits of each module and their current values.

- ➔ Right click on any module (controller module, gateway or extension module) and select **Edit...** in the context menu. If the system is online, the dialog window for the selected module opens with the additional Diagnostics file card where all available status bits for the selected module and their values are displayed.
- ➔ Click on the Refresh button to update the values of the module status bits.

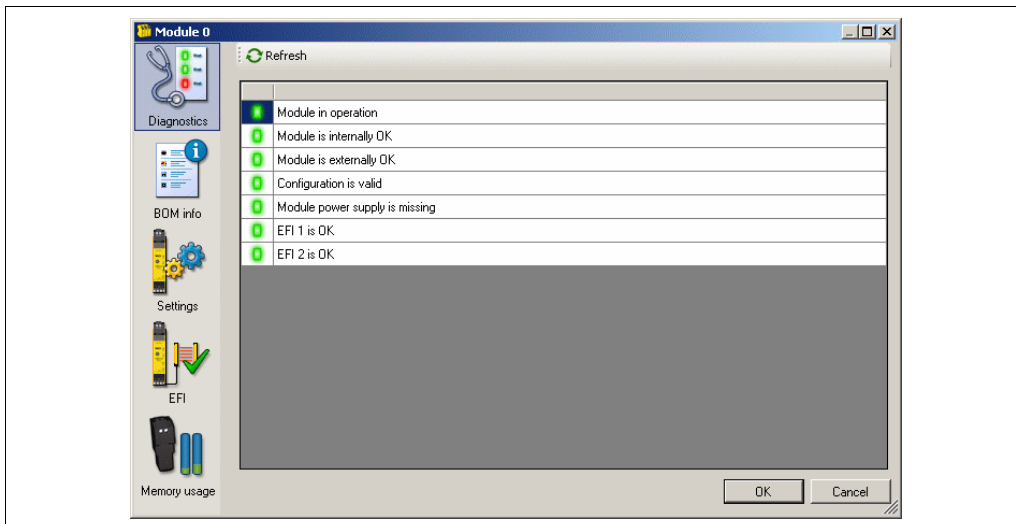


Fig. 9: CPU status bits in the Hardware configuration view

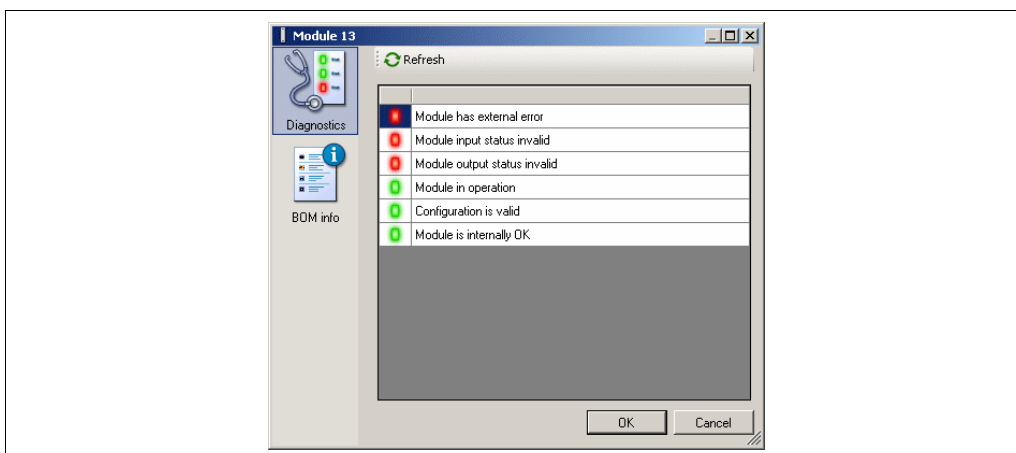


Fig. 10: Gateway status bits in the Hardware configuration view

The Graphical User Interface

Fig. 11: SP-SDIO module status bits in the Hardware configuration view

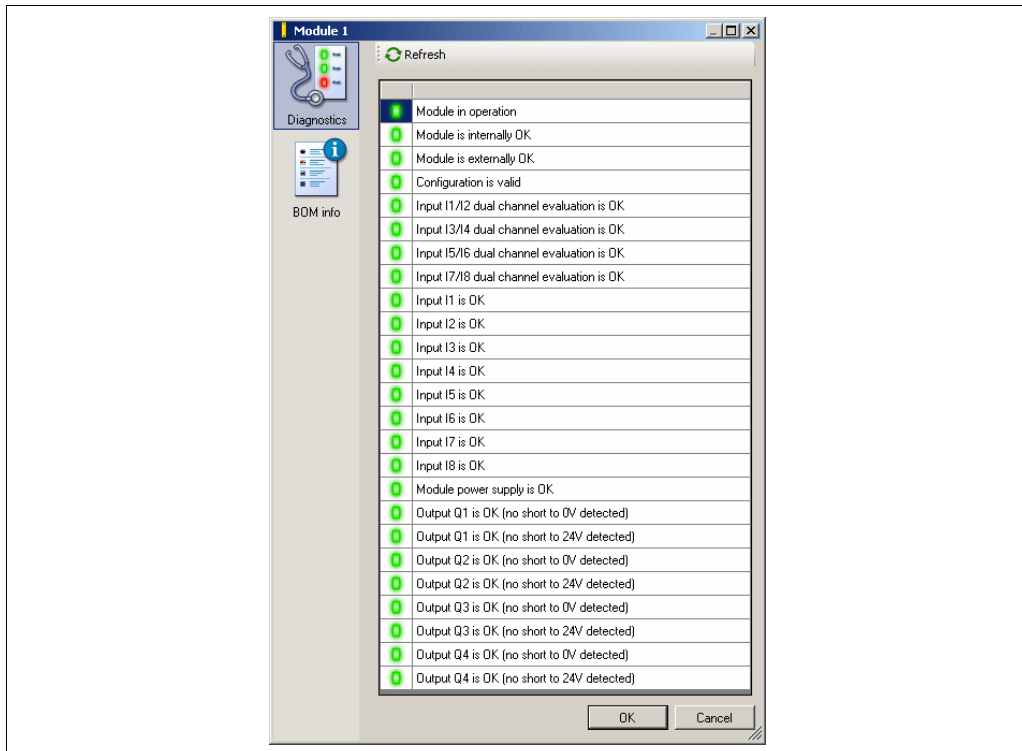
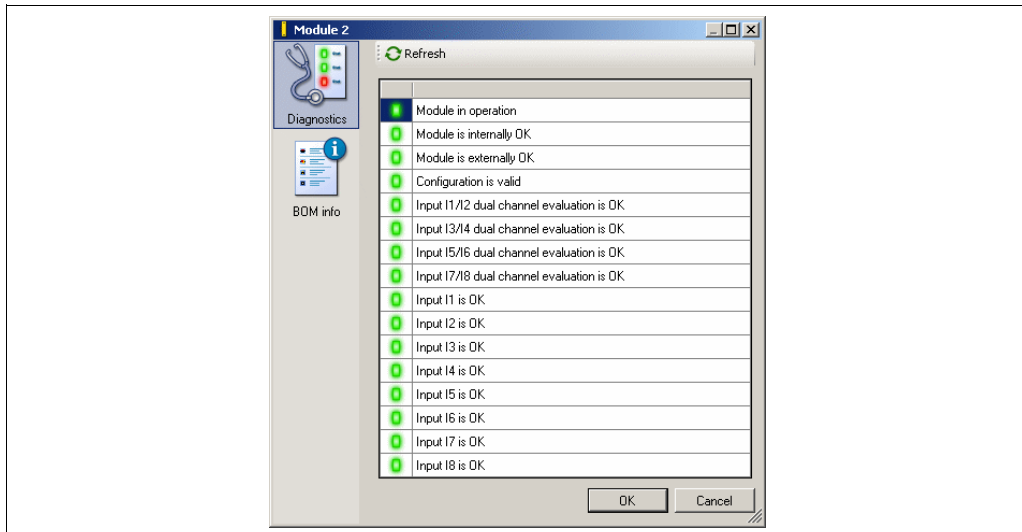


Fig. 12: SP-SDI module status bits in the Hardware configuration view



5.5.3 Exercise for configuring the connected devices

Exercise

- ➔ The selection tree in the **Elements** selection window can be expanded and collapsed by means of a mouse click. Optional: Right-click a device and select **Edit current element** in the pop-up menu. Assign a user-defined **Internal device number** if you want to. This **Internal device number** is stored for this device.
- ➔ Select some devices from the list and drag them into the **Parking area**.

NOTE

The **Parking area** serves to increase clarity. You can compile all required devices here so that you do not forget any of them during the configuration. Alternatively, you can drag the devices directly from the **Elements** selection window into the **Configuration area**.

- ➔ Then drag a device from the **Parking area** into the **Configuration area**.
- ➔ If the **Configuration area** does not contain a module with suitable free inputs/outputs, the device cannot be placed there. In this case, place at least one hardware module with inputs or outputs, e.g. SP-SDIO or SP-SDI, in the configuration area.

- ➔ When the device is moved over suitable free inputs or outputs, they light up green. The software automatically considers the required number of inputs or outputs. Drop the device on a suitable position. The device icon is now displayed in the view at this point.
- ➔ Drag the device to other suitable inputs or outputs or back into the **Parking area**.
- ➔ Delete the device by right-clicking the device icon and clicking **Delete** in the pop-up menu. Alternatively, use the mouse to drag the device to the recycle bin icon at the bottom left of the **Configuration area**.
- ➔ A device can be parametrized when it is located in the **Parking area** or in the **Configuration area**. Right-click a device in the **Parking area** or **Configuration area** and select **Edit...** from the pop-up menu or double-click a device. The **Element settings** window is opened. Depending on the type of device you can:
 - assign a tag name (identifying name for the element)
 - set evaluation parameters for the element, for example the discrepancy time, ON-OFF filter or OFF-ON filter, connection to a test output, test pulses enabled/disabled, etc.(See also section 5.5.5 “Parametrization of connected elements” on page 21.)
- ➔ Close the **Element settings** window by clicking **OK**.

5.5.4 Expanding elements

Some elements consist of a group of two or more sub-elements, such as an interlock that consists of a safety switch as input element and an interlock with locking as output element. Normally these elements must be connected to one module (e.g. SP-SDIO), but some of these elements can be expanded so that the individual sub-elements can be connected to different modules.

How to expand an element:

- ➔ Place the element (e.g. an interlock) in the Parking area.
- ➔ Right click the element to open the context menu.
- ➔ Select the **Expand** command. The element in the Parking area is replaced by its sub-elements which can be treated like individual elements.

5.5.5 Parametrization of connected elements

Input and output elements can be parametrized when they are located in the **Parking area** or in the **Configuration area**. Depending on the type of element you can:

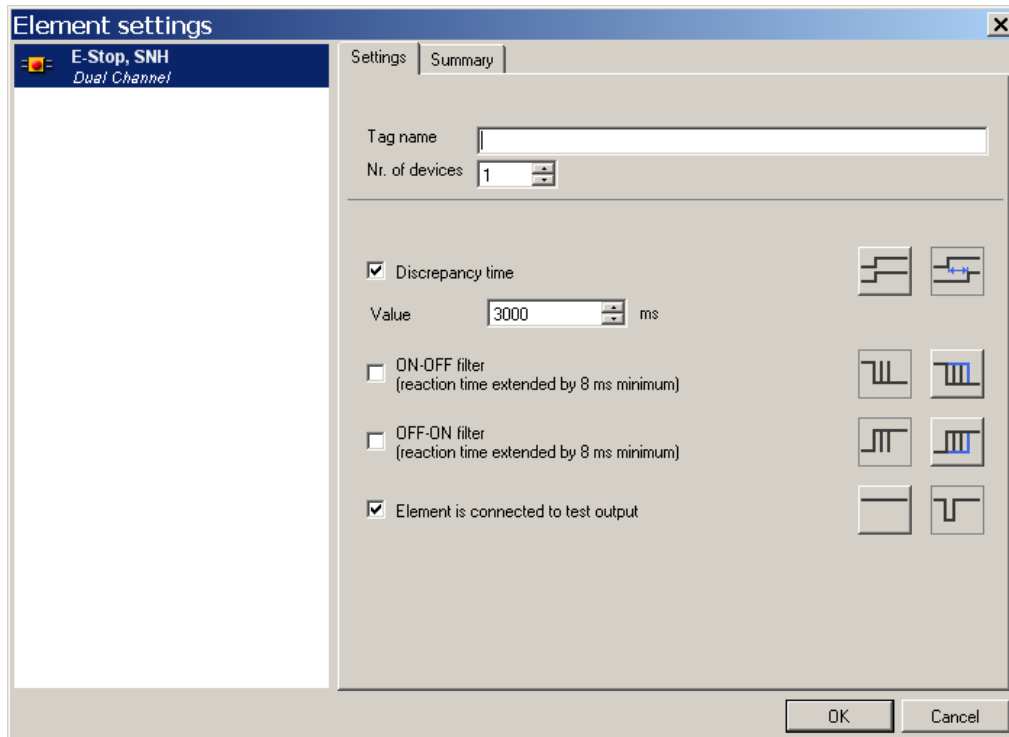
- assign a tag name (identifying name for the element)
- set evaluation parameters for the element, for example the discrepancy time, ON-OFF or OFF-ON filter, connection to a test output, test pulses enabled/disabled, etc.

The Graphical User Interface

How to parametrize a connected element:

- ➔ Double click on the element or right click an element in the **Parking area** or in the **Configuration area** and select **Edit...** from the pop-up menu. The **Element settings** window is opened.

Fig. 13: Element settings window for an SNH emergency stop button



Tag name

- ➔ Enter a **Tag name** for the element, if desired. Otherwise the default tag name is used.

Nr. of devices

- ➔ Adjust the **Nr. of devices**, if necessary. E.g. if you have connected a cascade of several testable type 2 sensors to one input, you can use this function to adjust the number of devices that will appear on the bill of material in the project report to match the actual number of devices used,

Discrepancy time

Dual channel elements can be evaluated with or without a **Discrepancy time**. The discrepancy time defines for how long the two inputs may have discrepant values after one of the both input signals has changed without this being considered as an error. For detailed information on discrepancy monitoring on the I/O modules see section 9.1 "Dual channel evaluation and discrepancy time" on page 193.

- ➔ To activate or to deactivate the **Discrepancy time** either click on the checkbox or on the 3D buttons on the right side of the element settings dialog.

For elements connected to SP-SDI and SP-SDIO modules the following restrictions apply:

- The **Value** for the discrepancy time can be set to 0 = disabled or to a value from 4 ms to 30 s. It will be rounded automatically to the next greater multiple of 4 ms due to the internal sampling frequency of the modules.
- If signals of tested sensors are connected to SP-SDI and SP-SDIO modules, the discrepancy time has to be greater than the *test gap* + the *max. Off-on delay* of the used test output. You can find these values in the project report under **Configuration, I/O module, Test pulse parameter**.
- If you try to set a lower discrepancy time than allowed, the minimum value will be shown in the dialog window.

ON-OFF filter and OFF-ON filter

Several unintentional brief signal changes occur when opening or closing a component fitted with contacts as the result of the bouncing of the contacts. As this may influence the evaluation of the input, you can use the **ON-OFF filter** for falling edges (i.e. transitions from High to Low) and the **OFF-ON filter** for rising edges (i.e. transitions from Low to High) to eliminate this effect.

- ➔ To activate or deactivate the **ON-OFF filter** or the **OFF-ON filter** either click on the checkbox or on the 3D buttons on the right side.

If the **ON-OFF filter** or the **OFF-ON filter** is active, a signal change will be recognized only if it is confirmed by three consecutive identical samples of the input with a sample rate of 4 ms, meaning 8–12 ms constant signal.

Consider extended reaction times when using the input filters!

- Due to the modules' internal sampling rate of 4 ms, the ON-OFF filter and the OFF-ON filter extend the reaction time by at least 8 ms.
- If the signal is alternating within these initial 8 ms, the signal change can be delayed for much longer, i.e. until a constant signal of at least 8 ms has been detected.



For dual-channel elements with complementary evaluation the respective filter (ON-OFF or OFF-ON) is always related to the leading channel. Filtering of the complementary channel is active automatically.

NOTE

Element is connected to test outputs

By activating or deactivating the option **Element is connected to test outputs** you can determine whether the respective element shall be tested or not. By connecting an element to the test outputs ...

- short circuits to 24 V in the sensor wiring which could inhibit the switch-off condition can be detected,
 - electronic sensors with test inputs can be tested.
- ➔ To activate or deactivate the connection to the test outputs either click on the checkbox or on the 3D buttons on the right side.

One SP-SDI has two test sources only, even if it has eight test output terminals.

NOTE

Protect single channel inputs against short circuits and cross circuits!

If a stuck-at-high error occurs on a single channel input with test pulses that was previously Low, the logic may see a pulse for this signal. The stuck-at-high first causes the signal to become High and then after the error detection time back to Low again. Due to the error detection a pulse may be generated. Therefore single channel signals with test pulses need special attention:

- If the stuck-at-high occurs on a single channel signal input with test pulses that was previously High, the logic will see a delayed falling edge (High to Low transition).
- If a single channel input is used and an unexpected pulse or a delayed falling edge (High to Low) at this input may lead to a dangerous situation, the following measures have to be taken:
 - Protected cabling of the related signal (to exclude cross circuits to other signals)
 - No cross circuit detection, i.e. no connection to test output.

This needs especially to be considered for the following inputs:

- Reset input on the Reset function block
- Restart input on the Restart function block
- Restart input on the Press function blocks (Eccentric Press Contact, Universal Press Contact, N-break, Press Setup, Press Single Stroke, Press Automatic)
- Override input on a Muting function block
- Reset input on a Valve function block
- Reset input and Reload input on a Counter function block



The Graphical User Interface

Disabling the test pulses of SP-SDIO outputs

It is possible to disable the test pulses on one or several outputs of SP-SDIO modules with firmware version V2.00 and higher.



Disabling the test pulses of any output reduces the safety parameters of all outputs!

Disabling the test pulses of one or more outputs of an SP-SDIO module will reduce the safety parameters for all outputs Q1...Q4 of this module. Consider this to ensure that your application conforms to an appropriate risk analysis and avoidance strategy!

For detailed information on the safety parameters see the samos®PRO Hardware operating instructions.

How to disable the test pulses of an SP-SDIO output:

- ➔ Connect an output element to the SP-SDIO module.
- ➔ Right click the output element and select **Edit...** from the context menu.
- ➔ Deactivate the option **Enable test pulses of this output**. The test pulses of this output are switched off. A notice will be displayed in the hardware configuration area under the respective SP-SDIO module.

5.5.6 Customized elements

In addition to the standard input and output elements that are installed with the samos®PLAN software, it is possible to create, configure, import and export customized elements. This function allows you to create element templates with preset configuration options (e.g. single-channel or dual-channel evaluation, discrepancy time, on-off filtering, connection to test outputs etc.) that are adapted to your specific equipment needs.

NOTES

The following constraints apply when modifying new elements:

- T_i (Test pulse): 2ms, 4ms, $n \times 4ms$ ($n = 1, 2, ..$)
- T_p (Test period): 40ms, 200ms, $n \times 200ms$ ($n = 1, 2,..$)
- $T_i \leq 0,5 \times T_p$

How to enable customized elements:

- ➔ In the **Hardware configuration** view, click on the **Settings** icon at the left of the **Configuration area** to open the **Settings** dialog.
- ➔ In the **General** tab, activate the **Enable customized elements** option.
- ➔ Click on **OK**.

How to create a customized element:

- ➔ In the **Hardware configuration** view, right click on any element (in the **Elements** window, in the **Configuration area** or in the **Parking area**). It is recommended to choose an element that is as similar as possible to the customized element you want to create.

- ➔ From the context menu select the command **Save as customized element...** The **Create custom element template** window opens.

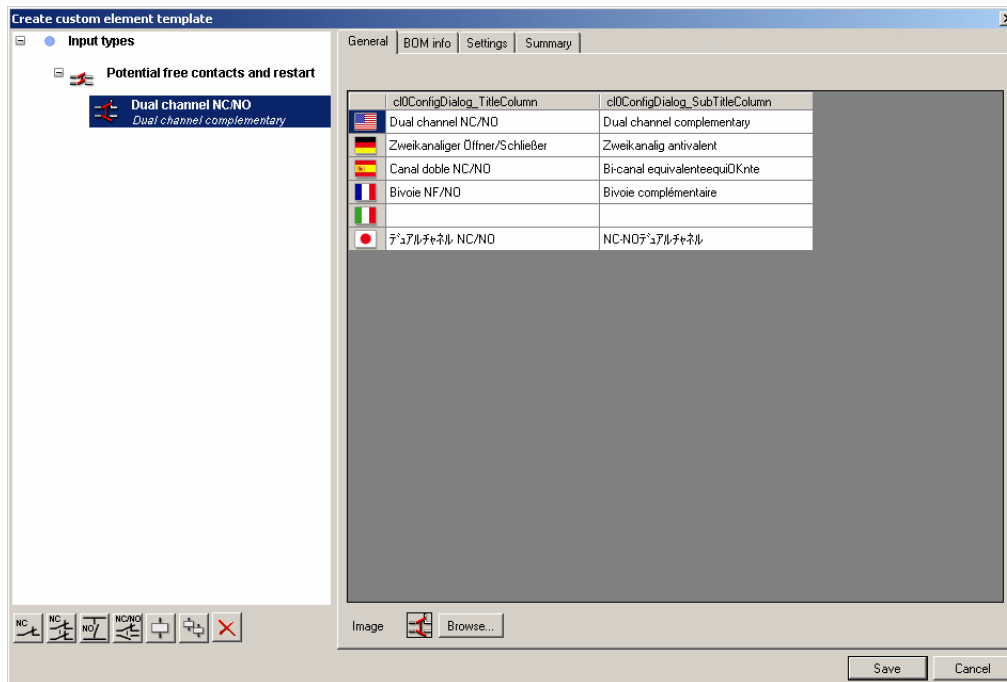


Fig. 14: Create custom element template window

- ➔ Rename and configure the element as needed (see below for details).
- ➔ Click on **Save** to save the new element and close the window.
- You have to enter a new name for the template in order to save it.
- Make sure that *all* settings are complete and correct before you save the new template. It is not possible to change an existing element template within the samos®PLAN, no matter whether it is a standard or a customized template.

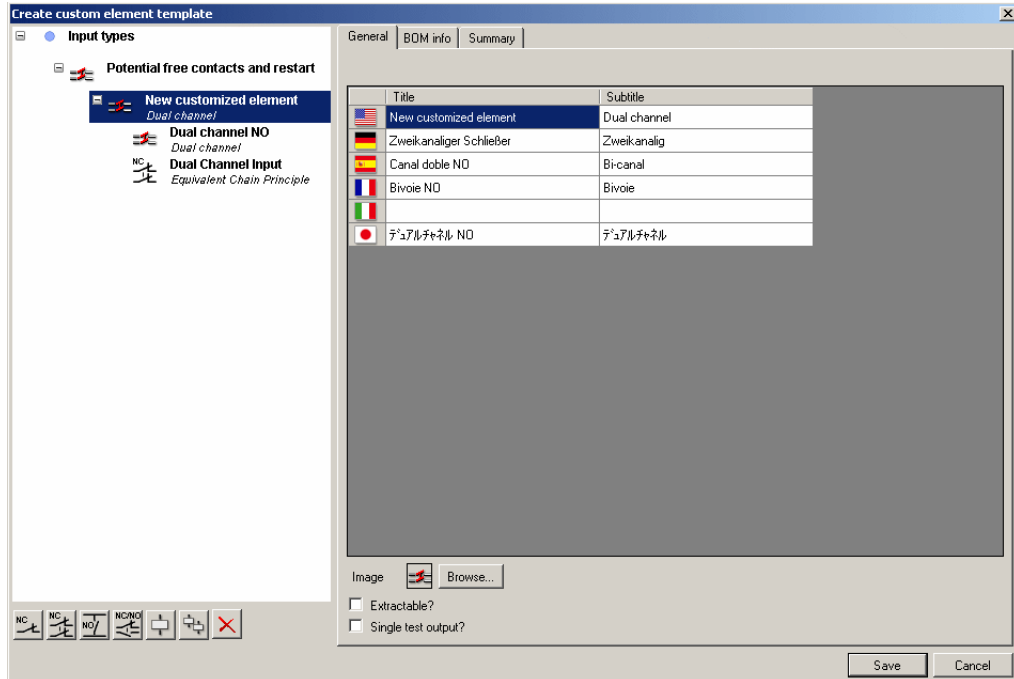
NOTES

The Graphical User Interface

How to configure a customized element:

- ➔ Select the new customized element in the element tree and use the sub-element buttons under the element tree to add additional inputs or outputs. You can choose between single-channel and various dual-channel input and output types. If you add sub-elements, these will appear in the element tree one level below the customized element.

Fig. 15: Add or delete sub-elements for a customized element



- ➔ Select the new element in the element tree and enter a new name for it in the **General** file card. It is not possible to save an element with a name that is already used by another element. However, it is not necessary to enter the new element name in all languages shown. You only have to change the element name in the language that is currently set in the samos®PLAN on your PC.
- ➔ Use the **Browse...** button at the bottom of the screen to assign a custom image to any element or sub-element.
- ➔ If an element contains two or more sub-elements, the **Extractable?** option is available. Elements based on a template that has been configured with this option active can be expanded or “split” into their sub-elements which can then be treated as individual elements (see section 5.5.4 “Expanding elements” on page 21).
- ➔ If the option **Single test output?** is activated, all sub-elements of the element must be connected to the same test output. Examples for this are the tested user mode switches which must use either inputs I1/I3/I5/I7 if test output X1 is used or inputs I2/I4/I6/I8 with test input X2.
- ➔ Enter the desired BOM (“bill of material”) information in the **BOM info** file card for the used elements and sub-elements. This information will be used in the samos®PLAN report in the material list.

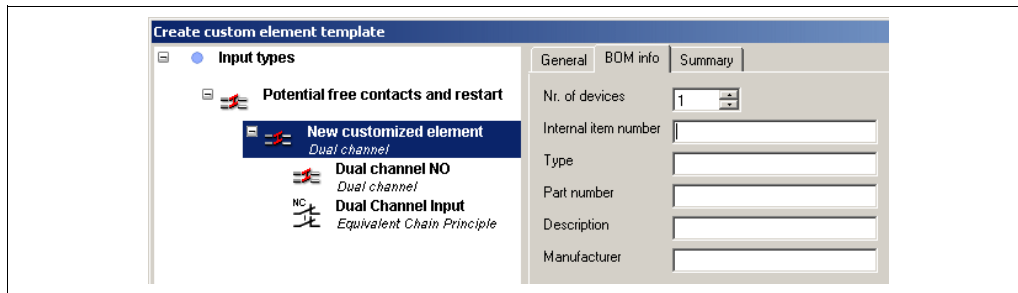


Fig. 16: Entering the BOM info for a customized element

- ➔ Select the customized element (or the sub-element) that you want to configure and click on the **Settings** file card to edit the configuration settings.

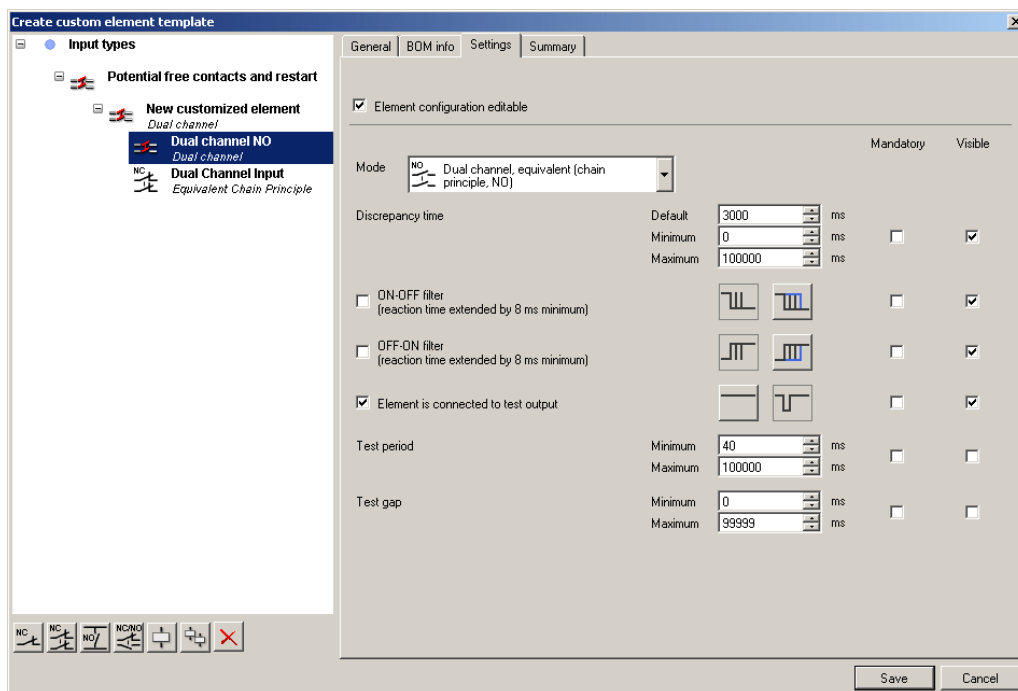


Fig. 17: Editing the configuration settings of a customized element

- ➔ Adjust the settings (e.g. Discrepancy time, ON-OFF filter, OFF-ON filter etc.) as described in section 5.5.4 “Expanding elements” on page 21. In addition to editing the configuration options, you can also activate or deactivate them completely and enter maximum, minimum and preset values.
- ➔ Check the **Mandatory** checkbox for a function, if elements based on this template must be used on modules supporting this function (e.g. to create an element that requires connection to a module with test outputs).
- ➔ Check the **Element configuration editable** checkbox, if Elements based on this new template shall be editable within the limits that are preset on the **Settings** file card.
- ➔ If only selected individual configuration options shall not be editable, uncheck the **Visible** checkbox for these options.

How to transfer a customized element to another PC:

- ➔ Save the project file and open it on the other PC. Customized elements contained in the project will be imported automatically.

Importing customized elements requires samos®PLAN version 1.3.0 or higher.

NOTE

How to delete a customized element:

- ➔ In the **Elements** window in the **Hardware configuration** view, right click on the customized element you want to delete.
- ➔ From the context menu select the command **Delete template....** You will be asked for confirmation.
- ➔ Click **OK**.

The Graphical User Interface

NOTE

It is not possible to delete a standard element template.

How to export customized elements as XML files:

- ➔ In the **Elements** window, right click on the customized element you want to export and choose the **Export...** command from the context menu. A folder selection dialog opens.
- ➔ Select or create the folder where you want to save the customized element and click on **OK**. The customized element is then saved as an XML file.

How to import customized elements as XML files:

- ➔ In the **Elements** window, right click on any element or element group and choose the **Import...** command from the context menu. A file selection dialog opens.
- ➔ Select the XML file for the customized element that you want to import and click on **OK**. The customized element will be imported.

5.5.7 Export and import of a configuration

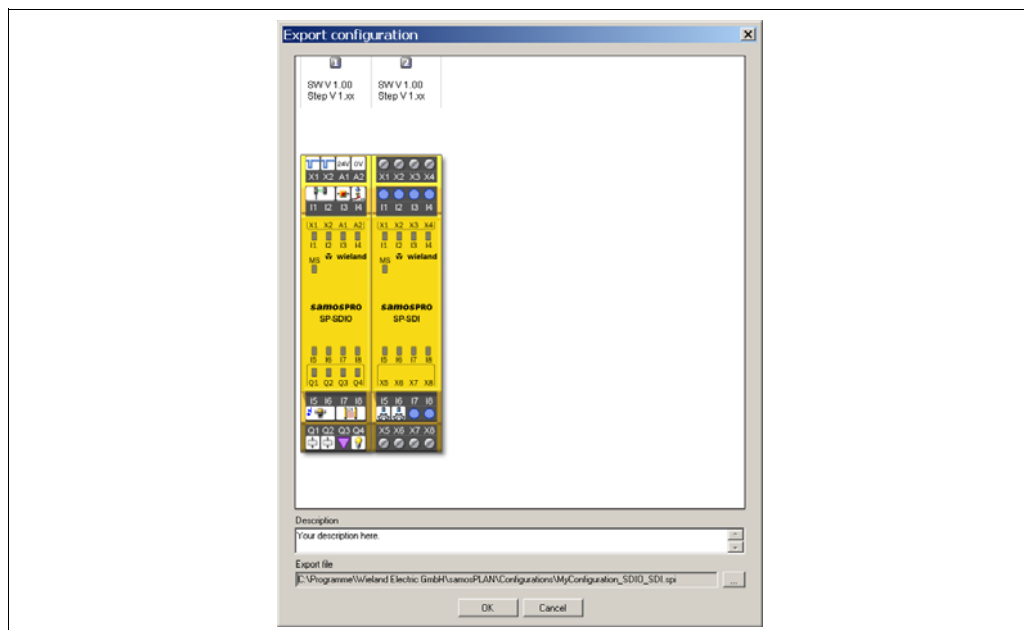
You can export or import a configuration. All modules with their associated inputs and outputs and logic are exported except for the SP-SCON/SP-SCON-NET. If you are exporting a project in which a SP-SCON-NET is present, the connected EFI compatible devices have to be reconfigured when you import the configuration into another project.

When you import a configuration into an existing project, the stored modules, elements and logic are added to the project while the rest of the project remains unchanged. This is especially useful if you need to replace a SP-SCON with a SP-SCON-NET (or vice versa) in an existing project without re-configuring the entire hardware and logic.

How to export a configuration:

- ➔ Right click on the controller module and select the **Export configuration...** command from the context menu. The following dialog opens.

Fig. 19: Export configuration dialog



- ➔ You can add a description of the configuration in the **Description** field.
- ➔ Click on the button on the right of the **Export file** field. A file selection dialog opens. Navigate to the folder where you want to save the export file, enter a file name for the export and click on **Save** to close the file selection dialog again.
- ➔ Click **OK** to save the exported configuration.

How to import a configuration

- Right click on the controller module and select the **Import configuration...** command from the context menu. The following dialog opens.

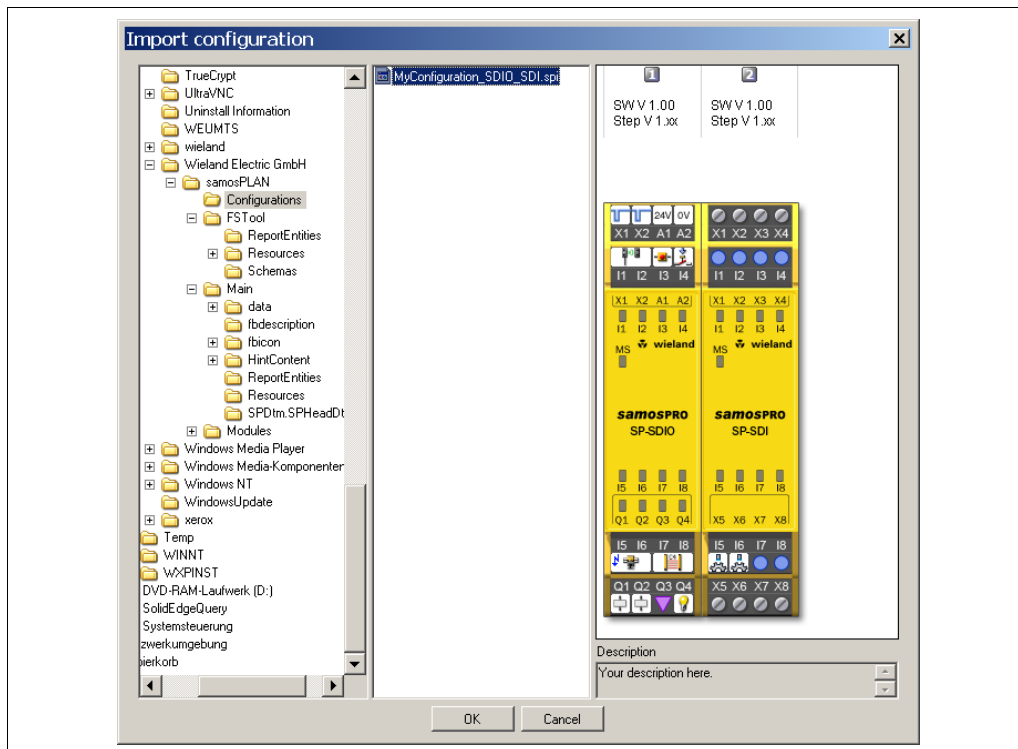


Fig. 20: Import configuration dialog

- Navigate to and select your project folder in the left panel. All samos®PRO import (*.ssi) files in the selected folder are displayed in the middle panel.
- Select any .ssi file to display the contained hardware configuration and description in the right panel.
- Click **OK** to import the selected import file. The hardware in the import file will be added to your projects hardware configuration while the imported logic will be inserted as one or several new separate pages in the logic editor.

Example: A project contains a SP-SCON-NET and an SP-SDIO module, a C4000, an emergency stop button, a robot and one page with the necessary logic in the logic editor. The configuration file to be imported contains another SP-SDIO module with a two hand control and a motor plus one logic editor page with the logic for controlling these devices. After the import has been completed, the project will contain *both* SP-SDIO modules with the respective devices connected and *both* logic programs on two separate pages.

How to exchange a controller module:

Using the export and import function, it is possible to exchange a controller module (e.g. SP-SCON to SP-SCON-NET or another firmware version) in an existing project without having to re-configure the project (hardware configuration, logic).

- Load the project with the controller module you want to exchange.
- Export the configuration as described above.
- In the **Project** menu, select the command **New, Standalone station project**.
- In the **Hardware configuration** view, add the desired new controller module to the new project.
- Right click on the new controller module and reimport the configuration.

The configuration of any connected EFI compatible devices is not included in the exported configuration. Therefore these devices have to be reconfigured.

NOTE

The Graphical User Interface

How to exchange an I/O module:

- ➔ Load the project with the I/O module you want to exchange.
- ➔ Add the desired new I/O module to the hardware configuration.
- ➔ Move the connected elements from the old module to the new module. This way the logic connections will be preserved.
- ➔ Delete the old module.

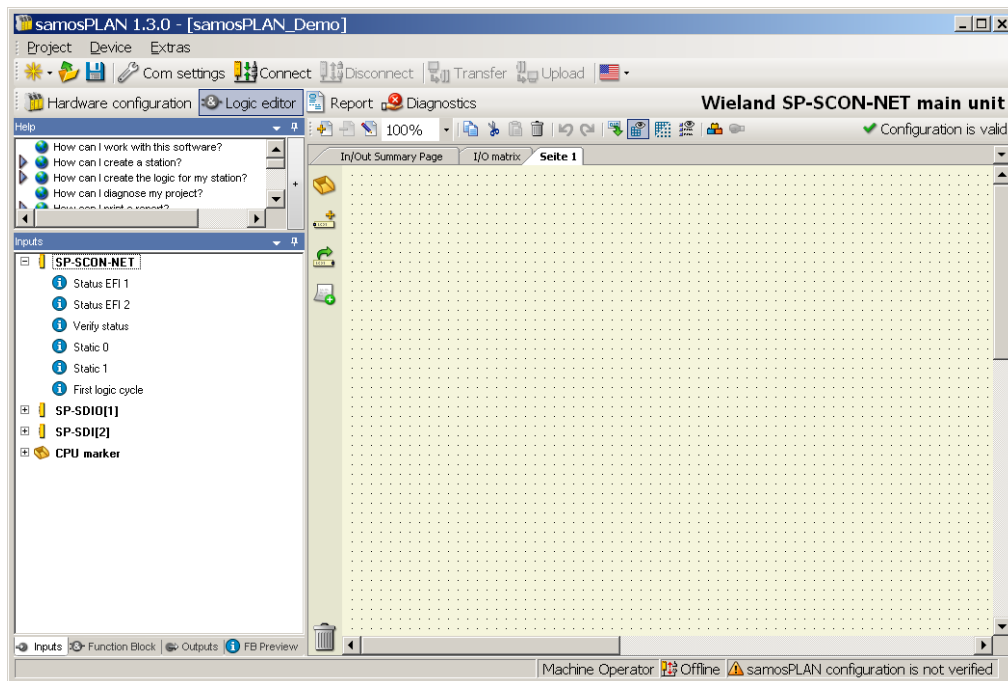
NOTE

This method does not work for elements that are used in combination with a Fast shut off function block, because these elements can not be moved to another module anymore.

5.6 Logic editor view

The samos®PLAN includes a graphical **Logic editor**. The function logic is program-med by using logic and application-specific function blocks. The inputs, function blocks and outputs are positioned on a worksheet and are connected correspondingly. As soon as a samos®PRO controller module is located in the Configuration area, the **Logic editor** can be accessed via the tab of the same name.

Fig. 21: The logic editor



The **Logic editor** window consists of the following sub-windows:

- Menu bar with the menus **Project**, **Device**, **Extras**
- Toolbar with icons for rapid access to menus that are often used
- Tabs for switching between the **Hardware configuration**, **Logic editor**, **Gateways** (if the project contains at least one gateway), **Report** and **Diagnostics** view.
- Specific toolbar for the logic editor with the following functions: **Add/Delete/Rename current page**, **Zoom**, **Copy/Cut/Paste/Delete elements**, **Undo/Redo last action**, **Open dialog to edit logic result markers**, **Show/Hide grid**, **Show grid of lines/dots**, **Show function block I/O description**, **Start simulation mode** and **Start forcing mode**
- Selection windows for **Function blocks**, **Inputs** and **Outputs** respectively
- **FB preview** window on the bottom left for displaying the important system resources such as the number of used/available function blocks or the current logic execution time (cycle time of the logic). When the cursor is moved over a function block in the worksheet, additional information on this function block is displayed in the **FB preview** window.
- Worksheets (**Pages**) for creating the logic, **In/Out summary page** and **I/O matrix** that can be selected alternatively by using tabs

5.6.1 Exercise for using the logic editor

Exercise

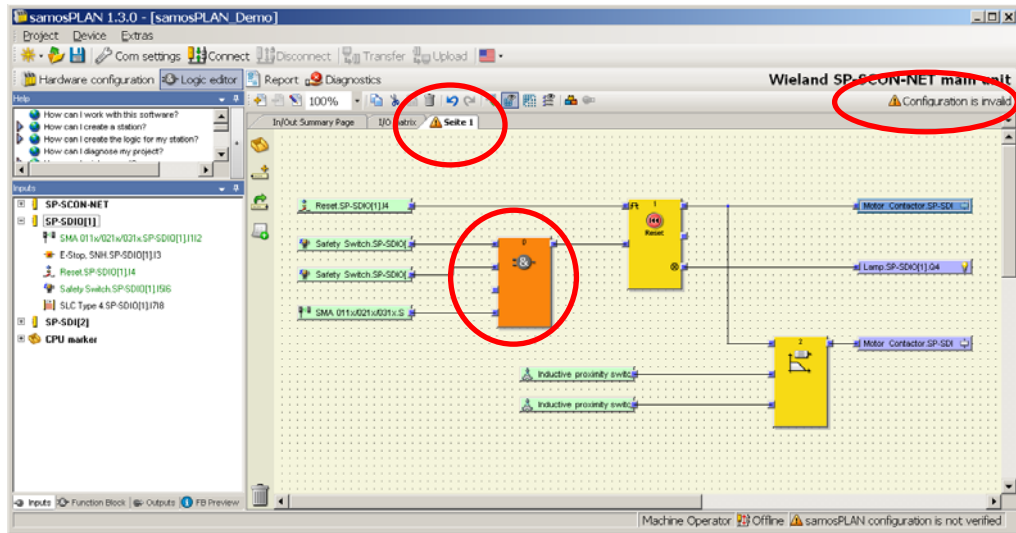
- In the **Hardware configuration** view combine a controller module, at least one SP-SDIO module and one element.
- Switch to the **Logic editor** by clicking the tab of the same name.
- In the selection window for **Inputs, Function blocks** and **Outputs**, click **Inputs** and drag an input from the list onto the worksheet. You can mark several inputs if you click them while holding down the [Ctrl] key or if you click the first and last in a series while holding down the [Shift] key. Then you can drag all marked inputs to the worksheet simultaneously.
- In the selection window for **Inputs, Function blocks** and **Outputs**, click **Function blocks** and drag an application-specific or logic function block from the list onto the worksheet. The function block will be displayed red as long as not all its inputs have been connected.
- In the selection window for **Inputs, Function blocks** and **Outputs** click **Outputs** and drag an output from the list onto the worksheet.
- Connect the node of the input with an input field of the function block (node) and an output (node) of the function block with the node of the output. To do so, click one node with the left mouse button, hold the left mouse button pressed and drag the cursor to the node with which the first node is to be connected. Once all inputs of the function block have been connected, the function block is displayed yellow.
- Alternatively it is possible to place and connect input or output elements in one step. Drag an input or output element while keeping the left mouse button pressed to the desired function block input or output node. If the mouse pointer hovers over the node it will be highlighted. Then drag the input or output to the place on the worksheet where it shall be positioned and release the left mouse button.
- While holding down the [Ctrl] key you can drag and drop the end of an already existing connection line from one node to another. This is useful to reassign a connection without having to delete it first.
- Mark the input, function block, output and the connections by clicking them or by dragging with the left mouse button pressed and then position as desired.
- In the selection window for **Inputs, Function blocks** and **Outputs**, click **FB preview**. A preview of the respective element or the details of a function block are displayed in the **FB preview** window when you move the cursor over it.
- In order to delete an element from the worksheet, right-click it and select the **Delete** command from the context menu.

The Graphical User Interface

5.6.2 Validation of the configuration

The samos®PLAN performs an automatic check of the logic program. If an error is detected, the configuration is marked as invalid and a warning icon appears in the upper right corner of the screen. Additionally, a warning icon marks the erroneous page of the logic program and the function block that is not connected correctly (e.g. one or more inputs are not connected) is displayed red.

Fig. 22: Invalid configuration warnings



As long as the configuration is invalid, it is not possible to start the simulation mode or to transfer the configuration to the samos®PRO system.

How to correct an invalid configuration:

- ➔ Connect all unconnected function block inputs. As soon as all function blocks are connected correctly, they will be displayed yellow and the invalid warnings will disappear.



Check your application thoroughly for correctness!

The samos®PLAN checks only for connection errors in your logic program. You are responsible to check whether your application conforms to your risk analysis and avoidance strategy and also fulfils all applicable standards and regulations. Otherwise the operator of the machine will be in danger.

5.6.3 CPU status bits in the logic editor

Verify status

This input is High if the configuration is verified (CV LED of the controller module is static yellow on).

Static 0 and Static 1

The **Static 0** input can be used to set a function block input permanently to 0 (Low). Respectively the **Static 1** input can be used to set a function block input permanently to 1 (High). This may be necessary to achieve a valid logic configuration if there are function block inputs that are not used but can not be disabled.

First logic cycle

This input is High for the very first logic cycle after every transition from the Stop state to the Run state. For all following logic cycles it remains Low. This input can be useful to trigger initialization functions in the logic program.

samos®NET status bits

NOTE

In a samos®NET system additional CPU status bits are available. For a description of these status bits see section 7.4.7 “samos®NET teaching status and diagnostics” on page 69.

Status EFI 1 and Status EFI 2

These inputs are available only with controller modules with an EFI interface, e.g. SP-SCON-NET. The inputs are set to Low as default.

The input is set to High if ...

- EFI input and output process data exchange to all EFI devices that are expected according to the configuration for the EFI interfaces (EFI 1, EFI 2) is started and faultless, or
- no EFI devices are expected according to the configuration.

The input is set to Low again if an error is detected at the EFI input/output process data exchange to any of the EFI devices (e.g. communication interruption).

With AOPD senders there is no EFI process data exchange. Therefore a communication error will not occur with these devices, i.e. a communication interruption can not be detected.

NOTE

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

An I/O error status bit for each connected EFI compatible device or samos®NET station is available in the **Inputs** tab of the logic editor under the respective EFI compatible device or samos®NET station and can be used as input for the logic programming. The I/O error status bit is High if the data or process image of the connected EFI compatible device or samos®NET station is set to Low. This may be the case e.g. if an error has been detected or if the samos®NET station is in the Stop state or being reconfigured.

Status bit	Value	Meaning
I/O error	Low	The corresponding EFI compatible device or samos®NET station is error-free (e.g. in the Run state)
	High	The process image of the corresponding EFI compatible device or samos®NET station is set to Low due to one of the following reasons: <ul style="list-style-type: none"> • Error detected on the EFI compatible device • The samos®NET station is not in the Run state • Suspended samos®NET station has been found • samos®NET station with different samos®NET ID found

Tab. 5: Meaning of the EFI I/O error status bits

See also section 7.4.7 “samos®NET teaching status and diagnostics” on page 69.

5.6.5 Module input and output status bits in the logic editor

The input and output status of the connected samos®PRO gateways and extension modules is available in the **Inputs** tab of the logic editor and can be used as input for the logic programming. In some applications an evaluation of this status information can be important in order to specify the behaviour of the logic functions of the samos®PRO safety controller. The input status specifies whether the data transferred from the input device to the samos®PRO controller module are:

- Low, because this is the output value at the input device or
- Low, because there is a fault at the input device.

Status bit	Value	Meaning
Status input data	Low	One or more input bits of the corresponding module are set to Low due to an detected error (e.g. cross-circuit detected or communication failure detected), meaning that the input bits may have different values as in error-free operation.
	High	The inputs of the corresponding module are error-free.
Status output	Low	For one or more outputs of the corresponding module an error

Tab. 6: Meaning of the module status bits

The Graphical User Interface

data		has been detected (e.g. overload detected, short circuit detected or communication failure detected), meaning that the outputs may have different values as in error-free operation.
	High	The outputs of the corresponding module are error-free.

NOTE

The input and output status for the SP-SDIO and SP-SDI modules is available only with firmware version V2.00 and higher.

5.6.6 CPU markers

CPU markers are available as inputs and outputs in the logic editor. They can be used e.g. for the creation of logic loop backs or to connect an output of a function block that is placed on one page of the logic editor to an input of a function block on another page of the logic editor.

A CPU marker consists of an output marker and an input marker. The input marker always takes the same value (High or Low) as the corresponding output marker with a delay of one logic execution time.

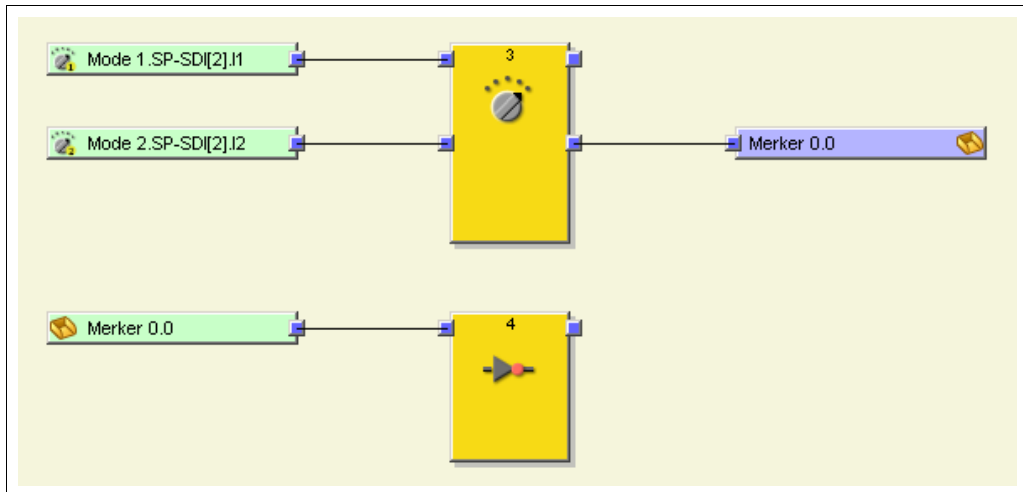


Take the delay caused by CPU markers into account!
 CPU markers always cause a delay of one logic execution time, because the input marker always uses the value of the output marker in the previous logic cycle. The resulting delay must be considered for the response time calculation and for the functionality.

How to use a CPU marker:

- ➔ Connect a CPU output marker (e.g. Marker 0.0) from the **Outputs** tab of the logic editor to the function block output that you want to use. Each CPU output marker can be used only once in a project.
- ➔ Connect the corresponding CPU input marker (e.g. Marker 0.0) from the **Inputs** tab of the logic editor to the function block input where you want to use the signal from the first function block as shown in the following screenshot. CPU input markers can be used several times in a project.

Fig. 23: Example for the usage of a CPU marker



5.6.7 Jump addresses

Jump addresses can be used basically in the same way as CPU markers. They consist of a source jump address and a destination jump address. The destination jump address takes the same value (High or Low) as the corresponding source jump address without delay – provided that it is not a loop back. In this way, jump addresses differ from CPU markers.



Take logic loop backs into account!
 A logic loop back is created if a function block input is connected to a destination jump address and the related source jump address is connected to an output of the same function block or to an output of another function block that has a higher function block index (the function block index is displayed at the top of each function block and shows the

function block's position in the logic execution sequence). In this case, the logic result from the current logic cycle is only available at the destination jump address in the following logic cycle, i.e. with a delay of the logic execution time. If a jump address causes a loop back, this is indicated automatically by an additional clock symbol shown on the destination jump address icon. The resulting delay is equal to the execution time and must be considered for the response time calculation and for the functionality.

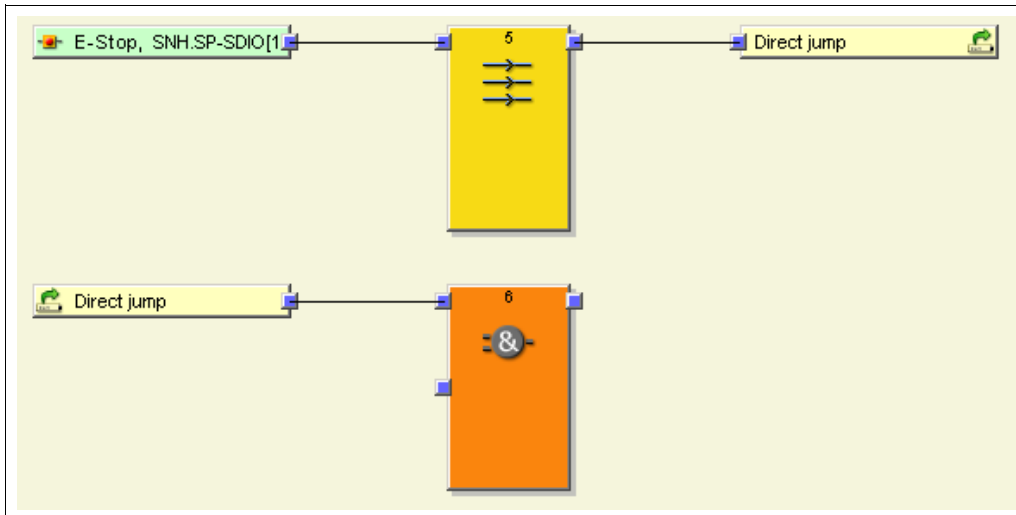


Fig. 24: Example of jump addresses with and without loop back

How to use a jump address:

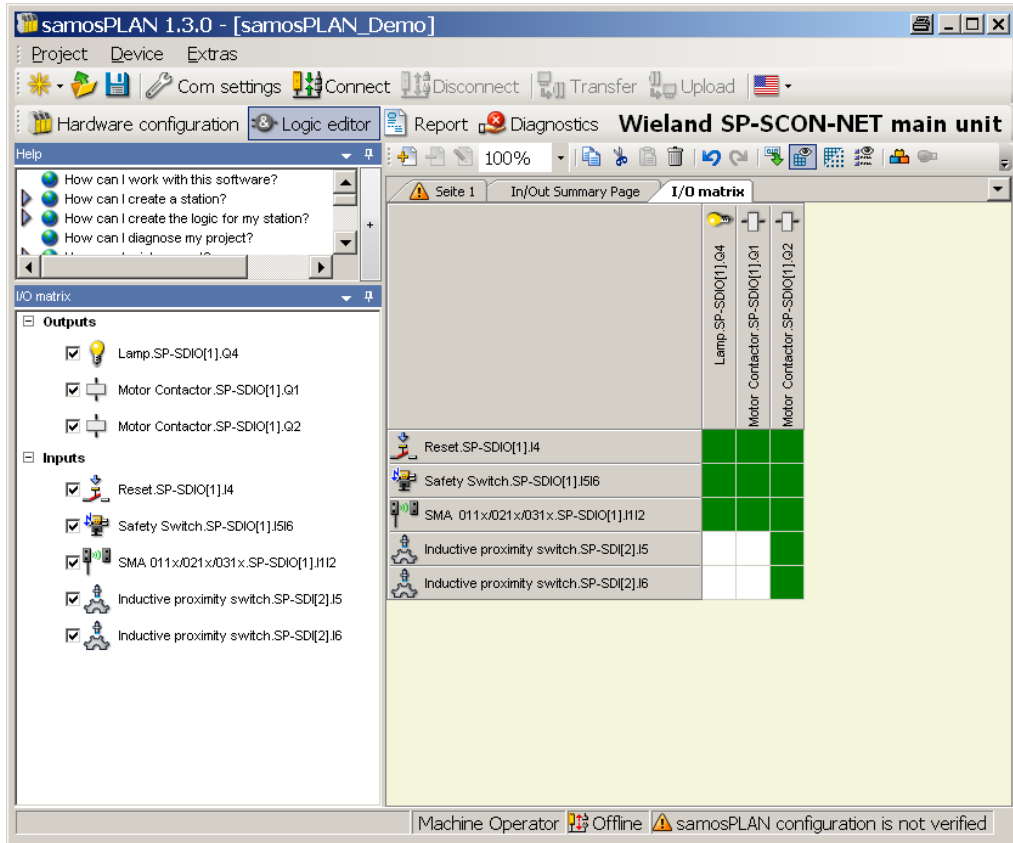
- First add a **source jump address** per drag & drop to your project. A dialog opens where you must enter a label for the new source jump address. Each source jump address label must be unique and can be used only once in a project. Typically, a source jump address is connected to any function block output.
- Then add one or several **destination jump addresses** per drag & drop. A dialog opens, where you can select the correspondent source jump address for the new destination jump address from the list of existing source jump address labels. A source jump address can have several destination jump addresses in a project. Typically, a destination jump address is connected to any function block input.

The Graphical User Interface

5.6.8 I/O matrix

The logic editor's **I/O matrix** file card displays which inputs have an effect on which outputs. This can be useful to check whether your logic program is complete. A green field indicates that the respective input has an effect on the respective output; a white field indicates that there is no relation between this input and output.

Fig. 25: I/O matrix in offline mode



In the **I/O matrix** window, all inputs and outputs are listed. By checking or unchecking the checkboxes you can select which inputs and outputs shall be displayed in the I/O matrix. This can be useful in complex projects with many inputs and outputs to condense the displayed information to the most important aspects.

I/O matrix in simulation mode

In simulation mode (see section 8.13 “Simulation of the configuration” on page 186), the I/O matrix displays the values of the used inputs and outputs. High inputs and outputs are displayed green. By clicking on an input you can toggle its value between High and Low and monitor the effect on the output values.

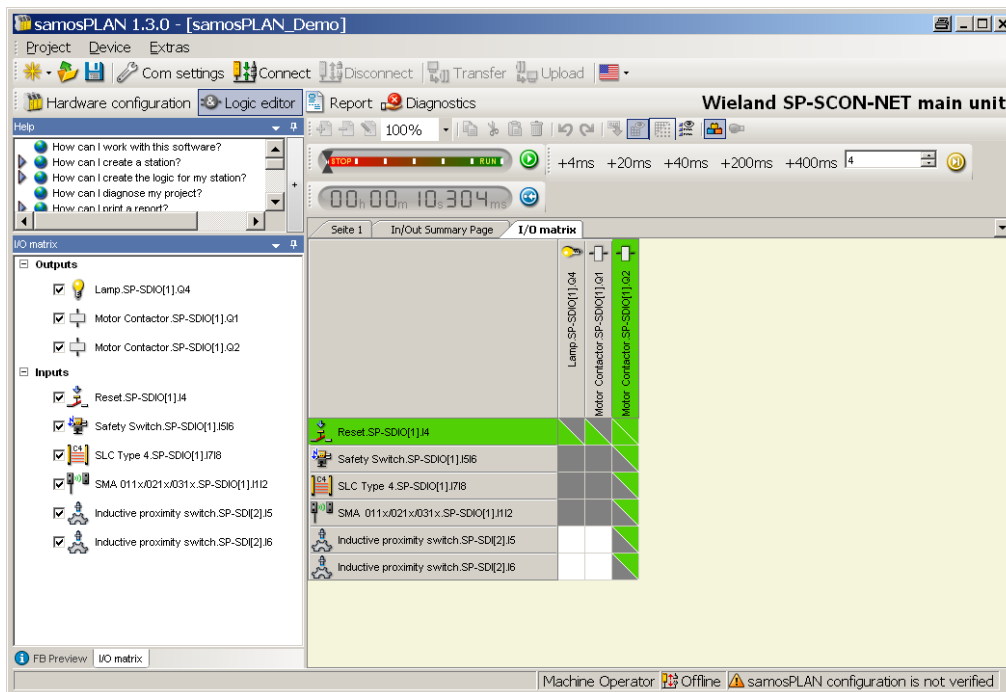


Fig. 26: I/O matrix in simulation mode

5.6.9 Tag name editor

The Tag name editor is the central place where you can edit all tag names in your project. To open the tag name editor either click on the **Edit tag names** button in the Hardware configuration view or click on the **Open dialog to edit logic result markers** button in the Logic editor toolbar.

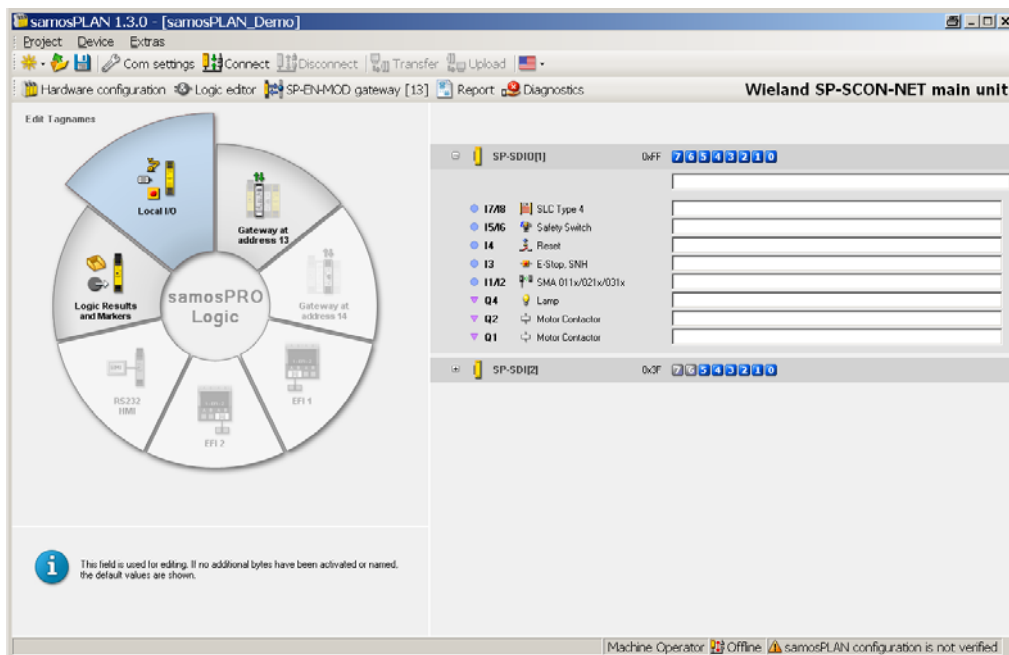


Fig. 27: Tag name editor

The Graphical User Interface

How to edit the tag names

- ➔ Click on one of the active sections on the left side to choose the device type whose tag names you want to edit:
 - **Logic results and markers:** Tag names in the logic editor (controller module)
 - **Local I/O:** Input and output element tag names in the hardware view (extension modules)
 - **Gateway at address 13/14:** Input and output data set tag names (gateways)
 - **EFI1/EFI2:** Input and output tag names for devices on EFI interface 1 and 2
 - **RS232 HMI:** RS-232 I/O tag names

The tag names of the selected type are listed in a tree view on the right side of the screen.

If a device type is not available in your project (e.g. no EFI compatible device connected), then the corresponding section is displayed grey, i.e. inactive.

- ➔ Navigate through the tree view on the right side to select the bit whose tag name you want to edit and enter the desired name in the input field.

5.7 Report view

In the **Report** view, a comprehensive report on the current project and all configuration settings including the logic program and detailed wiring information is available. You can individually configure the contents of the report.

Fig. 28: Report view



The information to be summarised in the report can be selected individually from an expandable selection list on the left-hand side. The selection is made by activating or deactivating the check boxes.

The toolbar in the **Report** standard contains the following commands:

- **Save:** Stores the report in .pdf format on a data medium.
- **Print:** Open the report in .pdf format. To this purpose, a PDF viewer (e.g. Acrobat Reader) must be installed on your computer.
- **Refresh report:** Updates the report after changing the report structure
- **Change report structure:** Switches between a hardware oriented and a function oriented report structure.

NOTE

Detailed information on using the wiring information at the end of the report is available in the samos®PRO hardware operating instructions. You will find example application reports in the annex of these operating instructions (see section 14.1 “Example application report” on page 205).

Exercise for the Report view

- Open the Report view by clicking on the **Report** button.
- Use the **Change report structure** button to switch between the two different report structures (hardware- or function-oriented).
- Activate or deactivate the check boxes for the components that shall be included in the report in the selection list on the left-hand side.
- After you have completed your selection, click **Refresh report**. The report is now assembled and displayed in the right-hand window section.

How to save or print a report:

The report can be printed or saved as PDF.

- To save the report as PDF, click on the **Save** button.
- To print the report, click on the **Print** button. A PDF preview of the report will be created that you can subsequently print.

5.8 Diagnostics view

Once you have completed your project and connected to your samos®PRO system, you can perform a diagnostics on your system. In the **Diagnostics** view, a complete history of all messages, information, warnings and error messages of a connected samos®PRO safety controller is available in the upper part of the window. If you click on one of the entries in the list, details on the selected message are displayed in the lower part of the window.

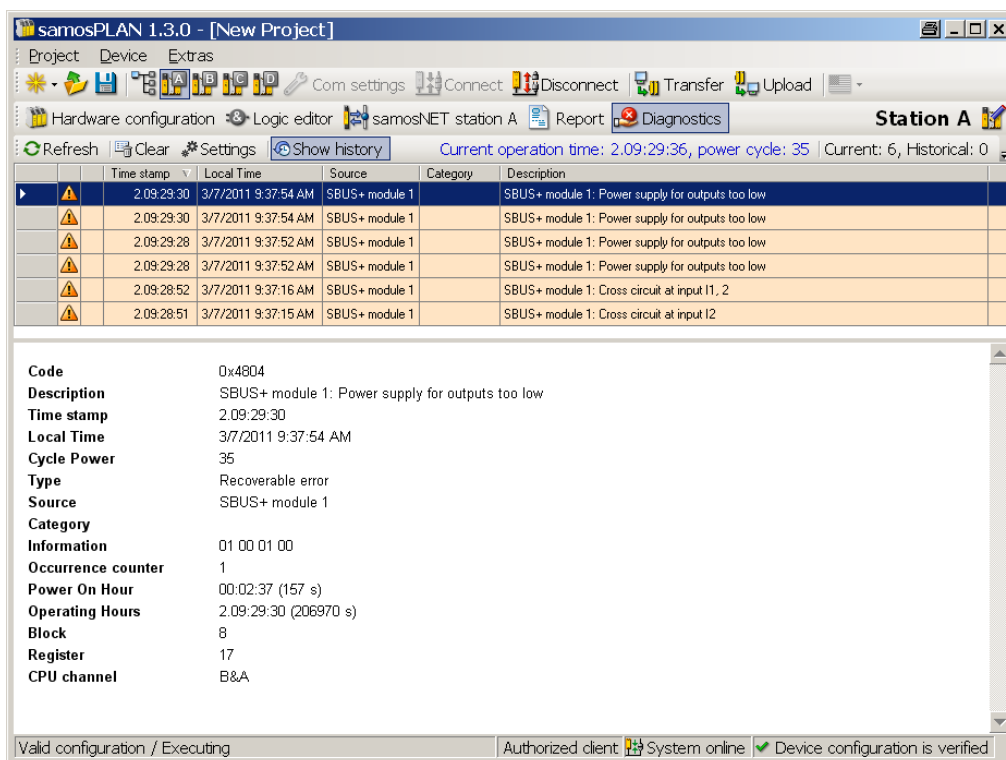


Fig. 29: Diagnostics view

Keyword	Description
Code	Hexadecimal error code.
Description	Error description.

Tab. 7: Meaning of the diagnostics information

The Graphical User Interface

Keyword	Description
Time stamp	Total controller module operation time at the time of error (days:hours:min:sec)
Local time	Time when error occurred (PC system time). This value is not displayed for historical errors.
Cycle power	Total number of times the controller module has been switched on.
Type	Error type (e.g. information, warning, recoverable error, critical error).
Source	Module that detected the error.
Category	Part of the module 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 a row, only the last occurrence will be recorded and the occurrence counter is increased.
Power on hour	Total time since the last power-on of the controller module. This value is reset at every restart.
Operating hours	Total power-on time of the controller module.
Block	Diagnostics memory area in the controller module. 8 = RAM (volatile, error occurred within the current power-on cycle) 88 = EEPROM (non-volatile, error occurred in a previous power-on cycle)
Register	Index in the diagnostics memory area.
CPU channel	Internal hardware channel (A or B) of the module that detected the error.

NOTE

For a list of the most important error codes, possible causes and potential rectification measures please see the “samos®PRO Hardware” operating instructions.

How to perform diagnostics:

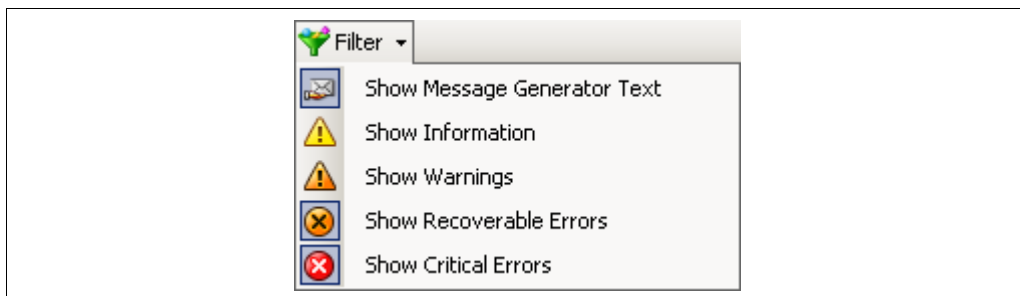
- ➔ Click on the **Diagnostics** button in the menu bar to open the **Diagnostics** view. In the toolbar, the following commands are available:

Fig. 30: Toolbar in the diagnostics view



- ➔ Click on **Refresh** to read the current message list from the system.
- ➔ Using the **Clear** button you can delete all messages stored in the system. You must be logged in as *Authorized client*.
- ➔ Under **Settings** you can configure an automatic refresh of the diagnostics and the time interval. In the **Diagnostic Settings** dialog, activate the **Automatic Refresh** checkbox and enter the desired refresh interval in seconds.
- ➔ Using the **Show history** button, you can display or hide older messages still stored in the samos®PRO system.
- ➔ The **Filter** pull-down menu enables you to display or hide different types of messages. In the menu, click on the different message types to activate or deactivate them.

Fig. 31: Filtering the diagnostics messages



NOTE

To save or print the diagnostic messages you can use the Report function (see section 5.7 “Report view” on page 38).

6 Connecting to the samos®PRO system

6.1 First steps for establishing a connection

This chapter describes how to establish a connection between the samos®PRO system and a PC or notebook.

6.1.1 Connecting the PC to the samos®PRO system via RS-232

- ➔ Connect a PC or notebook to the RS-232 interface of the SP-SCON(-NET).
- ➔ Power on the samos®PRO system.
- ➔ Open the samos®PLAN configuration tool installed on the PC.
- ➔ Click on **Edit com. interface settings** to ensure the correct communication interface has been selected. The following dialog appears:

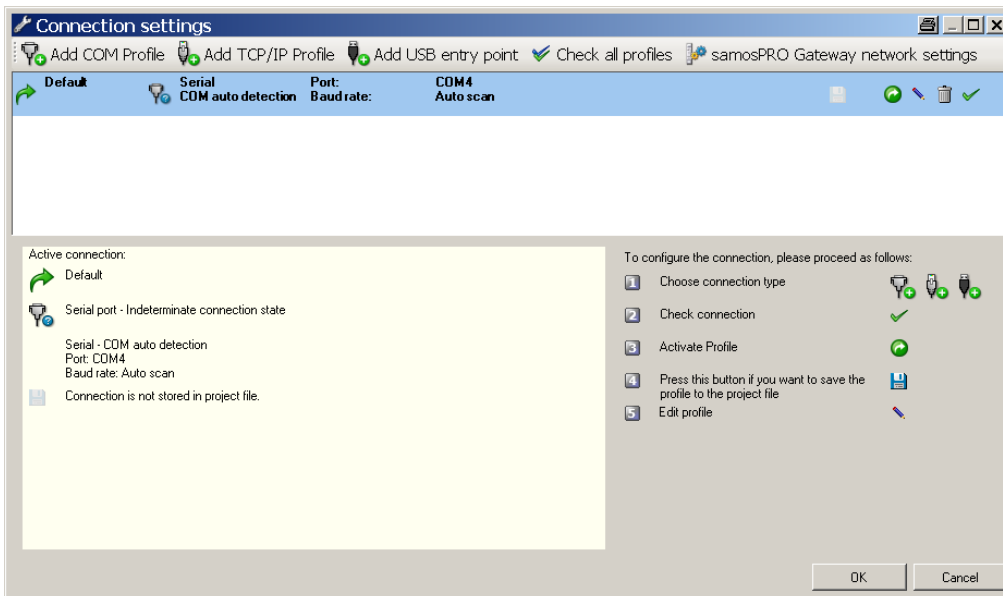


Fig. 32: Connection settings dialog

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

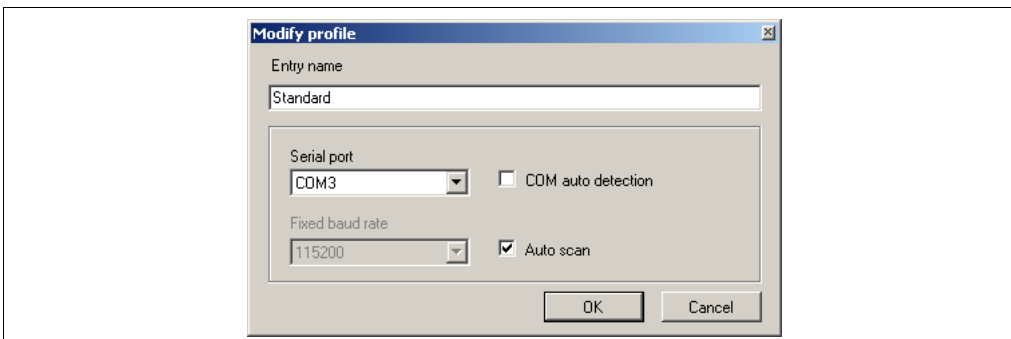


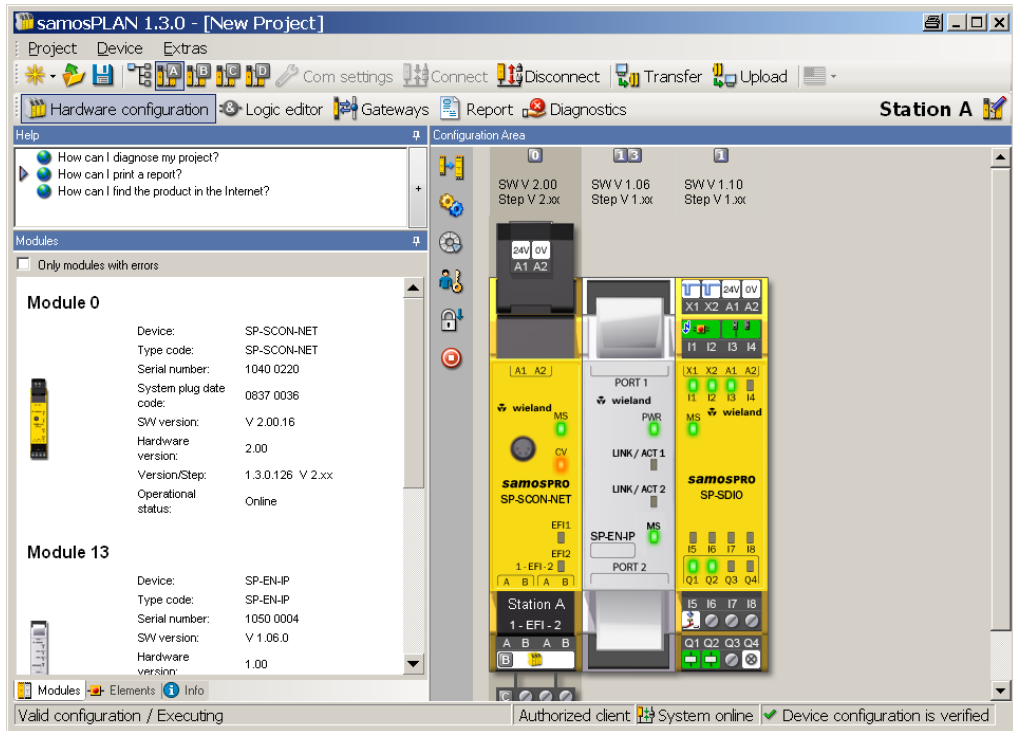
Fig. 33: Modify profile dialog

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

Connecting to the samos®PRO system

As an example, the following hardware configuration may appear:

Fig. 34: Hardware configuration dialog (example)



- ➔ Click **Disconnect** to go into the offline mode if you want to change the configuration of the samos®PRO modules. Alternatively, you can click on the online edit field to make minor changes without having to disconnect and connect each time.

NOTE

Configuration and verification of devices that are connected to the samos®PRO system is generally not carried out using the samos®PLAN software, even if they can be addressed via the RS-232 interface of a samos®PRO controller module. These devices have their own mechanisms for configuration and verification. An exception are EFI compatible devices connected to the samos®PRO controller module SP-SCON-NET (EFI elements from the elements window). These devices can be configured directly in the samos®PLAN by double-clicking the icon.

6.1.2 Online status and background color

The background color displayed in the samos®PLAN indicates the current online or offline status of the samos®PRO system as shown in the following table:

Background color	Online status	Configuration status
Light yellow	Offline	Any
Blue	Online	Invalid and/or different to the device configuration
Grey	Online	Valid and equal to the device configuration

Tab. 8: Meaning of the background color

6.2 Editing the communication settings

Using the **COM settings** command, you can create, edit and delete connection profiles. To edit the connection profiles, the software must be in offline mode.

- ➔ If you are in online mode, click on the **Disconnect** button to change into offline mode.
- ➔ Click on **COM settings**. The dialog for editing the connection profiles is opened:

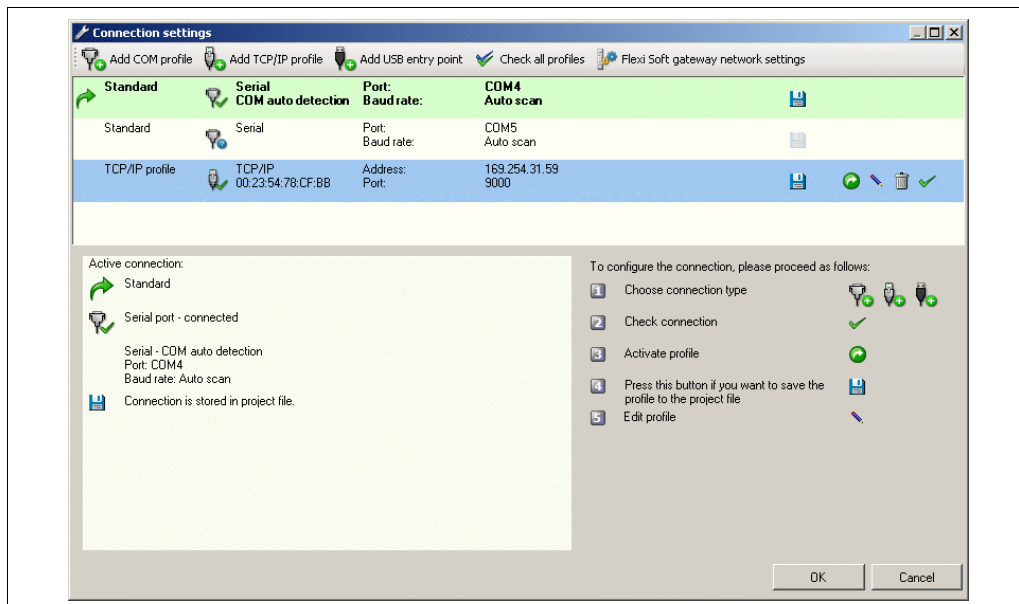


Fig. 35: Connection settings dialog

All existing connection profiles are displayed here. The currently activated profile is marked light green and with bold typeface; the profile selected for editing is marked blue. At the bottom of the dialog an overview of the current settings is displayed along with some options that you have.

The symbols for editing the profiles have the following meaning:

Symbol	Meaning
	Save profile with the current project
	Activate profile
	Edit profile
	Remove profile
	Check connection

Tab. 9: Symbols for editing the connection profiles in the Connection settings dialog

How to add a COM profile (serial port):

- ➔ Click on the **Add COM profile** button. The **Create new profile** dialog is opened.

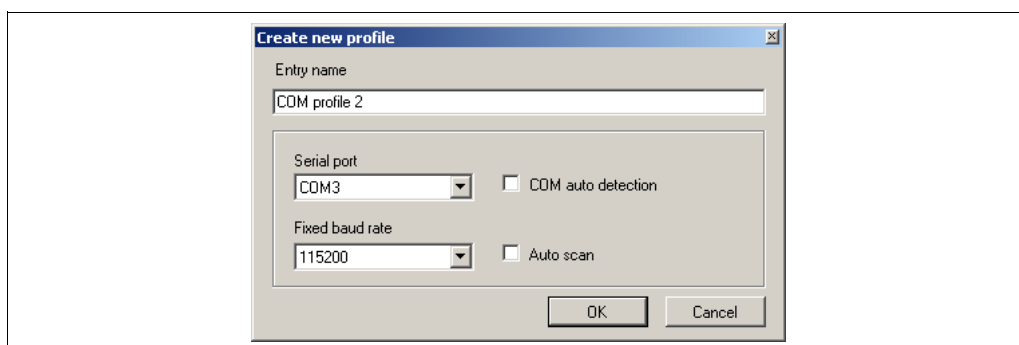


Fig. 36: Create new profile dialog (serial port)

- ➔ Enter a name for the new profile.
- ➔ Select the serial port for the new profile or activate the **COM auto detection** checkbox.
- ➔ Select a fixed baudrate or activate the **Auto scan** checkbox.

Connecting to the samos[®]PRO system

- ➔ Click **OK**. The dialog is closed and the new profile is displayed in the list.
- ➔ To activate the new profile, select it using the left mouse button and click on the green arrow symbol on the right. From now on, the samos[®]PLAN will use this profile.

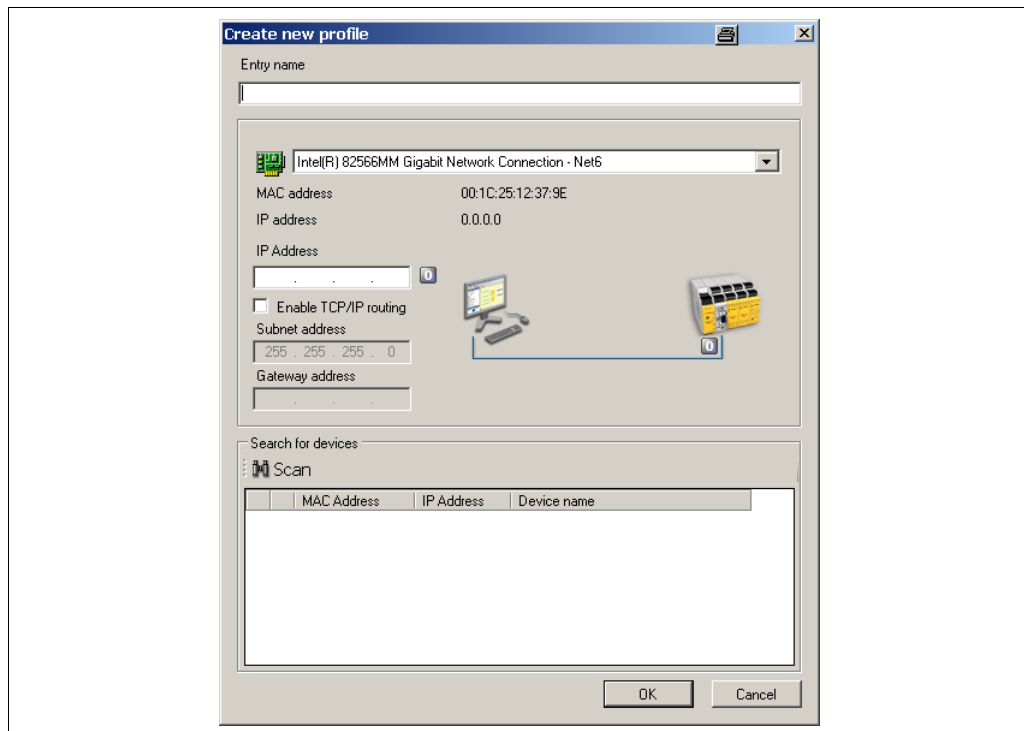
How to add a TCP/IP profile:

NOTE

To create a TCP/IP profile it is necessary that your samos[®]PRO system contains an Ethernet gateway (e.g. SP-EN-IP, SP-EN-PN or SP-EN-MOD) which must be configured with a valid IP address for your network. For detailed instructions on the gateway configuration please see the samos[®]PRO gateways operating instructions (Wieland Electric doc. no. BA000587).

- ➔ Click on the **Add TCP/IP profile** button. The **Create new profile** dialog is opened.

Fig. 37: Create new profile dialog (TCP/IP)



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

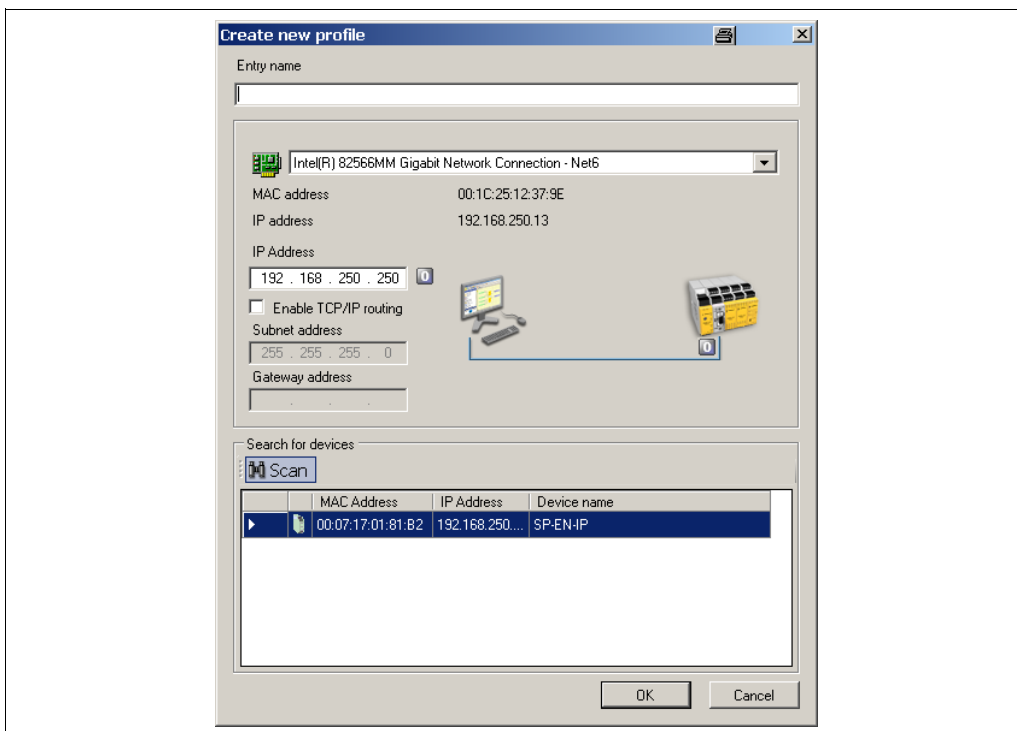


Fig. 38: List of gateways found

- ➔ Click on the desired gateway. The IP address of the device is displayed in the **IP address** field.
- ➔ Enter a name for the new profile.
- ➔ Click **OK**. The dialog is closed and the new profile is displayed in the list.
- ➔ To activate the new profile, select it using the left mouse button and click on the green arrow symbol at the right. From now on, the samos®PLAN will use this profile.

How to check a profile:

- ➔ Click on the green tick on the right side of the profile to be checked.
- ➔ To check all profiles, click on the **Check all profiles** button.

The samos®PLAN checks the connection settings and marks faulty profiles.

Profile type	Profile not checked	Profile OK	Profile faulty
Serial (COM)			
TCP/IP			

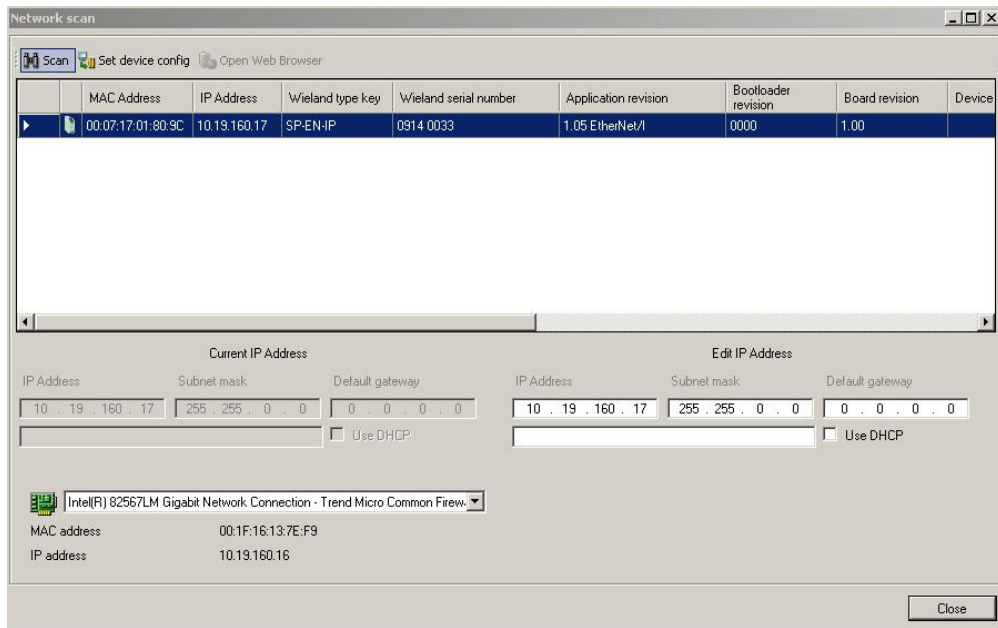
Tab. 10: Status symbols for connection profiles

How to change the network settings of a samos®PRO gateway:

- ➔ Click on the **samos®PRO gateway network settings** button. The **Network scan** dialog is opened.
- ➔ If necessary, select the correct network adapter from the dropdown list at the bottom of the dialog.
- ➔ Click on the **Scan** button. Your network is scanned for connected gateways and the devices found are displayed in the list.

Connecting to the samos®PRO system

Fig. 39: List of gateways found



- ➔ Click on the gateway you want to edit.
- ➔ Enter the new settings in the **Edit IP Address** area.
- ➔ Click on the **Set device config** button to transfer the new settings to the device.

NOTE

If the samos®PLAN identifies a samos® series gateway (i.e. SA-EN-IP) in the network, this will be displayed in the list as well. These gateways are equipped with an internal web server and can be addressed using the **Open Web Browser** button.

6.3 Establishing a connection with the samos®PRO system



Do not connect to the samos®PRO system via the RS-232 and the Ethernet interface at the same time!

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

- ➔ Click on the **Connect** button. The samos®PLAN will try to connect to your samos®PRO system using the currently activated connection profile.

If a connection is established successfully, the software goes into online mode and you can perform the following activities depending on your user level:

- *Log in* (see section 6.4 “User levels in the samos®PLAN” on page 47).
- *Transfer* the configuration to the device, *upload* it from the device or *verify* the configuration (see chapter 10 “Transferring the System Configuration” on page 195).
- *Run* or *stop* the device (see section 11.1 “Changing the device state” on page 200).
- Start the *force mode* (see section 8.14 “Force mode” on page 188).

6.4 User levels in the samos®PLAN

If the samos®PLAN is connected to the devices in a project (i.e. is in online mode), you can switch to the user levels of the samos®PLAN. These user levels have different authorisations for the transfer of configurations to the devices:

User level	Password	Authorisation
Machine operator	None	May create and edit configurations offline Can connect to the system (upload, diagnostics) Cannot transfer any configuration Cannot verify a configuration
Maintenance personnel	Default: None (i.e. no login possible) Can be changed by Authorized client	May create and edit configurations offline Can transfer verified configuration Can connect to the system (transfer, diagnostics) Cannot verify a configuration
Authorized client	Default: SAMOSPRO Can be changed by Authorized client	May create and edit configurations offline Can transfer verified and non-verified configuration May perform diagnostics Can connect to the system (transfer, diagnostics) Can use Force mode Can verify a configuration

Tab. 11: User level authorisations

Switch to the user group Machine operator!

If you leave the PC connected to devices without personal attendance or supervision, you must log off from the user levels *Maintenance personnel* or *Authorized client* and switch to the user level *Machine operator* to make sure that no unauthorized person can transfer configurations to the devices!



WARNING

The password protection relates to the configuration of the current devices. The password is saved in the removable memory plug. This means that the password will remain the same even if the controller module is replaced.

NOTE

How to change the user level:

- In the **Hardware configuration** view, click on the **Log in** symbol on the left side of the configuration area while you are online. The **Change user group** dialog will open.
- Select the desired user level, enter the password and click on **Log On**.

Connecting to the samos®PRO system

How to assign or to change the password for a user level:

- ➔ Go into online mode.
- ➔ Open the **Hardware configuration** view.
- ➔ With the right mouse button, click on the SP-SCON(-NET).
- ➔ From the context menu, select the **Change password...** command. If you are not logged in as *Authorized client*, you will be prompted to log in now.
- ➔ In the **Change password** dialog, select the user level for which you want to change the password, enter the new password twice and confirm with **OK**.

NOTE

The password may consist of 8 characters maximally.

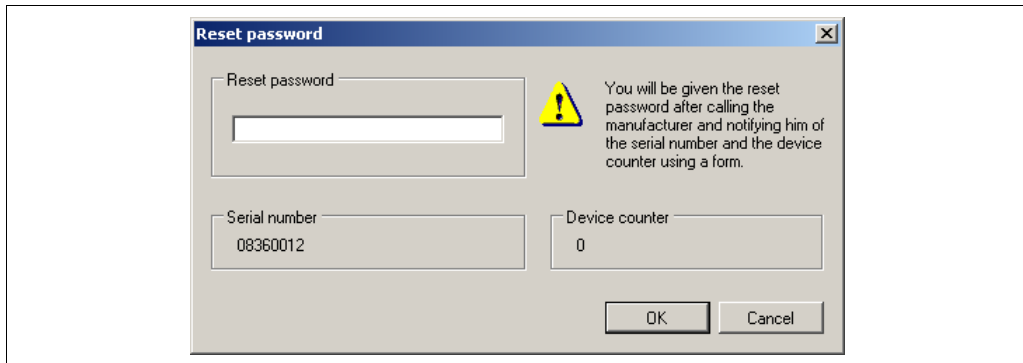
Fig. 40: Change password dialog



How to reset the password:

- ➔ Go into online mode.
- ➔ Open the **Hardware configuration** view.
- ➔ With the right mouse button, click on the SP-SCON(-NET).
- ➔ From the context menu, select the **Reset password...** command. The **Reset password** dialog opens.

Fig. 41: Reset password dialog



- ➔ Note the serial number of the removable memory plug and the device counter displayed in this dialog and contact Wieland Electric support. You will receive a reset password that will allow you to reset all passwords to default factory settings.
- ➔ Enter the reset password in the Reset password dialog and confirm with **OK**.

6.5 Identify project

The **Identify project** command is equivalent to the **Connect to physical device** command that can be executed upon program start of the samos®PLAN.

- ➔ In the **Device** menu, choose the **Identify project** command. The current project will be closed.
- ➔ The samos®PLAN will search for connected samos®PRO devices and load the hardware configuration into the hardware configuration dialog. Once all modules have been identified, the samos®PLAN will ask whether the configuration shall be uploaded.
- ➔ Click **Yes** if you want to upload the configuration.

7 samos®NET

7.1 samos®NET overview

samos®NET allows you to combine up to four samos®PRO stations via EFI for safe data communication. Only SP-SCON-NET controller modules can be used in a samos®NET system, the connection of SP-SCON modules is not possible.

The process data of each station (inputs and outputs, logic results etc.) can be made available immediately to all other stations in the samos®NET system. The Teach function allows to temporarily deactivate single stations without impairing the function of the overall system.

Features

- Safe connection of up to four samos®PRO stations via EFI
- Connection via EFI 1 or EFI 1 and 2
- Transfer/receive up to 52 bit of information per station (26 bit per EFI channel)
- Each bit can be assigned a global tag name
- Teaching simulates the presence of temporarily suspended (switched off) stations
- Any station can be used to address and configure the entire system
- The configuration of the entire samos®NET system is stored in a single project file

7.1.1 System requirements and restrictions for samos®NET

The minimum system requirements for samos®NET are as follows:

System component	Minimum version
Hardware	SP-SCON-NET with firmware version V2.00 or higher
Software	samos®PLAN version 1.3.0 or higher

Tab. 12: Minimum system requirements for samos®NET

The samos®NET system can be connected using only EFI 1 or using both EFI 1 and 2. The overall number of status bits per station that can be made available to the other stations in the samos®NET system depends on the connection method:

Connection method	Available status bits per station
EFI 1	26
EFI 1/2	52

Tab. 13: Available status bits depending on the connection method

You can not use samos®NET and EFI communication at the same time, i.e. it is not possible to connect other EFI compatible devices on the EFI 2 connection while EFI 1 is used for samos®NET.

NOTE

7.2 Function principle

The configuration of a samos®NET project is divided in two parts.

- The first step is the configuration of the network settings and the samos®NET address. Wiring errors or the presence of devices that are not suitable for samos®NET projects are detected in this step automatically by the system.
- The second step is the configuration of the individual stations in the system: Main module, extension modules, connected elements, gateways, logic and the process image for the samos®NET network.

7.2.1 samos®NET address

The samos®NET *address* is required by the samos®PLAN to uniquely identify each of the up to 4 stations in a samos®NET system. This is the first important setting in order to configure a samos®NET system.

The samos®NET address ranges from A to D and is freely configurable. For detailed information on how to assign samos®NET addresses to connected stations please see section 7.4.3 “samos®NET system: Network settings” on page 63.

7.2.2 samos®NET ID

The samos®NET *ID* is necessary for the stations in a samos®NET network to communicate with each other. All stations in a samos®NET system must have an identical samos®NET ID in order to exchange their process image information. This ensures that only stations that belong to the same samos®NET system can communicate with each other. If a differing samos®NET ID is detected in a samos®NET system, all connected stations will change into “Invalid configuration” mode (MS LED flashing red at 1 Hz).

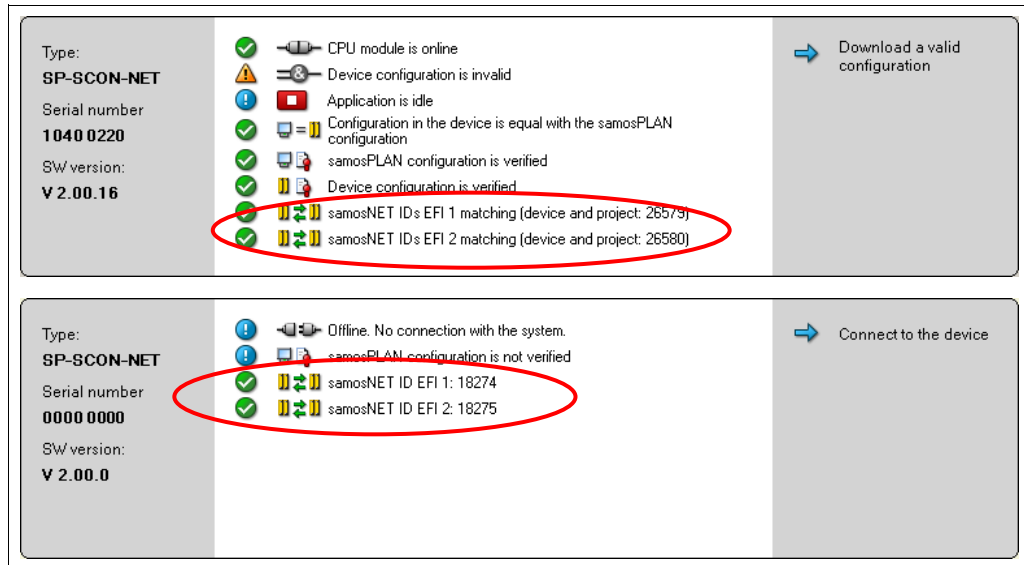
The samos®NET IDs are numeric values that are calculated from the default values for the process image. This means that a change of the default values for the process image of any station will change the samos®NET ID of all stations. Adding or deleting a station will also change the samos®NET ID of the system.

NOTE

If a change is made to any station’s process image, you must transfer the new configuration to all stations (transfer all in the samos®NET network settings view). This will set all samos®NET IDs simultaneously to the same value. Not doing so will lead to a samos®NET ID mismatch in the system and thus interrupt the safety communication between the stations.

The samos®NET IDs are a part of the configuration and are transferred and saved with the configuration in the removable memory plug of each connected SP-SCON-NET module. The samos®NET IDs for the *current configuration in the samos®PLAN* are always displayed in the samos®NET menu bar. The samos®NET IDs that are *currently stored in the individual stations* are displayed in the samos®NET **System overview** and compared to the samos®NET ID of the project on the PC, while the station is connected. If the samos®PLAN detects a samos®NET ID mismatch, a warning sign is displayed. On the right side of the screen a recommendation is given how to proceed:

Fig. 42: samos®NET IDs display in the samos®NET System overview



The samos®NET IDs are also displayed in the samos®NET **Network settings** view.

If the configuration of any station in the samos®NET system is changed in a way that affects the process image of the system (e.g. if a station is added to the system or if the default value for one of the transmitted bits is changed), then the samos®PLAN calculates a new samos®NET ID based on the changed process image. In this case you have to

transmit the configuration to *all stations in the system*, not only to the station whose configuration you have changed. Otherwise the new samos®NET IDs will be transferred only to this station while the other stations will keep the old samos®NET IDs. The resulting samos®NET ID mismatch between the stations will disrupt the process image communication in the system. If a differing samos®NET ID is detected, no process image transfer between the stations is possible and all controller modules (SP-SCON-NET) in the system will show a recoverable error (MS LED flashing **Red** at 1 Hz and EFI 1 and EFI 2 LEDs will light up **Red**). Only configuration and diagnostics of the stations is still possible then.

For more information on how to correct a samos®NET ID mismatch please see section 7.5 “samos®NET troubleshooting” on page 70.

7.3 Getting started

This chapter describes how to set up a new samos®NET system. In order to do this, you need to configure the hardware for your project first. You have two possibilities:

- You can first set up and wire the hardware, then connect your PC to the system and read in the hardware setup using the samos®PLAN software.
- If the required hardware is not available yet, you can set up the hardware configuration for your samos®NET project in the samos®PLAN and transfer the configuration later when you have assembled the hardware.

Either way, once the hardware configuration for your samos®NET project is complete, you can proceed with the software configuration. The last step is to transfer the finished configuration to the stations, to verify the configuration of the stations and to run the system.

7.3.1 Connecting to an existing samos®NET system

Step 1: Assemble and wire the hardware

- Set up the hardware for your samos®NET system (samos®PRO SP-SCON-NET controller modules, extension modules and connected devices such as sensors, switches, actuators etc.).

For information on wiring please see the samos®PRO hardware operating instructions (Wieland Electric doc. no. BA000497).

NOTE

Step 2: Establish a connection to your samos®NET system

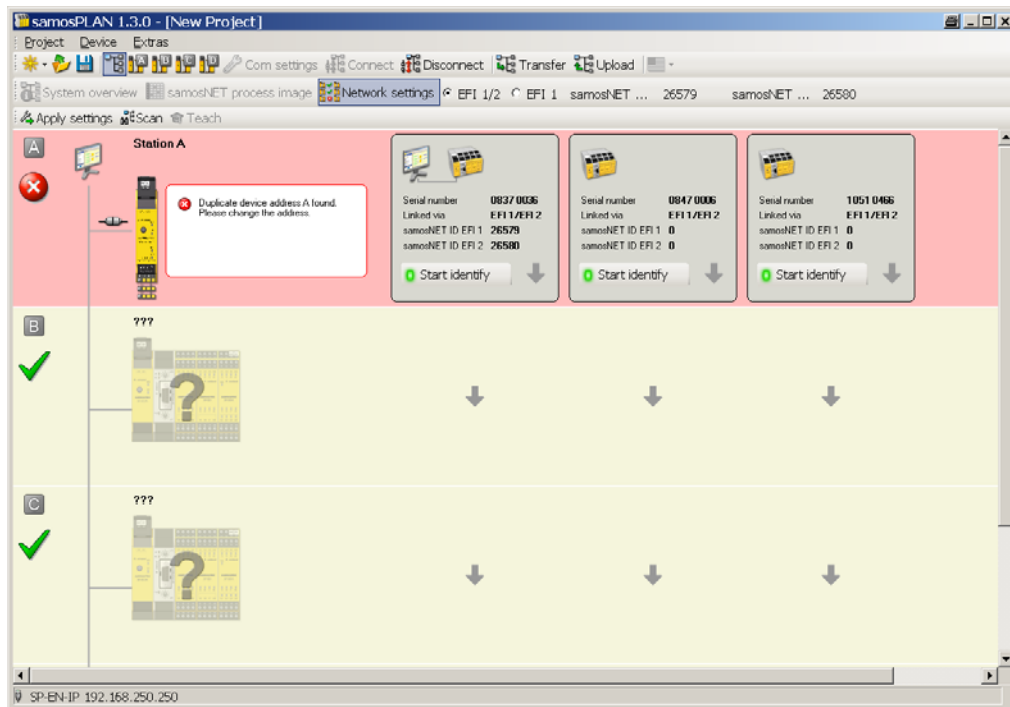
- Connect a PC or notebook to the RS-232 interface of any SP-SCON-NET in the system.
- Power on the samos®PRO system.
- Open the samos®PLAN configuration tool installed on the PC.
- If necessary, edit the communication settings (see chapter 6.2 “Editing the communication settings” on page 43).
- Click on **Connect to physical device** or choose the **Identify project** command from the **Device** menu. The samos®PLAN will then scan the network for connected devices.

If the error message “No valid samos NET station available” appears, check whether each of your SP-SCON-NET has at least the firmware version V2.00. The firmware version can be found on the type label of the module in the field *Software version*.

NOTE

- If the connected controller modules have not been configured for samos®NET before, they have no valid samos®NET addresses. In this case, the samos®PLAN will now open the samos®NET **Network settings** view and list the found stations:

Fig. 43: samos®NET
Network settings without
valid address assignment

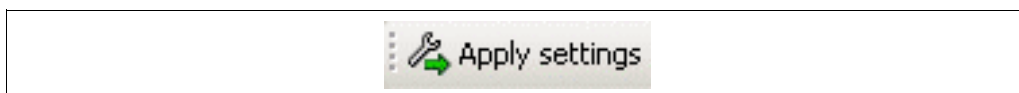


- ➔ Use the up and down arrow buttons or drag and drop the stations into the rows for Station A to D so that no two stations occupy the same address.
- ➔ If you are not sure which station is which, there are two possibilities to identify each station:

Click on the **Start identify** button of one of the displayed stations. The corresponding station's MS and EF1 LEDs will start blinking alternating with the EF2 LED (2 Hz). You will need the password for Authorized clients. The default password is "SAMOSPRO". To stop the LED blinking, click the button again (it is now labeled **Stop identify**).

Check the serial number on the removable memory plug and compare it with the serial number shown in the samos®PLAN.

Fig. 44: Apply settings
button



- ➔ Click on the **Apply settings** button in the upper left hand corner of the screen. The samos®NET addresses of the stations will be changed.

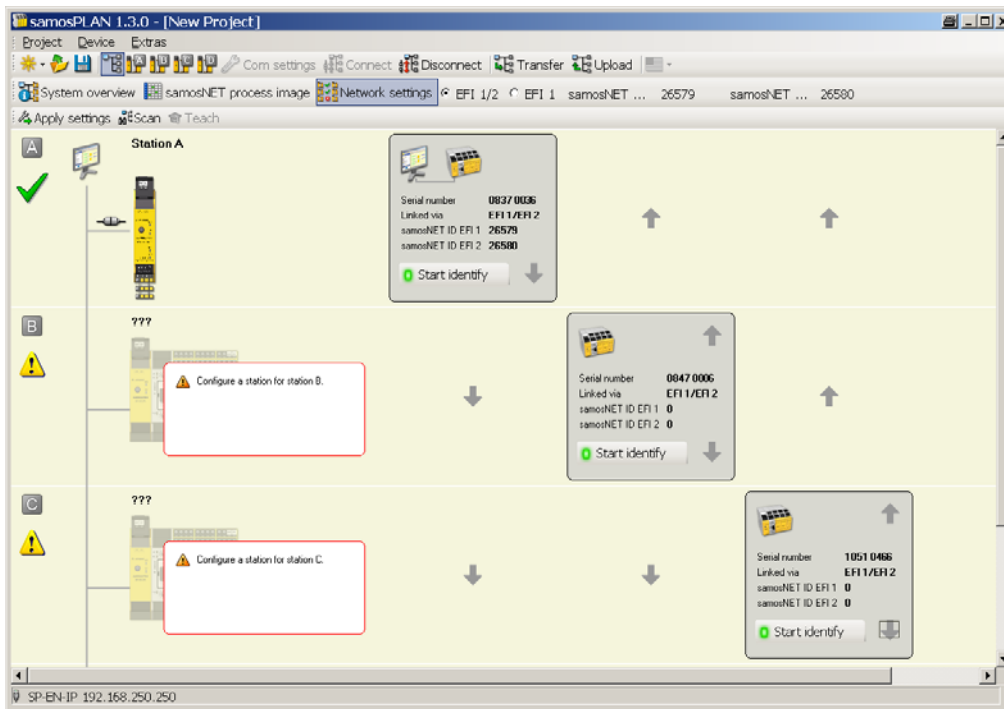


Fig. 45: samos®NET Network settings with valid address assignment

Step 3: Upload the hardware settings

- ➔ Click on the **System overview** tab. The following view opens:

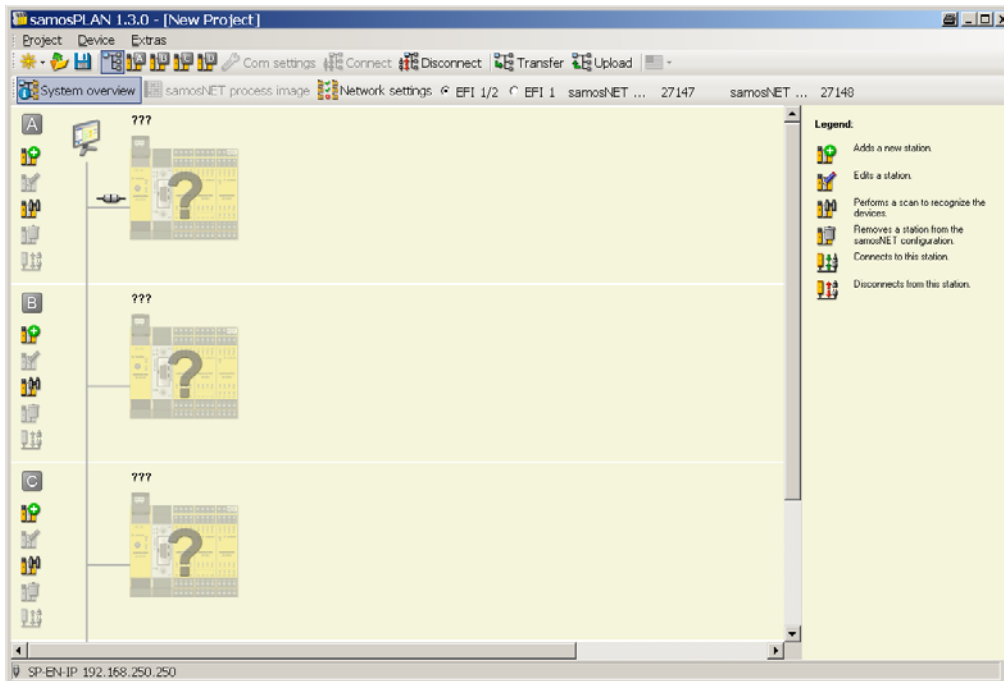


Fig. 46: samos®NET System overview view

- ➔ Click on the binocular button next to station A. The samos®PLAN will upload the hardware and configuration settings for all devices on this station. Repeat this for all stations.
- ➔ When the hardware configuration is complete, click **Disconnect**. You can now configure your project as described in section 7.3.3 “samos®NET configuration” on page 55.

Disconnecting in System overview will disconnect all samos®NET stations simultaneously. The buttons for **Transfer** and **Upload** react the same way as well. If you switch to the view for an individual station, these buttons will affect only that station.

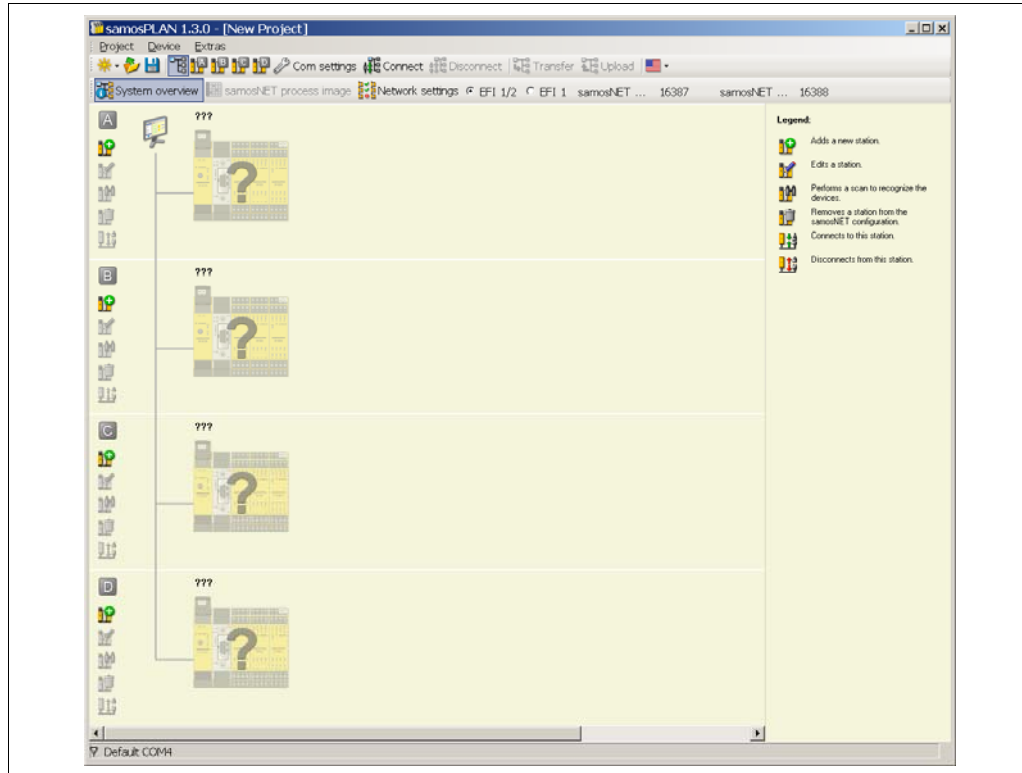
NOTE

7.3.2 Setting up a samos®NET project in the samos®PLAN

If the required hardware is not available yet, you can set up the hardware configuration for your samos®NET project in the samos®PLAN.

- ➔ Open the samos®PLAN software on your PC or laptop.
- ➔ In the startup dialog, click on **Create new samos®NET project** or choose from the **Project** menu the command **New > samos®NET system project**. The samos®NET **System overview** screen opens.

Fig. 47: samos®NET System overview screen



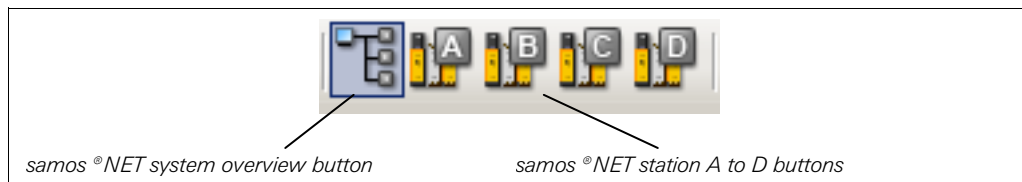
- ➔ Choose whether your samos®NET system will be wired using only EFI 1 or both EFI connections. EFI 1 allows to exchange up to 26 bits per station, using both EFI 1 and EFI 2 each station can share up to 52 bits with the other stations.

NOTE

You can change this setting any time later.

- ➔ Now add the first station to your project. Click one of the buttons for adding a new station on the left side of the screen. This will open the view for this individual station. Alternatively you can also switch to the view for an individual station using the station buttons in the toolbar at the top of the screen.

Fig. 48: samos®NET system overview and station buttons



- ➔ In the view for the individual station, add the desired hardware for this station as described in section 5.5.1 "Exercise for configuring the samos®PRO modules" on page 18 and in section 5.5.3 "Exercise for configuring the connected devices" on page 20.
- ➔ When the hardware configuration for the selected station is complete, use the **samos®NET system overview** button in the toolbar to switch back to the samos®NET system view.
- ➔ Then add the other required station(s) to your samos®NET project as described above.
- ➔ When the hardware configuration of your samos®NET system is complete, you can configure your project as described in the following section.

7.3.3 samos®NET configuration

This section shows in detail how data can be shared between the single stations in a samos®NET system. To this purpose, we will create a simple samos®NET project with two stations where an emergency stop button and a restart button on Station A will control two robots connected to Station A and Station B:

Setting up the hardware

- Create a new samos®NET project (see section 7.3.2 “Setting up a samos®NET project in the samos®PLAN” on page 54).
- In the samos®NET **System overview**, set the connection method to **EFI 1/2** (setting it to **EFI 1** will make no difference for this example project). Then click on the **Add a new station** button for Station A. The **Hardware configuration** view for Station A opens.
- Add a SP-SCON-NET and then an SP-SDIO module for Station A.
- Connect a single channel emergency stop button to input I1 and a single channel reset button to input I2 of the Station A SP-SDIO module.
- Connect a single channel robot to output Q1 and a lamp to output Q2 of the Station A SP-SDIO module.
- Now click on the button for Station B in the toolbar. The **Hardware configuration** view for Station B opens.
- Add a SP-SCON-NET and then an SP-SDIO module for Station B.
- Connect a single channel robot to output Q1 and a lamp to output Q2 of the Station B SP-SDIO module.

Setting up the logic for Station A

- Click on the button for Station A in the toolbar. Then switch to the **Logic editor** view for Station A.
- Using the connected input and output elements on the SP-SDIO module and a **Restart** function block, create the following logic configuration:

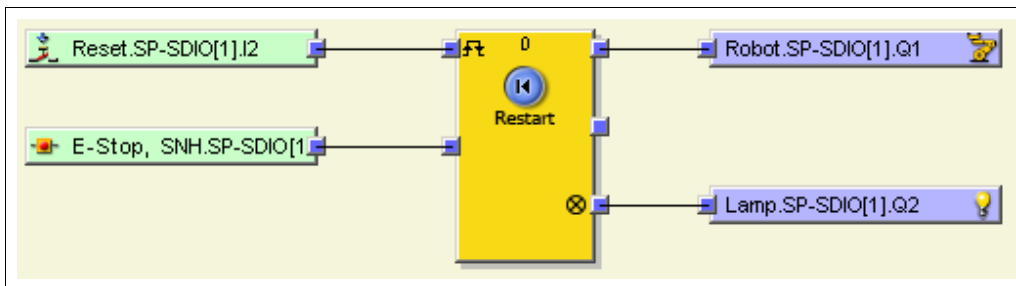


Fig. 49: Logic configuration example (Station A)

Setting up the samos®NET routing for Station A

- In the Logic editor for Station A, add an additional **Routing N:N** function block, configure it for two inputs and outputs and connect its inputs to the SP-SDIO inputs for the reset button and the emergency stop button.
- Drag two outputs of the Station A controller module on the logic editor worksheet. You will find the outputs in the output selection window under **SP-SCON-NET**.

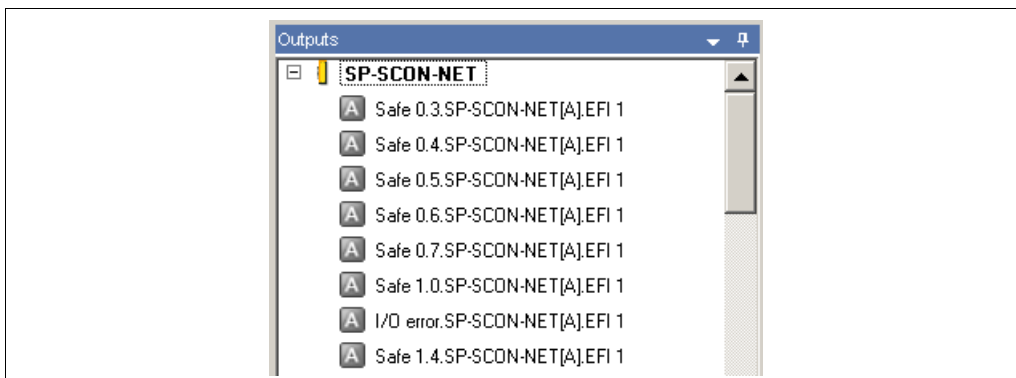
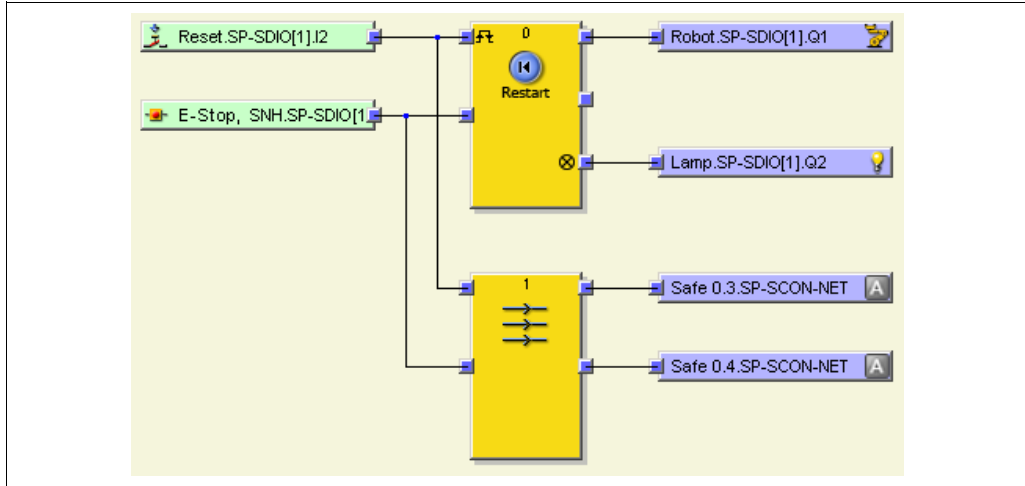


Fig. 50: CPU outputs in the output selection window

NOTES

- A square with the letter A–D within it denotes a bit within the samos®NET process image.
 - Each output can be used only once. Used outputs are displayed green.
- ➔ Connect the outputs of the **Routing N:N** function block to the two SP-SCON-NET[A] outputs (e.g. **Safe 0.3.SP-SCON-NET[A].EFI 1** and **Safe 0.4.SP-SCON-NET[A].EFI 1**) as shown in Fig. 51.

Fig. 51: samos®NET routing logic example



- ➔ Note which input is being routed to which output.

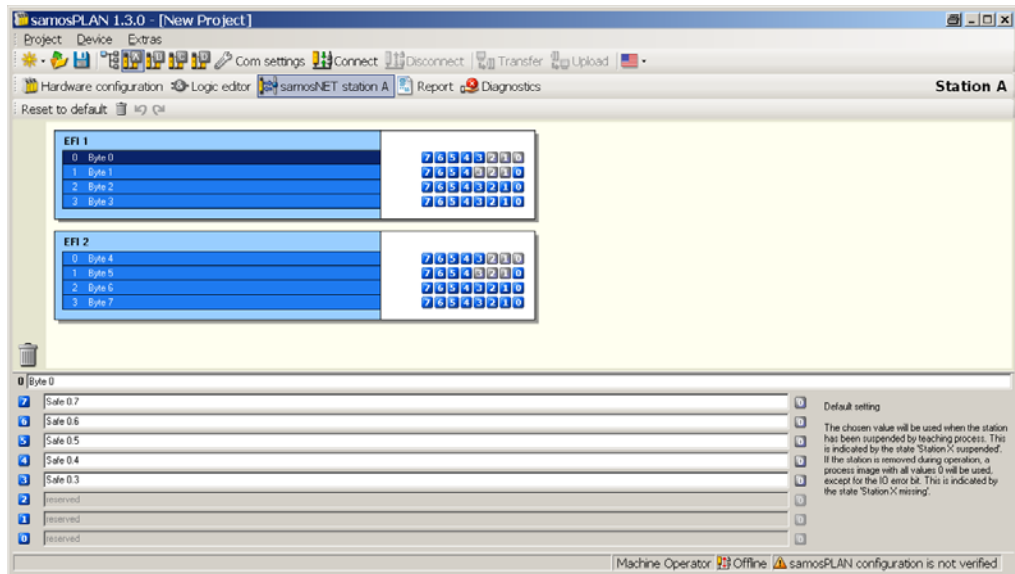
NOTE

In more complex projects it is recommended to configure the routing connections on a separate page in the logic editor. Otherwise the logic design could get confusing.

Assigning tag names for the samos®NET routing

- ➔ Still in the Station A view, switch to the samos®NET routing table using the **samos®NET station A** button in the toolbar (if your project contains also one or two gateways, you will find this entry in the submenu under **Gateways**).
- ➔ Click on **Byte 0** in the **EFI 1** area to display the tag names for Byte 0 and its bits in the lower half of the window.

Fig. 52: samos®NET routing table and tag names



- ➔ Now replace the default tag names (e.g. **Safe 0.3** and **Safe 0.4**) with more expressive tag names (e.g. **Global Reset** and **Global E-stop**). The assigned tag names will be displayed in the logic editor from now on.



Fig. 53: Assigned samos®NET tag names in the routing configuration and in the logic editor

It does not make a difference for the routing whether you use the default tag names or change them, but assigning clear tag names will help you to keep track of things in your projects.

NOTE

Setting up the logic for Station B

- Click on the button for Station B in the toolbar. Then switch to the **Logic editor** view for Station B.
- In the inputs selection window, find the two inputs from the Station A controller module that are routed via samos®NET. You can recognize them by their tag names:

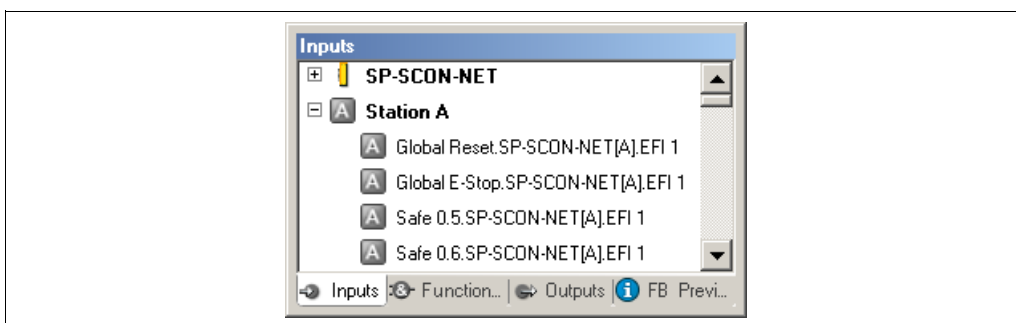


Fig. 54: Routed inputs from Station A in the Station B logic editor

- Using these inputs, the output elements on the Station B SP-SDIO module and a **Restart** function block, create the following logic configuration:

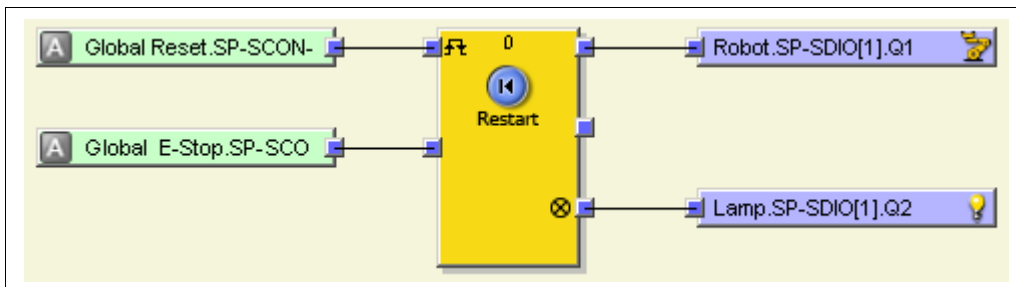


Fig. 55: Logic configuration example (Station B)

With this step the example project is finished. The input from the emergency stop button and from the reset button connected to Station A is routed to Station B via samos®NET so that the robots connected to both stations can be controlled simultaneously.

7.3.4 Transferring and verifying the samos®NET configuration

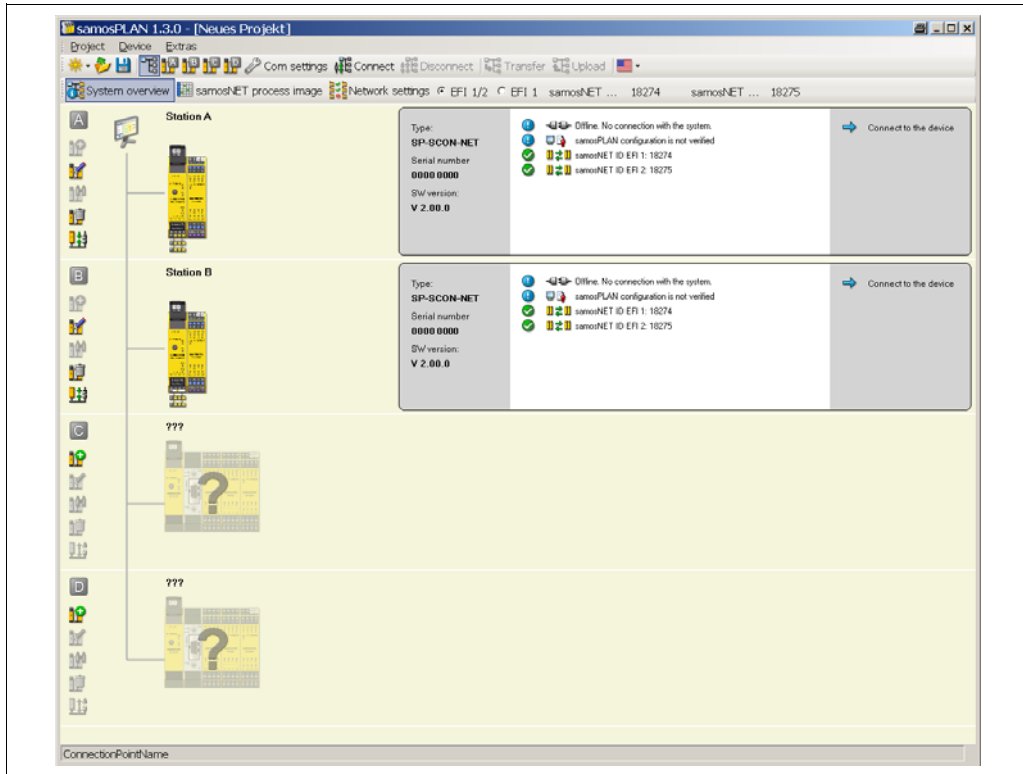
The last step in order to get your samos®NET system running is to connect the PC with the system, transfer and verify the configurations and switch the stations into Run state. This requires that you have finished the configuration as described in the previous section and that you have set up and connected the required samos®PRO modules and other hardware.

Establish a connection to your samos®NET system

- Connect a PC or notebook to the RS-232 interface of any SP-SCON-NET in the system.
- Power on the samos®NET system.
- Open the samos®PLAN configuration tool installed on the PC and load the project file with your configuration.
- If necessary, adjust the communication settings (see chapter 6.2 “Editing the communication settings” on page 43).

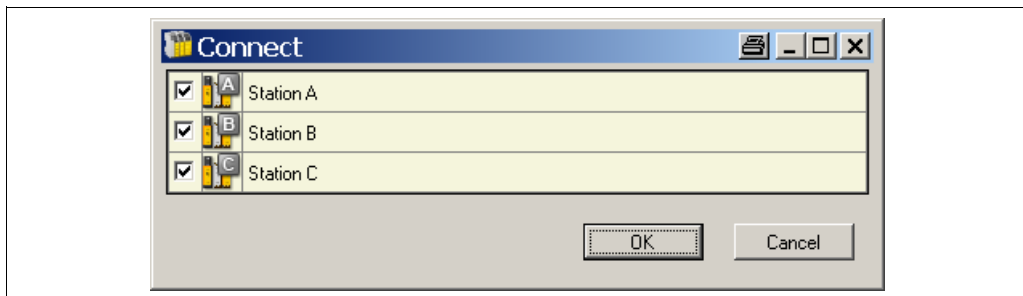
- ➔ Switch to the samos®NET system overview. The configured stations in your project will be displayed with a light yellow background.

Fig. 56: samos®NET system overview, system not connected



- ➔ Click on **Connect**. You will be prompted to select the stations you want to connect to. Activate all stations, then click **OK**.

Fig. 57: Connect dialog



- ➔ If the SP-SCON-NET modules of the connected stations have not been configured for samos®NET before, the samos®PLAN will now open the samos®NET **Network settings** view and list the found stations. In this case you will have to assign a unique samos®NET address from A to D to the individual stations as described in section 7.3.1 “Connecting to an existing samos®NET system” on page 51 (step 2).

The samos®PLAN will connect to the samos®NET system, compare the existing hardware and software configuration with the configuration in samos®PLAN and display the results. If the configuration in samos®PLAN is not identical to the configuration in the connected stations, these will be displayed with a blue background.

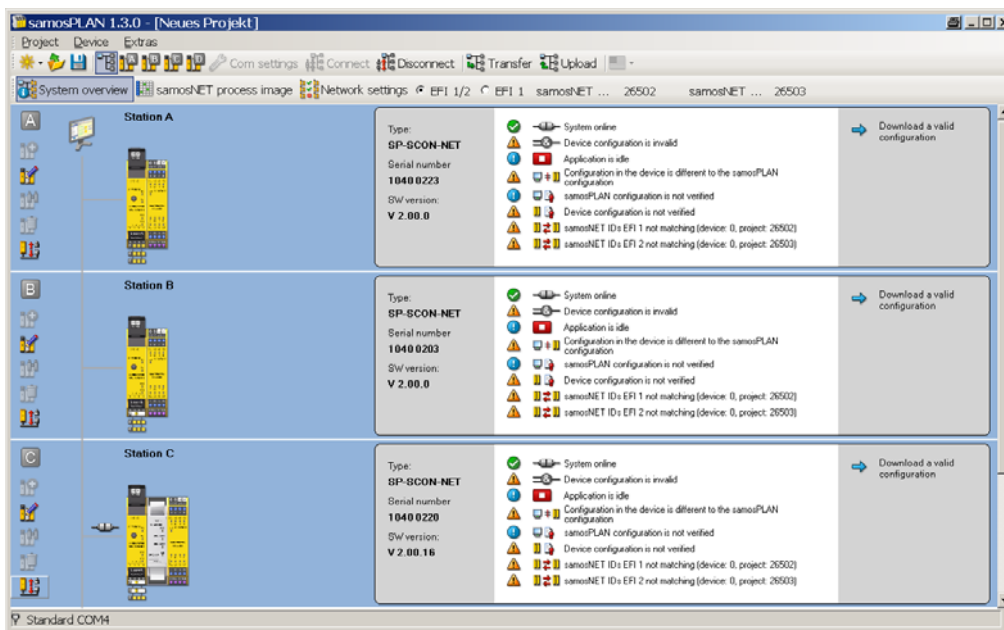


Fig. 58: samos®NET system overview, system connected, differing configurations

Transfer the configuration

- Now transfer the configuration to the stations by clicking on **Transfer**. Again you will be prompted to select which station(s) you want to transfer the configuration to.
- Select all stations and click on **OK**. The samos®PLAN will now transfer the configuration to each station. You will need the password for Authorized clients; the default password is "SAMOSPRO".
- Once a valid configuration has been transferred to a station successfully, the samos®PLAN will ask you whether you want to set this station into the Run state. Click either **Yes** or **No**. The station will be displayed in the samos®NET system overview with a grey background.

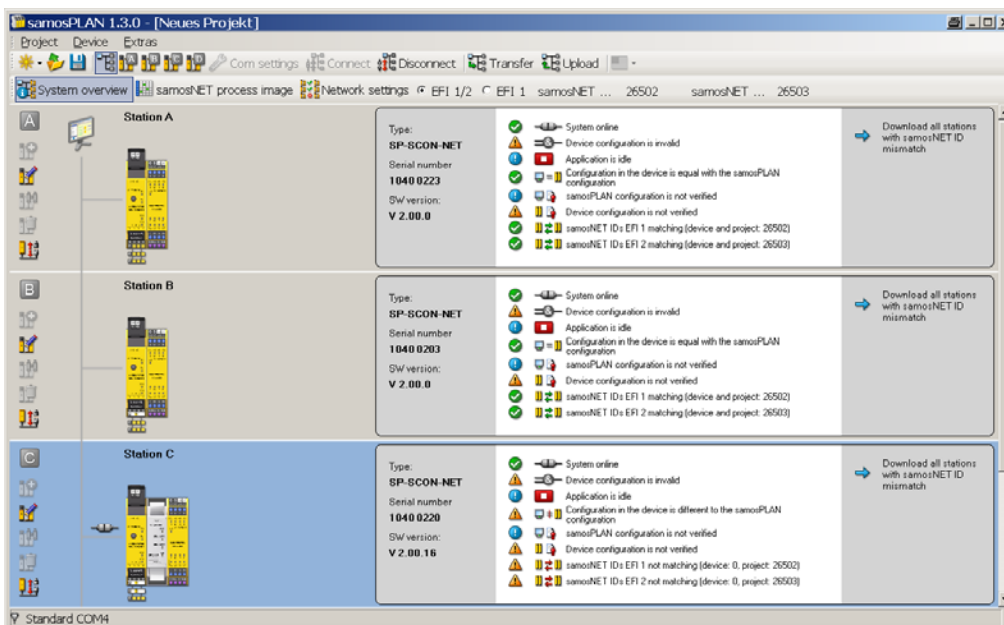


Fig. 59: samos®NET system overview, system connected, configuration identical but not verified

You can run or stop each station also in the hardware configuration view for the respective station.

NOTE

Verify the configuration

- Switch to the hardware configuration view for any station in your project. If the device configuration is valid and equal with the configuration in the samos®PLAN, but not yet

verified, the CV LED on the SP-SCON-NET will blink as well as the **Upload and verify configuration** button on the left side of the configuration area.

- ➔ Click the **Upload and verify configuration** button. The samos®PLAN will upload the configuration from the module and compare it to the configuration in the Software. If both configurations match, the result will be displayed as a report and you will be asked whether you want to set the device to verified. Read the report carefully.

NOTE

You can print the report or save it in PDF format.

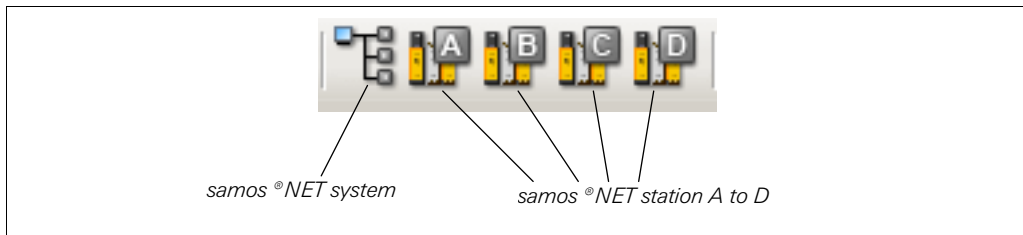
- ➔ If you have checked the report and want to set the device to verified, click **Yes**. The device will be set to verified, and, if it is currently not in the Run state you will now be asked whether you want to run it.
- ➔ Repeat this for all stations in your samos®NET system.

For more detailed information on transferring and verifying a configuration please see chapter 10 “Transferring the System Configuration” on page 195.

7.4 samos®NET functions

This chapter gives a comprehensive overview over the samos®NET functions in the samos®PLAN. These functions are divided into samos®NET system functions and functions that are related to the single stations in a samos®NET system. You can switch between the view for the samos®NET system and the individual station views using the additional buttons that appear in the toolbar of the samos®PLAN if a samos®NET project is open.

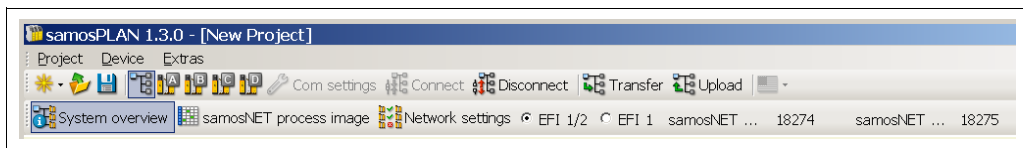
Fig. 60: samos®NET system and station buttons



samos®NET system functions

In the samos®NET system view, you can switch between the different samos®NET system functions using the buttons that are displayed under the menu bar:

Fig. 61: samos®NET system menu bar



- The samos®NET **System overview** provides information on the configured/connected stations and their status. See section 7.4.1 “samos®NET system: System overview” on page 62.
- The **samos®NET process image** allows you to monitor the information that is exchanged between the connected samos®NET stations. See section 7.4.2 “samos®NET system: Process image” on page 63.
- In the samos®NET **Network settings** view you can scan the samos®NET network for connected stations, view and assign the **samos®NET address** to the connected stations and distribute the stations to their individual positions (A to D) in the samos®NET network. See section 7.4.3 “samos®NET system: Network settings” on page 63.
- The **EF1 1/2** and **EF1 1** radio buttons are used to choose the connection method, i.e. whether one or two EFI strings are used.
- On the right side the **samos®NET IDs** for the current configuration in the samos®PLAN are displayed.

samos®NET station functions

The samos®NET functions for an individual station can be reached if the view for this station is active:

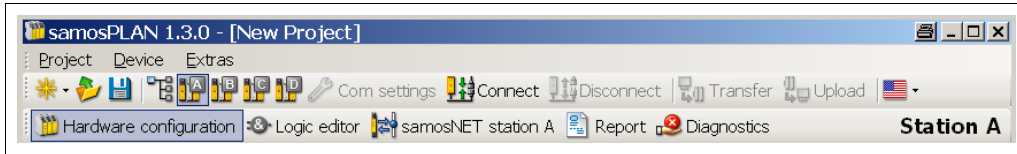


Fig. 62: samos®NET station menu bar

- Most of these functions as the **Hardware configuration**, **Report** or **Diagnostics** are working in the same way as in a standalone project. In this chapter only the additional functions that are relevant for samos®NET will be described.
- The **Logic editor** is used to configure which information each station will send to the other stations via the samos®NET network. This is also where the information provided by the other samos®NET stations in the network is available and can be used as input for logic applications. See section 7.4.4 “samos®NET stations: samos®NET data in the logic editor” on page 65.
- The **samos®NET station X** view is where you can assign tag names to the source information that the station sends to the samos®NET network and set the process image default values (High or Low) that will be used in case the **Teach function** is used to simulate the presence of this station. See section 7.4.5 “samos®NET stations: Station X view and process image” on page 66 and section 7.4.6 “samos®NET stations: Teach function” on page 67.

If a gateway is connected to the station, the **samos®NET Station X** button is integrated in the **Gateways** menu:

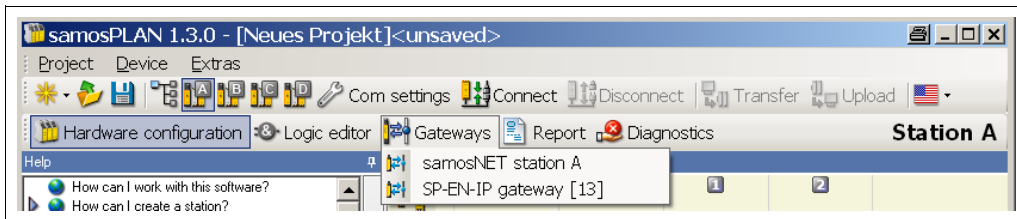
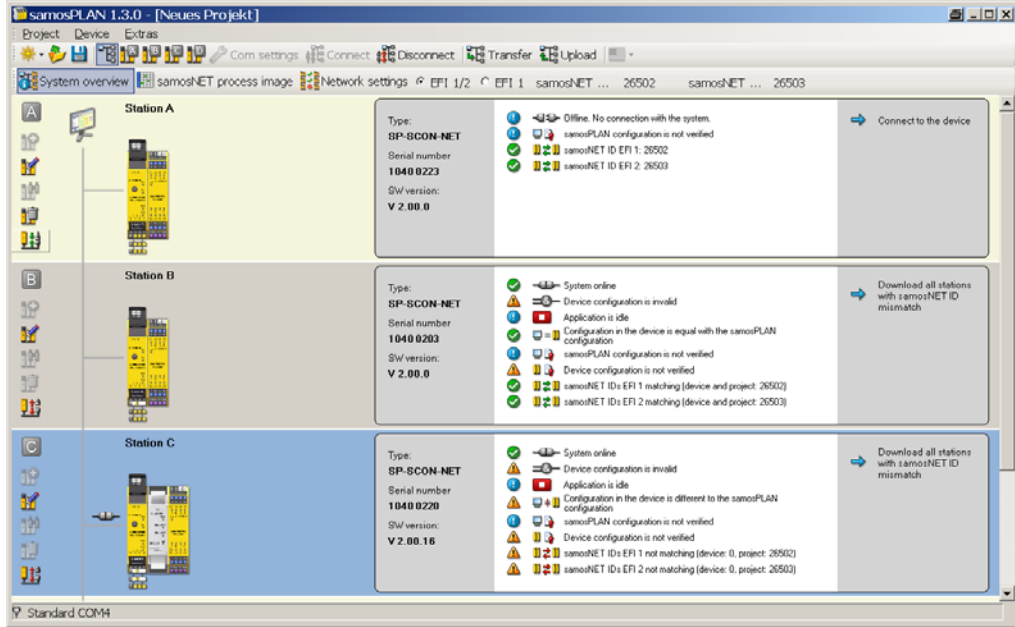


Fig. 63: samos®NET station menu bar with gateway

7.4.1 samos®NET system: System overview

The samos®NET **System overview** provides information on the configured/connected stations and their status. To open the system overview, first click on the **samos®NET system** button in the toolbar and then on the **System overview** button.

Fig. 64: samos®NET system overview



In this view each station is displayed with its current hardware configuration, information on the connected CPU, online status, configuration status and samos®NET IDs. The background color of each station also indicates its online status and configuration status as shown in Fig. 64.

In the example above, Station A is offline (yellow background), while station B is online with a valid configuration (grey background), station C is online with an invalid configuration (blue background) and no station D has been configured (yellow background).

To the left of each station you will find the following tool icons:

Tab. 14: samos®NET system overview tools

Icon	Function	Description
	Add	Adds a new station in the current slot (A to D) and switches to the view for the new station. This function is only available if no station has been configured for this slot yet.
	Edit	Switches to the view for this station where you can edit its properties including the station name, configure the connected hardware, program the logic, verify and lock the configuration etc.
	Recognize	Connects to this station, uploads its hardware configuration, then prompts you whether the software configuration shall be uploaded from this station.
	Delete	Deletes this station from the current configuration. Note: You will not be asked to confirm this command and there is no Undo function. Unsaved changes will be lost.
	Connect	Connects to this station so that you can upload, transfer or verify the configuration, run or stop the application etc.
	Disconnect	Disconnects from this station so that you can e.g. edit the configuration.

NOTES

- If a function is not available, the corresponding icon is displayed grey.
- Instead of clicking the Add or Edit button, you can also switch to the view for an individual station by clicking the corresponding button in the menu bar (station A to D) or by double clicking on that station's graphical representation.
- The Connect or Disconnect buttons next to each station perform their function only for this station while the **Connect** or **Disconnect** button in the menu bar for the samos®NET system overview will connect to or disconnect from the entire samos®NET system (when you click **Connect** there you will be asked which stations you want to connect to).
- It is not possible to set all stations into the Run state or to stop them all simultaneously in the samos®NET system overview. You need to switch to the station view for each station and to log in individually to each station as **Authorized client** with the password for this station in order to run it or to verify the configuration.

7.4.2 samos®NET system: Process image

The **samos®NET process image** allows you to monitor the information that is exchanged between the samos®NET stations. In the left area of the screen the hardware configuration for each station is displayed, on the right side the bits for EFI 1 and EFI 2 (if used) are displayed with their tag names. Bits that are currently High in the process image are highlighted green.

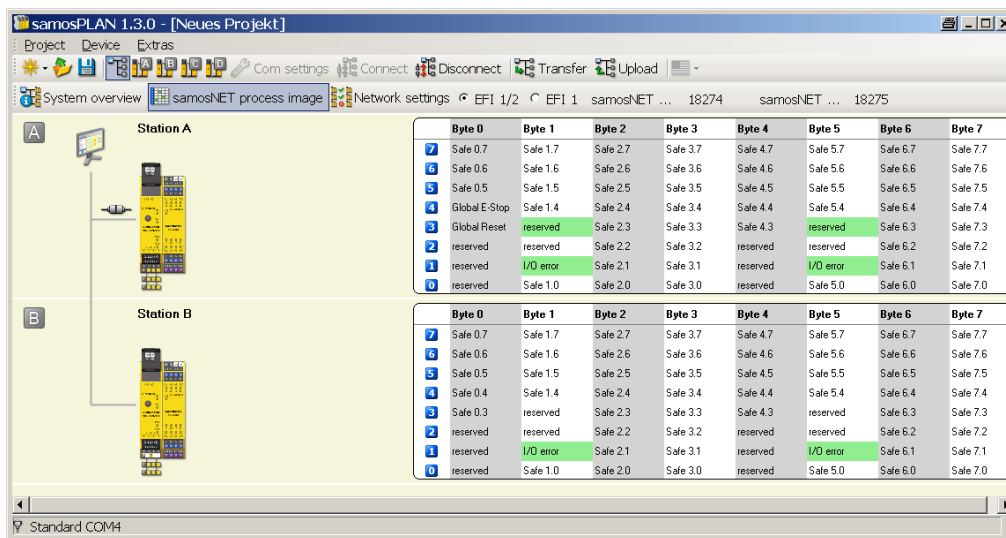


Fig. 65: samos®NET process image

NOTES

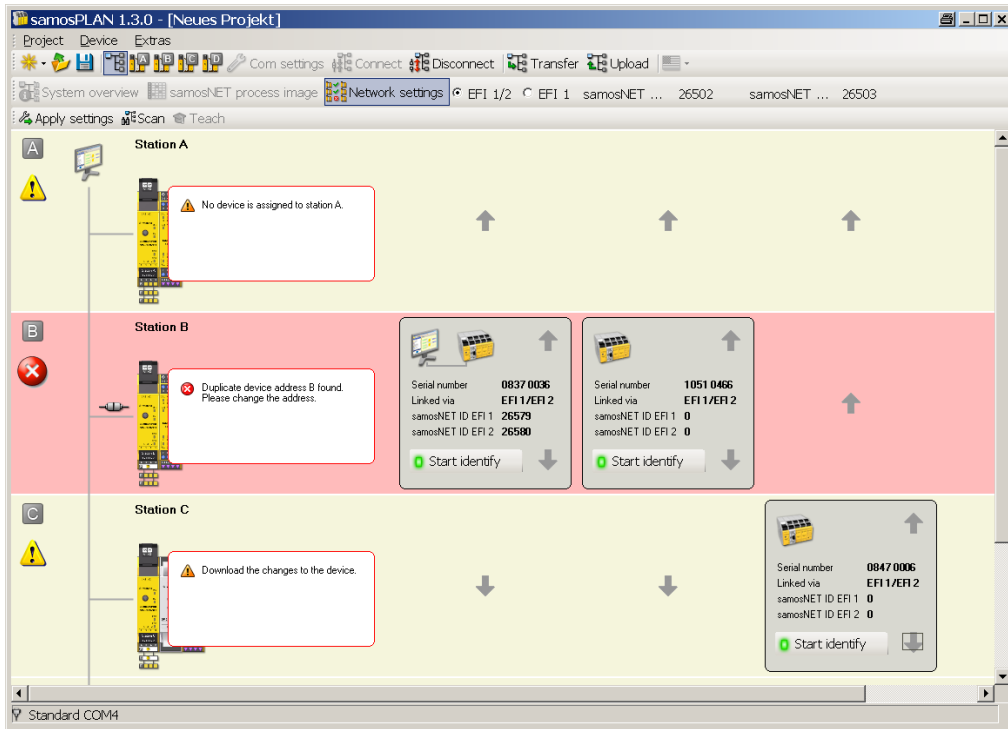
- If a station is not in the Run state, its process image will be set to Low and its I/O error status bits will be set to High (see section 5.6.4 “EFI I/O error status bits in the logic editor” on page 33).
- Double clicking on the hardware symbol for a station will open the routing view for this station where you can edit the tag names for the bits and bytes sent from this station (see section 7.4.4 “samos®NET stations: samos®NET data in the logic editor” on page 65).

7.4.3 samos®NET system: Network settings

The **Network settings** view is where you can assign the samos®NET address (A, B, C or D) to the individual stations in the samos®NET network. This is a prerequisite for the configuration because it enables the samos®NET to address each station and to identify the bits in the samos®NET process image, e.g. Station A, EFI 1, Byte 0, Bit 0.

The **Network settings** view is opened automatically if you connect to a samos®NET system and the software detects an erroneous address assignment, e.g. if two or more connected stations have the same samos®NET address. This will be the case if you have created a samos®NET system with new controller modules or if you have replaced one or more controller modules in an existing system.

Fig. 66: samos®NET Network settings view



If at least one station of a samos®NET system is online, all connected stations are shown with their current address assignment (address A to D). Additionally the removable memory plug serial number and the current samos®NET IDs for EFI 1 and EFI 2 of each station are displayed here. You can update this information using the **Scan** button at the top left of the window. Error messages and warnings regarding the current system status are displayed as a pop-up message for each station.

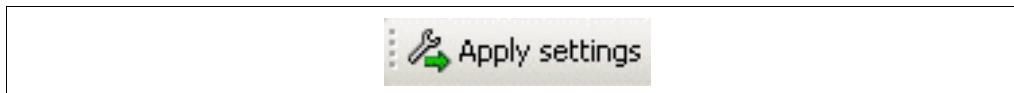
NOTES

- You have two possibilities to identify a station:
 - Click on the **Start identify** button of one of the displayed stations. The corresponding station’s MS and EFI 1 LEDs will start blinking alternating with the EFI2 LED (2 Hz). You will need the password for Authorized clients. The default password is “SAMOSPRO”. To stop the LED blinking, click the button again (it is now labeled **Stop identify**).
 - Check the serial number on the removable memory plug and compare it with the serial number shown in the samos®PLAN. The serial number displayed in the **Network settings** view is the serial number of the removable memory plug, not the serial number of the SP-SCON-NET module.

How to change the assigned samos®NET address (A to D):

- ➔ To change the address of a station, click on the up and down arrow buttons to move it up or down to the desired position in the window. Alternatively, you can drag and drop the station to the desired position.

Fig. 67: Apply settings button



- ➔ Click on the **Apply settings** button in the upper left hand corner of the screen. The samos®NET addresses of the stations will be changed.

NOTES

- The **Apply settings** button has no effect on the samos®NET IDs in the stations. The samos®NET IDs are transferred to the stations as a part of the configuration. That means that if you have made changes to the configuration of any station that have resulted in a change of the samos®NET IDs, then you have to transfer the configuration again to *all stations* in order to apply the new samos®NET IDs.

- It does not matter which samos®NET address is assigned to which station. One recommended possibility is to follow the mounting order in the switching cabinet from left to right for better orientation.
- If you change the address assignment of a samos®NET system, you may have to reconfigure the process image and the parts of the logic programming that use input bits from the samos®NET process image because the samos®NET address is part of the bit assignment in the process image.

7.4.4 samos®NET stations: samos®NET data in the logic editor

The logic editor is the central place where the available information in the samos®NET network is processed:

- Each station can use the information of the other stations as input data
- Each station can provide its own data as outputs

Ensure that all signals are present long enough!

Very short signals may not be recognized and transferred to other samos®NET stations, especially if the logic cycle time of the source is much shorter than the samos®NET system's logic execution time. Take appropriate measures to ensure that all signals are present long enough to be recognized in the samos®NET system (e.g. by using a delay function block in the logic).



Routing of data into the samos®NET network

In order to write data into the samos®NET network so that it can be used by other stations you need to define which bit in the samos®NET process image shall be set. You will find the bits that can be defined for each station in the **Outputs** panel in the logic editor under the symbol for the SP-SCON-NET module:



Fig. 68: samos®NET output bits for Station A in the logic editor

- A square with the letter A–D within it denotes a bit within the samos®NET process image.
- Each output bit can be used once. Outputs that are already in use are displayed green.
- You can edit the tag names of the output bits in the **samos®NET station X** view (see section 7.4.5 “samos®NET stations: Station X view and process image” on page 66).

NOTES

How to send information into the samos®NET network:

- ➔ Drag the bit you want to define on the worksheet and connect it to the output of a function block.
- ➔ To have the value of an input available directly for all stations in the samos®NET network, use a **Routing N:N** function block as shown in Fig. 69:

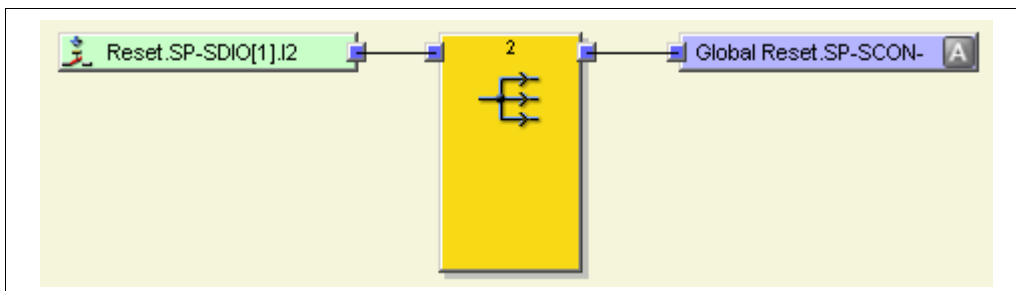
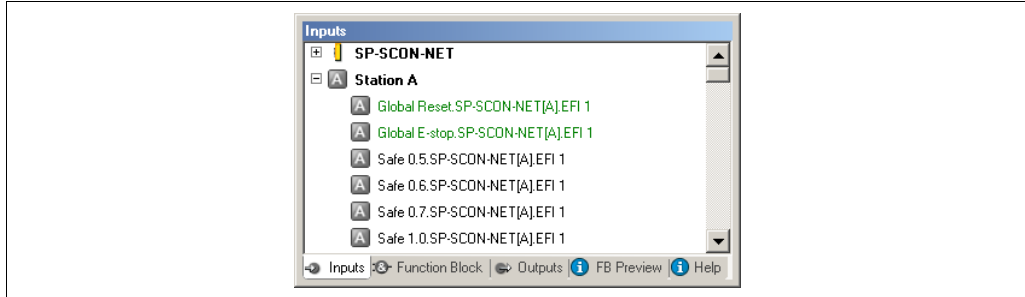


Fig. 69: Routing an input into the samos®NET network

Using data from the samos®NET network

You will find all available information from the other stations in the samos®NET network in the **Inputs** tab of the logic editor under the symbol of the respective station's controller module:

Fig. 70: samos®NET input bits from Station A in the logic editor of another station



You can use these inputs just as any other input.

NOTES

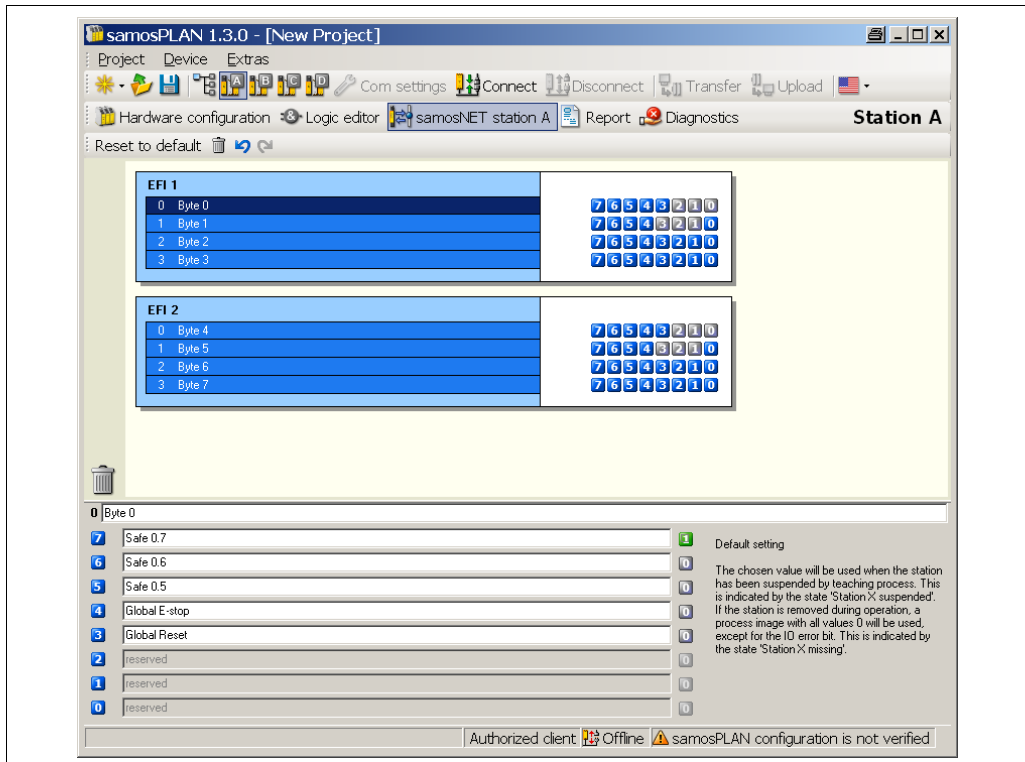
- Inputs can be used several times.
- Inputs that are already used at least once in this station's logic are displayed green.
- The samos®NET inputs are displayed with their respective tag names. You can edit the tag names in the **samos®NET station X** view of the originating station (see section 7.4.5 "samos®NET stations: Station X view and process image" on page 66).

7.4.5 samos®NET stations: Station X view and process image

In the **samos®NET station X** view you can do the following:

- Edit the tag names for the bits and bytes that this station sends to the samos®NET network.
- Set the default values for this station's process image bits to Low or High (see also section 7.4.6 "samos®NET stations: Teach function" on page 67).
- ➔ To open the **samos®NET station X** view press the corresponding button in the toolbar. If the station contains also a gateway, this button can be found in the **Gateway** menu.

Fig. 71: samos®NET station A view



The toolbar

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

- *Reset to default*: Sets the tag names and configured default settings for all bits and bytes to the default values.
- *Clear selected byte*: Deletes all tag names for the selected byte and its bits and sets the default value for all bits of this byte to Low.
- *Undo the last action*
- *Redo the last action*

How to edit the tag names:

- Click on a byte in the EFI 1 or EFI 2 area to display its bits in the lower half of the window.
- Change the displayed tag names as desired. The bits will appear in the logic editor in the **Outputs** tab with their new tag names.

- Some bits are reserved and can not be used or edited. These bits are displayed grey in the upper half of the window.
- You can also delete tag names. Bits that have no tag name can not be used. They will be displayed grey in the upper half of the window and they are not displayed in the samos®NET process image (see also section 7.4.2 “samos®NET system: Process image” on page 63).

NOTES

How to set the default values:

- Click on a byte in the EFI 1 or EFI 2 area to display its content in the lower half of the window.
- Now click once on the icon to the right of the tag name field of any bit to toggle this bit's default value between 0 (Low) and 1 (High).

The set default value will be used in the samos®NET system's process image if the corresponding station has been suspended (see section 7.4.6 “samos®NET stations: Teach function” on page 67).



Fig. 72: Default values for samos®NET input bits

Changing the default value of any bit changes the process image of the configuration and in consequence the samos®NET ID for the EFI string the changed bit belongs to. You will be warned that you have to transfer the changed configuration to **all** stations in order to apply the new samos®NET ID. Otherwise the communication in the samos®NET network will be disrupted due to a samos®NET ID mismatch (see also section 7.2.2 “samos®NET ID” on page 50 and section 7.5 “samos®NET troubleshooting” on page 70).

NOTE

7.4.6 samos®NET stations: Teach function

The Teach function allows you to keep a samos®NET system operable and running even if one or more stations in the system are missing (i.e. switched off). “Teaching” the missing station or stations will suspend them such that the other stations will simulate their existence. Each suspended station will be treated as if it were online and running. The samos®NET process image will contain the values that have been configured as default values for this station (see section 7.4.5 “samos®NET stations: Station X view and process image” on page 66). This can be useful e.g. while setting up a system or for maintenance purposes.

If the Teach function is active on any station and this station is connected to the system and in the Run state, it will trigger the complete system to perform a network scan and to treat all missing stations as suspended. I.e. the system will function as if these stations were still online and use their default process images.



Before you use the Teach function, check whether a dangerous state can occur!

If the Teach function is used, the safety outputs on any of the still active stations may be High.

- ➔ Analyze your application and check whether additional safety measures have to be implemented if the Teach function is enabled.
- ➔ Consider how to handle the disconnected machine modules. Point out that the control parts and sensors have no effect to the previously connected machine modules (e.g. install “out of order” signs at E-stops).
- ➔ The teach function must be considered as a configuration process. Therefore the teach function needs to be secured according to the respective safety requirements, e.g. by using a key operated switch that is wired to the logic teach input and a restart function block in the logic to check the timing condition.
- ➔ Only authorized and especially trained personnel are allowed to activate the Teach function.
- ➔ Before using the Teach function, make sure that nobody is in the hazardous area or has access to the hazardous area while the Teach function is active.

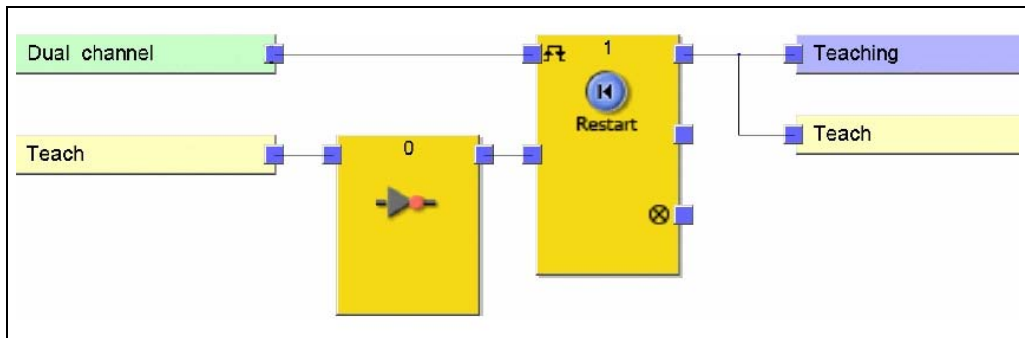
NOTES

- A station is considered as “missing” and can be suspended if its power supply is switched off or if its EFI connection to the samos®NET system has been interrupted completely. It is not possible to suspend a station if it is still connected and e.g. one of the following conditions applies:
 - The station is not in the Run state.
 - The station has generated an EFI error, e.g. due to a samos®NET ID mismatch.
- Teaching always affects all powered stations in the samos®NET system, not a single station. Therefore it may be sufficient to activate the Teach function just on one of the connected stations. However, if only one station in the system is equipped with a teach button and configured for teaching, then only the other stations in the system can be suspended since this particular station is needed for activating the Teach function.
- Each CPU in a samos®NET system signals the current system status via status bits that can be used as inputs in the logic editor (see section 7.4.7 “samos®NET teaching status and diagnostics” on page 69).

How to configure Teaching:

- ➔ Connect a Teach button to the inputs of each station in the samos®NET system that shall be able to activate the Teach function. The Teach button can be e.g. a dual channel key operated switch.
- ➔ In the logic editor for these stations, use a Restart function block to connect the Teach button input to this station’s Teaching output as shown in Fig. 73.

Fig. 73: Configuration of the Teach function in the logic editor



If the Teach button is pressed, the Teaching output will become High for one logic cycle. The rising edge (Low to High) on the Teaching output triggers the Teach function.



Ensure that the transitions of the signals for Teaching fulfil the requirements!

In case of a short-circuit to High (to 24 V DC) at a physical input, the evaluated signal can have a pulse when the signal is reset due to the short-circuit detection. If such a pulse can result in a dangerous state in the machine, the following points have to be observed:

- Ensure protected cable laying for the signal line (due to cross-circuiting to other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

How to use Teaching:

- ➔ In a running samos®NET system, switch off the power supply of one or more stations (e.g. Station C). The system will detect that these stations are missing and set their process image to failsafe values (zeros). The remaining stations will show an EFI error (EFI LEDs blinking red) and their EFI status bits for the switched off station(s) (e.g. **Station C missing**) will become High and **Station missing** will become High as well.
- ➔ Now press the Teach button on any of the remaining stations. The system will now continue to operate as if the missing stations were still present. Their “real” process image will however be replaced with the static default values that you have configured before (see section 7.4.5 “samos®NET stations: Station X view and process image” on page 66). The EFI status bits of the remaining stations will now show which stations have been suspended (e.g. **Station C missing** and **Station missing** will become Low again and **Station C suspended** becomes High).
- ➔ To bring a missing and consequently suspended station back into operation, reconnect its power supply. As soon as the station has finished powering up, the other stations will detect its presence and show an EFI error. The EFI status bit **Station C suspended** remains High while the system status bit **Suspended station found** becomes High.
- ➔ Now press the Teach button again. The system will reintegrate the suspended station and continue operation. The **Suspended station found** system status bit will become Low as well as the respective EFI status bits (e.g. **Station C missing** remains Low and **Station C suspended** also becomes Low).

If a station is missing not because its power supply has been switched off but due to an interruption of its EFI connection, it will most likely have gone into an error state. In this case you will have to reset this station by interrupting its power supply for at least 3 s before it can be reintegrated into the system.

NOTE

7.4.7 samos®NET teaching status and diagnostics

Each CPU in a samos®NET system signals via status bits whether Teaching is required and which station is missing or is suspended (= has been taught). These status bits are available as inputs of the respective CPU in the logic editor.

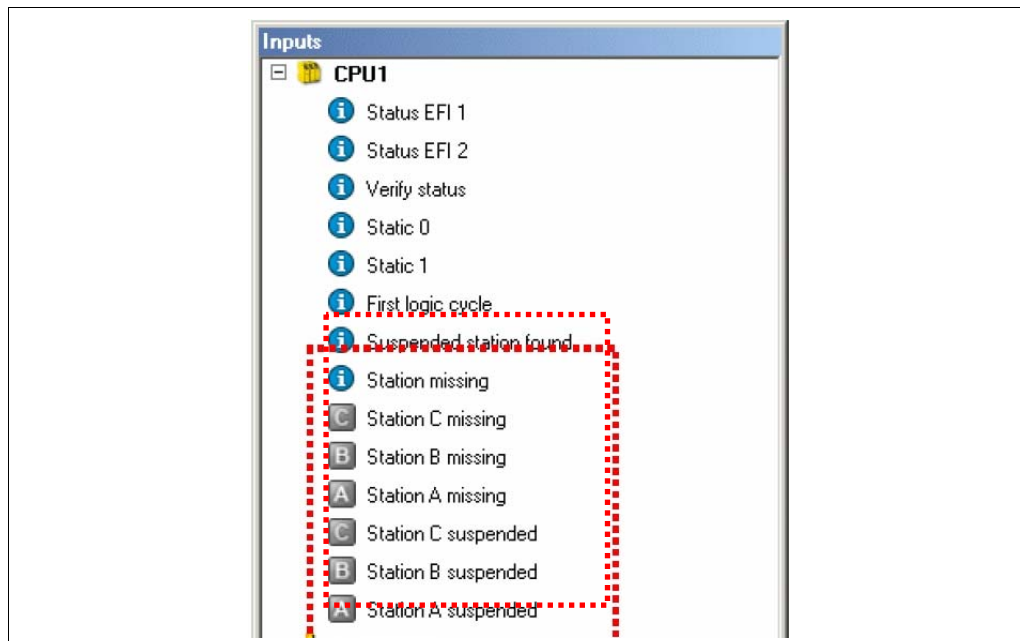


Fig. 74: samos®NET system status information in the logic editor

Tab. 15: Meaning of the Teaching status bits

CPU status bit	Meaning
Suspended station found	A station that has been suspended before, has reappeared in the system. In this case the process image of the corresponding samos®NET station is set to Low and the EFI I/O error bit is set to High. To continue operation, Teaching is required.
Station missing	At least one station in the system is missing. To continue operation, Teaching is required. This means that at least one of the Station X missing status bits (see below) is High as well.
Station X missing	The station with the samos®NET address X (= A, B, C or D) is missing. In this case the process image of the corresponding samos®NET station is set to Low and the EFI I/O error bit is set to High. This means that the Station missing status bit (see above) is High as well.
Station X suspended	The station with the samos®NET address X (= A, B, C or D) is suspended. In this case the default process image of the corresponding samos®NET station is used.

Using these status bits you can set up your own diagnostic system, e.g. by connecting these status bits to a Message generator function block or by switching on a warning lamp if Teaching is required or active.

NOTE

After the transition from the Stop state to the Run state, a station is considered as missing if it has not been found within 3 minutes.
For a description of the other CPU status bits see section 5.6.3 "CPU status bits in the logic editor" on page 32.

7.5 samos®NET troubleshooting

This chapter deals with some common causes for malfunctions of the samos®NET network and how to diagnose and correct them.

For an overview of the LED error displays please refer to the samos®PRO hardware operating instructions (Wieland Electric document no. BA000497).

7.5.1 samos®NET ID mismatch

Error description

If no process image transfer is possible between the stations in your system and all controller modules (SP-SCON-NET) show a recoverable error (MS LED flashing **Red** at 1 Hz and EFI 1 and EFI 2 LEDs light up **Red**), this could be due to a samos®NET ID mismatch. This means that at least one of the stations in the system has one or two samos®NET IDs that differ from the other stations' samos®NET IDs.

Diagnostics

- ➔ Switch to the samos®NET **System overview**.
- ➔ If the samos®PLAN is not connected to the system, connect to all stations.
- ➔ Check the status messages of all stations whether a samos®NET ID mismatch has been detected.

Correction of the error

If different samos®NET IDs exist in the system, the current configuration must be transferred to all stations again.

- ➔ Check whether the configuration in the samos®PLAN is correct.
- ➔ Connect to all stations.
- ➔ Transfer the configuration to all stations.
- ➔ Switch to the view for each station and verify the configuration in the device if necessary.

8 Logic programming – Function blocks

8.1 General safety notes for the logic programming

The function logic of the samos[®]PRO system is programmed by using function blocks. These function blocks are certified for use in safety-relevant functions if all safety standards are observed during implementation. The following sections provide information on important aspects of using function blocks in the samos[®]PRO system.

Observe the relevant safety standards and regulations!

All safety related parts of the installation (cabling, connected sensors and actuators, configuration settings) must be according to the relevant safety standards (e.g. IEC 62061 or EN ISO 13849-1) and regulations. Only safety-relevant signals may be used in safety-relevant logic. Ensure that the application fulfils all applicable standards and regulations!

You are responsible for checking that the right signal sources are used for these function blocks and that the entire implementation of the safety logic fulfils the applicable standards and regulations. Always check the mode of operation of the samos[®]PRO hardware and of the logic program in order to ensure that these behave in accordance with your risk avoidance strategy.

Take additional safety measures if the safe value may lead to a dangerous condition!

The safe value of process data and outputs is Low, which is applied if an error is detected. If the safe value (signal = Low) may lead to a dangerous condition in the application, additional measures must be taken, e.g. evaluation of the status of the process data and switching off the related output signals if the status evaluation detects an error. This needs especially to be considered for inputs with edge detections.

Take unexpected rising or falling edges into account!

Special attention is required for applications where an unexpected rising or falling edge at an input with edge detection may lead to a dangerous situation. An error on an input can generate such edges (e.g. network or EFI communication interruption, cable interruption at digital input, short circuit at digital input connected to test output). The safe value is applied until the error reset condition is fulfilled. Due to this the related signal can behave as follows:

- It changes temporarily to High, instead of remaining Low (rising edge and falling edge, i.e. transition from Low to High to Low), or
- it changes temporarily to Low, instead of remaining High as under faultless condition (falling edge and rising edge, i.e. transition from High to Low to High), or
- it remains Low, instead of changing to High as under faultless condition.

Take delays into account that are caused by CPU markers and jump addresses with logic loop backs!

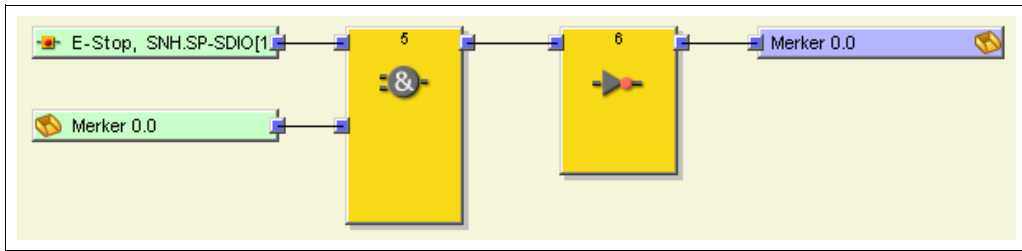
A loop back signal is an input signal that is connected to an output of a function block with the same or higher function block index (the function block index is displayed at the top of each function block). Therefore the input uses the output value of the previous logic cycle. This must be considered for the functionality and especially for the response time calculation.

To connect a loop back signal a jump address or a CPU marker must be used. A CPU marker generally causes a delay of one logic cycle.



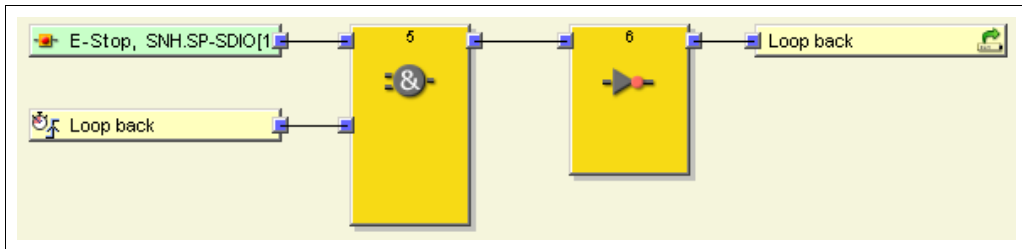
Logic programming – Function blocks

Fig. 75: CPU marker



A jump address causes a delay of one logic cycle if it constitutes a loop back. If this is the case, the input of the jump address is displayed with a clock symbol (with samos®PLAN V1.3.0 or higher).

Fig. 76: Jump address with loop back



8.2 Function block overview

The samos®PRO system uses function blocks to define the safety-oriented logic. A configuration can encompass a maximum of 255 function blocks. There are logic function blocks and application-specific function blocks. The following table lists all function blocks available for SP-SCON and SP-SCON-NET:

Tab. 16: Function block overview

Logic	
<ul style="list-style-type: none"> • AND • OR • XOR (exclusive OR) • XNOR (exclusive NOR) • NOT • RS Flip-Flop 	<ul style="list-style-type: none"> • JK Flip-Flop • Binary decoder • Binary encoder • Routing 1:N (signal duplication) • Routing N:N (N inputs to N outputs parallel)
Start/Edge	
<ul style="list-style-type: none"> • Reset • Restart 	<ul style="list-style-type: none"> • Start warning • Edge detection
Delays	
<ul style="list-style-type: none"> • On-delay timer • Off-delay timer 	<ul style="list-style-type: none"> • Adjustable on-delay timer • Adjustable off-delay timer
Counter and cycle	
<ul style="list-style-type: none"> • Event counter (up, down, up and down) • Clock generator 	<ul style="list-style-type: none"> • Ramp down detection • Message generator
EDM/Output blocks	
<ul style="list-style-type: none"> • EDM • Valve monitoring 	<ul style="list-style-type: none"> • Fast shut off with bypass • Fast shut off
Muting/Press	
<ul style="list-style-type: none"> • Sequential muting • Parallel muting • Cross muting • Universal press contact • Press single stroke 	<ul style="list-style-type: none"> • Press setup • Press automatic • N-break (PSDI — Press with N-PSDI mode) • Eccentric press contact
Other	
<ul style="list-style-type: none"> • User mode switch • Emergency stop 	<ul style="list-style-type: none"> • Two hand control type IIIA • Two hand control type IIIC

<ul style="list-style-type: none"> • Safety gate monitoring • Magnetic switch • Light curtain monitoring 	<ul style="list-style-type: none"> • Multi operator • Switch synchronization • Error output combination
User defined function blocks	
<ul style="list-style-type: none"> • Grouped function block 	<ul style="list-style-type: none"> • Customized function block

The samos®PRO system supports up to 255 function blocks in a specific application. The response time is influenced by the number of function blocks. Therefore, the number of function blocks in your application should be kept as low as possible.

8.3 Function block properties

Function blocks offer a number of different properties that you can use. The configurable parameters differ depending on the function block. You can double-click the function block to access the configurable parameters and select the tab with the desired properties. The following example shows the Safety gate monitoring function block:

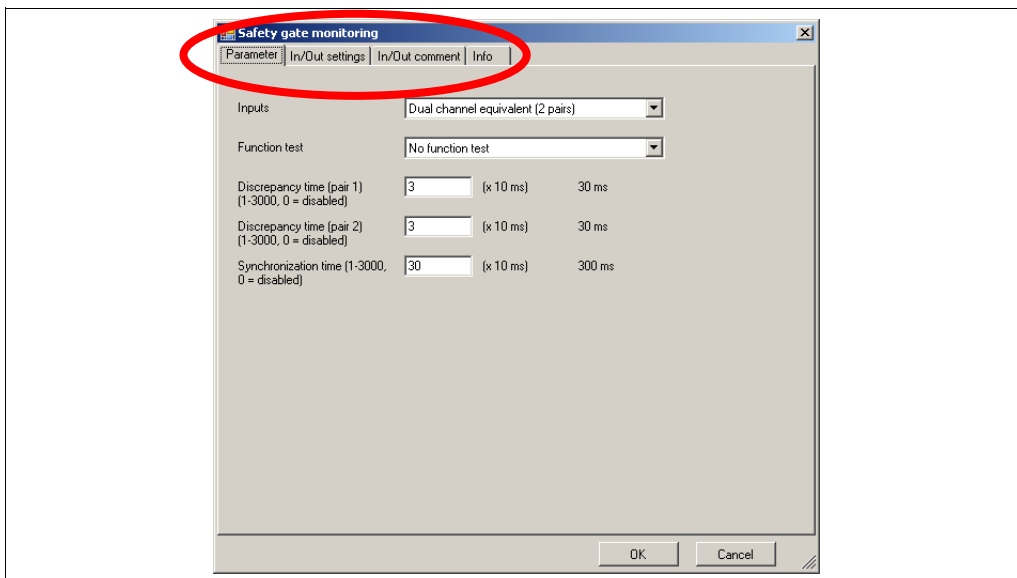


Fig. 77: Configurable parameters of function blocks

On the **Parameter** tab and on the **In/Out settings** tab you will find the configurable parameters depending on the function block. The **In/Out comment** tab allows you to replace the standard I/O descriptions of the function block with your own names and to add a name or a descriptive text to the function block that will be displayed under the function block in the logic editor. Under **Info** you will find a description of the function block and its parameters.

8.4 Input and output signal connections of function blocks

8.4.1 Function block input connections

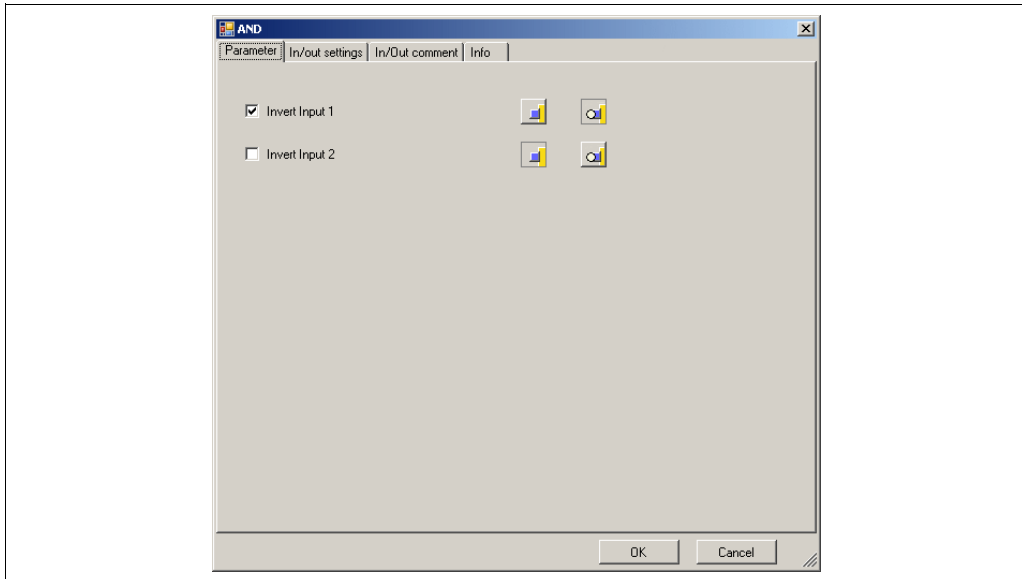
Possible sources for function block inputs are all input elements listed in the input selection tree of the logic editor as well as the outputs of function blocks.

8.4.2 Inversion of input connections

The input connections of some function blocks can be configured as *inverted*. This means that the function block evaluates a High signal at an inverted input as Low and vice versa. In order to invert an input, double click on the function block icon and check the desired input on the parameter page of the function block properties dialog:

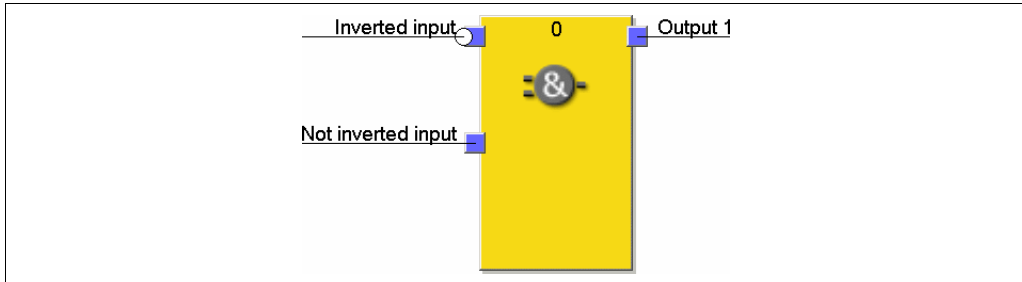
Logic programming – Function blocks

Fig. 78: Inversion of function block input connections



Inverted inputs are displayed with a small white circle:

Fig. 79: Example of an AND function block with one inverted input



Examples of function blocks with invertable inputs include the following function blocks:

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

8.4.3 Function block output connections

Function blocks provide various output signal connections for connecting to physical outputs or to other function blocks.

The output of a function block can be connected to several subordinate function blocks, but not to several output elements (physical outputs or EFI outputs). If you want to control several physical outputs with a single function block, use the **Routing 1:N** function block. The behaviour of the outputs is explained at the description of the individual function blocks.

You can choose whether error and diagnostics outputs are displayed. In the configuration basic setting of the function blocks only the Enable output and some further outputs are selected (e.g. Reset required). In order to display error and diagnostics outputs increase the number of outputs on the In/Out settings tab of the function block properties.

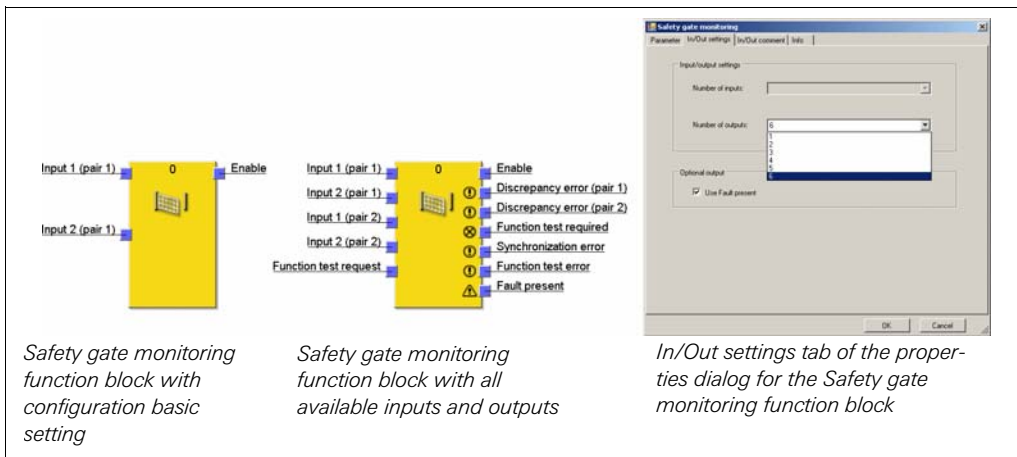


Fig. 80: In/Out configuration of the Safety gate monitoring function block

8.5 Parametrization of function blocks

In addition to the type of input (e.g. single-channel, dual-channel equivalent, etc.), function blocks can have further parameters that are defined on the properties page of the function block shown above.

8.5.1 Timer values and logic execution time

The following has to be observed when selecting time monitoring functions for the discrepancy time, synchronization time, pulse duration, muting time, etc.: The times

- have to be greater than the logic execution time
- have a precision of ± 10 ms in the evaluation in addition to the logic execution time

The logic execution time depends on the number and type of the function blocks used. It is a multiple of 4 ms. If the used logic execution time exceeds 100 %, then the logic execution time is increased by 4 ms. The logic execution time is displayed in the logic editor in the **FB preview** window. It has a precision of ± 100 ppm (parts per million).

NOTE

8.5.2 Fault present output

Various function blocks dispose of the Fault present diagnostics output. In order to use it, activate the **Use Fault present** checkbox on the In/Out settings tab of the function block properties. The additional output "Fault present" is now displayed in the function block.

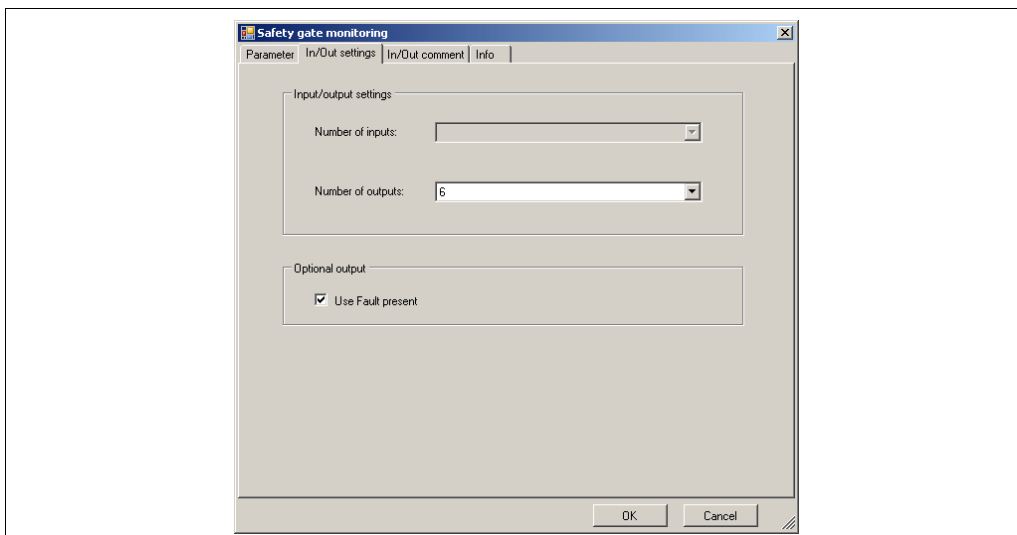
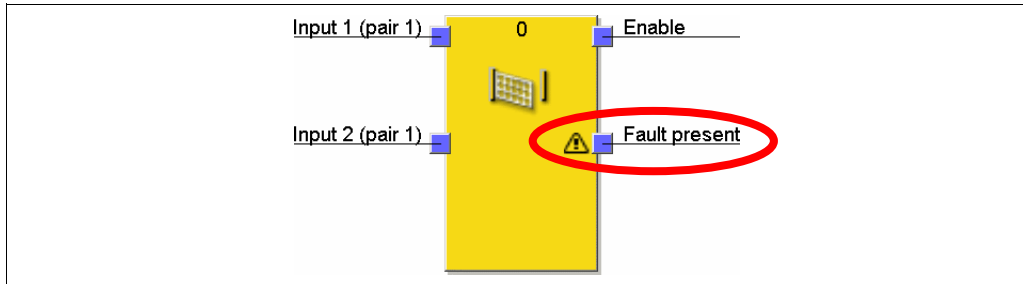


Fig. 81: Activating the Fault present output

Logic programming – Function blocks

Fig. 82: Fault present output



The Fault present output changes to High when an error has been detected on the basis of the configured function block parameters (e.g. discrepancy time error, function test error, synchronization error, etc.). When the Fault present output is High, the main output (e.g. the Enable output) changes to Low.

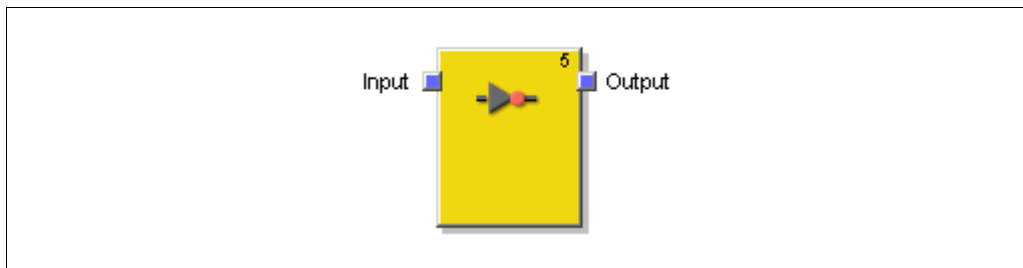
The Fault present output is set to Low if all errors are reset. The conditions to reset an error are described in the section of the respective function block.

8.6 Logic function blocks

8.6.1 NOT

Function block diagram

Fig. 83: Function block diagram for the NOT function block



General description

The inverted input value applies at the output. If, for example, the input is High, the output is Low.

Truth table

The following applies for the truth table in this section:

“0” means logic Low

“1” means logic High

Truth table for NOT

Input	Output
0	1
1	0

Tab. 17: Truth table for the NOT function block

8.6.2 AND

Function block diagram

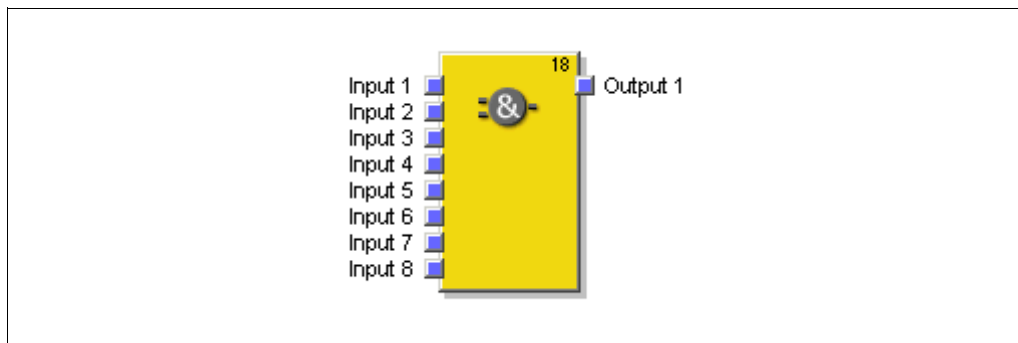


Fig. 84: Function block diagram for the AND function block

General description

The output is High if all evaluated inputs are High. Up to eight inputs are evaluated.

Example: If eight emergency stop buttons are attached to the inputs of the function block, the output would become Low as soon as one of the emergency stop buttons is pressed.

Parameters of the function block

Parameter	Possible values
Number of inputs	2 to 8
Invert input x	Each input of this function block can be inverted (see section 8.4.2 "Inversion of input connections" on page 73).

Tab. 18: Parameters of the AND function block

Truth table

The following applies for the truth tables in this section:

"0" means logic Low

"1" means logic High

"x" means "any" = "0" or "1"

Truth table for AND evaluation with one input

Input 1	Output
0	0
1	1

Tab. 19: Truth table for AND evaluation with one input

Truth table for AND evaluation with two inputs

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

Tab. 20: Truth table for AND evaluation with two inputs

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

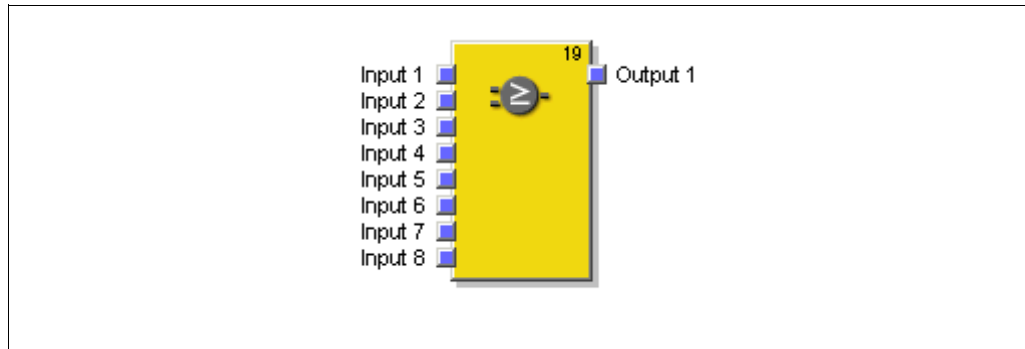
Tab. 21: Truth table for AND evaluation with eight inputs

Logic programming – Function blocks

8.6.3 OR

Function block diagram

Fig. 85: Function block diagram for the OR function block



General description

The output is High if any one of the evaluated inputs is High. Up to eight inputs are evaluated.

Example: If eight light curtains are attached to the inputs of the function block, the output would become High as soon as at least one of the light curtains is free.

Parameters of the function block

Tab. 22: Parameters of the OR function block

Parameter	Possible values
Number of inputs	2 to 8
Invert input x	Each input of this function block can be inverted (see section 8.4.2 "Inversion of input connections" on page 73).

Truth table

The following applies for the truth tables in this section:

"0" means logic Low

"1" means logic High

"x" means "any" = "0" or "1"

Truth table for OR evaluation with one input

Tab. 23: Truth table for OR evaluation with one input

Input 1	Output
0	0
1	1

Truth table for OR evaluation with two inputs

Tab. 24: Truth table for OR evaluation with two inputs

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

Truth table for OR evaluation with eight inputs

Tab. 25: 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

8.6.4 XOR (exclusive OR)

Function block diagram

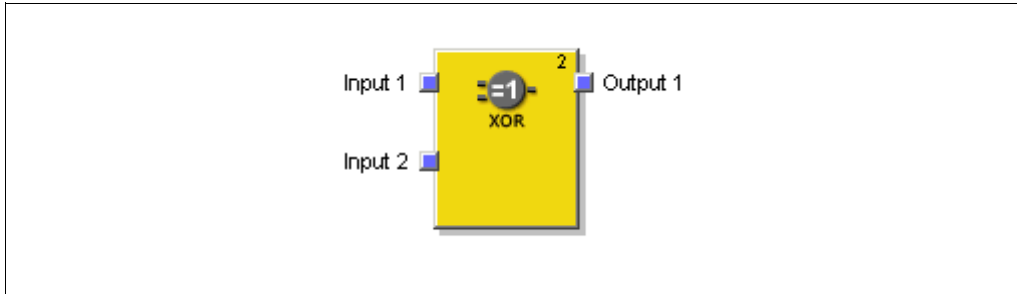


Fig. 86: Function block diagram for the Exclusive OR (XOR) function block

General description

The output is High only if both inputs are complementary (i.e. with contrary values: one input High and one input Low).

Truth table

The truth table uses the following designations:

"0" means logic Low

"1" means logic High

Truth table for XOR evaluation

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

Tab. 26: Truth table for XOR evaluation

8.6.5 XNOR (exclusive NOR)

Function block diagram

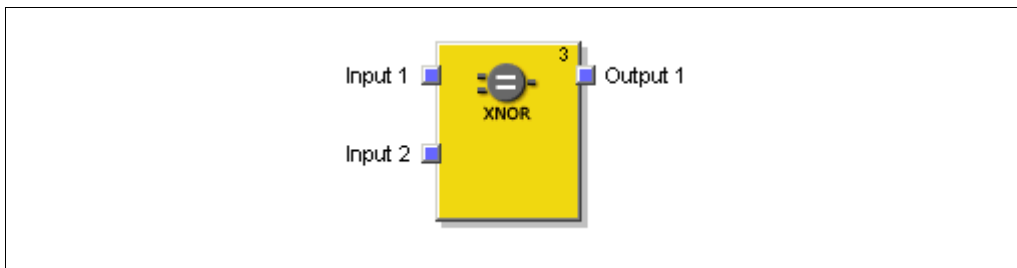


Fig. 87: Function block diagram for the Exclusive NOR (XNOR) function block

General description

The output is High only if both inputs are equivalent (have the same value: both inputs High or both inputs Low).

Truth table

The truth table uses the following designations:

"0" means logic Low

"1" means logic High

Truth table for XNOR evaluation

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

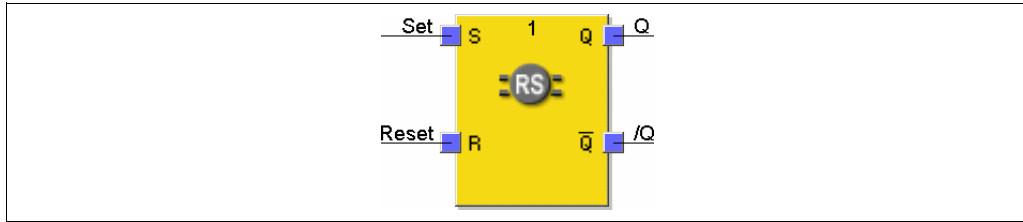
Tab. 27: Truth table for XNOR evaluation

Logic programming – Function blocks

8.6.6 RS Flip-Flop

Function block diagram

Fig. 88: Logic connections for the RS Flip-Flop function block



General description

The RS Flip-Flop function block stores the last value of the inputs **Set** or **Reset**. It is used as a simple storage cell. The **Reset** signal has a higher priority than the **Set** signal. If **Set** was High last, output **Q** is High and output **/Q** (Q inverted) is Low. If the **Reset** input was High last, output **Q** is Low and output **/Q** is High.

Parameters of the function block

Tab. 28: Parameters of the RS Flip-Flop function block

Parameter	Possible values
Invert Set	Each input of this function block can be inverted (see section 8.4.2 "Inversion of input connections" on page 73).
Invert Reset	

Truth table for the RS Flip-Flop function block

The following applies for the truth table in this section:

- "0" means logic Low
- "1" means logic High
- "n-1" references the preceding value
- "n" references the current value
- "x" means "any" = "0" or "1"

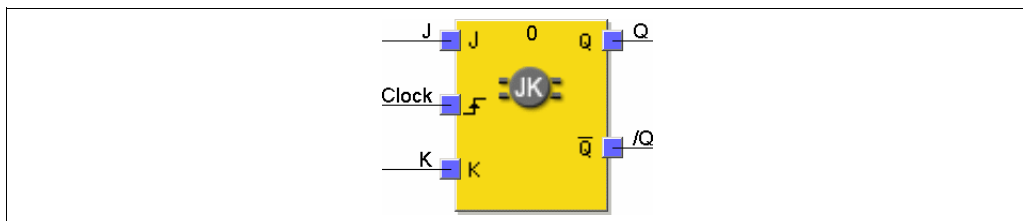
Tab. 29: Truth table for the RS Flip-Flop function block

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

8.6.7 JK Flip-Flop

Function block diagram

Fig. 89: Logic connections for the JK Flip-Flop function block



General description

The JK Flip-Flop function block has three inputs. The J and K inputs have only an effect on the outputs when a *rising edge* is detected at the Clock input. In this case ...

- If input **J** is High and input **K** is Low, the **Q** output will be High and the **/Q** (= Q inverted) output will be Low
- If input **J** is Low and input **K** is High, the **Q** output will be Low and the **/Q** output will be High
- If both inputs are Low, the last values of outputs **Q** and **/Q** will be held.

- If both inputs are High, the outputs will toggle, i.e. their last values will be inverted.

Parameters of the function block

Parameter	Possible values
Number of outputs	<ul style="list-style-type: none"> • 1 (Q) • 2 (Q and /Q)
Invert J	Each input of this function block can be inverted (see section 8.4.2 "Inversion of input connections" on page 73).
Invert Clock	
Invert K	

Tab. 30: Parameters of the JK Flip-Flop function block

Truth table for the JK Flip-Flop function block

The following applies for the truth table in this section:

- "0" means logic Low
- "1" means logic High
- " " means that a rising edge has been detected at the input
- " " means that a falling edge has been detected at the input
- "n-1" references the preceding value
- "n" references the current value
- "x" means "any" = "0" or "1"

The following truth table is valid for a configuration of the JK Flip-Flop function block with no input inverted.

NOTE

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

Tab. 31: Truth table for the JK Flip-Flop function block

8.6.8 Clock generator

Function block diagram

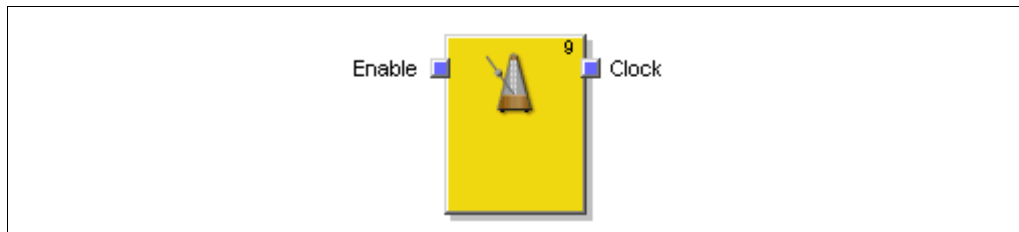


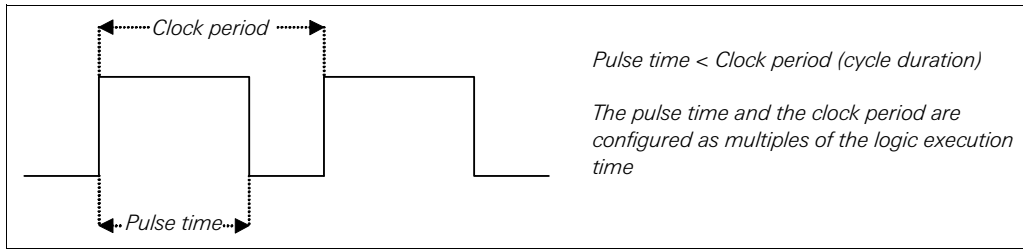
Fig. 90: Logic connections for the Clock generator function block

General description

The Clock generator function block is used to generate a pulsed signal. When the **Enable** input is High, the **Clock** output pulses from Low to High and back to Low in accordance with the parameter settings of the function block. When the **Enable** input is Low the **Clock** output becomes Low.

Logic programming – Function blocks

Fig. 91: Parameter diagram for the Clock generator function block



Parameters of the function block

Tab. 32: Parameters of the Clock generator function block

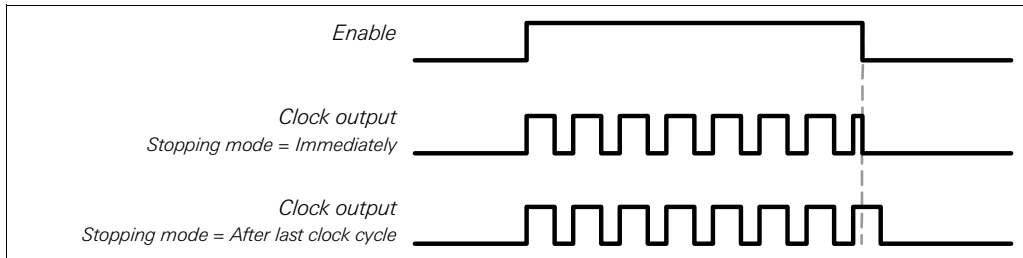
Parameter	Possible values
Stopping mode	<ul style="list-style-type: none"> Immediately After last clock
Clock period (cycle duration)	2 to 65·535 Duration = Parameter value × Logic execution time
Pulse time	1 to 65·534 Duration = Parameter value × Logic execution time The pulse time has to be lower than the clock period.

NOTE

If the logic execution time of the configuration changes (e.g. through addition or subtraction of function blocks), the clock period and the pulse time will change.

Sequence/timing diagram

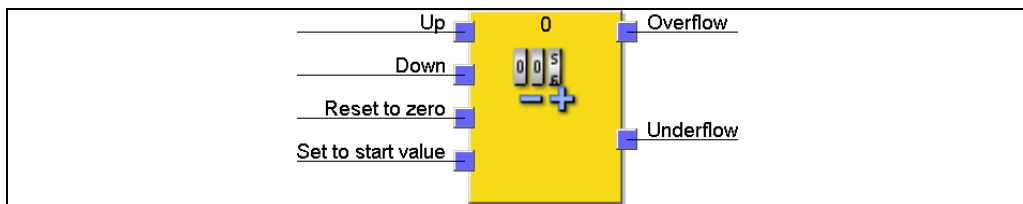
Fig. 92: Timing diagram for the Clock generator function block



8.6.9 Event counter (Up, Down and Up and down)

Function block diagram

Fig. 93: Logic connections for the Event counter (Up and down) function block



General description

The Event counter function blocks allow to count events, either upward and/or downward. When a pre-defined overflow value is reached, this will be indicated on the **Overflow** output or, when zero is reached, this will be indicated on the **Underflow** output. Depending on the required counting direction there are the function blocks Event counter (Up), Event counter (Down) and Event counter (Up and down).

Parameters of the function block

Tab. 33: Parameter for the Event counter function blocks

Parameter	Possible values
Reset to zero after overflow	<ul style="list-style-type: none"> Manual Automatic
Set to start value after underflow	<ul style="list-style-type: none"> Manual Automatic
Overflow value	Integer between 1 and 65,535. The overflow value has to be higher than or equal to the reload value.

Parameter	Possible values
Start value	Integer between 1 and 65,535
Min. pulse time for resetting to zero	<ul style="list-style-type: none"> • 100 ms • 350 ms
Min. pulse time for setting to start value	<ul style="list-style-type: none"> • 100 ms • 350 ms

Inputs Up and Down

A rising edge (Low to High) at the **Up** input increases the value of the internal counter by "1".

A rising edge (Low to High) at the **Down** input decreases the value of the internal counter by "1".

If a rising edge (Low to High) occurs at the **Up** input as well as at the **Down** input (applies only to the Up and down event counter function block), the value of the internal counter remains unchanged.

Reset to zero

A valid pulse sequence with a Low-High-Low transition at the **Reset to zero** input sets the internal counter to "0". This happens irrespective of whether the **Overflow value** has been reached or not and also irrespective of whether **Reset to zero after overflow** has been configured to **Manual** or **Automatic**.

The **Min. pulse time for resetting to zero** defines the minimum required duration of the pulse at the **Reset to zero** input. Valid values are 100 ms and 350 ms. If the pulse duration is shorter than the configured minimum pulse time or longer than 30 s, the pulse is ignored.

Set to start value

A valid pulse sequence with a Low-High-Low transition at the **Set to start value** input sets the internal counter to the configured value of the **Start value** parameter. This happens irrespective of whether **Set to start value after underflow** has been configured to **Manual** or **Automatic**.

The **Min. pulse time for setting to start value** defines the minimum required duration of the pulse at the **Set to start value** input. Valid values are 100 ms and 350 ms. If the pulse duration is shorter than the configured minimum pulse time or longer than 30 s, the pulse is ignored.

Ensure that the transitions of the signals for Resetting to zero or for Setting to start value fulfil the requirements!

In case of a short-circuit to High (to 24 V DC) at a physical input, the evaluated signal can have a pulse when the signal is reset due to the short-circuit detection. If such a pulse can result in a dangerous state in the machine, the following points have to be observed:

- Ensure protected cable laying for the signal line (due to cross-circuiting to other signal lines).
- Do not reference for short-circuit detection, i.e. not test outputs.



Overflow limit and Reset to zero after overflow

The **Reset to zero after overflow** parameter determines what happens when the counter value reaches the **Overflow value**. If this parameter is configured to **Automatic**, and the internal counter equals the **Overflow value**, the **Overflow** output becomes High for the duration of the logic execution time. The value of the internal counter is subsequently reset to zero.

If the **Reset to zero after overflow** parameter is configured to **Manual** and the **Overflow value** has been reached, the **Overflow** output is set to High and remains High until the counter value changes again, either by counting downward, by a valid **Reset to zero** pulse sequence or by a valid pulse sequence at the **Set to start value** input, if the start value is smaller than the overflow value. Until then, all further "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 value reaches "0". If this parameter is configured to **Automatic** and the internal counter equals "0", the **Underflow** output becomes High for the duration of the logic execution time. The value of the internal counter is subsequently set to the configured **Start value**.

If the **Set to start value after underflow** parameter is configured to **Manual** and the lower limit, i.e. "0", has been reached, the **Underflow** output is set to High and remains High until the counter value changes again, either by counting upward or by a valid pulse sequence at the **Set to start value** input. Until then, all further "Down" counting pulses are ignored.

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

The following applies for the truth table in this section:

- "0" means logic Low
- "1" means logic High
- "↑" means that a rising edge has been detected at the input
- "↓" means that a falling edge has been detected at the input
- "n-1" references the preceding value
- "n" references the current value
- "Y" references the value of the internal counter
- "X" means "any". E.g. the **Reset to zero** input and the **Set to start value** input have priority over the the **Up** input and the **Down** input.

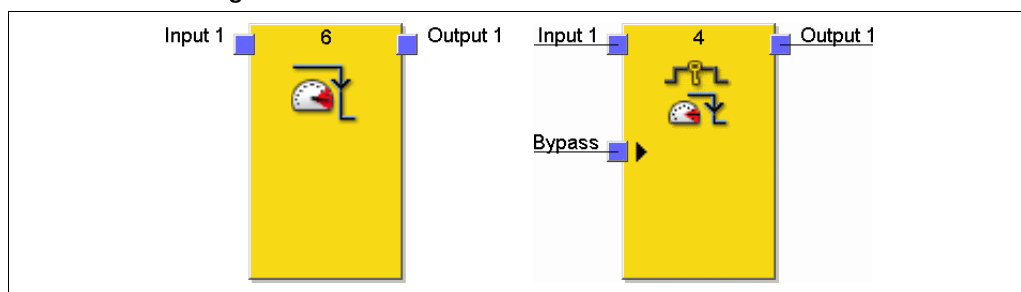
Tab. 34: Truth table for the Event counter function blocks (Up, Down and Up/Down)

Up	Down	Reset to zero	Set to start value	Counter value _{n-1}	Counter value _n	Over-flow _n	Under-flow _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

8.6.10 Fast shut off and Fast shut off with bypass

Function block diagram

Fig. 94: Logic connections for 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 samos®PRO system. To use one of these blocks, both the inputs and the output for the switching path must be connected to the same I/O module (i.e. SP-SDIO). This is necessary because the Fast shut off function

blocks generate a direct switch off at the extension module resulting in a shorter switch off time which is independent of the logic execution time.

For the Fast shut off function block, the consequence of this is that logic between the Fast shut off input and the Fast shut off output cannot hinder a switch off when the Fast shut off is activated.

The Fast shut off with bypass function block however allows to temporarily bypass the Fast shut off function using the bypass input.

The Fast shut off with bypass function block is only available with SP-SDIO modules with firmware version V2.00 or higher.

NOTE

Example: In the following logic example, the SLC will switch off the Q2 motor.

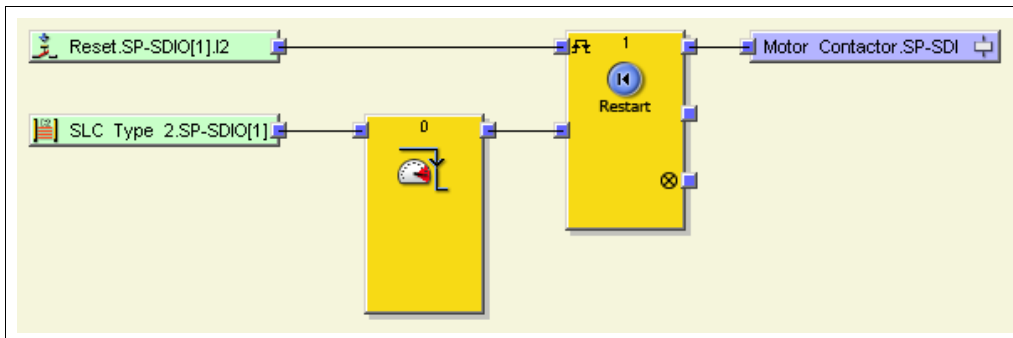


Fig. 95: Fast shut off example

Simple logic such as this can be accomplished within the Fast shut off function block itself (see how to configure below).

The signal path from the output of the Fast shut off function block to the physical output that is selected in the Fast shut off function block must be configured in such a way that switching off of the output of the Fast shut off function block always results in a direct switching off of the physical output as well. Typically the AND, Restart or EDM function blocks can be used in the signal chain for this. An OR function block however does not comply to this rule.

NOTE

Always consider the total response time of the entire safety function!

The response time of the Fast shut off function block is not the same as the total response time of the entire safety function. The total response time includes multiple parameters outside of this function block. For a description of how to calculate the total response time of the samos[®]PRO system please consult the “samos[®]PRO hardware” operating instructions.



Parameters of the function block

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	All outputs of the extension module whose inputs are connected to the function block, if the output is not already used for Fast shut off.

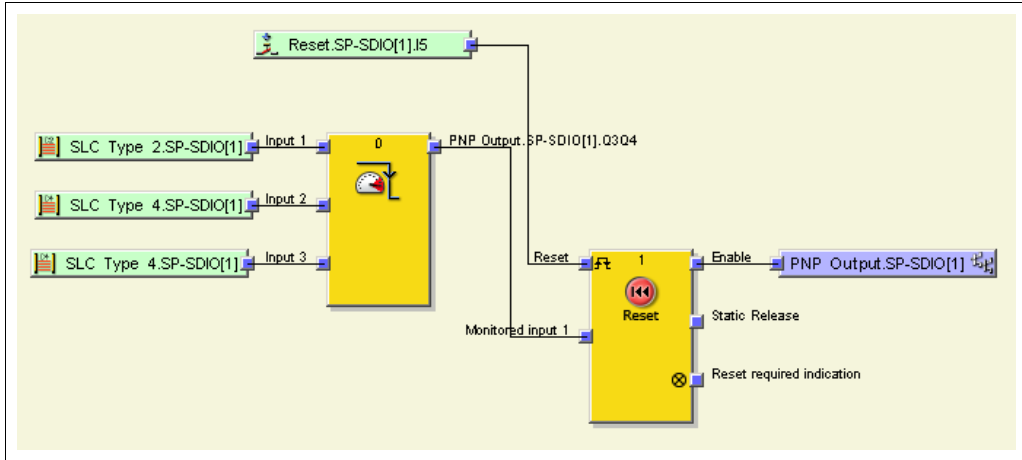
Tab. 35: Parameters of the Fast shut off function block

Logic programming – Function blocks

How to configure the Fast shut off function block:

The following example shows the function with three light curtains attached to a Fast shut off function block.

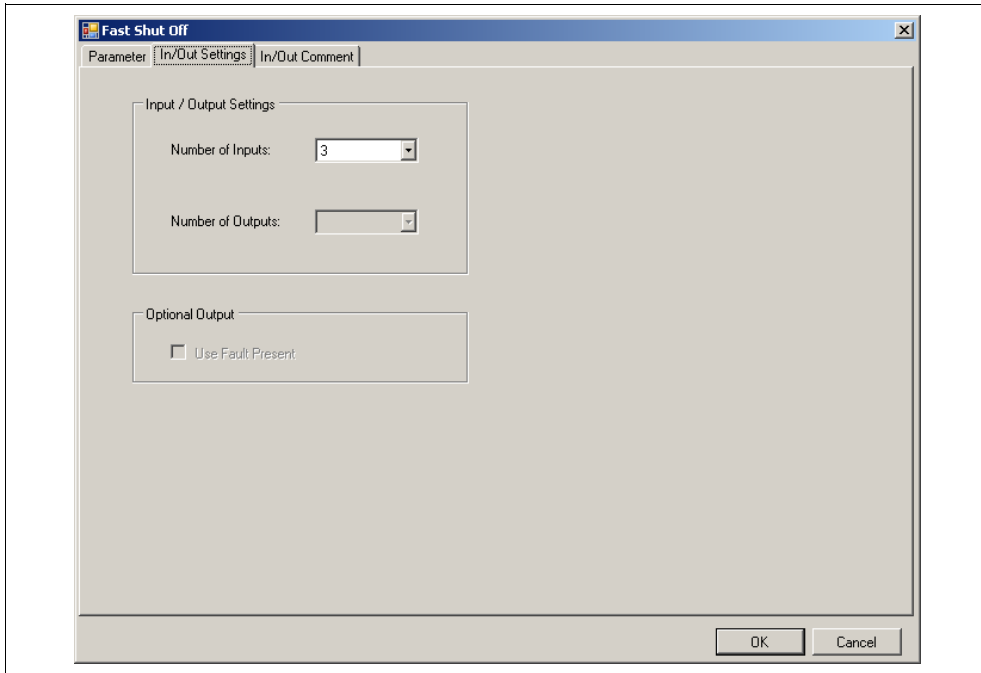
Fig. 96: Configuration example for Fast shut off with three light curtains



To configure the Fast shut off function block perform the following steps:

- ➔ Connect input elements to the function block. Double click the function block to open the configuration dialog and click the **In/Out Settings** tab.

Fig. 97: In/Out settings dialog for the Fast shut off function block



- ➔ Choose the number of inputs which you would like to attach to the function block.
- ➔ Then click the **Parameter** tab and choose the zone by checking the boxes.

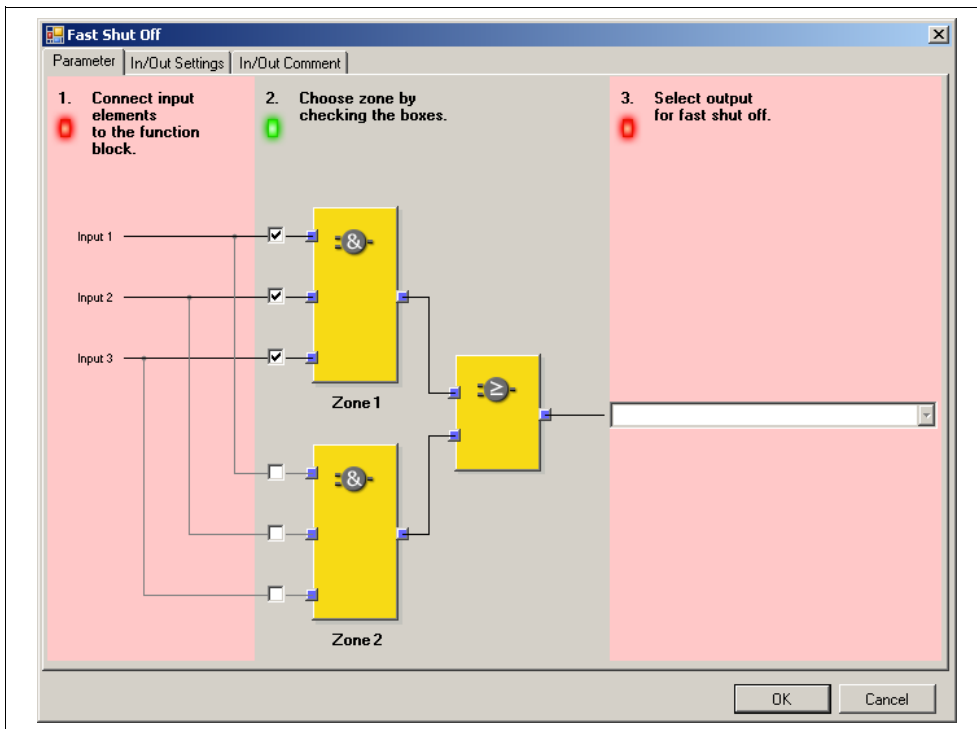


Fig. 98: Parameter settings dialog for the Fast shut off function block

If only AND logic is needed, leave the Zone 2 AND function block inputs unchecked. If additional OR logic is required in the application, the inputs can be combined using the Zone 1 and Zone 2 AND function blocks and subsequently connecting to the internal OR function block.

NOTE

➔ Finally select the output for Fast shut off.

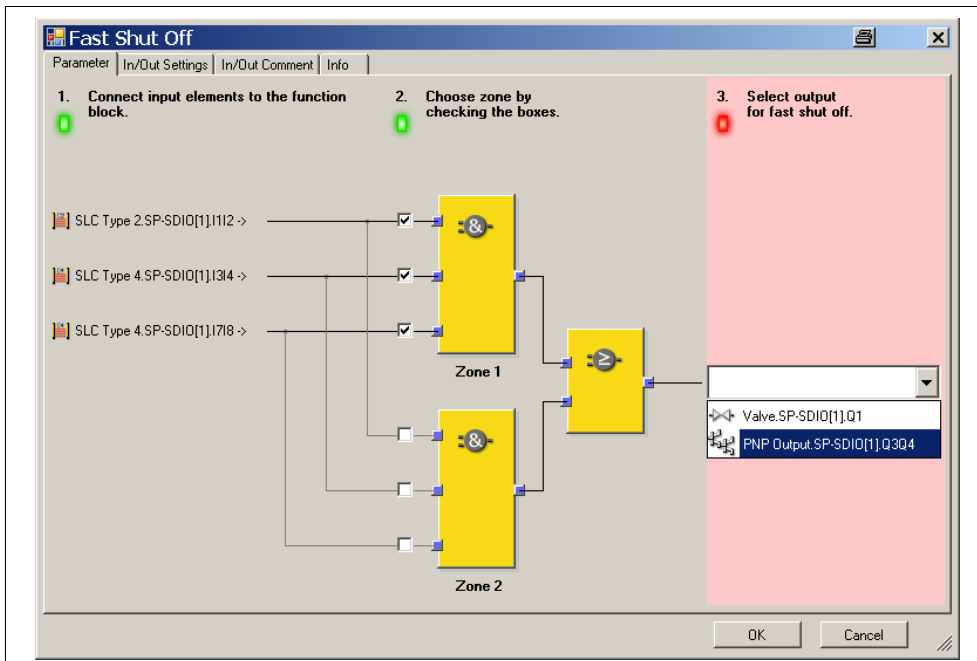
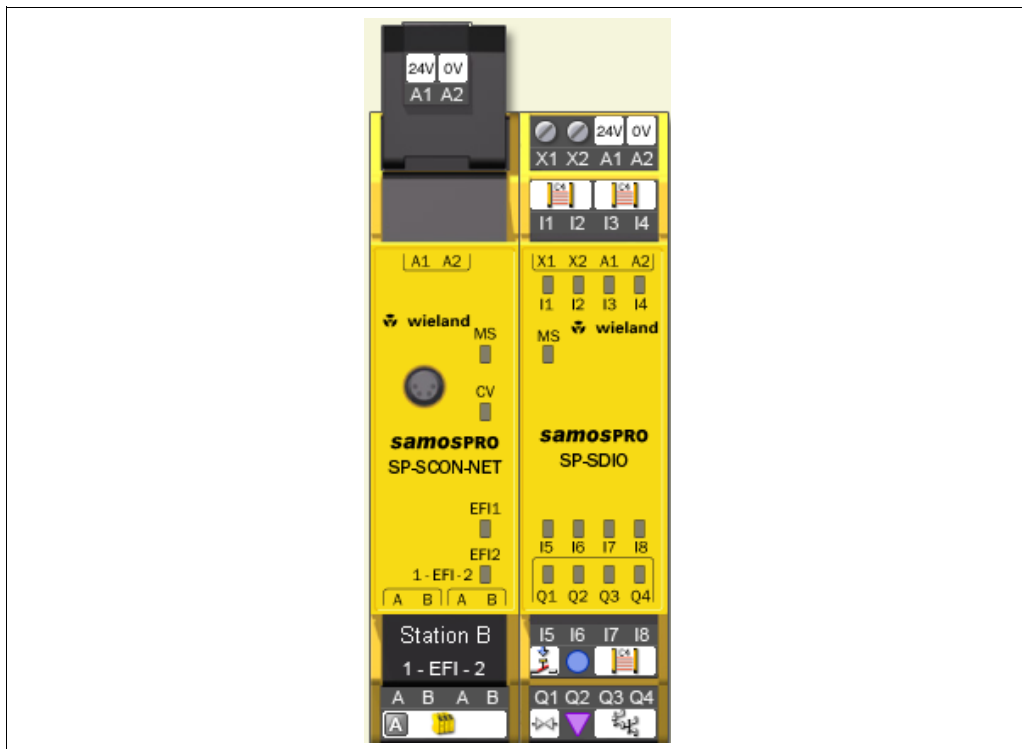


Fig. 99: Output selection for Fast shut off

At this point, the selected inputs and outputs are linked to each other such that the output cannot be moved to another position and the inputs must stay on the SP-SDIO module in the hardware configuration. The elements which are linked are shown in the hardware configuration in orange.

Logic programming – Function blocks

Fig. 100: Hardware configuration view of inputs and out-puts linked to Fast shut off



These links are broken when 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. This could be, e.g. in a safe machine setup mode, in which the machine can be operated only in jog mode. To this purpose, the Fast shut off with bypass function block is available. It is used and configured in 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 High, the Fast shut off with bypass function block is bypassed.



WARNING

Ensure the system or machine is in a safe condition when using the bypass function!

As long as the bypass function is active, any stopping condition, e.g. the infringement of a protective field will **not** lead to a shut-off of the machine. You must ensure that other protective measures are forcibly activated during the bypass, e.g. the safe machine setup mode, so that the machine cannot endanger persons or parts of the system during the bypass function.

Take the prolonged response time into account when the Bypass is deactivated!

If the **Bypass** input is being deactivated while a switching-off condition exists, the outputs will switch off only with the normal response time of the application. The minimized response time for Fast shut off does not apply for the **Bypass** input. Consider this for your risk analysis and avoidance strategy. Otherwise the operator of the machine will be in danger.

NOTES

- Unlike the other inputs and outputs of this function block, the **Bypass** input can be connected to an output of another function block as well as to any other input element which may also be moved to another module in the hardware configuration.
- The **Bypass** input has an on-delay of 3 logic cycles to compensate delays due to logic processing time and transmission time of the SBUS+. The delay ensures that the I/O module has received the bypass signal before it is used for the further logic processing in the Fast shut off function block. As a result of the delay, the **Bypass** input must be High 3 logic cycles in advance to successfully inhibit the fast shut off. If this condition is fulfilled, then the Fast shut off output of the function block and the physical output at the I/O module will both remain High.

- The Fast shut off directly switches off the connected output of the SP-SDIO module while subsequent logic will be ignored. Therefore it is not possible to implement further bypass conditions in the logic editor between the output of the Fast shut off function block and the connected SP-SDIO output.
- Be aware that in the logic online monitor the value of the connected SP-SDIO output may differ from the effective value at the physical output of the SP-SDIO module. For example the connected output can be Low as a result of the subsequent logic while the output of the Fast shut off function block and the physical output of the SP-SDIO module are High because the **Bypass** input is High.
- If your application requires that the output of the SP-SDIO module can be switched off independently of an existing bypass condition (e.g. emergency stop), then the underlying logic must be realised in a way that the respective shut-down signal (e.g. emergency stop) switches off the bypass signal for the function block as well, as shown in the following example:

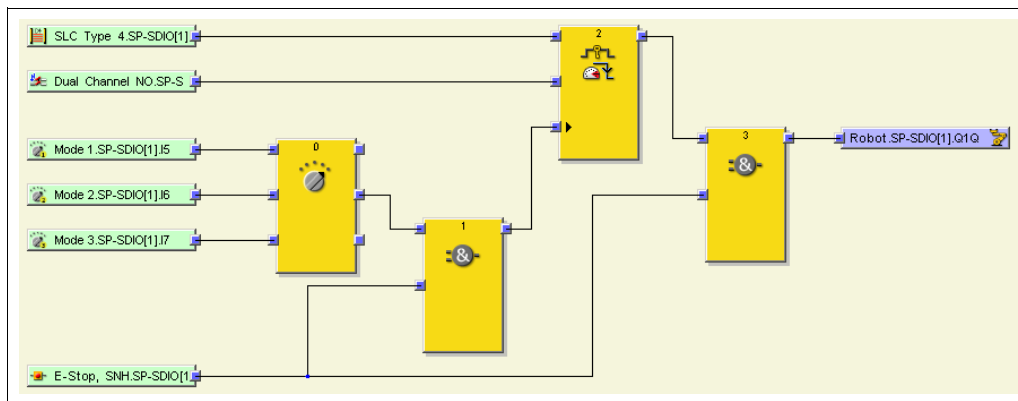


Fig. 101: Example for Fast shut off with bypass with more than one bypass conditions

8.6.11 Edge detection

Function block diagram

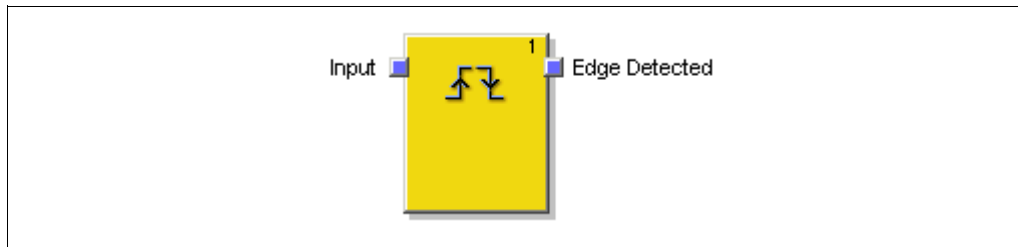


Fig. 102: Logic connections for the Edge detection function block

General description

The Edge detection function block is used to detect a positive (rising) or negative (falling) edge of the input signal. The function block can be configured to detect a positive edge, a negative edge or both. If an edge corresponding to the parameter settings is detected, the **Edge Detected** output changes to High for the duration of the logic execution time.

Parameters of the function block

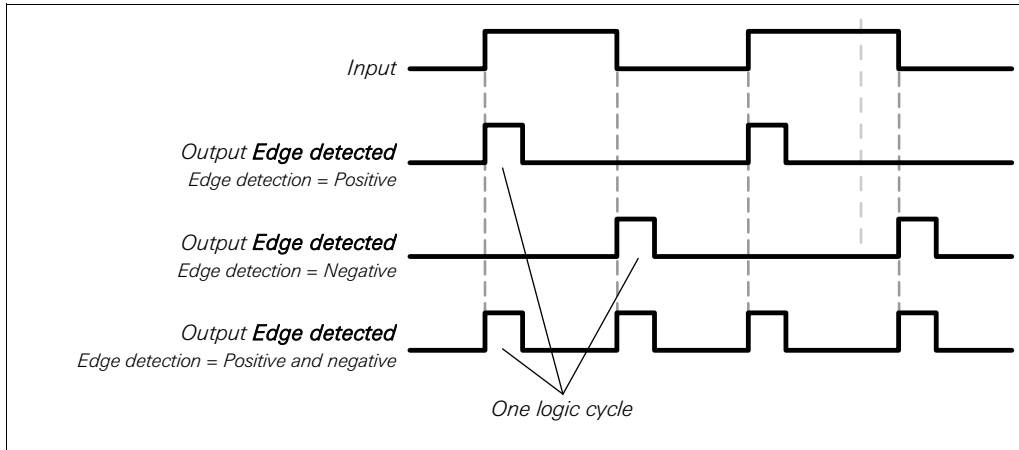
Parameter	Possible values
Edge detection	<ul style="list-style-type: none"> • Positive • Negative • Positive and negative

Tab. 36: Parameters of the Edge detection function block

Logic programming – Function blocks

Fig. 103: Timing diagram for the Edge detection function block

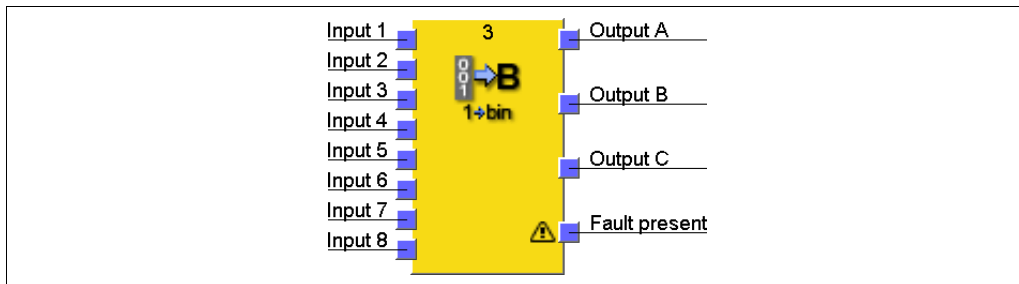
Sequence/timing diagram



8.6.12 Binary encoder

Fig. 104: Logic connections for the Binary encoder function block

Function block diagram



General description

The Binary encoder function block encodes depending on the current configuration a one-out-of-N (one-hot) or a priority code to a binary code (Output A = 2^0 , Output B = 2^1 , Output C = 2^2). 2 to 8 inputs can be configured. The number of outputs depends on the number of inputs. An optional **Fault present** output is available.

Parameters of the function block

Tab. 37: Parameters of the Binary encoder function block

Parameter	Possible values
Number of inputs	2 to 8
Encoder mode	<ul style="list-style-type: none"> One out of N Priority Priority-to-binary (Input 1 dominant)
Use Fault present	<ul style="list-style-type: none"> With Without

One out of N

In **One out of N** mode, only one input may be High at the same time. The outputs are set dependent on the index (input 1 = 1, input 2 = 2, ...) of the High input. If all inputs are Low or if more than one input is High at the same time, all outputs are set to Low and **Fault present** becomes High.

Priority

In **Priority** mode, more than one input may be High at a time. The outputs are set dependent on the High input with the highest index (input 1 = 1, input 2 = 2, ...). If all inputs are Low at the same time, all outputs are set to Low and **Fault present** becomes High.

Priority-to-binary (input 1 dominant)

In this mode, all outputs are set to Low, if input 1 is High, regardless of the other inputs. If input 1 is Low, the function block behaves as in **Priority** mode. If all inputs are Low at the same time, all outputs are set to Low and **Fault present** becomes High.

Truth tables for the Binary encoder function block

The following applies for the truth tables in this section:

- "0" means logic Low
- "1" means logic High
- "x" means "any" = "0" or "1"

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

Tab. 38: Truth table for the Binary encoder function block with 2 inputs in One out 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	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

Tab. 39: Truth table for the Binary encoder function block with 8 inputs in One out of N mode

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

Tab. 40: Truth table for the Binary encoder function block with 2 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

Tab. 41: Truth table for the Binary encoder function block with 8 inputs in Priority mode

Logic programming – Function blocks

Tab. 42: Truth table for the Binary encoder function block with 2 inputs in Priority mode with dominant input 1

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

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

Tab. 43: Truth table for the Binary encoder function block with 8 inputs in Priority mode with dominant input 1

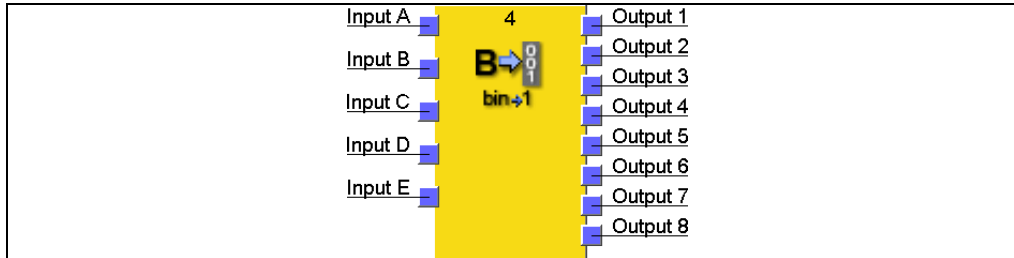


Evaluate Fault present if the Binary encoder function block is used for safety purposes!
 If you use the Binary encoder function block for safety relevant logic, you may have to evaluate the **Fault present** output depending on your application. This is the only way to distinguish whether only input 1 is High or if an invalid input state exists. In both cases, all outputs will be Low.

8.6.13 Binary decoder

Function block diagram

Fig. 105: Logic connections for the Binary decoder function block



General description

The Binary decoder function block decodes dependent on the current configuration a binary code to a one-out-of-N (one-hot) or to a priority code. Up to 5 inputs can be configured. The number of outputs depends on the number of inputs. Evaluating inputs A, B and C allows to encode binary codes with decimal values from 0 to 7 with a single Binary decoder function block (input A = 2⁰, input B = 2¹, input C = 2²). By using the optional inputs D and E it is possible to combine up to four Binary decoder function blocks in order to encode binary codes with decimal values from 0 to 31.

Parameters of the function block

Parameter	Possible values
Encoder mode	<ul style="list-style-type: none"> One out of N Priority
Inputs	<ul style="list-style-type: none"> Not inverted Inverted
Number of inputs	1 to 5
Value range	<ul style="list-style-type: none"> 0-7 8-15 (only available if more than 3 inputs are used) 16-23 (only available if 5 inputs are used) 24-31 (only available if 5 inputs are used)

Tab. 44: Parameters of the Binary decoder function block

One out of N

In **One out of N** mode, only the output with the number that corresponds to the current input values will be High.

Priority

In **Priority** mode, the output with the number that corresponds to the current input values and all outputs with lower numbers will be High.

Inputs inverted/not inverted

Using this parameter it is possible to invert all input values.

Truth tables for the Binary decoder function block

The following applies for the truth table in this section:

- “0” means logic Low
- “1” means logic High

Input A	Out 2	Out 1
0	0	1
1	1	0

Tab. 45: Truth table for the Binary decoder with 1 input in One out of N mode

Input B	Input A	Out 4	Out 3	Out 2	Out 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

Tab. 46: Truth table for the Binary decoder with 2 inputs in One out of N mode

Input C	Input B	Input A	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 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

Tab. 47: Truth table for the Binary decoder with 3 inputs in One out of N mode

Input A	Out 2	Out 1
0	0	1
1	1	1

Tab. 48: Truth table for the Binary decoder with 2 inputs in Priority mode

Input B	Input A	Out 4	Out 3	Out 2	Out 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

Tab. 49: Truth table for the Binary decoder with 2 inputs in Priority mode

Logic programming – Function blocks

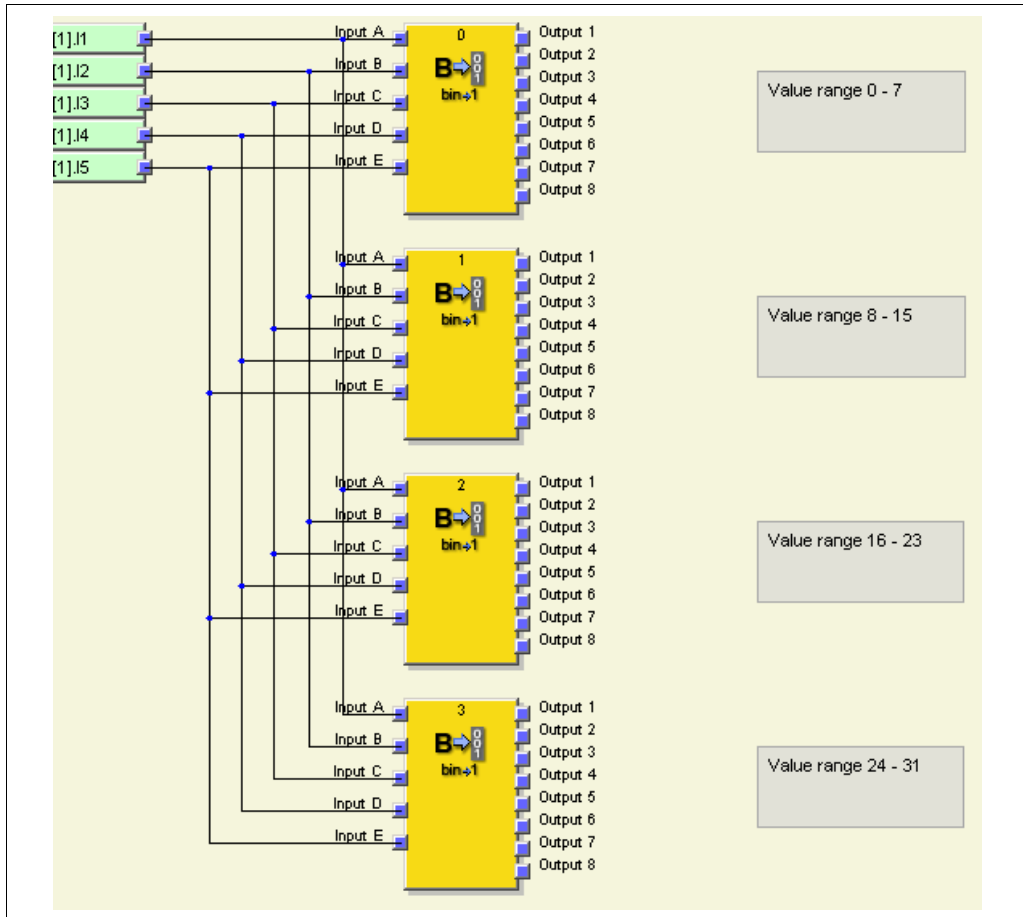
Tab. 50: Truth table for the Binary decoder with 3 inputs in Priority mode

Input C	Input B	Input A	Out 8	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 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

Evaluation of more than three inputs

If 4 or 5 inputs are used, up to four Binary decoder function blocks can be combined in order to encode binary codes with values from 0 to 31.

Fig. 106: Combination of four Binary decoder function blocks



When using multiple Binary decoder function blocks in combination you have to configure the **Value range** option of each function block for the value range it shall cover. This range depends on the values of inputs D and E.

Tab. 51: Value range of the Binary decoder function block depending on input D

Input D	Outputs
0	0-7
1	8-15

Tab. 52: Value range of the Binary decoder function block depending on inputs D and E

Input E	Input D	Outputs
0	0	1-7
0	1	8-15
1	0	16-23
1	1	24-31

- If **Input D** and **Input E** are set to the *same* value as the **Value range** parameter (e.g. if **Input E** = 1, **Input D** = 0 and **Value range** is set to 16-23), the function block will

behave as shown in the truth tables above, depending on the values of inputs A, B and C and the configured Decoder mode (**One out of N** or **Priority**).

- If **Input D** and **Input E** are set to a *lower* value than the **Value range** parameter (e.g. **Input E** = 0, **Input D** = 1 and **Value range** = 16-23) all outputs are Low independent of the configured Decoder mode (**One out of N** or **Priority**).
- If **Input D** and **Input E** are set to a *higher* value than the **Value range** parameter (e.g. **Input E** = 1, **Input D** = 1 and **Value range** = 16-23) ...

in **One out of N** mode, all outputs are set to Low,
in **Priority** mode, all outputs are set to High.

8.6.14 Message generator

Function block diagram

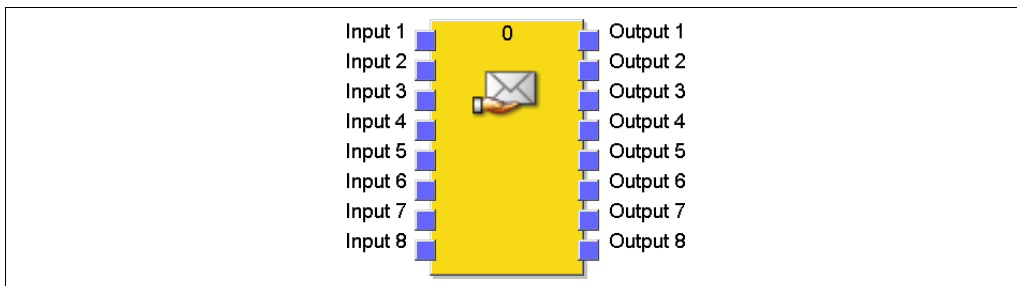


Fig. 107: Function block diagram for the Message generator function block

General description

The Message generator function block monitors up to eight inputs. If at one of these inputs an edge is detected according to the configuration, the function block sets the corresponding output to High for the duration of the logic execution time and adds a user defined text message to the diagnostics history. This can be read out in online mode using the samos®PLAN diagnostic function (see section 5.8 “Diagnostics view” on page 39).

These messages will be deleted when the voltage supply for the samos®PRO system is interrupted.

NOTE

Parameters of the function block

Parameter	Possible values
Number of inputs	1 to 8
Messages	Up to 64 user defined messages per project.
Input condition	<ul style="list-style-type: none"> • Rising edge • Falling edge • Rising or falling edge

Tab. 53: Parameters of the Message generator function block

How to configure the Message generator function block:

The following example shows the Message generator function block with two emergency stop buttons and a safety switch attached.

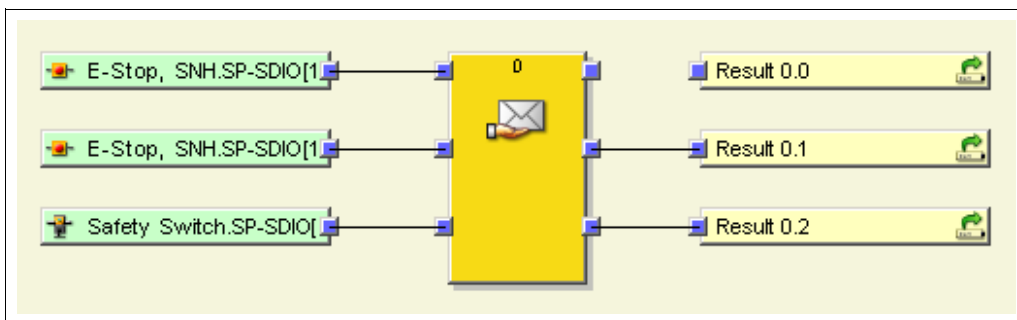


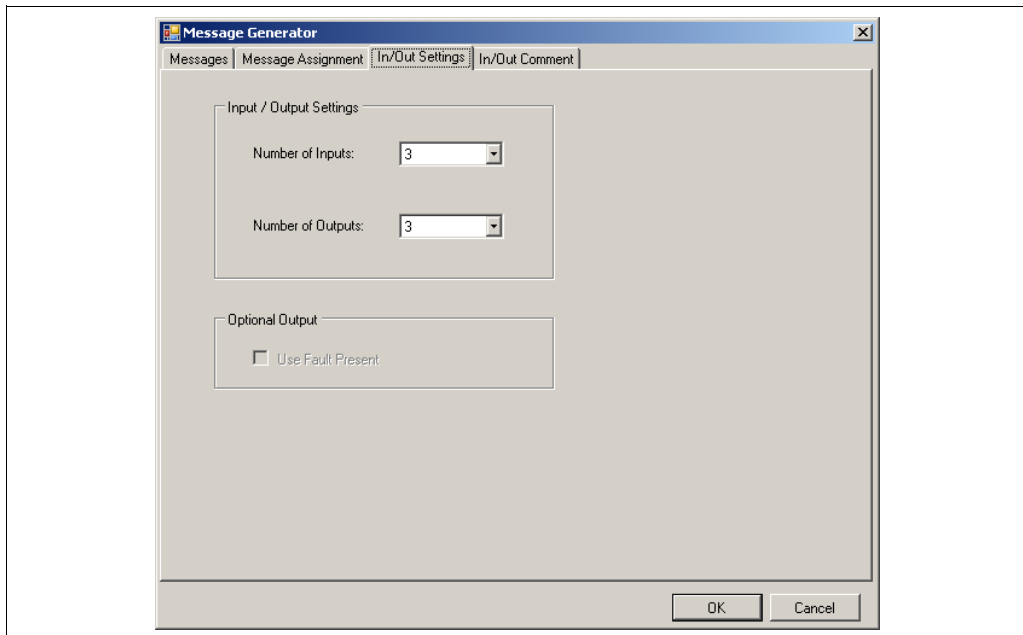
Fig. 108: Configuration example for Message generator with two emergency stop buttons and a safety switch

Logic programming – Function blocks

To configure the Message generator function block, proceed as follows:

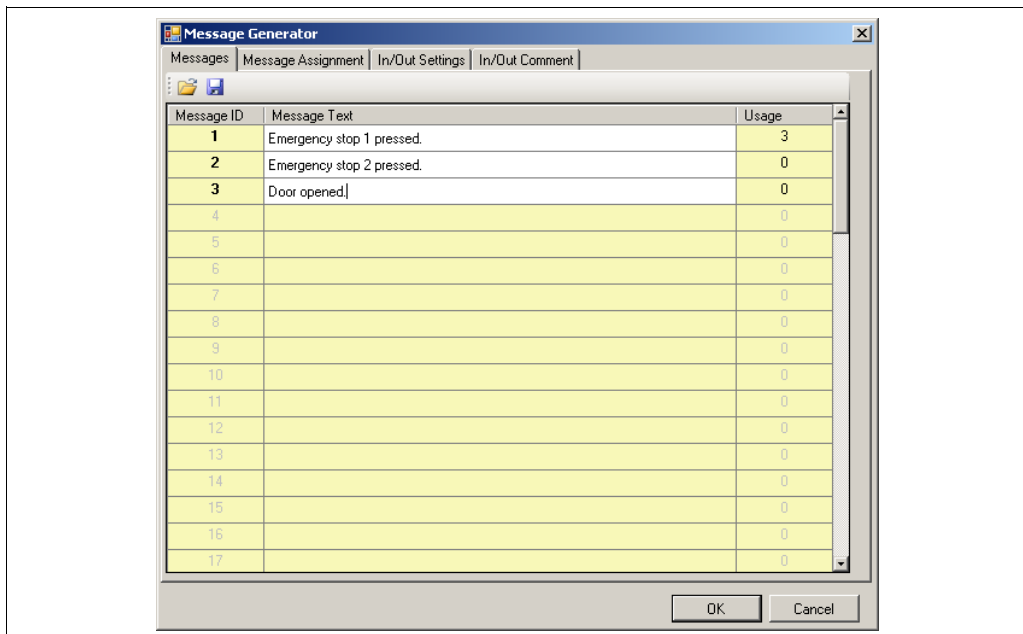
- ➔ Connect input elements to the function block. Double click on the function block to open the configuration dialog and click then on the **In/out settings** tab.

Fig. 109: In/out settings for the Message generator function block



- ➔ Choose the number of inputs that you wish to attach to the function block.
- ➔ Then click the **Messages** tab and enter the messages that shall be output in the diagnostics.

Fig. 110: Messages of the Message generator function block



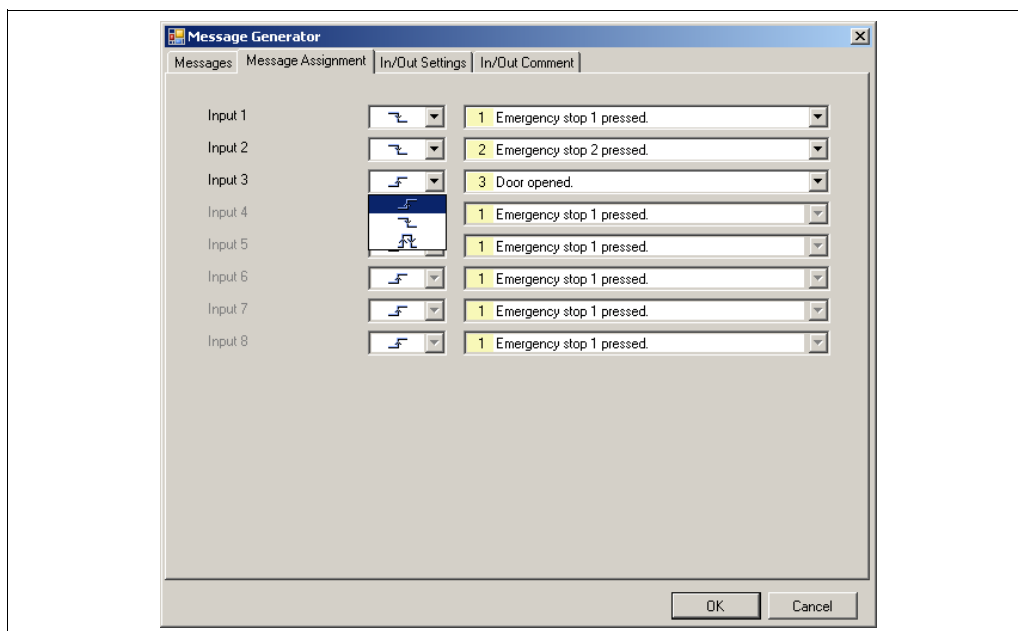
NOTES

- The messages entered are valid globally for all Message generator function blocks used in a project.
- In a single project, you can enter up to 64 different messages with a length of up to 32767 characters each.
- Using the **Import from CSV** and **Export to CSV** buttons at the top left of the window, you can save the messages to a text file in CSV (comma separated values) format or import messages from a CSV file.
- ➔ Then click the **Message assignment** tab. Assign the desired message to each used input and choose the input condition that must be fulfilled for the related message to be sent (rising edge, falling edge or rising and falling edge).

The message assignment can not be exported or imported.

NOTE

Fig. 111: Message assignment for the Message generator function block



Message priority

If more than one conditions are fulfilled at the same time, the following priorities apply:

- On a single Message generator function block, the input with the lower number is prioritized, i.e. the message triggered by this input will be logged first.
- If several Message generator function blocks are used, the function block with the lower function block index is prioritized, i.e. the messages generated by this function block will be logged first.

8.6.15 Routing 1:N

Function block diagram

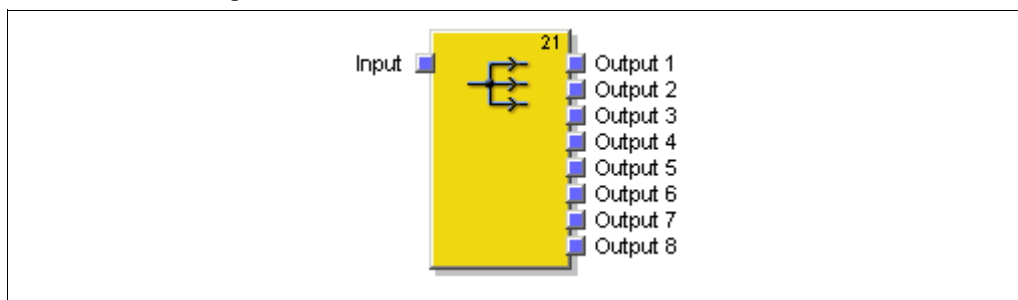


Fig. 112: Function block diagram for the Routing 1:N function block

General description

The Routing 1:N function block passes an input signal from a preceding function block to up to eight output signals. This function block makes it possible to connect an output of a function block or an input element with several output elements (e.g. outputs of an SP-SDIO module, CPU marker). It is not needed, however, for the connection to several function block inputs, because this can be done directly.

Parameters of the function block

Parameter	Possible values
Number of inputs	1 to 8

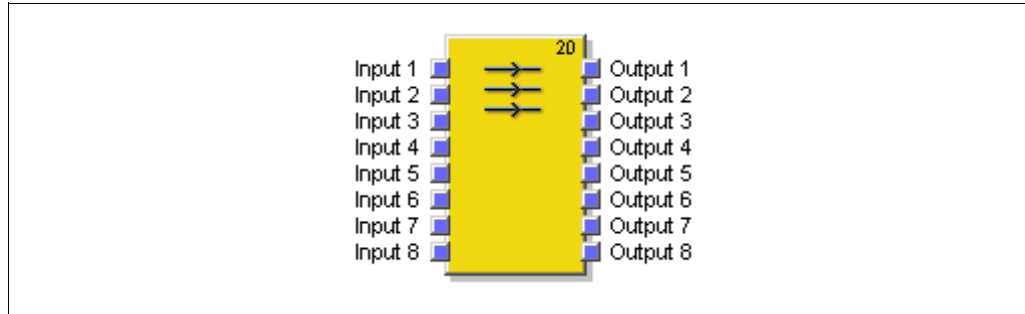
Tab. 54: Parameters of the Routing 1:N function block

Logic programming – Function blocks

8.6.16 Routing N:N

Function block diagram

Fig. 113: Function block diagram for the Routing N:N function block



General description

The Routing N:N function block passes up to eight input signals parallel to up to eight outputs. The input signal can originate from a preceding function block or directly from a physical input.

Parameters of the function block

Tab. 55: Parameters of the Routing 1:N function block

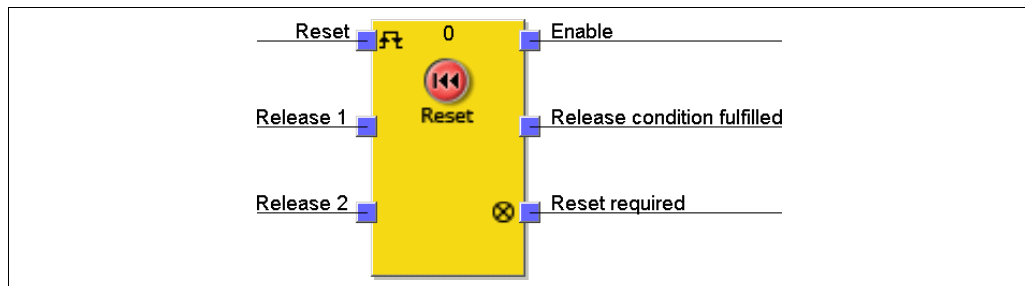
Parameter	Possible values
Number of inputs	1 to 8
Invert input x	Each input of this function block can be inverted (see section 8.4.2 "Inversion of input connections" on page 73).

8.7 Application-specific function blocks

8.7.1 Reset

Function block diagram

Fig. 114: Function block diagram for the Reset function block



General description

The Reset function block can be used to fulfil the normative requirements for safety applications on acknowledging a manual safety stop and the subsequent request to restart the application. Typically, each safety logic system of a samos®PRO safety controller contains a Reset function block.

Parameters of the function block

Tab. 56: Parameters of the Reset function block

Parameter	Possible values
Min. reset pulse time	<ul style="list-style-type: none"> 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 displays the result of an AND combination of all activated **Release** inputs. It is High if all activated **Release** inputs are High.

Reset required output

The **Reset required** output shows by pulsing at 1 Hz that the function block expects a valid reset pulse at the **Reset** input so that the **Enable** output can become High. This is the case if the **Release condition fulfilled** output is High, i.e. all activated **Release** inputs are High, but the **Enable** output is still Low. Typically this output is used to control a signal lamp.

Enable output

The **Enable** output becomes High, if the **Release condition fulfilled** output is High and a valid reset pulse has been detected at the **Reset** input, provided that all activated **Release** outputs remain High.

The **Min. reset pulse time** defines the minimum required duration of the pulse at the **Reset** input. Valid values are 100 ms and 350 ms. If the pulse duration is shorter than the configured minimum pulse time or longer than 30 s, the pulse is ignored.

The **Enable** output becomes Low, if one or more **Release** inputs become Low.

Ensure that the transitions of the signals for resetting fulfil the requirements!

In case of a short-circuit to High (to 24 V DC) at a physical input, the evaluated signal can have a pulse when the signal is reset due to the short-circuit detection. If such a pulse can result in a dangerous state in the machine, the following points have to be observed:

- Ensure protected cable laying for the signal line (due to cross-circuiting to other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.



WARNING

Sequence/timing diagram

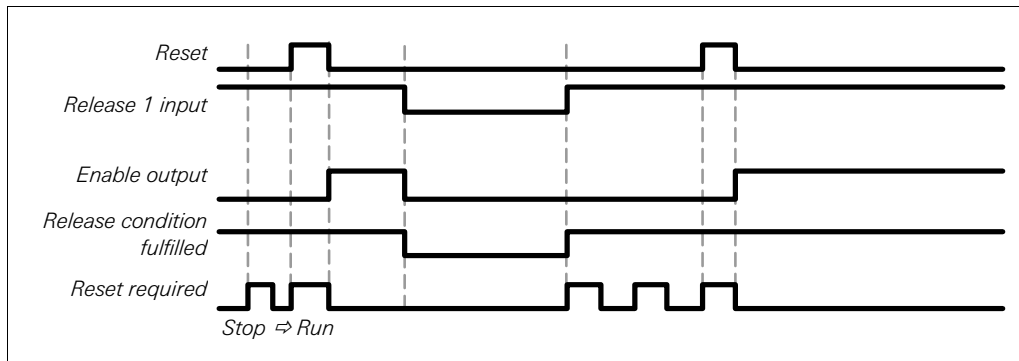


Fig. 115: Sequence/timing diagram for the Reset function block

8.7.2 Restart

Function block diagram

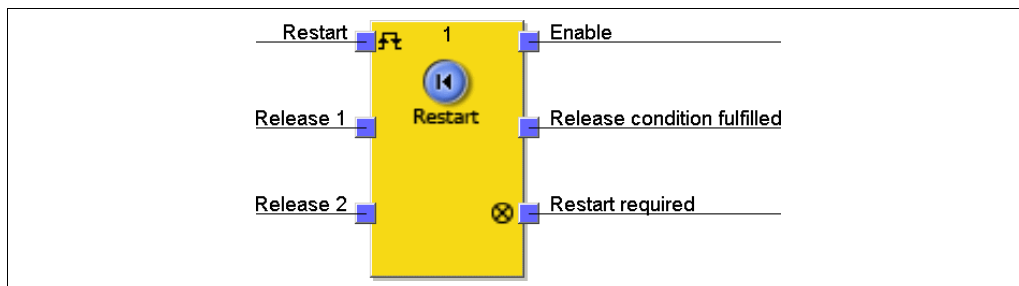


Fig. 116: Function block diagram for the Restart function block

General description

The internal logic of the Restart function block has the same functionality as the Reset function block. The Restart function block allows graphic differentiation between the function blocks with regard to the observation of application standards for acknowledging a manual restart request.

Logic programming – Function blocks

Tab. 57: Parameters of the Restart function block

Parameters of the function block

Parameter	Possible values
Min. restart pulse time	<ul style="list-style-type: none"> • 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 displays the result of an AND combination of all activated **Release** inputs. It is High if all activated **Release** inputs are High.

Restart required output

The **Restart required** output shows by pulsing at 1 Hz that the function block expects a valid restart pulse at the **Restart** input so that the **Enable** output can become High. This is the case if the **Release condition fulfilled** output is High, i.e. all activated **Release** inputs are High, but the **Enable** output is still Low. Typically this output is used to control a signal lamp.

Enable output

The **Enable** output becomes High, if the **Release condition fulfilled** output is High and a valid restart pulse has been detected at the **Restart** input, provided that all activated **Release** outputs remain High.

The **Min. restart pulse time** defines the minimum required duration of the pulse at the **Restart** input. Valid values are 100 ms and 350 ms. If the pulse duration is shorter than the configured minimum pulse time or longer than 30 s, the pulse is ignored.

The **Enable** output becomes Low, if one or more **Release** inputs become Low.



WARNING

Ensure that the transitions of the signals for restarting fulfil the requirements!

In case of a short-circuit to High (to 24 V DC) at a physical input, the evaluated signal can have a pulse when the signal is reset due to the short-circuit detection. If such a pulse can result in a dangerous state in the machine, the following points have to be observed:

- Ensure protected cable laying for the signal line (due to cross-circuiting to other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

Sequence/timing diagram

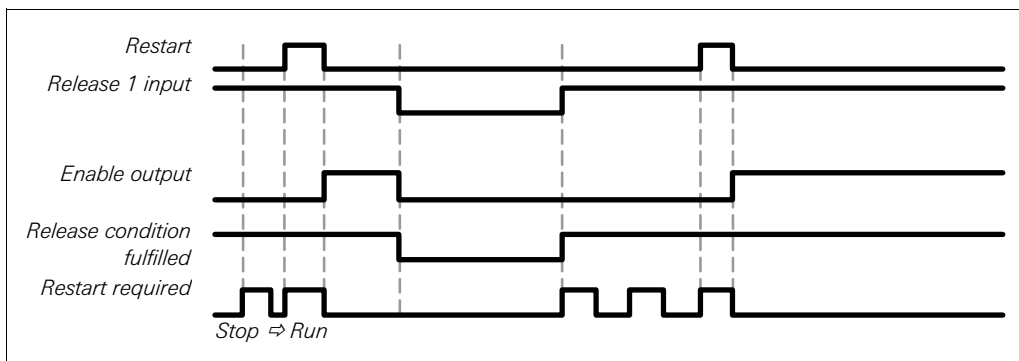


Fig. 117: Sequence/timing diagram for the Restart function block

8.7.3 Off-delay timer

Function block diagram

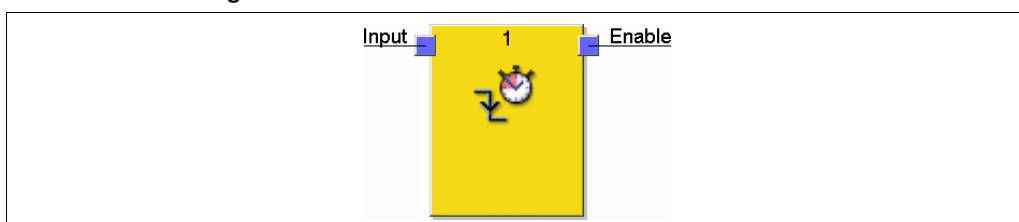


Fig. 118: Function block diagram for the Off-delay timer function block

General description

The Off-delay timer function block delays the switching-off of the output signal by a configurable duration.

Parameters of the function block

Parameter	Possible values
Delay time	0 to 300 seconds in steps of 10 ms. If the value is not 0, it has to be greater than the logic execution time.

Tab. 58: Parameters of the Off-delay timer function block

The timer begins with the delay sequence when a falling edge (High to Low) on the input occurs. If the timer has expired after the configured delay period, the **Enable** output changes also to Low, provided the input remains Low. If the input changes to High, the **Enable** output is set to High immediately and the delay timer is reset.

Sequence/timing diagram

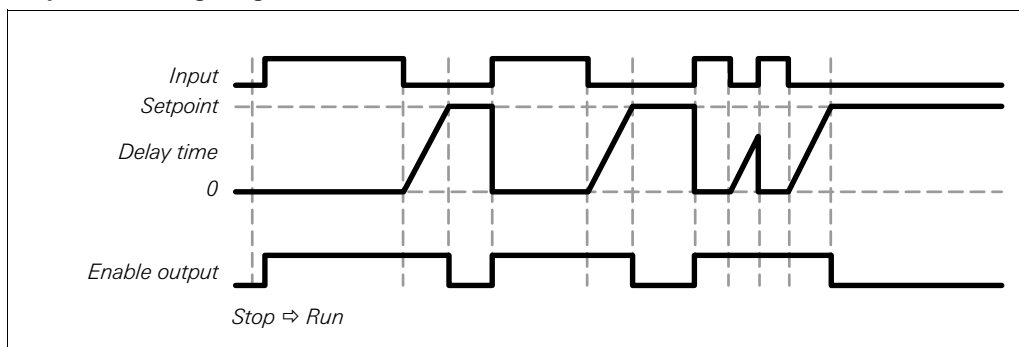


Fig. 119: Sequence/timing diagram for the Off-delay timer function block

8.7.4 Adjustable off-delay timer

Function block diagram

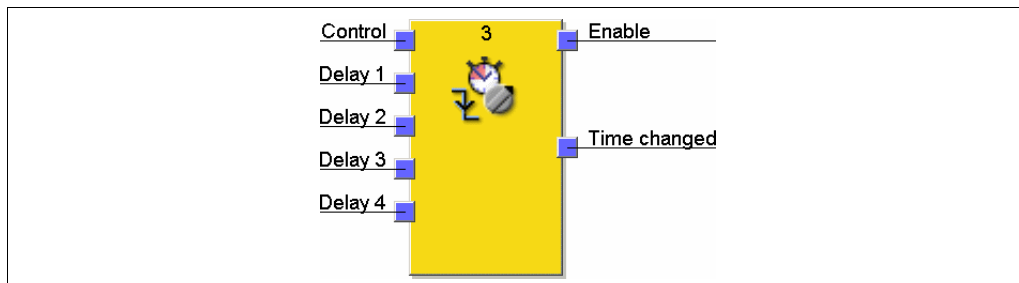


Fig. 120: Function block diagram for the Adjustable off-delay timer function block

General description

The Adjustable off-delay timer function block delays the switching-off of the **Enable** output by an adjustable duration. Four individual delay times can be configured, each of which can be activated via a related **Delay** input. The overall delay is the sum of all activated delay times.

Parameters of the function block

Parameter	Possible values
Off delay time 1	0 to 600 seconds in steps of 10 ms. If the value is not 0, the related input is activated. In this case, the value has to be greater than the logic execution time.
Off delay time 2	The overall delay (sum of all delay times) is limited to 600 seconds.
Off delay time 3	
Off delay time 4	

Tab. 59: Parameters of the Adjustable off-delay timer function block

The timer begins with the delay sequence when a falling edge (High to Low) occurs at the **Control** input. If the timer has expired after the selected overall delay period, the **Enable**

Logic programming – Function blocks

output changes also to Low, provided the **Control** input remains Low. If the **Control** input changes to High, the **Enable** output is set to High immediately and the delay timer is reset.

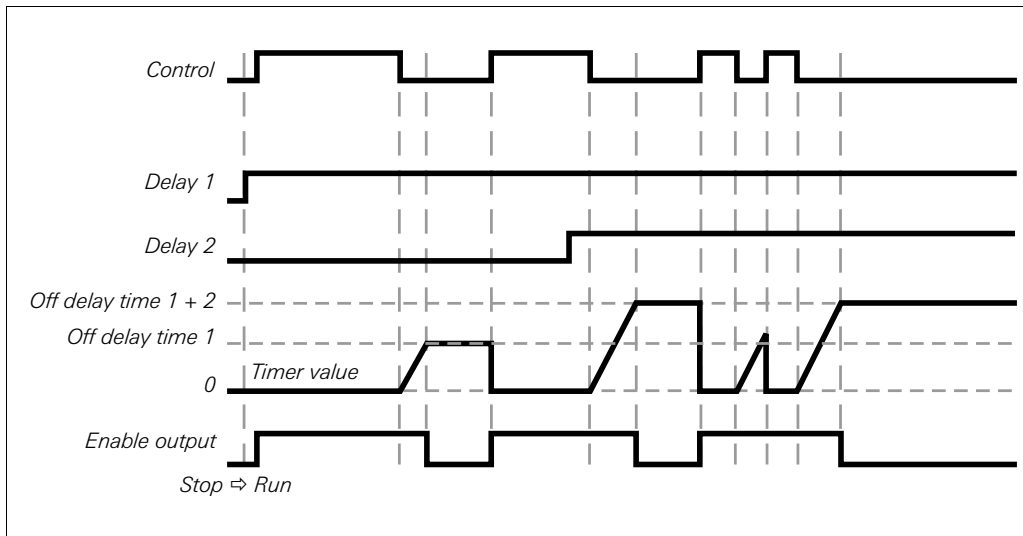
If during a running delay sequence any **Delay** input changes its value, the **Time changed** output changes to High and remains High until the **Control** input becomes High again.

The effective overall delay time depends on the **Delay** inputs that were High at the moment when the falling edge at the **Control** input has occurred. This means that a change on the **Delay** inputs during a delay sequence has no effect on the current delay sequence.

If the **Control** input is Low in the first logic cycle after transition from the Stop state to the Run state, the **Enable** output remains Low as well.

Sequence/timing diagram

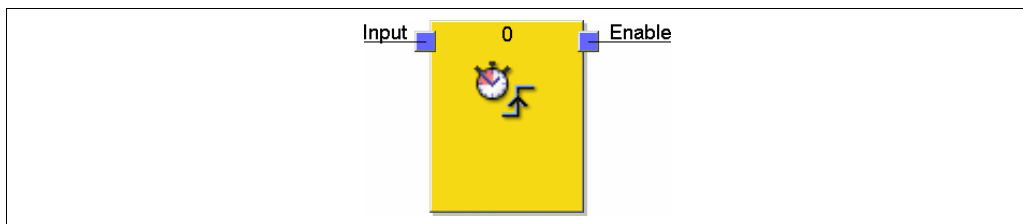
Fig. 121: Sequence/timing diagram for the Adjustable off-delay timer function block with Off delay time 1 and Off delay time 2



8.7.5 On-delay timer

Function block diagram

Fig. 122: Function block diagram for the On-delay timer function block



General description

The On-delay timer function block delays the switching-on of the output signal by a specified duration.

Parameters of the function block

Tab. 60: Parameters of the On-delay timer function block

Parameter	Possible values
Delay time	0 to 300 seconds in steps of 10 ms. If the value is not 0, it has to be greater than the logic execution time.

The timer begins with the delay sequence when a rising edge (Low to High) on the input occurs. If the timer has expired after the configured delay period, the **Enable** output changes also to High, provided the input remains High. If the input changes to Low, the **Enable** output is set to Low immediately and the delay timer is reset.

Sequence/timing diagram

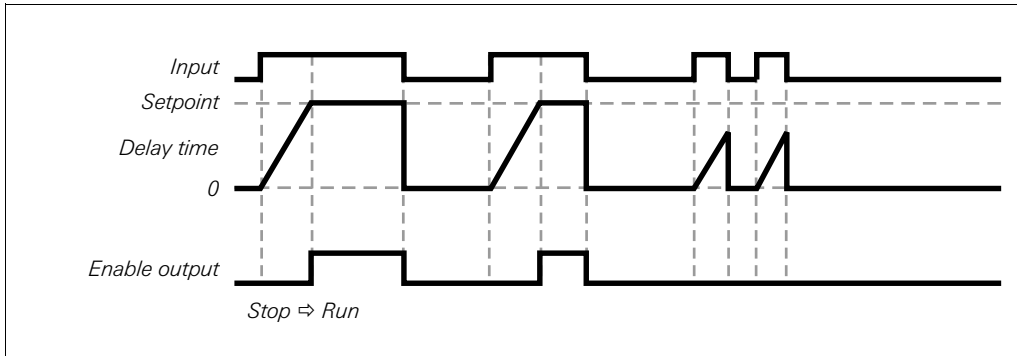


Fig. 123: Sequence/timing diagram for the On-delay timer function block

8.7.6 Adjustable on-delay timer

Function block diagram

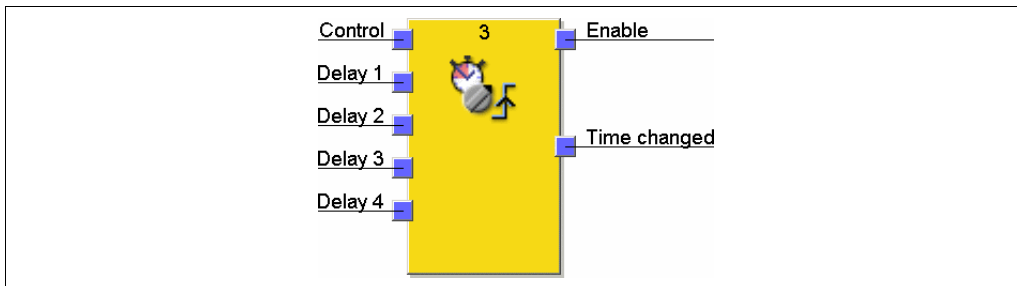


Fig. 124: Function block diagram for the Adjustable on-delay timer function block

General description

The Adjustable on-delay timer function block delays the switching-on of the **Enable** output by an adjustable duration. Four individual delay times can be configured, each of which can be activated via a related **Delay** input. The overall delay is the sum of all activated delay times.

Parameters of the function block

Parameter	Possible values
On delay time 1	0 to 600 seconds in steps of 10 ms.
On delay time 2	If the value is not 0, the related input is activated. In this case, the value has to be greater than the logic execution time.
On delay time 3	The overall delay (sum of all delay times) is limited to 600 seconds.
On delay time 4	

Tab. 61: Parameters of the Adjustable on-delay timer function block

The timer begins with the delay sequence when a rising edge (Low to High) occurs at the **Control** input. If the timer has expired after the selected overall delay period, the **Enable** output changes also to High, provided the **Control** input remains High. If the **Control** input changes to Low, the **Enable** output is set to Low immediately and the delay timer is reset.

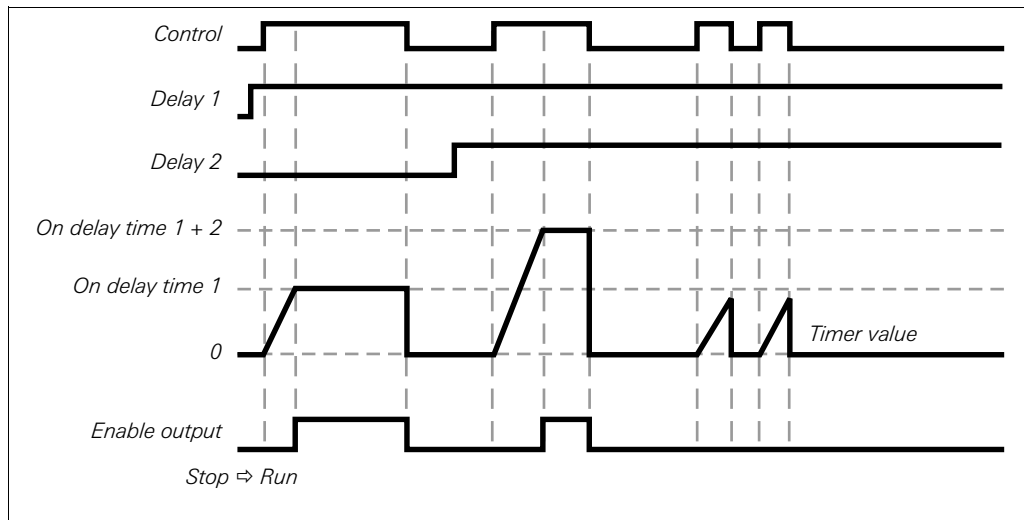
If during a running delay sequence any **Delay** input changes its value, the **Time changed** output changes to High and remains High until the **Control** input becomes Low again.

The effective overall delay time depends on the **Delay** inputs that were High at the moment when the rising edge at the **Control** input has occurred. This means that a change on the **Delay** inputs during a delay sequence has no effect on the current delay sequence.

If the **Control** input is High in the first logic cycle after transition from the Stop state to the Run state, the **Enable** output becomes High immediately without delay.

Fig. 125: Sequence/timing diagram for the Adjustable on-delay timer function block with On delay time 1 and On delay time 2

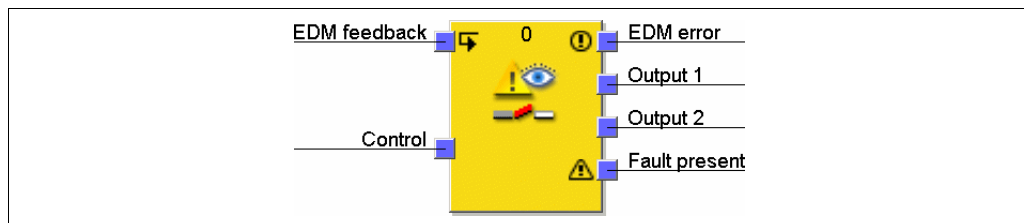
Sequence/timing diagram



8.7.7 EDM (External device monitoring)

Function block diagram

Fig. 126: Function block diagram for the EDM function block



General description

The EDM (External device monitoring) function block allows to control an external device (e.g. a contactor) and to check on the basis of its feedback signal whether it has switched as expected. To this purpose the external device is connected to **Output 1** and/or **Output 2**. The feedback signal is connected to the **EDM feedback** input. The **Control** input is connected to the logic signal that represents the desired state for the external device, e.g. the **Enable** output of a Reset function block.

Parameters of the function block

Tab. 62: Parameters of the EDM function block

Parameter	Possible values
Max. feedback delay	100 to 1,000 ms in 10 ms steps. The value has to be greater than the logic execution time.
Use Fault present	<ul style="list-style-type: none"> • With • Without

Output 1 und Output 2

Both outputs have always the same value. This way, two outputs are available to connect two output elements directly.

Output 1 and **Output 2** become High, if the **EDM feedback** is High and the **Control** input changes from Low to High subsequently.

Output 1 and **Output 2** become Low, if the **Control** input is Low or if an EDM error is present (**EDM error** output is High).

EDM error and Fault present

Generally it is expected that the **EDM feedback** input always takes the inverted value of the **Control** input before the configured Max. feedback delay (T_{EDM}) has expired.

The **EDM error** and **Fault present** outputs become High, if ...

- the **Control** input changes from Low to High and the **EDM feedback** input is Low (irrespective of T_{EDM}), or
- the **Control** input changes from Low to High and the **EDM feedback** input does not change from High to Low before T_{EDM} has expired, or
- the **Control** input changes from High to Low and the **EDM feedback** input does not change from Low to High before T_{EDM} has expired, or
- the **Control** input is Low and the **EDM feedback** input changes to Low for longer than T_{EDM} , or
- the **Control** input is High and the **EDM feedback** input changes to High for longer than T_{EDM} .

The **EDM error** and **Fault present** outputs become Low, if a signal sequence is detected that sets **Output 1** and **Output 2** to High.

If you require a delay of the **Output 1** and **Output 2** signals, then you have to realise the output delay with another function block before the EDM function block and not after it. Otherwise this can result in an EDM error.

NOTE

Sequence/timing diagram

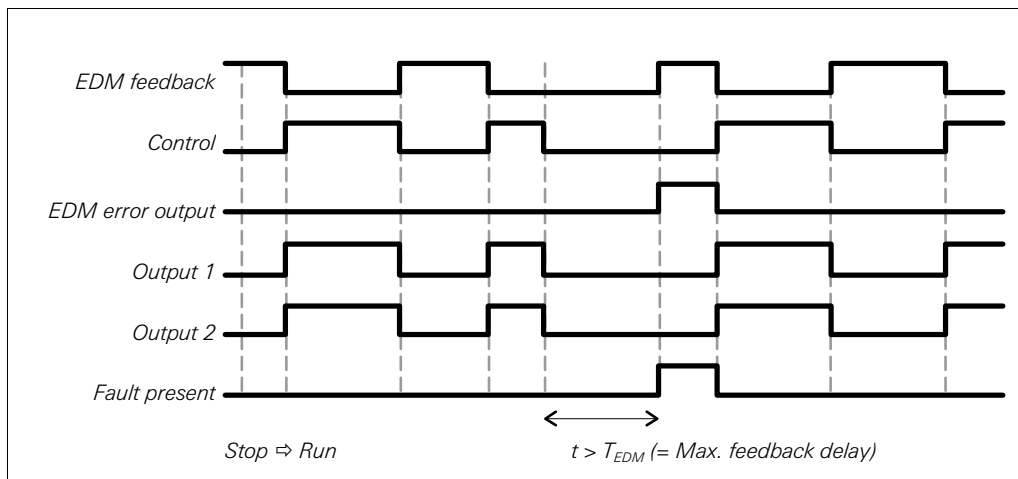


Fig. 127: Sequence/timing diagram for the External device monitoring (EDM) function block

8.7.8 Valve monitoring

Function block diagram

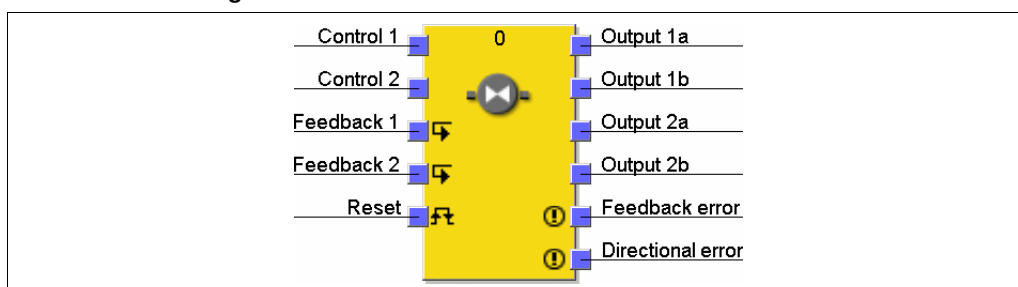


Fig. 128: Logic connections for the Valve monitoring function block, configured for a directional valve

General description

The Valve monitoring function block allows to control valves and to check on the basis of their feedback signals whether they have switched as expected.

To this purpose the valves are connected with **Output 1a** to **Output 2b**. The feedback signals are connected to the **Feedback 1** and **Feedback 2** inputs. The **Control 1** and **Control 2** inputs are connected to the logic signal that represents the desired state for the valve, e.g. the **Enable** output of a Reset function block. Depending on the valve type, some of the signals are not used.

Three different valve types are available: Single valves, double valves and directional valves.

Logic programming – Function blocks

Tab. 63: Parameters of the Valve monitoring function block

Parameters of the function block

Parameter	Possible values
Reset condition	<ul style="list-style-type: none"> Manual reset Auto reset
Continuous monitoring when valve is active	<ul style="list-style-type: none"> Active Inactive
Valve mode	<ul style="list-style-type: none"> Single (Control 1, Output 1a, Output 1b, Feedback 1 activated) Double (Control 1, Output 1a, Output 1b, Feedback 1, Output 2a, Output 2b, Rücklesen 2 activated) Directional (Control 1, Output 1a, Output 1b, Feedback 1, Control 2, Output 2a, Output 2b, Feedback 2, Directional error activated)
Max. switch-on feedback delay	50 ms to 10 s in steps of 10 ms (0 = disabled, only with CPU firmware V2.00.0 or higher). If this parameter is disabled, then the option Continuous monitoring when valve is active has to be deactivated as well. If this parameter is enabled, the value has to be greater than the logic execution time.
Max. switch-off feedback delay	50 ms to 10 s in steps of 10 ms (0 = disabled, only with CPU firmware V2.00.0 or higher). If enabled, the value has to be greater than the logic execution time.
Min. reset pulse time	<ul style="list-style-type: none"> 100 ms 350 ms
Use Fault present	<ul style="list-style-type: none"> With Without



Connect the feedback signals correctly!

The signals for **Feedback 1** and **Feedback 2** have to be protected against short-circuits to the signals for outputs (e.g. **Output 1a**, **1b**, **2a** and **2b**) as well as against each other (e.g. by means of protected wiring or wiring of these signals solely within the control cabinet).

Output 1a to Output 2b

Both outputs of a pair (**Output 1a** and **Output 1b** or **Output 2a** and **Output 2b**) have always the same value. This way, two outputs per valve are available to connect two output elements directly.

Output 1a/1b or **Output 2a/2b** become High, if the related **Feedback 1** or **Feedback 2** input is High and the related **Control** input changes from Low to High subsequently.

Output 1a/1b or **Output 2a/2b** become Low, if the related **Control** input is Low or if a fault is present (**Feedback error** output is High or **Directional error** output is High).

The related control input for **Output 1a/1b** is always **Control 1**.

The related control input for **Output 2a/2b** depends on the configured valve type:

- For *Directional valve*: **Control 1**
- For *Double valve*: **Control 2**

Feedback error, Directional error and Fault present

Generally it is expected that the **Feedback 1/2** input always takes the inverted value of the related **Control** input before the configured Max. switch-on feedback delay (T_{ON}) or Max. switch-off feedback delay (T_{OFF}) has expired.

The **Feedback error** output becomes High, if ...

- the **Control** input changes from Low to High and the related **Feedback** input is Low (irrespective of T_{ON} and T_{OFF}), or
- T_{ON} is greater than zero and the **Control** input changes from Low to High and the related **Feedback** input does not change from High to Low before T_{ON} has expired, or

- T_{OFF} is greater than zero and the **Control** input changes from High to Low and the related **Feedback** input does not change from Low to High before T_{OFF} has expired, or
- *Continuous monitoring when valve is active* is active and the **Control** input is High and the related **Feedback** input changes to High.

The **Directional error** output becomes High, if the *Valve type* parameter is = *Directional* and the **Control 1** and **Control 2** inputs are High at the same time.

The **Fault present** output becomes High, if **Feedback error** and/or **Directional error** is High.

The **Feedback error**, **Directional error** and **Fault present** outputs become Low, if all activated Control inputs are Low and all activated Feedback inputs are High. If the Reset condition is configured as *Manual reset*, a valid reset pulse must be applied at the **Reset** input additionally.

The **Min. reset pulse time** defines the minimum required duration of the pulse at the **Reset** input. Valid values are 100 ms and 350 ms. If the pulse duration is shorter than the configured minimum pulse time or longer than 30 s, the pulse is ignored.

Ensure that the transitions of the signals for resetting fulfil the requirements of the safety standards and regulations!

In case of a short-circuit to High (to 24 V DC) at a physical input, the evaluated signal can have a pulse when the signal is reset due to the short-circuit detection. If such a pulse can result in a dangerous state in the machine, the following points have to be observed:

- Ensure protected cable laying for the signal line (due to cross-circuiting to other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.



Sequence/timing diagrams

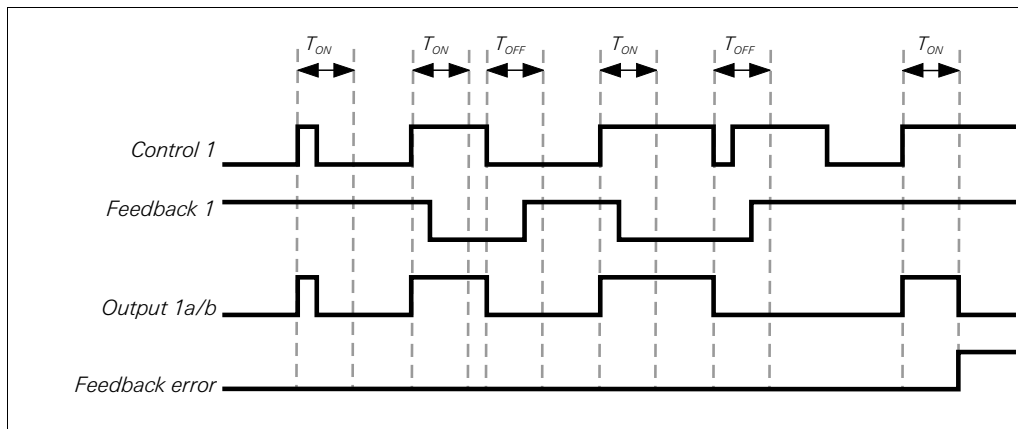


Fig. 129: Sequence/timing diagram for single valve in manual reset mode

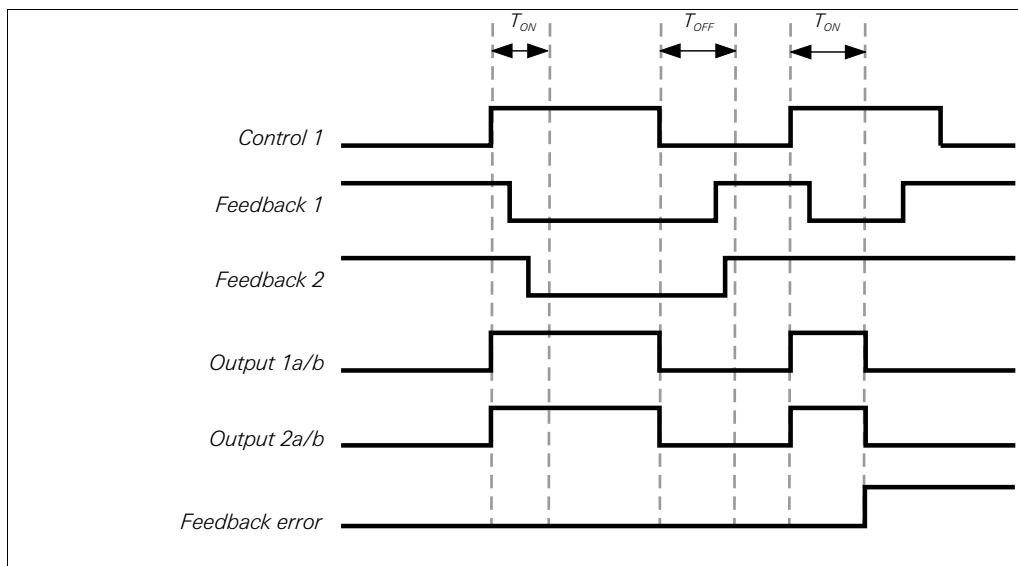
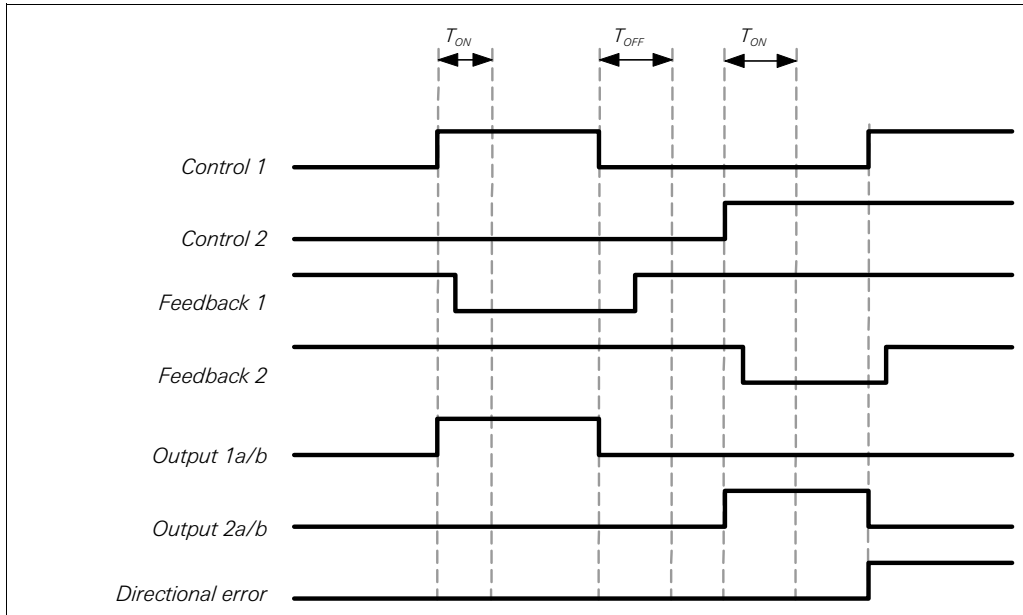


Fig. 130: Sequence/timing diagram for double valve in manual reset mode

Logic programming – Function blocks

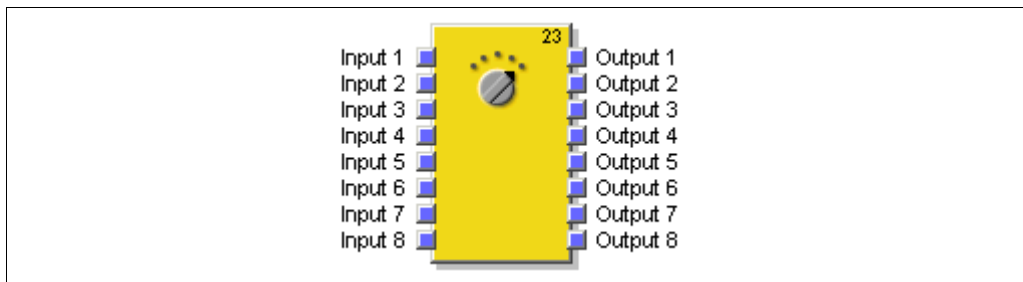
Fig. 131: Sequence/timing diagram for directional valve



8.7.9 User mode switch

Function block diagram

Fig. 132: Function block diagram for the User mode switch function block



General description

The User mode switch function block selects an output depending on an input value. Output x is High if Input x is High.

The function block supports 2 to 8 inputs and the corresponding outputs.

Only one input may be High at any time (1 out of n). If no input or more than one input is High, the output that was High last is kept High for the configured discrepancy time. After expiration of the discrepancy time, the **Fault present** output changes to High and all outputs change to the value defined in the error output combination.

If in the first logic cycle after the transition from the Stop state to the Run state there is no valid input combination, the configured error output combination is applied to the outputs and the **Fault present** output changes to High immediately.

Parameters of the function block

Tab. 64: Parameters of the User mode switch function block

Parameter	Possible values
Discrepancy time	0 to 10 seconds in steps of 10 ms
Error output combination	Checked outputs will be High and not checked outputs will be Low when Fault present is High.
Number of inputs or Number of outputs	2 to 8
Use Fault present	<ul style="list-style-type: none"> • With • Without

Truth table for the User mode switch function block

The truth table uses the following designations:

“0” means logic Low

“1” means logic High

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 High or no input High for shorter than the set discrepancy time								0	= Last output combination							
More than one input High or no input High for equal or longer than the set discrepancy time								1	= Error output combination							

Tab. 65: Truth table for the User mode switch function block

Sequence/timing diagram

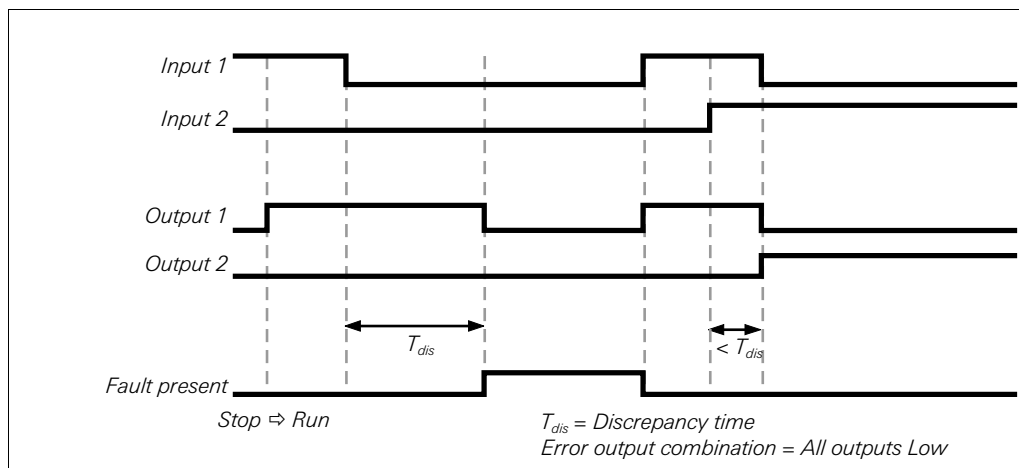


Fig. 133: Sequence/timing diagram for the User mode switch function block

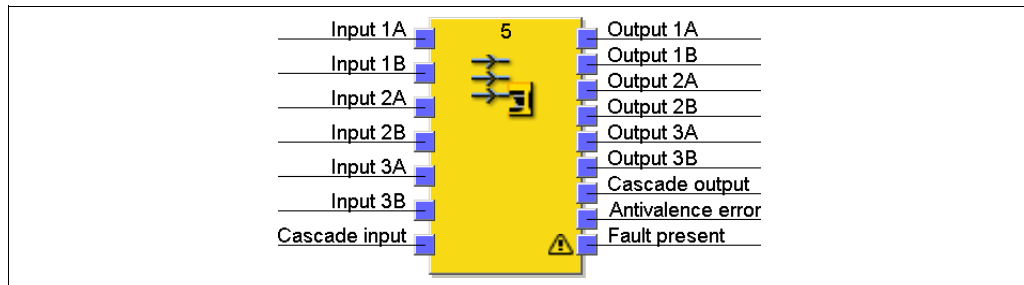
- If the inputs of the function block are connected to inputs of an extension module which are connected to test outputs, and the faulty input combination is the result of a test pulse error (stuck-at-high) which leads to a Low input value, the test pulse error needs to be reset first, e.g. by briefly interrupting the corresponding input line or test output line.
- If the inputs of the function block are connected to inputs of an extension module which are connected to test outputs, then a cross circuit between the used inputs is detected only if a user mode is selected that activates one of these inputs.

NOTES

8.7.10 Switch synchronization

Function block diagram

Fig. 134: Logic connections for the Switch synchronization function block



General description

The Switch synchronization function block was designed to improve the integration of Wieland Electric safety scanners (e.g. S3000). It monitors the input signals for changes. If a change of any input signal has been detected, the function block “freezes” the values of its outputs until the configurable **Hold time for outputs** has expired.

Parameters of the function block

Tab. 66: Parameters of the Switch synchronization function block

Parameter	Possible values
Cascade input	<ul style="list-style-type: none"> • With • Without
Antivalence check	<ul style="list-style-type: none"> • Active • Inactive <p>If this function is active, the optional Antivalence error output can be used.</p>
Hold time for outputs	10 ms to 10 s in steps of 10 ms. The value has to be greater than the logic execution time.
Invert input 1A ... Invert input 3B	Each input of this function block can be inverted (see section 8.4.2 “Inversion of input connections” on page 73).
Number of inputs or Number of outputs	1 to 6
Use Fault present	<ul style="list-style-type: none"> • With • Without

NOTE

An inverted input will invert the corresponding output signal as well. E.g. if input 1A is High, but configured as inverted, it will be evaluated as Low and output 1A will be set to Low.

Hold time for outputs

The **Hold time for outputs** defines the time delay between the first change of any input signal and the latching of the input signals, i.e. the reaction of the outputs. This can be used to compensate delays e.g. between the different contacts of mechanical switches.

Non cascading 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 of any input signal starts the timer. The outputs 1A to 3B keep their values during the configured **Hold time for outputs**. When the timer has elapsed, the values of the inputs 1A to 3B at this point of time are applied to output 1A to 3B, irrespective of the result of the antivalence check. The outputs keep these values until the next synchronization process occurs.

Cascading mode – with cascade input

Several Switch synchronization function blocks can be combined to a cascade so that all outputs will be switched at precisely the same time.

By cascading multiple Switch synchronization function blocks it is possible to synchronize more than six inputs. If the function block is configured with **Cascade** input, the **Cascade** output is present additionally.

All cascaded function blocks must be configured with the same **Hold time for outputs**.

NOTE

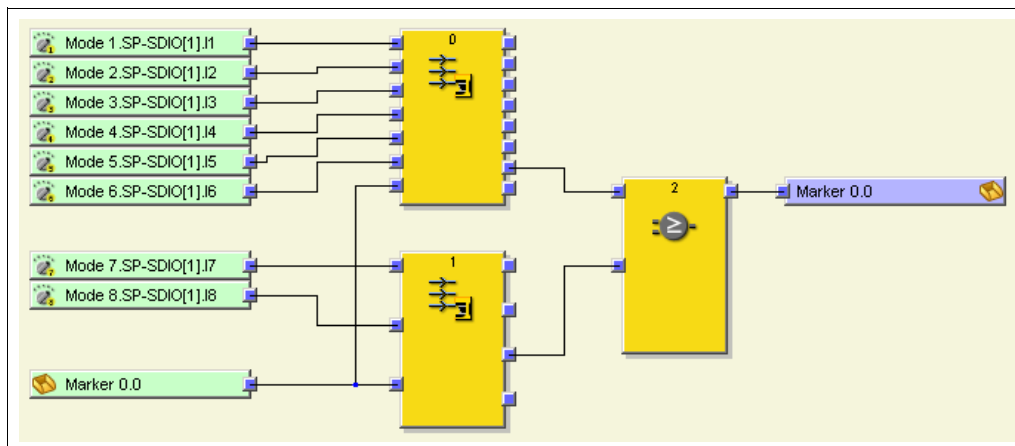


Fig. 135: Logic example for two cascaded Switch synchronization function blocks

All **Cascade output** signals must be connected back to the **Cascade inputs** of all used Switch synchronization function blocks via an OR function block and a CPU marker (see also Fig. 139).

- Do *not* use a jump address for the back routing of the cascade signal but a CPU marker. This ensures that all related Switch synchronization function blocks will process the signal at the same logic cycle.
- The timer is increased by the necessary value to compensate the delay resulting from the connection of the **Cascade input** via the CPU marker.

NOTE

A rising edge on the **Cascade input** starts the timer (timer start value is the system time from the *last* logic execution cycle). When the timer has elapsed, the values of the inputs 1A to 3B at this point of time are applied to output 1A to 3B, irrespective of the result of the antivalence check. The outputs keep these values until the next synchronization process occurs.

Antivalence check

If this function is enabled, an antivalence check is performed every time when the **Hold time for outputs** has elapsed (i.e. every time when the outputs take the current input values). If any of the used input pairs **Input 1A/Input 1B** to **Input 3A/Input 3B** does not have antivalent values (i.e. one input of each pair must be Low and the other input must be High), then the **Antivalence error** output changes to High. It becomes Low again when another synchronization process has been finished without an antivalence error. The behaviour of the outputs 1A to 3B however is independent of the result of the antivalence check.

In order to obtain a defined output value combination in case of an antivalence error, the Error output combination function block can be used (see chapter 8.7.11 "Error output combination" on page 113).

NOTE

Behaviour on startup

On transition from the Stop state to the Run state, the outputs will be set immediately according to the input values and the antivalence check will be executed, if configured. In this case the function block does not wait for the **Hold time for outputs** to expire.

Sequence/timing diagrams

Fig. 136: Sequence/timing diagram for the Switch synchronization function block without cascading

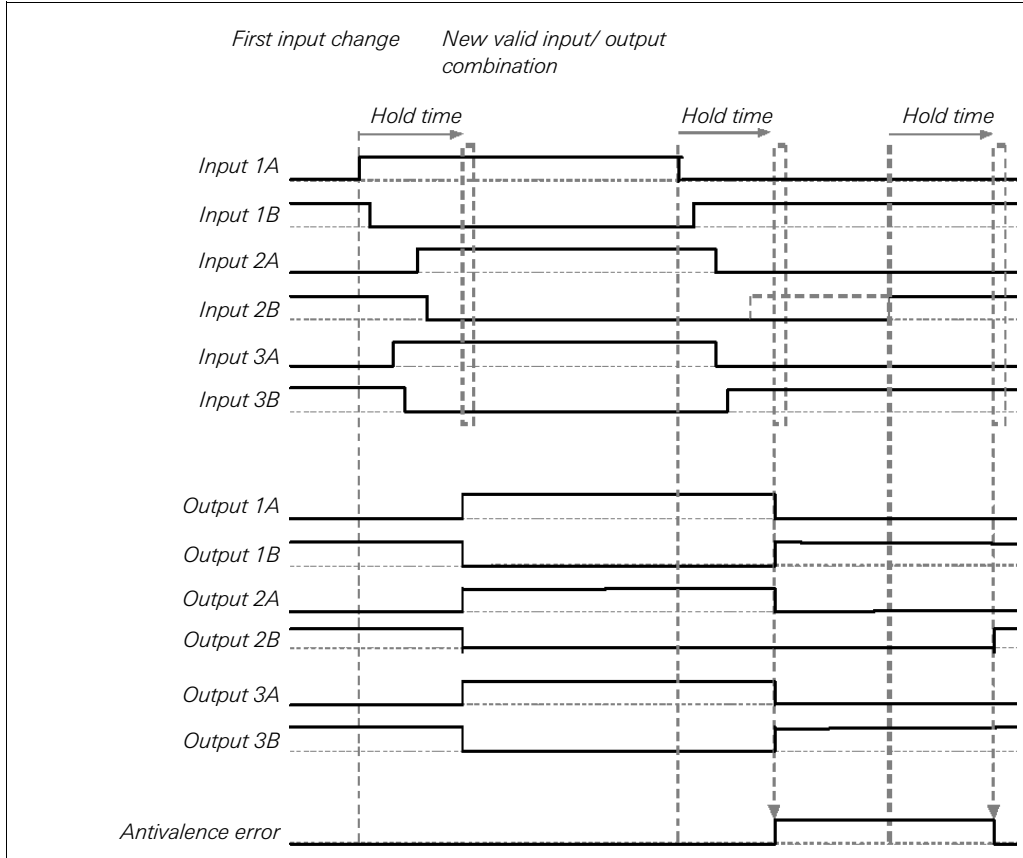
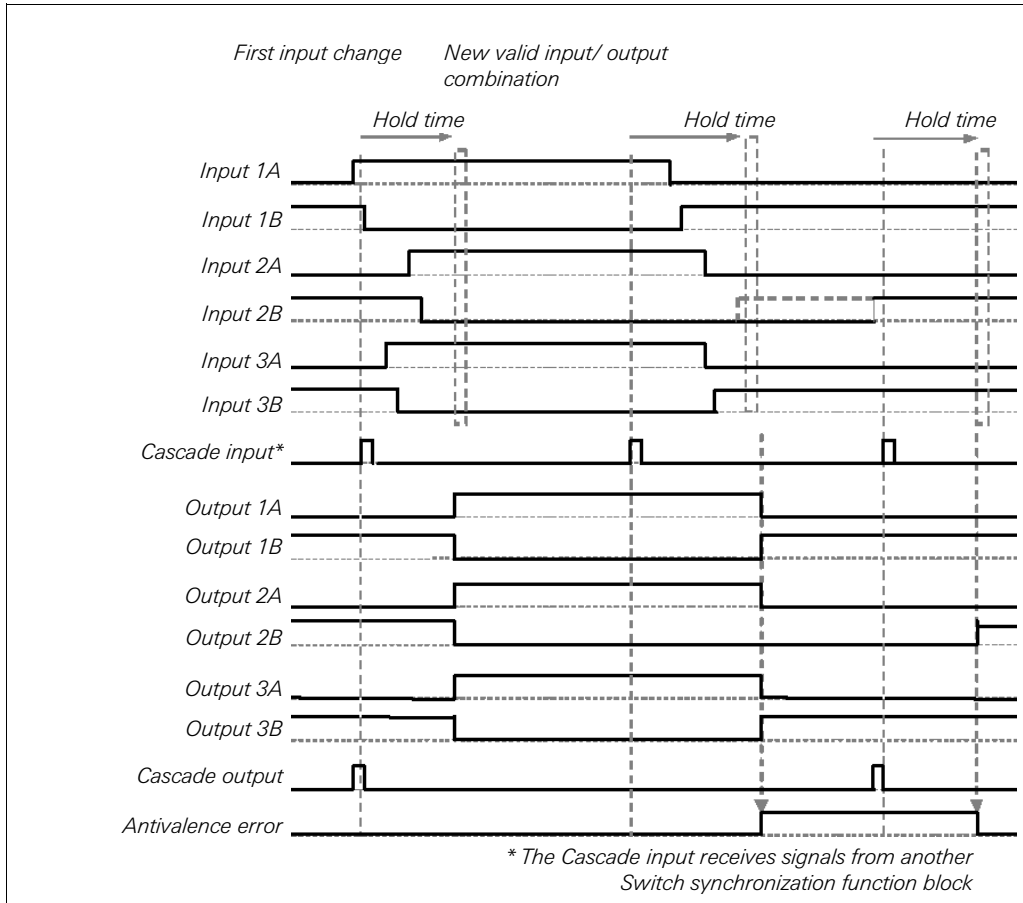


Fig. 137: Sequence/timing diagram for the Switch synchronization function block with cascading



8.7.11 Error output combination

Function block diagram

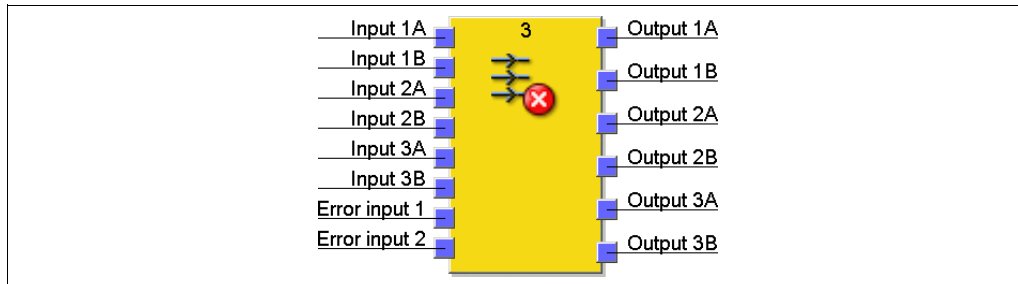


Fig. 138: Logic connections for the Error output combination function block

General description

The Error output combination function block was designed to improve the integration of safety scanners (e.g. S3000). It can be used to set the outputs to predefined values under certain conditions, e.g. in order to output a dedicated error output combination in case of an antivalence error of the Switch synchronization function block.

Parameters of the function block

Parameter	Possible values
Number of error inputs	<ul style="list-style-type: none"> 1 error input 2 error inputs
Number of inputs or Number of outputs	1 to 6
Error output combination	For each output individually: <ul style="list-style-type: none"> High Low

Tab. 67: Parameters of the Error output combination function block

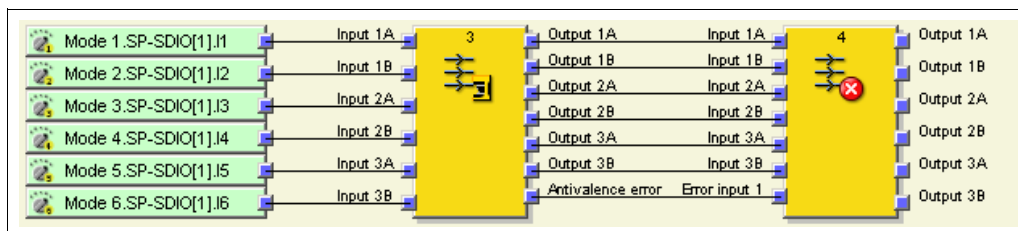


Fig. 139: Logic example for the Error output combination function block

Truth table

The following applies for the truth table in this section:

"0" means logic Low

"1" means logic High

"x" means "any" = "0" or "1"

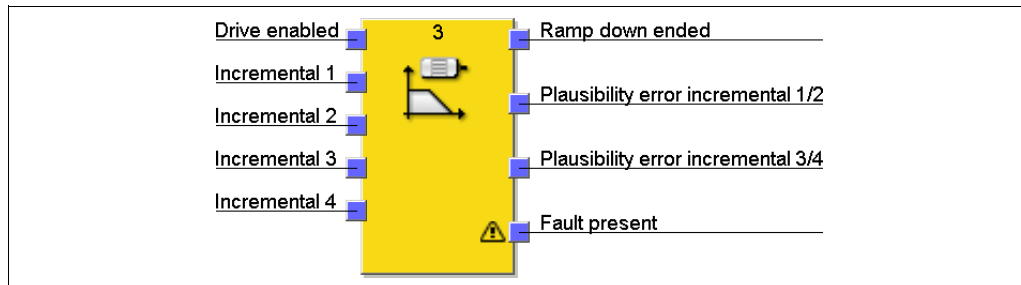
Error input 1	Error input 2	Output 1A	Output 1B	Output 2A	Output 2B	Output 3A	Output 3B
1	x	Error output combination					
x	1	Error output combination					
0	0	Input 1A	Input 1B	Input 2A	Input 2B	Input 3A	Input 3B

Tab. 68: Truth table for Error output combination

8.7.12 Ramp down detection

Function block diagram

Fig. 140: Logic connections for the Ramp down detection function block



General description

The Ramp down detection function block checks whether a connected drive has been stopped, meaning that for a configurable time no pulses have been detected from the encoder system (e.g. from a HTL encoder or proximity switches). Depending on the result of this check e.g. a safety door lock can be unlatched.

The ramp down detection is started by a falling edge of the **Drive enabled** input signal. A stop of the drive is detected when no signal change (falling or rising edge) at any **Incremental** input has occurred for at least the configured **Min. time between signal changes**. In this case the **Ramp down ended** output will change to High. If the **Drive enabled** input signal becomes High, this will immediately set the **Ramp down ended** output to Low and will also cancel a currently running ramp down detection.

During the running state (i.e. while the **Drive enabled** input is High) and the stop detected state (**Ramp down ended** output is High) the **Incremental** inputs are not monitored except for plausibility, if configured (see Fig. 145).

The function block allows an optional plausibility check of the **Incremental** inputs. If the plausibility check is enabled, at least one signal within a signal pair must be High at any time. The tolerance time for this evaluation is equal to the logic execution time. This means that both inputs of a pair may be Low for the duration of the logic execution time without causing an error (see also Fig. 146).

On transition from the Stop state to the Run state, this tolerance time is not active. This means that if both inputs of a pair are Low during the 1st logic cycle, a plausibility error will be detected. This way a cable interruption can be detected, provided that the encoder provides appropriate signals, e.g. complementary outputs or proximity switches and a tooth-wheel with 270° tooth width and 180° phase shift.

The **Fault present** output is set to High if any **Plausibility error incremental** output is High. The **Fault present** output is set to Low, if all error outputs are Low.

Parameters of the function block

Tab. 69: Parameters of the Ramp down detection function block

Parameter	Possible values
Number of incremental inputs	<ul style="list-style-type: none"> • 1 single encoder input • 1 pair of encoder inputs • 2 pairs of encoder inputs
Input plausibility checks	<ul style="list-style-type: none"> • Disabled • Enabled <p>If enabled, then the number of incremental inputs must be either 1 pair or 2 pairs.</p>
Min. time between signal changes	100 ms to 10 s in steps of 10 ms. If the value is not 0, it has to be greater than the logic execution time.
Use Fault present	<ul style="list-style-type: none"> • With • Without

Ensure that your application fulfils the following requirements!

- The drive must continually reduce its speed if the power to the drive is turned off. Any acceleration, e.g. caused by hanging loads, must be excluded.
- Connect the signal that controls the physical output for the drive to the **Drive enabled** input. It must be ensured that if this input is Low, the torque of the drive is switched off in any case.
- Encoders must be connected locally to an SP-SDIO or SP-SDI module on the same samos[®]PRO station (not via network or samos[®]NET etc.)
- The ON-OFF filter and OFF-ON filter of the inputs used for encoder signals must be disabled.



Configuration steps

- Encoder pulses must have a minimum duration of 2 ms. Shorter pulses will be eliminated by the SP-SDIO/SP-SDI modules (see step 1 below).
- Determine the time between signal changes for the speed limit (see step 2 below).
- Check the minimum value for **Min. time between signal changes** (see step 3 below).

Step 1: Check the maximum signal frequency for incremental signals

The encoder pulses t_{high} and t_{low} must both have a minimum duration of 2 ms. Shorter pulses will be eliminated by the SP-SDIO and SP-SDI modules. This limits the allowed signal frequency and encoder speed depending on the encoder type. The following figures show typical signal patterns for different encoder types:

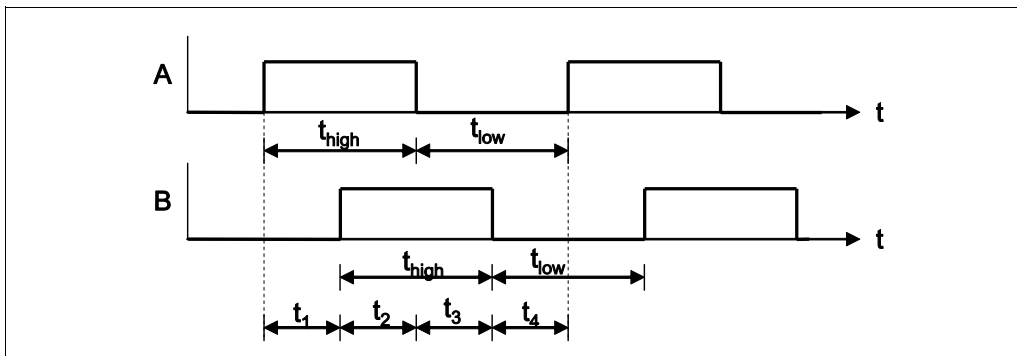


Fig. 141: Signal pattern for A/B 90° phase shift encoders

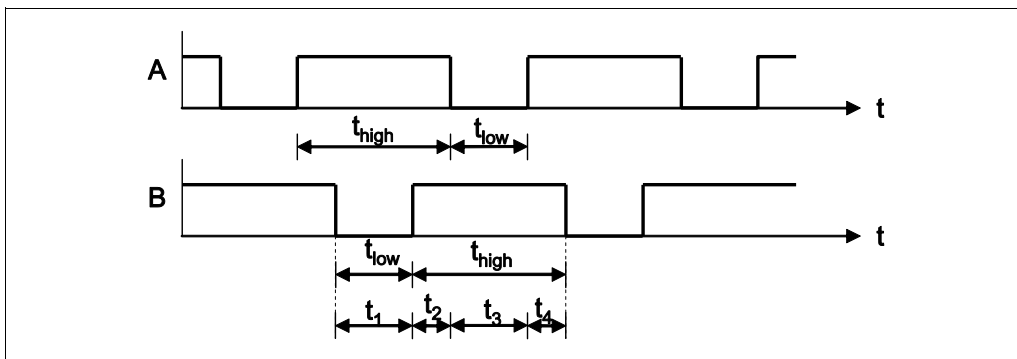


Fig. 142: Signal pattern for 1/3 gap 180° phase shift encoders

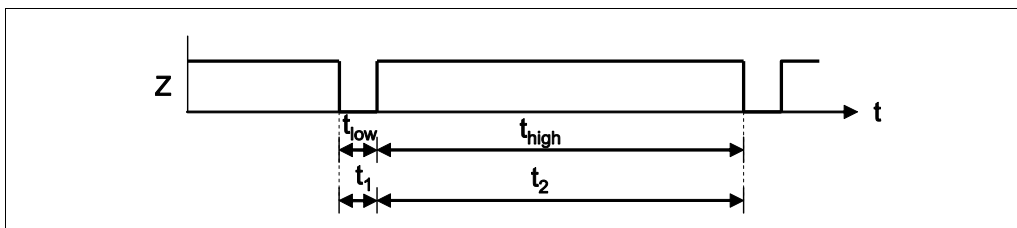


Fig. 143: Signal pattern for zero pulse encoders

It must be ensured by the design of your system that the minimum duration of the encoder pulses t_{high} and t_{low} must both be always > 2 ms. The following table shows typical values for different encoder types.

Logic programming – Function blocks

Encoder type	Max. signal frequency (Hz)	Max. allowed encoder speed (rpm) for x electrical signal periods/revolution							
		1	2	4	8	10	16	20	50
A/B, 90° phase shift	250	15,000	7,500	3,750	1,875	1,500	938	750	300
1/3 gap ¹⁾	166.7	10,000	5,000	2,500	1,250	1,000	625	500	200
Zero pulse 1°	1.4	83	42	21	10	8	5	4	2
Zero pulse 5°	7	416	208	102	52	41	26	21	8
Pulse 180°	250	15,000	7,500	3,750	1,875	1,500	938	750	300

Tab. 70: Maximum allowed encoder signal frequency and speed (rpm) depending on the encoder type

Step 2: Determine the time between signal changes for the speed limit

- ➔ Define the speed at which the **Ramp down ended** output shall be activated, e.g. to unlock a safety door.
- ➔ Define the maximum time that can pass between two signal changes at this speed (highest values of t_1-t_4). Take all possible tolerances into account, e.g. switching tolerances, tooth wheel tolerances etc.

$$\text{Min. time between signal changes} = \text{highest values of } t_1-t_4 + 10 \text{ ms}$$

Step 3: Check the minimum value for Min. time between signal changes

If one of the encoder pulses either t_{high} or t_{low} at the speed when the **Ramp down ended** output shall be activated is equal or lower than the logic execution time, the **Min. time between signal changes** must be calculated as follows:

$$\text{Min. time between signal changes} = \text{sqrt}(8 \times \tau_{\text{Decel}} \times T_{\text{Logic}} \times \alpha_{\text{max}}/360^\circ) + 10 \text{ ms}$$

Where ...

τ_{Decel} = Maximum deceleration time constant of the drive under worst conditions, meaning with highest inertia (e.g. with max. load for transportation) and lowest friction. You can determine this value by measuring the time the drive needs to come from maximum speed to a full stop. This time divided by 3 can be used as deceleration time constant (τ_{Decel}). This requires that the drive has an exponential deceleration, which is typically the case.

T_{logic} = Logic execution time

α_{max} = Maximum angle between two signal changes, related to an electrical signal period with 360° (same as the maximum value of t_1-t_4 but described as an angle). An electrical signal period may occur multiple times per revolution. If within one signal period the same input pattern exists twice (e.g. at 1/3 gap 180° phase shift in section t_2 and t_4), both sections have to be summarized and need to be considered as one section.

In any case the **Min. time between signal changes** must be greater than the logic execution time and must be rounded up to a multiple of 10 ms.



Take increased logic execution times into account!

Every time the logic program is changed, the logic execution time may increase. In this case it may be necessary to calculate the **Min. time between signal changes** again. Otherwise the operator of the machine will be in danger.

This minimum value for the **Min. time between signal changes** is required to ensure that a signal change is detected by the system, even if the minimum possible encoder pulses t_{high} and t_{low} are equal or lower than the logic execution time. This bases on the fact that when the power to the drive is turned off, it may still be running due to its inertia, but now it will steadily reduce its speed. In this case it is ensured that the frequency of the encoder signals will vary and that after a certain amount of time a signal change will be detected.

Example 1: A/B 90° phase shift

- 4 teeth per revolution
- Switching tolerances $\pm 5^\circ \rightarrow$ teeth 175° to 185° (corresponds to t_{low} , t_{high}); signal change 85° to 95° (corresponds to t_1-t_4)
- Maximum shaft speed = 3000 rpm = 50 Hz
- Shaft speed for release = 15 rpm = 0.25 Hz

¹⁾ 180° phase shift, 1 signal min. always High.

- Logic execution time = 8 ms
- Max. $\tau_{\text{Decel}} = 1 \text{ s}$
- ➔ Check the maximum signal frequency for incremental signals:
 - Max. signal frequency = $50 \text{ Hz} \times 4 \text{ teeth/revolution} = 200 \text{ Hz}$
 - Smallest $t_{\text{low}} = 1/200 \text{ Hz} \times 175^\circ/360^\circ = 2.4 \text{ ms} \rightarrow$ higher 2 ms ✓
 - Smallest $t_{\text{high}} = 1/200 \text{ Hz} \times 175^\circ/360^\circ = 2.4 \text{ ms} \rightarrow$ higher 2 ms ✓
- ➔ Determine the time between signal changes for the speed limit:
 - Signal frequency for release = $0.25 \text{ Hz} \times 4 \text{ teeth/revolution} = 1 \text{ Hz}$
 - Max. input pattern period = $1/1 \text{ Hz} \times 185^\circ/360^\circ = 514 \text{ ms}$
 - Time between signal changes = $514 \text{ ms} + 10 \text{ ms} = 524 \text{ ms}$
- ➔ Check the **Min. time between signal changes**:
 - Smallest $t_{\text{low}} = 2.4 \text{ ms}$ (result from above) \rightarrow higher 8 ms logic execution time? No!
 - Smallest $t_{\text{high}} = 2.4 \text{ ms} \rightarrow$ higher 8 ms logic execution time? No!
 - \rightarrow Check for **Min. time between signal changes** required
 - Min. time between signal changes = $\text{sqrt}(8 \times \tau_{\text{Decel}} \times T_{\text{Logic}} \times \alpha_{\text{max}}/360^\circ) + 10 \text{ ms}$
 - = $\text{sqrt}(8 \times 1 \text{ s} \times 8 \text{ ms} \times 95^\circ/360^\circ) + 10 \text{ ms} = 140 \text{ ms}$
 - \rightarrow Min. time between signal changes = 530 ms (rounded up to multiples of 10 ms)

Example 2: 1/3 gap 180° phase shift

- 8 teeth per revolution
- Switching tolerances $\pm 2^\circ \rightarrow$ teeth 118° to 122° (corresponds to $t_{\text{low}}, t_{\text{high}}$); signal change 118° to 122° (corresponds to t_1-t_4)
- Maximum shaft speed = $1200 \text{ rpm} = 20 \text{ Hz}$
- Shaft speed for release = $30 \text{ rpm} = 0.5 \text{ Hz}$
- Logic execution time = 20 ms
- Max. $\tau_{\text{Decel}} = 2 \text{ s}$
- ➔ Check the maximum signal frequency for incremental signals:
 - Max. signal frequency = $20 \text{ Hz} \times 8 \text{ teeth/revolution} = 160 \text{ Hz}$
 - Smallest $t_{\text{low}} = 1/160 \text{ Hz} \times 118^\circ/360^\circ = 2.04 \text{ ms} \rightarrow$ higher 2 ms ✓
 - Smallest $t_{\text{high}} = 1/160 \text{ Hz} \times 238^\circ/360^\circ = 4.13 \text{ ms} \rightarrow$ higher 2 ms ✓
- ➔ Determine the time between signal changes for the speed limit:
 - Signal frequency for release = $0.5 \text{ Hz} \times 8 \text{ teeth/revolution} = 4 \text{ Hz}$
 - Max. input pattern period = $1/4 \text{ Hz} \times 122^\circ/360^\circ = 85 \text{ ms}$
 - Time between signal changes = $85 \text{ ms} + 10 \text{ ms} = 95 \text{ ms}$
- ➔ Check the **Min. time between signal changes**:
 - Smallest $t_{\text{low}} = 2.04 \text{ ms}$ (result from above) \rightarrow higher 20 ms logic execution time? No!
 - Smallest $t_{\text{high}} = 4.13 \text{ ms} \rightarrow$ higher 20 ms logic execution time? No!
 - \rightarrow Check for **Min. time between signal changes** required
 - $\alpha_{1,\text{max}} = 122^\circ, \alpha_{2,\text{max}} + \alpha_{4,\text{max}} = 60^\circ + 60^\circ \pm 2^\circ = 122^\circ$ (same input pattern within one electrical signal period); $\alpha_{3,\text{max}} = 122^\circ$
 - $\rightarrow \alpha_{\text{max}} = 122^\circ$
 - Min. time between signal changes = $\text{sqrt}(8 \times \tau_{\text{Decel}} \times T_{\text{Logic}} \times \alpha_{\text{max}}/360^\circ) + 10 \text{ ms}$
 - = $\text{sqrt}(8 \times 2 \text{ s} \times 20 \text{ ms} \times 122^\circ/360^\circ) + 10 \text{ ms} = 339 \text{ ms}$
 - \rightarrow Min. time between signal changes = 340 ms (rounded up to multiples of 10 ms)
 - \rightarrow Approx. resulting shaft speed for release = $30 \text{ rpm} \times 85 \text{ ms}/340 \text{ ms} = 7.5 \text{ rpm}$

Example 3: Zero pulse 5°

- 1 tooth per revolution
- Switching tolerances $\pm 1^\circ \rightarrow$ teeth 4° to 6° (corresponds to $t_{\text{low}}, t_{\text{high}}$); signal change 354° to 56° (corresponds to t_1-t_4)
- Maximum shaft speed = $300 \text{ rpm} = 5 \text{ Hz}$
- Shaft speed for release = $3 \text{ rpm} = 0.05 \text{ Hz}$
- Logic execution time = 12 ms
- Max. $\tau_{\text{Decel}} = 1.5 \text{ s}$
- ➔ Check the maximum signal frequency for incremental signals:

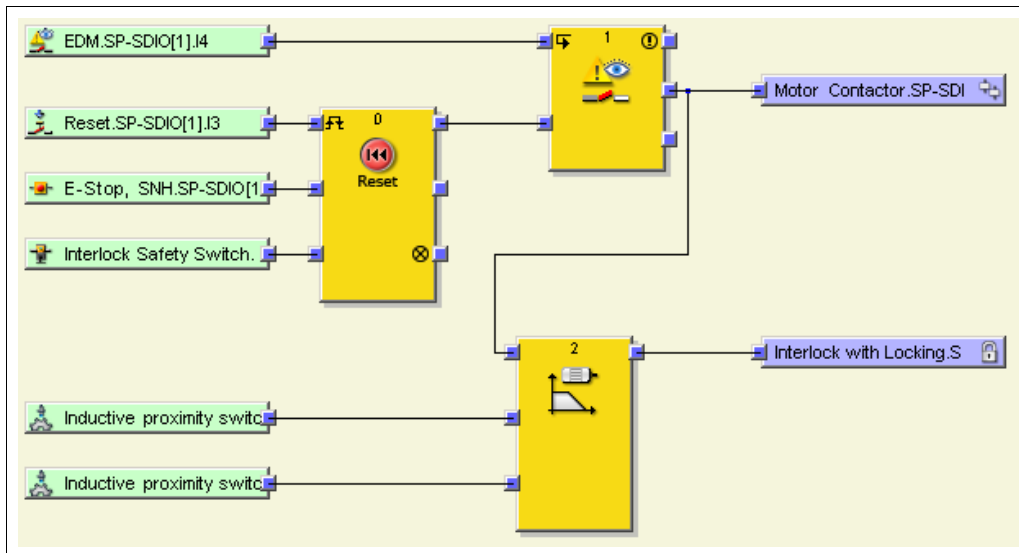
Logic programming – Function blocks

Max. signal frequency = 5 Hz × 1 tooth/revolution = 5 Hz
 Smallest $t_{low} = 1/5 \text{ Hz} \times 4^\circ/360^\circ = 2.2 \text{ ms} \rightarrow$ higher 2 ms ✓
 Smallest $t_{high} = 1/5 \text{ Hz} \times 356^\circ/360^\circ = 197.8 \text{ ms} \rightarrow$ higher 2 ms ✓

- ➔ Determine the time between signal changes for the speed limit:
 Signal frequency for release = 0.05 Hz × 1 tooth/revolution = 0.05 Hz
 Max. input pattern period = 1/0.05 Hz × 6°/360° = 333 ms
 Time between signal changes = 333 ms + 10 ms = 343 ms
- ➔ Check the **Min. time between signal changes**:
 Smallest $t_{low} = 2.2 \text{ ms}$ (result from above) → higher 12 ms logic execution time? No!
 Smallest $t_{high} = 197.8 \text{ ms}$ → higher 12 ms logic execution time? Yes!
 → Check for **Min. time between signal changes** required
 Min. time between signal changes = $\sqrt{8 \times \tau_{Decel} \times T_{Logic} \times \alpha_{max}/360^\circ} + 10 \text{ ms}$
 = $\sqrt{8 \times 1.5 \text{ s} \times 12 \text{ ms} \times 356^\circ/360^\circ} + 10 \text{ ms} = 387 \text{ ms}$
 → Min. time between signal changes = 390 ms (rounded up to multiples of 10 ms)
 → Approx. resulting shaft speed for release = 3 rpm × 333 ms/390 ms = 2.6 rpm

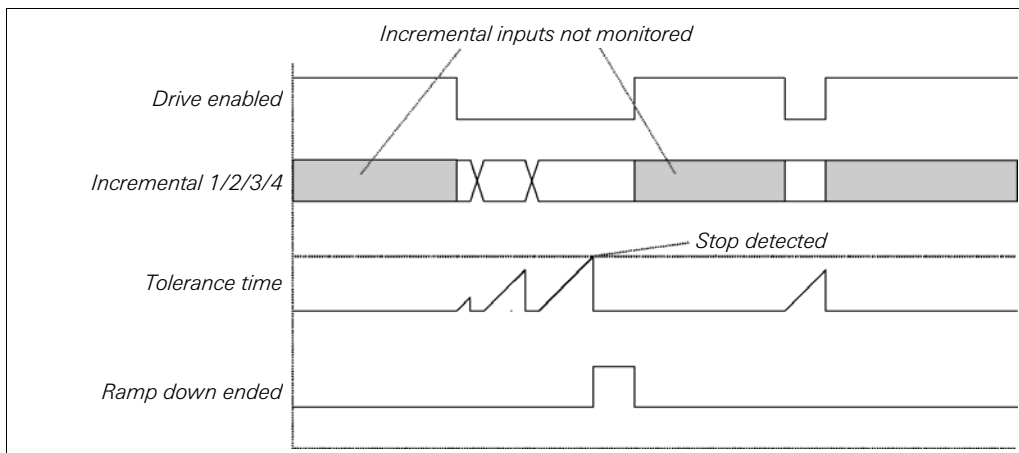
Logic example

Fig. 144: Logic example for the Ramp down detection function block



Sequence/timing diagrams

Fig. 145: Sequence/timing diagram for the Ramp down detection function block



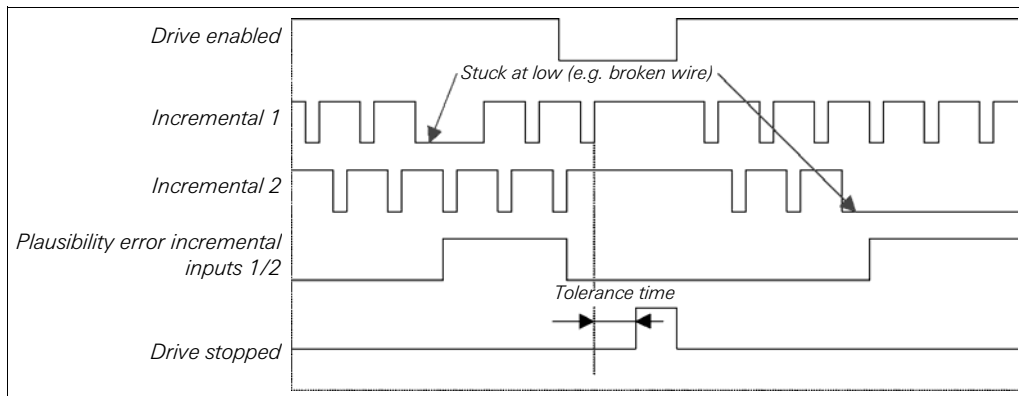


Fig. 146: Sequence/timing diagram for the Ramp down detection function block with plausibility check

8.7.13 Start warning

Function block diagram

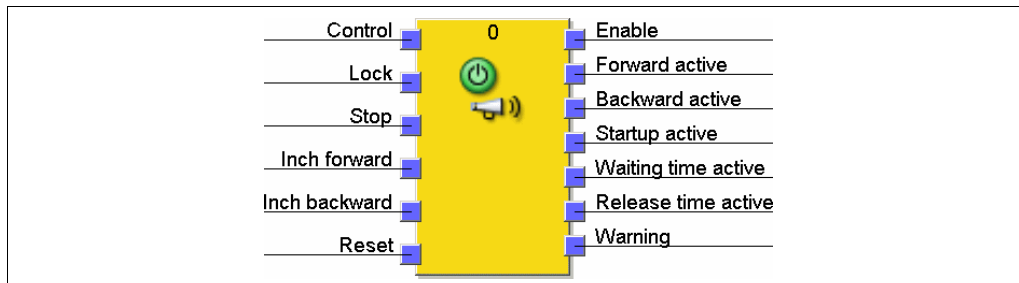


Fig. 147: Logic connections for the Start warning function block

General description

Many machines must be equipped with a start warning mechanism, e.g. if the machine operator can not see all dangerous areas from one location because of the size of the machine.

After a start button has been pressed, the waiting time starts and a warning signal is initiated. After expiration of the waiting time the release time starts and a second warning signal is initiated. During the release time it is possible to start the machine by pressing the start button a second time.

The start warning is required for automatic run mode as well as for the safe inch mode of the machine.

NOTE

Start sequence

1. On startup the function block is in *Inactive* mode. The **Startup active** output is High while all other outputs are Low.
2. If the **Control** input becomes Low and the **Lock** and **Stop** inputs are High, the start sequence is enabled and the function block goes into *Waiting for start* mode.
3. A rising edge on the **Inch forward** or on the **Inch backward** input will trigger the start sequence:
 - The **Startup active** output changes to Low, the waiting time and the signal time start and the **Waiting time active** output as well as the **Warning** output change to High for the duration of the signal time.
 - After expiration of the **Waiting time** the **Release time** and the **Impulse time** start. The **Waiting time active** output changes back to Low, the **Release time active** output changes to High and the **Warning** output changes to High again for the duration of the impulse time.
4. If *during the release time* a second rising edge of **Inch forward** occurs, the function block will transit to inch mode (*Forward*) and the **Enable** and **Forward active** outputs will change to High. Respectively, if during the release time an additional rising edge of **Inch backward** occurs, the function block will transit to inch mode (*Backward*), and the **Enable** and the **Backward active** outputs will change to High.

Logic programming – Function blocks

NOTE

For restrictions to this rule that apply in forward/backward locked mode see below.

5. If the release time has expired and no transition to inch mode has occurred, the function block will transit back to *Waiting for start* mode and a complete start sequence is required again.
6. The duration of the inch mode is not limited. It will be stopped if the High input (**Inch forward** or **Inch backward**) becomes Low again. In this case the **Enable** output and the **Forward active** or the **Backward active** output will become Low again. The inch mode will also be stopped if both inputs (**Inch forward** and **Inch backward**) become High at the same time. After the inch mode has been stopped, the release time starts again. This means that another rising edge on the **Inch forward** or on the **Inch backward** input will restart the inch mode immediately without a new start sequence. If the release time has expired and no transition to inch mode has occurred, the function block will transit back to *Waiting for start* mode and a complete start sequence is required again.
7. The inch mode will also be stopped by a falling edge on the **Reset** input or on the **Stop** input. In this case, the function block will transit back to *Waiting for start* mode and a complete start sequence is necessary again.

Parameters of the function block

Tab. 71: Parameters of the Start warning function block

Parameter	Possible values
Direction switching	<ul style="list-style-type: none"> • Locked • Not locked
Waiting time	1 to 60 s in steps of 10 ms. The value has to be greater than the logic execution time.
Release time	1 to 600 s in steps of 10 ms. The value has to be greater than the logic execution time.
Signal time	0 to 60 s in steps of 10 ms. If the value is not 0, it has to be greater than the logic execution time, but smaller than the Waiting time.
Impulse time	0 to 600 s in steps of 10 ms. If the value is not 0, it has to be greater than the logic execution time, but smaller than the Release time.

Direction switching

This parameter determines if it is possible to switch between the forward and backward direction with or without execution of the complete start sequence. If the setting **Not locked** is configured it is possible to initiate the start sequence with one of the inputs (e.g. **Inch forward**) and confirm the start sequence with the other input (e.g. **Inch backward**). In **Not locked** mode, it is also possible to switch the direction of the inch mode without the execution of the complete start sequence.

The setting **Locked** means that the start sequence must be confirmed (during the release time) with the *same* input (**Inch forward** or **Inch backward**) that has triggered the start sequence. A rising edge on the *other* input will restart the **Waiting time** instead. It is also not possible to switch between directions in inch mode. If the direction shall be changed a complete start sequence must be executed (see also Fig. 150).

Waiting time

The **Waiting time** parameter defines the time between the first rising edge of the **Inch forward** or **Inch backward** input and the start of the **Release time**.

Release time

After the **Waiting time** has expired, the **Release time** begins. During the **Release time** a rising edge on one of the **Inch forward/Inch backward** inputs will start the machine (depending on the setting of the **Direction switching** parameter).

Signal time

The **Signal time** starts at the same time as the **Waiting time**. During the **Signal time** the **Warning** output becomes High, indicating that a start sequence has been initiated.

Impulse time

The **Impulse time** starts at the same time as the **Release time**. During the impulse time the **Warning** output becomes High again, indicating that the inch mode can now be started. If during the **Impulse time** the inch mode is started this has no effect on the **Impulse time**, i.e. the **Warning** output will remain High until the configured **Impulse time** has expired.

The second warning signal is not mandatory and may be disabled by setting the impulse time to 0 s.

NOTE

Control input

A start sequence can only be initiated if the **Control** input is Low. If the **Control** input changes to High during a start sequence, the start sequence is aborted and another start sequence can only be initiated after the **Control** input has changed to Low again.

Lock input

A start sequence can only be initiated if the **Lock** input is High. If the **Lock** input changes to Low during a start sequence, the start sequence is aborted and another start sequence can only be initiated after the **Lock** input has changed to High again. This input can be used for safety stops.

If inch mode is active, a falling edge at the **Lock** input will stop inch mode and set the function block back into *Waiting for start* mode.

Stop input

A start sequence can only be initiated if the **Stop** input is High. If the **Stop** input changes to Low during a start sequence, the start sequence is aborted and another start sequence can only be initiated after the **Stop** input has changed to High again. This input can be used for safety stops.

If inch mode is active, a falling edge at the **Stop** input will stop inch mode and set the function block back into *Waiting for start* mode.

Inch forward/Inch backward

If a rising edge (transition from Low to High) is detected on the **Inch forward** or on the **Inch backward** input while the other input remains Low, the start sequence will begin.

A rising edge on both inputs or a rising edge on one of these inputs while the other input is High is considered an invalid input state. If this occurs during a start sequence (waiting time or release time running), these rising edges have no effect. If this occurs during inch mode, the inch mode will be stopped and the release time starts again.

NOTE

Reset

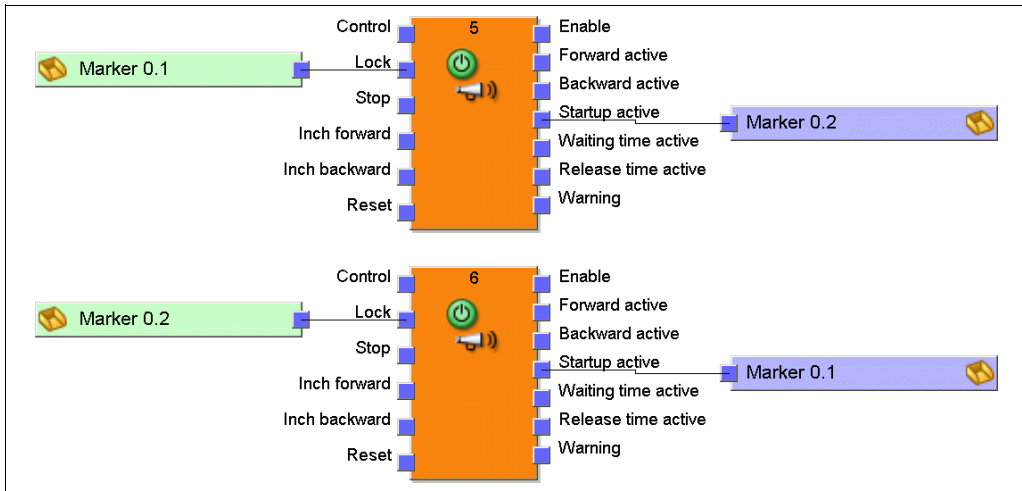
A falling edge on the **Reset** input restarts the start sequence. An active inch mode is stopped and the function block back goes into *Waiting for start* mode. The **Enable** output as well as the **Forward active** and the **Backward active** output will become Low while the **Startup active** output will become High.

Startup active output

The **Startup active** output is Low during the start sequence (waiting or release time is running) or if inch mode is active (the **Enable** output is High). The **Startup active** output can be used to lock other parallel Start warning function block instances. To this purpose, connect the **Startup active** output via a CPU marker to the **Lock** input of the other function block instance (see Fig. 148).

Logic programming – Function blocks

Fig. 148: Logic example for a combination of two Start warning function blocks

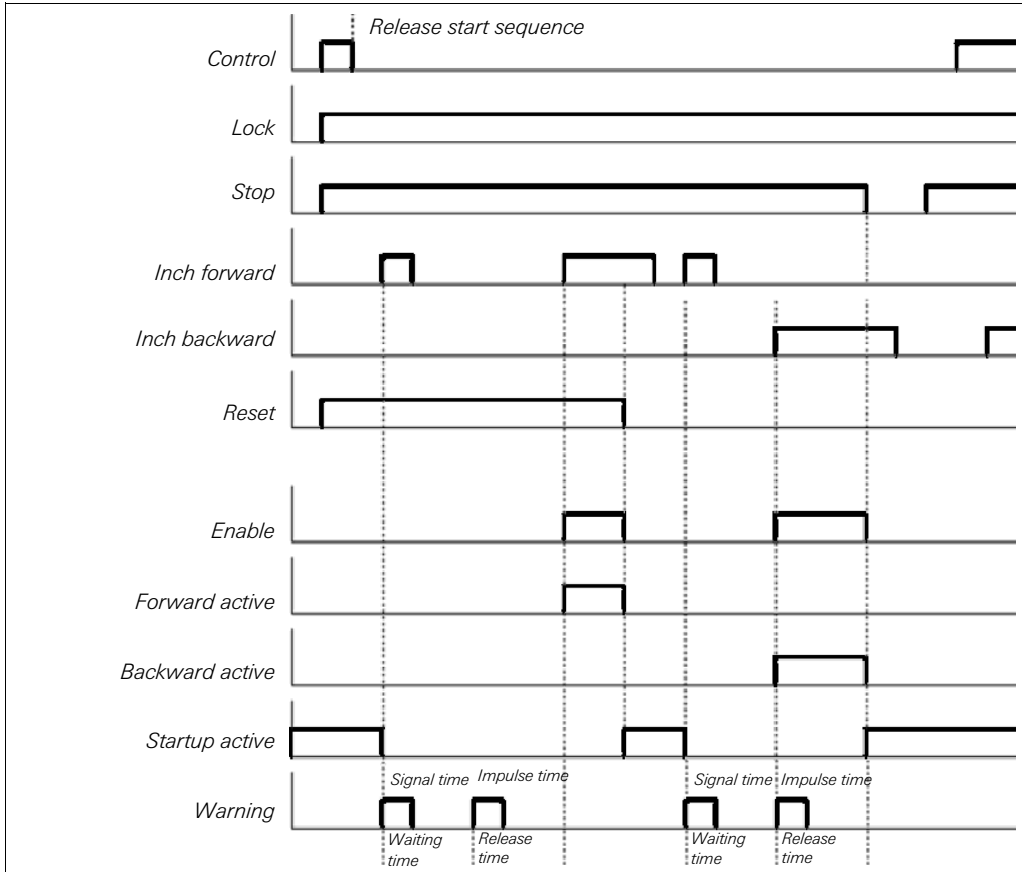


Waiting time active output and Release time active output

These outputs indicate whether the waiting time or the release time is active.

Sequence/timing diagrams

Fig. 149: Sequence/timing diagram for the Start warning function block in Not locked mode



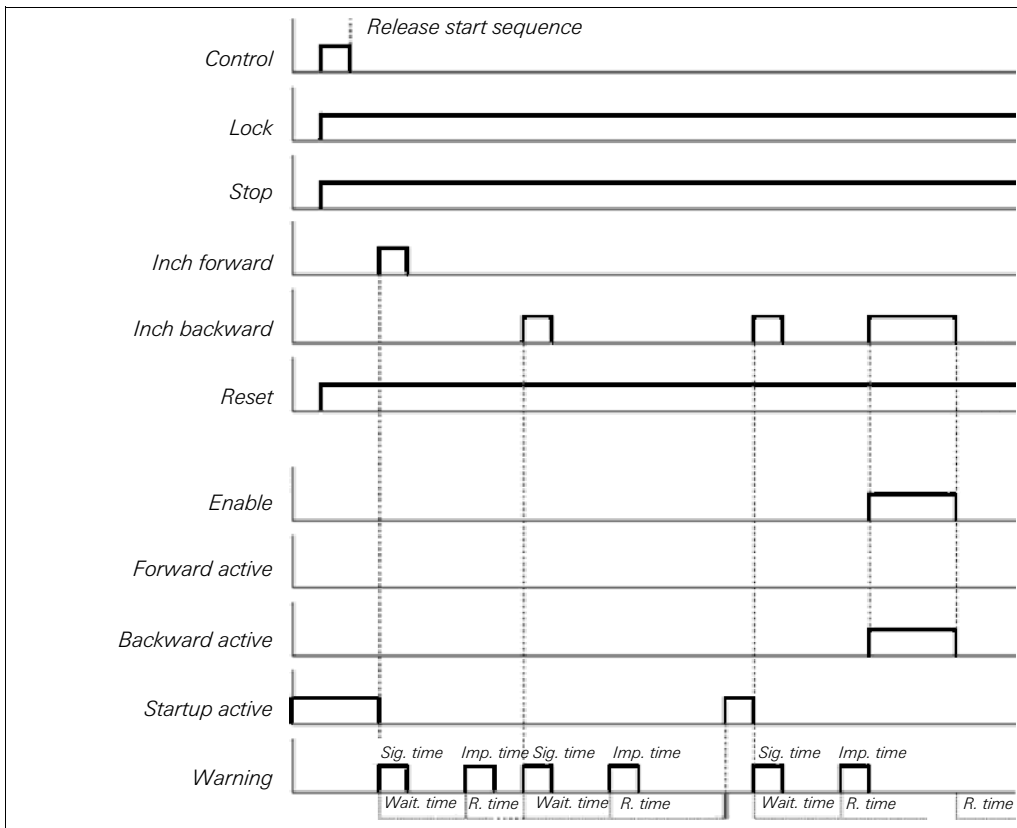


Fig. 150: Sequence/timing diagram for the Start warning function block in Locked mode

- The start sequence is started by a rising edge of the **Inch forward** input.
- A rising edge at the **Inch backward** input causes a restart of the waiting time during the start sequence.
- A rising edge at the **Inch backward** input causes a stop of the inch mode if the **Inch forward** input is High.

NOTES

8.8 Function blocks for dual channel evaluation

The samos[®]PRO system supports applications up to SIL3 (in accordance with EN 62061) and Performance Level (PL) e (in accordance with EN ISO 13-849-1). Possible sources for function block inputs are one or two safety signals connected locally to the samos[®]PRO safety controller. You can choose between the following input evaluations (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 the internal evaluation for the individual types of input signal evaluations of the samos[®]PRO safety controller.

Truth tables

The following applies for the truth tables in this section:

“0” means logic Low

“1” means logic High

“x” means “any” = “0” or “1”

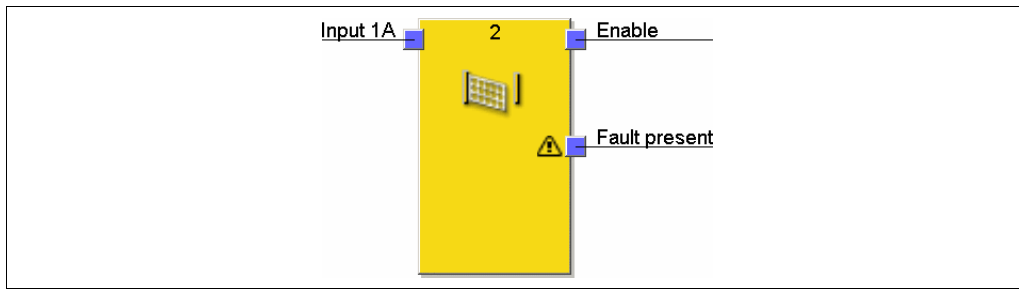
The Fault present output is High when the logic processing of the samos[®]PRO safety controller detects an error in the combination or in the sequence of the input signals.

NOTE

Logic programming – Function blocks

8.8.1 Single-channel evaluation

Fig. 151: Example for single-channel evaluation



NOTE

The following relates to the Safety gate monitoring and Emergency stop function blocks.

This evaluation type has no functional use since the **Enable** output always has the same value as **Input 1A** and the **Fault present** output is always Low. Therefore this option may be useful only for the graphic arrangement of the logic program. Regardless of that, the related input element can be used directly in the logic instead of connecting it to **Input 1A**.

8.8.2 Dual-channel evaluation (1 pair) and discrepancy time

NOTE

This section relates to the Safety gate monitoring, Emergency stop, Light curtain monitoring, Magnetic switch, Two hand control type IIIA and Two hand control type IIIC function blocks.

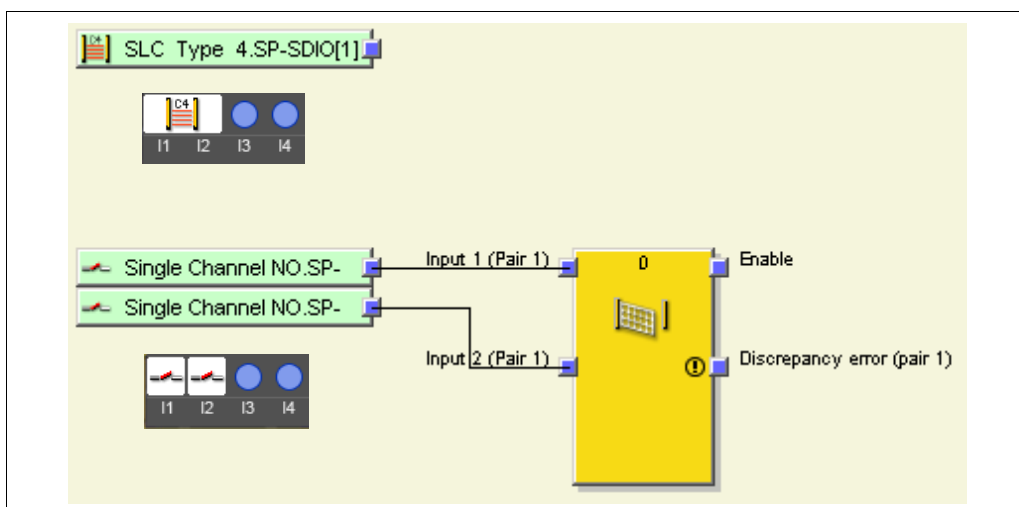
Note that the I/O extension modules, e.g. SP-SDIO or SP-SDI, can carry out a dual-channel evaluation when predefined input elements from the Elements window (e.g. SMA, SLC,...) are connected to them. If such an input element is selected, you do not need a separate function block for dual-channel evaluation (e.g. light curtain monitoring, safety gate monitoring or magnetic switch). For detailed information on discrepancy monitoring on the I/O modules see section 9.1 "Dual channel evaluation and discrepancy time" on page 193.

Alternatively, you can connect non pre-evaluated input signals to both input channels of a function block with a dual-channel input configuration. In this case the dual-channel evaluation takes place in the function block.

The disadvantage of this alternative is that it requires one function block more in the logic which can lead to a higher logic execution time. The advantage is that a discrepancy time error is made available via the output of the function block and can be evaluated in the logic.

The following function blocks generate the same output value for a dual-channel input signal that was pre-evaluated by the I/O device.

Fig. 152: Dual-channel evaluation with I/O module or with function block



The dual-channel evaluation evaluates the correct sequence of the two input signals. It is expected that if one of the two signals has caused a switching off, the other signal will follow accordingly. Which values the two signals must have depends on the type of the dual-channel evaluation. There are two possibilities:

- Equivalent evaluation
- Complementary evaluation

An optional discrepancy time can be configured. The discrepancy time defines for how long the two inputs may have discrepant values after one of the both input signals has changed without this being considered as an error.

The following truth table describes the discrepancy conditions for the dual-channel equivalent and the dual-channel complementary input evaluation:

Evaluation type	Input A	Input B	Discrepancy timer ¹⁾	Evaluation status	Enable output	Discrepancy error output
Equivalent	0	0	0	Inactive	0	Unchanged ²⁾
	0	1	< Discrepancy time	Discrepant	0	Unchanged
	1	0	< Discrepancy time	Discrepant	0	Unchanged
	1	1	0	Active ³⁾	1	0
	x	x	≥ Discrepancy time (timeout)	Error	0	1
Complementary	0	1	0	Inactive	0	Unchanged
	0	0	< Discrepancy time	Discrepant	0	Unchanged
	1	1	< Discrepancy time	Discrepant	0	0
	1	0	0	Active ³⁾	1	Unchanged
	x	x	≥ Discrepancy time (timeout)	Error	0	1

For the change between the different states of the dual-channel evaluation the following rules apply:

Tab. 72: Dual-channel evaluation

A dual-channel evaluation can only change to Active (**Enable** output changes from Low to High), if ...

- since the last Active status at least once the status was Inactive, i.e. it is not possible to switch from Active to Discrepant and back to Active, and
- the discrepancy time has not elapsed or the discrepancy time is deactivated, and
- after the state change of the samos[®]PRO system from the Stop state to the Run state the status has been at least once Inactive. So if at the time of the transit to the Run state the input values already stand for the Active status, the **Enable** output remains Low nevertheless.

A discrepancy error (timeout) is reset, if the Active status has been reached, i.e. the **Enable** output changes to High.

When defining values for the discrepancy time the following must be observed: The discrepancy time ...

NOTE

- must be greater than the logic execution time,
- have a tolerance of +/- 10 ms in addition to the logic execution time. The logic execution time depends on the number and type of function blocks used and is shown in the samos[®]PLAN in the logic editor on the **FB info** tab and also in the report.
- If signals from tested sensors are connected to SP-SDI or SP-SDIO modules the discrepancy time should be at least the set *Test gap [ms]* plus the *Max. off-on delay [ms]*, because a signal change at the module input can be delayed for this time. Both values are displayed in the samos[®]PLAN report for the used test output.
- If both inputs of a pair are connected to the same input signal, the evaluation corresponds to the single-channel evaluation, i.e. no equivalence check or antivalence check and no discrepancy time monitoring is carried out.

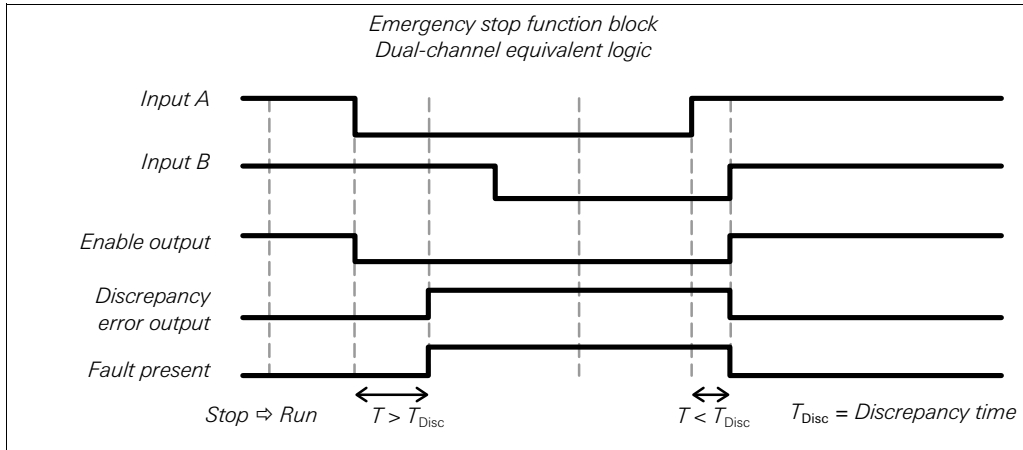
¹⁾ If the discrepancy time is active (>0), the discrepancy timer is restarted on the first signal change resulting in a discrepant status. If the discrepancy time is inactive (=0), the discrepancy timer does not start, i.e. a timeout will never occur.

²⁾ Unchanged = Last status is preserved.

³⁾ If the correct sequence has been observed.

Sequence/timing diagram

Fig. 153: Sequence/timing diagram for the Emergency stop function block

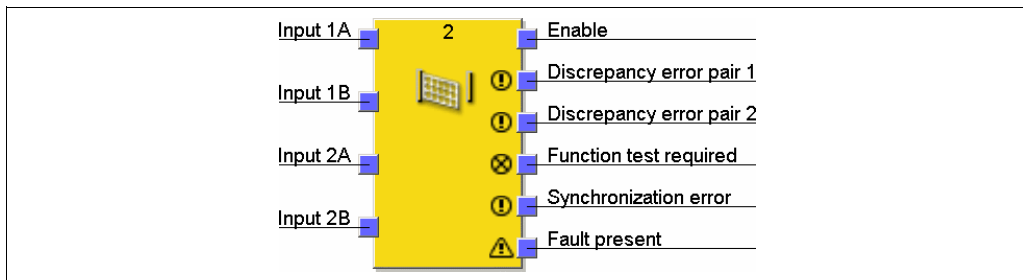


8.8.3 Double dual-channel evaluation (2 pair synchronization evaluation) and synchronization time

NOTE

This section relates to the Safety gate monitoring and Two hand control type IIC function blocks.

Fig. 154: Double dual-channel evaluation with the Safety gate monitoring function block



The double dual-channel evaluation (synchronization evaluation) evaluates the correct sequence of the two input signals for each of the two input pairs as described in chapter 8.8.2 "Dual-channel evaluation (1 pair) and discrepancy time" on page 124. Additionally the correct sequence of the two dual-channel evaluations in relation to each other is monitored. It is expected that if one of the two dual-channel evaluations has caused a switching off, the other dual-channel evaluation will follow accordingly.

An optional synchronization time can be defined. The synchronization time defines for how long the two dual-channel evaluations may have not synchronous states without this being considered as an error.

The synchronization time differs from the discrepancy time: It evaluates the relation between the two dual-channel evaluations while the discrepancy time applies to an input pair of one dual-channel evaluation.

The following truth table describes the synchronization conditions for double dual-channel evaluations (2 pairs):

Status of the dual-channel evaluation pair 1	Status of the dual-channel evaluation pair 1	Synchronization timer ¹⁾	Synchronization status	Enable output	Synchronization error output
Inactive or discrepant	Inactive or discrepant	0	Inactive	0	Unchanged ²⁾
Inactive or discrepant	Active	< Synchronization time	Discrepant	0	Unchanged
Active	Inactive or discrepant	< Synchronization time	Discrepant	0	Unchanged
Active	Active	0	Active ³⁾	1	0
x	x	≥ Synchronization time (timeout)	Error	0	1

For the change between the different states of the double dual-channel evaluation (synchronization evaluation) the following rules apply:

The synchronization evaluation can only change to Active (**Enable** output changes from Low to High), if ...

- since the last Active synchronization status at least once the status was Inactive. For the Two hand control type IIC function block both dual-channel evaluations must be Inactive at the same time, for the Safety gate monitoring function block this can occur at different times as well. It is not possible to switch from Active to Discrepant and back to Active.
- the synchronization time has not elapsed or the synchronization time is deactivated, and
- after the state change of the samos[®]PRO system from the Stop state to the Run state the synchronization status has been at least once Inactive. So if at the time of the transit to the Run state the inputs already stand for the Active status, the **Enable** output remains Low nevertheless.

A synchronization error (timeout) is reset, if the Active synchronization status has been reached, i.e. the **Enable** output changes to High.

When defining values for the synchronization time the following must be observed: The times ...

- must be greater than the logic execution time,
- have a tolerance of +/- 10 ms in addition to the logic execution time. The logic execution time depends on the number and type of function blocks used and is shown in the samos[®]PLAN in the logic editor on the **FB info** tab and also in the report.
- If signals from tested sensors are connected to SP-SDI or SP-SDIO modules, the synchronization time should be at least the set *Test gap [ms]* plus the *Max. off-on delay [ms]*, because a signal change at the module input can be delayed for this time. Both values are displayed in the samos[®]PLAN report for the used test output.

Tab. 73: Double dual-channel evaluation (synchronization evaluation)

NOTE

¹⁾ If the synchronization time is active (>0), the synchronization timer is restarted on the first signal change resulting in a discrepant synchronization status. If the synchronization time is inactive (=0), the synchronization timer does not start, i.e. a timeout will never occur.

²⁾ Unchanged = Last status is preserved.

³⁾ If the correct sequence has been observed.

For further information on the behaviour of this function block please refer to section 8.8.2 “Dual-channel evaluation (1 pair) and discrepancy time” on page 124.

8.8.5 Magnetic switch

Function block diagram

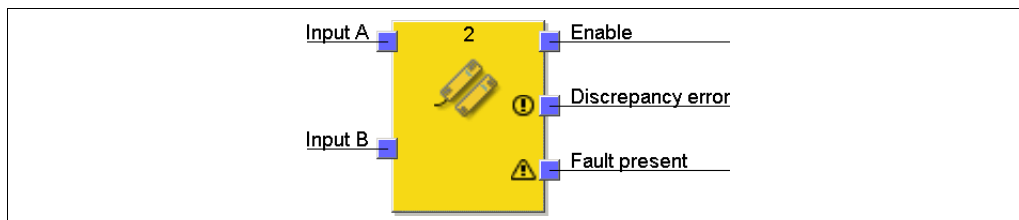


Fig. 157: Logic connections for the Magnetic switch function block

General description

The internal logic of the Magnetic switch function block corresponds to the functionality of the Emergency stop function block, only with a limited parameter selection. The function block allows graphic differentiation in accordance with the application.

The Magnetic switch function block is a predefined function block for reed switches or other sensors for which discrepancy time monitoring is required. When the evaluation of the complementary inputs is High, the **Enable** output is High (see section 8.8.2 “Dual-channel evaluation (1 pair) and discrepancy time” on page 124).

Parameters of the function block

Parameter	Possible values
Inputs	<ul style="list-style-type: none"> Dual-channel equivalent Dual-channel complementary
Discrepancy time	10 to 3000 ms in 10 ms steps. The value has to be greater than the logic execution time.
Number of outputs	<ul style="list-style-type: none"> 1 (Enable output) 2 (Enable output and Discrepancy error output)
Use Fault present	<ul style="list-style-type: none"> With Without

Tab. 75: Parameters of the Magnetic switch function block

8.8.6 Light curtain monitoring

Function block diagram

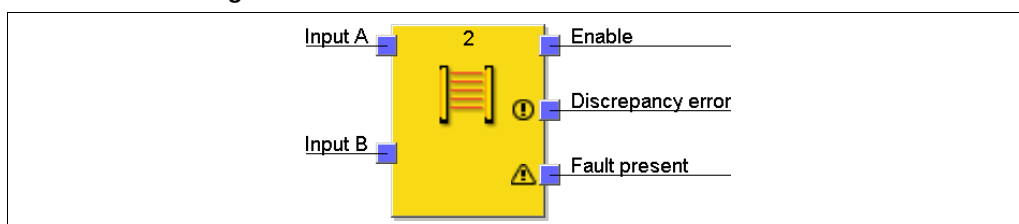


Fig. 158: Function block diagram for the Light curtain monitoring function block

General description

The Light curtain monitoring function block allows the implementation of a semiconductor protective device functionality with ESPE.

The internal logic of the Light curtain monitoring function block corresponds to the functionality of the Emergency stop function block, however with a limited parameter selection. The single-channel input type cannot be selected in the Light curtain monitoring function block. When the evaluation of the complementary inputs is High, the **Enable** output is High (see section 8.8.2 “Dual-channel evaluation (1 pair) and discrepancy time” on page 124).

If a corresponding dual-channel input element is configured in the hardware configuration of the samos®PLAN, this function block is no longer required in the logic since the pre-

NOTE

Logic programming – Function blocks

evaluation is then carried out directly on the extension module (e.g. SP-SDI or SP-SDIO module). But if the **Fault present** output is required for further processing, the function block can be used to this purpose. To this purpose the two input signals have to be configured as single-channel signals and applied to the inputs of the function block.

Parameters of the function block

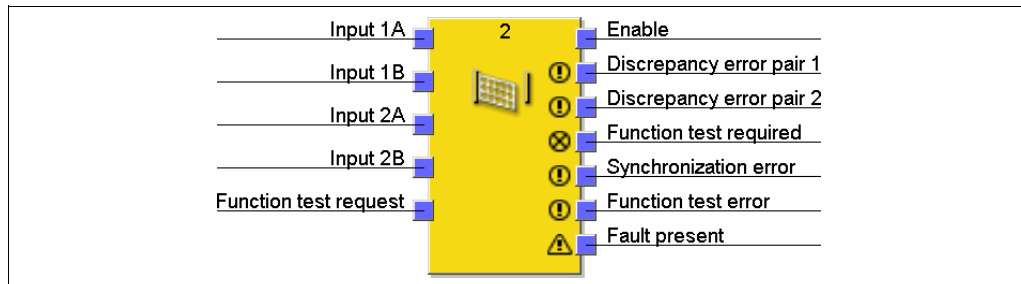
Tab. 76: Parameters of the Light curtain monitoring function block

Parameter	Possible values
Input type	Dual-channel equivalent
Discrepancy time	0 = disabled, 10 to 500 ms in 10 ms steps. If enabled, the value has to be greater than the logic execution time.
Number of outputs	<ul style="list-style-type: none"> • 1 (Enable output) • 2 (Enable output and Discrepancy error output)
Use Fault present	<ul style="list-style-type: none"> • With • Without

8.8.7 Safety gate monitoring

Function block diagram

Fig. 159: Function block diagram for the Safety gate monitoring function block



General description

The function block can be used for the evaluation of dual-channel switches. 1 pair or 2 pairs can be selected. For the behaviour of the dual-channel evaluation see section 8.8.2 “Dual-channel evaluation (1 pair) and discrepancy time” on page 124 and section 8.8.3 “Double dual-channel evaluation (2 pair synchronization evaluation) and synchronization time” on page 126.

Additionally the function block allows an optional function test monitoring.

Parameters of the function block

Tab. 77: Parameters of the Safety gate monitoring function block

Parameter	Possible values
Inputs	<ul style="list-style-type: none"> • Single-channel • Dual-channel equivalent (1 pair) • Dual-channel complementary (1 pair) • Dual-channel equivalent (2 pairs) • Dual-channel complementary (2 pairs)
Function test	<ul style="list-style-type: none"> • No function test • Function test required
Discrepancy time pair 1 Discrepancy time pair 2	Can be set separately for the inputs 1A/1B and 2A/2B. Values: 0 = disabled, 10 to 30,000 ms in 10 ms steps. If enabled, the value has to be greater than the logic execution time.
Synchronization time	0 = disabled, 10 to 30,000 ms in 10 ms steps. If enabled, the value has to be greater than the logic execution time.
Number of outputs	1 to 6
Use Fault present	<ul style="list-style-type: none"> • With • Without

Function test

In some applications, safeguarding devices require cyclic physical testing in order to verify that the device continues to operate properly.

If the Safety gate monitoring function block is configured with the *Function test required* parameter, the input signal(s) must change once per machine cycle in a way that no enable condition exists anymore and back (e.g. as a result of opening and closing of a safety gate).

Typically the **Function test request** input is connected to the machine cycle contact.

If according to the configuration a function test is required, this has to be performed under the following conditions:

- after the samos[®]PRO system has changed from the Stop state to the Run state, and
- after each rising edge (Low to High) at the **Function test request** input.

This is indicated by a High signal at the **Function test required** output. The **Function test required** output changes back to Low, if a signal sequence occurs at the inputs that causes the **Enable** output to change from Low to High, before the next rising edge at the **Function test request** input occurs.

The **Function test error** output becomes High and the **Enable** output becomes Low, if the next machine cycle starts before a function test has been performed, i.e. if the **Function test required** output is still High and another rising edge (Low to High) at the **Function test request** input occurs.

The Function test error output changes back to Low, if a signal sequence occurs at the inputs that causes the **Enable** output to change from Low to High.

Sequence/timing diagram

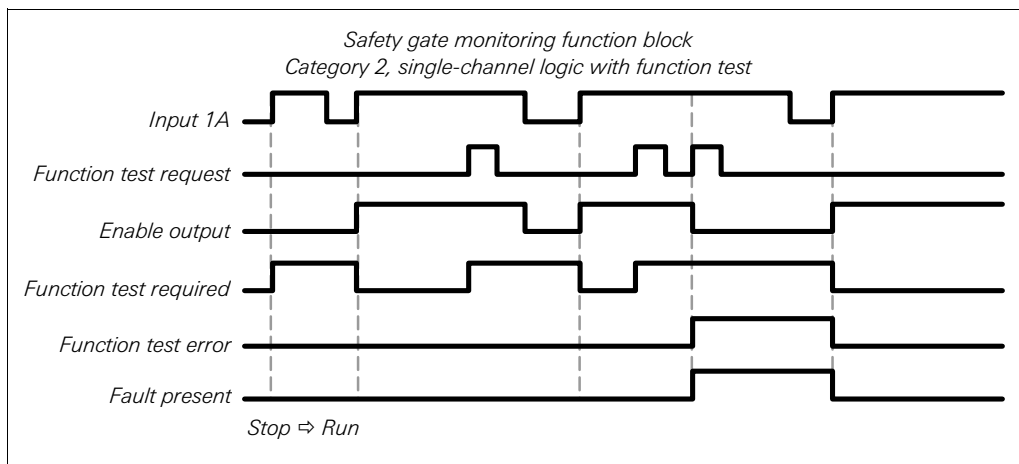


Fig. 160: Sequence/timing diagram for the Safety gate monitoring function block, Category 2, single-channel with function test

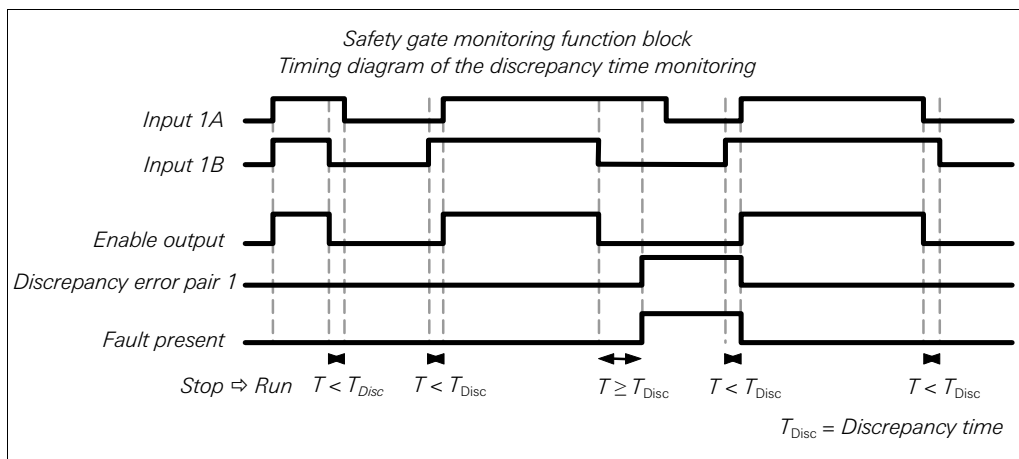


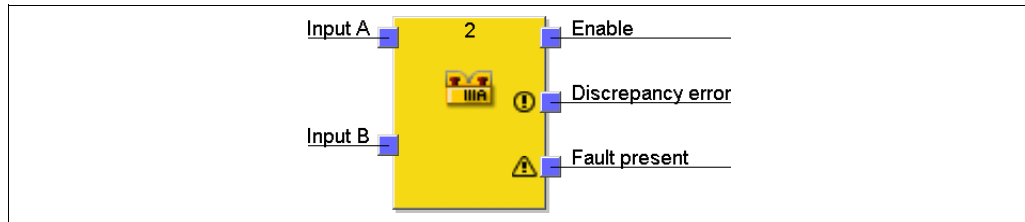
Fig. 161: Sequence/timing diagram for the Safety gate monitoring function block, Category 4, dual-channel equivalent (1 pair) without function test

Logic programming – Function blocks

8.8.8 Two-hand control type IIIA

Function block diagram

Fig. 162: Function block diagram for 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 sensors that require a discrepancy time monitoring of equivalent inputs. The internal logic of the Two-hand control type IIIA function block corresponds to the functionality of the Emergency stop function block, only with a limited parameter selection. The function block allows graphic differentiation in accordance with the application.

Input A and **Input B** form a dual-channel evaluation and must be equivalent. When the evaluation of the inputs is High, the **Enable** output is High (see section 8.8.2 “Dual-channel evaluation (1 pair) and discrepancy time” on page 124).

The discrepancy time is set to 500 ms (the discrepancy time is fixed and cannot be changed).

Parameters of the function block

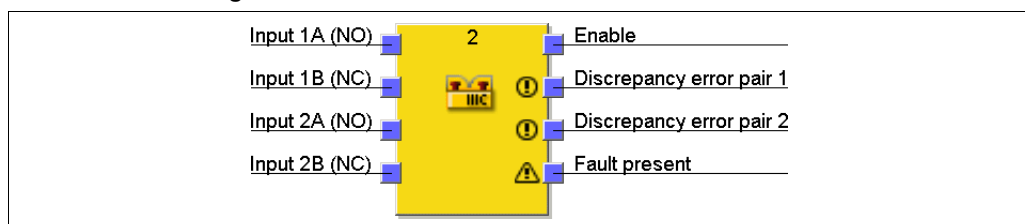
Tab. 78: Parameters of the Two hand control type IIIA function block

Parameter	Possible values
Inputs	Fixed value: Dual channel equivalent
Discrepancy time	Fixed value: 500 ms
Number of outputs	<ul style="list-style-type: none"> • 1 (Enable output) • 2 (Enable output and Discrepancy error output)
Use Fault present	<ul style="list-style-type: none"> • With • Without

8.8.9 Two-hand control type IIIC

Function block diagram

Fig. 163: Function block diagram for the Two hand control type IIIC function block



General description

The Two hand control type IIIC function block provides the logic for monitoring the inputs of a two-hand control in accordance with EN 574.

Use the Two hand control type IIIC function block only in combination with an SP-SDIO or an SP-SDI module!

The Two hand control type IIIC function block requires the usage of an SP-SDIO or an SP-SDI module. Otherwise the requirements of EN 574 are not fulfilled.



In the hardware configuration the used inputs must be configured as single channel signals, i.e. no dual channel input evaluation on the extension module.

Parameters of the function block

Parameter	Possible values
Discrepancy time (pair 1)	0 = disabled, 10 to 500 ms in 10 ms steps. If enabled, the value has to be greater than the logic execution time.
Discrepancy time (pair 2)	0 = disabled, 10 to 500 ms in 10 ms steps. If enabled, the value has to be greater than the logic execution time.
Synchronization time	Fixed value: 500 ms
Number of outputs	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • With • Without

Tab. 79: Parameters of the Two hand control type IIC function block

The function block evaluates its input signals in pairs. **Input 1A** and **Input 1B** form a dual-channel evaluation and have to be complementary. **Input 2A** and **Input 2B** form a dual-channel evaluation and also have to be complementary. A discrepancy time can be specified for each of both input pairs.

The synchronization time is the time during which the *input pairs* may have different values. As specified in the standards and regulations, the synchronization time for a two-hand switch evaluation may not exceed 500 ms (the synchronization time is preset and cannot be changed).

For the behaviour of the double dual-channel evaluation see section 8.8.2 “Dual-channel evaluation (1 pair) and discrepancy time” on page 124 and section 8.8.3 “Double dual-channel evaluation (2 pair synchronization evaluation) and synchronization time” on page 126.

The synchronization evaluation for the Two hand control type IIC function block differs from the Safety gate monitoring function block with regard to the condition for the Inactive synchronization status. For the Two hand control type IIC function block, both dual-channel evaluations must be Inactive, i.e. the inputs A/B of both input pairs must be Low/High at the same time.

Furthermore the Two hand control type IIC function block has no **Synchronization error** output, because with a two hand control it is not regarded as an error if not both manual switches are actuated within the specified 500 ms. However this synchronization time may not be exceeded, because otherwise the **Enable** output will not change to High.

Sequence/timing diagram

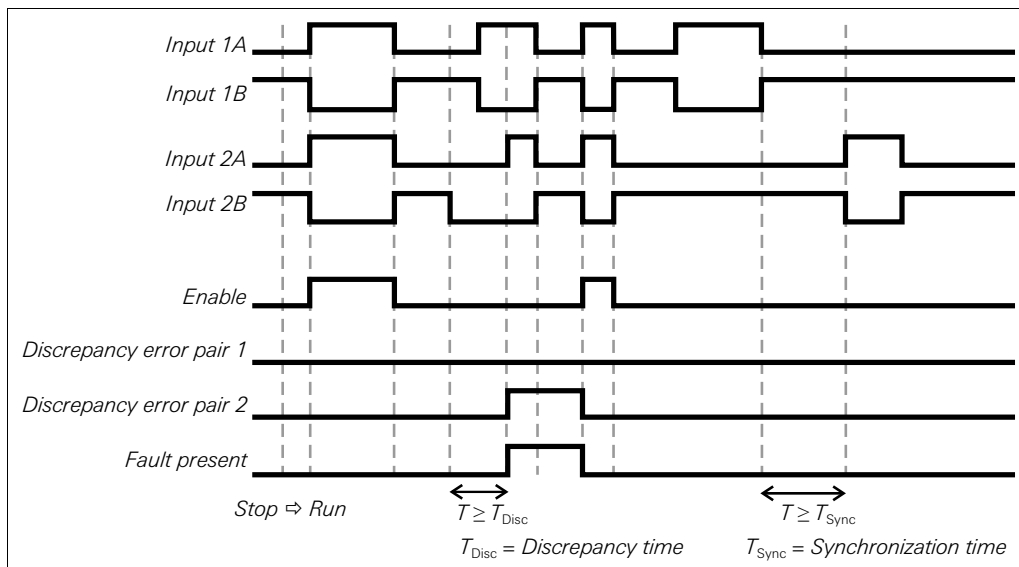
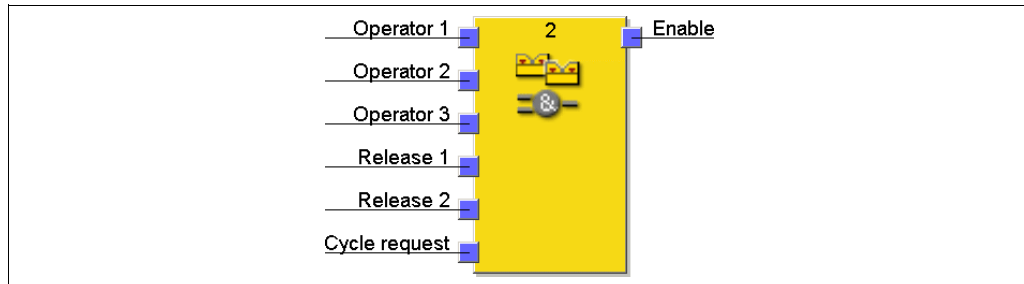


Fig. 164: Sequence/timing diagram for the Two hand control type IIC function block

8.8.10 Multi operator (multiple two-hand control)

Function block diagram

Fig. 165: Logic connections for the Multi operator function block



General description

The Multi operator function block is used to monitor simultaneous operation of up to three two-hand controls. For example, several two-hand controls or foot switches can be necessary in a press application with more than one operator in order to trigger the downward movement of the press. Typically, each **Operator** input is connected to a Two-hand control function block.

Release inputs (e.g. safety light curtains) can be connected optionally in order to ensure that the assigned devices are High before the **Enable** output can become High. Resetting and restarting need to be handled independently of this function block.

The **Cycle request** input can be used to enforce that each connected two hand control is released at least once before another start is possible. Typically this input is connected to a signal that generates a pulse with each machine cycle. In this way it can be prevented that one or more of the two hand controls remain actuated permanently.



The Operator inputs and the Release inputs must be pre-evaluated signals!

- Connect only safe pre-evaluated signals to the Operator inputs, e.g. the **Enable** output of a Two-hand control type IIIA or Two-hand control type IIIC function block. A safety-relevant evaluation of the inputs of a two-hand control has to be effected either by another function block (e.g. Two-hand control or Light curtain monitoring) or as a part of the configuration of the safety inputs (e.g. configuration of the inputs with dual-channel evaluation).
- The **Cycle request** input must not be used for safety functionality. This input is for automation control functionality only.

Parameters of the function block

Tab. 80: Parameters for the Multi operator function block

Parameter	Possible values
Cycle request condition	<ul style="list-style-type: none"> • Rising edge • Falling edge
Number of operators	<ul style="list-style-type: none"> • 2 operators • 3 operators
Number of release inputs	<ul style="list-style-type: none"> • 0 • 1 • 2

The **Enable** output changes to High, if ...

- all **Release** inputs are High and stay High, and
- each activated **Operator** input has changed to Low at least once (this may occur at different times as well) after the samos®PRO system has changed from the Stop state to the Run state or after a rising or a falling edge (depending on the configuration) has been detected at the **Cycle request** input, and
- all activated **Operator** inputs have changed to High subsequently.

The **Enable** output changes to Low, if ...

- one or more of the **Release** inputs is Low, or
- one or more of the **Operator** inputs is Low, or

- a rising or a falling edge (depending on the configuration) has been detected at the **Cycle request** input.

Sequence/timing diagram

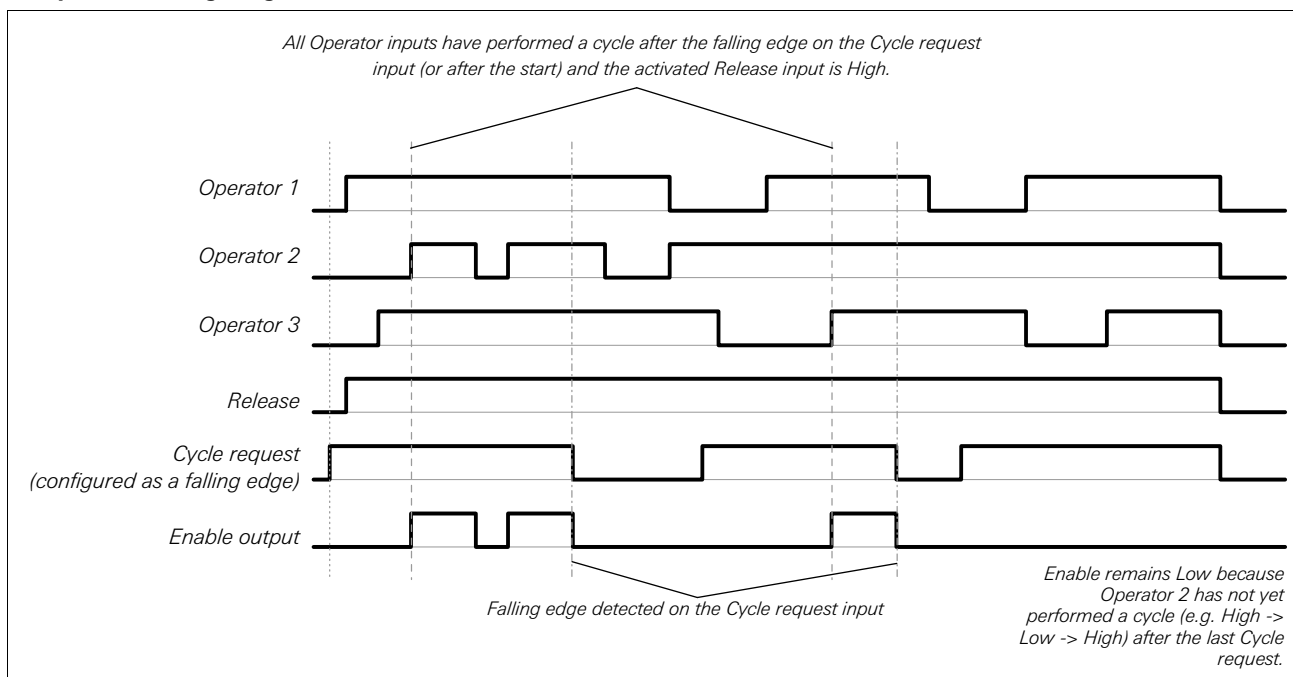


Fig. 166: Sequence/timing diagram for the Multi operator function block

8.9 Function blocks for Parallel muting, Sequential muting and Cross muting

8.9.1 Overview and general description

Muting is the automatic temporary suppression of safety-oriented area monitoring using electro-sensitive protective equipment (ESPE) while certain objects, e.g. pallets with material, are moved into the hazardous area.

Muting sensors monitor the presence of the material while it is being transported. Careful selection of the type and layout of the sensors makes it possible to differentiate between objects and persons.

In combination with the muting sensors and the ESPE the transported object generates an exactly defined signal sequence while it is moved through the hazardous area. The muting sensors have to ensure that all dangers are excluded when a person enters an area protected by the ESPE (i.e. any dangerous state has to be terminated immediately). It has to be impossible for a person to generate the same signal sequence as a transported object.

The placement of the muting sensors is determined by the form of the object to be detected. To this purpose the following options are, amongst others, available with differing numbers of sensor input signals:

- two sensors
- two sensors and an additional signal C1
- four sensors (two sensor pairs)
- four sensors (two sensor pairs) and an additional signal C1

Muting sensor signals can be generated by the following external sensors:

- optical sensors
- inductive sensors
- mechanical switches
- signals from the control system

If you use optical sensors for muting applications, use sensors with a background suppression in order to ensure that only the transported material fulfils the muting condition. These sensors detect material only up to a specific distance. Objects that are further away can therefore not fulfil the input conditions of the muting sensors.

Three different function blocks are available for muting:

- Parallel muting (muting with two parallel sensor pairs)
- Sequential muting (muting with two sequential sensor pairs)
- Cross muting (muting with one crossed sensor pair)

NOTES

The muting cycle is the specified sequence of all processes that are executed during muting.

- The muting cycle begins when the first muting sensor is activated. The muting cycle ends depending on the configuration in the function block for the muting end condition. It is not possible to activate muting again until the preceding muting cycle has been terminated.
- Material can be transported several times within one muting cycle if the muting conditions are maintained permanently in the process, meaning that at least one pair of sensors remain activated permanently.

Since muting bypasses the safety functions of a protective device, several requirements have to be fulfilled, as shown below, in order to ensure the safety of the application.



The general safety regulations and protective measures have to be observed!

If you use muting, be sure to observe the following information about the correct use of muting:

- Access to the hazardous area has to be detected reliably by the ESPE or be excluded through other measures. It has to be impossible for a person to pass by, pass over, pass under or cross the ESPE without being detected. Observe the operating instructions of the ESPE for the correct installation and use of the device.
- Always observe the valid applicable local, regional and national regulations and standards applying to your application. Ensure that your application conforms to an appropriate risk analysis and avoidance strategy.
- Muting may never be used to transport a person into the hazardous area.
- Mount the control devices for resetting and overriding outside the hazardous area so that they cannot be actuated by a person located in the hazardous area. Furthermore, when operating a control device, the operator must have full visual command of the hazardous area.
- The muting sensors have to be located in such a way that the hazardous area can only then be accessed after an intervention in the protective field, if the dangerous state has been terminated before. One condition here is that the required safety distances defined in prEN ISO 13·855 are observed. At least two muting signals that are independent of each other are required.
- Muting may only be activated for the period in which the object that triggered the muting condition blocks access to the hazardous area.
- The area between the ESPE and the muting sensors must be protected against standing behind:
 - For Parallel muting between ESPE and sensors A1/A2 and between ESPE and sensors B1/B2 (see Fig. 171).
 - For Sequential muting between ESPE and sensor A2 and between ESPE and sensor B1 (see Fig. 174).
 - For Cross muting between ESPE and sensor A1 and between ESPE and sensor A2 (see Fig. 177)
- Muting has to be carried out automatically, but may not depend on a single electrical signal.
- The material to be transported has to be detected along the entire length, meaning that an interruption of the output signals may not occur (see Suppression of sensor signal gaps).
- Muting must be triggered by at least two independently wired signals (e.g. by muting sensors) and may not depend completely on software signals (e.g. from a PLC).

- The muting condition has to be terminated immediately after the passage of the object so that the protective device returns to its normal state that was bypassed by muting (i.e. so that it comes back into force).
- The muting sensors have to be positioned in such a way that muting cannot be triggered unintentionally by anyone (see Fig. 167).

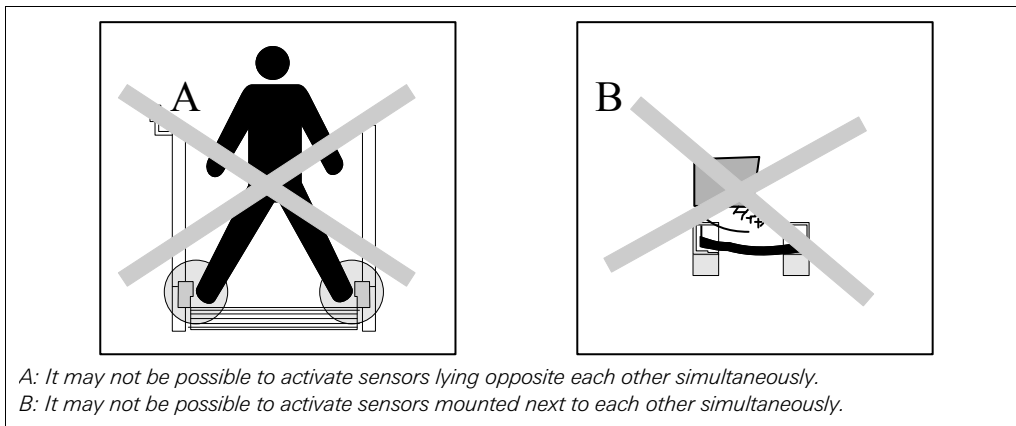


Fig. 167: Safety when mounting the muting sensors

- Always position the muting sensors in such a way that only the material is detected and not the conveyance means (pallet or vehicle).

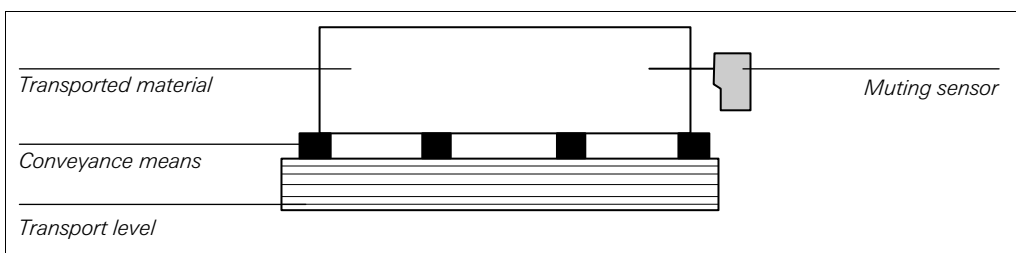


Fig. 168: Detection of material during muting

- Always position muting sensors in such a way that the material can pass unimpeded, but persons are detected reliably.
- Always position the muting sensors in such a way that a minimum distance to the detection area of the ESPE (e.g. to the light beams of a light curtain) is observed while the material is being detected.
- It has to be ensured that no persons are within the hazardous area before and during the activation of an override.
- Before you activate the override ensure that the equipment is in a perfect condition, in particular the muting sensors (visual inspection).
- When it has been necessary to activate an override, subsequently check whether the equipment functions properly and the layout of the muting sensors.
- During long muting cycles (i.e. longer than 24 hours) or during longer machine down-times check that the muting sensors function correctly.
- A muting and/or override lamp has to be used in order to signal that the muting or override function is active. It is possible to use an external muting/override lamp or one that is integrated in the protective device (ESPE).
- It may be necessary to monitor the muting/override lamp depending on your local, regional and national regulations and standards. If this is the case, this has to be realized by additional means. SP-SDIO and SP-SDI modules do not support lamp monitoring.
- Always position the muting or override lamp so that it can be seen well! It must be possible to see the muting or override lamp from all positions around the hazardous area and for the system operator.
- If safety-relevant information (i.e. distributed safety input values and/or safety output values) is transferred via a safety field bus network, always take the corresponding delays into account. These delays can influence both the system behaviour as well as the requirements for the minimum safety distances that are connected to the response times.

Logic programming – Function blocks

- When an override input is configured, test pulse outputs may not be used for the configuration of the safety inputs.
- Separate lines have to be used for the sensor signals A1 and A2 (B1 and B2).
- A line that is independent of other input signals has to be used for the signals for Reset and Reset required in order to exclude unintentional resetting of the system. The line must furthermore be laid protected.
- The total muting time cannot be set to indefinite (inactive) without additional precautions being taken. If the total muting time is set to indefinite, additional measures have to be taken to ensure that no one can access the hazardous area while muting is activated.

8.9.2 Parameters of the function blocks

The following table lists the possible configuration parameters of the muting function blocks.

Tab. 81: Parameters of the muting function blocks

Parameter	Possible values
Direction detection	<ul style="list-style-type: none"> • Disabled Only with Parallel muting and Sequential muting: <ul style="list-style-type: none"> • Forward (A1/A2 first) • Backward (B1/B2 first)
Sequence monitoring	Not selectable. Defined by selection of the muting function block: <ul style="list-style-type: none"> • Enabled: With Sequential muting • Disabled: With Parallel muting or Cross muting
Condition of other sensor pair for muting start	<ul style="list-style-type: none"> • Both inputs are clear Only with Parallel muting and Sequential muting: <ul style="list-style-type: none"> • At least one sensor is clear
Muting end condition	<ul style="list-style-type: none"> • With muting sensor pair • With ESPE
Total muting time	0 = disabled, 5 s to 3600 s, resolution 1 s
Additional muting time when ESPE is clear	0 ms, 200 ms, 500 ms, 1000 ms
Suppression of sensor signal gaps	0 = disabled, 10 to 1000 ms, resolution 10 ms. If enabled, the value has to be greater than the logic execution time.
Concurrency monitoring time	0 = disabled, 10 to 3000 ms, resolution 10 ms. If enabled, the value has to be greater than the logic execution time.
C1 input	<ul style="list-style-type: none"> • With • Without
Conveyor input	<ul style="list-style-type: none"> • With • Without
Override input	<ul style="list-style-type: none"> • With • Without
Min. override pulse time	<ul style="list-style-type: none"> • 100 ms • 350 ms

Direction detection

Direction detection is used when the transported material has to be moved in a specific direction. The direction depends on the sequence in which the muting sensors are activated.

If direction detection is disabled, the material to be transported can be moved in both directions in order to fulfil the muting conditions. In this case it is not relevant which sensor pair is activated first.

If **Forward (A1/A2 first)** was selected as the direction, the muting sensor pairs have to be activated in the sequence (A1/A2) before (B1/B2). Muting is not possible in the opposite

direction. Muting is terminated by a transition from four active sensors to an inactive sensor pair "B" (no sensor or one sensor active).

If **Backward (B1/B2 first)** was selected as the direction, the muting sensor pairs have to be activated in the sequence (B1/B2) before (A1/A2). Muting is not possible in the forward direction. Muting is terminated by a transition from four active sensors to an inactive sensor pair "A" (no sensor or one sensor active).

Condition of other sensor pair for muting start

The **Condition of other sensor pair for muting start** parameter determines when a valid muting sequence can begin. The **Condition of other sensor pair for muting start** can be defined for one of the following conditions:

- **Both inputs are clear:** All muting sensors have changed to Low together or individually and the OSSDs of the protective device (e.g. safety light curtain) are High (i.e. the protective field is clear),
or
- **At least one sensor is clear:** All muting sensors except the last muting sensor are Low and the OSSDs of the protective device (e.g. safety light curtain) are High (i.e. the protective field is clear).

If a higher throughput is required, it can be advantageous to allow the beginning of the next muting sequence as soon as the transported material has passed the protective device as well as all muting sensors with the exception of the last one (i.e. **At least one sensor is clear**).

Muting end condition

In contrast to the **Condition of other sensor pair for muting start**, the **Muting end condition** determines when a valid muting state ends. You can choose when the **Muting end condition** occurs:

- **With muting sensor pair:** When a muting sensor of the last muting sensor pair changes to Low (sensor clear),
or
- **With ESPE:** When the OSSDs of the protective device (e.g. safety light curtain) indicate that the protective field is no longer violated, i.e., the protective field is clear, and the OSSDs return to High.

If after the muting end the OSSD input of the ESPE becomes Low (e.g. by a violation of the protective field of the ESPE) before the next valid muting sequence has begun, the **Enable** output of the function block becomes Low. The next muting cycle cannot begin until the **Muting end condition** has been fulfilled.

Total muting time

The **Total muting time** is used in order to limit the maximum duration of the muting sequence. If the set value for the **Total muting time** is exceeded, the **Muting error** and **Fault present** outputs change to High and the **Enable** output changes to Low.

The timer for the **Total muting time** begins when the muting function is activated, indicated by the transition of the **Muting status** output to High. The timer for the **Total muting time** is stopped and reset to zero when the muting function changes to Low. If the optional **Conveyor** input is used, the timer for the **Total muting time** pauses when the **Conveyor** input is High, indicating that the conveyor belt has stopped.

Additional muting time when ESPE is clear

The **Additional muting time when ESPE is clear** parameter is used when the **Muting end condition** has been configured as **With ESPE**. If the ESPE does not always detect the muting end exactly because of irregularities in the material or the transport means, you can increase the availability of the machine by configuring an additional muting time of up to 1000 ms. Only in this case does the **Additional muting time when ESPE is clear** parameter determine the additional muting time after the OSSDs of the ESPE have returned to High, i.e. that the safety light curtain is no longer interrupted.

Concurrency monitoring time

The **Concurrency monitoring time** is used to check whether the muting sensors are activated simultaneously. This value specifies the maximum duration for which each of the two dual-channel evaluated muting sensor inputs may have different values without this being evaluated as an error. This means that input pair A1 and A2 or input pair B1 and B2 must have equivalent values before the **Concurrency monitoring time** has expired.

Concurrency monitoring begins with the first change of an input value of a muting sensor. If the **Concurrency monitoring time** has expired and the two inputs of an input pair still have different values, an error occurs.

If the concurrency monitoring determines an error for at least one input pair, the function block indicates this error by setting the **Muting error** output to High.

Suppression of sensor signal gaps

Occasionally faults occur in the output signals of muting sensors that may not be relevant for muting. The **Suppression of sensor signal gaps** function allows to filter out brief faults without muting being interrupted.

When **Suppression of sensor signal gaps** is enabled, a Low signal from a muting sensor input is ignored for the duration of the set value for **Suppression of sensor signal gaps**. The function block continues to interpret this signal as an uninterrupted High as long as only one sensor per pair A1/A2 or B1/B2 has a signal gap. If a signal gap has been detected at a sensor, the simultaneous occurrence of a further signal gap at another sensor results in the termination of muting.

Sequence monitoring

Sequence monitoring is used to define a special mandatory sequence in which the muting sensors have to be High. Tab. 82 shows the valid sequence for muting sensor input signals. This parameter is only available for configurations with four muting sensors, for example for parallel muting or sequential muting.

Tab. 82: Requirements for sequence monitoring

Direction detection	Requirement for the muting sensor signal inputs for sequence monitoring:
Disabled	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

This parameter depends on the function block. Deviations from the sequence shown above result in a muting error, indicated by the **Muting error** output. In order to avoid machine downtime the configured time for the **Suppression of sensor signal gaps** should furthermore be shorter than the time span that the transported object requires to pass a muting sensor pair (e.g. A1/A2 or B1/B2).

C1 input

The **C1** input is used as an additional measure to avoid manipulations. If **C1** is used, a transition from Low to High has to take place before the first muting sensor pair becomes High. Input **C1** must then remain High until both sensors of the muting sensor pair are High so that a valid muting condition can arise. If this condition is not fulfilled, this results in a muting error, indicated by the **Muting error** output. The **C1** input subsequently has to return to Low again before the next muting cycle is permitted.

Override input

An **Override** input signal is used to remove transported objects that have remained in the protective field of the protective device (e.g. safety light curtain) after power failures, triggering of an emergency stop, muting errors or similar circumstances.

The **Override required** output pulses at 2 Hz if the following conditions are fulfilled:

- Muting is currently inactive (i.e. **Muting status** is Low).

- At least one muting sensor is High.
- The OSSDs of the ESPE are Low (e.g. safety light curtain is interrupted).
- The **Enable** output is Low.

If the conditions for the **Override required** output are fulfilled and a valid override pulse sequence with a Low-High-Low transition (minimum 100 ms or 350 ms and maximum 3 s; longer or shorter pulses are ignored) has occurred at the **Override** input, the **Enable** output becomes High, as if the muting conditions were fulfilled. When all muting sensors have returned to Low and the OSSD input of the ESPE is High (e.g. 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 fulfil the conditions for a muting cycle, but the conditions for the **Override required** output, a further override cycle can be used in order to remove the transported material. The number of override cycles is limited (see Tab. 84 “Number of permissible override cycles” on page 142).

A reset button can also be suitable for the override function. Check the requirements of your application in order to ensure that the safety-relevant logic fulfils the requirements of the local, regional, national and international regulations.

NOTE

Tab. 83 provides information about the **Override required** output and when Override is possible under the shown conditions and when not.

Muting status	At least one muting sensor is High	OSSDs of the ESPE are High	Override required output	Override possible
0	No	0	No	No
0	No	1	No	No
0	Yes	0	Pulses at 2 Hz	Yes, if the maximum permissible number of override cycles is not exceeded
0	Yes	1	No	No
1	No	0	No	No
1	No	1	No	No
1	Yes	0	No	No
1	Yes	1	No	No

Tab. 83: Conditions for Override required and Override possible

Fig. 169 shows an example sequence for **Override** and **Override required**.

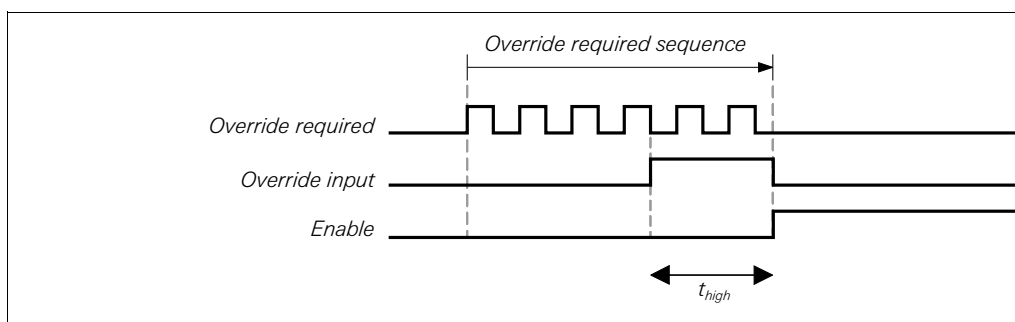


Fig. 169: Logic diagram for Override and Override required

t_{high} has to be equal to or greater than the minimum override pulse time (100 ms or 350 ms), but less than or equal to 3 s. If t_{high} is shorter than the minimum override pulse time or greater than 3 s, the **Override input** is ignored.

NOTE

If you use Override, check whether the system is in a safe state!

The Override function is used to activate the safety output (i.e. the **Enable** output) of the muting function block although the safety device (e.g. a safety light curtain) signals that a dangerous state may exist. The **Override** input should only be used when the hazardous area has been checked visually and nobody is in the hazardous area or has access to the hazardous area while the **Override** input is being used.

Ensure that the transitions of the signals for Override fulfil the requirements of the safety standards and regulations!



Logic programming – Function blocks

In case of a short-circuit to High (to 24 V DC) at a physical input, the evaluated signal can have a pulse when the signal is reset due to the short-circuit detection. If such a pulse can result in a dangerous state in the machine, the following points have to be observed:

- Ensure protected cable laying for the signal line (due to cross-circuiting to other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

During an override cycle, the **Enable** output is set to High as during a valid muting sequence. The number of permissible override cycles is limited in order to prevent excessive use of the override function. The number of permissible override cycles depends on the value for the total muting time. Tab. 84 summarises the number of permissible override cycles:

Tab. 84: Number of permissible override cycles

Total muting time	Number of permissible override cycles	Remarks
5 s	360	Maximum number of override cycles = 360 = 60 min/total muting time
10 s	360	
20 s	180	
30 s	120	
1 min	60	
5 min	12	
15 min	5	Minimum number of override cycles = 5
30 min	5	
60 min	5	
Disabled (unlimited)	5	

The number of override cycles is stored in the function block. The value is incremented each time the **Override required** output starts pulsing. The value is reset to "0", after a valid muting cycle has occurred, after a system reset (e.g. using the samos®PLAN) or after a transition from the Stop state to the Run state.

After the **Override required** output has started pulsing at 2 Hz and a subsequent **Override** signal has become High, Muting begins again and the **Enable** output becomes High.

If the muting cycle is stopped because of a faulty input signal of a muting sensor, **Override required** changes to High for the duration of the logic execution time, provided that the remaining conditions for Override required are fulfilled. If the faulty input of the muting sensor returns to High and subsequently to Low, the muting cycle is stopped again and **Override required** becomes High, provided that the remaining conditions for **Override required** are fulfilled.

During a valid override state, the direction detection, sequence monitoring (depending on the function block) and concurrency monitoring are not carried out for the duration of an override cycle.

Conveyor input

If the movement of the transported material is stopped during the muting cycle, the total muting time and other parameters that can result in a muting error may be exceeded. This can be avoided by using the **Conveyor** input. This input is used to stop time-related functions connected with muting when the material to be transported does not move further. The **Conveyor** input has to fulfil EN 61131 and has the following properties:

- 0 V DC = conveyor belt stopped, e.g. Low
- 24 V DC = conveyor belt running, e.g. High

The following timer functions are influenced by the value of the **Conveyor** input:

Tab. 85: Effects of the conveyor belt monitoring on the timer functions

Monitoring of the total muting time	<ul style="list-style-type: none"> • If a belt stop is detected, these timer functions pause. • If the conveyor belt starts up again, the timer continues its function with the value stored before the belt stop has been detected. If this occurs the first time, the total muting time is increased once by 5 seconds.
Concurrency monitoring	

NOTE

The **Suppression of sensor signal gaps** is not influenced by a belt stop.

Min. override pulse time

The **Min. override pulse time** determines the minimum High duration for a valid pulse sequence at the **Override** input.

Muting status output

The **Muting status** output indicates the state of the muting function in accordance with the following table:

Condition	Muting status output
Muting cycle inactive, no error	Low
Muting cycle active, no error	High
Muting error detected	Low
Override active, no error	High

Tab. 86: Output values for muting status

Muting lamp output

The **Muting lamp** output is used in order to indicate an active muting cycle. The value for the **Muting lamp** output depends directly on the value of the **Muting status** output as shown in the following table:

Status of the Muting function block	Value of the Muting lamp output
Muting status output is Low	Low
Muting status output is High	High
Override cycle active	High
Override required	Pulses at 2 Hz

Tab. 87: Output values for the Muting lamp output

Muting error output

The **Muting error** output is used to indicate that an error connected to the muting function block has been detected. The **Muting error** output becomes High when any muting error has been detected. In order to reset a muting error it is necessary that all muting sensors return to Low and that the OSSD signal of the ESPE is High.

Enable output

The **Enable** output is High if a valid muting condition exists, a valid override cycle occurs or if the OSSD input of the ESPE is clear and no error/error state is active.

Logic programming – Function blocks

8.9.3 Information on wiring

If muting functions are to be implemented, possible errors in the wiring have to be taken into consideration. If certain signal combinations are to be transferred in a common wire, additional precautions have to be taken in order to ensure that the respective signals are correct. Suitable measures have to be taken (e.g. protected wiring) in order to ensure that errors cannot arise through this wiring.

Tab. 88: Wiring combinations for muting and prerequisites

Signal description	A1	A2	B1	B2	C1	Conveyor	ESPE	Override input	Enable output	Muting lamp	Muting status	Override required
A1	-	A	B	B	A	A	A	A	A	A	A	C
A2	A	-	B	B	A	A	A	A	A	A	A	C
B1	B	B	-	A	A	A	A	A	A	A	A	C
B2	B	B	A	-	A	A	A	A	A	A	A	C
C1	A	A	A	A	-	A	A	A	A	C	C	C
Conveyor	A	A	A	A	A	-	C	A	A	C	C	C
ESPE	A	A	A	A	A	C	-	C	A	C	C	C
Override input	A	A	A	A	A	A	C	-	A	A	C	A

- A** The specified signals may not be installed in a common wire unless protected wiring is used.
- B** The specified signals may not be installed in a common wire unless protected wiring or sequence monitoring is used.
- C** The specified signals may be installed in a common wire.
- Not applicable

8.9.4 State transition from Stop to Run

If the samos[®]PRO safety controller changes from the Stop state to the Run state, the following behavioural patterns can be realised, depending on the state of the muting sensors and of the OSSDs of the sensors (e.g. safety outputs of a safety light curtain). Tab. 89 shows details of the system behaviour during the transition from Stop to Run.

Tab. 89: Stop-to-Run transition behaviour for muting functions

State after the change from the Stop state to the Run state		System behaviour	
Input ESPE	State of the muting sensors	Run	Next action
High (e.g., no object in the protective field)	All muting sensors are Low	A normal muting sequence is possible.	Muting is possible after correct activation/sequence of the muting sensors.
	The muting condition is partially fulfilled.		All muting sensors have to return to Low, before the OSSDs of the sensor become Low. If the OSSDs of the sensors become Low before all muting sensors have become Low, Override has to be used.
	The muting condition is fulfilled.		
Low (e.g., object detected)	All muting sensors are Low	Muting is blocked.	The sensor OSSDs have to become High before muting can take place.
	The muting condition is partially fulfilled.	Override is required, if configured.	Either transition to normal behaviour (in case of a cyclically correct sequence of sensor states) or the total override time is exceeded.
	The muting condition is fulfilled.		

8.9.5 Error states and information on resetting

Diagnostic outputs	Resetting the error state	Remarks
Muting error: <ul style="list-style-type: none"> • Error in the concurrency monitoring function • Error in the total muting time monitoring • Error in the direction detection • Sequence error detected • Error in the sensor gap monitoring 	A complete valid muting cycle has to occur before any muting error can be reset. To this purpose either Override has to be used or all muting sensors and OSSDs of the ESPE have to be clear and a subsequent valid muting sequence has to be applied. If one of these two conditions is fulfilled, the Muting error output returns to Low, provided that no other error exists.	The Enable output changes to Low and Fault present changes to High, if the Muting error output is High.

Tab. 90: Error states and information on resetting for Muting function blocks

8.9.6 Parallel muting

Function block diagram

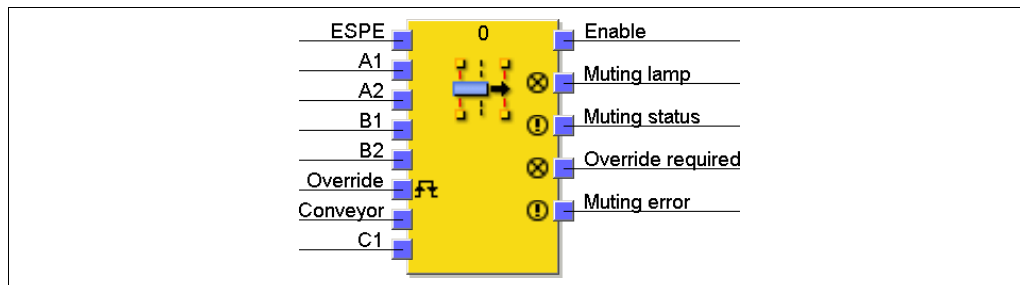


Fig. 170: Logic connections for the Parallel muting function block

Representation of the application

Fig. 171 shows an example of the placement of sensors for Parallel muting.

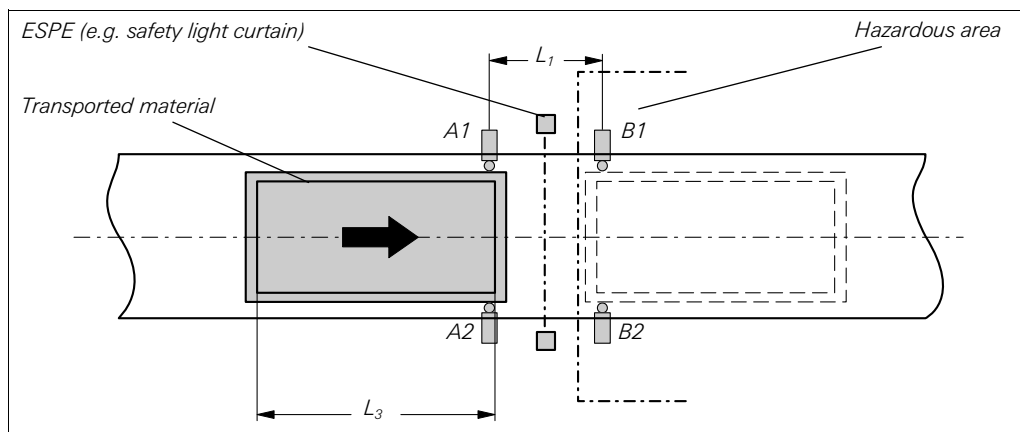


Fig. 171: Muting with two parallel sensor pairs

In this example the material moves from the left to the right. As soon as the first muting sensor pair A1 & A2 is activated, the protective effect of the protective device (ESPE) is muted. The protective effect remains muted until the muting sensor pair B1 & B2 is clear again.

Input conditions for muting sensors

Condition	Description
A1 & A2 (or B1 & B2)	Starts the muting cycle. The first sensor pair is activated depending on the direction of transportation of the material.
A1 & A2 & B1 & B2	Condition for transferring the muting function to the second sensor pair.
B1 & B2 (or A1 & A2)	Muting applies as long as this condition is fulfilled. The second sensor pair is activated depending on the direction of transportation of the material.

Tab. 91: Conditions for Parallel muting

Logic programming – Function blocks

Equations and prerequisites for calculating the distance:

$$L_1 \geq v \times 2 \times T_{IN \text{ Muting sensor}}$$

$$v \times t > L_1 + L_3$$

$$L_1 < L_3$$

$$T_{IN \text{ Light curtain}} < T_{IN \text{ Muting sensor}}$$

Where ...

L_1 = Distance between the sensors (layout symmetrical to the detection area of the ESPE)

L_3 = Length of material in conveyor direction

v = Velocity of the material (e.g. of the conveyor belt)

t = Set total muting time [s]

$T_{IN \text{ Light curtain}}, T_{IN \text{ Muting sensor}}$ = Response time of the light curtain or the muting sensors *in the samos®PRO system* (see chapter "Response times of the samos®PRO system" in the samos®PRO Hardware operating instructions).

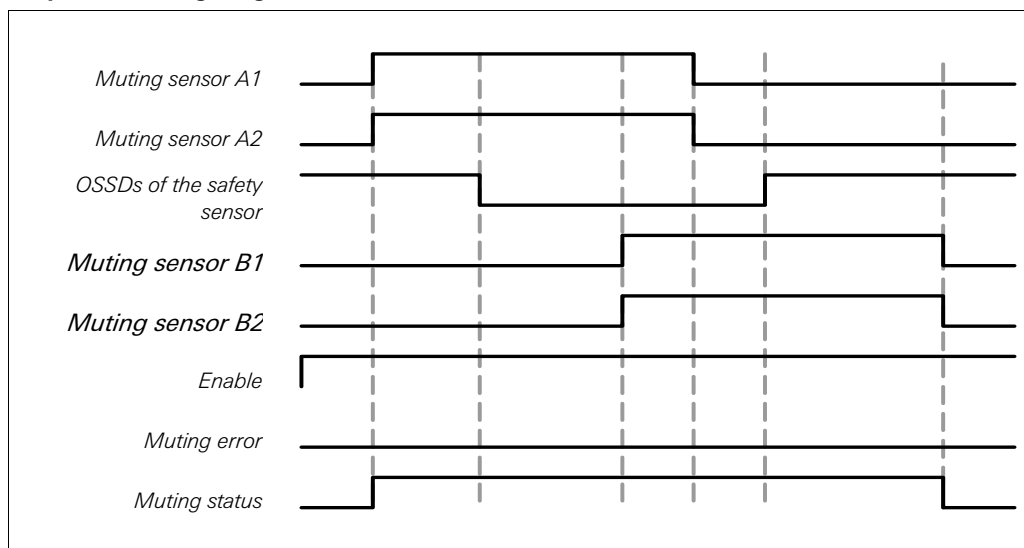
NOTES

- The material can be moved in both directions or a fixed direction of transportation can be defined for it as follows:
 - With the optional input **C1**. If used, the **C1** input always has to be activated before both muting sensors of the first sensor pair (e.g. A1 and A2) become High.
 - By means of the **Direction detection** configuration parameter
- In parallel layout, the position of the muting sensors also determines the width of the permissible object. The objects always have to pass the muting sensors with an identical width.
- Optical probes and all types of non-optical sensors can be used for this application. Use sensors and probes with background suppression.
- Avoid mutual interference of the sensors.
- Increase the protection against manipulation and the safety level by using the following configurable functions:
 - Concurrency monitoring
 - Monitoring of the total muting time
 - Muting end via ESPE
- Information on wiring can be found in section 8.9.3.

The function block requires that a valid muting sequence takes place. Fig. 172 shows an example of a valid muting sequence based on the basic parameter setting for this function block.

Sequence/timing diagram

Fig. 172: Valid muting sequence using the configuration basic setting



8.9.7 Sequential muting

Function block diagram

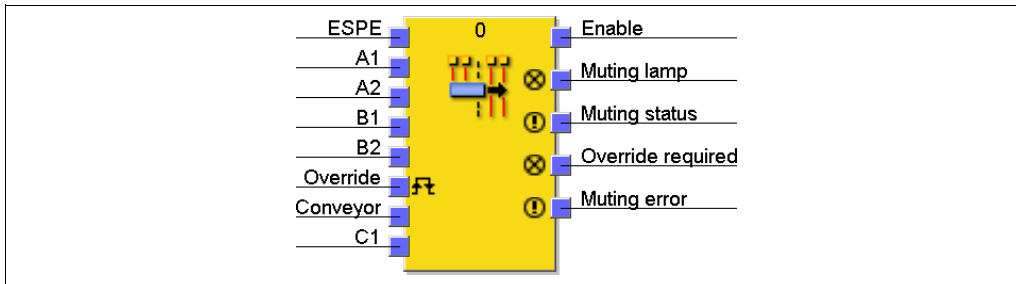


Fig. 173: Logic connections for the Sequential muting function block

Representation of the application

Fig. 174 shows an example of the placement of sensors using the Sequential muting function block.

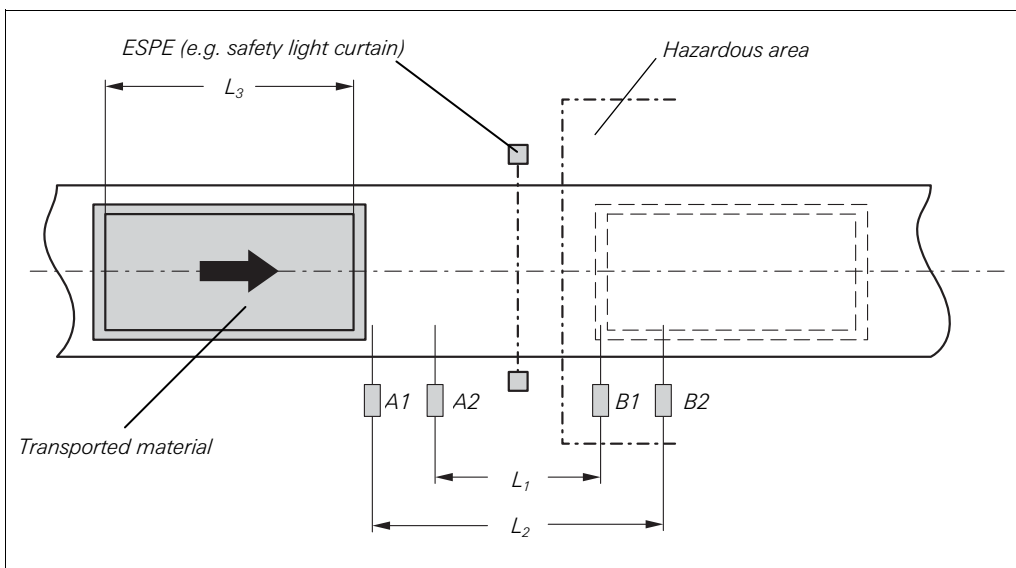


Fig. 174: Example of the sequential layout of muting sensors

In the example, the material moves from the left to the right. As soon as the muting sensors A1 & A2 are activated, the protective effect of the protective device (ESPE) is muted. The protective effect remains muted until a sensor of the muting sensor pair B1 & B2 becomes clear again.

Input conditions for muting sensors

Condition	Description
A1 & A2 (or B1 & B2)	Starts the muting cycle. The first sensor pair is activated depending on the direction of transportation of the material.
A1 & A2 & B2 & B1	Condition for transferring the muting function to the second sensor pair.
B1 & B2 (or A1 & A2)	Muting applies as long as this condition is fulfilled. The second sensor pair is activated depending on the direction of transportation of the material.

Tab. 92: Conditions for Sequential muting

Equations and prerequisites for calculating the distance:

$$L_1 \geq v \times 2 \times T_{IN \text{ Muting sensor}}$$

$$v \times t > L_1 + L_3$$

$$L_2 < L_3$$

$$T_{IN \text{ Light curtain}} < T_{IN \text{ Muting sensor}}$$

Where ...

L_1 = Distance between the inner sensors (layout symmetrical to the detection area of the ESPE)

Logic programming – Function blocks

- L_2 = Distance between the outer sensors (layout symmetrical to the detection area of the ESPE)
- L_3 = Length of the material in conveyor direction
- v = Velocity of the material (e.g. of the conveyor belt)
- t = Set total muting time [s]

$T_{IN \text{ Light curtain}}, T_{IN \text{ Muting sensor}}$ = Response time of the light curtain or the muting sensors *in the samos® PRO system* (see chapter "Response times of the samos® PRO system" in the samos® PRO Hardware operating instructions).

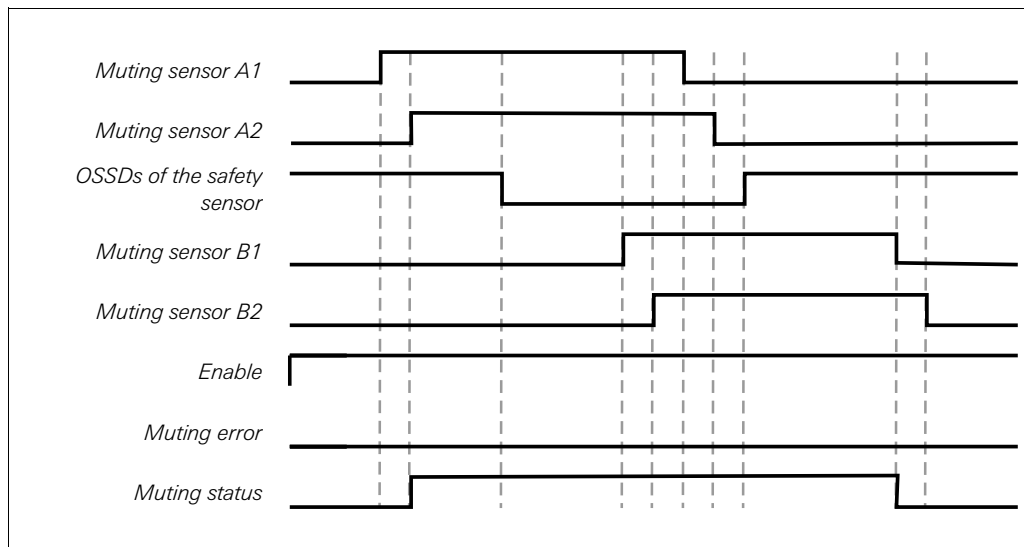
NOTES

- In this example the material can either be moved in both directions or a fixed direction of transportation can be defined as follows:
 - With the optional input **C1**. If used, the **C1** input always has to be activated before both muting sensors of the first sensor pair (e.g. A1 and A2) become High.
 - By means of the **Direction detection** configuration parameter
- The sensor layout shown in this example is suitable for all types of sensors.
- Avoid mutual interference of the sensors.
- Increase the protection against manipulation and the safety level by using the following configurable functions:
 - concurrency monitoring
 - monitoring of the total muting time
 - muting end via ESPE
 - sequence monitoring
- Information on wiring can be found in section 8.9.3.

Sequence/timing diagram

The function block requires that a valid muting sequence takes place. Fig. 175 shows an example of a valid muting sequence based on the basic parameter setting for this function block.

Fig. 175: Valid muting sequence using the basic configuration setting



8.9.8 Cross muting – direction of movement only forwards or backwards

Function block diagram

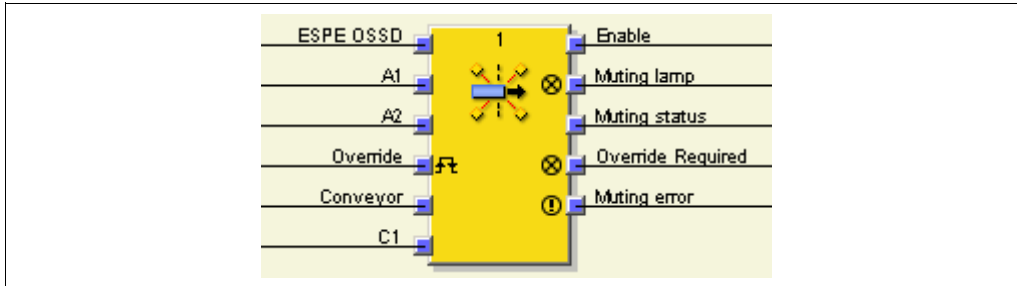


Fig. 176: Logic connections for the Cross muting function block with optional input C1

Representation of the application

Fig. 177 shows an example of the sensor layout for the Cross muting function block. The optional input **C1** is used as an additional protection for the muting system against manipulation.

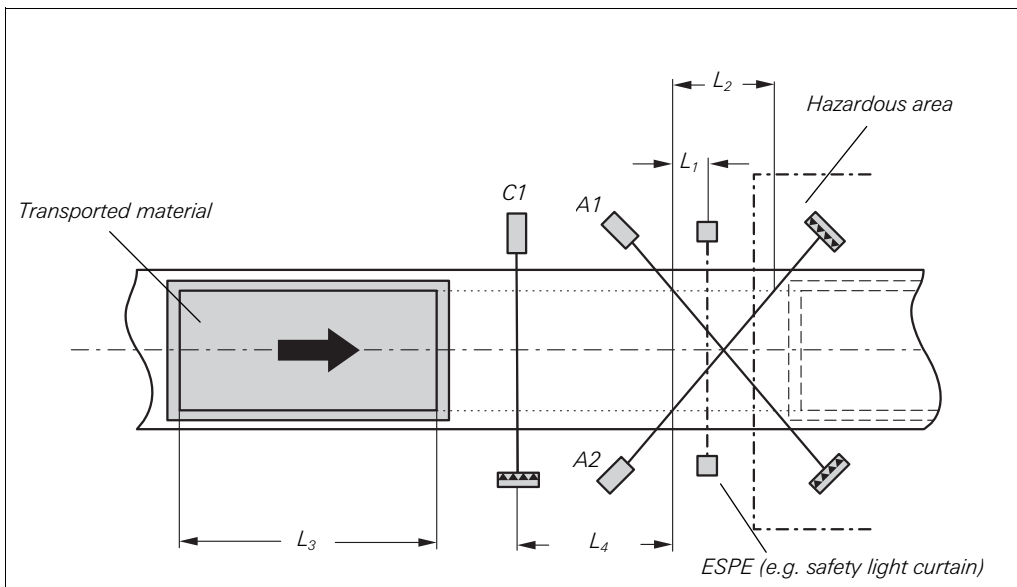


Fig. 177: Example of Cross muting with optional input C1

The protective effect of the protective device is muted when the muting sensors are operated in a defined sequence. The optional input C1 always has to be activated before **both** muting sensors of the first sensor pair (e.g. A1 and A2) become High.

Input conditions for muting sensors

Condition	Description
C1 & A1 & A2	Input C1 always has to be activated before both muting sensors of the first sensor pair (e.g. A1 and A2) become High.
A1 & A2	Muting applies as long as this condition is fulfilled and the requirement mentioned above existed.

Tab. 93: Conditions for Cross muting with optional input C1

Equations and prerequisites for calculating the distance:

$$L_1 \geq v \times T_{\text{IN Muting sensor}}$$

$$v \times t > L_2 + L_3$$

$$L_3 > L_4$$

$$T_{\text{IN Light curtain}} < T_{\text{IN Muting sensor}}$$

Where ...

L_1 = Minimum distance between the detection line of the ESPE and the detection by A1, A2

Logic programming – Function blocks

- L_2 = Distance between the two detection lines of the sensors (sensors activated/sensors clear)
- L_3 = Length of material in conveyor direction
- L_4 = Maximum distance between C1 and the detection line of A1, A2
- v = Velocity of the material (e.g. of the conveyor belt)
- t = Set total muting time [s]

$T_{IN \text{ Light curtain}}$, $T_{IN \text{ Muting sensor}}$ = Response time of the light curtain or the muting sensors *in the samos® PRO system* (see chapter “Response times of the samos® PRO system” in the samos® PRO Hardware operating instructions).

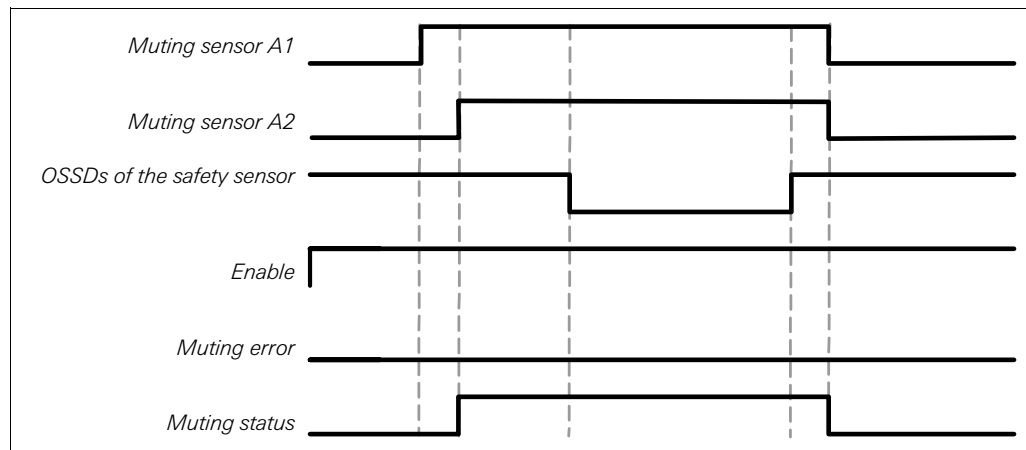
NOTES

- Material flow is only possible in one direction in this example.
- In order to move material in both directions (i.e. bidirectionally), place the intersection directly in the light beams of the ESPE (see section 8.9.9 “Cross muting – material transport in both directions” on page 150).
- The sensor layout shown in this example is suitable for both through-beam photoelectric switches and photoelectric reflex switches.
- Avoid mutual interference of the sensors.
- Increase the protection against manipulation and the safety level by using the following configurable functions:
 - concurrency monitoring
 - monitoring of the total muting time
 - muting end via ESPE
- Information on wiring can be found in section 8.9.3.

Sequence/timing diagram

The function block requires that a valid muting sequence takes place. Fig. 178 shows an example of a valid muting sequence based on the basic parameter setting for this function block. The optional input **C1** is not contained in the sequence shown below.

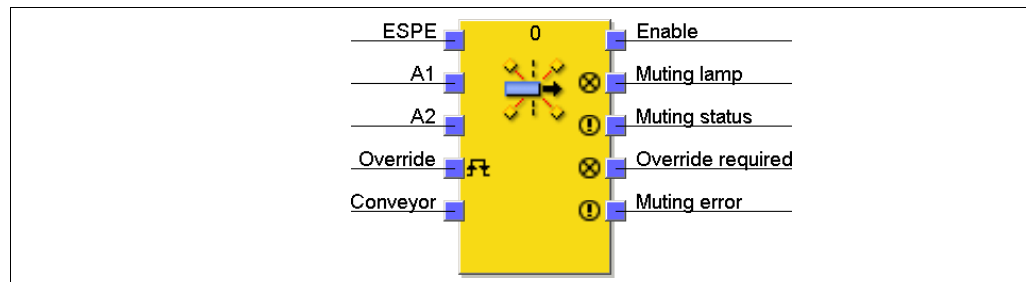
Fig. 178: Valid muting sequence using the configuration basic setting



8.9.9 Cross muting – material transport in both directions

Function block diagram

Fig. 179: Logic connections for the Cross muting function block



Representation of the application

The sensors can be located as follows in the case of muting applications with one crossed sensor pair where the material has to be moved in both directions. The optional signal **C1** is not used in this application example.

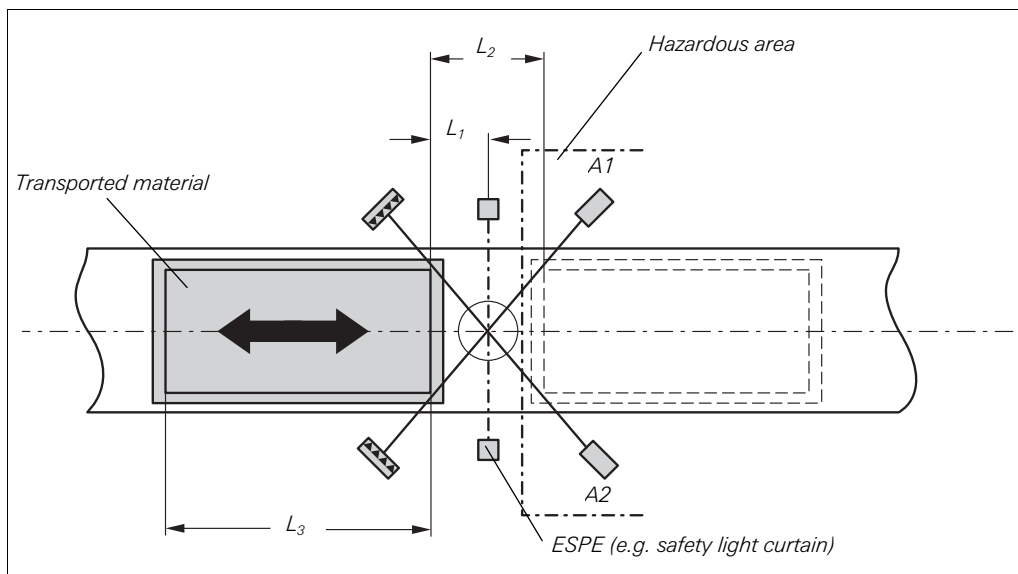
Ensure that the muting sensors detect only the moved material!

Ensure that the muting sensors are positioned in such a manner that no one can enter the hazardous area by fulfilling the muting conditions (meaning that they activate both muting sensors and thus create the required conditions for muting).



WARNING

Fig. 180: Cross muting for bidirectional movement of material



Input conditions for muting sensors

Condition	Description
A1 & A2	Muting applies as long as this condition is fulfilled and, furthermore, the requirements mentioned above existed.

Tab. 94: Conditions for Cross muting without optional input C1

Equations and prerequisites for calculating the distance:

$$L_1 \geq v \times T_{\text{IN Muting sensor}}$$

$$v \times t > L_2 + L_3$$

$$T_{\text{IN Light curtain}} < T_{\text{IN Muting sensor}}$$

Where ...

- L_1 = Minimum distance between the detection line of the ESPE and the detection by A1, A2
- L_2 = Distance between the two detection lines of the sensors (sensors activated/sensors clear)
- L_3 = Length of material in conveyor direction
- v = Velocity of the material (e.g. of the conveyor belt)
- t = Set total muting time [s]

$T_{\text{IN Light curtain}}, T_{\text{IN Muting sensor}}$ = Response time of the light curtain or the muting sensors *in the samos® PRO system* (see chapter "Response times of the samos® PRO system" in the samos® PRO Hardware operating instructions).

- Material flow is possible in both directions in this example.
- In order for material to be moved in both directions, place the intersection of the muting sensors exactly in the course of the light beams of the ESPE.
- In order to move material in one direction only, place the intersection, with regard to the direction of the transport, behind the light beams of the ESPE (see section 8.9.8 "Cross muting – direction of movement only forwards or backwards" on page 149).
- The sensor layout shown in this example is suitable for both through-beam photoelectric switches and photoelectric reflex switches.

NOTES

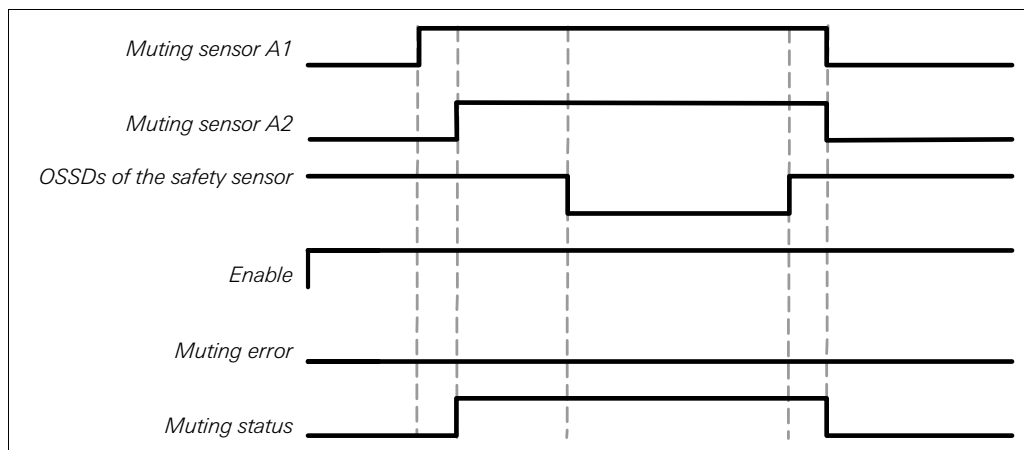
Logic programming – Function blocks

- Avoid mutual interference of the sensors.
- Increase the protection against manipulation and the safety level by using the following configurable functions:
 - concurrency monitoring
 - monitoring of the total muting time
 - muting end via ESPE
- Information on wiring can be found in section 8.9.3.

Sequence/timing diagram

The function block requires that a valid muting sequence takes place. Fig. 181 shows an example of a valid muting sequence based on the basic parameter setting for this function block.

Fig. 181: Valid muting sequence using the basic configuration setting



8.10 Function blocks for press contact monitoring

8.10.1 Overview and general description

For press applications there are two complementary types of function blocks offered. This chapter describes the contact monitoring function blocks, which provide signals for the press cycle control function blocks.

There are two different function blocks for press contact monitoring which can be used to monitor the correct cam signal sequence and the correct stopping (overrun) of the press. The outputs of these function blocks indicate the current press cycle phase in which the press is operating (e.g. upstroke or top). Typically the **Enable** output, **Top** output and **Upstroke** output of a contact monitoring function block are connected to the corresponding inputs of one or more press cycle control function blocks.

Tab. 95: Overview of the press contact monitoring function blocks

	Eccentric press	Universal press
Typical press types	Eccentric press	Eccentric press Hydraulic press
Press travelling directions	Forward	Forward and backward
Cams	Overrun cam Upstroke cam Dynamic cam	Top dead center (TDC) Bottom dead center (BDC) Stop control contact (SCC)
Top position condition	When Overrun cam = High	When TDC = Low
Upstroke condition	When Upstroke cam = High	When BDC = High
Overrun monitoring	Optional	Optional
Disable monitoring	Optional	Optional

8.10.2 Eccentric press contact

Function block diagram

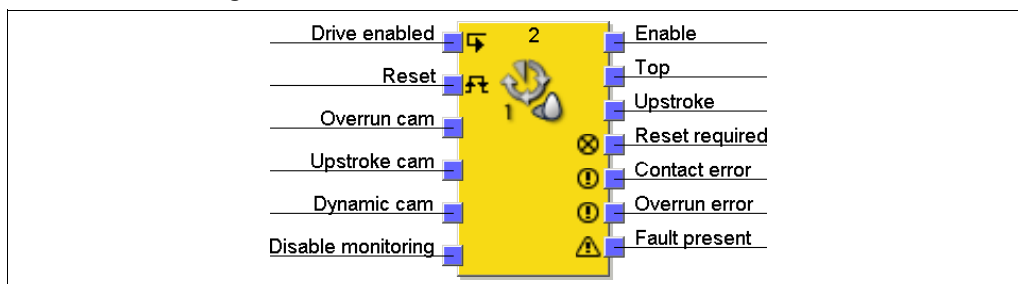


Fig. 182: Logic connections for the Eccentric press contact function block

General description

The Eccentric press contact function block can be used for specific types of eccentric presses (i.e. mechanical presses). The minimum configuration requires an **Overrun cam** and the **Upstroke cam**. Optionally a **Dynamic cam** can also be connected.

Parameters of the function block

Parameter	Possible values
Dynamic cam input	<ul style="list-style-type: none"> • With • Without
Minimum reset pulse time	<ul style="list-style-type: none"> • 100 ms • 350 ms
Reset input	<ul style="list-style-type: none"> • With • Without
Disable monitoring input	<ul style="list-style-type: none"> • With • Without
Use Fault present	<ul style="list-style-type: none"> • With • Without

Tab. 96: Parameters of the Eccentric press contact function block

Enable output

The **Enable** output is used to stop the press and is connected to another complementary press function block, e.g. Press setup or Press single stroke. If no error was detected, the **Enable** output is High.

If any error in the contact signal sequence is detected, the **Enable** output changes to Low, the corresponding error output changes to High and the **Reset required** output changes to High. A valid reset sequence at the **Reset** input is then required.

The **Enable** output also changes to Low if Monitoring gets disabled.

Reset input

A valid reset sequence at the **Reset** input is a Low-High-Low transition with a minimum pulse duration of 100 ms or 350 ms and maximum 30 s. Shorter and longer pulses are ignored.

Ensure that the transitions of the signals for resetting fulfil the requirements of the safety standards and regulations!

In case of a short-circuit to High (to 24 V DC) at a physical input, the evaluated signal can have a pulse when the signal is reset due to the short-circuit detection. If such a pulse can result in a dangerous state in the machine, the following points have to be observed:

- Ensure protected cable laying for the signal line (due to cross-circuiting to other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

Safety-relevant signals have to conform to standards and regulations to be applied!

Always take the valid national, regional and local regulations and standards into consideration for your application. Type C standards such as EN 692 and EN 693 contain requirements how safety-relevant signals have to be used. For example, the restart signal may



have to be protected by suitable means in case of overrun errors (e.g. by a key switch or in a closed control cabinet).

If the **Reset** input is disabled, an error can be reset only by stopping the logic execution, e.g. by a power cycle or by changing the system state from the Run state to the Stop state and back to the Run state with the samos®PLAN.

Top output and Upstroke output

The **Top** output is typically used to stop the press and is connected to another complementary press function block, e.g. Press setup or Press single stroke.

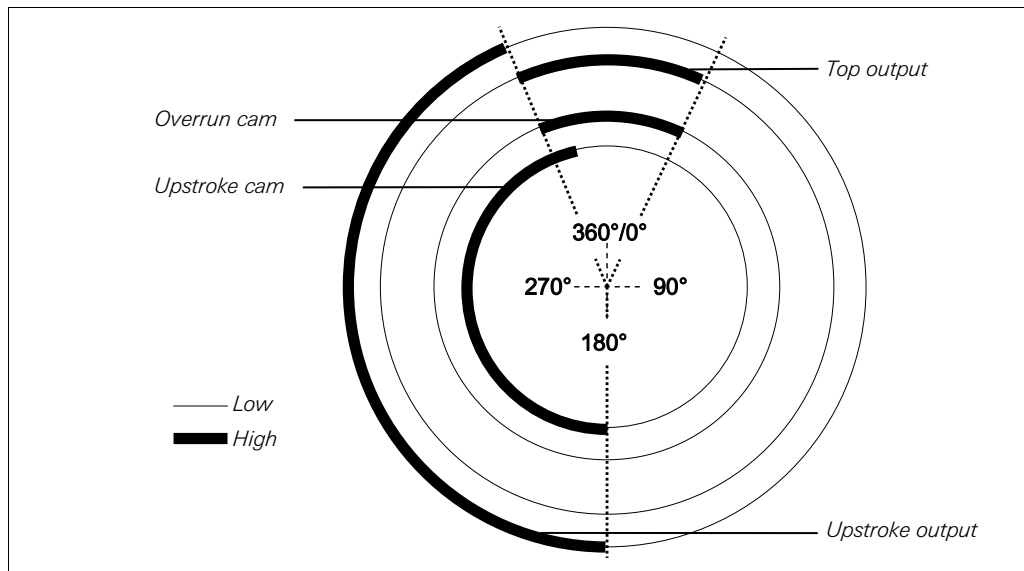
The **Upstroke** output is typically connected to another complementary press function block, e.g. Press single stroke or Press Setup. Additionally it can be used to initiate upstroke muting.

This function block sets the **Upstroke** output and the **Top** output based on value changes of the cam inputs. If the function block detects an error, both outputs are set to Low.

Without Dynamic cam

The **Upstroke** output is set to High with a rising edge (transition from Low to High) of the **Upstroke cam** input and is set to Low with a rising edge of the **Overrun cam** input. The **Top** output is set to High if the **Overrun cam** input is High.

Fig. 183: Press cycle for the Eccentric press contact function block without Dynamic cam



With Dynamic cam

If this function block is configured with **Dynamic cam**, the start of the Top phase can be brought forward with a falling edge (transition from High to Low) of the **Dynamic cam** input.

The **Upstroke** output is set to High with a rising edge (transition from Low to High) of the **Upstroke cam** input. It is set to Low either with a rising edge of the **Overrun cam** input or with a falling edge of the **Dynamic cam** input, whichever occurs first.

The **Top** output is set to High with a rising edge of the **Overrun cam** or with a falling edge of the **Dynamic cam** input, whichever occurs first. The **Top** output is set to Low with a falling edge of the **Overrun cam** input.

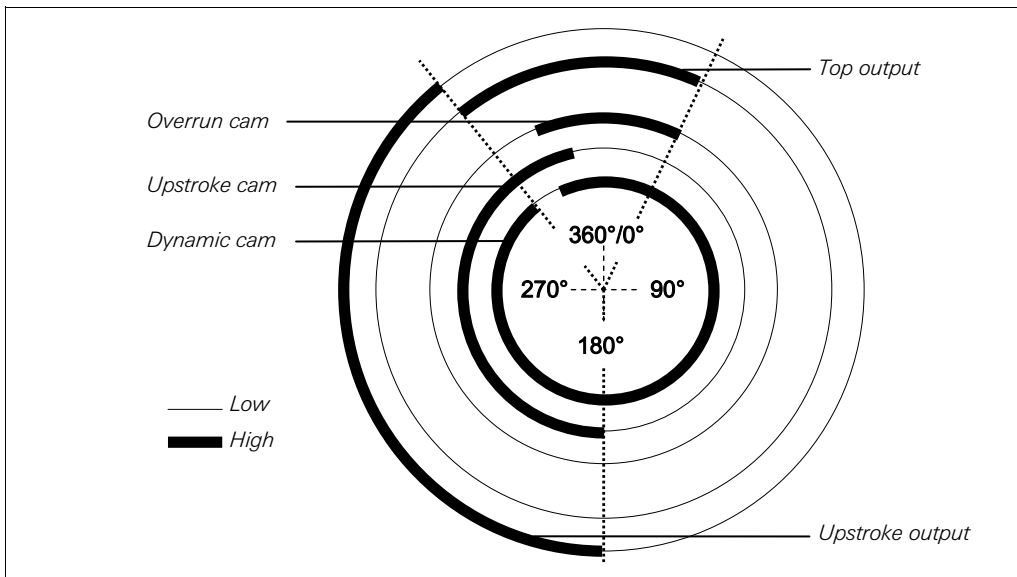


Fig. 184: Press cycle for the Eccentric press contact function block with Dynamic cam on upstroke

If there is a falling edge of the **Dynamic cam** input when the **Upstroke cam** input is Low, i.e. in the Run-down phase of the press cycle, the **Top** output will be set to High until a rising edge of the **Upstroke cam** input is detected. The **Upstroke** output will remain Low for the rest of the press cycle.

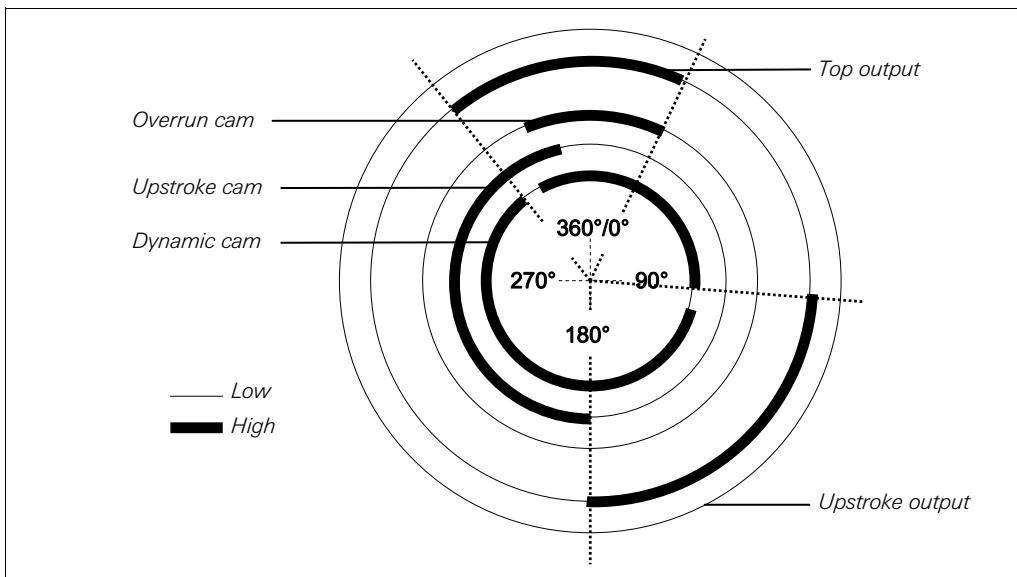


Fig. 185: Press cycle for the Eccentric press contact function block with Dynamic cam on upstroke and downstroke

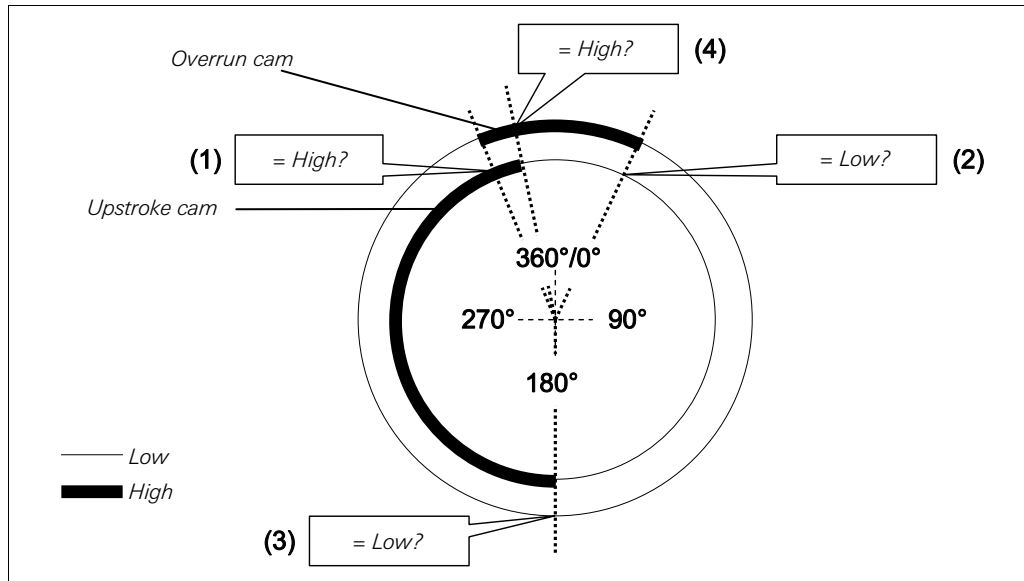
If the **Upstroke cam** input is already High when the monitoring of the cam inputs begins (e.g. in the first logic cycle, after resetting an error or after enabling monitoring with **Disable monitoring** input), the **Upstroke** output will remain Low until the first actual transition from Low to High has been detected at the **Upstroke cam** input.

NOTE

Contact monitoring

The input signals for the **Overrun cam** input, **Upstroke cam** input and the **Drive enabled** input have to accord with Fig. 186 and the following rules.

Fig. 186: Contact monitoring with the Eccentric press contact function block



- (1) The overrun must begin during the upstroke phase: The rising edge at the **Overrun cam** input (Low-High transition) must occur while the **Upstroke cam** input is High.
- (2) The overrun must end after the end of the upstroke phase: The falling edge at the **Overrun cam** input (High-Low transition) must occur when the **Upstroke cam** input is Low.
- (3) The upstroke phase must begin after the overrun has ended: The rising edge at the **Upstroke cam** input (Low-High transition) must occur while the **Overrun cam** input is Low.
- (4) The upstroke phase must end during the overrun: The falling edge at the **Upstroke cam** input (High-Low transition) must occur while the **Overrun cam** input is High.

If at least one of these conditions is not fulfilled during operation, the **Enable** output becomes Low and the **Contact error** output becomes High.

A valid sequence to fulfil the conditions is as follows:

- 0) Start condition: **Overrun cam** input = High, **Upstroke cam** input = Low
- 1) **Overrun cam** input: High → Low
- 2) **Upstroke cam** input: Low → High
- 3) **Overrun cam** input: Low → High
- 4) **Upstroke cam** input: High → Low



Observe the relevant safety standards and regulations!

All safety related parts of the installation (cabling, connected sensors and actuators, configuration settings) must be according to the relevant safety standards (e.g. EN 62-061 or EN ISO 13-849-1 or Type C standards such as EN 692 and EN 693) and regulations. Only safety-relevant signals may be used in safety-relevant logic. Ensure that the application fulfils all applicable standards and regulations!

This needs to be considered especially for the **Upstroke cam**, if the **Upstroke** output is used for upstroke muting, e.g. in combination with a press cycle control function block.

In order to fulfil the safety standards it may be necessary to use tested switches for the cam input signals, each with different test sources. To use different test sources for the cam signals, the **Overrun cam**, **Upstroke cam** and **Dynamic cam** need to be connected to different SP-SDI or SP-SDIO modules.

NOTE

One SP-SDI has two test sources only, even if it has eight test output terminals.

Overrun monitoring

The Eccentric press contact function block monitors the overrunning of the press. If the **Overrun cam** is left although the press is supposed to have stopped, then the function block detects an Overrun error.

The **Drive enabled** input signal must then accord to Fig. 187 and the following rule.

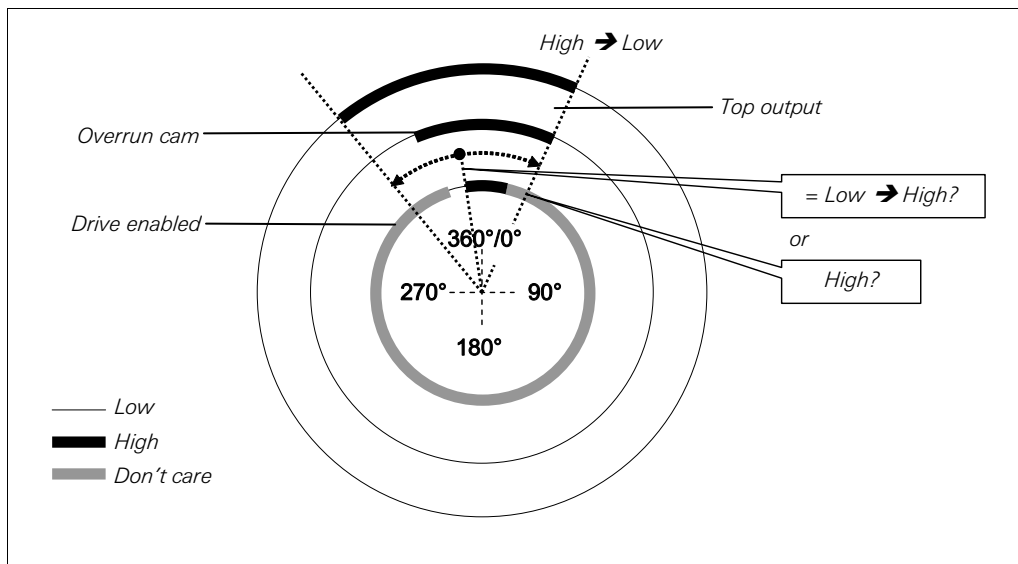


Fig. 187: Overrun monitoring with the Eccentric press contact function block

At the **Drive enabled** input there has to be either a Low to High transition while the **Top** output is High or the **Drive enabled** input has to be High at the end of the **Overrun cam** (High-Low transition). If none of these two conditions is fulfilled, the **Enable** output becomes Low and the **Overrun error** output becomes High.

The **Drive enabled** input must be connected to the signal that controls the physical output of the press drive, so that the function block can determine whether the press is currently supposed to be running or has been stopped. Typically this is the **Enable** output of a subsequent Press setup or Press single stroke function block.

Do not connect any physical input signals to the **Drive enabled** input. Connect the signal that controls the physical output for the press drive using a jump address or a CPU marker.

NOTE

- If a jump address is used, make sure that this signal is a loop-back. This is indicated by a clock icon on the destination tag of the jump address. To this purpose, connect the outputs of this function block to the following function blocks before you connect the jump address to the **Drive enabled** input. This applies especially if all connections to the following function blocks are also realized using jump addresses.
- If a CPU marker is used, then a Routing function block must be used to split the signal to the physical output for the press drive and to the CPU marker output.

Disable monitoring

Using this optional input it is possible to deactivate the monitoring functionality under certain conditions in order to prevent the function block to go into an error state. This can be useful for certain operating modes, e.g. during the setup of the machine or when the press moves backwards.

If the **Disable monitoring** input is High, the **Enable** output of the Eccentric press contact function block is Low and the monitoring of the cam signal sequence and overrun is inhibited, provided there is no error pending. The error output states are not affected by this.

If the **Disable monitoring** input is High and an error is pending, a reset of the error is possible.

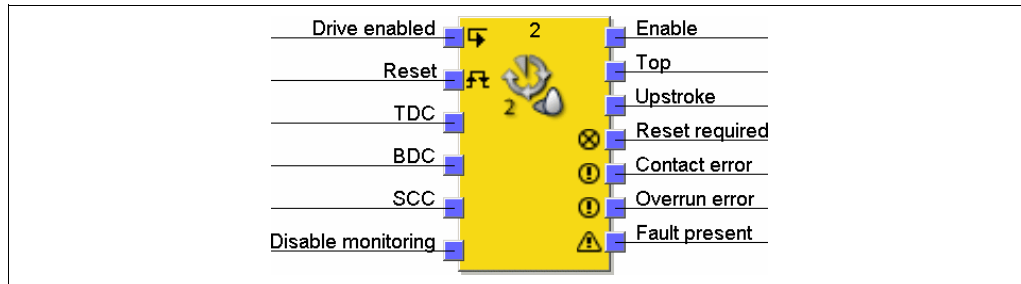
When the **Disable monitoring** input changes from High to Low, the function block behaves in the same way as after a change from the Stop state to the Run state, i.e. the **Enable** output will become High again.

Logic programming – Function blocks

8.10.3 Universal press contact

Function block diagram

Fig. 188: Logic connections for the Universal press contact function block



General description

The Universal press contact function block can be used for different press types (e.g. hydraulic presses and eccentric presses (i.e. mechanical presses)). The minimum configuration requires only **TDC** (Top dead center). Optionally, the **BDC** (Bottom dead center) and **SCC** (Stop control contact) inputs can be connected.

- The **Upstroke** output is available only if the **BDC** input is enabled.
- Overrun monitoring is possible only if the **SCC** input is enabled.
- If **BDC** and **SCC** are not used, a plausibility check is not possible for the function block. In this case, a check for overrunning cannot be carried out. The only remaining function in this case is the provision of the **Top** output signal.



Do not use this function block for safety purposes without BDC and SCC!

If this function block is used without the **BDC** and **SCC** inputs, then it must be used for automation control only, i.e. not for safety functionality. Otherwise the operator of the press will be in danger.

Tab. 97: Parameters of the Universal press contact function block

Parameters of the function block

Parameter	Possible values
SCC input	<ul style="list-style-type: none"> • With • Without
BDC input	<ul style="list-style-type: none"> • With • Without
Number of BDC signals per cycle	<ul style="list-style-type: none"> • 1 (e.g. eccentric press) • 0-2 (e.g. hydraulic press)
Min. reset pulse time	<ul style="list-style-type: none"> • 100 ms • 350 ms
Reset input	<ul style="list-style-type: none"> • With • Without
Disable monitoring input	<ul style="list-style-type: none"> • With • Without
Use Fault present	<ul style="list-style-type: none"> • With • Without

Enable output

The **Enable** output is used to stop the press and is connected to another complementary press function block, e.g. Press setup or Press single stroke. If no error was detected, the **Enable** output is High.

If any error in the contact signal sequence is detected, the **Enable** output changes to Low, the corresponding error output changes to High and the **Reset required** output changes to High. A valid reset sequence at the **Reset** input is then required.

The **Enable** output also changes to Low if Monitoring gets disabled.

Reset input

A valid reset sequence at the **Reset** input is a Low-High-Low transition with a minimum pulse duration of 100 ms or 350 ms and maximum 30 s. Shorter and longer pulses are ignored.

Ensure that the transitions of the signals for resetting fulfil the requirements of the safety standards and regulations!

In case of a short-circuit to High (to 24 V DC) at a physical input, the evaluated signal can have a pulse when the signal is reset due to the short-circuit detection. If such a pulse can result in a dangerous state in the machine, the following points have to be observed:

- Ensure protected cable laying for the signal line (due to cross-circuiting to other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

Safety-relevant signals have to conform to standards and regulations to be applied!

Always take the valid national, regional and local regulations and standards into consideration for your application. Type C standards such as EN 692 and EN 693 contain requirements how safety-relevant signals have to be used. For example, the restart signal may have to be protected by suitable means in case of overrun errors (e.g. by a key switch or in a closed control cabinet).

If the **Reset** input is disabled, an error can be reset only by stopping the logic execution, e.g. by a power cycle or by changing the system state from the Run state to the Stop state and back to the Run state with the samos®PLAN.

Top output and Upstroke output

The **Top** output is typically used to stop the press and is connected to another complementary press function block, e.g. Press setup or Press single stroke.

The **Upstroke** output is typically connected to another complementary press function block, e.g. Press single stroke or Press Setup. Additionally it can be used to initiate upstroke muting.

This function block sets the **Upstroke** output and the **Top** output based on value changes of the cam inputs. If the function block detects an error, both outputs are set to Low.

The **Top** output is set to High if the **TDC** input is Low. The **Upstroke** output is set to High with a rising edge (transition from Low to High) of the **BDC** input. It is set to Low either with a falling edge of the **TDC** input, or with a falling edge of the **BDC** input, whichever occurs first.

If the function blocks starts (power up, disable → enable) with the **BDC** input High, the **Upstroke** output will remain Low for the first press cycle.

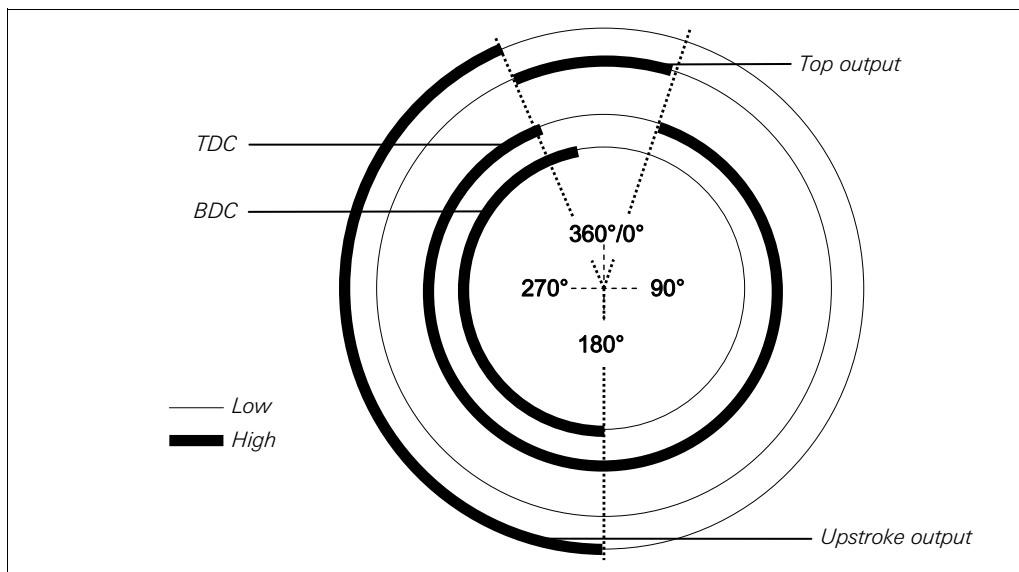
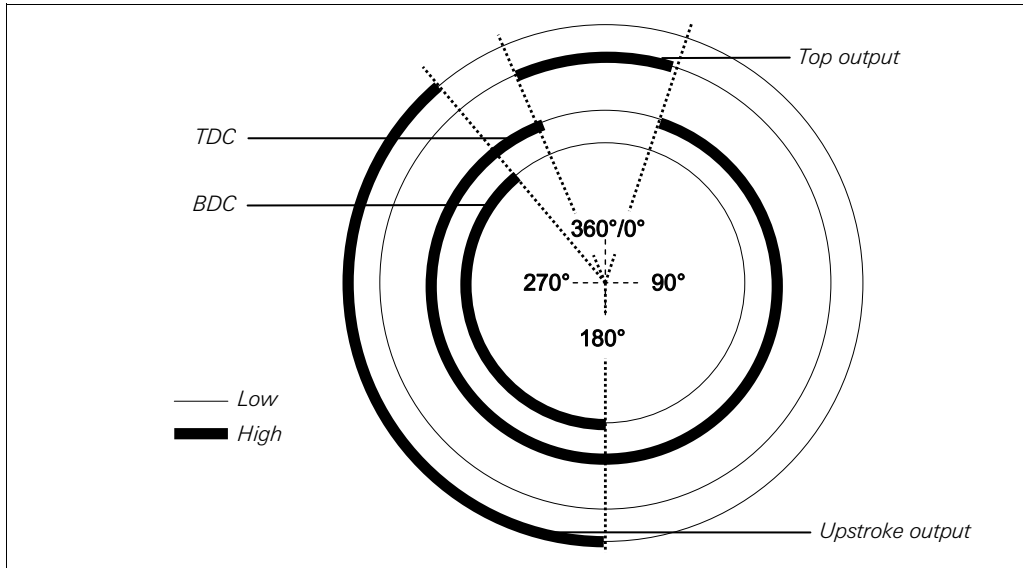


Fig. 189: Press cycle for the Universal press contact function block with falling edge of TDC before BDC



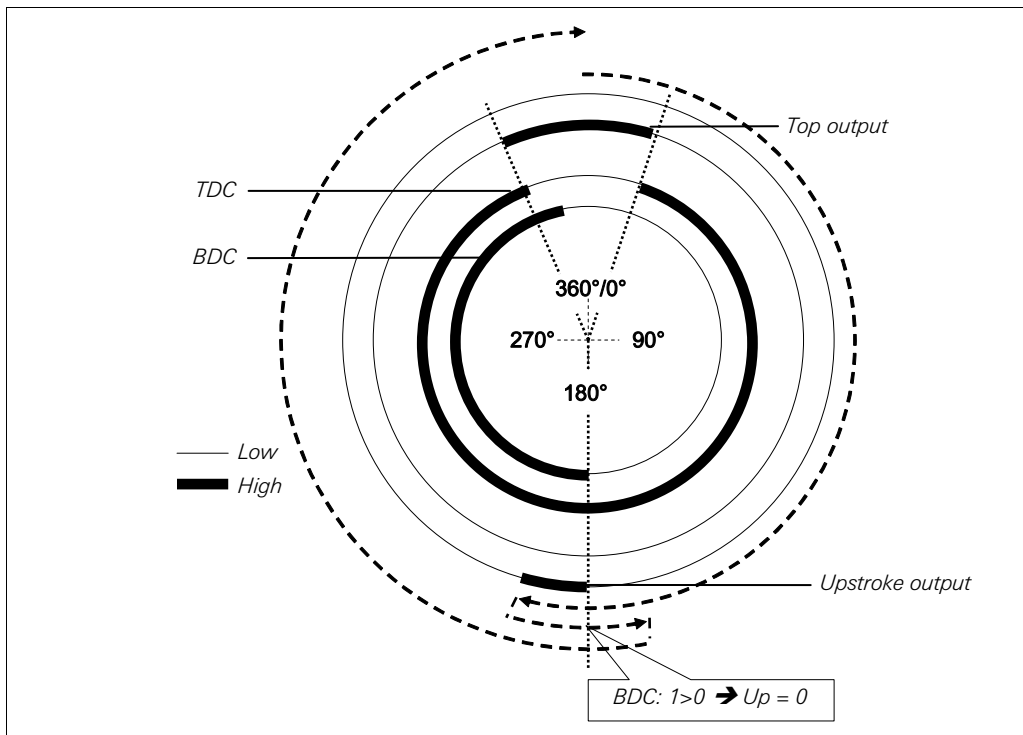
Logic programming – Function blocks

Fig. 190: Press cycle for the Universal press contact function block with falling edge of BDC before TDC



A second rising edge of the **BDC** input does not restart the upstroke phase. This is the case if **Number of BDC signals per cycle** is 0-2 (e.g. hydraulic press), and the press moves back and forth in the bottom section.

Fig. 191: Press cycle for the Universal press contact function block with 2 BDC transitions



If in this setting no **BDC** pulse at all occurs during the cycle, the **Upstroke** output will remain Low for the complete cycle.

NOTE

If the **BDC** input is already High when the monitoring of the contact inputs begins (e.g. in the first logic cycle, after resetting an error or after enabling monitoring with **Disable monitoring input**), the **Upstroke** output will remain Low during the first cycle. The next **BDC** transition from Low to High is only accepted after a transition from High to Low has occurred at the **Top** output.

TDC monitoring

There must be exactly one **TDC** pulse per cycle. A violation of this rule can be detected only if the **SCC** input is enabled and/or the **BDC** input is enabled and the parameter **Number of BDC signals per cycle** is set to 1 (e.g. eccentric press).

SCC monitoring

If SCC is enabled, the **SCC** input signals must accord with Fig. 192 and the following rules:

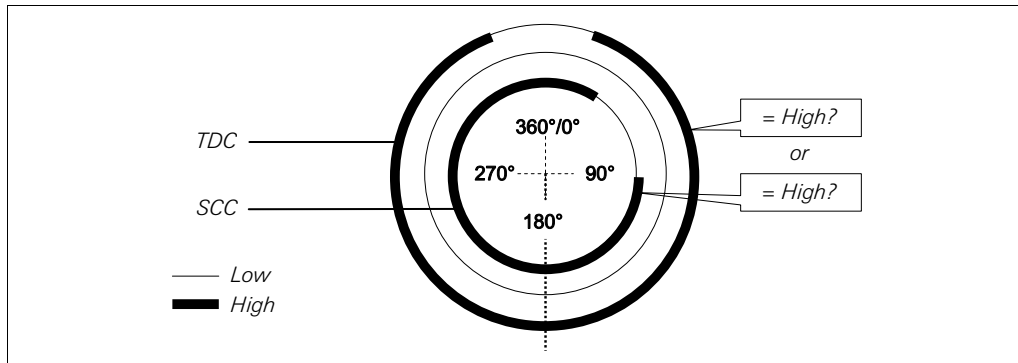


Fig. 192: Contact monitoring with the Universal press contact function block with SCC enabled

There must be exactly one SCC pulse per cycle. The rising edge of the **SCC** input (Low-High transition) must occur before the falling edge of the **TDC** input. The falling edge of the **SCC** input (High-Low transition) must occur after the rising edge of the **TDC** input. This means that at any time at least one of both inputs must be High.

BDC monitoring

If BDC is enabled and SCC is disabled, the **BDC** input signals must accord with Fig. 193 and the following rules.

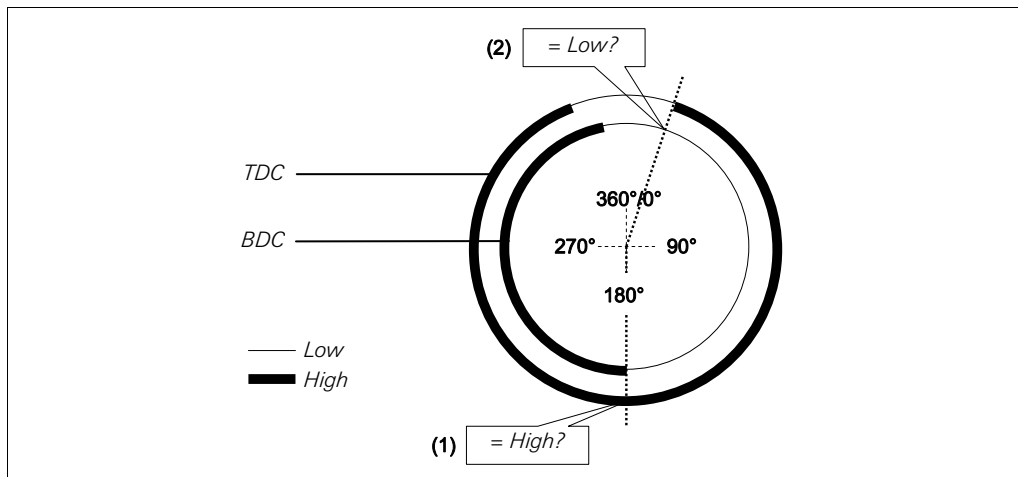


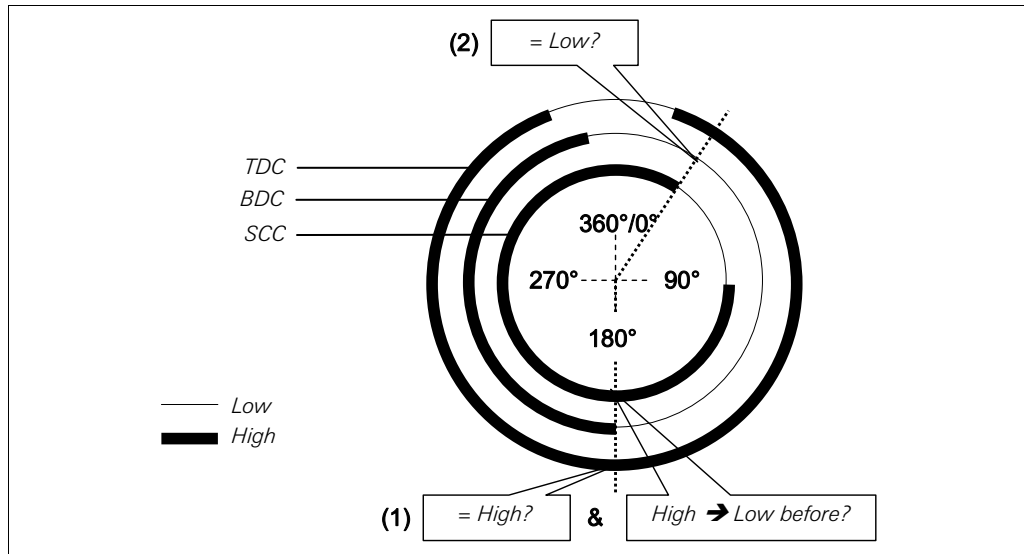
Fig. 193: Contact monitoring with the Universal press contact function block with BDC enabled

- (1) The beginning of the **BDC** (Low-High transition) must be close to 180° and has to occur while the **TDC** input is High.
- (2) The end of the **BDC** (High-Low transition) has to occur before the rising edge (Low-High transition) of the **TDC** input. In other words, **BDC** must be Low when a Low-High transition at the **TDC** input occurs.

BDC and SCC monitoring

If BDC and SCC are enabled, the **BDC** input signals must accord with Fig. 194 and the following rules.

Fig. 194: Contact monitoring with the Universal press contact function block with BDC enabled



(1) The beginning of the **BDC** (Low-High transition) must be close to 180° and has to occur while the **TDC** input is High and after the falling edge (High-Low transition) of the **SCC** input (the **SCC** input may have changed back to High already).

(2) The end of the **BDC** (High-Low transition) has to occur before the falling edge (High-Low transition) of the **SCC** input. In other words, **BDC** must be Low when a High-Low transition at the **SCC** input occurs.

A valid sequence that fulfils the conditions for BDC and SCC is:

1. Start condition: TDC = Low, BDC = Low, SCC = High
2. TDC: Low → High
3. Drive enabled = High (this is to fulfil the condition for overrun monitoring)
4. SCC: High → Low
5. BDC: Low → High
6. SCC: Low → High
7. TDC: High → Low and BDC: High → Low (sequence does not matter)

Depending on the type of press (e.g. hydraulic press), it may occur that the beginning of the BDC (step 5 above) occurs not once but twice or even not at all. To avoid that this leads to a contact error, the parameter **Number of BDC signals per cycle** must be set to the value 0-2 (e.g. hydraulic press). With this setting, the conditions for the BDC still apply for every BDC pulse with the exception of the falling edge of the **SCC** input (step 4 above).

Additionally, the number of BDC signals (Low-High-Low) must be according to the configured value, i.e. either exactly one or any value between 0 and 2.

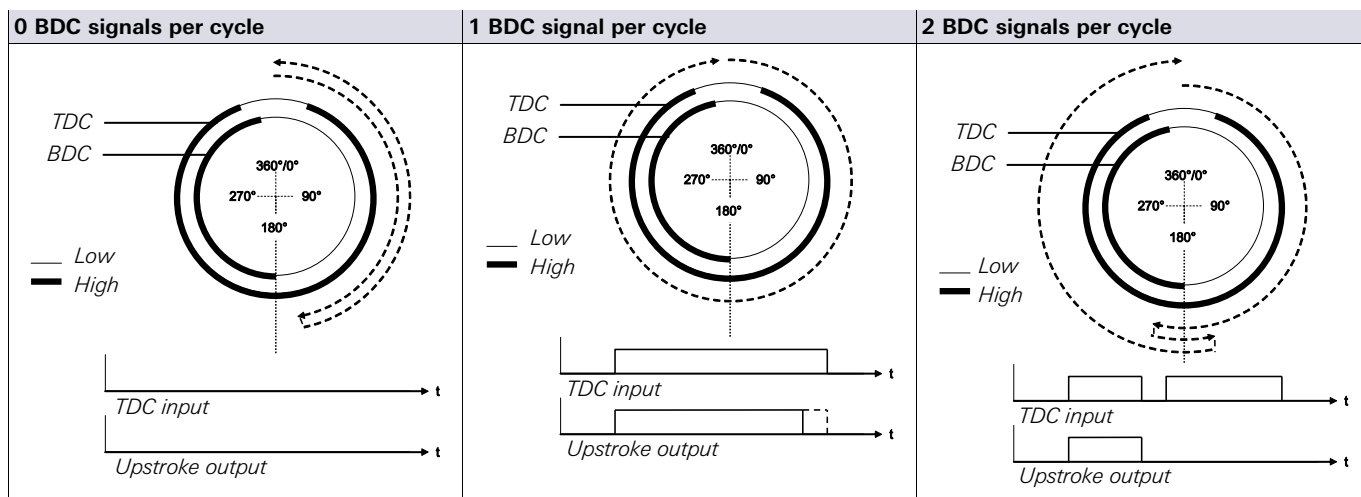


Fig. 195: Timing diagrams for 0, 1 and 2 BDC signals per cycle

If one or more of the conditions mentioned above is not fulfilled during operation, the **Enable** output becomes Low and the **Contact error** output becomes High.

Observe the relevant safety standards and regulations!

All safety related parts of the installation (cabling, connected sensors and actuators, configuration settings) must be according to the relevant safety standards (e.g. EN 62-061 or EN ISO 13-849-1 or Type C standards such as EN 692 and EN 693) and regulations. Only safety-relevant signals may be used in safety-relevant logic. Ensure that the application fulfils all applicable standards and regulations!

This needs to be considered especially for the **BDC**, if the **Upstroke** output is used for upstroke muting, e.g. in combination with a press cycle control function block.

If the **Number of BDC signals per cycle** is configured to 0-2 (e.g. hydraulic press), the function block's error detection is reduced and not all input errors can be detected (e.g. stuck-at-low at the **BDC** input).

In order to fulfil the safety standards it may be necessary to use tested switches for the cam input signals, each with different test sources. To use different test sources for the cam signals, the **TDC**, **BDC** and **SCC** inputs need to be connected to different SP-SDI or SP-SDIO modules.

One SP-SDI has 2 test sources only, even if it has eight test output terminals.

Overrun monitoring

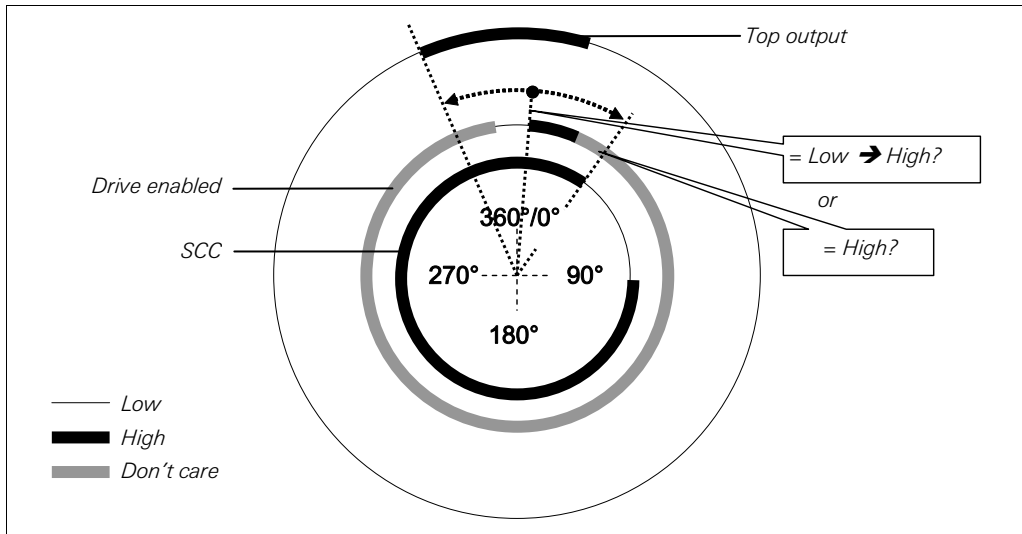
If the **SCC** input is enabled, the Universal press contact function block monitors the overrunning of the press, i.e. it checks whether the **SCC** cam is left although the press is expected to have stopped.



NOTE

Fig. 196: Overrun monitoring with the Universal press contact function block

The **Drive enabled** input signal must then accord with Fig. 196 and the following rule.



At the **Drive enabled** input there has to be either a Low to High transition between the Low-High transition of the **Top** output and the end of **SCC** (High-Low transition), or the **Drive enabled** input has to be High at the end of the **SCC** (High-Low transition). If none of these two conditions is fulfilled, the **Enable** output becomes Low and the **Overrun error** output becomes High.

The **Drive enabled** input must be connected to the signal that controls the physical output of the press drive, so that the function block can determine whether the press is currently supposed to be running or has been stopped. Typically this is the **Enable** output of a subsequent Press setup or Press single stroke function block.

NOTE

Do not connect any physical input signals to the **Drive enabled** input. Connect the signal that controls the physical output for the press drive using a jump address or a CPU marker.

- If a jump address is used, make sure that this signal is a loop-back. This is indicated by a clock icon on the destination tag of the jump address. To this purpose, connect the outputs of this function block to the following function blocks before you connect the jump address to the **Drive enabled** input. This applies especially if all connections to the following function blocks are also realized using jump addresses.
- If a CPU marker is used, then a Routing function block must be used to split the signal to the physical output for the press drive and to the CPU marker output.

Disable monitoring

Using this optional input it is possible to deactivate the monitoring functionality under certain conditions in order to prevent the function block to go into an error state. This can be useful for certain operating modes, e.g. during the setup of the machine or when the press moves backwards.

If the **Disable monitoring** input is High, the **Enable** output of the Universal press contact function block is Low and the monitoring of the cam signal sequence and overrun is inhibited, provided there is no error pending. The error output states are not affected by this.

If the **Disable monitoring** input is High and an error is pending, a reset of the error is possible.

When the **Disable monitoring** input changes from High to Low, the function block behaves in the same way as after a change from the Stop state to the Run state, i.e. the **Enable** output will become High again.

8.11 Function blocks for press cycle control

8.11.1 Press setup

Function block diagram

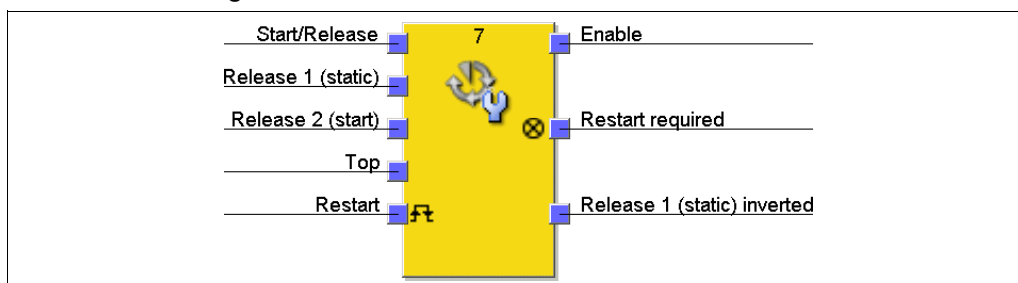


Fig. 197: Logic connections for the Press setup function block

General description

The Press setup function block is generally used together with the Universal press contact or the Eccentric press contact function block in order to set up the press and in order to provide the information of the Top output as input for this function block. The Top output is required for single-stroke operation. Control of the press can, for example, be effected by means of a two-hand control.

Parameters of the function block

Parameter	Possible values
Restart interlock condition	<ul style="list-style-type: none"> • Never • When Release 1 or Start/Release is Low • When Release 1 is Low or Top changes to High • Always
Release 2 (start) input	<ul style="list-style-type: none"> • With • Without
Single stroke protection	<ul style="list-style-type: none"> • Active • Inactive
Min. restart pulse time	<ul style="list-style-type: none"> • 100 ms • 350 ms

Tab. 98: Parameters of the Press Setup function block

Ensure that the transitions of the signals for restarting fulfil the requirements of the safety standards and regulations!

In case of a short-circuit to High (to 24 V DC) at a physical input, the evaluated signal can have a pulse when the signal is reset due to the short-circuit detection. If such a pulse can result in a dangerous state in the machine, the following points have to be observed:

- Ensure protected cable laying for the signal line (due to cross-circuiting to other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.



Input signals of the function block

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 edge (Low to High) at the **Start/Release** input signals a start of the press. A Low **Start/Release** input signals a stop of the press. If **Restart interlock condition** is set to **When Release 1 or Start/Release is Low**, a valid restart sequence is required after a stop that was caused by a Low **Start/Release** input signal.

Release 1 (static)

The input signal **Release 1 (static)** is mandatory. The **Enable** output always changes immediately to Low, if **Release 1 (static)** is Low.

If this function block is used together with a press contact function block (e.g. Eccentric press contact or Universal press contact), the **Enable** output of this press contact function block must be connected with the **Release 1 (static)** input of the Press setup function block.

Release 2 (start)

The input signal **Release 2 (start)** is optional. If **Release 2 (start)** is configured, the **Enable** output can only change to High (e.g. during switching on), if **Release 2 (start)** is High. If the **Enable** output is High, **Release 2 (start)** is no longer monitored.



WARNING

Do not use the Release 2 (start) input for safety purposes!

Do not use the **Release 2 (start)** input for initiating safety stops, because this input is evaluated temporarily during the start sequence only. Otherwise the operator of the press will be in danger.

Top

The **Top** input signal is used in order to determine the end of the press cycle (i.e. the press has reached the top dead center). This signal is available at the function blocks Eccentric press contact or Universal press contact. The **Top** input signal is used for single-stroke protection. When the **Single stroke protection** configuration parameter is set to **Active**, the **Enable** output changes to Low when the **Top** input changes from Low to High.



WARNING

Do not use the Top input for safety purposes!

Connect the **Top** input only with the **Top** output of an Eccentric press contact or Universal press contact function block or to an equivalent signal source. Do not use the **Top** input for initiating safety stops. Otherwise the operator of the press will be in danger.

Restart input

If the **Restart interlock condition** parameter has been set to **Never**, a Restart signal is not required in order to restart the press after any kind of stop. The **Restart interlock condition** parameter can also be set to the following values:

- When Release 1 or Start/Release is Low
- When Release 1 is Low or Top changes to High
- Always

This parameter determines when a **Restart** signal is expected as input signal for the function block.

If the **Enable** output changes to Low because of the above-mentioned settings of the configuration parameters for **Restart interlock condition**, the **Enable** output can only be reset after a valid restart sequence with a Low-High-Low transition (minimum 100 ms or 350 ms; shorter pulses and pulses longer than 30 s will be ignored) has been carried out.

Output signals of the function block

Restart required

The **Restart required** output is High, when a valid restart sequence is expected at the **Restart** input.

Enable

The **Enable** output is High, when **Restart required** is Low (i.e. a restart is not required) and the following conditions are fulfilled:

- When **Single stroke protection** is set to **Inactive**, **Release 1 (static)** is High and **Release 2 (start)** (if configured) is High; and a rising edge (Low to High) is detected at the **Start/Release** input; or
- If **Single stroke protection** is set to **Active**, **Start/Release** changes from Low to High, **Release 1 (static)** is High and **Release 2 (start)** (if configured) is High. In this case the **Enable** output changes to Low when the **Top** input changes from Low to High.

Release 1 (static) inverted

The **Release 1 (static) inverted** output signals whether an enable signal for the Press setup function block is present. If **Release 1 (static)** is High, **Release 1 (static) inverted** is Low and vice versa.

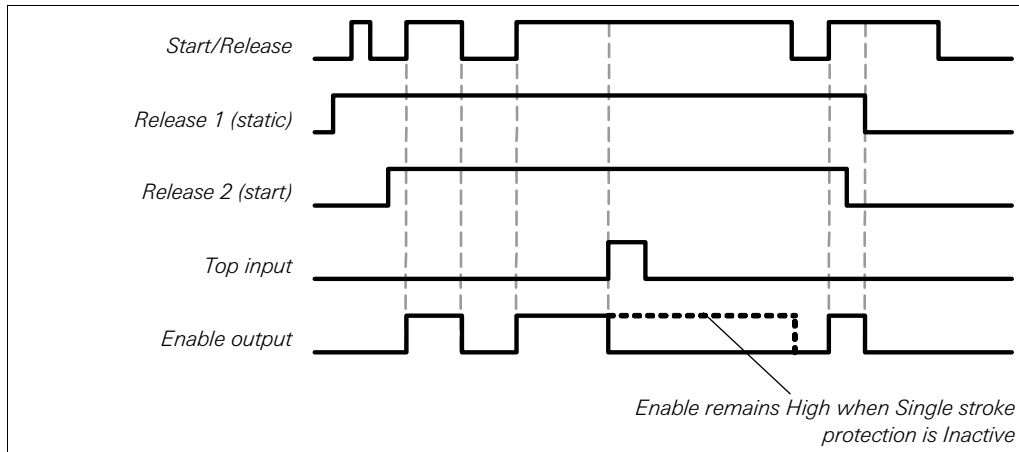


Fig. 198: Sequence/timing diagram for the Press setup function block

8.11.2 Press single stroke

Function block diagram

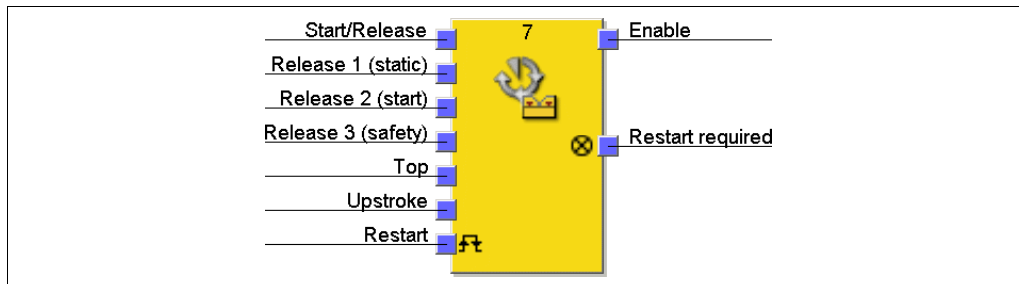


Fig. 199: Logic connections for the Press single stroke function block

General description

The Press single stroke function block is generally used together with the Universal press contact or the Eccentric press contact function block in order to provide the information of the **Top** and **Upstroke** outputs as input for this function block. The **Top** output is required for single-stroke operation. Controlling of the press can, for example, be implemented by means of a Two hand control or an N-break function block in connection with a safety light curtain.

Single-stroke protection is always active and cannot be configured. This means: When the signal of the **Top** input changes to High, the **Enable** output is always set to Low. The requirements for a restart depend of the configuration of the parameter for **Restart interlock condition**.

Parameters of the function block

Parameter	Possible values
Restart interlock condition	<ul style="list-style-type: none"> • Never • When Release 1 or Release 3 or Start/Release is Low • When Release 1 or Release 3 is Low or Top changes to High • Always • When Release 1 or Release 3 is Low
Release 2 (start) input	<ul style="list-style-type: none"> • With • Without
Release 3 (safety) input	<ul style="list-style-type: none"> • With • Without

Tab. 99: Parameters of the Press single stroke function block

Parameter	Possible values
Mode for Start/Release input	<ul style="list-style-type: none"> • Start and static release (stepping) • Start only (stopping not possible)
Mode for upstroke muting	<ul style="list-style-type: none"> • Inactive • For Release 3 • For Release 3 and Start/Release
Max. time for upstroke muting	0 = disabled, 1 to 7200 s. The Upstroke input is available only if the value is not set to 0.
Min. restart pulse time	<ul style="list-style-type: none"> • 100 ms • 350 ms
Ignore Release 3 (safety) for Restart interlock while in top position	<ul style="list-style-type: none"> • Yes • No



Ensure that the transitions of the signals for restarting fulfil the requirements of the safety standards and regulations!

In case of a short-circuit to High (to 24 V DC) at a physical input, the evaluated signal can have a pulse when the signal is reset due to the short-circuit detection. If such a pulse can result in a dangerous state in the machine, the following points have to be observed:

- Ensure protected cable laying for the signal line (due to cross-circuiting to other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

Input parameters and input signals of the function block

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 edge (Low to High) at the **Start/Release** input signals a start of the press. A Low **Start/Release** input signals a stop of the press. If the **Mode for Start/Release input** parameter is set to **Start only (stopping not possible)**, the press cannot be stopped by the **Start/Release** input signal.



Use additional safety measures when Start/Release is set to Start only (stopping not possible)!

If the **Mode for Start/Release input** parameter is set to **Start only (stopping not possible)**, you must use additional safety measures (e.g. hazardous point protection with a light curtain). Otherwise the operator of the press will be in danger.

If the **Mode for Start/Release input** parameter is set to **Start and static release (stepping)** and **Restart interlock condition** is set to **When Release 1, Release 3 or Start/Release is Low** or **Always**, a valid restart sequence is required after a stop that was caused by a Low **Start/Release** input signal.

The **Enable** output of a Two hand control or of an N-break function block is particularly suitable for connection to the **Start/Release** input.

Release 1 (static)

The **Release 1 (static)** input signal is mandatory. The **Enable** output always changes immediately to Low, if **Release 1 (static)** is Low.

If this function block is used together with a press contact function block (e.g. Eccentric press contact or Universal press contact), its **Enable** output must be connected with the **Release 1 (static)** input of this function block.

Release 2 (start)

The input signal **Release 2 (start)** is optional. If **Release 2 (start)** is configured, the **Enable** output can only change to High (e.g. during switching on), if **Release 2 (start)** is High. If the **Enable** output is High, **Release 2 (start)** is no longer monitored.

Do not use the Release 2 (start) input for safety purposes!

Do not use the **Release 2 (start)** input for initiating safety stops, because this input is evaluated temporarily during the start sequence only. Otherwise the operator of the press will be in danger.



Release 3 (safety)

The **Release 3 (safety)** input signal is an optional signal. The **Enable** output can only change from Low to High if **Release 3 (safety)** is High. If **Release 3 (safety)** is Low and **Upstroke** is Low, the **Enable** output is set to Low and a restart sequence has to occur in accordance with the settings.

If **Release 1 (static)** and **Upstroke** are High and the maximum upstroke muting time is configured to a value higher than 0, the **Release 3 (safety)** signal is muted.

Top

The **Top** input signal is used in order to determine the end of the press cycle (i.e. the press has reached the top dead center). This signal is available at the function blocks Eccentric press contact or Universal press contact. The **Top** input signal is used for single-stroke protection. The **Enable** output changes to Low when the **Top** input signal changes from Low to High.

Do not use the Top input for safety purposes!

Connect the **Top** input only with the **Top** output of an Eccentric press contact or Universal press contact function block or to an equivalent signal source. Do not use the **Top** input for initiating safety stops. Otherwise the operator of the press will be in danger.



Mode for upstroke muting

If the **Max. time for upstroke muting** is not set to 0, the **Upstroke** input has to be connected.

Connect the **Upstroke** input only with the **Upstroke** output of an Eccentric press contact or a Universal press contact function block.

NOTE

In this case, the **Release 3 (safety)** and **Start/Release** input signals are muted (muting of the **Start/Release** input depends on the parameter settings) when the **Enable** output is High and the **Upstroke** input is High. This function block does not carry out a plausibility check of the **Upstroke** input signal. If the **Upstroke** input is High several times during a single press cycle, it is possible to mute the corresponding input of the function block several times. If a signal shall not be muted, it should be connected to the **Release 1 (static)** input by means of an AND function block together with other signals that have to be connected to the **Release 1 (static)** input.

Exclude any danger during the upstroke movement of the press!

If you use upstroke muting, you must ensure that during the upstroke period no hazards are present, e.g. by the up movement itself.



Max. time for upstroke muting

The **Max. time for upstroke muting** can be configured. This time begins with the rising edge (Low to High) of the signal at the **Upstroke** input. If the timer reaches the configured **Max. time for upstroke muting** before a falling edge (High to Low) occurs at the **Upstroke** input, the function block terminates the muting of the **Release 3 (safety)** and **Start/Release** inputs. If from this moment on one of these two inputs becomes Low, the **Enable** output is also set to Low.

Restart input

If the **Restart interlock condition** parameter has been set to **Never**, a **Restart** signal is not required in order to restart the press after any kind of stop. The **Restart interlock condition** parameter can also be set to the following values:

- When Release 1, Release 3 or Start/Release is Low
- When Release 1 or Release 3 is Low or Top changes to High
- Always

- When Release 1 or Release 3 is Low

This parameter determines when a **Restart** signal is expected as input signal for the function block.

If the **Enable** output changes to Low because of the above-mentioned settings of the configuration parameters for the **Restart interlock condition**, the **Enable** output can only be reset after a valid restart sequence with a Low-High-Low transition (minimum 100 ms or 350 ms; shorter pulses and pulses longer than 30 s will be ignored) has been carried out.

Ignore Release 3 (safety) for Restart interlock while in Top position

The **Ignore Release 3 (safety) for Restart interlock while in Top position** parameter prevents the restart interlock being activated if the **Release 3 (safety)** input changes to Low during a regular stop of the press. This means that if the **Ignore Release 3 (safety) for Restart interlock while in Top position** parameter is configured as **Yes** and the **Enable** output changes to Low as a result of the **Top** input changing to High, then the **Restart required** output will not change to High if the **Release 3 (safety)** input changes to Low as long as the press has not been restarted.

Output signals of the function block

Restart required

The **Restart required** output is High, when a valid restart sequence is expected at the **Restart** input.

Sequence/timing diagrams

Fig. 200: Sequence/timing diagram for the Press single stroke function block when Start/Release is configured in stepping mode

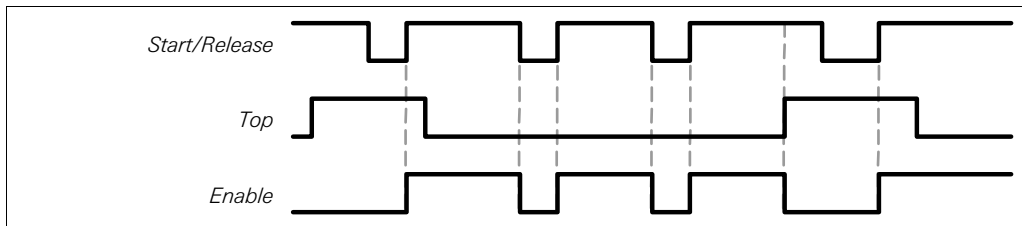


Fig. 201: Sequence/timing diagram for the Press single stroke function block when Start/Release is configured in start only mode

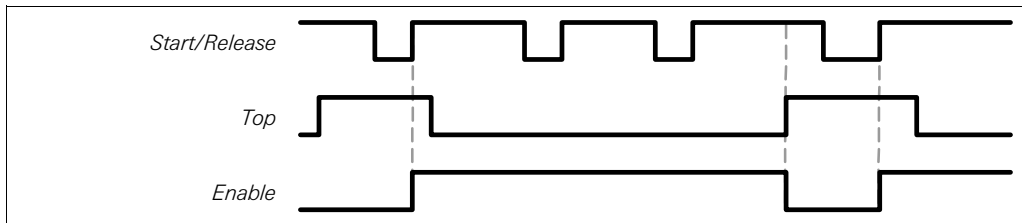
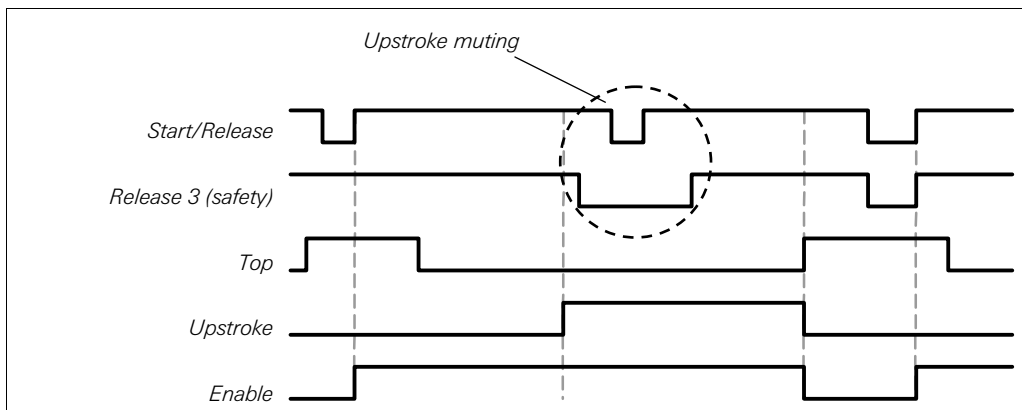


Fig. 202: Sequence/timing diagram for the Press single stroke function block with upstroke muting of Start/Release and Release 3 (safety)



8.11.3 Press automatic

Function block diagram

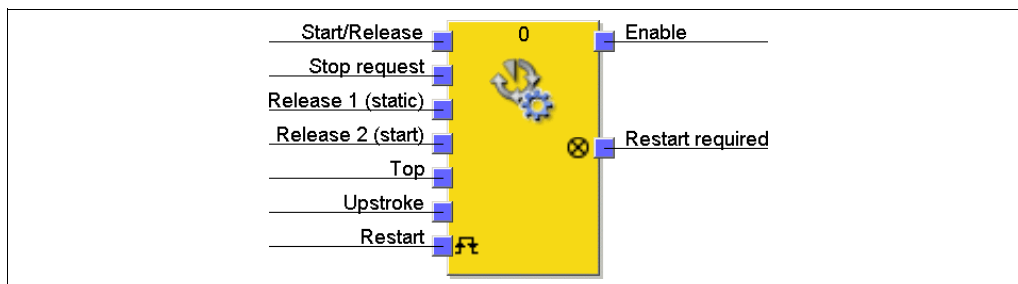


Fig. 203: Logic connections for the Press automatic function block

General description

The Press automatic function block is used in connection with press applications in which the workpieces are moved automatically to and from the press, but where occasionally access to the press is required, for example to change a tool.

To this purpose the function block can generate a stop signal for the press (i.e. the **Enable** output changes to Low) in a position in which the tool can be changed easily (e.g. in the top position), when a stop has been requested before.

Parameters of the function block

Parameter	Possible values
Restart interlock condition	<ul style="list-style-type: none"> • After every stop • Never
Stop request condition	<ul style="list-style-type: none"> • When Start/Release input is Low • When Stop input is High
Upstroke input	<ul style="list-style-type: none"> • With • Without
Release 2 (start) input	<ul style="list-style-type: none"> • With • Without
Min. restart pulse time	<ul style="list-style-type: none"> • 100 ms • 350 ms

Tab. 100: Parameters of the Press automatic function block

Ensure that the transitions of the signals for restarting fulfil the requirements of the safety standards and regulations!

In case of a short-circuit to High (to 24 V DC) at a physical input, the evaluated signal can have a pulse when the signal is reset due to the short-circuit detection. If such a pulse can result in a dangerous state in the machine, the following points have to be observed:

- Ensure protected cable laying for the signal line (due to cross-circuiting to other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.



Input parameters and input signals of the function block

Stop request condition

The **Stop request condition** parameter determines the Stop state of the Press automatic function block. If this parameter is configured as **When Start/Release input is Low**, the **Start/Release** input signal is used to control the **Enable** output directly. If configured to **When Stop input is High**, the **Enable** output changes to Low, when the **Stop request** input is High.

In both cases the **Enable** output changes to High, when the following conditions are fulfilled:

- A transition from Low to High occurs at the **Start/Release** input; and
- the **Stop request** input is Low, if it is connected; and
- no other reason is present that would normally trigger a stop signal, e.g. **Release 1 (static)** is Low.



Do not use the Start/Release and Stop inputs for safety stops!

Independent of the configured stop request mode, the inputs **Start/Release** and **Stop request** must not be used for initiating safety stops. These inputs can only be used for setting automation control stop requests. Signals initiating safety stops (e.g. safety stop) must be connected to the **Release 1 (static)** input of the function block.

Upstroke input

If the **Upstroke input** parameter is configured as **With**, a High signal at the **Upstroke** input allows the press to stop both during the downstroke and in the top position. If this parameter is set to **Without**, regular stops are only possible in the top position.

NOTE

Connect the **Upstroke** input only to the **Upstroke** output of an Eccentric press contact or a Universal press contact function block.

Start/Release

The **Start/Release** input signal is used to provide signals for the beginning and end of the press movement. If a rising edge (Low to High) is detected at the **Start/Release** input, the **Enable** output becomes High, provided that the **Stop request** input is Low and no other reason is present that would normally trigger a stop signal, e.g. **Release 1 (static)** is Low. A valid restart sequence can be required before a signal transition of **Start/Release** if the **Restart interlock condition** parameter is set to **After every stop**. If you connect a command device (e.g. a two-hand control) to the **Start/Release** input, you must ensure that unintentional restarting is not possible.

Stop request

If the **Stop request condition** parameter is set to **When Stop input is High**, the **Stop request** input is used to signal a stop to the press. When the **Stop request** input is High, the **Enable** output is set to Low.

This input should only be used if the **Stop request condition** parameter has been set to **When Stop input is High**. The **Stop request** input is not used when the **Stop request condition** parameter has been set to **When Start/Release input is Low**. A valid restart sequence can be required before a signal transition of **Start/Release** when the **Restart interlock condition** parameter is set to **After every stop**. The **Stop request** input is designed for the connection of signals that are not safety-relevant (e.g. from a programmable logic controller (PLC)). Safety-relevant signals may only be connected to the **Release 1 (static)** input, not to the **Stop request** input.

Release 1 (static)

The input signal **Release 1 (static)** is mandatory. The **Enable** output always changes immediately to Low, if **Release 1 (static)** is Low.

If this function block is used together with a press contact function block (e.g. Eccentric press contact or Universal press contact), its enable signal must be connected with the **Release 1 (static)** input of this function block.

Release 2 (start)

The input signal **Release 2 (start)** is optional. When **Release 2 (start)** is configured, the **Enable** output can only change to High (e.g. during switching on), when **Release 2 (start)** is High. When the **Enable** output is High, **Release 2 (start)** is no longer monitored.

Do not use the Release 2 (start) input for safety purposes!

Do not use the **Release 2 (start)** input for initiating safety stops, because this input is evaluated temporarily during the start sequence only. Otherwise the operator of the press will be in danger.



Top

The **Top** input signal is used in order to determine the end of the press cycle (i.e. the press has reached the top dead center). This signal is available at the function blocks Eccentric press contact or Universal press contact.

Do not use the Top input for safety purposes!

Connect the **Top** input only with the **Top** output of an Eccentric press contact or Universal press contact function block or to an equivalent signal source. Do not use the **Top** input for initiating safety stops. Otherwise the operator of the press will be in danger.



Restart input

If the **Restart interlock condition** parameter has been set to **Never**, a **Restart** signal is not required in order to restart the press after any kind of stop.

If the **Restart interlock condition** parameter has been set to **After every stop** and the **Enable** output changes to Low, the **Enable** output can only be reset after a valid restart sequence with a Low-High-Low transition (minimum 100 ms or 350 ms; shorter pulses and pulses longer than 30 s will be ignored) has been carried out.

Output signals of the function block

Restart required

The **Restart required** output is High, when a valid restart sequence is expected at the **Restart** input.

Sequence/timing diagram

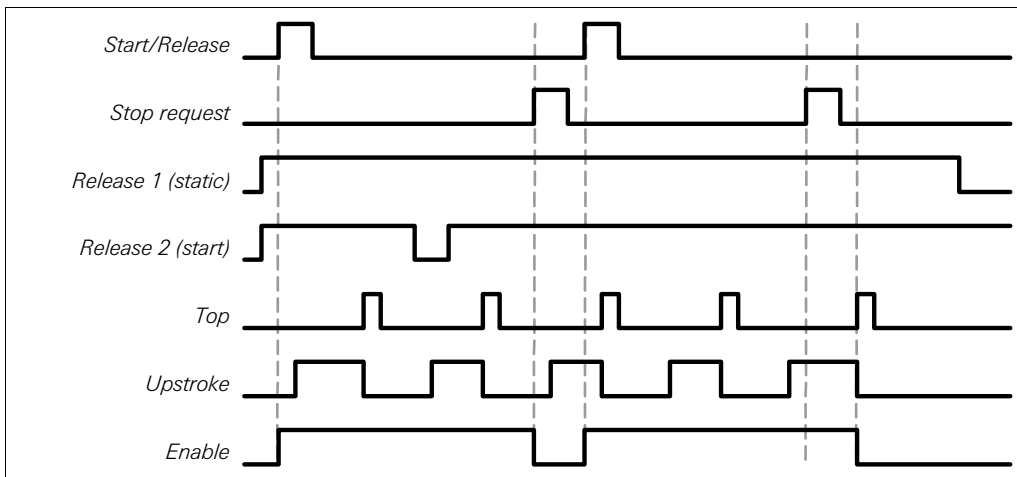


Fig. 204: Sequence/timing diagram for the Press automatic function block using the Stop request and Upstroke inputs

8.11.4 N-break (press with N-PSDI mode)

Function block diagram

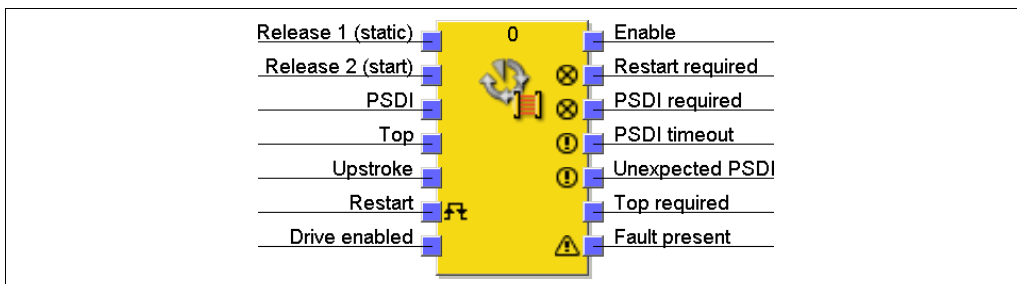


Fig. 205: Logic connections for the N-break function block

General description

The N-break function block is used for press applications with Press Sensing Device Initiation (PSDI) mode.

Conform to the safety regulations for PSDI mode!

The requirements for PSDI mode are specified in local, regional, national and international standards. Always implement PSDI applications in conformity with these standards and regulations as well as in conformity with your risk analysis and avoidance strategy. If more than one mode is set up in which the ESPE (e.g. safety light curtain) is not used, the ESPE has to be deactivated in this mode so that it is clear that the ESPE is currently not active in protective operation.



Logic programming – Function blocks

If more than one ESPE (e.g. safety light curtain) is used in an application that uses the N-PSDI functions, only one of the ESPEs may be used to fulfil the requirements for N-PSDI mode.

In conformity with EN 692 and EN 693 for press applications the number of breaks is limited to 1 or 2. Other applications depend on the applicable standards.

Prevent access to hazardous movements!

Press systems with a configuration that would allow a person to enter, to cross through and to leave the protective field of an ESPE are not permitted for PSDI mode.

This function block defines a specific sequence of events that trigger a press cycle. "Breaks" are defined as the transition from High to Low to High of the **PSDI** input signal. In PSDI mode of a press an indirect manual triggering of a press cycle is carried out based on a predefined number of "breaks" in the ESPE. If the ESPE (e.g. safety light curtain) detects that the operating movements of the operator related to the insertion or removal of parts have ended and that the operator has withdrawn all body parts from the protective field of the ESPE, the press may trigger automatically.

The N-break function block can be used in connection with the Universal press contact or Press single stroke function blocks and an input for a safety light curtain. The **Enable** signal of this function block controls, for example, the **Start/Release** input of a Press single stroke function block.

The N-break function block checks whether the start sequence is valid and when the break counter or the function block have to be reset.

Parameters of the function block

Parameter	Possible values
Number of PSDI pulses	1 to 8
Mode	<ul style="list-style-type: none"> • Standard • Sweden
Max. time for upstroke muting	0 = disabled, 1 to 7200 s. The Upstroke input is available only if the value is not set to 0.
Max. time for PSDI pulses (timeout)	0 = disabled, 1 to 500 s
Condition for Release 2 (start) input	<ul style="list-style-type: none"> • Without • Necessary for first start • Necessary for every start
Start of first PSDI pulse (PSDI input Low => High)	<ul style="list-style-type: none"> • After top has been reached • After the start of upstroke
Restart interlock	<ul style="list-style-type: none"> • For all stops • For stops in downstroke and in top (in upstroke ignored) • Without
Min. restart pulse time	<ul style="list-style-type: none"> • 100 ms • 350 ms
Valid start position (Restart and PSDI pulses)	<ul style="list-style-type: none"> • Everywhere • Only in top
Min. PSDI pulse time (Low time)	<ul style="list-style-type: none"> • 100 ms • 350 ms
Use Fault present	<ul style="list-style-type: none"> • With • Without

Tab. 101: Parameters of the N-break function block

Ensure that the transitions of the signals for restarting fulfil the requirements!

In case of a short-circuit to High (to 24 V DC) at a physical input, the evaluated signal can have a pulse when the signal is reset due to the short-circuit detection. If such a pulse can result in a dangerous state in the machine, the following points have to be observed:

- Ensure protected cable laying for the signal line (due to cross-circuiting to other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.



Input parameters and input signals of the function block

Standard or Sweden mode

The **Mode** parameter specifies the complete start sequence for the N-Break function block. **Standard** mode requires that the configured number of breaks is carried out, followed by a valid restart sequence.

Sweden mode first requires a valid restart sequence, followed by the configured number of breaks.

Requirements for the start sequence

If the **Enable** output changes to Low because of one of the following conditions, a complete start sequence can be necessary:

- **Release 1 (static)** is Low,
- the **Unexpected PSDI** output is High, while Cycle = 0 and there is no active upstroke muting and no stop at the top dead center,
- in case of a PSDI timeout,
- after the **Drive enabled** has been switched on

If the **Unexpected PSDI** output is High and the **Enable** output is Low and the **PSDI** input is also Low and **Restart interlock** is set to **Without**, a restart is possible without a complete restart sequence. This can also apply during the press upstroke if **Restart interlock** is set to **For stops in downstroke and in top (in upstroke ignored)**.

The minimum break time at the **PSDI** input is 100 ms or 350 ms. Shorter breaks are not evaluated as valid, i.e. they are ignored. If the **Condition for Release 2 (start) input** is configured as **Necessary for first start** or as **Necessary for every start**, the **Release 2 (start)** input also must be High if a complete start sequence is required.

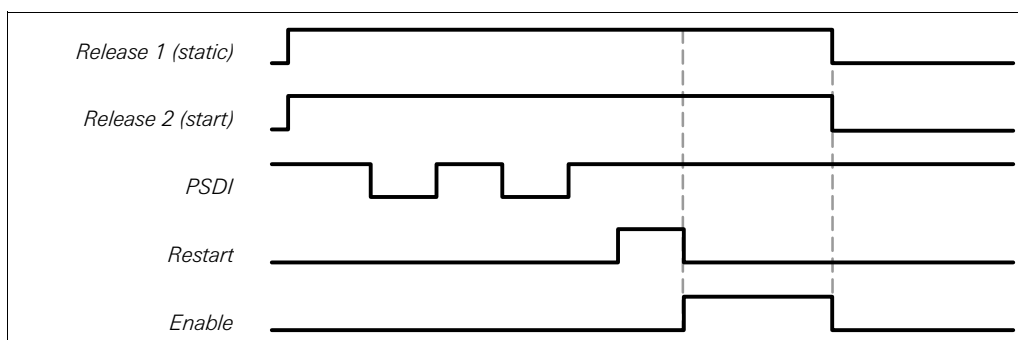


Fig. 206: Sequence/timing diagram for a complete start sequence in Standard mode in two-cycle mode

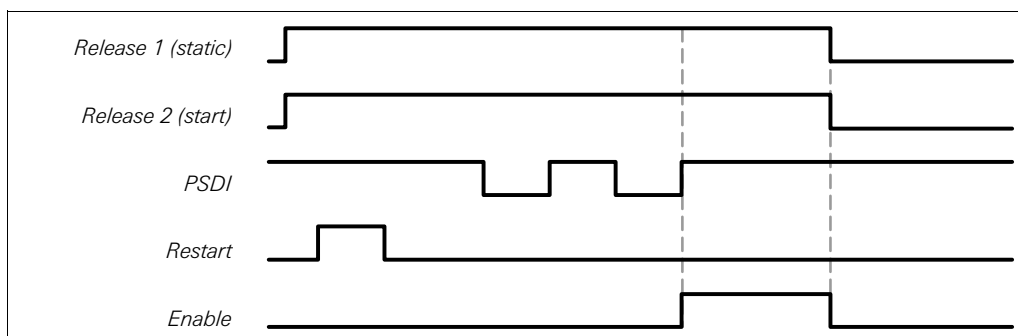


Fig. 207: Sequence/timing diagram for a complete start sequence in Sweden mode in two-cycle mode

After the initial complete start sequence has been executed and the press has completed a press cycle, the **Top** input must indicate that the press has currently reached the top dead center. This is indicated by a rising edge (Low to High) of the **Top** input. When this happens, the internal break counter is reset.

A cycle start sequence is required in order to trigger a subsequent cycle. In this case, the **Enable** output is set to High when the configured number of breaks has occurred and the remaining configured conditions have been fulfilled (e.g., **Condition for Release 2 (start) input** can be configured as **Necessary for every start**).

Max. time for PSDI pulses (timeout)

The **Max. time for PSDI pulses (timeout)** parameter specifies the required time both for a complete start sequence and for a cycle start sequence. If the **Max. time for PSDI pulses (timeout)** is exceeded, the **PSDI timeout** output changes to High. In this case, a complete start sequence is necessary so that the **Enable** output can return to High (e.g. in order to start the press). The PSDI timer starts when the press is stopped at the top dead center (i.e. the **Top** input changes from Low to High) and after all other stop conditions have been fulfilled.

The basic setting for the **Max. time for PSDI pulses (timeout)** is 30 s in accordance with the maximum PSDI time allowed for eccentric presses (defined in EN 692). If the **Max. time for PSDI pulses (timeout)** is set to 0, PSDI time monitoring is disabled.

Start of first PSDI pulse (PSDI input Low => High)

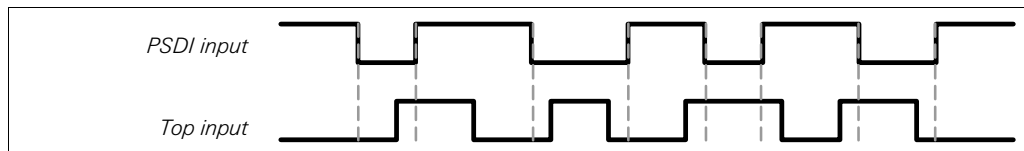
The **Start of first PSDI pulse** parameter determines under which circumstances a break is regarded as valid.

If the **Start of first PSDI pulse** parameter is set to **After the start of upstroke**, a break is valid if the beginning of the break (i.e. falling edge (High to Low) at the **PSDI** input) occurs after the rising edge at the **Upstroke** input, no matter whether the **Top** input has changed to High already.

If the **Start of first PSDI pulse** parameter is set to **After top has been reached**, a break is only valid if the beginning of the break (i.e. falling edge (High to Low) at the **PSDI** input) occurs after the rising edge at the **Top** input.

In both cases, the end of the break (i.e. rising edge (Low to High) at the **PSDI** input) must occur after the rising edge at the **Top** input, no matter whether the **Top** input is still High or has changed back to Low already.

Fig. 208: Valid breaks 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 enabled. Otherwise the **Enable** output changes to Low as soon as the **PSDI** input changes to Low (i.e. at the beginning of the break).

Upstroke muting and Maximum time for upstroke muting

Upstroke muting allows bypassing of 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 higher than 0. Upstroke muting is inactive when the **Max. time for upstroke muting** parameter is set to 0.

When upstroke muting is activated ...

- it is imperative that the **Upstroke** input is connected to a suitable signal. This can be the **Upstroke** output e.g. of the Eccentric press contact or of the Universal press contact function block.
- the **PSDI** input of the function block is bypassed if the **Upstroke** input is High and the **Top** input remains Low.

The function block does not check the **Upstroke** input for plausibility. This means that it is possible to bypass the **PSDI** input several times if the **Upstroke** input is activated several times during a single press cycle.

Exclude any danger during the upstroke movement of the press!

If you use upstroke muting, you must ensure that during the upstroke period no hazards are present, e.g. by the up movement itself.

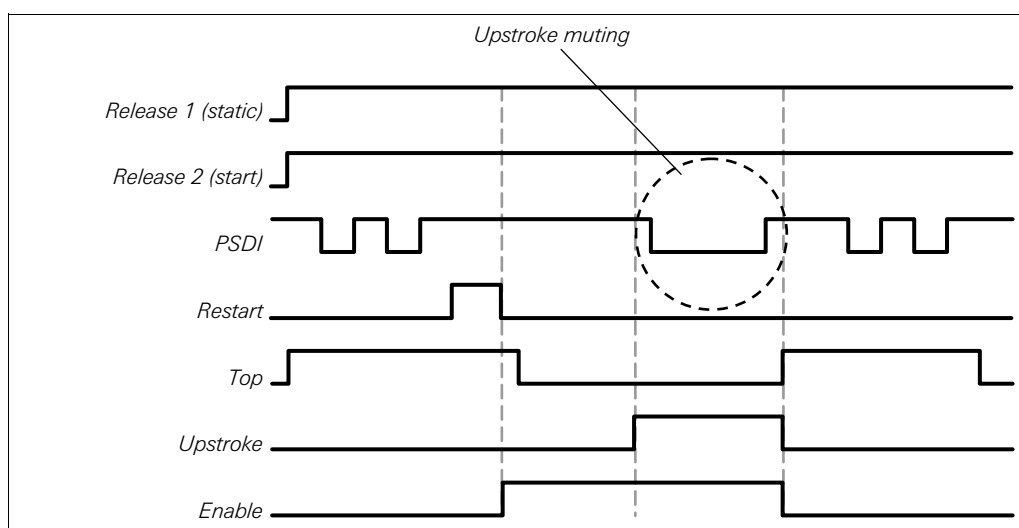


Fig. 209: Sequence/timing diagram for Upstroke muting in Standard mode in two-cycle mode

The **Max. time for upstroke muting** can be configured. The upstroke muting timer starts with a rising edge (Low to High) at the **Upstroke** input. If the timer reaches the configured **Max. time for upstroke muting** before another rising edge occurs at the **Upstroke** input, upstroke muting is interrupted and, if the **PSDI** input is Low, the **Enable** output is set to Low. When a second rising edge occurs at the **Upstroke** input, upstroke muting begins again.

Valid start position (Restart and PSDI pulses)

If the **Valid start position (Restart and PSDI pulses)** parameter is set to **Only in top**, a start of the press is possible only in the top position. In any other position a start is inhibited. If the press has been stopped e.g. by interruption of the light curtain during a downstroke, you must change to another press operating mode (e.g. in combination with the press setup function block) to move the press back to the top position, because the N-break function block inhibits a restart with this parameter setting.

If the **Valid start position (Restart and PSDI pulses)** parameter is set to **Only in top**, the optional **Drive enabled** input must be connected in order to monitor whether the press is currently running or has been stopped. This must be the signal which directly controls the press. Typically the **Drive enabled** input will be connected via a jump address or a CPU marker to the logic editor signal which is connected to the physical output for the press.

Do not connect any physical input signals to the **Drive enabled** input. Connect the signal that controls the physical output for the press drive using a jump address or a CPU marker.

NOTE

- If a jump address is used, make sure that this signal is a loop-back. This is indicated by a clock icon on the destination tag of the jump address. To this purpose, connect the outputs of this function block to the following function blocks before you connect the jump address to the **Drive enabled** input. This applies especially if all connections to the following function blocks are also realized using jump addresses.
- If a CPU marker is used, then a Routing function block must be used to split the signal to the physical output for the press drive and to the CPU marker output.

If the **Enable** output changes to Low as a result of either the **Release 1 (static)** input or the **PSDI** input has changed to Low, the diagnostic output **Top required** changes to High.

A restart of the press is prevented until the **Top** input has changed back to High and no restart in another operating mode has occurred.

Release 1 (static)

The input signal **Release 1 (static)** is mandatory. The **Enable** output always changes immediately to Low, if **Release 1 (static)** is Low.

If this function block is used together with a press contact function block (e.g. Eccentric press contact or Universal press contact), its **Enable** output must be connected with the **Release 1 (static)** input of this function block.

Release 2 (start)

The input signal **Release 2 (start)** is optional. If **Release 2 (start)** is configured, the **Enable** output can only change to High (e.g. during switching on), when **Release 2 (start)** is High. When the **Enable** output is High, **Release 2 (start)** is no longer monitored.



Do not use the Release 2 (start) input for safety purposes!

Do not use the **Release 2 (start)** input for initiating safety stops, because this input is evaluated temporarily during the start sequence only. Otherwise the operator of the press will be in danger.

Top

The **Top** input signal is used in order to determine the end of the press cycle (i.e. the press has reached the top dead center). This signal is available at the function blocks Eccentric press contact or Universal press contact.



Do not use the Top input for safety purposes!

Connect the **Top** input only with the **Top** output of an Eccentric press contact or Universal press contact function block or to an equivalent signal source. Do not use the **Top** input for initiating safety stops. Otherwise the operator of the press will be in danger.

Upstroke input

If upstroke muting is enabled (i.e. if the **Max. time for upstroke muting** is greater than 0), the **PSDI** input of the function block is bypassed if the **Upstroke** input is High and the **Top** input remains Low.

NOTE

Connect the **Upstroke** input only with the **Upstroke** output of an Eccentric press contact or a Universal press contact function block.

Restart input

If **Restart interlock** has been set to **Without**, a **Restart** signal is not required in order to restart the press after the **Enable** output has changed to Low.

If **Restart interlock** has been set to **For all stops** and the **Enable** output changes to Low, the **Enable** output can only be reset after a valid restart sequence with a Low-High-Low transition (minimum 100 ms or 350 ms; shorter pulses and pulses longer than 30 s will be ignored) has been carried out. The only exception to this rule is formed by the cycle beginning. In this case the **Restart interlock** parameter does not have any effect on the function block.

If **Restart interlock** has been set to **For all stops** and the **Max. time for upstroke muting** has been configured to 0 s, a Low signal at the **PSDI** input during the runup sets the **Enable** output immediately to Low.

If **Restart interlock** has been set to **For all stops** and upstroke muting is active, the **Enable** output remains High until the **Top** input becomes High, thus indicating that the press cycle has been completed. In this case, a complete restart sequence is required.

If **Restart interlock** has been set to **For stops in downstroke and in top (in upstroke ignored)** and the **Upstroke** input is High, the **Enable** output remains High until **Top** becomes High, thus indicating that the press cycle has been completed. In this case, a cycle start sequence is required.

If the **PSDI** input changes after the **Max. time for upstroke muting** has elapsed from High to Low and back to High, the **Enable** output also changes from High to Low and

back to High. The setting for this parameter does not have any effect when the **Restart** and **Upstroke** input signals remain unconnected.

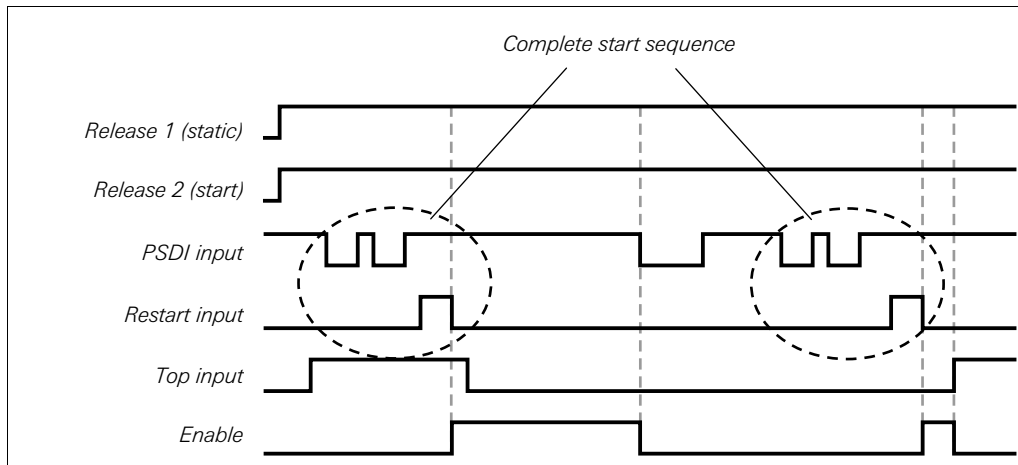


Fig. 210: Sequence/ timing diagram, when the PSDI input is Low, Upstroke muting is inactive and Restart interlock is set to "For all stops".

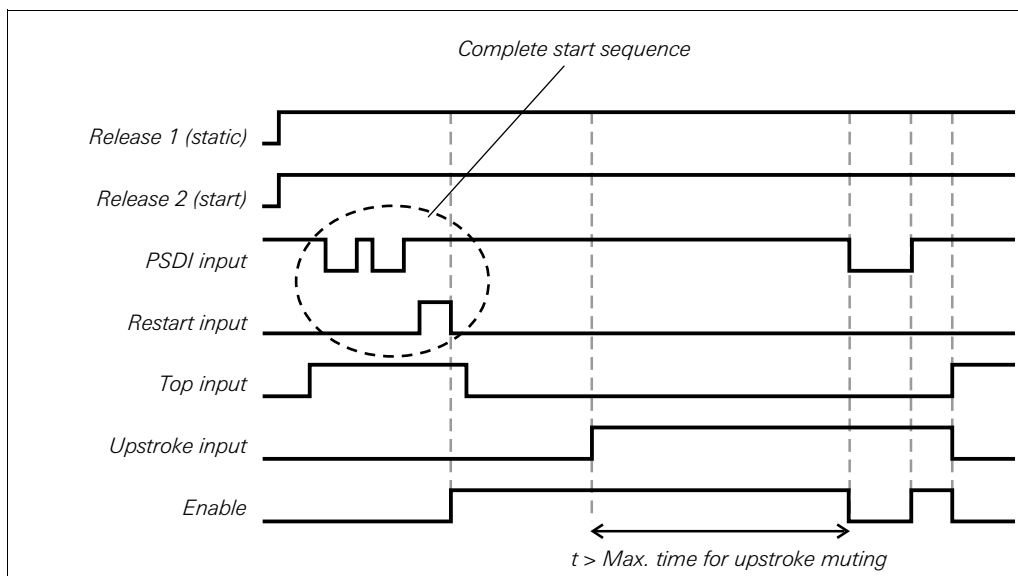


Fig. 211: Sequence/ timing diagram when the PSDI input is Low, Max. time for upstroke muting > 0 and Restart interlock is set to "For stops in downstroke and in top (in upstroke ignored)".

Output signals of the function block

Restart required output

The **Restart required** output is High when a valid restart sequence is expected at the **Restart** input.

PSDI required output

The **PSDI required** output is High when a break is expected at the **Restart** input.

Unexpected PSDI

The **Unexpected PSDI** output is High when a valid start sequence has been carried out and the **PSDI** input changes from High to Low while no muting is active and no break is expected. If **Unexpected PSDI** is High, a valid restart sequence generally has to be carried out before the **Enable** output can be set to High.

If the **Unexpected PSDI** output is High and the **Enable** output is Low and the **PSDI** input is also Low and **Restart interlock** is set to **Without**, a restart is possible without a complete restart sequence. This can also apply during the press upstroke if **Restart interlock** is set to **For stops in downstroke and in top (in upstroke ignored)**.

Logic programming – Function blocks

Tab. 102: Error states and information on resetting for the N-break function block

Error states and information on resetting

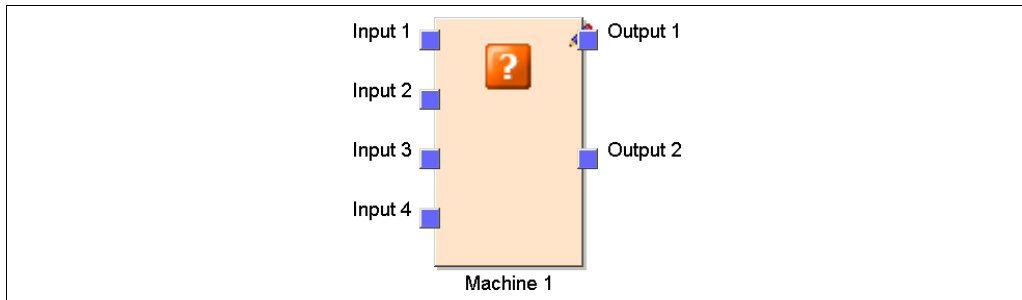
Diagnostics outputs	Resetting the error state	Remarks
Unexpected PSDI	If an interruption of the protective field occurs, the PSDI input generally has to return to High, followed by a valid restart sequence, in order to reset the error. If the Unexpected PSDI output is High and the Enable output is Low and the PSDI input is also Low and Restart interlock is set to Without or For stops in downstroke and in top (in upstroke ignored) , a restart is possible without a complete restart sequence.	The Enable output changes to Low and Fault present changes to High, if Unexpected PSDI or PSDI timeout is High.
PSDI timeout	For PSDI timeout the error is reset by a valid restart sequence.	

8.12 User defined function blocks

8.12.1 Grouped function block

Function block diagram

Fig. 212: Logic connections for the Grouped function block



You can select groups of function blocks in order to create a single grouped function block. The typical purpose of a grouped function block is to simplify the re-use of groups of logic and to reduce the number of function blocks on a page.

A grouped function block has the following characteristics:

- It can have a maximum of 8 inputs and 8 outputs.
- It cannot contain the Fast Shut-off function block nor another grouped or customized function block.
- The icon representing the grouped function block is chosen from a fixed library within the samos®PLAN.
- It is created within the logic editor, but is not listed in the function block list.
- It is saved with the project file. Upon re-opening the project file on another PC, the grouped function block will be displayed.
- It can be saved as a customized function block.

NOTE

When determining the total number of function blocks within a project, the grouped function block is not counted as a single function block, but rather the total number of blocks used within it.

How to create a grouped function block:

- ➔ Select the function blocks which are to be grouped.
- ➔ Right click on one of the selected function blocks to call up the context menu.

Logic programming – Function blocks

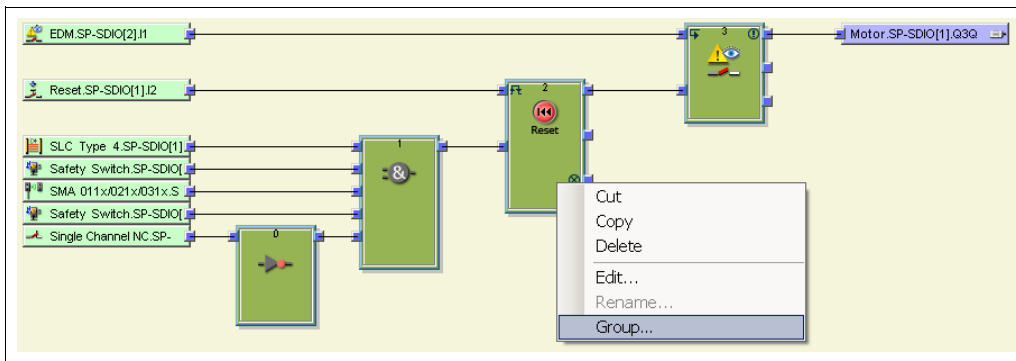


Fig. 213: Creating a grouped function block

- ➔ Click on **Group**. The **Edit Function Block Details** dialog opens.

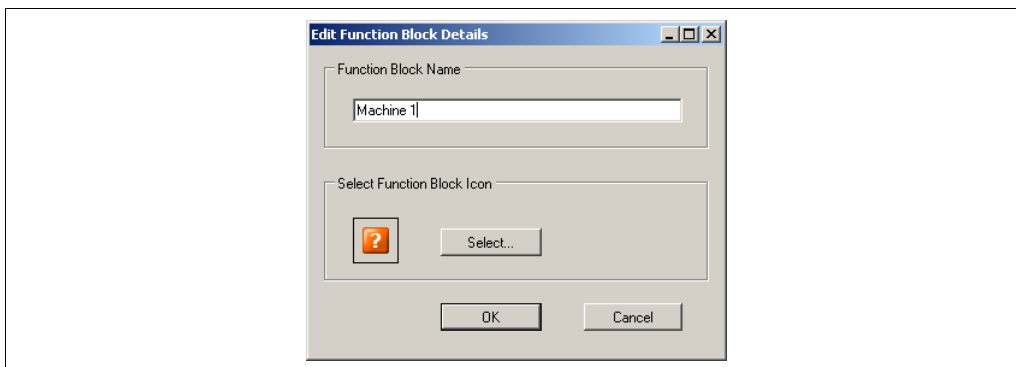


Fig. 214: Edit Function Block Details dialog for the Grouped function block

- ➔ Enter a name for the new grouped function block.
- ➔ If you want to assign another icon to the new grouped function block, click on **Select...** to open the **Select Icon** dialog. You can choose the icon from a fixed library.

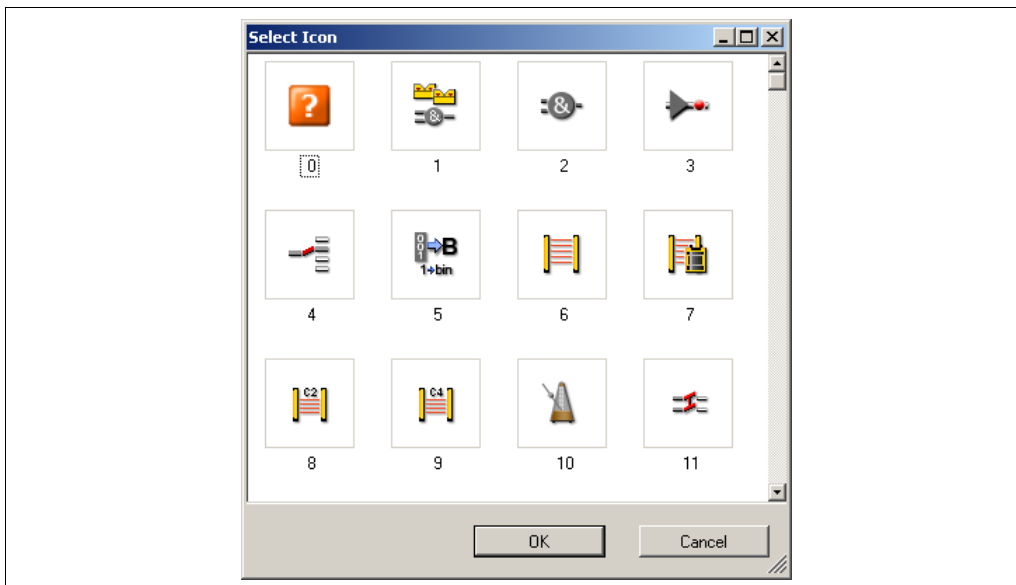
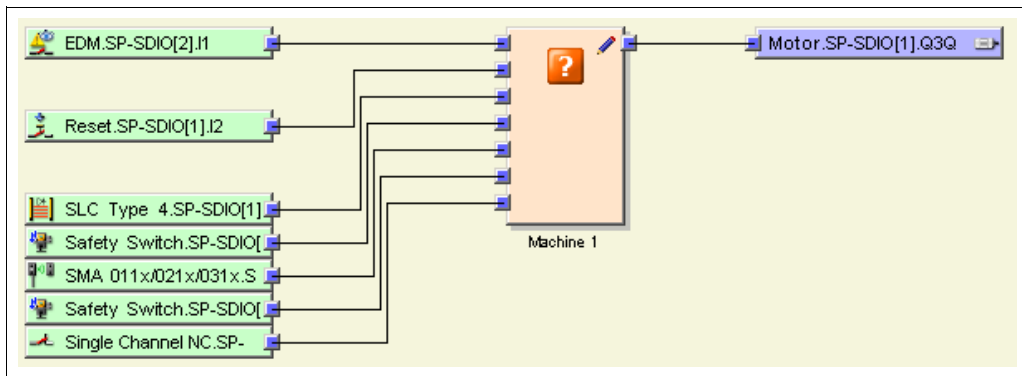


Fig. 215: Select Icon dialog for the Grouped function block

- ➔ Select the desired icon and click **OK**.
- ➔ Back in the **Edit Function Block Details** dialog, click **OK** to confirm your changes and to leave the dialog. The selected function blocks will be reduced to a single grouped function block on the worksheet for the main program.

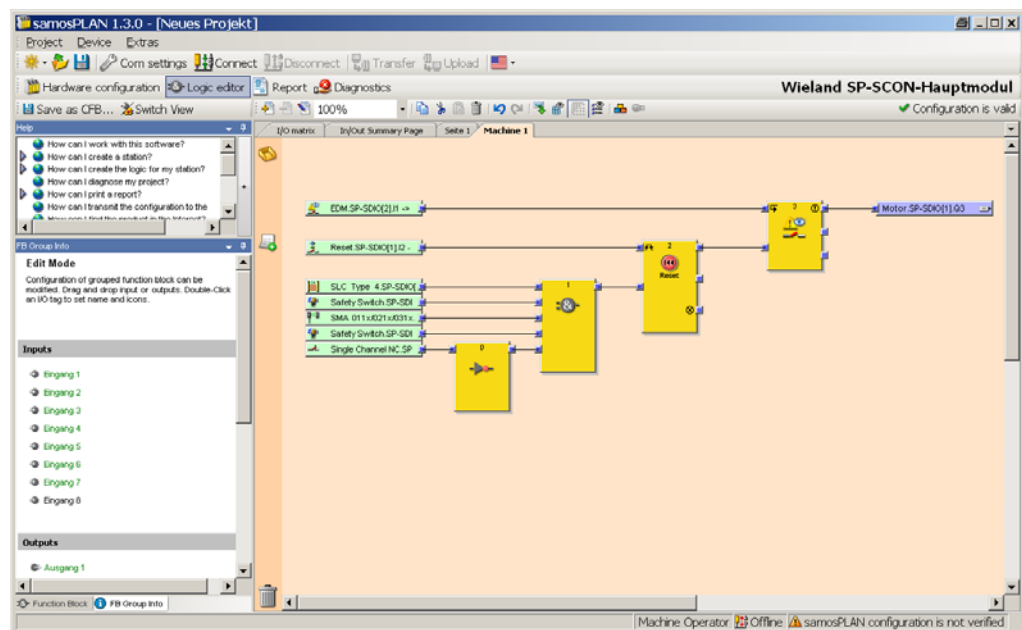
Logic programming – Function blocks

Fig. 216: New grouped function block on the worksheet



The content of the new grouped function block is stored on a new page. In the example, the name of the new grouped function block is *Machine 1*. The worksheet for the grouped function block is displayed orange.

Fig. 217: New logic editor page for the new grouped function block



NOTE

The name and the icon associated with a grouped function block can be edited here by clicking on the function block's icon in the **FB Group Info** view.

By clicking on the associated tab (here: *Machine 1*), you can edit the grouped function block.

How to add inputs and outputs to a grouped function block:

- ➔ Click on the tab for the grouped function block.
- ➔ Switch to the **FB Group Info** view on the left side of the screen.
- ➔ Drag and drop inputs or outputs onto the function block worksheet and attach them within the logic as needed.
- ➔ Double click on an input or output to edit its tag name.

Logic programming – Function blocks

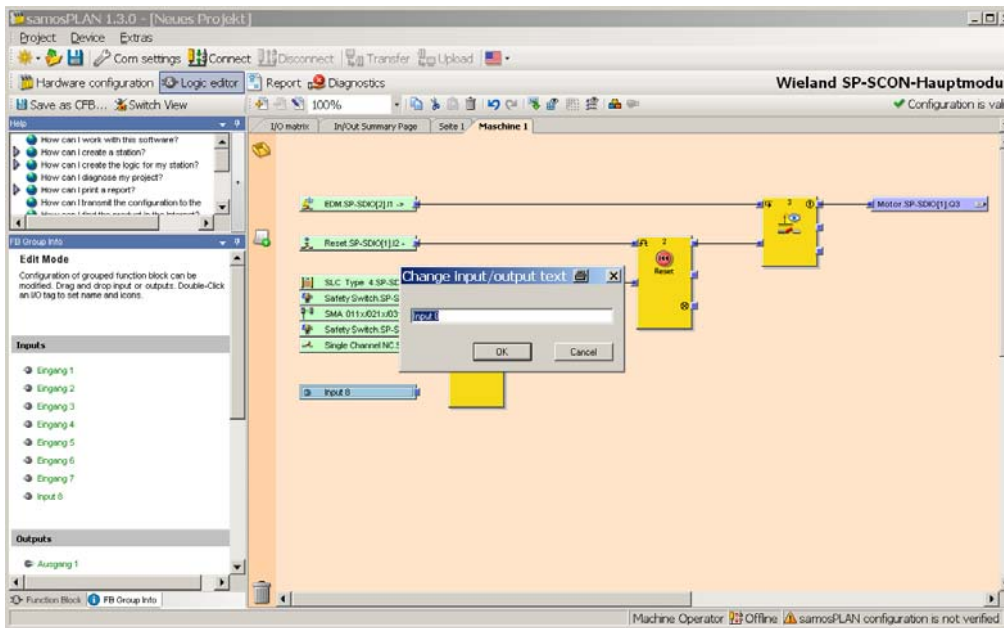


Fig. 218: Adding inputs and outputs to a grouped function block

The inputs and outputs that have been added to the grouped function block will appear on the function block itself in the main program and devices can be attached to them. Once a device is attached, it will be displayed in the logic of the grouped function block when the view is switched to *external view*.

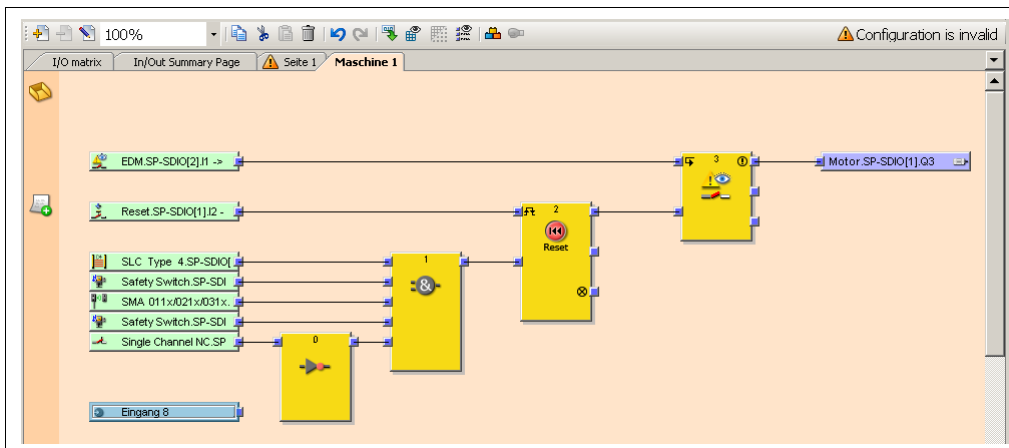


Fig. 219: Grouped function block with devices attached

To switch between the internal tag names of the grouped function block (internal view) and the external I/O descriptions (external view), click on **Switch View** in the toolbar.

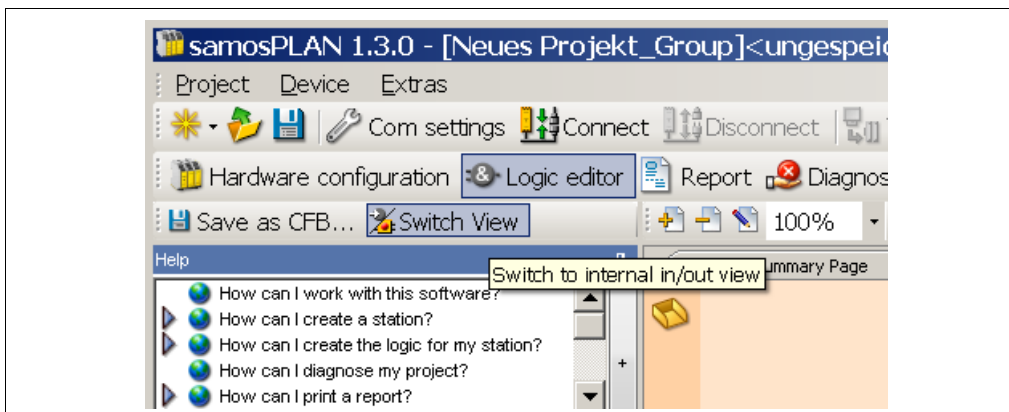


Fig. 220: Switching between internal and external view

- The *internal view* shows the grouped function block's tag names for its inputs and outputs.
- The *external view* shows what is connected to the grouped function block.

Logic programming – Function blocks

How to transfer a grouped function block to another PC:

- ➔ Save the project file and open it on the other PC. Grouped function blocks contained in the project will be imported automatically.

NOTES

- The import of function blocks must be enabled in the hardware configuration view (**Settings, General** tab, disable the option **Don't ask to import customized function blocks**). You will be asked to confirm the import, when you open the project file.
- Importing grouped function blocks requires samos®PLAN version 1.1.0 or higher.

8.12.2 Customized function block

Once a grouped function block is created, it is possible to lock and import it into the function block selection field for use in future project files. The resulting function block is called a customized function block.

Function block diagram

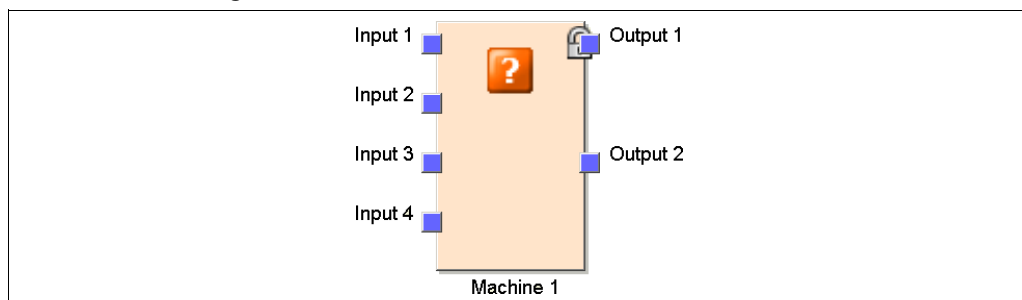


Fig. 221: Logic connections for the customized function block

A customized function block has the following characteristics:

- It can have a maximum of 8 inputs and 8 outputs.
- It cannot contain the Fast Shut-off function block nor another grouped or customized function block.
- The icon representing the customized function block may either be user defined or chosen from a fixed library within the samos®PLAN software.
- It is created within the logic editor, will be listed with the other function blocks in the function block list and will be available in all new projects on the same PC.
- Upon opening a project file containing customized function blocks on another PC, you have the following options:

You can import the customized function blocks into the function block listing on the new PC for further use in new projects.

Or you can import the customized function blocks for this project only. In this case, they will not be listed in the function block list.

NOTE

When determining the total number of function blocks within a project, the customized function block is not counted as a single function block, but rather the total number of blocks used within it.

How to create a customized function block:

In order to create a customized function block, you must have already created it as a grouped function block (see section 8.12.1 “Grouped function block” on page 180).

- ➔ Open the grouped function block view by clicking on its tab,
- ➔ Click on **Save as CFB...** in the toolbar. The **Edit Function Block Details** dialog opens.

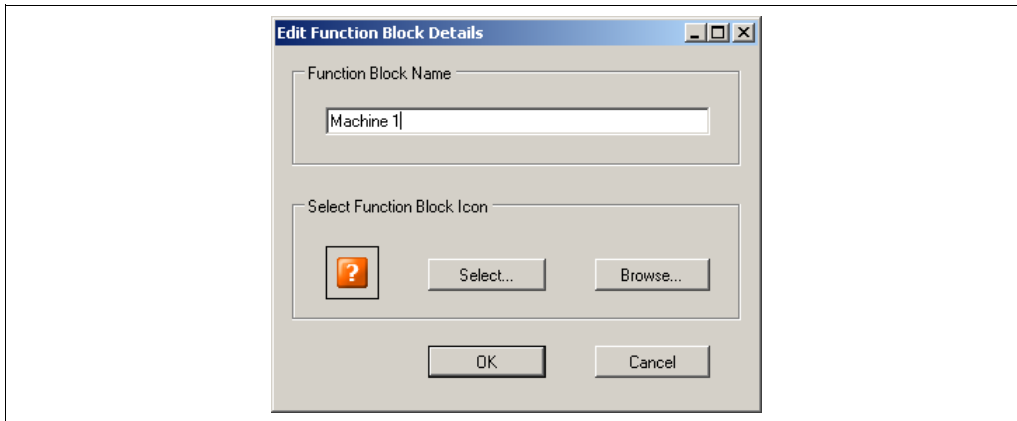


Fig. 222: Edit Function Block Details dialog for the customized function block

- ➔ Enter a name for the new customized function block.
- ➔ If you want to assign another icon to your new customized function block, you have two possibilities:

Click on **Browse...** to choose a user defined icon.

Or click on **Select...** to open the **Select Icon** dialog. You can choose the icon from a fixed library.

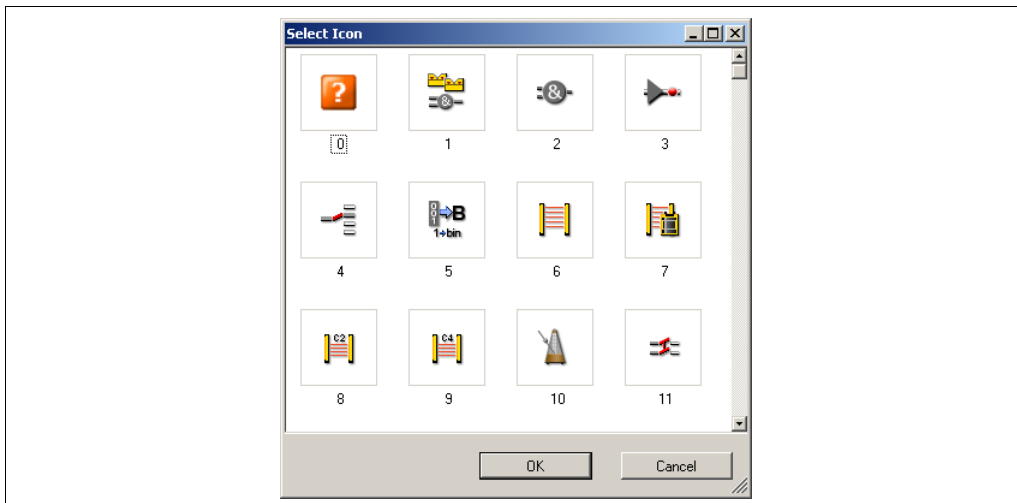


Fig. 223: Select Icon dialog for the customized function block

- ➔ Select the desired icon and click **OK**.
- ➔ Back in the **Edit function block details** dialog, click **OK** to confirm your changes. The selected grouped function block will appear in the function block list as a customized function block and will be available in all new projects on the same PC.

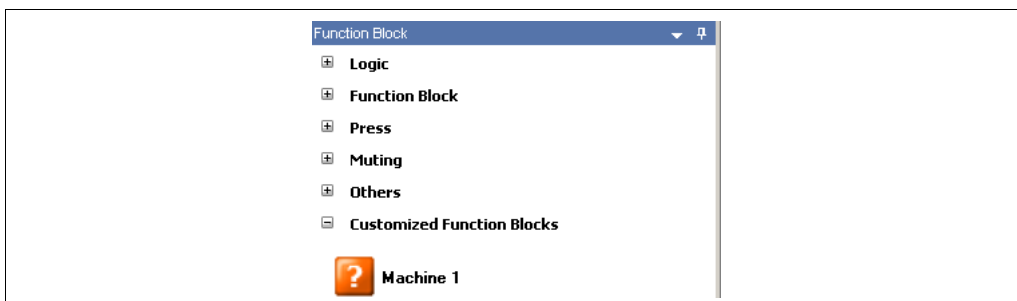


Fig. 224: New customized function block in the function block list

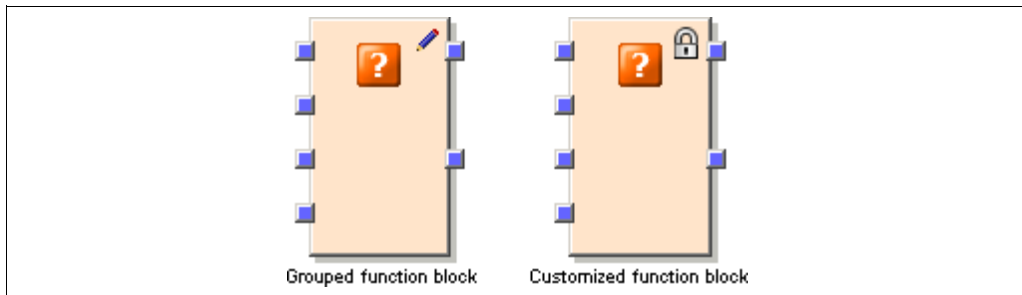
Once a customized function block has been placed on the worksheet, its content is displayed on a new page. In the example, the name of the customized function block is *Machine 1*. The worksheet for the customized function block is displayed orange. The customized function block can not be edited.

NOTE

Logic programming – Function blocks

A grouped function block contains a small pencil icon in the upper right corner indicating that it can be edited. The customized function block contains a padlock indicating that it is locked against modification.

Fig. 225: Icons for the grouped function block and for the customized function block



How to edit a customized function block:

- ➔ Open the customized function block's page by clicking on its tab,
- ➔ Click on **Edit...** in the toolbar. You will be prompted for confirmation. If you click on **Yes**, the customized function block will be transformed to a grouped function block which can be edited (see section 8.12.1 "Grouped function block" on page 180).
- ➔ In order to make the modified function block available for re-use in the function block list, save it again as a customized function block by clicking **Save as CFB...** in the toolbar.

How to transfer customized function blocks to another computer:

- ➔ Drag and drop the desired customized function blocks into the logic editor and save the project file.
 - ➔ Open the project file on another computer. You will be prompted to accept an import of all customized function blocks used in the project file.
 - ➔ Click on **Yes** to import the customized function blocks. They will be listed in the function block list and will be available in all new projects on the same PC.
- Or:**
- ➔ Click on **No** to import the customized function blocks as grouped function blocks only. In this case, they will not be listed in the function block list and will be available for the current project only.

How to delete a customized function block permanently from your PC:

- ➔ Delete all instances of the customized function block from your project or transform each of them to a grouped function block by clicking **Edit...** in the toolbar.
- ➔ In the function block list, right click on the customized function block you want to delete. The context menu opens.
- ➔ Choose **Delete Custom Function Block....**

NOTES

- You cannot undo this action.
- Other projects containing customized function blocks that have been deleted can still be used. When opening an older project that contains customized function blocks that have been deleted from your PC, it will be treated like a project that has been transferred from another PC. You will be prompted whether you want to import the customized function blocks contained in the project permanently as customized function blocks or as grouped function blocks for use in the current project only.

8.13 Simulation of the configuration

Within the logic editor, it is possible to simulate the programmed logic offline. Inputs can be set to High or Low and the resulting switching of the outputs can be monitored. Additionally the timer and counter values of the used function blocks are displayed on the function blocks while the simulation is running.

Logic programming – Function blocks

- ➔ Click on the **Start simulation mode** icon (🏠) in the toolbar to activate the simulation mode. The background of the logic editor will change to green and the simulation toolbar will appear.

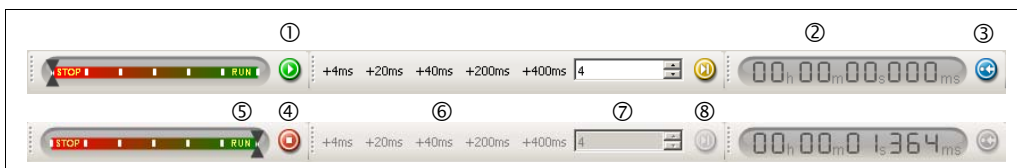


Fig. 226: Simulation toolbar

To start a simulation of the logic, click the green **Play** button (1) for simulation at full speed (near to real time). The timer (2) keeps track of the elapsed time. The timer can be reset using the blue **Reset** button (3). To stop a simulation, click the red **Stop** button (4).

Time control of the simulation

For logic processes which are too fast to see at normal speed there are two possibilities:

- Use the sliding bar (5) to slow the simulated passage of time.
- It is possible to execute a simulation in time increments. To do so, stop the simulation by clicking on the red **Stop** button and click on one of the time increment buttons to the right of the sliding bar (6). The following time periods are available by default: +4 ms, +20 ms, +40 ms, +200 ms and +400 ms. These values will be adapted automatically respective to the size of the programmed logic since they represent multiples of the logic execution time. By clicking on one of these time buttons, the simulation jumps forward by the specified time increment.

Additionally, the input field on the right (7) allows you to enter a user specific time period in ms, by which the simulation will jump forward when the yellow button (8) beside the input field is pressed. By entering a large number such as 40000 (40 s) into this field, you can jump forward in order to avoid waiting for timers to complete their cycle, for example.

The entered time will be rounded to the nearest possible cycle time.

NOTE

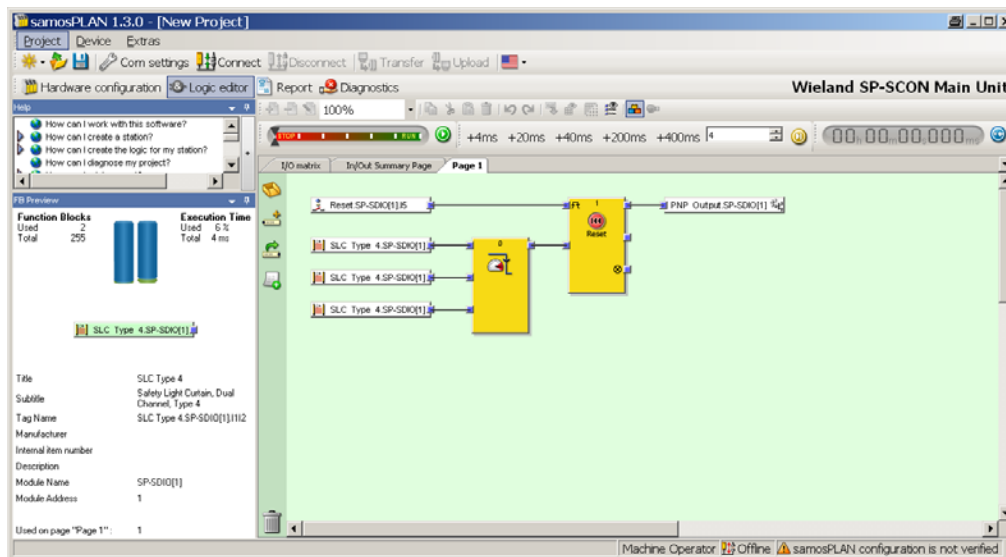
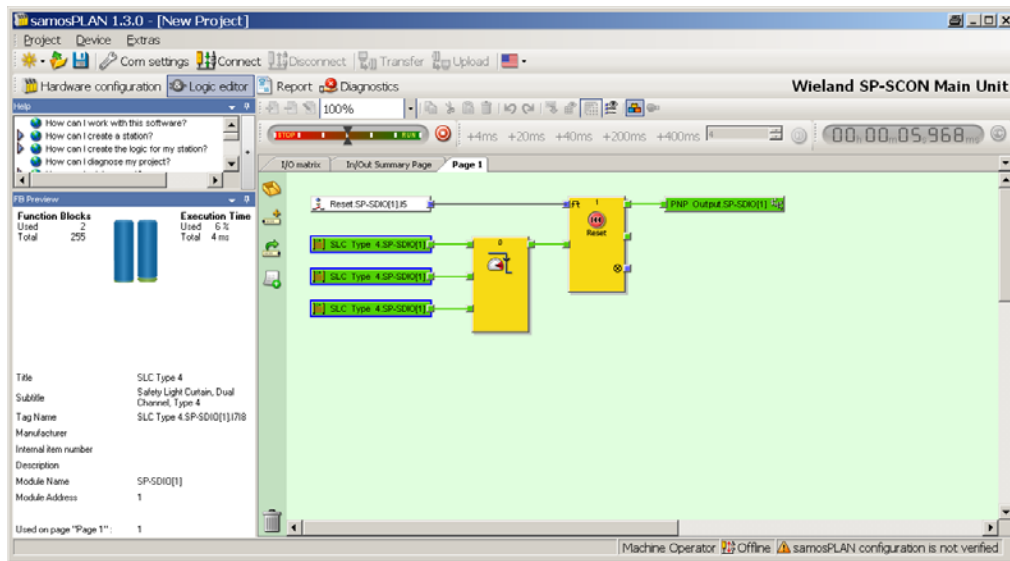


Fig. 227: Simulation mode started, simulation off

While the simulation is running you can set an input to High by clicking on it. High inputs will be displayed green with a blue frame. Another click will set the input back to Low again.

Logic programming – Function blocks

Fig. 228: Simulation mode started, simulation running



When the simulation is stopped, it is possible to select inputs to switch at the next possible moment. When clicking on an input in stopped mode, a blue box will appear around it denoting that it is ready to switch at the next cycle of the simulation. This makes it possible to switch one or more elements at the same time and see their direct effect on the logic.

After activation of the desired inputs, the simulation must be continued either by pressing the green **Start** button or by using one of the time increment buttons in order for the logic and outputs to switch accordingly.

NOTE

If the EDM or the Valve monitoring function block is being used, it is recommended to remove them from the logic before simulating. These function blocks expect a high signal on their feedback input within 300 ms of their associated output being activated. This can only be simulated by using small increments and not in real time.

8.14 Force mode

In force mode you can set the inputs in the samos[®]PRO logic program to High or Low via software independently of the values of the actual physical inputs while the samos[®]PRO system is in the Run state. The samos[®]PRO system and the programmed logic will react exactly in the same way as if the physical inputs had actually the respective values.

This enables you e.g. during commissioning or maintenance to test the wiring of your system and the function of your programmed logic in online operation.

NOTES

- You can force only the logic inputs of a samos[®]PRO system directly, but not outputs or logic results such as function blocks or jump addresses.
- Forcing effects only function block inputs. Therefore it is not possible to influence signals that do not depend on the outputs of a function block, such as inputs of I/O modules that are routed directly to a PLC via a gateway.



Exclude any danger for persons or equipment!

In force mode you can freely influence the status of the safety inputs. As a result thereof, the safety function of your safety equipment can be impaired and a dangerous state may occur.


- Ensure that no person is present in the dangerous area of your machine or system before activating force mode.
- Ensure that no person can intrude into the dangerous area of the machine or system while force mode is active.
- Additional safety measures may be required if forcing is used.


Do not use force mode from several PCs simultaneously!

- When using force mode, ensure that no person activates the force mode from a second PC. Otherwise, a dangerous state may occur.

How to activate the force mode:

The following requirements must be met in order to use the force mode:

- The firmware version of your samos[®]PRO CPUx must be version V.1.10 or higher.
- You must be logged in to the system as *Authorized client*.
- Your PC must be connected to the samos[®]PRO system via the COM interface. Forcing via an Ethernet connection is not possible.
- The configuration of your samos[®]PRO project may not be verified (CV LED  **Yellow** flashing with 1 Hz).

If you try to activate the force mode although the configuration has been already verified (CV LED of controller module  **Yellow** on), a dialog appears that allows you to reset the status to *Not validated*.

NOTE

- ➔ Click on **Connect** to establish a connection to your samos[®]PRO system.
- ➔ In the **Hardware configuration** view, click on the **Run application** button. If you have not yet logged in as *Authorized client*, you will be prompted now to do so.
- ➔ Go to the **Logic editor** view and click on the **Start force mode** button. A dialog is opened where you can enter the time after which the force mode will be automatically left if no actions are taken.

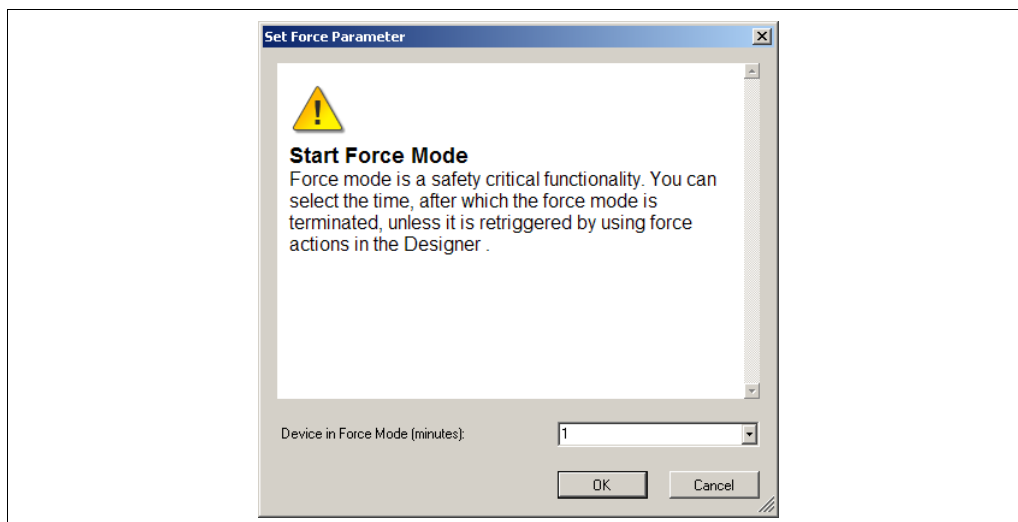
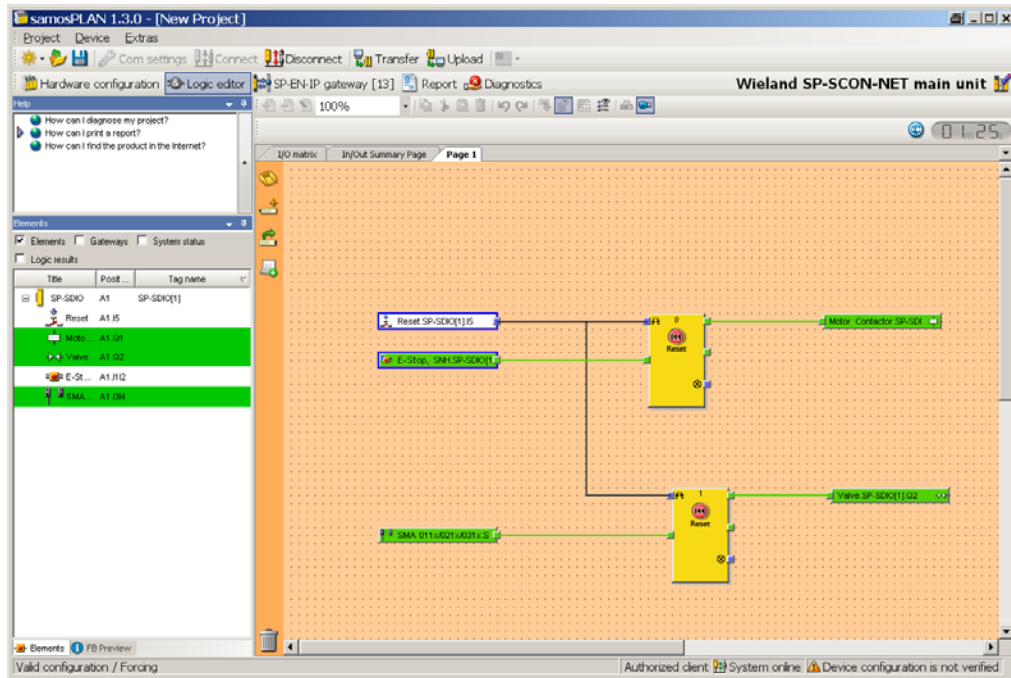


Fig. 229: Dialog window when starting force mode

- ➔ Choose the desired time span from the drop box list and click **OK**. Force mode will be activated and the background colour of the logic editor changes to red.

Logic programming – Function blocks

Fig. 230: Logic editor with force mode activated



NOTE

While force mode is active, it is not possible to logout, to upload and compare a configuration or to stop the device

How to force an input:

- ➔ Click on an input with the left mouse button. A context menu with the following options will appear:

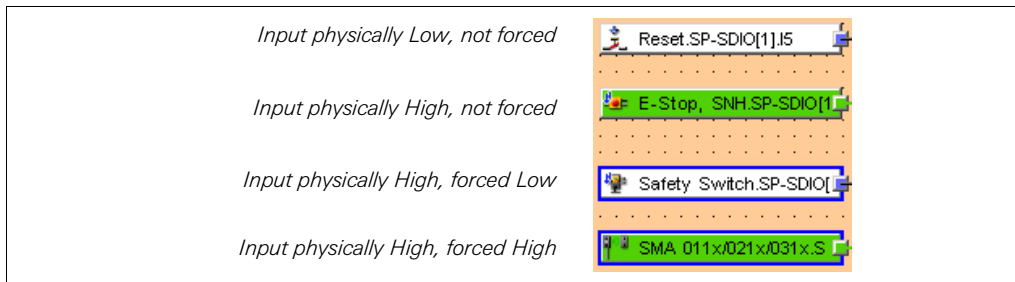
Force low...: The samos®PRO system will evaluate the input independently of its actual physical value as Low.

Force high...: The samos®PRO system will evaluate the input independently of its actual physical value as High.

Remove force...: The samos®PRO system will evaluate the input with its actual physical value.

Forced inputs are marked with a dark blue frame. High inputs are displayed green, Low inputs are displayed white. Inputs whose forced value differs from their physical value are displayed light blue.

Fig. 231: Forced and not forced inputs



NOTES

- While an input is forced in the logic, the actual value on the physical input is not displayed in the logic editor but only in the **Hardware configuration** view. Forcing affects only the inputs in the logic program, but not the physical inputs of extension modules. Examples:

Forcing has no effect on the inputs of an SP-SDIO module that are used for Fast shut off. Therefore the hardware output may remain Low although the inputs are forced High in the logic because the Fast shut off on the SP-SDIO is controlled directly via the physical inputs.

Forcing has no effect on inputs whose values are not being controlled by the logic program but are transferred directly to a PLC via a gateway.

- Force mode always applies to the complete project. This means for logic programs using more than one page in the logic editor, that a forced input will be set to the same value on each page of the logic editor where it is used, not only on the currently displayed page.
- If forcing an input in a logic program causes more than 16 outputs to switch at the same time, then some of these outputs will be switched with a delay of one or more logic cycles due to the limited transmission capacity of the RS-232 interface. The logic execution time depends on the size of your logic program. It is calculated automatically in the logic editor and is displayed in the top right corner of the **FB preview** window.
- Unlike the simulation mode, the force mode allows you to use the **EDM** or **Valve monitoring**, if corresponding devices are connected that will send the required feedback signal when the outputs are activated.
- When using a samos[®]PRO gateway, please note that the process image of the samos[®]PRO gateways *always* reflects the actual physical value of the inputs and outputs of the connected devices and not the (virtual) forced value of an input in the logic program. If by forcing of an *input* in the logic program (e.g. from High to Low) the value of an *output* is changed (e.g. from High to Low), the actually changed value of the *output* (in the example Low) will be transferred to the PLC in the process image, but *not* the forced Low value of the input in the logic program. Instead still the actual physical value of the input on the device (in the example High) will be transferred. Take this into account when you evaluate the transferred data in the PLC.

Termination of the force mode

The force mode can be terminated in the following ways:

- manually through the user
- automatically after the defined time delay
- automatically after 30 seconds if the samos[®]PRO system detects an error (e.g. if the connection to the PC is interrupted)

When the force mode is terminated, the samos[®]PRO system switches all outputs to Low and the active application is stopped.

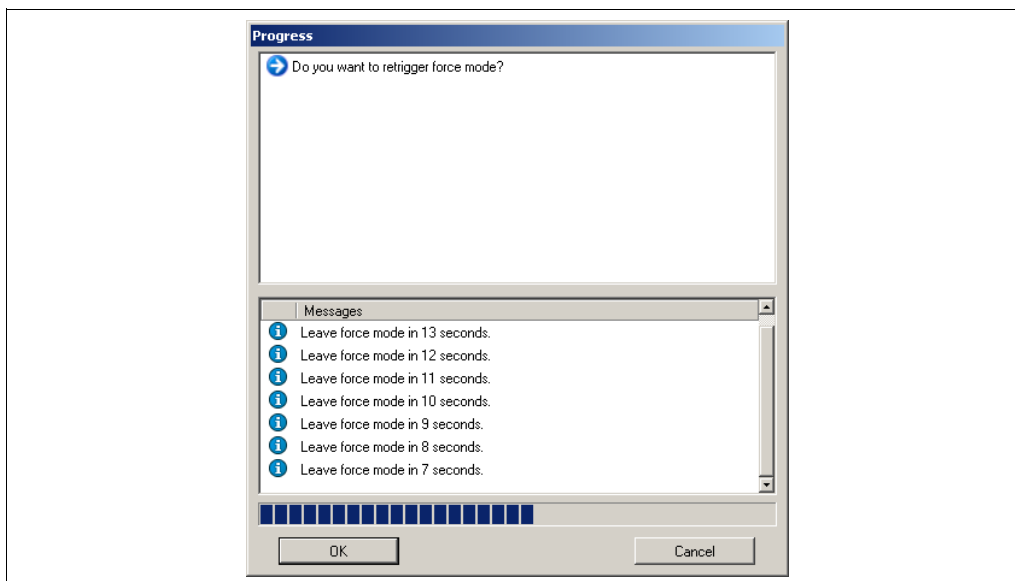
Ensure that no dangerous situation can occur when the force mode is terminated!

- Ensure that your machine or system is transferred to a safe state and can not be damaged when the force mode is terminated.
 - While the force mode has been active, the actual value of an input may have changed (e.g. button pressed, safety door opened etc.). Ensure that this can not cause any dangerous situation before restarting your machine or system.
- ➔ Click on the **Stop force mode** button. A safety message will appear. Click on **Yes** to confirm and terminate the force mode or click on **No** to keep the force mode active.
- ➔ If no force action (e.g. forcing of an input) has occurred for the time defined when the force mode was started, the force mode will be automatically terminated. During force mode, a timer in the top right corner shows the time remaining until the force mode is automatically terminated. Each action resets this timer. You can also reset the timer using the **Trigger force mode** button on its left. 15 seconds before the timer expires, a dialog is displayed that reminds you of the imminent termination of the force mode.



Logic programming – Function blocks

Fig. 232: Dialog before auto-matic termination of the force mode



- ➔ If you ignore this dialog, the force mode will be terminated after the defined time delay has expired.
Or:
- ➔ Click on **Cancel**. The dialog will close and the force mode will be terminated after the defined time delay has expired.
Or:
- ➔ Click **OK** in order to close the dialog, reset the timer and keep the force mode active.

9 I/O modules

9.1 Dual channel evaluation and discrepancy time

The I/O extension modules, e.g. SP-SDIO or SP-SDI, can carry out a dual-channel evaluation when predefined input elements from the Elements window (e.g. SMA, SLC,...) are connected to them. If such an input element is selected, you do not need a separate function block for dual-channel evaluation (e.g. light curtain monitoring, safety gate monitoring or magnetic switch).

The dual-channel evaluation evaluates the correct sequence of the two input signals. It is expected that if one of the two signals has caused a switching off, the other signal will follow accordingly. Which values the two signals must have depends on the type of the dual-channel evaluation. There are two possibilities:

- Equivalent evaluation
- Complementary evaluation

An optional **Discrepancy time** can be configured. The discrepancy time defines for how long the two inputs may have discrepant values after one of the both input signals has changed without this being considered as an error.

When configuring the discrepancy time the following must be observed: The discrepancy time ...

NOTE

- must be a multiple of 4 ms,
- If signals from tested sensors are connected to SP-SDI or SP-SDIO modules the discrepancy time should be at least the set *Test gap [ms]* plus the *Max. off-on delay [ms]*, because a signal change at the module input can be delayed for this time. Both values are displayed in the samos®PLAN report for the used test output.

The following truth table describes the discrepancy conditions for the dual-channel equivalent and the dual-channel complementary input evaluation:

Evaluation type	Input A	Input B	Discrepancy timer ¹⁾	Status of the dual-channel evaluation	I/O module input in the logic editor	Discrepancy error
Equivalent	0	0	0	Inactive	0	Unchanged ²⁾
	0	1	< Discrepancy time	Discrepant	0	Unchanged
	1	0	< Discrepancy time	Discrepant	0	Unchanged
	1	1	0	Active ³⁾	1	0
	x	x	Discrepancy time (timeout)	Error	0	1
Complementary	0	1	0	Inactive	0	Unchanged
	0	0	< Discrepancy time	Discrepant	0	Unchanged
	1	1	< Discrepancy time	Discrepant	0	0
	1	0	0	Active ³⁾	1	Unchanged
	x	x	Discrepancy time (timeout)	Error	0	1

For the change between the different states of the dual-channel evaluation the following rules apply:

Tab. 103: Dual-channel evaluation

¹⁾ If the discrepancy time is active (>0), the discrepancy timer is restarted on the first signal change resulting in a discrepant status. If the discrepancy time is inactive (=0), the discrepancy timer does not start, i.e. a timeout will never occur.

²⁾ Unchanged = Last status is preserved.

³⁾ If the correct sequence has been observed.



A dual-channel evaluation can only change to Active (the input of the I/O module in the logic editor changes from Low to High), if ...

- since the last Active status at least once the status was Inactive, i.e. it is not possible to switch from Active to Discrepant and back to Active, and
- the discrepancy time has not elapsed or the discrepancy time is deactivated

NOTE

If the correct sequence to reach the Active status has not been observed (i.e. if the status has changed from Active to Discrepant to Active), SP-SDIO and SP-SDI modules with firmware V2.00.0 and higher will display this sequence error after 100 ms at the latest, if the discrepancy time has not elapsed earlier (i.e. if the discrepancy time is set to 0 or to a value > 100 ms). Older modules will not display the sequence error, although their input in the logic editor remains Low as well.

In case of a discrepancy error or sequence error the module will behave as follows:

- The MS LED of the affected module will start flashing  **Red** (1 Hz),
- the LEDs of the affected inputs will start flashing  **Green** (1 Hz),
- the module's **Status input data** in the logic editor will be Low,

Resetting the error:

A discrepancy error (timeout) or sequence error is reset, if the Inactive status has been reached.

10 Transferring the System Configuration

Initially, the configuration of the safety controller only exists as a project, meaning as a samos®PRO configuration file. The configuration has to be transferred to the samos®PRO removable memory plug via the CPU.

The samos®PRO removable memory plug and the controller module communicate via an internal interface. Direct connection of a PC to the removable memory plug is not possible. Data can only be loaded to the removable memory plug or read from it via a compatible controller module.

NOTE

The configuration data are checked for compatibility during transfer to the removable memory plug and can subsequently be verified (through reading and comparing) and optionally have a write protection assigned to it.

With the removable memory plug the project data can be transferred without further processing using the samos®PLAN to any number of samos®PRO safety controllers. The configuration data are copied exactly in the process, including the verification and any write-protection information that were set during the configuration of the first safety controller with these data.

10.1 Transferring project data to the safety controller

After the transfer, the configuration data are read back from the removable memory plug if the verification has been activated in the samos®PLAN (see chapter 10.3 "Verification of the configuration" on page 195).

The reading back of the configuration data from the removable memory plug requires some time. The removable memory plug may not be removed during this time. The samos®PLAN displays a corresponding warning as long as the process takes.

NOTE

10.2 Compatibility check

The configuration data contain an electronic type code and a version code for each module that is to be configured. During the transfer each module checks whether it is compatible with the configuration data. The compatibility check only applies to the functional part of the respective module, not to the hardware variant, the implementation of the terminals, for example, remains unconsidered.

If the compatibility check is negative, a corresponding error message is generated in the respective module and in the controller module.

In samos®PLAN different version numbers are assigned to some modules so that a compatible module can be selected from a list below the module.

NOTE

10.3 Verification of the configuration

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

If the configuration is verified, the samos®PRO system will change into the Run state automatically after the voltage supply has been switched on. If the configuration is not

NOTE

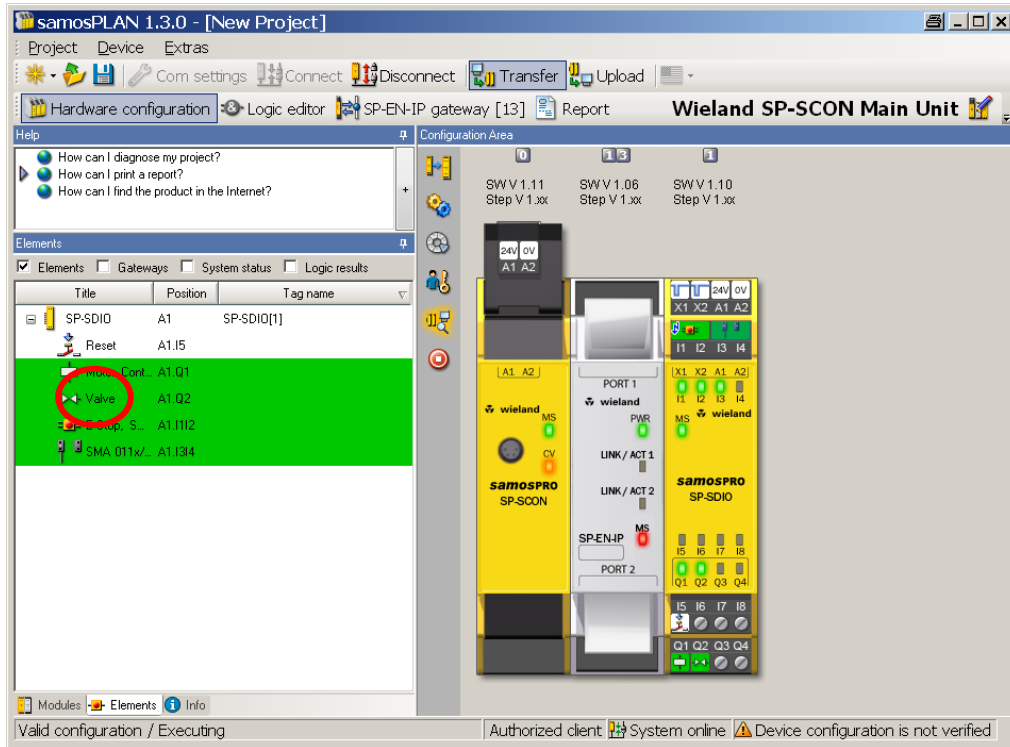
Transferring the System Configuration

verified, the system must be set into the Run state manually using the samos®PLAN (see section 11.1 “Changing the device state” on page 200).

How to verify the configuration:

- ➔ Click the **Upload and verify configuration** button.

Fig. 233: Upload and verify configuration



- ➔ The **Upload and verify result** window is opened. Click **Yes** below at the question **Set device to verified?** if the displayed configuration is the expected configuration. The system is then considered to be verified.

Transferring the System Configuration

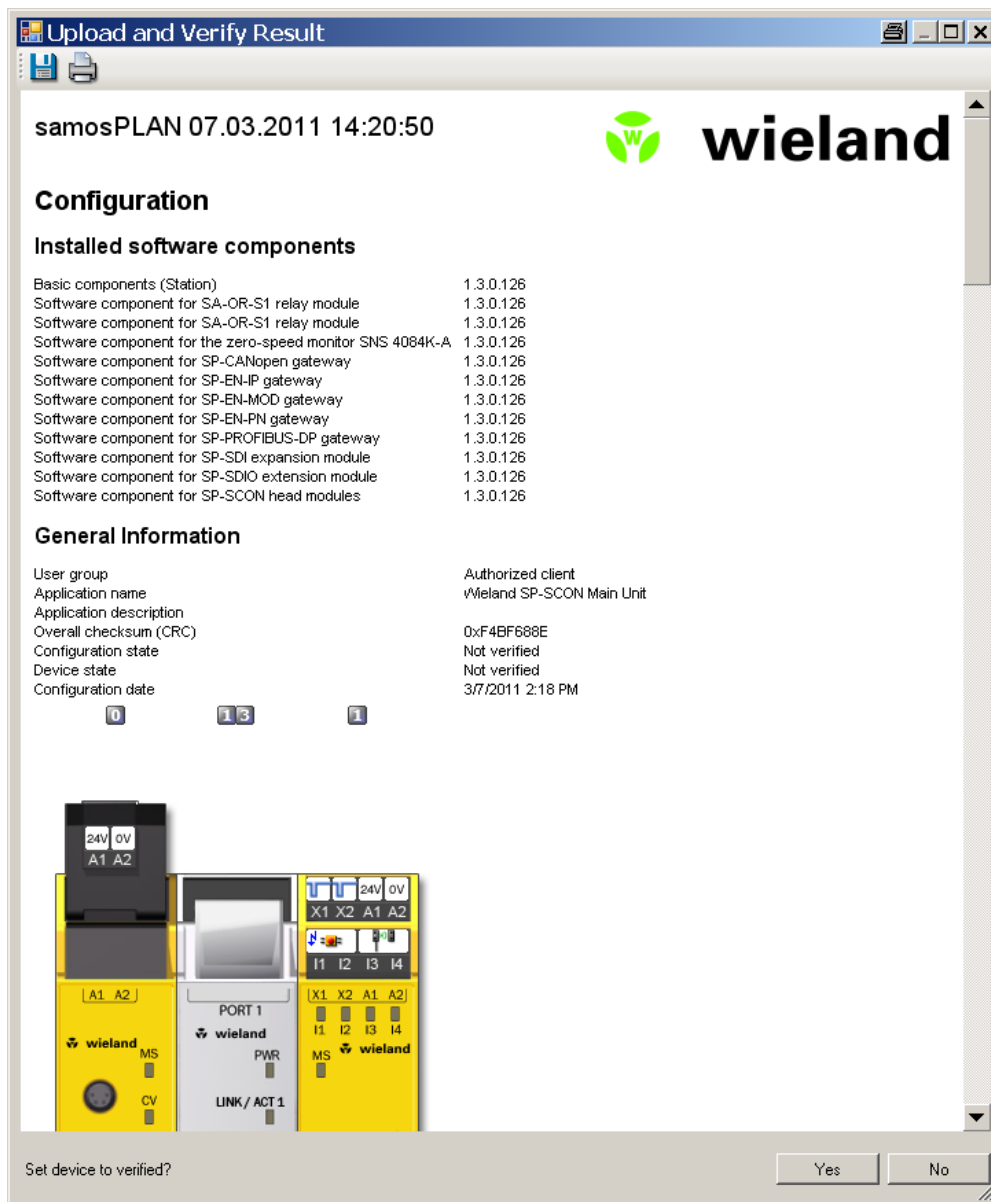


Fig. 234: Setting a device to verified.

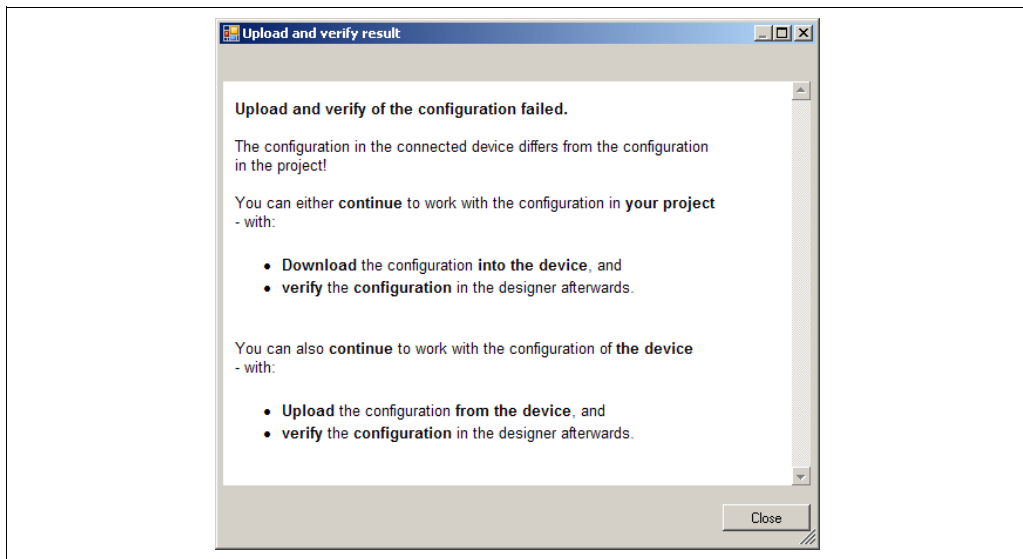
The configuration of the connected elements, for example EFI sensors, is not included in the process. Their verification is carried out analogous to the configuration and verification via the serial interfaces of the devices.

NOTE

- ➔ If differences between the project data and the read-back configuration data are detected, a corresponding message including information about possible actions is displayed. Verification of the configuration is not possible then. Observe the information in the error message for the further procedure. Terminate the dialog box by clicking **Close**.

Transferring the System Configuration

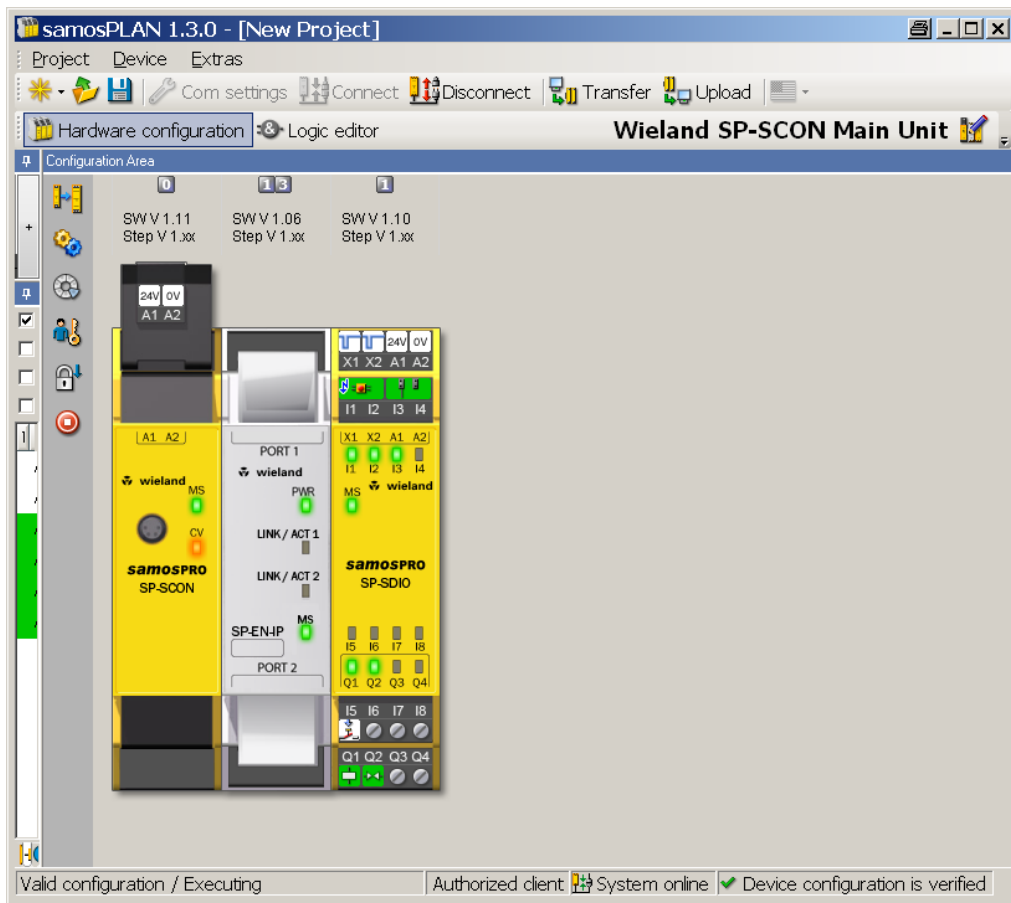
Fig. 235: Upload and verify failed



If the verification is completed successfully, a project report is created subsequently that you can print and save.

The query whether the device is to be marked as verified is displayed in the lower part of the report window. You must be logged in as an Authorized client to mark the device as “verified”. The status verified/not verified is indicated in the lower right-hand corner of the samos®PLAN and by the CV LED at the samos®PRO controller module lighting up.

Fig. 236: Verification successful



The verification flag is included in copying when the data are read back into the removable memory plug and are also transferred automatically to each safety controller to which the configuration data are duplicated.

The safety controller is also ready to use when the configuration is only validated and not verified or does not have a write protection. However the automatic transition of the

samos[®]PRO system into the Run state after the voltage supply has been switched on is not possible then.

The dialog box for verification is only displayed after a request by the user in order to ensure that the verification process does not have to be passed through every time the configuration is changed or new project data uploaded.

NOTE

In order to validate the samos[®]PRO system, the safety functions at the machine or system have to be checked completely and have to function perfectly. With regard to its content, the validation is identical to the the technical test taking place when the samos[®]PRO system is commissioned.

10.4 Activating the write protection of the configuration in the control system

A verified configuration can be protected against accidental changes by activating the write protection. The write protection can be set and deactivated in the samos[®]PLAN by using the lock symbol in the hardware configuration to the left of the controller module.

The write protection is included in copying when the data are transferred to the removable memory plug and are also transferred automatically to each safety controller to which the configuration data are duplicated.

11 Device States of the samos®PRO System

The samos®PRO system knows different device states during operation. Some device states require a user intervention, e.g. the state transition from **Stop** to **Run** or vice versa using the samos®PLAN. Other states are based on the internal self-test of the samos®PRO system, e.g. **Internal error**. The following table summarises the device states of the samos®PRO system.

Tab. 104: Device status and LED displays on the CPUx

MS LED	Meaning	Notes
○	Supply voltage is outside range	Switch on the supply voltage and check it at the terminals A1 and A2.
☀ Red/Green (1 Hz)	A self test is being carried out or the system is being initialized	Please wait ...
☀ Green (1 Hz)	System is in Stop state	Start the application in the samos®PLAN.
☀ Green (2 Hz)	Identify (e.g. for samos®NET)	
● Green	System is in Run state	
☀ Red (1 Hz)	Invalid configuration	Check the module type and version of the controller module and extension modules whose MS LED flashes red/green. If appropriate, adapt the configuration using the samos®PLAN. For detailed diagnostics information refer to the samos®PLAN.
☀ Red (2 Hz)	Critical error in the system, possibly in this module. Application is stopped. All outputs are switched off.	Switch the supply voltage off and on again. If the error is not eliminated after multiple repetition, replace this module. For detailed diagnostics information refer to the samos®PLAN.
● Red	Critical error in the system, possibly in another module. Application is stopped. All outputs are switched off.	Switch the supply voltage off and on again. If the error is not eliminated after multiple repetition, replace the module which displays ☀ Red (2 Hz). If this is not the case, use the diagnostic functions of the samos®PLAN to narrow down the respective module.
CV LED	Meaning	
○	Configuration in progress	
☀ Yellow (2 Hz)	Storing of configuration data in the non-volatile memory	Supply voltage may not be interrupted until the storage process has been completed.
☀ Yellow (1 Hz)	Unverified configuration	Verify configuration with the samos®PLAN software.
● Yellow	Verified configuration	



11.1 Changing the device state

Specific state changes in the samos®PRO system are carried out manually in the samos®PLAN. These changes in the device state are:

- Change from **Stop** to **Run**
- Change from **Run** to **Stop**

In order to change the device state, click on the **Stop application** or **Run application** button next to the representation of the modules in the hardware configuration.

Device States of the samos[®]PRO System

Icon	Function	Description
	Run	Sets the system into the Run state.
	Stop	Sets the system into the Stop state.

Tab. 105: Run button and Stop button

If the configuration is verified, the samos[®]PRO system will go into the Run state automatically after the voltage supply has been switched on. If the configuration is not verified, the system must be set into the Run state manually using the samos[®]PLAN.

NOTE

11.2 Behaviour on startup

When the samos[®]PRO safety controller changes from the Stop state to the Run state:

- The **First logic cycle** status bit of the controller module is High for the logic execution time. This status bit is available as a CPU input element in the logic editor.
- All timers and states including the error states of the function blocks are reset.

11.3 Software reset of the controller module

It is possible to reset the controller module via software (i.e. without interrupting the voltage supply), if the samos[®]PLAN is connected with the controller module.

How to perform a software reset:

- In the Hardware configuration view, right click on the controller module in the configuration area and select the **Software reset** command from the context menu.
- If you are not logged in as *Authorized client*, you will be prompted to log in now.
- A safety message will appear. Click on **Yes** to reset the controller module.

Before you reset the controller module, check whether the system is in a safe state!

If you reset the controller module, the outputs (e.g. the **Enable** output) could change their status. The **Software reset** command should only be used when the hazardous area has been checked visually and nobody is in the hazardous area or has access to the hazardous area while the controller module is being reset.



If the configuration is verified, the controller module will change back into the Run state automatically after the reset. If the configuration is not verified, you will have to start the controller module manually using the samos[®]PLAN.

NOTE

12 Technical Commissioning

The configuration of the samos[®]PRO system has to be completed before you begin with the technical commissioning.

12.1 Wiring and voltage supply



When connecting the samos[®]PRO system, observe the technical data in the samos[®]PRO hardware operating instructions!

- ➔ Connect the individual field devices to the corresponding signal connections and check for each safety input, test/signal output and safety output whether these behave as required for the application. Diagnostics information from the samos[®]PRO LEDs support you in validating the individual field signals. Check whether the external circuit, the realisation of the wiring, the choice of the pick-ups and their location at the machine fulfil the required safety level.
- ➔ Eliminate any faults (e.g. incorrect wiring or crossed signals) at each safety input, test/signal output or safety output before you continue with the next step.
- ➔ Switch on the voltage supply. As soon as the supply voltage is applied to the connections A1 and A2 of the controller module SP-SCON/SP-SCON-NET or the SP-SDIO modules, the samos[®]PRO system automatically carries out the following steps:
 - internal self-test
 - loading of the saved configuration
 - testing of the loaded configuration for validity

The system does not start up if the steps described above could not be carried out successfully. If there is an error, this is indicated correspondingly by the LEDs (see the samos[®]PRO hardware operating instructions) and the samos[®]PRO system only transfers Low values.

12.2 Transferring the configuration

After you have configured the hardware and the logic in the samos[®]PRO system and have checked whether they are correct, transfer the configuration to the samos[®]PRO system via the samos[®]PLAN.

12.3 Technical test and commissioning

The machine or system that is protected by a samos[®]PRO safety controller may only be started up after a successful technical check of all safety functions. The technical test may only be performed by qualified safety personnel.

The technical test includes the following test items:

- ➔ Uniquely mark all connection cables and connectors at the samos[®]PRO system in order to avoid confusion. Since the samos[®]PRO system has several connections of the same design, you must ensure that loosened connection cables are not connected back to the wrong connection.
- ➔ Verify the configuration of the samos[®]PRO system.
- ➔ Check the signal paths and the correct inclusion in higher-level controllers.
- ➔ Check the correct data transfer from and to the samos[®]PRO safety controller.
- ➔ Check the logic programme of the safety controller.
- ➔ Completely document the configuration of the entire system, of the individual devices and the results of the safety check.

- Check the safety functions of the machine or system completely and ensure that the safety functions function perfectly.
- In order to prevent unintentional overwriting of the configuration, activate the write protection of the configuration parameters of the samos®PRO system. Modifications are now no longer possible unless the write protection has been deactivated.

13 Troubleshooting

In case of an error please refer to the “samos®PRO Hardware” operating instructions, chapter “Error displays of the status LEDs”. There you will find a list of LED error displays, error codes, error causes and rectification measures.

Error codes and error messages can also be displayed in the **Diagnostics** view if you are connected to the samos®PRO system. For more information on how to perform diagnostics, see section 5.8 “Diagnostics view” on page 39.

14 Annex

14.1 Example application report: Wood processing machine

samosPLAN

Wieland Electric

Application name: Wieland SP-SCON Main Unit

CRC Tool: 0x74F99157 - CRC Device: 0x00000000

Content

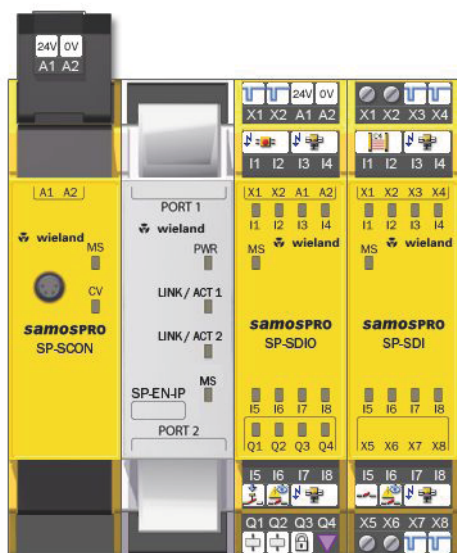
- 1. Bill Of Material.....3
- 2. Diagnostics.....4
- 3. Configuration.....4
 - 3.1. Installed software components.....4
 - 3.2. General Information.....4
 - 3.3. CPU.....5
 - 3.3.1. SP-SCON - General Information.....5
 - 3.3.2. CPU logic.....5
 - 3.3.2.1. E-Stop.....5
 - 3.3.2.2. Doors.....6
 - 3.3.2.3. Guardlocking.....6
 - 3.3.3. CPU0 - IO.....6
 - 3.3.3.1. Wiring diagram.....7
 - 3.3.4. CPU0 - Routing.....7
 - 3.3.4.1. CPU to RS232.....7
 - 3.3.4.2. RS232 to CPU.....7
 - 3.4. I/O module.....8
 - 3.4.1. SP-SDIO[1].....8
 - 3.4.1.1. General information.....8
 - 3.4.1.2. Inputs.....8
 - 3.4.1.3. Outputs.....8
 - 3.4.1.4. Power supply.....8
 - 3.4.1.5. Wiring diagram.....9
 - 3.4.1.6. Test pulse parameter.....9
 - 3.4.2. SP-SDI[2].....9
 - 3.4.2.1. General information.....9
 - 3.4.2.2. Inputs.....9
 - 3.4.2.3. Wiring diagram.....10
 - 3.4.2.4. Test pulse parameter.....10
 - 3.5. Gateway.....10
 - 3.5.1. SP-EN-IP - General Information.....10
 - 3.5.2. SP-EN-IP - Routing.....10
 - 3.5.2.1. CPU to Network.....11
 - 3.5.2.2. Network to CPU.....11
 - 3.5.3. SP-EN-IP - Device configuration.....11

samosPLAN 3/10/2011 3:28:06 PM



1. Bill Of Material

0 1 3 1 2



QTY	Title	Tag Name	Part Number	Internal Item Number	Description
1	SP-SCON	SP-SCON[0]	R1.190.0010.0		Wieland SP-SCON Module
1	SP-MEMORY	System plug	R1.190.0080.0		samosPRO Memory Plug
1	SP-EN-IP	SP-EN-IP[13]	R1.190.0150.0		Wieland SP-EN-IP module
1	SP-SDIO	SP-SDIO[1]	R1.190.0030.0		Wieland SP-SDIO input/output expansion module
1	Reset / Single Channel	Reset.SP-SDIO[1]J5			
1	EDM / Single Channel	EDM.SP-SDIO[1]J6			
1	Motor Contactor / Single Channel	GE 3S1OEMCR S.1.SP-SDIO[1].Q1			
1	Motor Contactor / Single Channel	GE 3S1OEMCR S.6.SP-SDIO[1].Q2			
1	Interlock / Dual Channel	Interlock.SP-SDIO[1].Q3/I3I4			
1	E-Stop, SNH / Dual Channel	E-Stop, SNH.SP-SDIO[1]J1I2			
1	Safety Switch / Dual Channel	barrier-safety switches.SP-SDIO[1]J7I8			
1	SP-SDI	SP-SDI[2]	R1.190.0050.0		Wieland SP-SDI input expansion module
1	Single Channel NO / Single Channel	standstill.SP-SDI[2]J5			

samosPLAN 3/10/2011 3:28:06 PM



QTY	Title	Tag Name	Part Number	Internal Item Number	Description
1	EDM / Single Channel	contactor S.6.SP-SDI[2].I6			
1	SLC Type 4 / Safety Light Curtain, Dual Channel, Type 4	SLC Type 4.SP-SDI [2].I112			
1	Safety Switch / Dual Channel	door middle.SP-SDI [2].I314			
1	Safety Switch / Dual Channel	door right.SP-SDI[2].I718			

2. Diagnostics

No error history available.

3. Configuration

3.1. Installed software components

Basic components (Station)	1.3.0.127
Software component for SA-OR-S1 relay module	1.3.0.127
Software component for SA-OR-S2 relay module	1.3.0.127
Software component for the zero-speed monitor SNS 4084K-A	1.3.0.127
Software component for SP-CANopen gateway	1.3.0.127
Software component for SP-EN-IP gateway	1.3.0.127
Software component for SP-EN-MOD gateway	1.3.0.127
Software component for SP-EN-PN gateway	1.3.0.127
Software component for SP-PROFIBUS-DP gateway	1.3.0.127
Software component for SP-SDI expansion module	1.3.0.127
Software component for SP-SDIO extension module	1.3.0.127
Software component for SP-SCON head modules	1.3.0.127

3.2. General Information

User group	Machine Operator
Application name	Wieland SP-SCON Main Unit
Application description	
Overall checksum (CRC)	-
Configuration state	Not verified
Configuration date	-

Page 4

samosPLAN 3/10/2011 3:28:06 PM

wieland



Module	Type code	Step	Address
SP-SCON	SP-SCON	V 1.xx	0
SP-EN-IP	SP-EN-IP	V 1.xx	13
SP-SDIO	SP-SDIO84-P1-K-x	V 1.xx	1
SP-SDI	SP-SDI8-P1-K-x	V 1.xx	2

3.3. CPU

3.3.1. SP-SCON - General Information

Type code	Serial number	SW version	Hardware version	Version/Step	Memory usage (UI/Logic)	Address
SP-SCON	0000 0000	V 1.10.0	0.00	1.3.0.127 V 1.xx	5.66% / 7.29%	0
SP-MEMORY	0701 0000	-	-	1.3.0.127	-	-

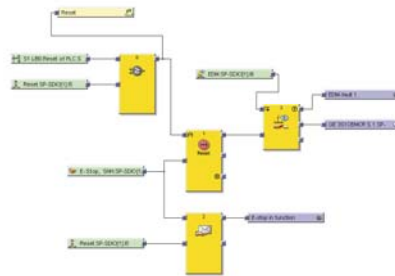
3.3.2. CPU logic

	Used
Function Blocks	9
Execution Time (ms)	4

3.3.2.1. E-Stop

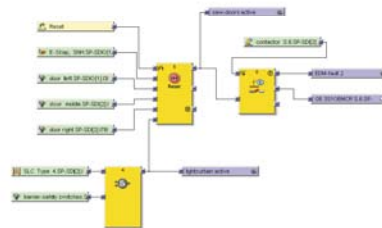
No	Name	Input	Output	Settings
0	OR	I.0 S1.LB0.Reset of PLC.SP-EN-IP [13].S1.LB0.0	O.0 POB 1.Input 0	
1	Reset	I.1 Reset.SP-SDIO[1].I5 I.0 POB 0.Output 0 I.1 E-Stop. SNH.SP-SDIO[1].I1I2	O.0 POB 5.Input 0 O.0 POB 3.Input 2 O.1 Not connected O.2 Not connected	Min. reset pulse time: 100 ms
2	Message generator	I.0 E-Stop. SNH.SP-SDIO[1].I1I2 I.1 Reset.SP-SDIO[1].I5	O.0 E-stop in function O.1 Not connected	Input 1: Edge = Negative ; Message Text = Not-Halt betätigt Input 2: Edge = Positive & Negative ; Message Text = Reset manuell betätigt Max. feedback delay: 300 ms
3	EDM	I.0 EDM.SP-SDIO[1].I6 I.2 POB 1.Output 0	O.0 EDM-fault 1 O.1 GE 3S1OEMCRS.1.SP-SDIO[1].Q1 O.2 Not connected	

Page 5



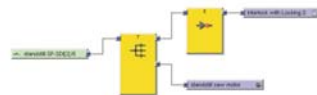
3.3.2.2. Doors

No	Name	Input	Output	Settings
4	AND	I.0 SLC Type 4 SP-SDI[2].I1I2 I.1 barrier-safety switches.SP-SDIO[1].I7I8	O.0 POB 5.Input 5 O.0 lightcurtain active	
5	Reset	I.0 POB 0.Output 0 I.1 E-Stop SNH SP-SDIO[1].I1I2 I.2 door left.SP-SDIO[1].I3I4 I.3 door middle.SP-SDI[2].I3I4 I.4 door right.SP-SDI[2].I7I8 I.5 POB 4.Output 0	O.0 POB 6.Input 2 O.0 saw-doors active O.1 Not connected O.2 Not connected	Min. reset pulse time: 100 ms
6	EDM	I.0 contactor S.6.SP-SDI[2].I6 I.2 POB 5.Output 0	O.0 EDM-fault 2 O.1 GE 3S1OEMCR S.6.SP-SDIO[1].Q2 O.2 Not connected	Max. feedback delay: 300 ms



3.3.2.3. Guardlocking

No	Name	Input	Output	Settings
7	Routing 1:N	I.0 standstill.SP-SDI[2].I5	O.0 POB 8.Input 0 O.1 standstill saw motor	
8	NOT	I.0 POB 7.Output 0	O.0 Interlock with Locking.SP-SDIO[1].Q3 [1].Q3	



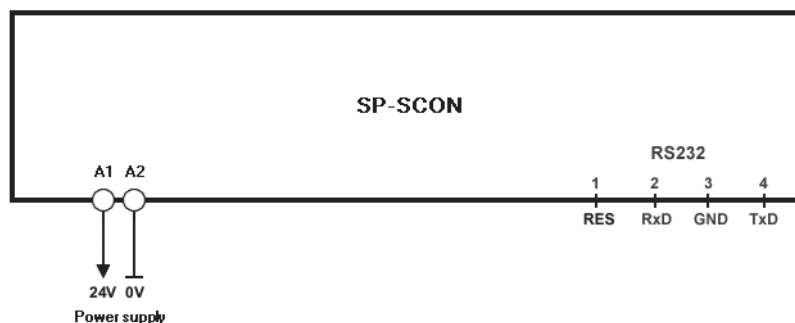
3.3.3. CPU0 - IO

samosPLAN 3/10/2011 3:28:06 PM

wieland

Tag name
 24V A1 SP-SCON[0] Power Supply
 0V A2

3.3.3.1. Wiring diagram



3.3.4. CPU0 - Routing

3.3.4.1. CPU to RS232

[0-99]

[Byte 0] Module 0 [SP-SDIO[1]]

[0.7]	-	[0.6]	barrier-safety switches.SP-SDIO [1].I718	[0.5]	EDM.SP-SDIO[1].I6	[0.4]	Reset.SP-SDIO[1].I5
[0.3]	-	[0.2]	door left.SP-SDIO [1].I314	[0.1]	-	[0.0]	E-Stop, SNH.SP-SDIO [1].I112

[Byte 1] Module 1 [SP-SDI[2]]

[1.7]	-	[1.6]	door right.SP-SDI [2].I718	[1.5]	contactor S.6.SP-SDI [2].I6	[1.4]	standstill.SP-SDI[2].I5
[1.3]	-	[1.2]	door middle.SP-SDI [2].I314	[1.1]	-	[1.0]	SLC Type 4.SP-SDI [2].I112

[Byte 12] Module 0 [SP-SDIO[1]]

[12.7]		[12.6]		[12.5]		[12.4]	
[12.3]		[12.2]	Interlock with Locking.SP-SDIO [1].Q3	[12.1]	GE 3S1OEMCR S.6.SP-SDIO[1].Q2	[12.0]	GE 3S1OEMCR S.1.SP-SDIO[1].Q1

[Byte 24] Byte 0

[24.7]	result 0.7	[24.6]	result 0.6	[24.5]	saw-doors active	[24.4]	lightcurtain active
[24.3]	standstill sawmotor	[24.2]	EDM-fault 2	[24.1]	EDM-fault 1	[24.0]	E-stop in function

3.3.4.2. RS232 to CPU

Page 7

[0-3]

[0] Byte 0

[0.7]	RS232 0.7	[0.6]	RS232 0.6	[0.5]	RS232 0.5	[0.4]	RS232 0.4
[0.3]	RS232 0.3	[0.2]	RS232 0.2	[0.1]	RS232 0.1	[0.0]	RS232 0.0

[1] Byte 1

[1.7]	RS232 1.7	[1.6]	RS232 1.6	[1.5]	RS232 1.5	[1.4]	RS232 1.4
[1.3]	RS232 1.3	[1.2]	RS232 1.2	[1.1]	RS232 1.1	[1.0]	RS232 1.0

[2] Byte 2

[2.7]	RS232 2.7	[2.6]	RS232 2.6	[2.5]	RS232 2.5	[2.4]	RS232 2.4
[2.3]	RS232 2.3	[2.2]	RS232 2.2	[2.1]	RS232 2.1	[2.0]	RS232 2.0

[3] Byte 3

[3.7]	RS232 3.7	[3.6]	RS232 3.6	[3.5]	RS232 3.5	[3.4]	RS232 3.4
[3.3]	RS232 3.3	[3.2]	RS232 3.2	[3.1]	RS232 3.1	[3.0]	RS232 3.0

3.4. I/O module

3.4.1. SP-SDIO[1]

3.4.1.1. General information

Type code	Serial number	SW version	Hardware version	Version/Step	Address
SP-SDIO84-P1K-	0401 0000	V 1.00.0	0.00	1.3.0.127 V 1.xx	1

3.4.1.2. Inputs

	Mode	Title/tag name	ON-OFF	OFF-ON	Dis. [ms]
2 24V	I5	Reset (Single Channel)	-	-	-
3 24V	I6	EDM (Single Channel)	-	-	-
8 X1	I3	Interlock (Dual Channel) / door left	-	-	3000
X2	I4				
7 X1	I1	E-Stop, SNH (Dual Channel)	-	-	3000
X2	I2				
9 X1	I7	Safety Switch (Dual Channel) / barrier-safety switches	-	-	3000
X2	I8				

3.4.1.3. Outputs

	Mode	Title/tag name
4	Q1	Motor Contactor (Single Channel) / GE 3S10EMCR S.1
5	Q2	Motor Contactor (Single Channel) / GE 3S10EMCR S.6
6	Q3	Interlock (Dual Channel)

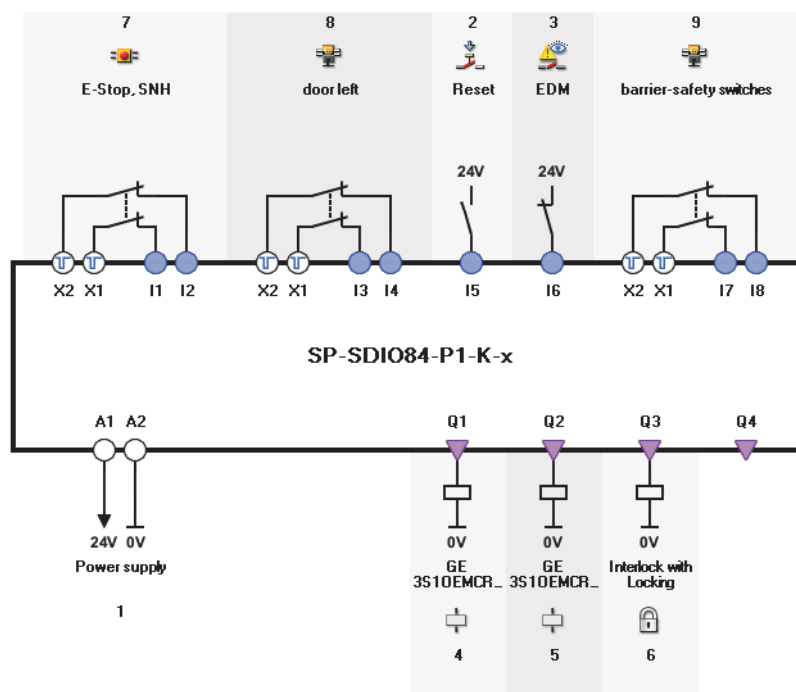
3.4.1.4. Power supply

	Title/tag name
1 24V	SP-SDIO[1] Power Supply
0V	

samosPLAN 3/10/2011 3:28:06 PM



3.4.1.5. Wiring diagram



3.4.1.6. Test pulse parameter

Test outputs	Test period [ms]	Test gap [ms]	Max. off-on delay [ms]
X1	200	1	-
X2	200	1	-

3.4.2. SP-SDI[2]

3.4.2.1. General information

Type code	Serial number	SW version	Hardware version	Version/Step	Address
SP-SDI8-P1-K-x	0401 0000	V1.00.0	0.00	1.3.0.127 V1.xx	2

3.4.2.2. Inputs

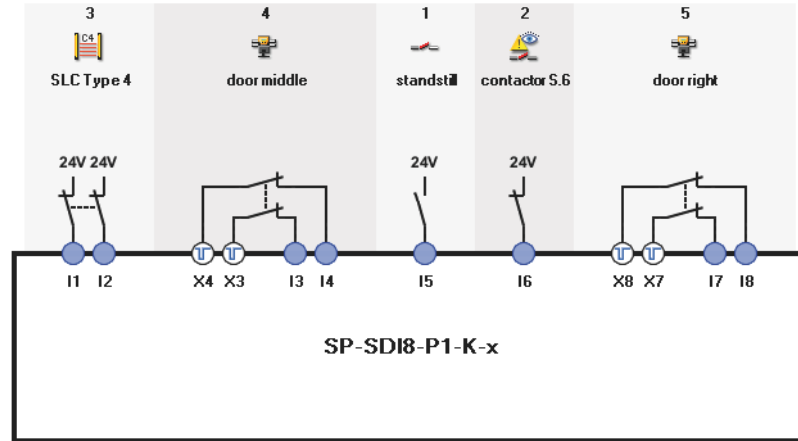
	Mode	Title/tag name	ON-OFF	OFF-ON	Dis. [ms]
1 24V	NO	Single Channel NO (Single Channel) / standstill	-	-	-
2 24V	NC	EDM (Single Channel) / contactor S.8	-	-	-
3 24V 24V	NC	SLC Type 4 (Safety Light Curtain, Dual Channel, Type 4)	-	-	3000

samosPLAN 3/10/2011 3:28:06 PM

wieland

	Mode	Title/tag name	ON-OFF	OFF-ON	Dis. [ms]
4	X3 X4	Safety Switch (Dual Channel) / door middle	-	-	3000
5	X7 X8	Safety Switch (Dual Channel) / door right	-	-	3000

3.4.2.3. Wiring diagram



3.4.2.4. Test pulse parameter

Test outputs	Test period [ms]	Test gap [ms]	Max. off-on delay [ms]
X1/X3/X5/X7	200	1	-
X2/X4/X6/X8	200	1	-

3.5. Gateway

3.5.1. SP-EN-IP - General Information

Type code	Serial number	SW version	Hardware version	Version/Step	Address
SP-EN-IP	0401 0000	V1.00.0	0.00	1.3.0.127 V1.xx	13

3.5.2. SP-EN-IP - Routing

samosPLAN 3/10/2011 3:28:06 PM

wieland

3.5.2.1. CPU to Network

Update interval [ms]

4

Input Data Set 1

[Byte 0] Byte 0

[0.7] result 0.7	[0.6] result 0.6	[0.5] saw-doors active	[0.4] lightcurtain active
[0.3] standstill saw motor	[0.2] EDM-fault 2	[0.1] EDM-fault 1	[0.0] E-stop in function

[Byte 4] Module 0 [SP-SDIO[1]]

[4.7] -	[4.6] barrier-safety switches.SP-SDIO [1].I7.I8	[4.5] EDM.SP-SDIO[1].I6	[4.4] Reset.SP-SDIO[1].I5
[4.3] -	[4.2] door left.SP-SDIO [1].I3.I4	[4.1] -	[4.0] E-Stop. SNH.SP-SDIO [1].I1.I2

[Byte 5] Module 1 [SP-SDI[2]]

[5.7] -	[5.6] door right.SP-SDI [2].I7.I8	[5.5] contactor S.6.SP-SDI [2].I6	[5.4] standstill.SP-SDI[2].I5
[5.3] -	[5.2] door middle.SP-SDI [2].I3.I4	[5.1] -	[5.0] SLC Type 4.SP-SDI [2].I1.I2

[Byte 16] Module 0 [SP-SDIO[1]]

[16.7]	[16.6] Interlock with Locking.SP-SDIO [1].Q3	[16.5] GE 3S1OEMCR S.6.SP-SDIO[1].Q2	[16.4] GE 3S1OEMCR S.1.SP-SDIO[1].Q1
[16.3]		[16.1]	

3.5.2.2. Network to CPU

Update interval [ms]

4

Output Data Set 1

[W0.LB (Low Byte)] S1.LB0

[0.7] Bit7	[0.6] Bit6	[0.5] Bit5	[0.4] Bit4
[0.3] Bit3	[0.2] Bit2	[0.1] Bit1	[0.0] Reset of PLC

[W0.HB (High Byte)] S1.HB0

[1.7] Bit7	[1.6] Bit6	[1.5] Bit5	[1.4] Bit4
[1.3] Bit3	[1.2] Bit2	[1.1] Bit1	[1.0] Bit0

[W1.LB (Low Byte)] S1.LB1

[2.7] Bit7	[2.6] Bit6	[2.5] Bit5	[2.4] Bit4
[2.3] Bit3	[2.2] Bit2	[2.1] Bit1	[2.0] Bit0

[W1.HB (High Byte)] S1.HB1

[3.7] Bit7	[3.6] Bit6	[3.5] Bit5	[3.4] Bit4
[3.3] Bit3	[3.2] Bit2	[3.1] Bit1	[3.0] Bit0

3.5.3. SP-EN-IP - Device configuration

General configuration

IP address	Subnet mask	Default gateway	DHCP	Timeout (s)	Device name
192.168.250.250	255.255.0.0	0.0.0.0	off	0	SP-EN-IP

Page 11

samosPLAN 3/10/2011 3:28:06 PM

TCP/IP configuration - Socket 1

Listen on	Connect to	Auto update	Update rate	Data set 1	Data set 2	Data set 3	Data set 4
9100	-	Off	-	On	Off	Off	Off

Ethernet/IP configuration - samosPRO to PLC

Connection mode	COS update	Heartbeat rate	Data set 1/ Update rate (ms)	Data set 2/ Update rate (ms)	Data set 3/ Update rate (ms)	Data set 4/ Update rate (ms)
PLC requests	Off	On	InDataSet1 / 5000	InDataSet2 / 5000	InDataSet3 / 5000	InDataSet4 / 5000

Ethernet/IP configuration - PLC to samosPRO

Connection mode	Data set 1/ Update rate (ms)	Data set 2/ Update rate (ms)	Data set 3/ Update rate (ms)	Data set 4/ Update rate (ms)	Data set 5/ Update rate (ms)
PLC writes	OutDataSet1 / 5000	OutDataSet2 / 5000	OutDataSet3 / 5000	OutDataSet4 / 5000	OutDataSet5 / 5000

Page 12

14.2 List of tables

Note: Table count omits "4" for reasons of consistency with samos®PLAN software

Tab. 1:	Overview of the samos®PRO documentation	8
Tab. 2:	Required firmware and software versions	11
Tab. 3:	Errors and error elimination	13
Tab. 5:	Meaning of the EFI I/O error status bits	33
Tab. 6:	Meaning of the module status bits	33
Tab. 7:	Meaning of the diagnostics information	39
Tab. 8:	Meaning of the background color	42

Tab. 9:	Symbols for editing the connection profiles in the Connection settings dialog	43
Tab. 10:	Status symbols for connection profiles	45
Tab. 11:	User level authorisations	47
Tab. 12:	Minimum system requirements for samos®NET	49
Tab. 13:	Available status bits depending on the connection method	49
Tab. 14:	samos®NET system overview tools	62
Tab. 15:	Meaning of the Teaching status bits	70
Tab. 16:	Function block overview.....	72
Tab. 17:	Truth table for the NOT function block.....	76
Tab. 18:	Parameters of the AND function block.....	77
Tab. 19:	Truth table for AND evaluation with one input	77
Tab. 20:	Truth table for AND evaluation with two inputs	77
Tab. 21:	Truth table for AND evaluation with eight inputs.....	77
Tab. 22:	Parameters of the OR function block	78
Tab. 23:	Truth table for OR evaluation with one input	78
Tab. 24:	Truth table for OR evaluation with two inputs	78
Tab. 25:	Truth table for OR evaluation with eight inputs	78
Tab. 26:	Truth table for XOR evaluation	79
Tab. 27:	Truth table for XNOR evaluation.....	79
Tab. 28:	Parameters of the RS Flip-Flop function block.....	80
Tab. 29:	Truth table for the RS Flip-Flop function block	80
Tab. 30:	Parameters of the JK Flip-Flop function block	81
Tab. 31:	Truth table for the JK Flip-Flop function block.....	81
Tab. 32:	Parameters of the Clock generator function block.....	82
Tab. 33:	Parameter for the Event counter function blocks.....	82
Tab. 34:	Truth table for the Event counter function blocks (Up, Down and Up/Down).....	84
Tab. 35:	Parameters of the Fast shut off function block	85
Tab. 36:	Parameters of the Edge detection function block	89
Tab. 37:	Parameters of the Binary encoder function block	90
Tab. 38:	Truth table for the Binary encoder function block with 2 inputs in One out of N mode.....	91
Tab. 39:	Truth table for the Binary encoder function block with 8 inputs in One out of N mode.....	91
Tab. 40:	Truth table for the Binary encoder function block with 2 inputs in Priority mode.....	91
Tab. 41:	Truth table for the Binary encoder function block with 8 inputs in Priority mode.....	91
Tab. 42:	Truth table for the Binary encoder function block with 2 inputs in Priority mode with dominant input 1	92
Tab. 43:	Truth table for the Binary encoder function block with 8 inputs in Priority mode with dominant input 1	92
Tab. 44:	Parameters of the Binary decoder function block	93
Tab. 45:	Truth table for the Binary decoder with 1 input in One out of N mode	93
Tab. 46:	Truth table for the Binary decoder with 2 inputs in One out of N mode	93
Tab. 47:	Truth table for the Binary decoder with 3 inputs in One out of N mode	93
Tab. 48:	Truth table for the Binary decoder with 1 input in Priority mode	93
Tab. 49:	Truth table for the Binary decoder with 2 inputs in Priority mode.....	93
Tab. 50:	Truth table for the Binary decoder with 3 inputs in Priority mode.....	94
Tab. 51:	Value range of the Binary decoder function block depending on input D.....	94
Tab. 52:	Value range of the Binary decoder function block depending on inputs D and E.....	94
Tab. 53:	Parameters of the Message generator function block	95
Tab. 54:	Parameters of the Routing 1:N function block	97
Tab. 55:	Parameters of the Routing 1:N function block	98
Tab. 56:	Parameters of the Reset function block	98
Tab. 57:	Parameters of the Restart function block.....	100

Tab. 58:	Parameters of the Off-delay timer function block.....	101
Tab. 59:	Parameters of the Adjustable off-delay timer function block.....	101
Tab. 60:	Parameters of the On-delay timer function block.....	102
Tab. 61:	Parameters of the Adjustable on-delay timer function block	103
Tab. 62:	Parameters of the EDM function block.....	104
Tab. 63:	Parameters of the Valve monitoring function block	106
Tab. 64:	Parameters of the User mode switch function block	108
Tab. 65:	Truth table for the User mode switch function block.....	109
Tab. 66:	Parameters of the Switch synchronization function block	110
Tab. 67:	Parameters of the Error output combination function block.....	113
Tab. 68:	Truth table for Error output combination.....	113
Tab. 69:	Parameters of the Ramp down detection function block.....	114
Tab. 70:	Maximum allowed encoder signal frequency and speed (rpm) depending on the encoder type	116
Tab. 71:	Parameters of the Start warning function block.....	120
Tab. 72:	Dual-channel evaluation.....	125
Tab. 73:	Double dual-channel evaluation (synchronization evaluation).....	127
Tab. 74:	Parameters of the Emergency stop function block	128
Tab. 75:	Parameters of the Magnetic switch function block.....	129
Tab. 76:	Parameters of the Light curtain monitoring function block.....	130
Tab. 77:	Parameters of the Safety gate monitoring function block.....	130
Tab. 78:	Parameters of the Two hand control type IIIA function block.....	132
Tab. 79:	Parameters of the Two hand control type IIIC function block.....	133
Tab. 80:	Parameters for the Multi operator function block	134
Tab. 81:	Parameters of the muting function blocks	138
Tab. 82:	Requirements for sequence monitoring	140
Tab. 83:	Conditions for Override required and Override possible.....	141
Tab. 84:	Number of permissible override cycles	142
Tab. 85:	Effects of the conveyor belt monitoring on the timer functions.....	142
Tab. 86:	Output values for muting status	143
Tab. 87:	Output values for the Muting lamp output	143
Tab. 88:	Wiring combinations for muting and prerequisites	144
Tab. 89:	Stop-to-Run transition behaviour for muting functions.....	144
Tab. 90:	Error states and information on resetting for Muting function blocks	145
Tab. 91:	Conditions for Parallel muting.....	145
Tab. 92:	Conditions for Sequential muting	147
Tab. 93:	Conditions for Cross muting with optional input C1	149
Tab. 94:	Conditions for Cross muting without optional input C1	151
Tab. 95:	Overview of the press contact monitoring function blocks.....	152
Tab. 96:	Parameters of the Eccentric press contact function block.....	153
Tab. 97:	Parameters of the Universal press contact function block.....	158
Tab. 98:	Parameters of the Press Setup function block	165
Tab. 99:	Parameters of the Press single stroke function block	167
Tab. 100:	Parameters of the Press automatic function block.....	171
Tab. 101:	Parameters of the N-break function block.....	174
Tab. 102:	Error states and information on resetting for the N-break function block	180
Tab. 103:	Dual-channel evaluation.....	193
Tab. 104:	Device status and LED displays on the CPUx.....	200
Tab. 105:	Run button and Stop button	201

14.3 List of illustrations

Note: Figure count omits "18" for reasons of consistency with samos®PLAN software

Fig. 1:	Start view with selection of the action	14
Fig. 2:	The view can be selected below the menu bar	15
Fig. 3:	Sub-windows can be converted to flyout menus.....	16

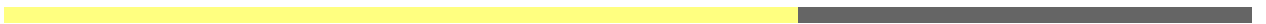
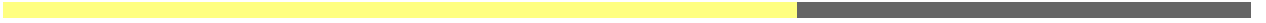
Fig. 4:	The “Hardware configuration” view.....	17
Fig. 5:	Switch view button.....	17
Fig. 6:	Settings button	18
Fig. 7:	Edit tag names button	18
Fig. 8:	Edit button in the hardware view	18
Fig. 9:	CPU status bits in the Hardware configuration view	19
Fig. 10:	Gateway status bits in the Hardware configuration view	19
Fig. 11:	SP-SDIO module status bits in the Hardware configuration view	20
Fig. 12:	SP-SDI module status bits in the Hardware configuration view	20
Fig. 13:	Element settings window for an SNH emergency stop button	22
Fig. 14:	Create custom element template window	25
Fig. 15:	Add or delete sub-elements for a customized element.....	26
Fig. 16:	Entering the BOM info for a customized element.....	27
Fig. 17:	Editing the configuration settings of a customized element.....	27
Fig. 19:	Export configuration dialog	28
Fig. 20:	Import configuration dialog	29
Fig. 21:	The logic editor	30
Fig. 22:	Invalid configuration warnings	32
Fig. 23:	Example for the usage of a CPU marker	34
Fig. 24:	Example of jump addresses with and without loop back	35
Fig. 25:	I/O matrix in offline mode.....	36
Fig. 26:	I/O matrix in simulation mode	37
Fig. 27:	Tag name editor.....	37
Fig. 28:	Report view.....	38
Fig. 29:	Diagnostics view.....	39
Fig. 30:	Toolbar in the diagnostics view	40
Fig. 31:	Filtering the diagnostics messages	40
Fig. 32:	Connection settings dialog	41
Fig. 33:	Modify profile dialog.....	41
Fig. 34:	Hardware configuration dialog (example)	42
Fig. 35:	Connection settings dialog	43
Fig. 36:	Create new profile dialog (serial port)	43
Fig. 37:	Create new profile dialog (TCP/IP).....	44
Fig. 38:	List of gateways found	45
Fig. 39:	List of gateways found	46
Fig. 40:	Change password dialog	48
Fig. 41:	Reset password dialog	48
Fig. 42:	samos®NET IDs display in the samos®NET System overview.....	50
Fig. 43:	samos®NET Network settings without valid address assignment	52
Fig. 44:	Apply settings button	52
Fig. 45:	samos®NET Network settings with valid address assignment.....	53
Fig. 46:	samos®NET System overview view.....	53
Fig. 47:	samos®NET System overview screen.....	54
Fig. 48:	samos®NET system overview and station buttons.....	54
Fig. 49:	Logic configuration example (Station A)	55
Fig. 50:	CPU outputs in the output selection window	55
Fig. 51:	samos®NET routing logic example	56
Fig. 52:	samos®NET routing table and tag names.....	56
Fig. 53:	Assigned samos®NET tag names in the routing configuration and in the logic editor	57
Fig. 54:	Routed inputs from Station A in the Station B logic editor.....	57
Fig. 55:	Logic configuration example (Station B)	57
Fig. 56:	samos®NET system overview, system not connected	58
Fig. 57:	Connect dialog.....	58
Fig. 58:	samos®NET system overview, system connected, differing configurations.....	59
Fig. 59:	samos®NET system overview, system connected, configuration identical but not verified.....	59
Fig. 60:	samos®NET system and station buttons	60

Fig. 61:	samos®NET system menu bar.....	60
Fig. 62:	samos®NET station menu bar	61
Fig. 63:	samos®NET station menu bar with gateway	61
Fig. 64:	samos®NET system overview	62
Fig. 65:	samos®NET process image	63
Fig. 66:	samos®NET Network settings view	64
Fig. 67:	Apply settings button	64
Fig. 68:	samos®NET output bits for Station A in the logic editor	65
Fig. 69:	Routing an input into the samos®NET network	65
Fig. 70:	samos®NET input bits from Station A in the logic editor of another station	66
Fig. 71:	samos®NET station A view.....	66
Fig. 72:	Default values for samos®NET input bits	67
Fig. 73:	Configuration of the Teach function in the logic editor	68
Fig. 74:	samos®NET system status information in the logic editor	69
Fig. 75:	CPU marker	72
Fig. 76:	Jump address with loop back	72
Fig. 77:	Configurable parameters of function blocks.....	73
Fig. 78:	Inversion of function block input connections	74
Fig. 79:	Example of an AND function block with one inverted input	74
Fig. 80:	In/Out configuration of the Safety gate monitoring function block.....	75
Fig. 81:	Activating the Fault present output	75
Fig. 82:	Fault present output.....	76
Fig. 83:	Function block diagram for the NOT function block	76
Fig. 84:	Function block diagram for the AND function block.....	77
Fig. 85:	Function block diagram for the OR function block.....	78
Fig. 86:	Function block diagram for the Exclusive OR (XOR) function block	79
Fig. 87:	Function block diagram for the Exclusive NOR (XNOR) function block	79
Fig. 88:	Logic connections for the RS Flip-Flop function block	80
Fig. 89:	Logic connections for the JK Flip-Flop function block	80
Fig. 90:	Logic connections for the Clock generator function block.....	81
Fig. 91:	Parameter diagram for the Clock generator function block	82
Fig. 92:	Timing diagram for the Clock generator function block.....	82
Fig. 93:	Logic connections for the Event counter (Up and down) function block.....	82
Fig. 94:	Logic connections for the Fast shut off and Fast shut off with bypass function blocks.....	84
Fig. 95:	Fast shut off example.....	85
Fig. 96:	Configuration example for Fast shut off with three light curtains.....	86
Fig. 97:	In/Out settings dialog for the Fast shut off function block	86
Fig. 98:	Parameter settings dialog for the Fast shut off function block	87
Fig. 99:	Output selection for Fast shut off	87
Fig. 100:	Hardware configuration view of inputs and out-puts linked to Fast shut off.....	88
Fig. 101:	Example for Fast shut off with bypass with more than one bypass conditions.....	89
Fig. 102:	Logic connections for the Edge detection function block	89
Fig. 103:	Timing diagram for the Edge detection function block	90
Fig. 104:	Logic connections for the Binary encoder function block	90
Fig. 105:	Logic connections for the Binary decoder function block	92
Fig. 106:	Combination of four Binary decoder function blocks	94
Fig. 107:	Function block diagram for the Message generator function block.....	95
Fig. 108:	Configuration example for Message generator with two emergency stop buttons and a safety switch	95
Fig. 109:	In/out settings for the Message generator function block	96
Fig. 110:	Messages of the Message generator function block.....	96
Fig. 111:	Message assignment for the Message generator function block	97
Fig. 112:	Function block diagram for the Routing 1:N function block	97
Fig. 113:	Function block diagram for the Routing N:N function block.....	98
Fig. 114:	Function block diagram for the Reset function block.....	98
Fig. 115:	Sequence/ timing diagram for the Reset function block	99

Fig. 116: Function block diagram for the Restart function block.....	99
Fig. 117: Sequence/ timing diagram for the Restart function block	100
Fig. 118: Function block diagram for the Off-delay timer function block	100
Fig. 119: Sequence/ timing diagram for the Off-delay timer function block.....	101
Fig. 120: Function block diagram for the Adjustable off-delay timer function block.....	101
Fig. 121: Sequence/ timing diagram for the Adjustable off-delay timer function block with Off delay time 1 and Off delay time 2.....	102
Fig. 122: Function block diagram for the On-delay timer function block	102
Fig. 123: Sequence/ timing diagram for the On-delay timer function block	103
Fig. 124: Function block diagram for the Adjustable on-delay timer function block.....	103
Fig. 125: Sequence/ timing diagram for the Adjustable on-delay timer function block with On delay time 1 and On delay time 2	104
Fig. 126: Function block diagram for the EDM function block	104
Fig. 127: Sequence/ timing diagram for the External device monitoring (EDM) function block.....	105
Fig. 128: Logic connections for the Valve monitoring function block, configured for a directional valve	105
Fig. 129: Sequence/ timing diagram for single valve in manual reset mode	107
Fig. 130: Sequence/ timing diagram for double valve in manual reset mode.....	107
Fig. 131: Sequence/ timing diagram for directional valve.....	108
Fig. 132: Function block diagram for the User mode switch function block.....	108
Fig. 133: Sequence/ timing diagram for the User mode switch function block	109
Fig. 134: Logic connections for the Switch synchronization function block	110
Fig. 135: Logic example for two cascaded Switch synchronization function blocks	111
Fig. 136: Sequence/	112
Fig. 137: Sequence/ timing diagram for the Switch synchronization function block with cascading.....	112
Fig. 138: Logic connections for the Error output combination function block	113
Fig. 139: Logic example for the Error output combination function block	113
Fig. 140: Logic connections for the Ramp down detection function block	114
Fig. 141: Signal pattern for A/B 90° phase shift encoders	115
Fig. 142: Signal pattern for 1/3 gap 180° phase shift encoders.....	115
Fig. 143: Signal pattern for zero pulse encoders.....	115
Fig. 144: Logic example for the Ramp down detection function block	118
Fig. 145: Sequence/ timing diagram for the Ramp down detection function block	118
Fig. 146: Sequence/ timing diagram for the Ramp down detection function block with plausibility check	119
Fig. 147: Logic connections for the Start warning function block.....	119
Fig. 148: Logic example for a combination of two Start warning function blocks.....	122
Fig. 149: Sequence/ timing diagram for the Start warning function block in Not locked mode	122
Fig. 150: Sequence/ timing diagram for the Start warning function block in Locked mode.....	123
Fig. 151: Example for single-channel evaluation.....	124
Fig. 152: Dual-channel evaluation with I/O module or with function block.....	124
Fig. 153: Sequence/ timing diagram for the Emergency stop function block.....	126
Fig. 154: Double dual-channel evaluation with the Safety gate monitoring function block	126
Fig. 155: Sequence/ timing diagram for the Safety gate monitoring function block, Category 4, dual-channel equivalent (2 pairs) without function test	128
Fig. 156: Function block diagram for the Emergency stop function block	128
Fig. 157: Logic connections for the Magnetic switch function block.....	129
Fig. 158: Function block diagram for the Light curtain monitoring function block	129
Fig. 159: Function block diagram for the Safety gate monitoring function block	130
Fig. 160: Sequence/ timing diagram for the Safety gate monitoring function block, Category 2, single-channel with function test.....	131
Fig. 161: Sequence/ timing diagram for the Safety gate monitoring function block, Category 4, dual-channel equivalent (1 pair) without function test	131

Fig. 162: Function block diagram for the Two-hand control type IIIA function block.....	132
Fig. 163: Function block diagram for the Two hand control type IIIC function block.....	132
Fig. 164: Sequence/ timing diagram for the Two hand control type IIIC function block.....	133
Fig. 165: Logic connections for the Multi operator function block	134
Fig. 166: Sequence/ timing diagram for the Multi operator function block	135
Fig. 167: Safety when mounting the muting sensors.....	137
Fig. 168: Detection of material during muting.....	137
Fig. 169: Logic diagram for Override and Override required	141
Fig. 170: Logic connections for the Parallel muting function block	145
Fig. 171: Muting with two parallel sensor pairs	145
Fig. 172: Valid muting sequence using the configuration basic setting.....	146
Fig. 173: Logic connections for the Sequential muting function block.....	147
Fig. 174: Example of the sequential layout of muting sensors.....	147
Fig. 175: Valid muting sequence using the basic configuration setting.....	148
Fig. 176: Logic connections for the Cross muting function block with optional input C1	149
Fig. 177: Example of Cross muting with optional input C1	149
Fig. 178: Valid muting sequence using the configuration basic setting.....	150
Fig. 179: Logic connections for the Cross muting function block.....	150
Fig. 180: Cross muting for bidirectional movement of material	151
Fig. 181: Valid muting sequence using the basic configuration setting.....	152
Fig. 182: Logic connections for the Eccentric press contact function block.....	153
Fig. 183: Press cycle for the Eccentric press contact function block without Dynamic cam	154
Fig. 184: Press cycle for the Eccentric press contact function block with Dynamic cam on upstroke	155
Fig. 185: Press cycle for the Eccentric press contact function block with Dynamic cam on upstroke and downstroke	155
Fig. 186: Contact monitoring with the Eccentric press contact function block	156
Fig. 187: Overrun monitoring with the Eccentric press contact function block.....	157
Fig. 188: Logic connections for the Universal press contact function block	158
Fig. 189: Press cycle for the Universal press contact function block with falling edge of TDC before BDC.....	159
Fig. 190: Press cycle for the Universal press contact function block with falling edge of BDC before TDC.....	160
Fig. 191: Press cycle for the Universal press contact function block with 2 BDC transitions.....	160
Fig. 192: Contact monitoring with the Universal press contact function block with SCC enabled.....	161
Fig. 193: Contact monitoring with the Universal press contact function block with BDC enabled	161
Fig. 194: Contact monitoring with the Universal press contact function block with BDC enabled	162
Fig. 195: Timing diagrams for 0, 1 and 2 BDC signals per cycle.....	163
Fig. 196: Overrun monitoring with the Universal press contact function block.....	164
Fig. 197: Logic connections for the Press setup function block.....	165
Fig. 198: Sequence/ timing diagram for the Press setup function block.....	167
Fig. 199: Logic connections for the Press single stroke function block.....	167
Fig. 200: Sequence/ timing diagram for the Press single stroke function block when Start/Release is configured in stepping mode	170
Fig. 201: Sequence/ timing diagram for the Press single stroke function block when Start/Release is configured in start only mode	170
Fig. 202: Sequence/ timing diagram for the Press single stroke function block with upstroke muting of Start/Release and Release 3 (safety)	170
Fig. 203: Logic connections for the Press automatic function block	171
Fig. 204: Sequence/ timing diagram for the Press automatic function block using the Stop request and Upstroke inputs	173

Fig. 205: Logic connections for the N-break function block	173
Fig. 206: Sequence/ timing diagram for a complete start sequence in Standard mode in two-cycle mode	175
Fig. 207: Sequence/ timing diagram for a complete start sequence in Sweden mode in two-cycle mode	175
Fig. 208: Valid breaks when the Start of first PSDI pulse parameter is set to After the start of upstroke	176
Fig. 209: Sequence/ timing diagram for Upstroke muting in Standard mode in two-cycle mode	177
Fig. 210: Sequence/ timing diagram, when the PSDI input is Low, Upstroke muting is inactive and Restart interlock is set to "For all stops"	179
Fig. 211: Sequence/ timing diagram when the PSDI input is Low, Max. time for upstroke muting > 0 and Restart interlock is set to "For stops in downstroke and in top (in upstroke ignored)"	179
Fig. 212: Logic connections for the Grouped function block	180
Fig. 213: Creating a grouped function block	181
Fig. 214: Edit Function Block Details dialog for the Grouped function block	181
Fig. 215: Select Icon dialog for the Grouped function block	181
Fig. 216: New grouped function block on the worksheet	182
Fig. 217: New logic editor page for the new grouped function block	182
Fig. 218: Adding inputs and outputs to a grouped function block	183
Fig. 219: Grouped function block with devices attached	183
Fig. 220: Switching between internal and external view	183
Fig. 221: Logic connections for the customized function block	184
Fig. 222: Edit Function Block Details dialog for the customized function block	185
Fig. 223: Select Icon dialog for the customized function block	185
Fig. 224: New customized function block in the function block list	185
Fig. 225: Icons for the grouped function block and for the customized function block	186
Fig. 226: Simulation toolbar	187
Fig. 227: Simulation mode started, simulation off	187
Fig. 228: Simulation mode started, simulation running	188
Fig. 229: Dialog window when starting force mode	189
Fig. 230: Logic editor with force mode activated	190
Fig. 231: Forced and not forced inputs	190
Fig. 232: Dialog before auto-matic termination of the force mode	192
Fig. 233: Upload and verify configuration	196
Fig. 234: Setting a device to verified.	197
Fig. 235: Upload and verify failed	198
Fig. 236: Verification successful	198





wieland

Wieland Electric GmbH
Brennerstraße 10-14
D-96052 Bamberg
Tel. +49 (0) 951 / 9324 -0
Fax +49 (0) 951 / 9324 -198

Email info@wieland-electric.com
www.wieland-electric.com

