

# ***samos***<sup>®</sup> PRO

***samos***<sup>®</sup> PLAN5+ Software

## **Manual**

Doc. no. BA000968

Last Update: 07/2016 (Rev. D)



# Info

## Copyright

This document is copyright-protected. The rights derived from this copyright are reserved for Wieland Electric GmbH. Reproduction of this document or parts of this document is only permissible within the limits of the statutory provision of the Copyright Act. Any modification or abridgment of the document is prohibited without the express written agreement of Wieland Electric GmbH.

samos is a registered trademark of WIELAND Electric GmbH

Allen-Bradley, CompactBlock Guard I/O, CompactLogix, ControlFLASH, ControlLogix, DH+, FactoryTalk, FLEX, GuardLogix, Kinetix, Logix5000, MicroLogix, PanelBuilder, PanelView, PhaseManager, PLC-2, PLC-3, PLC-5, POINT I/O, POINT Guard I/O, Rockwell Automation, Rockwell Software, RSBizWare, RSFieldbus, RSLinx, RSLogix 5000, RSNetWorx, RSView, SLC, SoftLogix, Stratix, Stratix 2000, Stratix 5700, Stratix 6000, Stratix 8000, Stratix 8300, Studio 5000, Studio 5000 Logix Designer, SynchLink, and Ultra are registered trademarks of Rockwell Automation, Inc.

ControlNet, DeviceNet, and EtherNet/IP are registered trademarks of ODVA, Inc.

TwinCAT is a registered trademark of Beckhoff Automation GmbH.

EtherCAT registered trademark and patented technology, licensed by Beckhoff Automation GmbH.

Microsoft, Windows 98, Windows NT, Windows 2000, Windows XP, Windows 7, Windows 8, and .NET Framework are registered trademarks of the Microsoft Corporation.

Any other product or trade names listed in this manual are the trademarks or registered trademarks of the respective owners.

## Subject to change.

Subject to technical changes for reasons of continued development.

# Table of Contents

<b>1</b>	<b>About this manual</b>	<b>9</b>
1.1	Function of this document	9
1.2	Scope of application and version	9
1.3	Target group	10
1.4	Function and setup of this software manual	10
1.4.1	Recommendations for getting to know the software	10
1.4.2	Recommendations for experienced users	10
1.5	Symbols/icons and writing style/spelling standard used	11
<b>2</b>	<b>Safety</b>	<b>12</b>
2.1	Qualified persons	12
2.2	Proper use	12
<b>3</b>	<b>Version, compatibility, and features</b>	<b>13</b>
3.1	Version info	13
<b>4</b>	<b>Installation and removal</b>	<b>14</b>
4.1	System requirements	14
4.2	Installation	14
4.3	Update	14
4.4	Removal	14
4.5	Troubleshooting and eliminating errors	15
<b>5</b>	<b>User interface</b>	<b>16</b>
5.1	Setup and function	16
5.1.1	Areas at a glance	16
5.1.2	Menu bar	17
5.1.3	Window behavior of sidebars	19
5.1.4	Commands	20
5.1.5	Favorites for hardware and logic	21
5.1.6	Overview safety and security mechanism	21
5.1.7	Program help	22

<b>5.2</b>	<b>Cross-project settings (main menu)</b>	<b>24</b>
5.2.1	User interface language	25
5.2.2	Configuration of display names	26
5.2.3	Displaying start view	27
5.2.4	Instructions for automating (saving, logging off, updating)	28
5.2.5	Specifications for the logic editor	29
5.2.6	Updates	30
5.2.7	Proxy settings	30
5.2.8	Importing/Exporting settings	31
<b>5.3</b>	<b>"Overview" view</b>	<b>32</b>
5.3.1	Setup and functions	32
5.3.2	Creating a new project (exercise)	33
5.3.3	Log messages	34
5.3.4	User administration	35
<b>5.4</b>	<b>"Hardware" view</b>	<b>36</b>
5.4.1	Setup and functions	36
5.4.2	Selecting and parameterizing modules (exercise)	39
5.4.2.1	Special case: SP-XX expansion module	40
5.4.3	Placing actuators and sensors (exercise)	41
5.4.4	Parameterizing actuators and sensors (parameter reference)	42
5.4.5	Creating user-defined elements (exercise)	45
5.4.6	Exporting and importing user-defined elements	46
<b>5.5</b>	<b>"Logic" view</b>	<b>47</b>
5.5.1	Setup and functions	47
5.5.2	Programming logic incrementally (exercise)	51
5.5.3	Status bits for type SP-COP modules (reference)	53
5.5.4	Automatic logic check	55
5.5.5	Using CPU flags	56
5.5.6	Using jump addresses	56
5.5.7	Views of logic programming (tip)	57
<b>5.6</b>	<b>"Gateway" view</b>	<b>59</b>
<b>5.7</b>	<b>"Name" view</b>	<b>59</b>
5.7.1	Setup and functions	59
5.7.2	Assigning Tag names/changing display names (exercise)	61
<b>5.8</b>	<b>"Report" view</b>	<b>62</b>
5.8.1	Setup	62
5.8.2	Functions	63
5.8.3	Edit front page	64

<b>5.9</b>	<b>"Diagnostics" view</b>	<b>65</b>
5.9.1	Setup	65
5.9.2	Synchronize time for diagnostic purposes	67
<b>6</b>	<b>Connecting to the samosPRO system</b>	<b>68</b>
<b>7</b>	<b>Logic programming – Function blocks</b>	<b>72</b>
<b>7.1</b>	<b>General safety information regarding logic programming</b>	<b>72</b>
<b>7.2</b>	<b>Function block overview</b>	<b>73</b>
<b>7.3</b>	<b>Function block properties</b>	<b>74</b>
<b>7.4</b>	<b>Input and output signal connections of function blocks</b>	<b>75</b>
7.4.1	Function block input connections	75
7.4.2	Inverting inputs	75
7.4.3	Output connections of function blocks	76
<b>7.5</b>	<b>Parameterization of function blocks</b>	<b>77</b>
7.5.1	Time values and logic execution time	77
7.5.2	Error outputs	77
<b>7.6</b>	<b>Logical function blocks</b>	<b>78</b>
7.6.1	NOT	78
7.6.2	AND	78
7.6.3	OR	80
7.6.4	XOR (exclusive OR)	81
7.6.5	T flip-flop	82
7.6.6	RS flip-flop	83
7.6.7	JK flip-flop	84
7.6.8	Clock generator	86
7.6.9	Event counter (upward, downward, up- and downward)	88
7.6.10	Fast shut-off and fast shut-off with bypass	91
7.6.11	Edge detection	96
7.6.12	Binary encoder	97
7.6.13	Binary decoder	100
7.6.14	Log generator	104
7.6.15	Residual memory	107
<b>7.7</b>	<b>Application-specific function blocks</b>	<b>108</b>
7.7.1	Reset	108
7.7.2	Restart	110
7.7.3	Switch-off delay timer	112
7.7.4	Adjustable switching-off delay	113
7.7.5	Switch-on delay timer	114
7.7.6	Adjustable switching-on delay	115
7.7.7	EDM (Contact monitor)	116

7.7.8	Valve monitoring	118
7.7.9	Operation mode selection switch	121
7.7.10	Ramp down detection	123
<b>7.8</b>	<b>Function blocks for two-channel evaluation</b>	<b>128</b>
7.8.1	Emergency stop	129
7.8.2	Solenoid switch	130
7.8.3	Light curtain evaluation	131
7.8.4	Switch evaluation	132
7.8.5	Two-hand control, type IIIA	134
7.8.6	Two-hand control, type IIIC	135
7.8.7	Multi-operator start	137
<b>7.9</b>	<b>Function blocks for parallel muting, sequential muting, and cross muting</b>	<b>139</b>
7.9.1	Overview and general description	139
7.9.2	Parameters of the function blocks	142
7.9.2.1	Directional detection	143
7.9.2.2	Condition for muting start	143
7.9.2.3	Condition for muting end	143
7.9.2.4	Total muting time	143
7.9.2.5	Additional muting time after the contactless safety device is free	144
7.9.2.6	Simultaneity monitoring time	144
7.9.2.7	Suppressing sensor signal gaps	144
7.9.2.8	Sequence monitoring	144
7.9.2.9	C1 input	145
7.9.2.10	Override input	145
7.9.2.11	Belt signal	147
7.9.2.12	Min. override pulse time	147
7.9.2.13	Muting status output	147
7.9.2.14	Muting lamp output	148
7.9.2.15	Muting error output	148
7.9.2.16	Release output	148
7.9.3	Information regarding cabling	149
7.9.4	State transition from stop to run	150
7.9.5	Error states and information regarding reset	150
7.9.6	Parallel muting	151
7.9.7	Sequential muting	153
7.9.8	Cross muting (one-way) – Movement direction only forwards or only backwards	155
7.9.9	Cross muting (two-way) – Material transport in both directions	157

<b>7.10</b>	<b>Function blocks for presses</b>	<b>159</b>
7.10.1	Function blocks for press contact monitoring	159
7.10.1.1	Overview and general description	159
7.10.1.2	Eccentric press contact monitor	160
7.10.1.3	Universal press contact monitor	169
7.10.2	Function blocks for press cycle control	180
7.10.2.1	Press set-up	180
7.10.2.2	Single-stroke press	186
7.10.2.3	Press automatic mode	195
7.10.2.4	Clock mode	202
<b>7.11</b>	<b>User-defined function blocks</b>	<b>216</b>
7.11.1	Grouped function block	216
7.11.2	User-defined function block	219
<b>8</b>	<b>Simulating configuration</b>	<b>223</b>
<b>8.1</b>	<b>Logic analyzer</b>	<b>225</b>
8.1.1	Starting the logic analyzer	225
8.1.2	Adding inputs and outputs	226
8.1.3	Recording data	227
8.1.4	Exporting, importing and printing recorded data	227
<b>9</b>	<b>Force mode</b>	<b>228</b>
<b>10</b>	<b>Two-channel evaluation and synchronous time</b>	<b>232</b>
<b>11</b>	<b>Transferring the system configuration</b>	<b>234</b>
11.1	Transferring project data into the safety control	234
11.2	Compatibility check	234
11.3	Verifying the configuration	235
<b>12</b>	<b>Device states in the samosPRO system</b>	<b>238</b>
12.1	Device state and LED displays in the COMPACT modules (SP-COP1, SP-COP2-ENx)	239
12.2	Device state and LED displays in the expansion modules (SP-SDIO, SP-SDI)	241
12.3	Changing the device state	242
12.4	Behavior at system start	242
<b>13</b>	<b>Technical commissioning</b>	<b>243</b>
13.1	Wiring and supply voltage	243
13.2	Transferring the configuration	243
13.3	Technical check and commissioning	243

14	Troubleshooting	244
15	Appendix	245
15.1	Complete list of error messages	245



# 1 About this manual

Please read this section carefully before you work with this manual for the software and the samosPRO system.

## 1.1 Function of this document

There are manuals with clearly delineated application areas for the samosPRO-System and operating instructions for each module and for the samosPLAN5+ software.

- The "samosPLAN5+ software" manual describes the programming of the samosPRO system when using COMPACT modules.
- Each samosPRO module includes operating instructions. These instructions provide information on the fundamental technical specifications of the modules and contain simple installation instructions. Please use the operating instructions when installing samosPRO safety controls.
- The "samosPRO hardware" manual contains an extensive description of all samosPRO modules and their functions. Use the "samosPRO hardware" manual mainly for planning samosPRO safety controls.
- The "samosPRO gateway" manual contains an extensive description of the samosPRO gateways and their functions.
- The "samosPLAN5+ software" manual contains a description of the software-supported configuration and parameterization of samosPRO safety controls. In addition, the software manual contains a description of the important diagnostic functions for operation and detailed information for identifying and eliminating errors. Use the software manual mainly when configuring, commissioning, and operating samosPRO safety controls.

## 1.2 Scope of application and version

This software manual is valid for samosPLAN5+ software versions V1.0.0 and higher and SP-COPx module versions A-01 and higher.

This version of the manual describes version V1.4.0 of the samosPLAN5+ software.

This manual contains original operating instructions in accordance with the Machinery Directive.

*Table 1: Overview of the samosPRO documentation*

Document	Title	Item number
Manual	samosPLAN5+ software	BA000968
Manual	samosPRO hardware	BA000966
Manual	samosPRO gateways	BA000970
Operating instructions	SP-COPx (COMPACT modules of the modular samosPRO safety control)	BA000978
Operating instructions	SP-SDI/SP-SDIO (Expansion modules of the modular samosPRO safety control)	BA000515
Operating instructions	SP-DIO (Expansion module of the modular samosPRO safety control)	BA001033

Document	Title	Item number
Operating instructions	SP-PROFIBUS-DP (samosPRO-PROFIBUS-DP gateway)	BA000572
Operating instructions	SP-CANopen (samosPRO-CANopen gateway)	BA000683
Operating instructions	SP-EN-ETC (samosPRO EtherCAT gateway)	BA000959

## 1.3 Target group

The samosPRO software manual is targeted toward PLAN5+ users, developers, and operators of systems into which a modular samosPRO safety control is integrated. It is also targeted toward persons commissioning a samosPRO system for the first time or maintaining such a system.

This software manual not only provides instructions for operating the machine or system into which a samosPRO safety control is integrated. There are also instructions on how to operate the machine or system.

## 1.4 Function and setup of this software manual

The software manual guides technical personnel from the machine manufacturer or machine operator regarding the software configuration, operation, and diagnostics of a samosPRO COMPACT system having the samosPLAN5+ software. It is only valid together with the "samosPRO hardware" manual (BA000966).

The basic safety information can be found here:

- Section: *Safety [ch. 2, p. 12]*
- Please ensure that you read this information.

### NOTICE

Also consult our website on the Internet at the following link:

<http://www.wieland-electric.de>

There you will find the following files available for download:

- samosPLAN5+ software
- Hardware manual (BA000966) and software manual (BA000968)
- EDS and GSD files

### 1.4.1 Recommendations for getting to know the software

We recommend the following procedure for users wishing to familiarize themselves for the first time with samosPLAN5+:

- Please read the section titled *User interface [ch. 5, p. 16]* in order to familiarize yourself with the user interface.
- Carry out the exercises for configuring example applications.

### 1.4.2 Recommendations for experienced users

We recommend the following procedure for experienced users who have already worked with samosPLAN5+:

- Please familiarize yourself with the current version of the software in the section titled *Scope of application and version [ch. 1.2, p. 9]*.
- The table of contents lists all of the functions provided by the samosPLAN5+. Please use the table of contents to find information about the basic functions.

### 1.5 Symbols/icons and writing style/spelling standard used

**NOTICE** These are notes that provide you with information regarding particularities of a device or a software function.



**ATTENTION**

#### **Warning!**

A warning lets you know about specific or potential hazards. It is intended to protect you from accidents and help prevent damage to devices and systems.

- **Please read and follow the warnings carefully!**

Failure to do so may negatively impact the safety functions and cause a hazardous state to occur.

#### **Menus and commands**

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

## 2 Safety

This section is intended to support your safety and the safety of the system users.

- ➔ Please read this section carefully before you work with a samosPRO system.

### 2.1 Qualified persons

The samosPRO system may only be installed, configured, commissioned and maintained by qualified persons. Qualified persons are those who

- have suitable technical training **and**
- have been trained by the machine operator in the operation and applicable safety guidelines **and**
- have access to the samosPRO operating instructions and have read said instructions and have duly noted these.

### 2.2 Proper use

The samosPLAN5+ software is used to configure a safety control comprising modules in the samosPRO-COMPACT system.

The samosPRO system may only be operated by qualified persons and may only be used on a machine on which the hardware and software have been installed and commissioned for the first time by a qualified person in accordance with the samosPRO operating instructions.



**In the event of any other use or any changes to the software or the devices – including within the scope of installation – this shall nullify any sort of warranty claim with respect to Wieland Electric GmbH.**

- Regard the safety information and protective measures of the samosPRO hardware and software operating instructions.
- When implementing safety-relevant control logic, ensure that the regulations of national and international standards are adhered to, particularly the control strategies and measures for reducing risk that are prescribed for your application.

**NOTICE** Please follow the standards and guidelines valid in your country when installing and using a samosPRO system.

**NOTICE** The national and international legal regulations apply to the installation and use of the samosPRO safety control as well as for the commissioning and repeated technical testing, particularly the following:

- Machinery Directive 2006/42/EC,
- EMC Directive 2014/30/EC,
- Work Equipment Directive 2009/104/EC and the supplementary directive 35/63/EC,
- Low-Voltage Directive 2014/35/EC, and
- The accident prevention regulations and safety rules.

**NOTICE** The samosPRO hardware and software operating instructions must be provided to the operator of the machine on which a samosPRO system is being used. The machine operator must be trained by a qualified person and required to read the operating instructions.

### 3 Version, compatibility, and features

There are various module versions and function packages for the samosPRO product family that enable various functions. This section will give you an overview as to which module version, which function package, and/or which version of the samosPLAN5+ you will need to be able to use a certain function or a certain device.

Table 2: Module and software versions required

Feature / functionality	Available from module version			
	SP-COP1-x	SP-COP2-EN-x	SP-COP2-ENI-x	samosPLAN5+
Safe I/O (SP-SDIO, SP-SDI)	A-01	A-01	A-01	V1.0
Modbus TCP	---	---	A-01	V1.0
Profinet IO	---	---	B-01.xx	V1.2
Non-secure I/O (SP-DIO)	C-01.xx	C-01.xx	C-01.xx	V1.3
EtherCAT (SP-EN-ETC)	C-01.xx	C-01.xx	C-01.xx	V1.3
EtherNet/IP	---	---	D-01.xx	V1.4
Press functions <sup>1)</sup>	D-01.xx	D-01.xx	D-01.xx	V1.4

<sup>1)</sup> only available with module variants **-P** (example: SP-COP2-EN-P-x)

#### Info

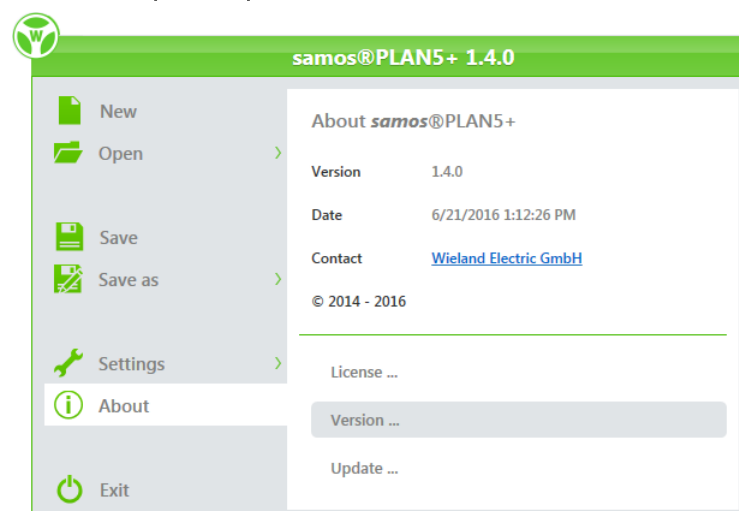
- You will find the module version on the type plate of the samosPRO modules.
- You can find the samosPLAN5+ version in the green **File menu** under **About**.
- You can obtain the latest version of the samosPLAN5+ on the Internet at <http://www.wieland-electric.de>.
- Newer modules are backwards-compatible, which means that each module can be replaced with a module having a higher module version.
- You can find the date of manufacture for a device on the type plate in the S/N field in the format <Product no.>yywwnnnnn (yy = year, ww = calendar week).

#### 3.1 Version info

All changes in the current version of samosPLAN5+ compared to previous versions can be viewed directly in the software in the **Version Information** window.

#### Activate

Main menu | About | Version



## 4 Installation and removal

### 4.1 System requirements

Recommended system configuration:

- Windows XP (32 bit/64 bit), Windows Vista (32 bit/64 bit), Windows 7 (32 bit/64 bit), Windows 8.1 (32 bit/64 bit) or Windows 10 (32 bit/64 bit)
- 2.2 GHz processor
- 3 GB RAM
- 1280 × 800 pixels screen resolution
- 150 MB available hard drive space

samosPLAN5+ is a .NET Framework application. It requires .NET Framework version 4.0 or higher (contained on the samosPRO CD-ROM) (you can find information on the current .NET Framework versions and supported operating systems on the Internet at <http://www.microsoft.com/>).

Microsoft .NET Framework version 4.0 or higher and any other required components can also be downloaded from <http://www.microsoft.com/downloads/>.

### 4.2 Installation

CD installation:

- ➔ Insert the samosPLAN5+ CD-ROM into the drive of your computer to start the installation. If the autostart function for CDs is activated on your PC, a start screen will appear after the CD has been inserted.

**NOTICE** If the autostart function for CDs has not been activated on your PC, start the installation manually by executing the setup.exe file on the CD-ROM.

- ➔ Click on **Install samosPLAN5+** and follow the additional instructions. There are two different versions of the setup file:  
If you have a 32-bit system, start the installation by double-clicking on the samosPLAN5+\_%Version%\_Setup.x86.msi file.  
If you have a 64-bit system, start the installation by double-clicking on the samosPLAN5+\_%Version%\_Setup.x64.msi file.
- ➔ Then follow the further instructions from the installation program.

### 4.3 Update

You can obtain the latest version of the samosPLAN5+ on the Internet at:  
<http://www.wieland-electric.de>

New software versions may contain new functions and support new samosPRO modules.

The removal of a previously installed, older software version is not required. However, if an installed, newer version has to be replaced by an older software version, the previously installed version must be removed beforehand.

### 4.4 Removal

The software can be removed as follows:

- ➔ In the **samosPLAN5+** program folder, in the Windows start menu, select **Remove samosPLAN5+**.

## 4.5 Troubleshooting and eliminating errors

Table 3: Error and error elimination

Error/error message	Cause	Elimination
This setup requires .NET Framework version 4.0 or higher. Please download the .Net installer from <a href="http://www.microsoft.com">http://www.microsoft.com</a>	Microsoft .NET Framework is not installed on the PC.	Install suitable version of Microsoft .NET Framework; contact your system administrator if necessary. .NET Framework is available for download on the Microsoft websites. <b>Note:</b> Install .NET Framework version 4.0 or higher.
This installer is intended for use on 64-bit operating systems. Please use the 32-bit installer from the manufacturer.	The 64-bit installer was used on a 32-bit computer.	Use the installer for the 32-bit computer.
This installer is intended for use on 32-bit operating systems. Please use the 64-bit installer from the manufacturer.	The 32-bit installer was used on a 64-bit computer.	Use the installer for the 64-bit computer.
Please remove the newer version before you install this one.	A newer version of the program has been installed on the computer.	Removal of the installed version of samosPLAN5+
This product requires Windows XP SP3 or higher.	An older Windows operating system is installed on the computer.	Please install service pack 3 for Windows XP. You can find this on the support page of Microsoft: <a href="http://www.microsoft.com">http://www.microsoft.com</a>

## 5 User interface

This section explains the most important concepts and function of the samosPLAN5+ user interface.

### 5.1 Setup and function

This section describes the basic setup and the central control concepts:

#### 5.1.1 Areas at a glance

The user interface is divided into a total of 8 areas:

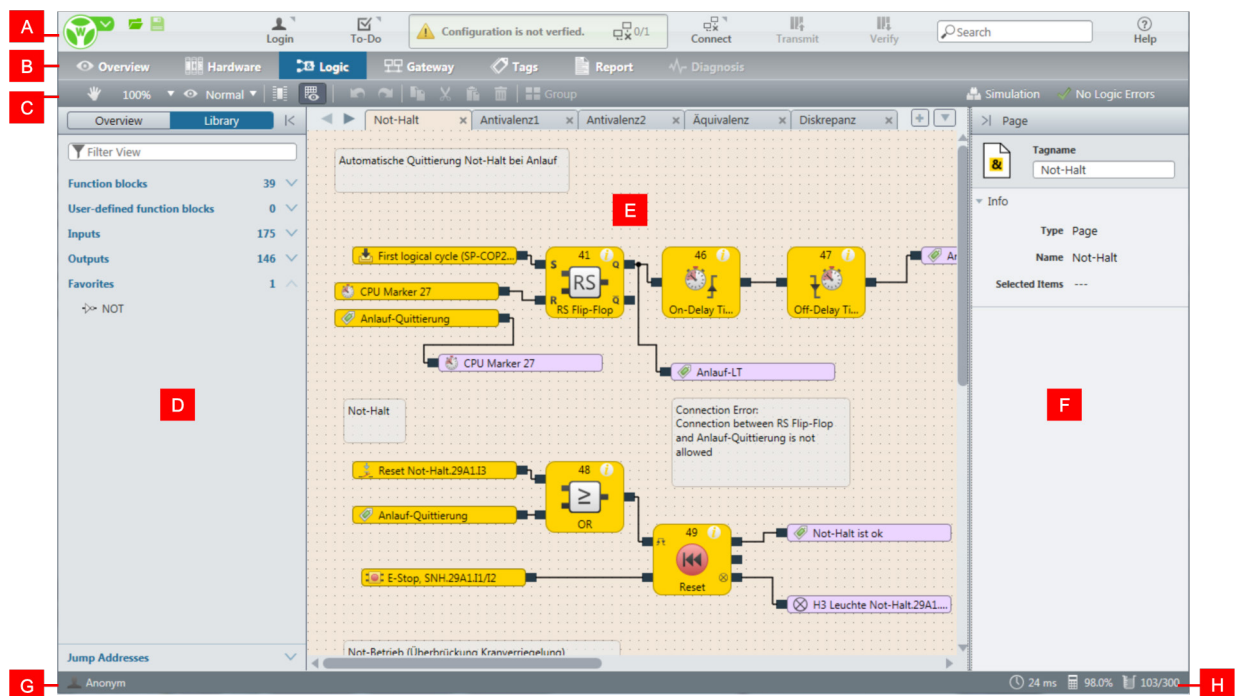





Illustration 1: User interface areas

Table 4: Key

Area	Description
A	<b>Menu bar</b> Settings and functions across projects. Setup in detail: <i>Menu bar [ch. 5.1.2, p. 17]</i>
B	<b>Tab bar</b> Toggles between views.
C	<b>Command bar</b> Depends on the selected view: Available commands
D	<b>Left sidebar</b> Depends on the selected view: Hierarchical tree structure with selection of the project components available or used. Further information: <i>Window behavior of sidebars [ch. 5.1.3, p. 19]</i>
E	<b>Work area</b> Depends on the currently selected view: Graphical representation and configuration of the project content.















Area	Description	
F	<p><b>Right sidebar</b></p> <p>Configuration dialog or display of the properties for a selected element in the work area.</p> <p>Further information: <i>Window behavior of sidebars [ch. 5.1.3, p. 19]</i></p>	
G	<p><b>Status bar on left</b></p> <p>Displays the user currently logged on</p>	
H	<p><b>Status bar on right</b></p> <p>Central status data on your samosPLAN5+ project:</p>	
	 24 ms	CPU cycle time resulting from your logic programming
	 0,0%	CPU usage as a percent
	 0/300	Number of function blocks used

## 5.1.2 Menu bar

The menu bar contains the following commands and functions:

Table 5: Reference

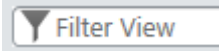








Element	Description
	<p><b>Main menu:</b></p> <p>This menu contains the basic functions and basic settings across projects.</p> <p>Setup in detail: <i>Cross-project settings (main menu) [ch. 5.2, p. 24]</i></p>
	<p>This enables quick access to the following commands:</p> <p><b>Open project</b> and <b>Save project</b></p>
	<p><b>User menu:</b></p> <p>This enables logon with user and password.</p> <p>Further information: <i>User administration [ch. 5.3.4, p. 35]</i></p>
	<p><b>To-do menu</b></p> <p>This enables you to manually document the progress of the project.</p> <p><b>Please note:</b> This is not an automatic display. The presence or absence of checkmarks has no effect on system behavior.</p>

Element	Description
 Configuration is not verified.	<p><b>Status display/text message</b></p> <p>This indicates the verification status and the connection status of a project.</p> <ul style="list-style-type: none"> <li>• <b>Configuration has errors:</b> There are one or more logic components in the logic at which not all inputs are connected (connection error). Further information: <i>Automatic logic check [ch. 5.5.4, p. 55]</i></li> <li>• <b>Configuration is not verified:</b> There is no connection error in the configuration. However, the process of verification was not completed successfully. Further information: <i>Verifying the configuration [ch. 11.3, p. 235]</i></li> <li>• <b>Configuration verified:</b> The process of verification has been completed successfully.</li> </ul>
	<p><b>Status display/connection status</b></p> <p>This indicates whether samosPLAN5+ is connected to a safety control.</p>
 Connect	<p><b>Connect menu</b></p> <p>This opens a menu that you can use to connect to a safety control. Further information: <i>Connecting to the samosPRO system [ch. 6, p. 68]</i></p>
 Transmit	<p><b>Send</b></p> <p>This sends logic programming to the currently connected safety control.</p>
 Verify	<p><b>Verify</b></p> <p>This starts the verification of your logic programming. Further information: <i>Verifying the configuration [ch. 11.3, p. 235]</i></p>
 Falsify	<p><b>Falsify</b></p> <p>Cancels a verification of the currently connected safety controller.</p>
 Search	<p><b>Find</b></p> <p>This will initiate a search of the display names of all of the project components according to the character string entered.</p> <ul style="list-style-type: none"> <li>• The hit list will show all of the components whose display name contains the character string.</li> <li>• When you click on a hit, the configuration dialog for the selected object will open in the right-hand sidebar.</li> </ul>
 Help	<p><b>Help</b></p> <p>Provides you with direct access to the support services of WIELAND:</p> <ul style="list-style-type: none"> <li>• Contact sensitive help that opens directly from the samosPLAN interface (alternative: &lt;F1&gt; key).</li> <li>• HTML help with full contents of the samosPRO manuals for <b>software, hardware and gateway</b>.</li> <li>• Support request in the form of an automatically generated e-mail. Further information: <i>Function description [ch. 5.8.2, p. 63]</i></li> </ul>

## 5.1.3 Window behavior of sidebars

You can set the following view options for the left and right sidebars.

Table 6: View options for the left and right sidebars

Target	Procedure
Filtering the content of tree view	<ul style="list-style-type: none"> <li>Enter a character string into the input field.   </li> <li>The content of the tree view will be reduced to the entries containing this character string. The content of the work area will not be influenced by this filter function.</li> <li>In order to cancel the filter, delete the character string so that the input field is once again empty.</li> </ul>
Changing the width	<ul style="list-style-type: none"> <li>Place the mouse on the inner edge of the sidebar.</li> <li>Once the double arrow appears, drag the edge to the left or right until the sidebar reaches the desired width.</li> </ul>
Hide/minimize	<p>Click on the <b>Hide</b> icon in the header of the sidebar.</p> <ul style="list-style-type: none"> <li>Left sidebar:  </li> <li>Right sidebar:  </li> </ul>
Display once	<p>Once the sidebar is hidden: Click on the arrow icon at the edge of the screen.</p> <ul style="list-style-type: none"> <li>Left sidebar:  </li> <li>Right sidebar:  </li> </ul>
Continuous display	<p>Once the sidebar is hidden:</p> <ul style="list-style-type: none"> <li>Click on the arrow icon at the edge of the screen. <ul style="list-style-type: none"> <li>Left sidebar:  </li> <li>Right sidebar:  </li> </ul> </li> <li>Click on the <b>Display</b> icon in the header of the sidebar. <ul style="list-style-type: none"> <li>Left sidebar:  </li> <li>Right sidebar:  </li> </ul> </li> </ul>

## 5.1.4 Commands

You can activate commands in three different ways:

- Mouse click on buttons
- Commands in the context menu
- Activate via keyboard

### Frequently used commands

The following commands are effective in all samosPLAN5+ views:

*Table 7: List of frequently used commands*

Command	Activate
Save project	<ul style="list-style-type: none"> <li>• Menu bar: <b>Save button</b></li> <li>• Keyboard: <b>&lt;Ctrl&gt; + &lt;S&gt;</b></li> </ul>
Create new project	<ul style="list-style-type: none"> <li>• <b>Settings menu: New</b></li> <li>• Keyboard: <b>&lt;Ctrl&gt; + &lt;N&gt;</b></li> </ul>
Undo last action	<ul style="list-style-type: none"> <li>• Command bar: <b>Undo button</b></li> <li>• Keyboard: <b>&lt;Ctrl&gt; + &lt;Z&gt;</b></li> </ul>
Redoing an action that was undone	<ul style="list-style-type: none"> <li>• Command bar: <b>Redo button</b></li> <li>• Keyboard: <b>&lt;Ctrl&gt; + &lt;Y&gt;</b></li> </ul>
Delete a selection in the work area	<ul style="list-style-type: none"> <li>• Context menu: <b>Delete</b></li> <li>• Keyboard: <b>&lt;Del&gt;</b></li> </ul>
Change the size of the display in the work area	<ul style="list-style-type: none"> <li>• Keyboard: <b>&lt;Ctrl&gt; + &lt;mouse wheel&gt;</b></li> </ul>
Find	<ul style="list-style-type: none"> <li>• Menu bar: <b>Find input field</b></li> <li>• Keyboard: <b>&lt;Ctrl&gt; + &lt;F&gt;</b></li> </ul>
Open help	<ul style="list-style-type: none"> <li>• Menu bar: <b>? Icon</b></li> <li>• Keyboard: <b>&lt;F1&gt;</b></li> </ul>

## 5.1.5 Favorites for hardware and logic

You can create favorites for frequently used elements in the **Hardware** and **Logic** views:

### Procedure

- ➔ Click on the desired element with the right mouse key.  
The context menu will open.
- ➔ Select **Add to favorites** in the context menu.

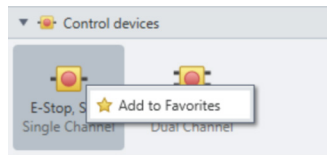


Illustration 2: Defining favorites via the context menu

You can then activate the element from the left sidebar at:

### Library | Favorites

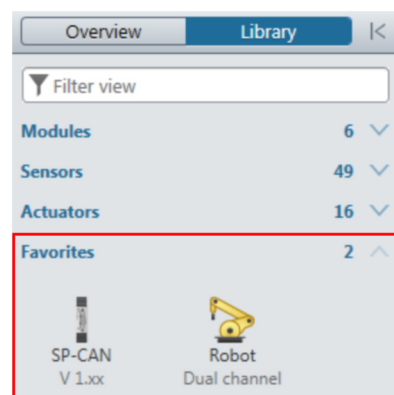


Illustration 3: Location of favorites in the left sidebar

## 5.1.6 Overview safety and security mechanism

As software for configuring and programming safety controls, samosPRO COMPACT fulfills the pertinent requirements for safety products (e.g. normative requirements of IEC 61508).

### Functional safety

In the area of functional safety (**safety** aspect), the following samosPRO COMPACT mechanisms apply:

- Automatic review of the logic configuration for connection errors  
Additional information: *Automatic logic check [ch. 5.5.4, p. 55]*
- Warning in the event of test pulse deactivation  
Additional information: *Parameterizing actuators and sensors [ch. 5.4.4, p. 42]*
- Blocking of functions when safety-relevant preconditions are not fulfilled  
Example: Verification is not possible until there are no more connection errors
- Default value ranges for configuration parameters
- Automatic calculation of the required CPU cycle time for the entire project (status bar on right)  
This means that you always can see the effects that your logic programming are having on the CPU cycle time.
- Checksums (CRC) for central safety-relevant project components:
  - CRC for the report
  - CRC for user-defined logic components

## Access security

With respect to security, samosPLAN5+ additionally provides protection of project data regarding the following:

- User administration that you can scale the access options to project content systematically.

Important information in this context:

- When you start working with samosPLAN5+, define which user groups you wish to set up with which access rights.
- Change the default password for the previously set up user groups.

Further information: *User administration [ch. 5.3.4, p. 35]*

- Password protection for user-defined libraries.  
You can precisely determine who can view or change the components you have developed yourself.
- Encryption of project files  
Project files cannot be read or evaluated without the samosPLAN5+ software. Those who do not have the appropriate user rights or password also cannot open project files, even with samosPLAN5+.
- Password-protected connection to the samosPRO hardware  
Additional information: *Connecting to the samosPRO system [ch. 6, p. 68]*

## 5.1.7 Program help

The program help supports you in working with samosPRO. Here, you will quickly find additional information, such as safety instructions, handling instructions, module descriptions and summary tables.

### Open help

You open program help with the **F1** key on your keyboard. The help window opens and displays the information on the selected object:

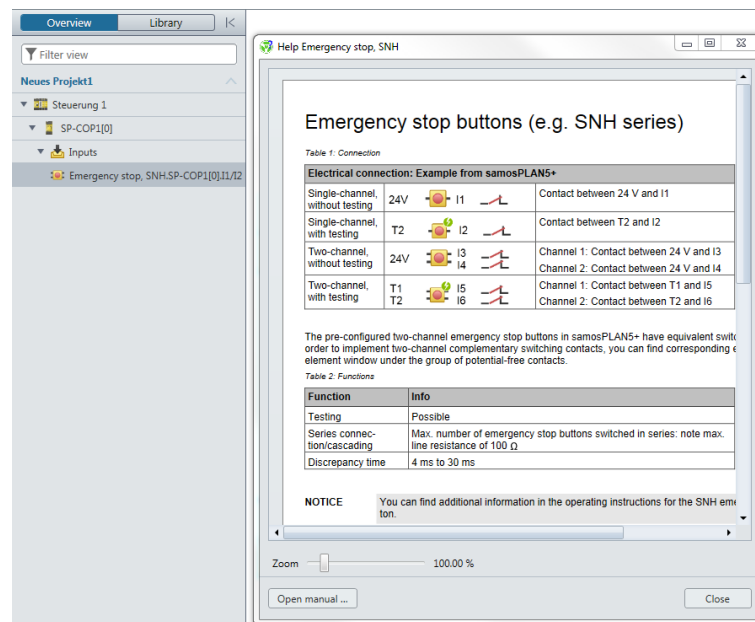


Illustration 4: Program help on the selected "Emergency stop button" object

### **Additional information in the HTML help**

If you have opened the help, you are forwarded to the online help by clicking on **To the Manual...** It opens in your browser. Here, you will find the complete manuals for the samosPRO COMPACT system:

- samosPRO hardware (BA000966)
- samosPLAN5+ software (BA000968)
- samosPRO gateways (BA000970)

## 5.2 Cross-project settings (main menu)

The **Settings** menu provides you with access to cross-project commands and functions.

### Overview

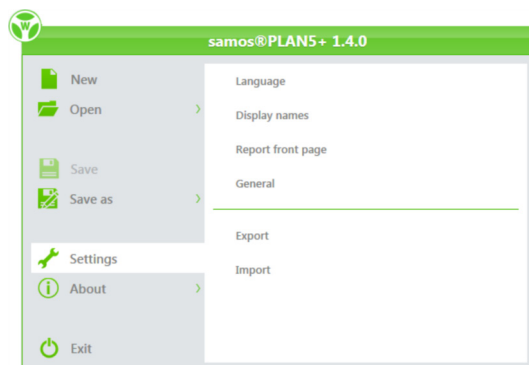


Illustration 5: "Settings" menu

Table 8: Key

Element	Function
New	This closes the currently open project (with a prior save prompt) and opens a new empty project.
Open	<ul style="list-style-type: none"> <li>When selecting without mouse click: Shows a list of available projects in the right menu area that you can open with a mouse click.</li> <li>When selecting with mouse click: Opens Windows Explorer. From there you can look for a previously existing project and open it.</li> </ul>
Save	Saves the current status of the open project.
Save as	<p>This opens an Explorer window that you can use to save the currently open project under a new name or at a new target location.</p> <p>You can choose from two save options:</p> <ul style="list-style-type: none"> <li>as a project file (file format *.SPF)</li> <li>as a control project file (file format *.XML)</li> </ul>
Settings	<p>This contains cross-project basic settings that you can make for samosPLAN5+.</p> <p><b>IMPORTANT:</b> Any changes made to the settings will only affect the currently logged-in Windows user. You can <i>export and import</i> [ch. 5.4.6, p. 46] the settings and thus make them available to other users or computers.</p> <p>You can make the following settings:</p> <ul style="list-style-type: none"> <li>User interface language [ch. 5.2.1, p. 25]</li> <li>Configuration of display names [ch. 5.2.2, p. 26]</li> <li>Edit report front page [ch. 5.8.3, p. 64]</li> <li>General automation [ch. 5.2.4, p. 28], display of start view [ch. 5.2.3, p. 27], Specifications for the logic editor [ch. 5.2.5, p. 29]</li> </ul>
About	<p>This opens a menu in the right area of the window containing the following:</p> <ul style="list-style-type: none"> <li>Version information</li> <li>License information</li> <li>Option of a software update via the Internet</li> </ul>
End	This closes the project and the software after a save prompt.



## 5.2.1 User interface language

You can choose from among six different languages for the user interface.

### Activate

Main menu | Settings | Language

### Overview

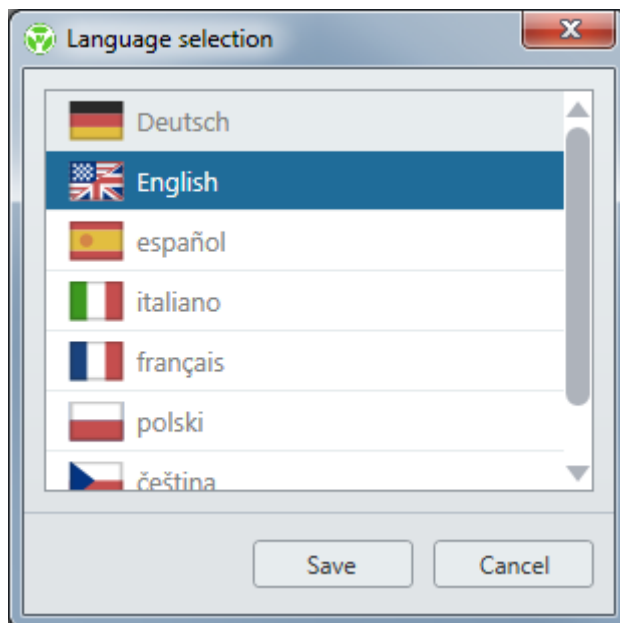


Illustration 6: Dialog window for language selection

## 5.2.2 Configuration of display names

You can define a naming scheme for the display names for modules, sensors, actuators and gateways. The display name is the designation with which the project components are displayed on the samosPLAN5+ interface.

### Activate

Main menu | Settings | Display names

### Overview of functions

The configuration window consists of two areas:

- Left: Configuration for modules, sensors and actuators
- Right: Configuration for gateways

You define the relevant naming scheme in the input field with a white background. You thus define the components that make up the display name. You can obtain the correct syntax for the components by clicking on the corresponding button above the input field.

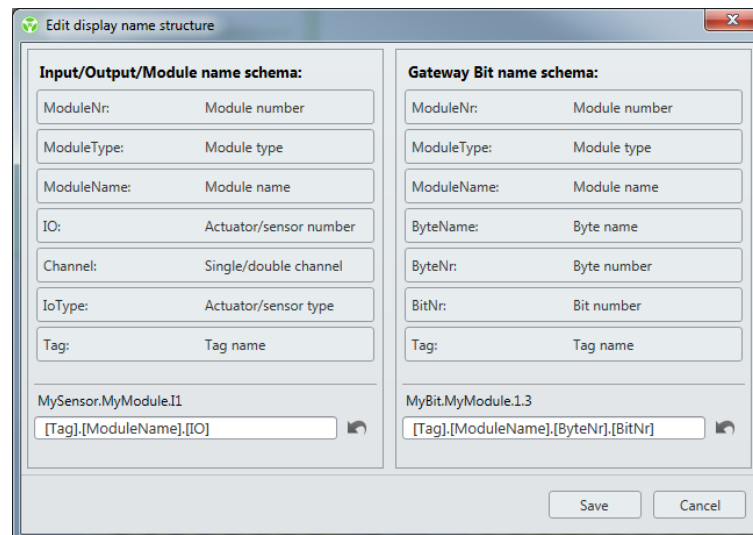


Illustration 7: Dialog window for configuring display names

### NOTICE "Tag" name element

The **Tag** name element allows you to integrate a user-defined character string in the display name. You can define which character string is used for the **Tag** name element at two locations:

- The corresponding project component in the configuration window. You will find the configuration dialog in the **Hardware** view and in the **Logic** view.
- In the **Name** view: *Assigning Tag names/changing display names (exercise) [ch. 5.7.2, p. 61]*

## 5.2.3 Displaying start view

After the start of the software, a start view will appear by default. From there you can select which of the following actions you wish to begin your work with in samosPLAN5+.

- Create a new empty project.
- Select one of the recently edited projects from a list.
- Select a freely selected project (file format \*.SPF) in Windows Explorer and open.
- Deactivate the start view.

### Overview

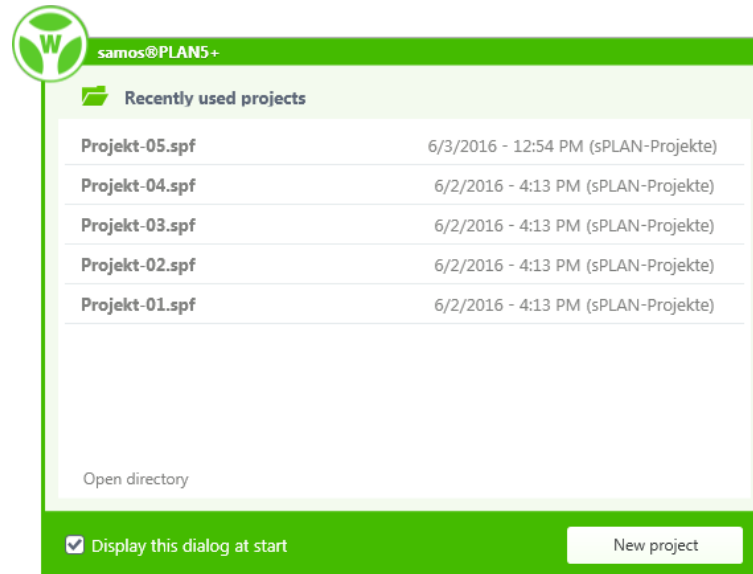


Illustration 8: Start view with list of the recently edited projects

### Reactivating start view

Once you have deactivated start view, you can reactivate it here:

- ➔ Open the main menu.
- ➔ Click on **Settings | General**.
- ➔ Activate the **Display project selection dialog at start** checkbox.

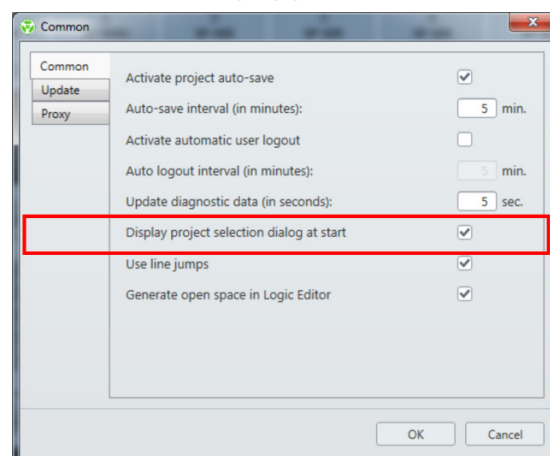


Illustration 9: Checkbox for activating start view

## 5.2.4 Instructions for automating (saving, logging off, updating)

samosPLAN5+ offers you the following automatic features in the program characteristics:

- Automatic saving of the project file
- Automatic log-off
- Updating of the diagnostic data

You can make the settings for this here:

### Activate

Main menu | Settings | General | General tab

### Overview

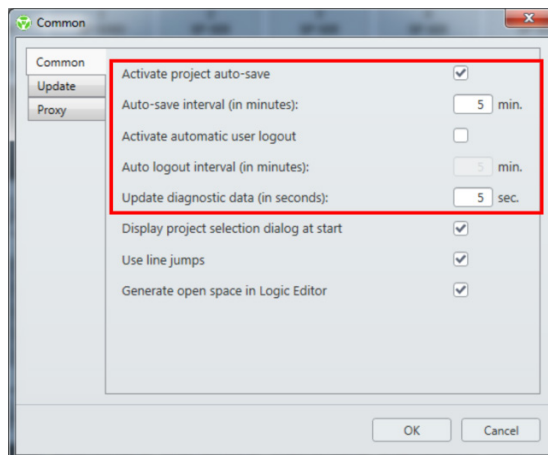


Illustration 10: Setting automatic features in the "General" dialog

## 5.2.5 Specifications for the logic editor

Using the two options below, you can configure the behavior of the logic editor:

### Activate

Main menu | Settings | General | General tab

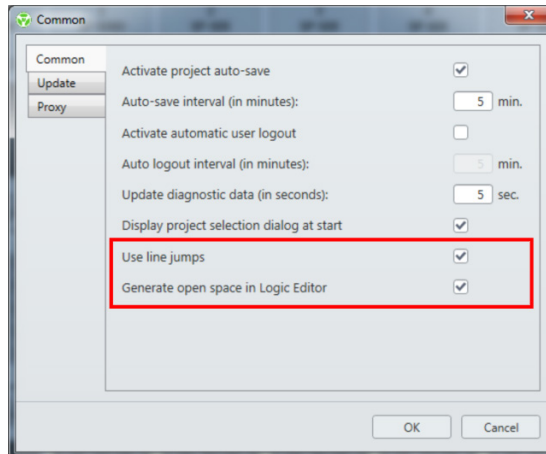


Illustration 11: Specifications for the logic editor

Table 9: Reference

Element	Function
Uses line jumps	<p>Defines line jumps for the intersecting lines displayed in the logic editor.</p> <p><b>Example</b></p> <p>The diagram shows two yellow input boxes: 'Acknowledgement of start-...' and 'Input status (25A1)'. Their lines connect to an OR gate (yellow box labeled '49 OR'). The lines use line jumps to connect to the gate inputs without crossing.</p>
Generate open space in the Logic Editor	<p>Defines that open space is generated around the new object when inserting a new object (if possible).</p> <p>The available project components move to the side or down.</p>

## 5.2.6 Updates

In the **General** window you can specify whether and when samosPLAN5+ should check for new versions of the program.

### Activating the function

Main menu | Settings | General | Update tab

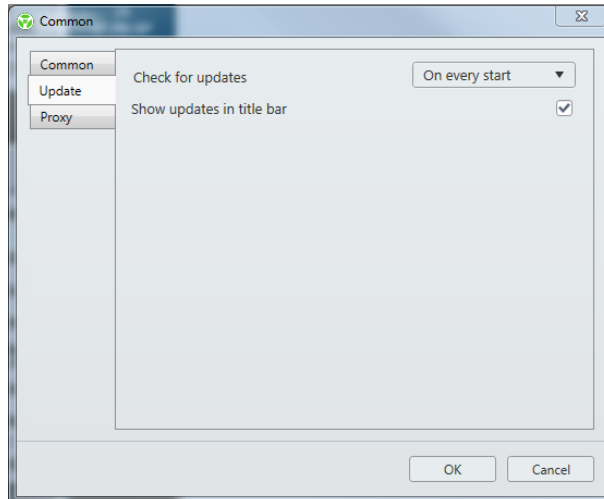


Illustration 12: Update settings

## 5.2.7 Proxy settings

You can define a proxy server for use with samosPLAN5+.

### Activating the function

Main menu | Settings | General | Proxy tab

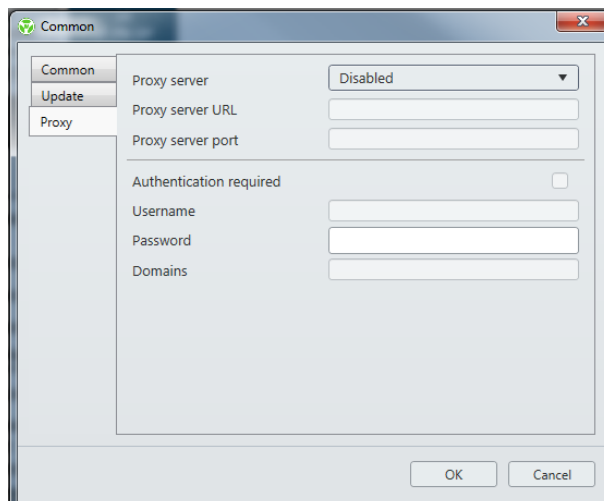


Illustration 13: Proxy settings

## 5.2.8 Importing/Exporting settings

You can save and transfer user settings that you have made in the samosPLAN5+ menu in file format via an export/import interface.

### Function

- What is saved?
  - Language setting
  - Naming schema for display names
  - Settings from the settings dialog
    - Exception:** Password for the proxy server, if defined
- Saving format (file format):
  - \*.SPS**

### Activating the function

Main Menu | Settings | Export / Import

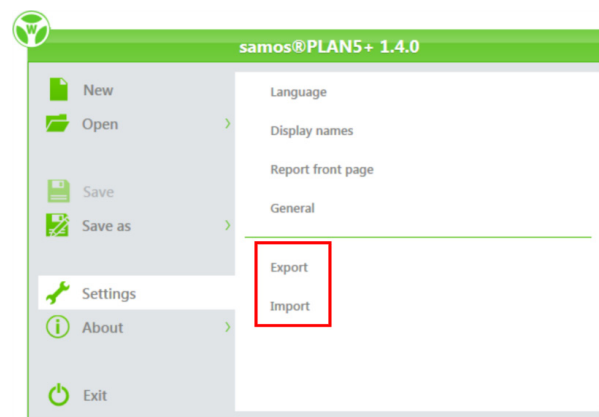


Illustration 14: Export and import function in the "Settings" menu

## 5.3 "Overview" view

Normally, you only work in **Overview** view when you are creating a new project or wish to change superordinate properties of an existing project.

### 5.3.1 Setup and functions

#### Work area

The work area in the **Overview** view shows the hardware configuration of the currently open project and selected status information.

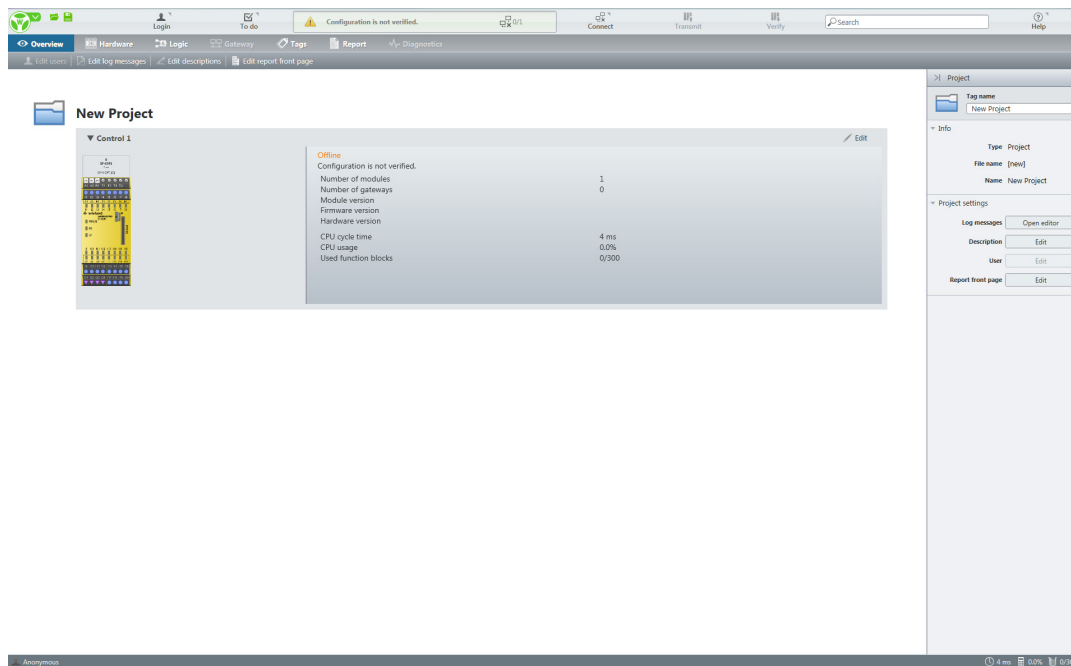


Illustration 15: Display of project content in the "Overview" view

If you create a new project and still have not defined any hardware, the work area will be empty.

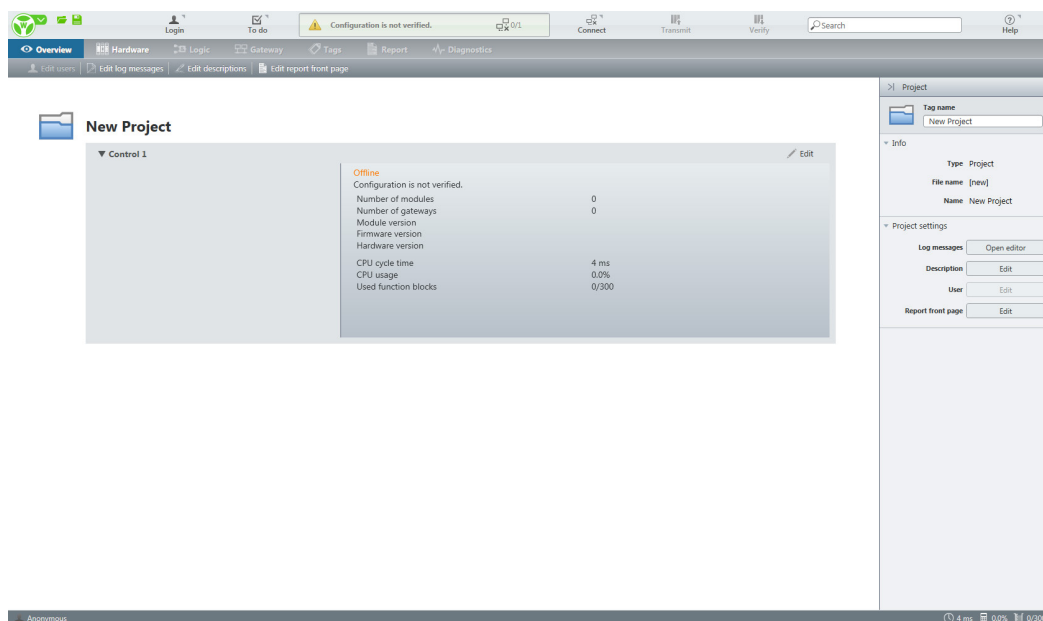


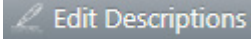
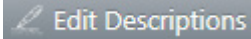


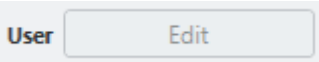
Illustration 16: New empty project in the "Overview" view



## Functions and commands

The following functions are available to you in the command bar and in the right sidebar:

Table 10: Reference

Function	Command bar	Right sidebar
Assign or change project name		
Create or change project description		
Add notes regarding the project (display via project description)		
Define log messages Details: <i>Log messages [ch. 5.3.3, p. 34]</i>		
Manage users Only active if you have the corresponding user rights. Details: <i>User administration [ch. 5.3.4, p. 35]</i>		

### 5.3.2 Creating a new project (exercise)

You can create a new project in just 3 steps:

#### Procedure

- ➔ Click on **New** in the **samosPLAN5+** menu.
  - ⇒ A new empty project will be created.
  - ⇒ Except for the **Overview** and **Hardware** views, all of the other views will be grayed out and unavailable.
- ➔ Switch to the **Overview** view, and assign a name for the project in the right sidebar.
- ➔ If necessary, store the following information:
  - A project description (**Edit** button)
  - Log messages (**Open Editor** button)

## 5.3.3 Log messages

samosPLAN5+ will provide you with a set of 64 log messages, which you can freely define. The messages apply to 1 project. If you wish to use a set of messages in other projects, use the export/import function for transferring.

### Editor for log messages

You can manage the log messages in their own editor.

Activate:

- **Overview** | Command line **Edit log messages**
- **Overview** | Right sidebar | **Open editor**

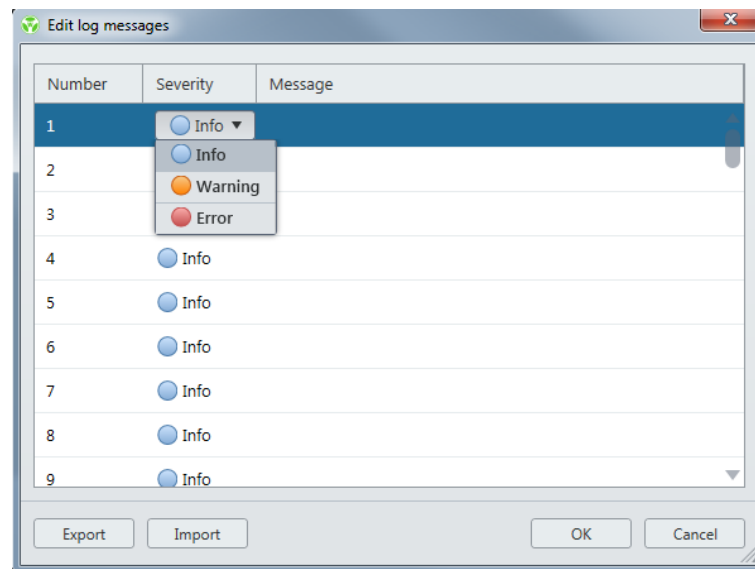


Illustration 17: Editor for log messages

Two values must be set for each log message:

- Severity level (selection list in the table column of the same name)
- Message text in the **Message** column

### Exporting and importing log messages

Within the editor, click on the corresponding buttons in order to export or import log messages.

- File format: **\*.CSV**

### Integrating into the "Log generator" function block

In the **Logic** view, you can reference the log messages in function blocks of the **Log generator** type.

To do this, select the number of the desired log message in the configuration dialog of the component (right sidebar) in the **Inputs 1** selection list.

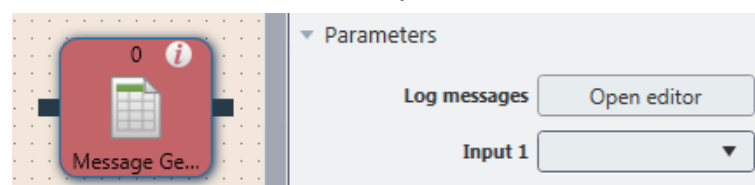


Illustration 18: Referencing log messages

## 5.3.4 User administration

You can centrally control access and editing rights via user administration.

### Scope of application of access rights

The access rights only apply to the respective project.

### Requirement

In order to activate user administration, you have to be registered in a user function that has the following authorization: **May edit users**

Default setting of samosPLAN5+:

- By default, only the **Authorized client** user group may activate user administration.
- Default password for all user groups (change under all circumstances): **SAMOSPRO**

### Activate

#### Overview view | Edit user command

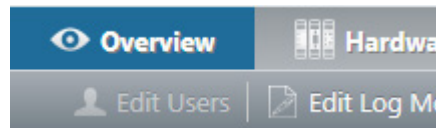


Illustration 19: Activates user administration

### Setup and function overview

In user administration you can do the following: Change existing user rights, create new users, and copy. In addition, you have the option of exporting and importing users.

You can obtain the rights concept of samosPLAN5+ in the right sidebar.

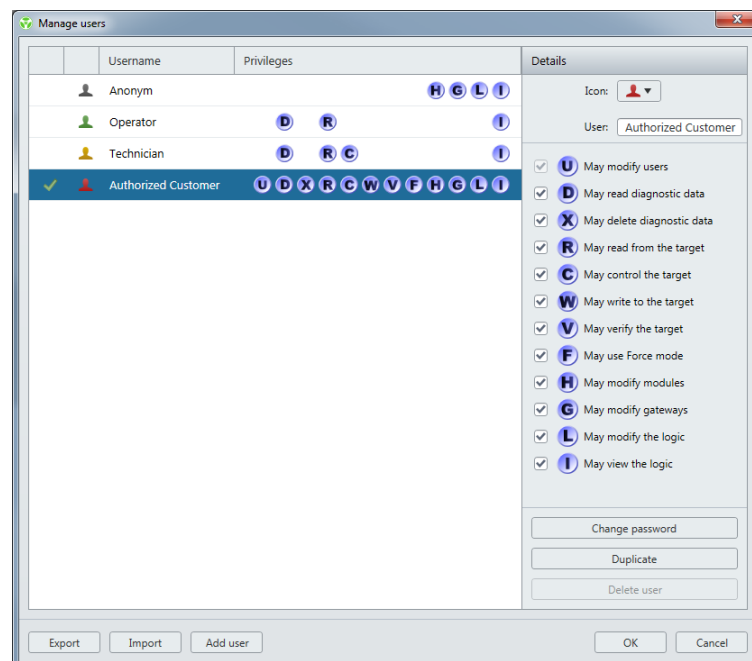


Illustration 20: Rights concept upon delivery

## 5.4 "Hardware" view

In **Hardware** view, you can combine hardware components into a safety control. You can name modules and elements, give them tag names, and carry out parameterization.

### 5.4.1 Setup and functions

#### Work area

You can combine the available elements into a safety control in the work area. All of the connected devices are graphically depicted in the work area.

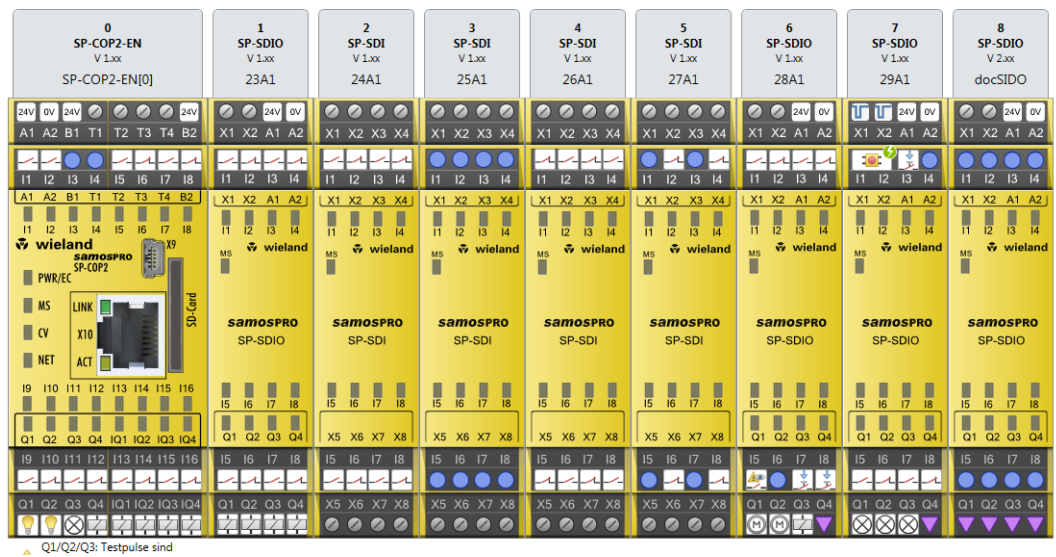


Illustration 21: Display in the work area

#### Left sidebar | Library

You can find the devices used in the **Library** tab in the left sidebar.

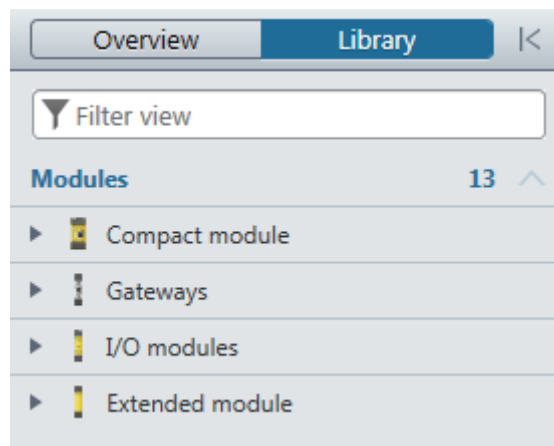


Illustration 22: Library in the "Hardware" view

Using the drag-and-drop function, you can place modules in the work area and also drag sensors and actuators and place them onto available inputs and outputs of the module.

You can define your own hardware elements (**Library** tab) and specify favorites.

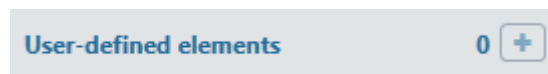


Illustration 23: Location of user-defined elements

## Left sidebar | Overview

In the **Overview** tab of the left sidebar, you can see all of the project components used as a hierarchical tree structure.

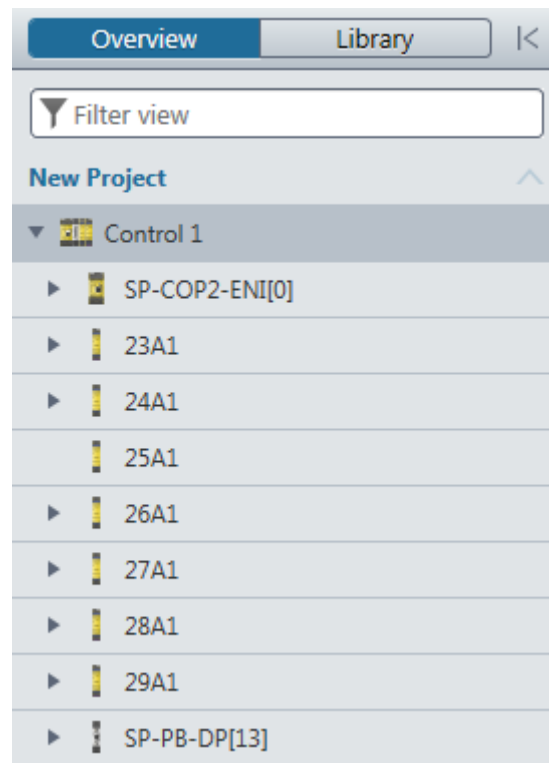


Illustration 24: Overview of the left sidebar

## Right sidebar

The configuration dialog for the currently selected element will appear in the right sidebar.

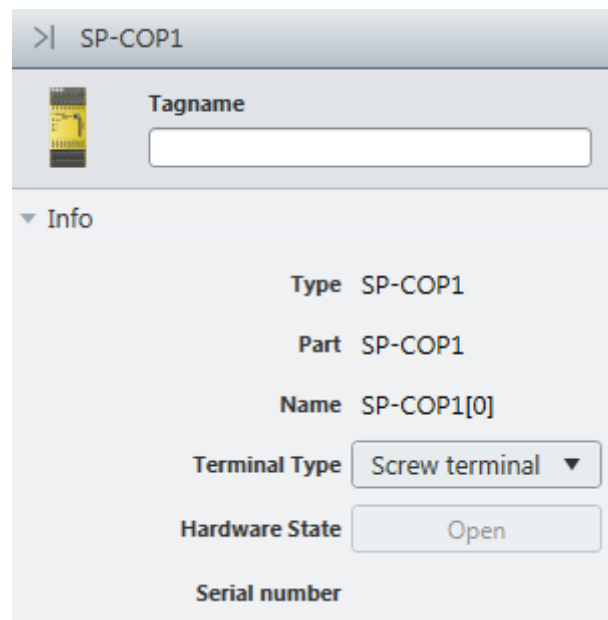


Illustration 25: Configuration dialog in the right sidebar

## Command bar

Via the command bar, you have access to the following view-specific functions:

Table 11: Key

Element	Description
	<b>Zoom</b> This determines the size of the display in the work area.
	<b>Start</b> If you are connected to a safety control: This starts or stops the processing of the control logic.
	<b>Undo</b> This renders the last action undone.
	<b>Redo</b> This makes an action that has been undone redone.
	<b>Delete</b> This deletes the currently selected element.

## Context menu

You can change the module type for main modules (modules of type **SP-COPx**).

To do this, right-click on the desired module in the work area and select the **Change module type** option.

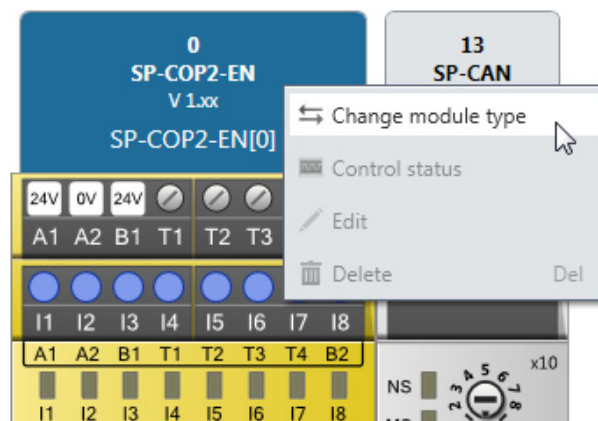


Illustration 26: Changing the module type using the context menu

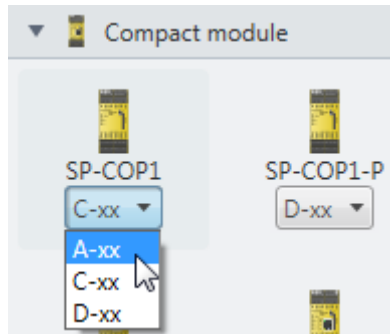
## 5.4.2 Selecting and parameterizing modules (exercise)

### Requirement

- You have *created* [ch. 5.3.2, p. 33] and opened a project.

### Step 1: Drag modules around in the work area and sort them

- ➔ Switch into **Hardware | Library | Modules**.  
All of the modules except for the compact modules will be grayed out.
- ➔ Click the compact module you want to use and select the desired module version in the selection list.



- ➔ Using the mouse, drag the compact module into the work area.
  - ⇒ The other modules (e.g. under gateways) will be active and can be selected.
- ➔ Drag the desired additional modules into the work area.
  - ⇒ The modules will be automatically added in the correct sequence.  
The controller module will be located on the far left. Up to two gateways will follow directly to the right next to the controller module. The expansion modules will not come until afterwards. The relay modules will be positioned all the way to the right.
  - ⇒ Using the drag-and-drop function, you can change the sequence of the modules within these groupings.
- ➔ Assign a module name for each module.  
To do this, click on the module and enter the name in the right sidebar under **Tag name**.
- ➔ If you wish to delete a module:
  - Click on it and press the **<Del>** button.
  - Or delete the module via the context menu.

### Step 2: Parameterizing modules

For the options available for configuration of the modules, see here:  
"samosPRO hardware" manual (BA000966)

## 5.4.2.1 Special case: SP-XX expansion module

If you want to use an **SP-XX** expansion module, proceed as follows:

### Procedure

- ➔ Select a compact module and drag it into the work area.  
See: *Selecting and parameterizing modules (exercise)* [ch. 5.4.2, p. 39]
- ➔ Drag the **SP-XX** expansion module into the work area.
- ➔ In the configuration dialog (right sidebar) click on the following button:



⇒ The **Available elements** selection dialog opens.



- ➔ Select the desired module image for the **SP-XX** expansion module.

### NOTICE Use your own module images

You can create your own module images for the **SP-XX** expansion module and use them in samosPLAN5+. To do this, proceed as follows:

- Create an image with the dimensions 1600 x 384 pixels (height x width).
- Insert the image into the selection dialog with the **Select module image** button.



## 5.4.3 Placing actuators and sensors (exercise)

Using the drag-and-drop function, you can drag the actuators and sensors and place them onto available outputs and inputs of modules.

### Procedure

- ➔ Switch to **Hardware | Library | Sensors or Actuators**.
- ➔ Using the mouse, drag the desired elements (e.g. **Emergency stop**) to a suitable available output or input. Suitable outputs and inputs are indicated by a green frame during the drag-and-drop process.

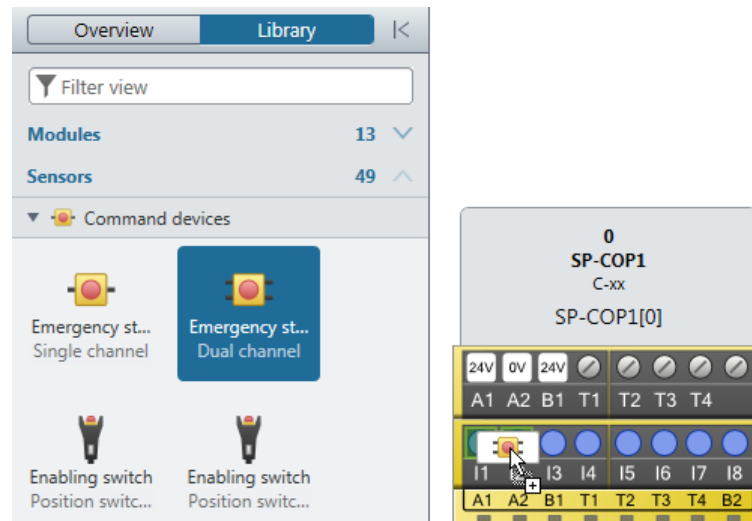


Illustration 27: Drag an element onto inputs and outputs

- ➔ Click on the particular actuator or sensor.
  - ⇒ The configuration dialog will open in the right sidebar.

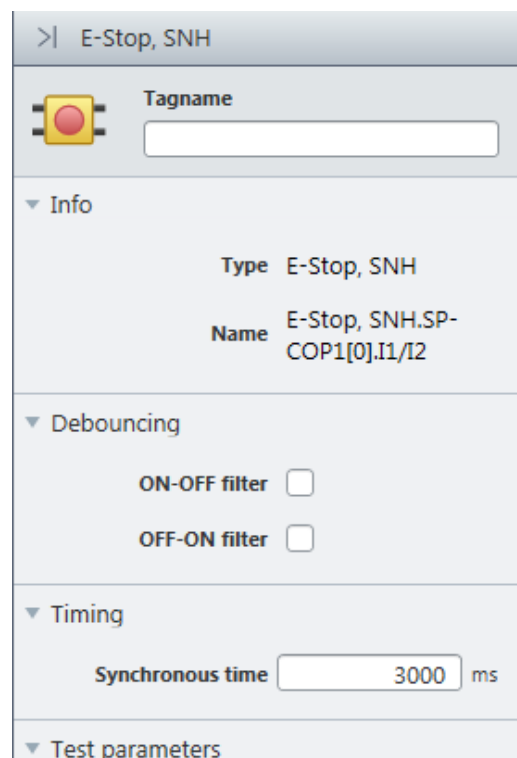


Illustration 28: Configuration dialog for inputs and outputs

- ➔ Make the desired settings in the configuration dialog (right sidebar). Parameter reference: *Parameterizing actuators and sensors [ch. 5.4.4, p. 42]*

## 5.4.4 Parameterizing actuators and sensors (parameter reference)

The following parameters are available to you in actuators and sensors. Depending on the type of element, they will vary in scope and selection:

### Tag name

If you do not assign your own tag name, the default tag name will be used.

### Number of devices

If you have connected, for example, a cascade of several testable type-2 L21 sensors at an input, you can use this function in order to set the number of devices that will appear in the quantity in the report so that it corresponds to the actual number of devices used.

### Synchronous time

Two-channel elements can be selected with or without synchronous time. Synchronous time determines how long the two inputs will be allowed to have discrepant values after one of the two input signals has changed its value without this being evaluated as an error.

Detailed information on synchronous time monitoring by the I/O modules: *Two-channel evaluation and synchronous time [ch. 10, p. 232]*

### Procedure

- ➔ Enter a value between 0 and 30000.

### Info

For elements that are connected to SP-SDI and SP-SDIO modules, the following limitations apply:

The value for synchronous time can be set to 0 = deactivated or to a value of 4 ms to 30 s. Due to the internal evaluation frequency of the modules, the value is automatically rounded up to the next-higher multiple of 4 ms.

If signals of tested sensors are connected to SP-SDI or SP-SDIO modules, then the synchronous time must be greater than the test gap + the max. Off-On delay of the test output used. You can find these values in the project report under **Configuration, I/O module, Test pulse parameter**.

If you attempt to set a synchronous time that is less than permitted, the minimum value will be displayed in the dialog window.

### On-Off filter or Off-On filter

Upon the opening or closing of a contact-based component, multiple short signal changes will undesirably result due to the bounce of the contacts. Because this can 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) in order to eliminate this effect.

### Procedure

- ➔ Activate or deactivate the corresponding checkboxes.

### Info

- Once the **On-Off** filter or the **Off-On** filter is active, then a change in the signal is only recognized as such when it is confirmed by three identical evaluations of the input directly after one another with an evaluation frequency of 4 ms, i.e. when the signal is constantly present for 8-12 ms.
- When there are two-channel elements with a discrepancy evaluation, the respective filter (On-Off or Off-On) always relates to the lead channel. The filter for the complementary channel is automatically active.



### Be aware of the extended response times if you use the input filter!

- Due to the internal evaluation frequency of the modules of 4 ms, the On-Off filter and the Off-On filter will extend the response time by at least 8 ms.
- If the signal changes within this initial 8 ms, then the signal change can be delayed significantly longer, i.e. until a constant signal of at least 8 ms has been detected.

### Using test outputs

By activating or deactivating the **Use test outputs** option, you can determine whether the respective element will be tested or not. By connecting an element to the test outputs ...

- short-circuits in the sensor cabling downstream of the 24 V that could prevent the switch-off condition can be detected
- electronic sensors with test inputs (e.g. L21) can be tested.

### Procedure

➔ Activate or deactivate the corresponding checkboxes.

### Note

An SP-SDI only has two test sources even if it has more than eight test output terminals.

### Safety information



#### Protect single-channel inputs against short-circuits and cross-connections!

When a short-circuit to high occurs at a single-channel input with test pulses that were previously low, this signal can then look like a pulse for the logic. A short-circuit to high means that the signal is first to high and then is back to low after the error detection time. A pulse can be generated due to the error detection.

Because of this, note the following specifications for single-channel signals with test pulses:

- If the short-circuit to high occurs at a single-channel input with test pulses that was previously high, this signal for the logic then looks like a delayed falling edge (transition from high to low).
- When a single-channel input is used and an unexpected pulse or a delayed falling edge (high to low) at this input can lead to a state causing a risk, then you must undertake the following measures:
  - Protected cabling for the signal in question (in order to prevent cross-connections with other signals)
  - No cross-connection detection, i.e. no connection with a test output**This must be noted in particular for the following inputs:**
  - Input reset at the function block reset
  - Input restart at the function block restart
  - Input restart at the function blocks for press applications (eccentric press contact monitor, contact monitor for universal presses, cycle mode, press setup, single stroke monitoring, press automatic mode)
  - Input override at a function block for muting
  - Input reset at a function block for valve monitoring
  - Resetting of inputs to zero and setting at a start value on an event counter function block

## Deactivating test pulses

It is possible to deactivate test pulses at one or more outputs of SP-COPx modules or SP-SDIO models with module version B-01 and higher.



**ATTENTION**

### The deactivating of the test pulses at any output reduces the security parameters of all outputs!

- The deactivating of the test pulses at one or more outputs of an SP-SDIO module reduces the security parameters of all Q1 to Q4 outputs of this module. Be aware of this in order to ensure that your application corresponds to an appropriate risk analysis and risk avoidance strategy.
- The deactivating of the test pulses at one or more outputs of an SP-COPx module reduces the security parameters of the relevant outputs. Be aware of this in order to ensure that your application corresponds to an appropriate risk analysis and risk avoidance strategy.

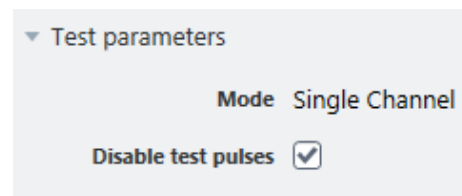
You will find more detailed information about the security parameters in the "samosPRO Hardware" manual (BA000966).

## Procedure

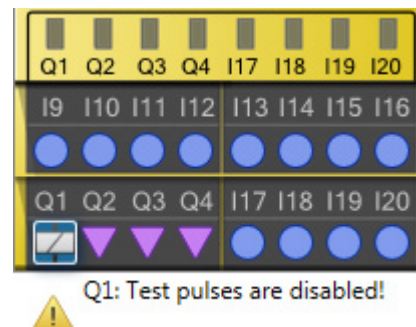
- ➔ Activate the **Deactivate test pulses** checkbox in the configuration dialog (right sidebar).  
⇒ A warning message will appear in the work area.

## Example

The test pulses will be deactivated in the configuration dialog.



A corresponding warning message will appear in the work area.




## 5.4.5 Creating user-defined elements (exercise)


### Background

In addition to the standard elements in the **Hardware** view installed with samosPLAN5+, it is also possible to create, configure, import, and export user-defined elements. This function makes it possible for you to create your own elements with preset configuration options (e.g. single-channel or two-channel evaluation, synchronous time, On-Off filtering, connection to test outputs, etc.) that will meet to the requirements of your individual equipment.

### Step 1: Creating a new library

- ➔ Click on the **User-defined elements** section heading.
  - ⇒ The **Add** icon will appear to the right next to the section heading.  

- ➔ Click on the **Add** icon.
  - ⇒ The **Please enter name** window will open.
- ➔ Enter a name and click on **OK**.
  - ⇒ Underneath the **User-defined elements**, a new library will appear.
- ➔ On the left sidebar, click on the newly created library and configure the following properties within the right sidebar.
  - **Security** section:  
Make the following security settings for the library.  
You can define different access modes and assign respective of passwords for them.
  - **Info** section:  
Click on the **Save** button and back up the library (file format: \*.SPI).

### Step 2: Creating a new element

- ➔ In the left sidebar, open the newly created library by clicking on the **Arrow** icon.
  - 
  - ⇒ The library will be opened.
- ➔ Click on **Add element**.
  - ⇒ The configuration dialog will open.
- ➔ Click on **Add** and double-click on the desired element type within the **Available elements** dialog.
  - ⇒ The new element will be displayed in the configuration dialog overview.
- ➔ Click on the element in the overview and assign the desired parameters in the right sidebar of the configuration dialog.
  - The type and scope of the parameters are different depending on the element type.

You can obtain more detailed information on the element types and their parameters in the **Hardware** manual.

## 5.4.6 Exporting and importing user-defined elements

You can export libraries with user-defined elements that you have created in samosPLAN5+ and import them into a different installation of samosPLAN5+.

### File format

\*.SPI

### Transferring user-defined libraries to a different PC (exporting)

- ➔ Switch to the **Hardware | Library | User-defined elements** area.
- ➔ From there, click on the library you wish to use in a different installation of samosPLAN5+.
  - ⇒ The configuration dialog for the selected library will open in the right sidebar.

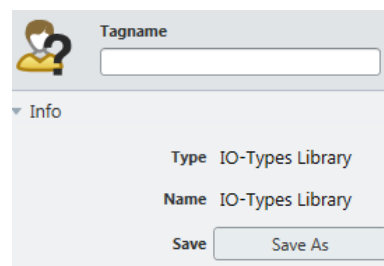


Illustration 29: Configuration dialog with "Save as" button

- ➔ Click on the **Save as** button in the right sidebar.
  - ⇒ Windows Explorer will open at the location you wish to place the selected library.
- ➔ Select the desired library (file in \*.SPI format).
  - ⇒ You can copy the SPI file to the clipboard using Windows commands (context menu) or send it via e-mail.

### Importing user-defined libraries

- ➔ Switch to the **Hardware | Library** view.
- ➔ Click on **User-defined elements** with the right mouse key.
  - ⇒ The context menu will open.

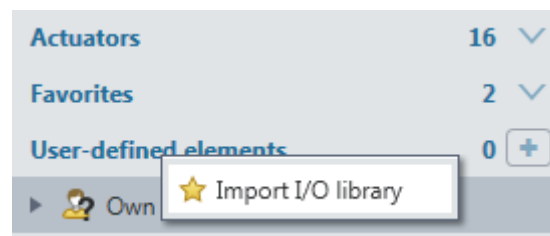


Illustration 30: Context menu with activation of import function

- ➔ In the context menu, click on **Import IO library**.
  - ⇒ Windows Explorer will open.

You can search for the library (file in \*.SPI format) using Windows Explorer and select it.

## 5.5 "Logic" view

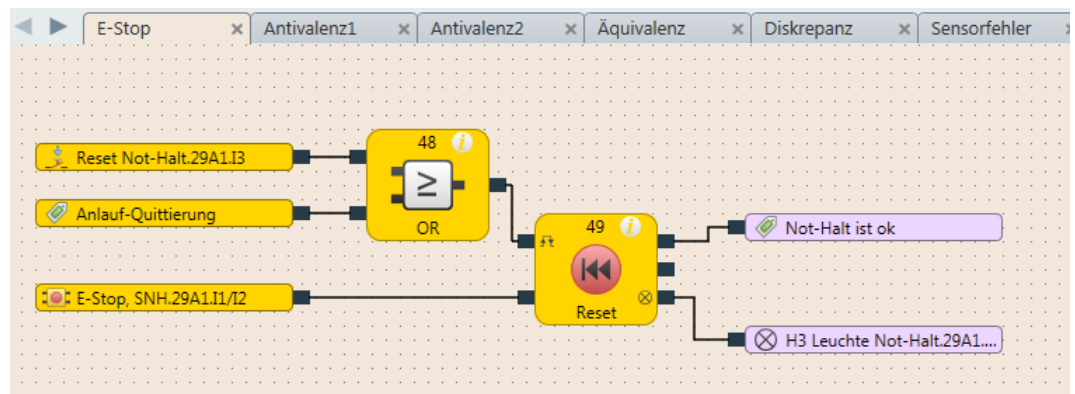
In **Logic** view, you can program the functional logic with the assistance of logical function blocks and application-specific function blocks.

The **Logic** view will not be available until you have dragged at least 1 controller module and placed it into the work area in the **Hardware** view.

### 5.5.1 Setup and functions

#### Work area

The **Logic** view work area is a graphical interface. This is where you can connect inputs, function blocks, and outputs with one another and thus program the complete functional logic with the assistance of the configurable parameters.



By default, the first page of the work area is created with the designation **Page 1**. You can do the following using the buttons on the upper right edge:

Create additional pages



Toggle between pages



You can differentiate between secure and non-secure IOs using the color scheme. The report also uses these colors:

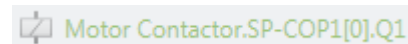
	secure input	secure output	non-secure input	non-secure output
0 (Low)				
1 (High)				

- 0 (Low): **Offline, Not simulated, Simulated and inactive** or **Online and inactive** state
- 1 (High): **Online and active** or **Simulated and active** state

## Left sidebar | Library

The **Library** tab in the left sidebar will provide you with the necessary components for the logic programming. Using the drag-and-drop function, you can move the function blocks (predefined or user-defined) and the inputs and outputs resulting from the elements configured in the hardware into the work area.

Outputs and inputs that have been moved to the work area with the drag-and-drop function are indicated in green.



Right-clicking on an element highlighted in green brings up a usage list in the context menu:

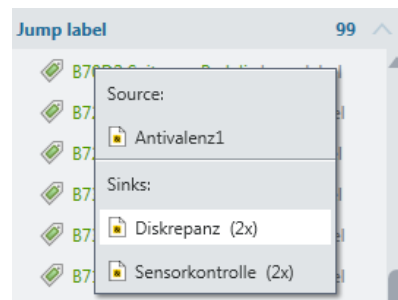


Illustration 31: Usage information on a jump label

The usage list documents how often and per element on what page of the **Logic** view the element is used. The usage list is available for inputs, outputs, jump labels and CPU flags.

Clicking on a source or target takes you directly to the corresponding point in the work area in the **Logic** view.

## Left sidebar | Overview

In the **Overview** tab of the left sidebar, you can see all of the project components used as a hierarchical tree structure.

## Right sidebar

The configuration dialog for the currently selected element will appear in the right sidebar.

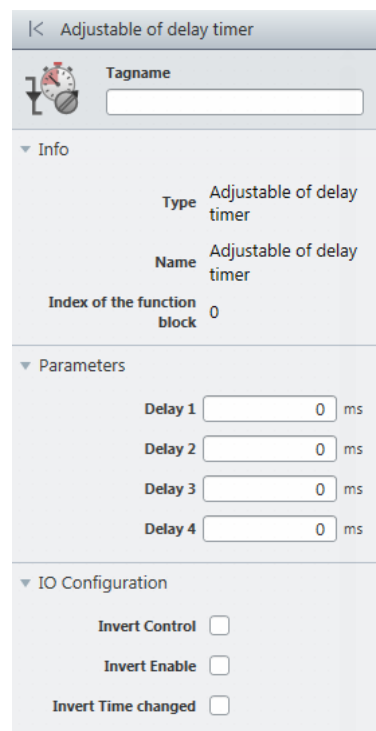













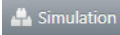

Illustration 32: Right sidebar in the "Logic" view



## Command bar and keyboard commands

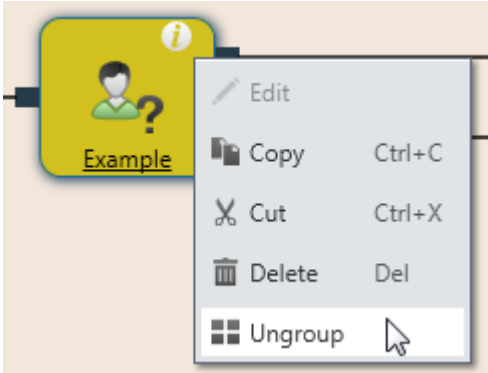
In the **Logic** view, you have the following view-specific commands and functions available:

Table 12: Reference

Element	Function
	Activates/deactivates the drag mode in the work area. Keyboard command: <b>Space</b> + <mouse movement>
100% ▾	Scales the display in the work area. Keyboard command: <b>Ctrl</b> + <mouse wheel>
 Normal ▾	Switches among the normal, overview, and matrix views. Further information: <i>Three views for logic programming [ch. 5.5.7, p. 57]</i>
	Activates/deactivates the marking of inputs and outputs.
	Activates/deactivates the grid points on the pages in the work area.
	Makes the last action undone (undo).
	Redoes the action that was recently undone (redo).
	Copies the currently selected project component. Keyboard command: <b>Ctrl + C</b>
	Cuts the currently selected project component. Keyboard command: <b>Ctrl + X</b>
	Inserts a copied or cut project component into the currently open page of the work area. Keyboard command: <b>Ctrl + V</b>
	Deletes the current selection in the work area. Keyboard command: <b>Del</b>
 Group	Groups multiple function blocks selected in the work area into one complex switching logic. <ul style="list-style-type: none"> <li>You can only group function blocks.</li> <li>Grouped function blocks are shown as an abstract element (black box) on the pages of the logic editor. The content of the grouping can be seen in editable form on its own page in the work area. From there, you can further edit and parameterize the interconnection.</li> <li>You can save a grouping as a user-defined function block directly from this grouping page.</li> <li>You will find the command to cancel a grouping in the context menu.</li> </ul> Keyboard command: <b>Ctrl + G</b> Further information: <i>User-defined function blocks [ch. 7.11, p. 216]</i>
 Simulation	Starts simulation mode. Further information: <i>Simulating configuration [ch. 8, p. 223]</i>
 No Logic Errors	Indicates whether there is a logic error (connection error). Further information: <i>Automatic logic check [ch. 5.5.4, p. 55]</i>

## Context menu

Table 13: Reference

Element	Function
Cancel or undo grouping	<p>Cancels a grouping from two or more function blocks. Example:</p> 

## Simulation mode

Once the logic programming is complete (error-free), you can test this via the samosPLAN5+ simulation mode.

You can obtain additional information on this topic here: *Simulating configuration [ch. 8, p. 223]*

## Creating and managing user-defined function blocks

You can create your own libraries in the **Logic** view library under **User-defined function blocks**.



Illustration 33: User-defined function blocks

You can obtain additional information on this topic here: *User-defined function blocks [ch. 7.11, p. 216]*

## 5.5.2 Programming logic incrementally (exercise)

This section shows you how you can program a simple logical function in samosPLAN5+ using an example.

### Requirements

You have created the following hardware elements in **Hardware** view.

- 1 compact module
- 1 SP-SDIO module
- 2 populated inputs
  - Sensor, emergency stop, single-channel
  - Sensor, reset, single-channel
- 1 populated output
  - Actuator, bulb

### Step 1: Moving elements into the work area with drag-and-drop

- ➔ Switch to the **Logic | Library** view.
- ➔ Move the two inputs (emergency stop and reset) out of the **Inputs** section and into the work area using drag-and-drop.
  - ⇒ The inputs will be marked in green in the library.
- ➔ Move the function block into the work area.  
Here: **Function blocks | Logic | Application | Reset**
  - ⇒ Because not all of the inputs are connected, the function component will be marked in red. In addition, the status display will indicate an error in the menu bar.
- ➔ Move the bulb from the **Outputs** section into the work area using drag-and-drop.

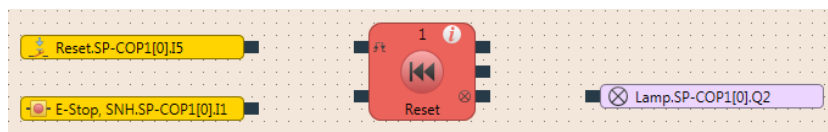


Illustration 34: All hardware elements will then be moved to the work area.

## Step 2: Interconnecting elements

- ➔ Using the mouse, click on the node of an input and press and hold the mouse key.
  - ⇒ A hand icon will appear.

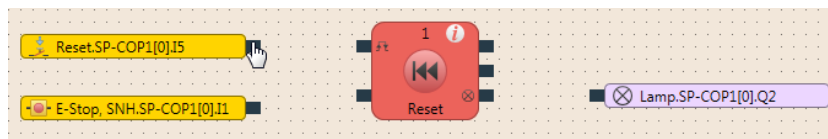


Illustration 35: Hand icon

- ➔ Move the hand icon on to the input node of the function component and release the mouse key.
  - ⇒ The input and the function component will be connected with a line (logical connection).



Illustration 36: Connection between output and input

- ➔ In the same manner, establish the connection for the other input and output.
  - ⇒ Because all inputs of the function component are connected, the component will be marked in yellow and the status display will no longer show an error in the menu bar.

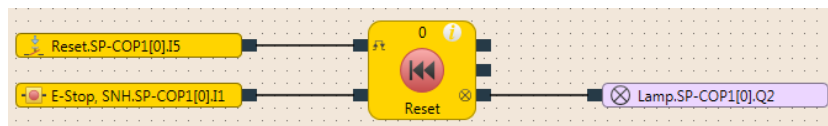


Illustration 37: All hardware elements will then be connected to the function component.

## Tips for working in the "Logic" area

Topic	Tip
Interconnecting elements	An alternative to the previously described procedure: Using drag-and-drop, move an input or output over the node of the function block to which the connection is to be established.
Placing components	You can move function blocks, inputs, and outputs anywhere within the work area. Click on them with the mouse and move them to the desired spot. As a placement aid, you can activate the grid points in the command bar.
Adding notes	You can save notes at any place in the work area. To do this, click with the right mouse key on the empty background of the work area and select the <b>Add note</b> entry in the context menu.
Working with multiple pages	In order to maintain an overview of an extensive project, you can create additional pages within the work area. <ul style="list-style-type: none"> <li>• Creating a new page: Click on the + button.</li> <li>• You can create connections between the pages using a CPU flag or jump addresses.</li> </ul>
Deleting connections	To delete a project component or a connection: <ul style="list-style-type: none"> <li>• Click on the element.</li> <li>• Press the <b>Del</b> button.</li> </ul>

## 5.5.3 Status bits for type SP-COP modules (reference)

For type SP-COP modules that you install as hardware, you have the following status bits available in the logic.

### Location

You can find the status bits in the left sidebar under: **Logic | Library**

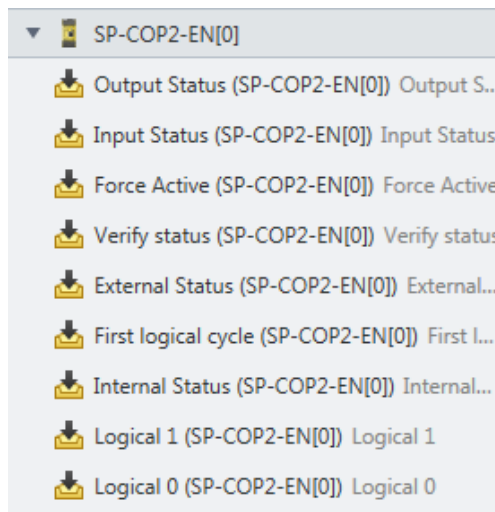


Illustration 38: Status bits

### Reference

Table 14: Status bits and their values

Bit	Possible values
Output status	<p>Indicates whether there is an error with respect to the output test pulses and the plausibility check or with internal outputs.</p> <ul style="list-style-type: none"> <li>• <b>1 (High):</b> No error</li> <li>• <b>0 (Low):</b> Error present.</li> </ul>
Input status	<p>Indicates whether there is a procedure error, synchronous time error, test pulse error, or an internal error at one of the inputs.</p> <ul style="list-style-type: none"> <li>• <b>1 (High):</b> No error</li> <li>• <b>0 (Low):</b> Error present.</li> </ul>
First logic cycle	<p>You can trigger initialization functions in the logic program using this input.</p> <ul style="list-style-type: none"> <li>• <b>1 (High):</b> First logic cycle after the transition from the <b>Stop</b> state into the <b>Run</b> state.</li> <li>• <b>0 (Low):</b> State during all other logic cycles.</li> </ul>

Bit	Possible values
External status	<p>Indicates whether there is a B1, B2 voltage error and/or overcurrent error at the outputs.</p> <p>From A-03 onwards, errors of the output state or the input state can also be detected on this state bit.</p> <ul style="list-style-type: none"> <li>• <b>1 (High):</b> No error</li> <li>• <b>0 (Low):</b> Error present.</li> </ul>
Force active	<p>Indicates whether the force mode is active.</p> <ul style="list-style-type: none"> <li>• <b>1 (High):</b> Force mode is active.</li> <li>• <b>0 (Low):</b> Force mode is not active.</li> </ul>
Internal status	<p>Indicates whether a critical error is present.</p> <ul style="list-style-type: none"> <li>• <b>1 (High):</b> No error</li> <li>• <b>0 (Low):</b> Error present.</li> </ul>
Logical 0 and logical 1	<p>You can use these status bits in order to obtain a valid logical configuration when it contains functional block inputs that a) are not required and b) cannot be deactivated.</p> <ul style="list-style-type: none"> <li>• <b>Logical 0:</b> Permanently sets the input of a function block to <b>0</b> (low).</li> <li>• <b>Logical 1:</b> Permanently sets the input of a function block to <b>1</b> (high).</li> </ul>
Verified	<p><b>Indicates whether the configuration is verified.</b></p> <ul style="list-style-type: none"> <li>• <b>1 (High):</b> Configuration verified. The CV-LED on the COMPACT module is permanently lit yellow.</li> <li>• <b>0 (Low):</b> Configuration is not verified. The CV-LED on COMPACT module is flashing yellow in a 1 Hz cycle.</li> </ul>

## 5.5.4 Automatic logic check

samosPLAN5+ automatically checks the logical programming in your project. Connection errors are checked; other errors are not detected.

As long as the configuration is not valid, it will not be possible to start simulation mode or to transfer the configuration.

### Important security information



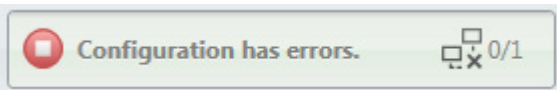
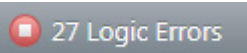

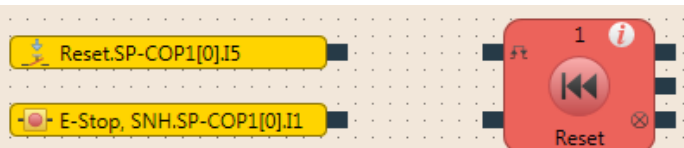
#### Check your application thoroughly for correctness!

Because the samosPLAN5+ only checks for logic-internal connection errors, you have to check the following aspects systematically yourself:

- Does your application correspond to the results from the risk analysis and the avoidance strategy?
- Have all of the applicable standards and guidelines been complied with? If not, you are placing the machine's operator in danger.

### Displaying errors

If there is a connection error, you can see this at several places in samosPLAN5+:

Display	Explanation
Status display in the menu bar	<p>If there is a connection error, the status display will show the following message:</p>  <p>You will always see this display regardless of which samosPLAN5+ view is currently active.</p>
Logic command bar, outside right	<p>If there are errors present, this display shows the number of function blocks having the connection errors.</p> 
Logic work area	<p>Pages that have a connection error have a warning symbol displayed in the tab:</p>  <p>Function components at which one or more inputs are not connected are shown in red.</p> 

### Eliminating connection errors

- ➔ Connect all of the inputs at function blocks to the corresponding hardware inputs.
  - ⇒ The function blocks are shown in yellow.
  - ⇒ The error message in the **Logic** command bar and in the menu bar will be deleted.

### 5.5.5 Using CPU flags

CPU flags are available as inputs and outputs in the logic editor. They can be used, for example, to create logical loopbacks.

A CPU flag consists of an output flag and an input flag. The input flag always assumes the same value (**high** or **low**) as the corresponding output flag with a delay of a logic cycle (i.e. the logic execution time).

You can use input flags multiple times.



#### **Increased delay times can make the control unstable.**

CPU flags always cause a delay of 1x the logic execution time. The reason for this is that the input flags always assume the value that the output flag had in the previous logic cycle.

- Always calculate the resulting delay when computing the response time and functionality.

### 5.5.6 Using jump addresses

Jump addresses consist of a source jump address and a target jump address. The target jump address assumes the same value (**high** or **low**) as the corresponding source jump addressed without delay – provided it does not involve a loopback. In this respect, jump addresses differ from CPU flags.

You can use jump addresses in order to interconnect components that are on different logic pages.

You can use target jump addresses multiple times.



## 5.5.7 Views of logic programming (tip)

In order to ensure an optimal overview, primarily for extensive projects, samosPLAN5+ offers three different views of the logic programming in the work area.

You can change the view in the command bar:

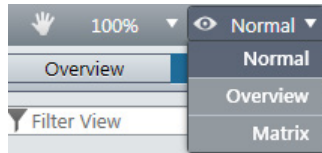


Illustration 39: Selection list for changing views

### "Normal" view

In this view, you can carry out programming. You can see all of the programming components, you can move them to any location, and, if necessary, you can distribute them to multiple pages.

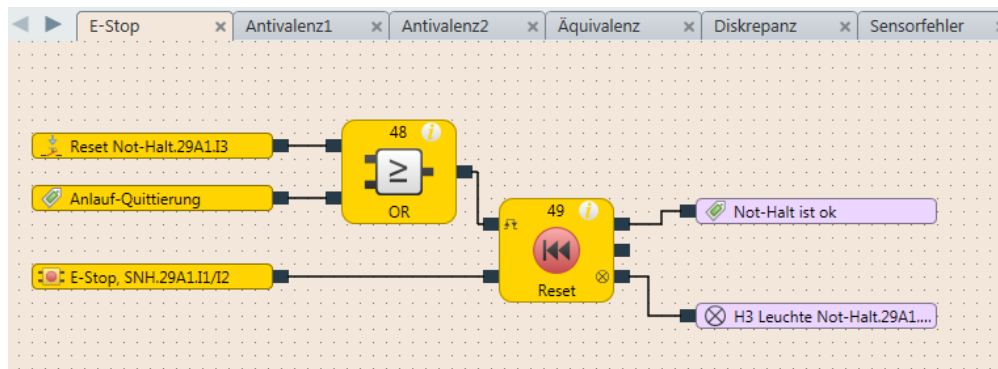


Illustration 40: "Normal" view

### "Overview" view

This view shows all of the inputs and outputs, per page, that you use in logic programming. The logical links (logic components and connections) are shown in abstract as a black box.

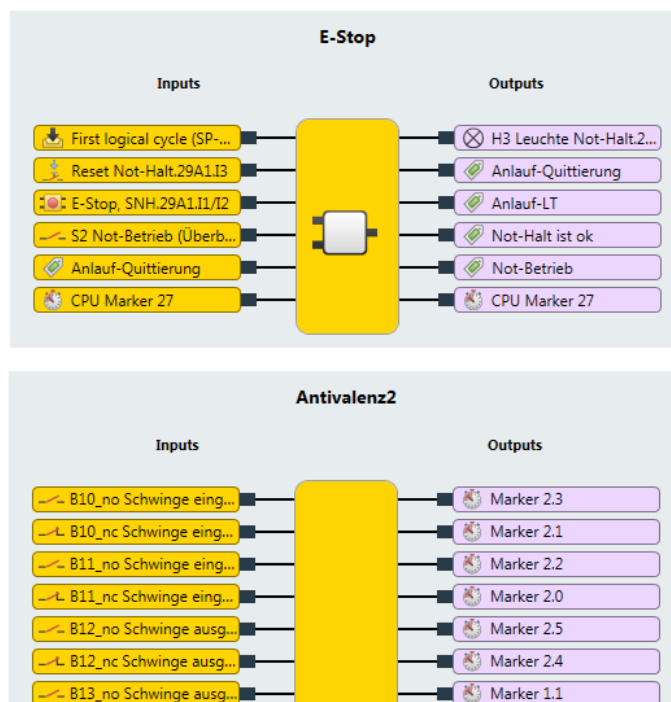


Illustration 41: "Overview" view

## "Matrix" view

This view shows you a detailed view as to which inputs act on which outputs. This will help you in checking whether your logic programming is complete.

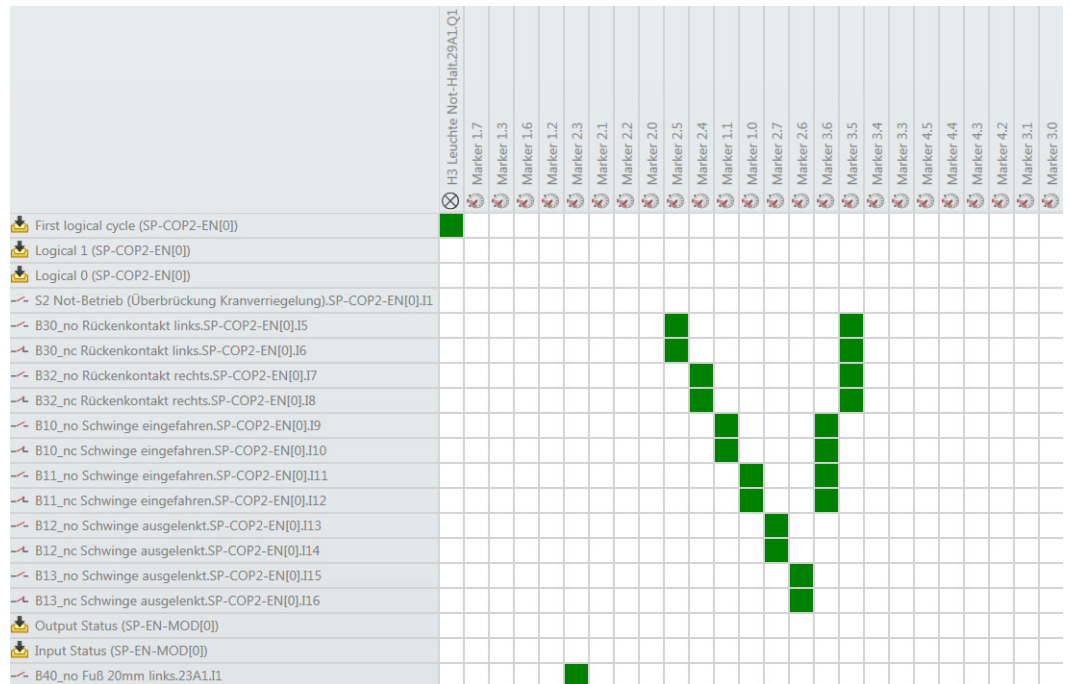


Illustration 42: "Matrix" view

You can determine the relationship of inputs and outputs using the color marking of the matrix intersections:

Table 15: Color key

Marking	Explanation
Green intersection	Input (line) acts on output (column)
White intersection	No logical relationship between input and output

## 5.6 "Gateway" view

You can find extensive information regarding the **Gateway** view in the following manual: "samosPRO COMPACT gateways" (BA000970)

## 5.7 "Name" view

The **Name** view contains a list of all project components. Here, you can configure the designation of the project components on the samosPLAN5+ interface.

### Info

- **Requirement**

To individually configure the names, you have to use the **Tag** element in the naming scheme. You define the naming scheme in the overall setting for samosPRO5+: *Configuration of display names [ch. 5.2.2, p. 26]*

- **Effect**

The effect your entries have on the designation of the project components depends on where you position the **Tag** element in the naming scheme.

### 5.7.1 Setup and functions

#### Work area

The work area contains a list of all project components.



Test-Project	Test-Project	Project
Station 1	Station 1	Station
E-Stop, SNH.SP-COP1[0].I1		Input
Enabling Switch.SP-COP1[0].I3/I4		Input
Reset.SP-COP1[0].I5		Input
Motor Contactor.SP-COP1[0].Q1		Output
Lamp.SP-COP1[0].Q2		Output

Illustration 43: Work area in the "Name" view

## Command bar

In the **Name** view, you have the following view-specific commands and functions available:

Table 16: Reference of commands and functions

Element	Function
	This enables the export or import of the display names defined here. Thus, you can also use your specifications in other projects.
	This reduces the display in the work area to the selected types of project components.  <div style="border: 1px solid gray; padding: 5px; background-color: #f0f0f0;"> <p><b>Types to display:</b></p> <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Project</li> <li><input checked="" type="checkbox"/> Controls</li> <li><input checked="" type="checkbox"/> Compact modules</li> <li><input checked="" type="checkbox"/> Modules</li> <li><input checked="" type="checkbox"/> Gateways</li> <li><input checked="" type="checkbox"/> I/O containers</li> <li><input checked="" type="checkbox"/> Inputs</li> <li><input checked="" type="checkbox"/> Outputs</li> <li><input checked="" type="checkbox"/> Logic pages</li> <li><input checked="" type="checkbox"/> Function blocks</li> <li><input checked="" type="checkbox"/> Logic results</li> <li><input checked="" type="checkbox"/> Ports</li> <li><input checked="" type="checkbox"/> Gateway inputs</li> <li><input checked="" type="checkbox"/> Gateway outputs</li> </ul> </div>

## Left sidebar

The left sidebar shows all of the project components as a hierarchical tree structure.

You can reduce the content of the tree structure to entries that contain the input character string using the **Filter view** input field.

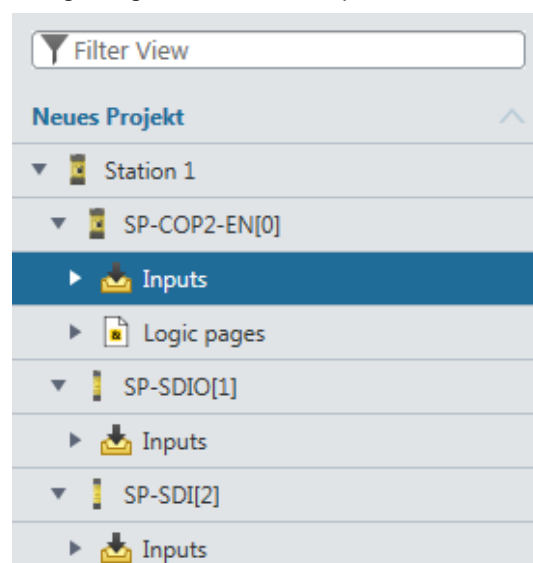


Illustration 44: Left sidebar in the "Name" view

## 5.7.2 Assigning Tag names/changing display names (exercise)

### Procedure

- ➔ Select a project component containing the display names you would like to adapt and click on the **Tag name** in the input field.
- ➔ Enter any desired character string into the input field.
  - ⇒ Depending on the selected project component, either the entire name or only part of the display name changes.
  - See below for details.

### Background

samosPLAN5+ automatically assigns a display name for all project components. This name may consist of 1 or more digits (classes of information).

When you have a 1-digit display name, a tag name will overwrite the entire display name. When you have a multi-digit display name, a tag name will only overwrite the component of the display name that you have explicitly defined as a tag name in the main menu (see *Configuration of display names [ch. 5.2.2, p. 26]*).

Table 17: Examples

Example	Explanation				
Names of pages	<p>Pages in samosPLAN5+ have a 1-digit display name. By default, it is constructed according to the template "page + &lt;number of page&gt;"</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <span style="border: 1px solid #ccc; padding: 2px;">Page 1</span> <span style="border: 1px solid #ccc; padding: 2px; margin-left: 100px;">Page 1</span> </div> <p>If you overwrite the tag name, the new display name will correspond precisely to your input:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">System Name</th> <th style="width: 50%;">Tag-Name</th> </tr> </thead> <tbody> <tr> <td style="background-color: #0056b3; color: white;">Test-Project</td> <td style="border: 1px solid #ccc; padding: 2px;">Test-Project</td> </tr> </tbody> </table>	System Name	Tag-Name	Test-Project	Test-Project
System Name	Tag-Name				
Test-Project	Test-Project				
Names of inputs	<p>The display name of inputs consist of three digits together by default.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <span style="border: 1px solid #ccc; padding: 2px;">E-Stop, SNH.SP-COP1[0].I7</span> <span style="border: 1px solid #ccc; padding: 2px; margin-left: 100px;"></span> </div> <p>When you assign a tag name, only the digit that is defined as a tag name will change. Here is an example using the first digit:</p> <div style="border: 1px solid #ccc; padding: 5px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td style="background-color: #0056b3; color: white;">NH3.SP-COP1[0].I3</td> <td style="border: 1px solid #ccc; padding: 2px;">NH3</td> </tr> </tbody> </table> </div>	NH3.SP-COP1[0].I3	NH3		
NH3.SP-COP1[0].I3	NH3				

## 5.8 "Report" view

The **Report** view contains complete information regarding the currently loaded project and all of the settings including the logic programming and the interconnection diagrams.

All information can be saved in standard file formats and printed. The scope of the report can be individually compiled depending on the selection.

### 5.8.1 Setup

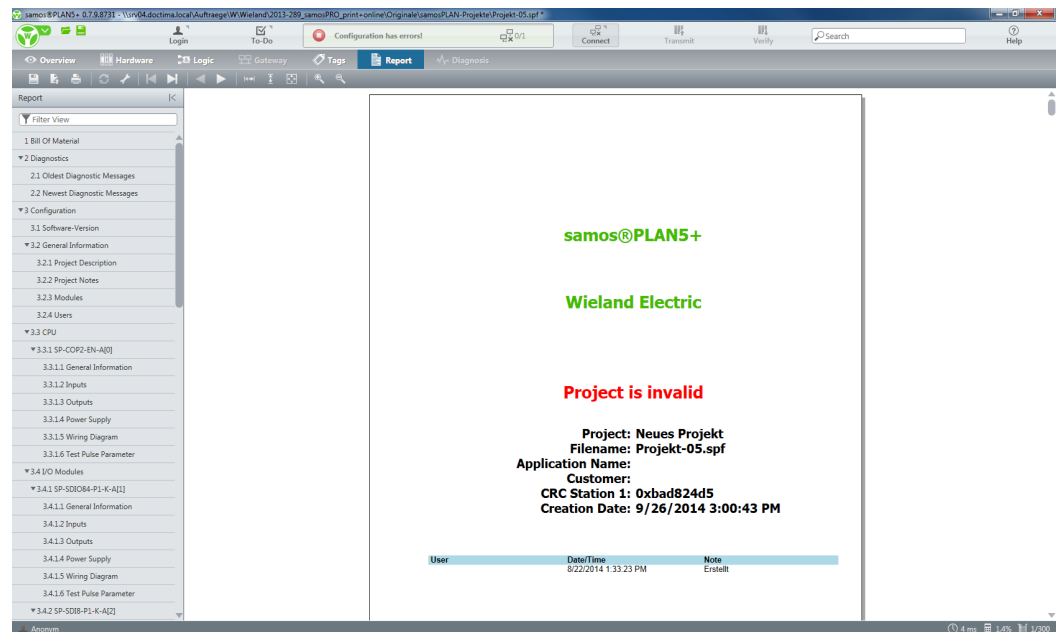


Illustration 45: Setup of the "Report" view

#### Work area

You can see the content of the report in the work area.

All of the content is automatically compiled from your project configuration.

- You are not able to manually change individual parts of the report however.
- You can only determine which of the potential sections in the report will be generated.

#### Left sidebar




The left sidebar shows the section breakdown of the report. You can switch between sections by clicking on the corresponding entry in the tree view.

## 5.8.2 Functions

The command line provides you with the following commands and functions:



### Save and print functions

Table 18: Reference

Element	Function
	Saves the report as a PDF file.
	Saves the report as an XML file.
	Opens a dialog window that you can use to send the report to a printer.



### Configuration

Table 19: Reference

Element	Function
	Opens a dialog in which you can define which section the report should contain.
	Updates the breakdown of the report after you have changed the section breakdown.

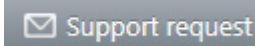
### Navigation and scaling

Table 20: Reference

Elements	Function
	Enables navigation within the report: <ul style="list-style-type: none"> <li>• From page to page</li> <li>• To the first page or to the last page of the report</li> </ul>
	This determines the size of the display in the work area.

### Support

Table 21: Reference

Elements	Function
	Generates an e-mail that is addressed to Wieland Support. <b>Note</b> You can define which information you would like to send to Wieland Support: <ul style="list-style-type: none"> <li>• Text message</li> <li>• Current report from samosPLAN5+</li> <li>• File of the project currently open</li> </ul>

## 5.8.3 Edit front page

You can customize the report front page. Three areas can be configured:

- Company logo
- Company name
- Company address

The company logo, company name and company address are displayed in this order in the middle of the front page. The company name also appears in the report footer.

### Activate

Main menu | Settings | Edit Front Page menu

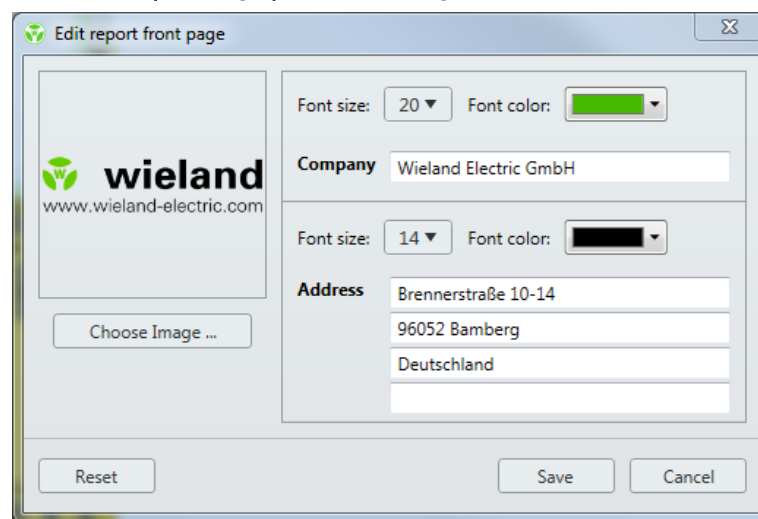


Illustration 46: Configuration example

### Configuration options

Table 22: Reference

Element	Description
Choose image ...	You select the desired graphic from your local directory by clicking on the button. This graphic is saved in the project file.
Company	Enter the company name here. You can define a separate font size and font color for the company name.
Address	Enter the company address here. You can define a separate font size and font color for the company address.



## 5.9 "Diagnostics" view

In the **Diagnostics** view, you can execute diagnostics for an existing safety control.

**NOTICE** **Before diagnostics: Connects to samosPRO system**  
 In order to execute diagnostics, you must be connected to the samosPRO system.  
 To obtain additional information on this topic: *Connecting to the samosPRO system [ch. 6, p. 68]*

### 5.9.1 Setup

#### Work area

In the work area, you will see a list with all of the messages that are currently pending in the connected system.

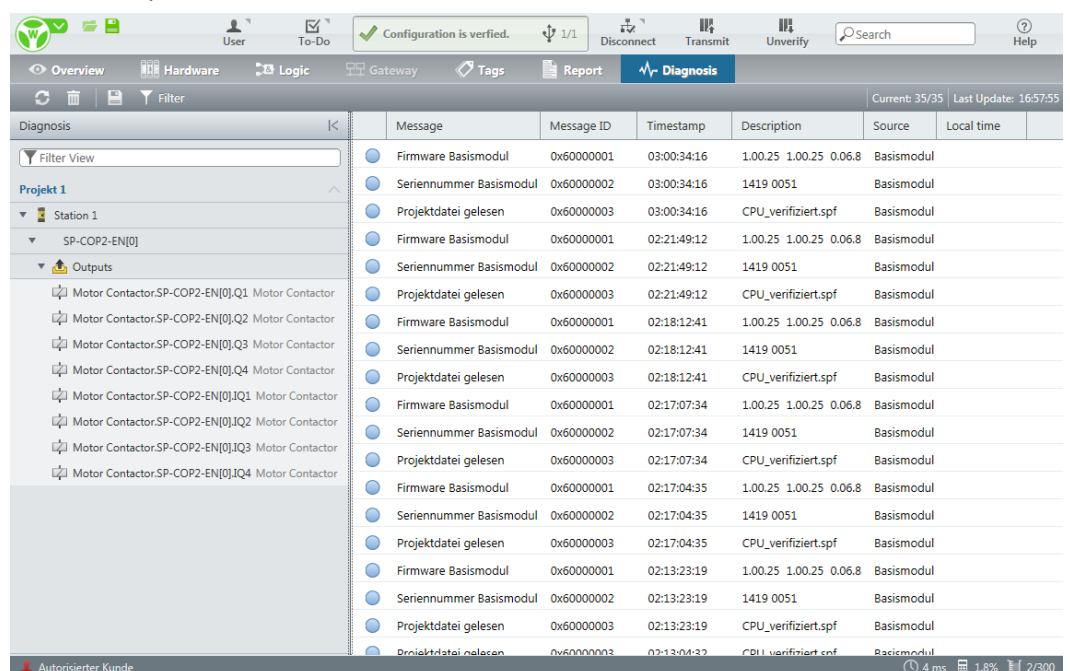


Illustration 47: Setup of the "Diagnostics" view

The following information is displayed regarding the messages in the columns of the message list:

Table 23: Reference

Column	Description
	Severity level of the message: <ul style="list-style-type: none"> <li>Blue: Info</li> <li>Orange: Warning</li> <li>Red: Error</li> </ul>
Message	Text title of the message
Message ID	Unique ID as a hexadecimal number throughout the system
Timestamp	Total operating time of the main module when the message occurred
Description	Detailed information for the diagnostics
Source	Module that detected the error
Local time	Time at which the message occurred (system time of your PC)

**NOTICE** If an error occurs, you can find additional information here:

- *Complete list of error messages* [ch. 15.1, p. 245] (Error codes, error causes and troubleshooting measures)
- "samosPRO Hardware" manual (BA000966)

**NOTICE** A maximum of 5000 of the latest error entries are automatically transferred to the samosPLAN5+ software.  
 If you want to transfer older error entries: Import the **history.csv** file into samosPLAN5+. This file is located on the SP-COP-CARD1 SD card.

### Left sidebar

The left sidebar shows all of the project components as a hierarchical tree structure.

You can reduce the content of the tree structure to entries that contain the input character string using the **Filter view** input field.

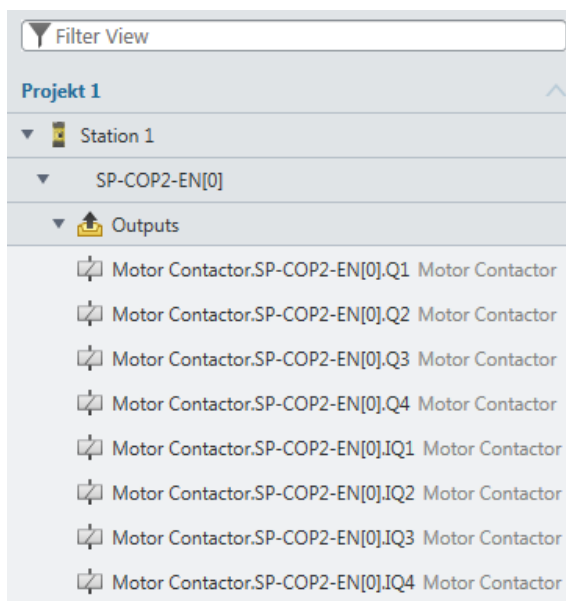


Illustration 48: Left sidebar in the "Diagnostics" view

### Command bar

Table 24: Reference of commands and functions

Element	Function
	Updates the list of messages in the work area.
	This deletes the currently selected element.
	This opens a save dialog. From there you can save the message list of the work area as a CSV file.
	This reduces the list in the work area to the selected message types. <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;"> <p><b>Messages to show:</b></p> <p><input checked="" type="checkbox"/> Information</p> <p><input checked="" type="checkbox"/> Warning</p> <p><input checked="" type="checkbox"/> Error</p> <p><input checked="" type="checkbox"/> Stopping Error</p> <p><input checked="" type="checkbox"/> Critical Error</p> </div>

## 5.9.2 Synchronize time for diagnostic purposes

You can synchronize the time on the safety control with the time on the connected diagnostics computer for diagnostics purposes. This provides you with an absolute time for error tracking.

If you use this function, a corresponding absolute time value is displayed in the **Diagnostics** view for each message in the **Local Time** column.

**NOTICE** The time synchronized here does not affect the control function. The time continues running as long as the module is live, and it is noted for the subsequent diagnostics messages.

### Example

Message	Message ID	Timestamp	Description	Source	Local time
Base-Module	0x2B082FF0	03:16:14:37	(00002FF0)	Base-Module	7/14/2015 5:28:23 PM
Base-Module	0x22010226	03:16:14:37	(00000226)	Base-Module	7/14/2015 5:28:23 PM
Communication Error (Ethernet/USB)	0x6A0B0023	03:16:14:37	35	Base-Module	7/14/2015 5:28:23 PM
Communication Error (Ethernet/USB)	0x6A0B3101	03:16:14:37	49 1	Base-Module	7/14/2015 5:28:23 PM

Illustration 49: The "Local Time" column displays the time when the message occurred

### Synchronizing the time

This is how you synchronize the time on the safety control with the time on the connected diagnostics computer:

#### Requirement

- You are *connected* [ch. 6, p. 68] to samosPRO.
- You have logged on as a user with the authorization **D (output diagnostic data)**.  
Further information: *User administration* [ch. 5.3.4, p. 35]

#### Procedure

- ➔ Switch to the **Hardware** view.
- ➔ Click on an empty background area in the work area.  
⇒ The right side bar displays the configuration dialog of the control.

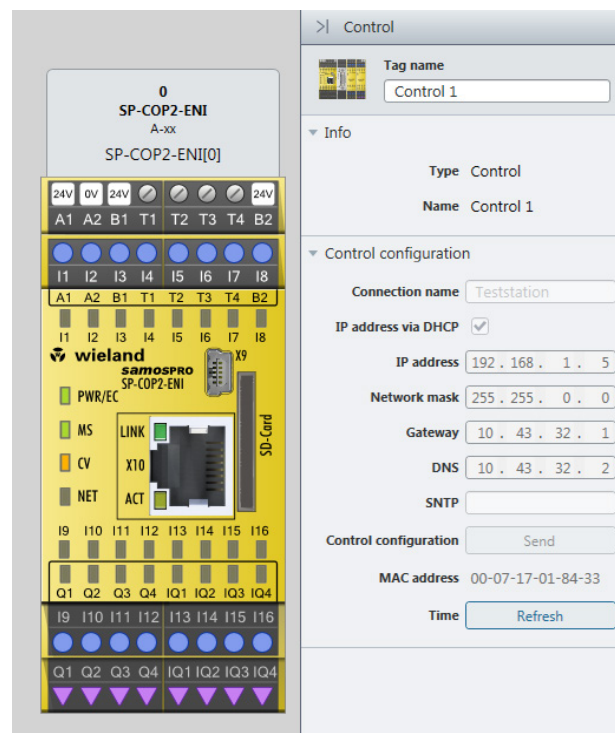


Illustration 50: Updating the time

- ➔ In the configuration dialog under **Time**, click on **Update**.

## 6 Connecting to the samosPRO system

This section describes how you can set up an initial connection between the samosPRO system and a PC or notebook.

As well as a USB interface, the samosPRO COMPACT SP-COP2-ENx main module has a TCP/IP configuration interface that makes it possible to configure the samosPRO COMPACT system via samosPLAN5+. This interface operates parallel to the Ethernet TCP/IP or other Ethernet protocols.

### Important information



**ATTENTION**

**Do not connect to the samosPRO system via the USB and the Ethernet interface simultaneously!**

The samosPRO system can only communicate with one samosPLAN5+ at a time. If you wish to establish multiple connections to the samosPRO system, either from a single PC or from multiple PCs, this can lead to inconsistencies in the configuration and diagnostics or to errors in the operation. This applies to both USB and Ethernet connections.



**ATTENTION**

**Also take note of the signal run-times in the case of remote TCP/IP connections!**

Remote TCP/IP connections to the gateway may be unstable if the signal run-time is too long.

- Use the ping command to check the signal run-time to the gateway. Signal run times > 250 ms may result in a connection loss.

#### Possible solutions:

- Ensure that the connection is fast enough or change the routing if this is possible.

#### Or:

- Use a program such as Teamviewer to control a local computer on which the samosPLAN5+ has been installed and that is locally connected to the samosPRO COMPACT system.

#### Or:

- Contact Wieland Electric Support.

### Step 1: Establish the cable connection and start the software

- ➔ Connect a PC or notebook to the USB or Ethernet interface of the SP-COPx.
- ➔ Switch the samosPRO system on.
- ➔ Open the samosPLAN5+ configuration software installed on the PC.

### Step 2: Establish the connection from samosPLAN5+

- ➔ Click on **Connect** in the menu bar.
- ⇒ The following dialog will appear.

Illustration 51: 'Connect' dialog

## Connecting to the samosPRO system

**NOTICE** You have two options for establishing the connection to the samosPRO.

- In the next step, select the procedure suitable for you.

➔ **Version 1:**

Press **Update** and then select the desired connection in the pulldown menu. This procedure is possible when the computer and the safety control are located in the same sub-network.

– or –

**Version 2:**

Press **Edit** and manually enter the IP address of the control in the **Configurable controls** dialog window.

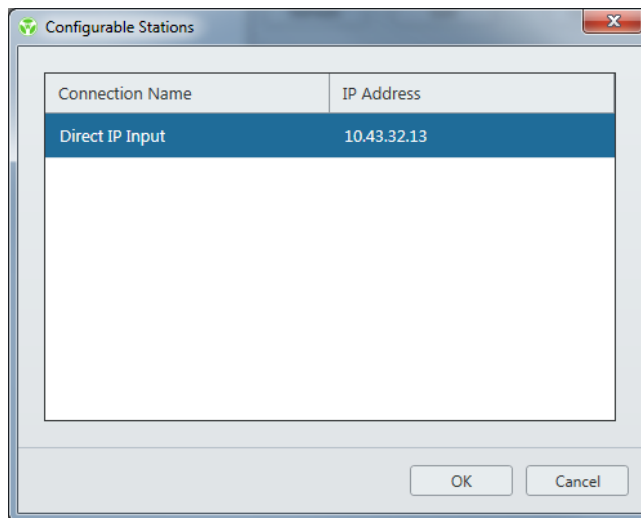


Illustration 52: Dialog for manual entry of the IP address

➔ Press the **Connect** button.

⇒ Depending on the particular user settings, the dialog for user logon will appear. There, you will have to log on with the password of the project located on the control.

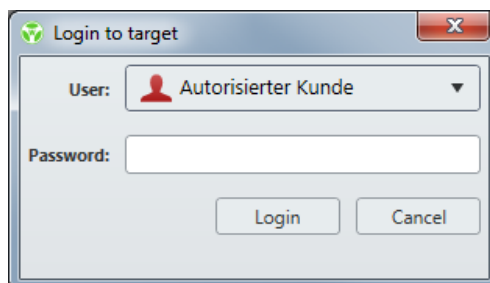


Illustration 53: Password logon

## Connecting to the samosPRO system

⇒ After successful connection, the **Connect** button will change to **Disconnect**, and the following dialog will appear in which you can select an action for the connection:

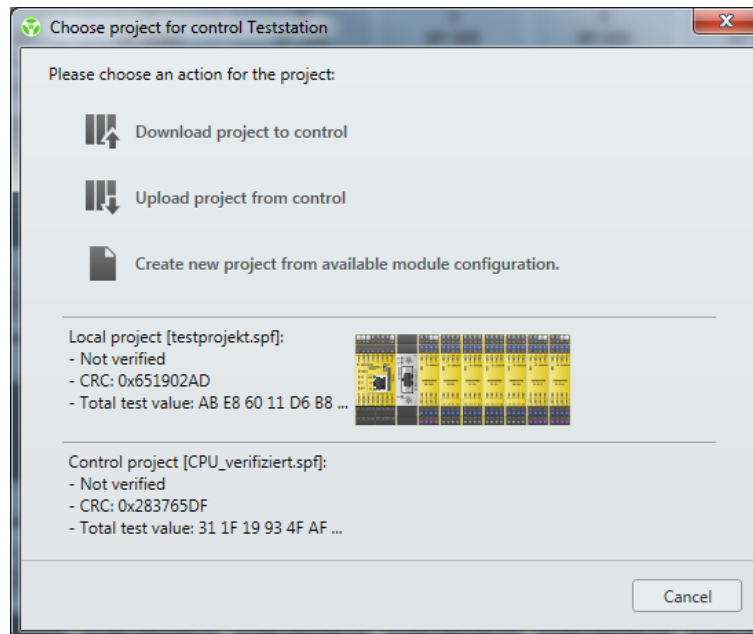
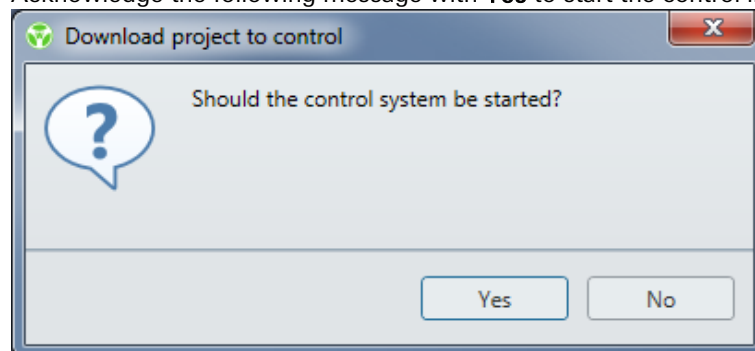


Illustration 54: Action view for connection (example)

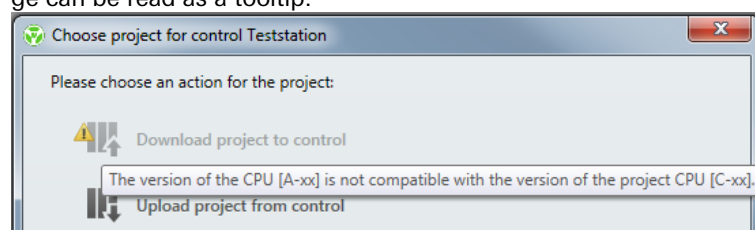
### Step 3: Execute action

- ➔ Select the desired action.
- ➔ If you have selected the action **Transfer project into control**: Acknowledge the following message with **Yes** to start the control immediately.



### Note

samosPLAN5+ prevents the transfer of a project that is not compatible in terms of the module version with the compact module onto which it is to be loaded. Example: You want to transfer a project from samosPLAN5+ to the controller in which the compact module has a higher module version (e.g. B-xx) than the compact module on the controller (e.g. A-xx). In such a configuration the connection dialog shows an exclamation mark. The error message can be read as a tooltip.



# Connecting to the samosPRO system

⇒ After successful transfer of a project from the station to the PC, this project will appear in the various views of samosPLAN5+ and you will be in online mode.

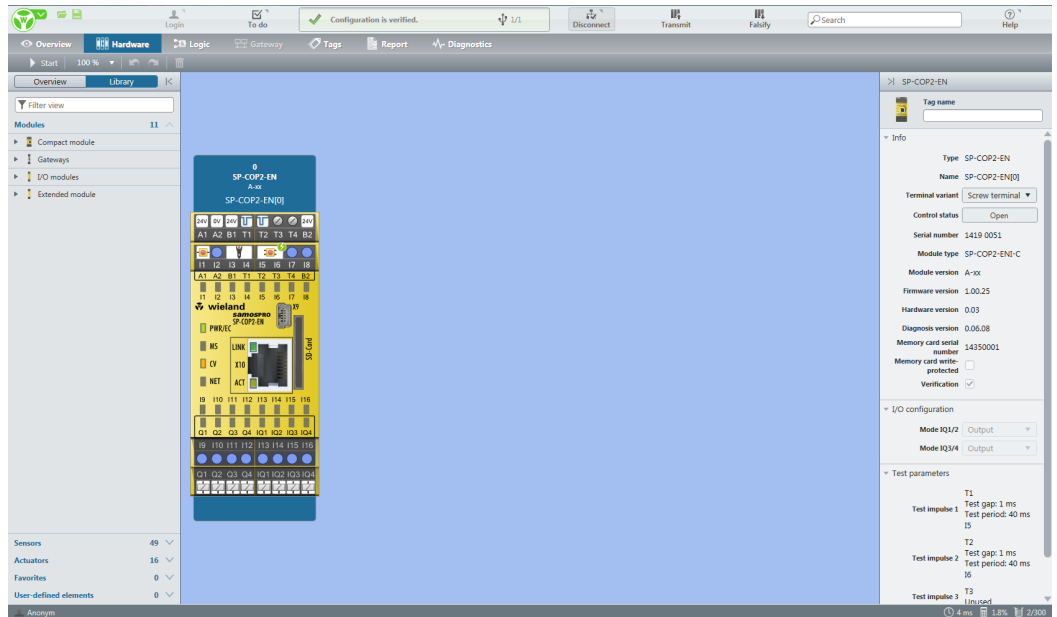


Illustration 55: Hardware configuration view (example)

## 7 Logic programming – Function blocks

### 7.1 General safety information regarding logic programming

The functional logic of the samosPRO system is programmed with the assistance of function blocks. These function blocks are certified for use in safety-relevant functions when all safety standards are being maintained during implementation. The following sections provide information on the important aspects of using function blocks in the samosPRO system.



ATTENTION

#### Be aware of the corresponding standards and safety regulations!

All of the safety-related parts of the system (wiring, connected sensors and control devices, configuration) must meet the respective standards (e.g. EN 62061 or EN ISO 13849-1) and safety regulations. Safety-relevant signals must be used for safety-input and safety-output signals in safety-relevant applications. Make sure that the application meets all of the applicable standards and regulations.

You are responsible for checking that the correct signal sources are being used for these function blocks and that the entire implementation of the safety logic meets the applicable standards and regulations. Always check the function of the samosPRO hardware and logic program in order to ensure that they are behaving according to your risk reduction strategy.



ATTENTION

#### Implement additional safety measures if the safety value could lead to a hazardous state.

The safety value for process data and outputs is low and is set if an error is determined. If the safety value (signal = low) could lead to a hazard-inducing state in the application, additional measures must be implemented, such as evaluation of the status of the process data and switch-off of the output signals in question, if the status evaluation detects an error. This applies particularly to inputs with edge detection.



ATTENTION

#### Pay particular attention to unexpected rising or falling edges.

Particular care must be taken during the planning of applications in which an unexpected rising or falling edge could lead to a hazard-inducing situation at an input with edge detection. An error at an input can generate such types of edges (e.g. interruption in network communication, cable break at a digital input, short-circuit at a digital input that is connected to a test output). The safe value is set until the conditions have been met for resetting the error. For this reason, the signal in question may behave as follows:

- It temporarily switches to high instead of remaining in the error-free low state (falling edge and rising edge, i.e. transition from low to high to low), or
- It temporarily switches to low instead of remaining in the error-free high state (falling edge and rising edge, i.e. transition from high to low to high), or
- It remains low instead of changing to the error-free state to high.



ATTENTION

#### Note the delays from the CPU flag in the reverse path.

A reverse path signal is an input signal that is connected at the output of a function block with the same or a higher function block index (the function block index is displayed at the top in each function block). Thus, the input uses the output value of the prior logic cycle. This must be considered for the functionality and particularly when calculating the response time. In order to connect a reverse path signal, you have to use a CPU flag. A CPU flag generally causes a delay of a logic cycle (see below: *Example*).

#### Example: CPU flag in the reverse path

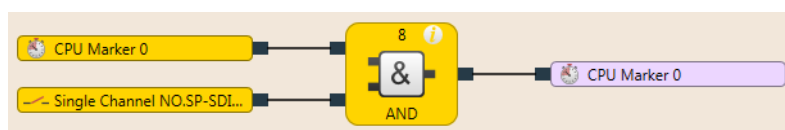


Illustration 56: CPU flag in the reverse path



## 7.2 Function block overview

The samosPRO system uses function blocks to define the safety-based logic. A configuration can consist of no more than 300 function blocks. There are logical function blocks and application-specific function blocks. The following table lists all of the available function blocks for SP-COPx:

Table 25: Overview of function blocks

<b>Logic</b>	
<ul style="list-style-type: none"> <li>• AND (And operation)</li> <li>• OR (Or operation)</li> <li>• XOR (exclusive OR)</li> <li>• NOT (negation)</li> </ul>	<ul style="list-style-type: none"> <li>• RS flip-flop</li> <li>• JK flip-flop</li> <li>• T flip-flop</li> <li>• Binary decoder</li> <li>• Binary encoder</li> </ul>
<b>Start/edge</b>	
<ul style="list-style-type: none"> <li>• Reset</li> <li>• Restart</li> </ul>	<ul style="list-style-type: none"> <li>• Edge detection</li> </ul>
<b>Delays</b>	
<ul style="list-style-type: none"> <li>• Switch-on delay</li> <li>• Switch-off delay</li> </ul>	<ul style="list-style-type: none"> <li>• Adjustable switch-on delay</li> <li>• Adjustable switch-off delay</li> </ul>
<b>Counter and cycle</b>	
<ul style="list-style-type: none"> <li>• Counter (upward, downward, upward and downward)</li> <li>• Cycle generator</li> </ul>	<ul style="list-style-type: none"> <li>• Ramp down detection</li> <li>• Log generator</li> <li>• Retentive memory</li> </ul>
<b>EDM/output components</b>	
<ul style="list-style-type: none"> <li>• EDM</li> <li>• Valve monitoring</li> </ul>	<ul style="list-style-type: none"> <li>• Fast shut-off with bypass</li> <li>• Fast shut-off</li> </ul>
<b>Muting/presses</b>	
<ul style="list-style-type: none"> <li>• Sequential muting</li> <li>• Parallel muting</li> <li>• Cross muting</li> <li>• Contact monitor, universal presses</li> <li>• Single-stroke press</li> </ul>	<ul style="list-style-type: none"> <li>• Press set-up</li> <li>• Press automatic mode</li> <li>• Cycle mode</li> <li>• Eccentric press contact monitor</li> </ul>
<b>Others</b>	
<ul style="list-style-type: none"> <li>• Mode selection switch</li> <li>• Emergency stop</li> <li>• Switch evaluation</li> <li>• Solenoid switch</li> </ul>	<ul style="list-style-type: none"> <li>• Light grid evaluation</li> <li>• Two-hand, type IIIA</li> <li>• Two-hand, type IIIC</li> <li>• Multi-two-hand</li> </ul>
<b>User-defined function blocks</b>	
<ul style="list-style-type: none"> <li>• Grouped function block</li> </ul>	<ul style="list-style-type: none"> <li>• User-defined function block</li> </ul>

The samosPRO system can support up to 300 function blocks in a given application. The response time is influenced by the number of function blocks. Therefore, you should keep the number function blocks in your application as low as possible.

## 7.3 Function block properties

Function blocks have a host of different properties that you can utilize. The configurable parameters are different depending on the function block. By clicking on the function block, you can access the configurable parameters on the right-hand of the page within the window and select the desired properties (info, parameters, I/O configuration). The following example shows the function block switch evaluation with the Parameters tab open.

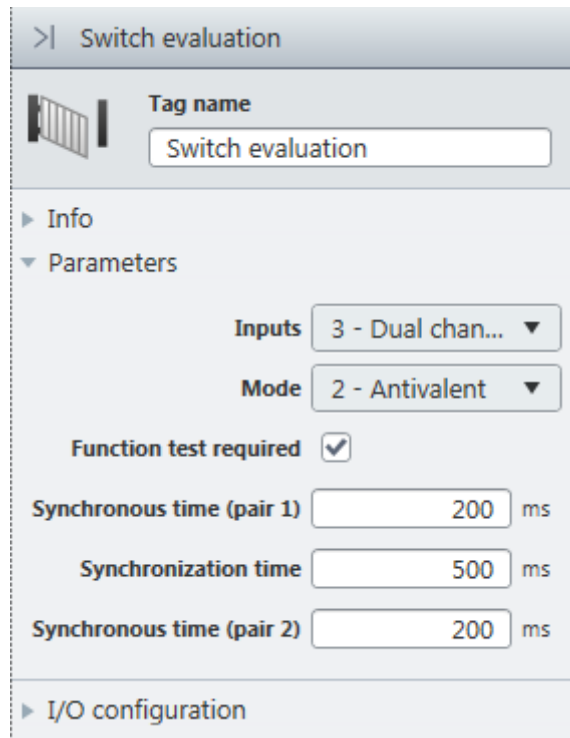


Illustration 57: Configurable parameters of function blocks

You can find the configurable parameters, depending on the respective function block, on the **Parameters** and **I/O configuration** tabs. The **Tag name** field makes it possible for you to replace the specified description of the function block with your own description and to add a name or descriptive text to the function block, which will be displayed in the logic editor under the function block. The description of the inputs and outputs can be changed by clicking on the respective connection of the function block. You can find a general description and the index of the function block under **Info**.

## 7.4 Input and output signal connections of function blocks

### 7.4.1 Function block input connections

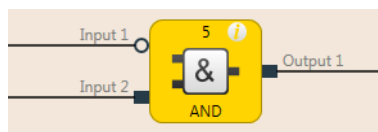
Potential sources for function block inputs are all of the input elements that are listed in the selection tree of the inputs of the logic as well as the output of the function blocks.

### 7.4.2 Inverting inputs

The inputs of some function blocks can be configured in an inverted manner. 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, click on the icon for the function block and mark the desired input in the right-hand window of the page as inverted under I/O configuration.

Inverted inputs are displayed with a small white circle.



*Illustration 58: An example of an AND function block with an inverted input*

Function blocks with invertible inputs include, among other things, the following function blocks:

<ul style="list-style-type: none"><li>• AND</li><li>• OR</li><li>• XOR</li></ul>	<ul style="list-style-type: none"><li>• RS flip-flop</li><li>• JK flip-flop</li><li>• T flip-flop</li></ul>
--	---


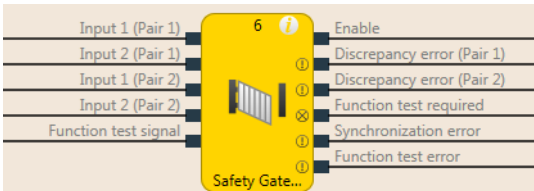
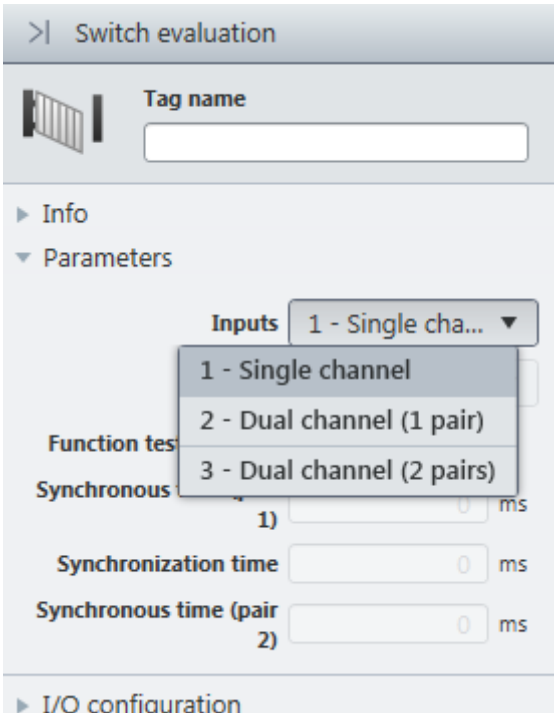
## 7.4.3 Output connections of function blocks

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 multiple subordinate function blocks or to multiple output elements. The behavior of the outputs is explained in the description of the individual function blocks.

You can select whether errors and diagnostic outputs are displayed. In the configuration basic settings of the function blocks, only the output release and a few other outputs are selected. (e.g. reset necessary). In order to display other error outputs and diagnostic outputs, increase the number of outputs on the I/O Settings tab for function block properties.

Table 26: I/O configuration of the function block, switch evaluation

<p>Switch evaluation function block with configuration basic setting</p>	
<p>Switch evaluation function block with all available inputs and outputs</p>	
<p>I/O settings tab of the properties dialog for the switch evaluation function block</p>	

## 7.5 Parameterization of function blocks

Except for the type of input (e.g. single-channel, two-channel equivalent etc.), function blocks can have additional parameters, which are defined on the property page of the function blocks shown above.

### 7.5.1 Time values and logic execution time

**NOTICE** When selecting time monitoring for synchronous time, pulse duration, muting time, etc., the following should be noted.

The times

- must be greater than the logic execution time and
- have an accuracy of  $\pm 10$  ms, plus the logic execution time, during the evaluation.

The logic execution time depends on the number and type of function blocks used. It is a multiple of 4 ms. If the logic execution time used exceeds 100%, it is increased by 4 ms. The logic execution time is displayed in the logic editor. It has an accuracy of  $\pm 100$  ppm (parts per million).

### 7.5.2 Error outputs

Various function blocks have one or more error outputs. These error outputs are either present

- by default
- or
- appear automatically at the function block as soon as the corresponding parameters (e.g. synchronous time) are set
- or
- appear at the function block once they have been selected in the parameterization of the function block.

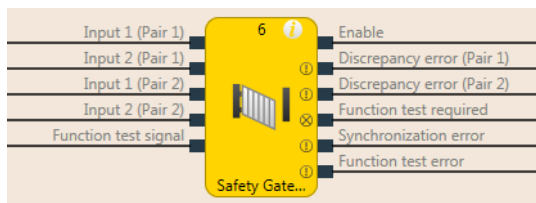


Illustration 59: Error output

The error outputs are high if an error has been detected based on the configured function block parameter (e.g. synchronous time error, function test error, synchronization error, etc.). If an error output is high, the main output (e.g. the output release) is low.

The error outputs are low when the respective error has been reset. The conditions for resetting an error are described in the section for the respective function block.

## 7.6 Logical function blocks

### 7.6.1 NOT

#### Function block diagram

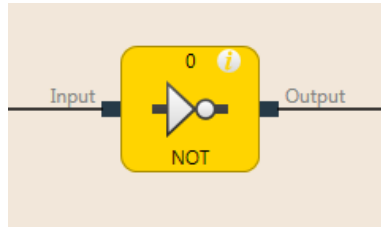


Illustration 60: Function block diagram for the NOT function block

#### General description

The inverted value of the input appears at the output. When the input, for example, is high, then the output is low.

#### Truth table

The following applies to the truth table in this section:

0 means logical low

1 means logical high

#### Truth table for NOT

Table 27: Truth table for the NOT function block

Input	Output
0	1
1	0

### 7.6.2 AND

#### Function block diagram

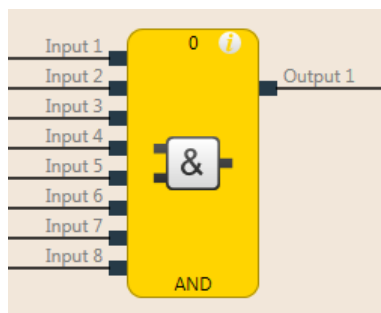


Illustration 61: Function block diagram for the AND function block

#### General description

The output is high when all of the evaluated inputs are high. Up to eight inputs are evaluated.

**Example:** When eight emergency stop buttons are connected at the inputs of the function block, then the output is low as soon as one of the emergency stop buttons is pressed.

## Parameters of the function block

Table 28: Parameters of the AND function block

Parameters	Possible values
Number of inputs	2 to 8
Inverting input x	Any input of this function block can be inverted.

Further information: *Inverting inputs* [ch. 7.4.2, p. 75]

## Truth table

The following applies to the truth tables in this section:

0 means logical low

1 means logical high

x means "any" = 0 or 1

## Truth table for AND evaluation with one input

Table 29: Truth table for AND evaluation with one input

Input 1	Output
0	0
1	1

## Truth table for AND evaluation with two inputs

Table 30: Truth table for AND evaluation with two inputs

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

## Truth table for AND evaluation with eight inputs

Table 31: Truth table for AND evaluation with eight inputs

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

## 7.6.3 OR

### Function block diagram

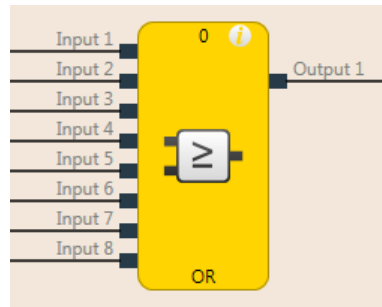


Illustration 62: Function block diagram for the OR function block

### General description

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

Example: When eight light curtains are connected at the inputs of the function block, then the output is high as soon as at least one of the light curtains is free.

### Parameters of the function block

Table 32: Parameters of the OR function block

Parameters	Possible values
Number of inputs	2 to 8
Inverting input x	Any input of this function block can be inverted.

Further information: *Inverting inputs* [ch. 7.4.2, p. 75]

### Truth table

The following applies to the truth tables in this section:

0 means logical low

1 means logical high

x means "any" = 0 or 1

### Truth table for OR evaluation with one input

Table 33: Truth table for OR evaluation with one input

Input 1	Output
0	0
1	1

### Truth table for OR evaluation with two inputs

Table 34: 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

Table 35: Truth table for OR evaluation with eight inputs

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

### 7.6.4 XOR (exclusive OR)

#### Function block diagram

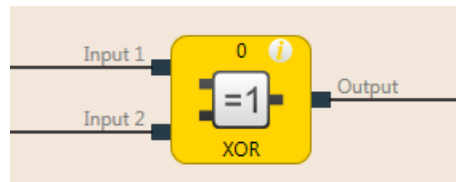


Illustration 63: Function block diagram for the XOR function block (exclusive OR)

#### General description

The output is only high if the two inputs are discrepant (i.e. have opposite values; one input high and one input low).

#### Truth table

The truth table uses the following designations:

0 means logical low

1 means logical high

#### Truth table for XOR evaluation

Table 36: Truth table for XOR evaluation

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

## 7.6.5 T flip-flop

### Function block diagram

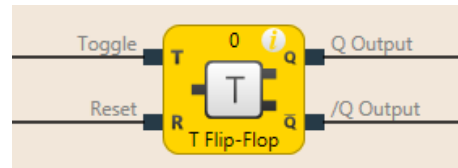


Illustration 64: Function block diagram for the T flip-flop function block

### General description

The Q output switches its state at the **Toggle** input with any rising edge. The reset input resets the Q output, regardless of the state at the **Toggle** input.

### Truth table

The truth table uses the following designations:

- 0 means logical low
- 1 means logical high
- "n-1" relates to the previous value
- "n" relates to the current value
- x means "any" = 0 or 1
- "/" stands for the negated value

### Truth table for T flip-flop evaluation

Table 37: Truth table for T flip-flop evaluation

Toggle	Reset	Output $Q_{n-1}$	Output $/Q_{n-1}$	Output $Q_n$	Output $/Q_n$
0 → 1	0	0	1	1	0
0 → 1	0	1	0	0	1
1 → 0	0	$Q_{n-1}$	$/Q_{n-1}$	$Q_n$	$/Q_n$
x	1	x	x	0	1

## 7.6.6 RS flip-flop

### Function block diagram



Illustration 65: Logical connections for the RS flip-flop function block

### General description

The RS flip-flop function block saves the last value of the **Set** and **Reset** inputs. It is used as a simple memory cell. **Reset** has a higher priority than **Set**. If **Set** was most recently high, output **Q** is high and output **/Q** (Q inverted) is low. If input **Reset** was most recently high, output **Q** is low and output **/Q** is high.

### Parameters of the function block

Table 38: Parameters of the RS flip-flop function block

Parameters	Possible values
Inverting Set	Any input of this function block can be inverted.
Inverting Reset	

Further information: *Inverting inputs [ch. 7.4.2, p. 75]*

### Truth table for the RS flip-flop function block

The following applies to the truth table in this section:

- 0 means logical low
- 1 means logical high
- "n-1" relates to the previous value
- "n" relates to the current value
- x means "any" = 0 or 1
- "/" stands for the negated value

Table 39: Truth table for the RS flip-flop function block

Set	Reset	Output $Q_{n-1}$	Output $Q_n$	Output $/Q_n$
0	0	0	0	1
0	0	1	1	0
0	1	x	0	1
1	0	x	1	0
1	1	x	0	1

## 7.6.7 JK flip-flop

### Function block diagram



Illustration 66: Logical connections for the JK flip-flop function block

### General description

The JK flip-flop function block has three inputs. Inputs **J** and **K** only act upon the outputs if a rising edge is detected at the **Clock** input. In this case ...

- If the **J** input is high and the **K** input is low, the **Q** output is high and the **/Q** output (= **Q** converted) is low
- If the **J** input is low and the **K** input is high, the **Q** output is low and the **/Q** output is high
- If both inputs are low, outputs **Q** and **/Q** retain the last value.
- If both inputs are high, the outputs switch over, i.e. their last values are inverted.

### Parameters of the function block

Table 40: Parameters of the JK flip-flop function block

Parameters	Possible values
Number of outputs	2 (Q and /Q)
Inverting J	Any input of this function block can be inverted.
Inverting cycle	
Inverting K	

Further information: *Inverting inputs [ch. 7.4.2, p. 75]*

### Truth table for the RS flip-flop function block

The following applies to the truth table in this section:

- 0 means logical low
- 1 means logical 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" relates to the previous value
- "n" relates to the current value
- x means "any" = 0 or 1
- "/" stands for the negated value

**NOTICE** The following truth table applies to a configuration of the JK flip-flop function block without inverted inputs.

Table 41: Truth table for the JK flip-flop function block

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

## Logic programming – Function blocks

J	K	Clock	Output $Q_{n-1}$	Output $Q_n$	Output / $Q_n$
0	1	↑	0	0	1
0	1	↑	1	0	1
1	0	↑	0	1	0
1	0	↑	1	1	0
1	1	↑	0	1	0
1	1	↑	1	0	1

## 7.6.8 Clock generator

### Function block diagram

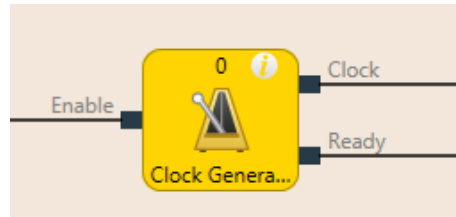


Illustration 67: Logical connections for the clock generator function block

### General description

The clock generator function block makes it possible to generate a pulsed signal. When the **Release** input is high, the **Clock** output pulses from low to high and back to low, according to the parameter settings of the function block.

The **Ready** output indicates with a pulse (duration: 1 logic cycle) the end of the cycle generation.

The pulse generator function block has 3 different operating modes, which determine the period, pause, and sequence parameters.

Endless cycle

Parameter setting: Pulses in a sequence = 0, pauses in a sequence = any, sequences to be generated = any

Endless sequence

Parameter setting: Pulses in a sequence  $\neq 0$ , pauses in a sequence  $\neq 0$ , sequences to be generated = 0

Sequence run

Parameter setting: Pulses in a sequence  $\neq 0$ , pauses in a sequence  $\neq 0$ , sequences to be generated  $\neq 0$

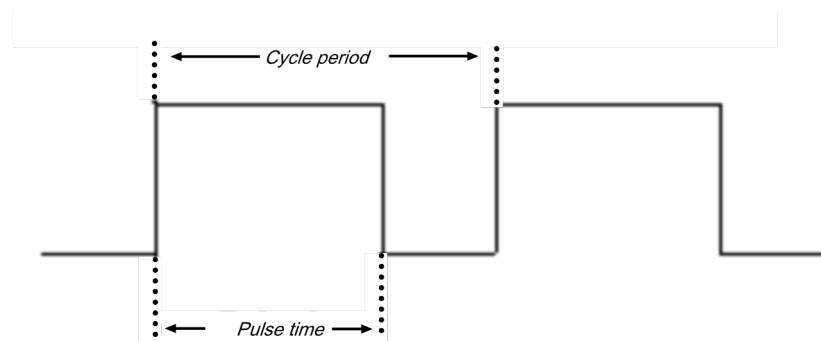


Illustration 68: Parameter diagram for the "Clock generator" function block

Pulse time < cycle period (cycle duration)

Pulse time and cycle period are configured as a multiple of the logic execution time

## Parameters of the function block

Table 42: Parameters of the "Clock generator" function block

Parameters	Possible values
Stop mode	<ul style="list-style-type: none"> <li>• Immediately after removal of the start signal</li> <li>• After current cycle period of the last cycle</li> <li>• After current sequence</li> <li>• After completion of all sequences</li> </ul>
Duration of high pulse	4 to 65535 ms
Duration of low pulse	4 to 65535 ms
Pulses in a sequence	0 to 65535
Pauses in a sequence	0 to 65535
Sequences to be generated	0 to 65535

## Sequence/timing diagram

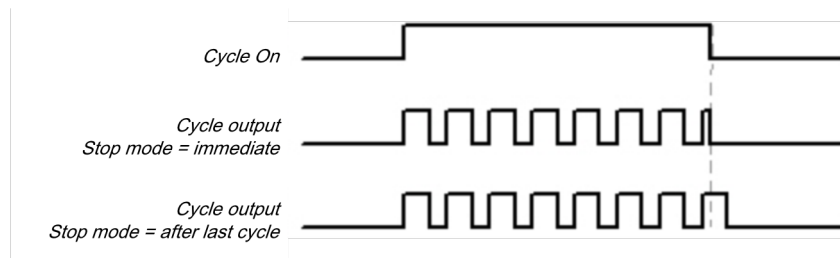


Illustration 69: Timing diagram for the clock generator function block

## 7.6.9 Event counter (upward, downward, up- and downward)

### Function block diagram

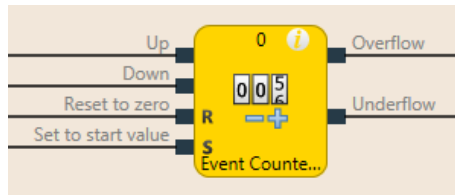


Illustration 70: Logical connections for the event counter function block (upward and downward)

### General description

The event counter function blocks make it possible to count events, either upward and/or downward, in order to show them when a preset threshold value is reached at the **Upper threshold** output or when zero is reached at the **Lower threshold** output. Depending on the required counting direction, there are Up counter, Down counter, and Up and Down counter function blocks.

### Parameters of the function block

Table 43: Parameters of the Up counter, Down counter, and Up and Down counter function blocks

Parameters	Possible values
Reset to zero after upper threshold exceeded	<ul style="list-style-type: none"> <li>Manually</li> <li>Automatically</li> </ul>
Set to start value after lower threshold reached	<ul style="list-style-type: none"> <li>Manually</li> <li>Automatically</li> </ul>
Upper threshold value	Integer between 1 and 65535 The upper threshold value must be greater than or equal to the reset value.
Start value	Integer between 1 and 65535
Min. pulse time for reset to zero	<ul style="list-style-type: none"> <li>100 ms</li> <li>350 ms</li> </ul>
Min. pulse time for set to start value	<ul style="list-style-type: none"> <li>100 ms</li> <li>350 ms</li> </ul>

### Up and down inputs

A rising edge (low to high) at the **Up** input increases the value of the internal event counter by 1.

A rising edge (low to high) at the **Down** input decreases the value of the internal event counter by 1.

If a rising edge (low to high) occurs at both the **Up** input and the **Down** input (only relates to the Up and Down counter function block), then the value of the internal event counter remains unchanged.

### Reset to zero

A valid pulse sequence with a transition of low to high to low at the **Reset to zero** input sets the internal event counter to "0". This occurs regardless of whether the **Upper threshold value** was reached or not and likewise regardless of whether **Reset to zero after upper threshold** was configured with the **Manual** setting or **Automatic**.

The **Minimum pulse time for reset to zero** determines the minimum duration of the pulse at the **Reset to zero** input. Valid values are 100 ms and 350 ms. If the pulse duration is less than the configured minimum pulse time or longer than 30 seconds, then the pulse is ignored.



### Set to start value

A valid pulse sequence with a transition from low to high to low at the **Set to start value** input sets the internal event counter to the configured value of the **Start value** parameter. This occurs regardless of whether **Set to start value after lower threshold** was configured with **Manual** setting or **Automatic**.

The **Minimum pulse time for reset to start time** determines the minimum duration of the pulse at the **Reset to start value** input. Valid values are 100 ms and 350 ms. If the pulse duration is less than the configured minimum pulse time or longer than 30 seconds, then the pulse is ignored.



### Make sure that the transitions of the signals for reset to zero and for set to start value meet the requirements.

When there is 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 as a result of short-circuit detection. If this type of pulse can lead to a hazard-inducing state in the machine, then the following points should be noted:

- Ensure that the cable routing is protected for the signal lines (due to cross-connection with other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

### Upper threshold value and reset to zero after reaching upper threshold

The **Reset to zero after upper threshold** parameter determines what occurs when the event counter reaches the **Upper threshold** value. If this parameter is configured as **Automatic** and the internal event counter is equal to the **Upper threshold value**, the **Upper threshold** output will be at high for the duration of the logic execution time. Following this, the value of the internal event counter will be reset to zero.

When the **Reset to zero after upper threshold** parameter has been configured as **Manual** and the **Upper threshold value** has been reached, the **Upper threshold** output is set at high and remains on high until the event counter value changes, either through a downward count due to a valid pulse sequence at the **Reset to zero** input, or due to a valid pulse sequence at the **Set at start value** input when the start value is less than the upper threshold value. Until then, all of the additional "Upward" count pulses will be ignored.

### Start value and set to start value after lower threshold

The **Set at start value after lower threshold** parameter determines what occurs when the event counter reaches the zero value. If this parameter is configured as **Automatic** and the internal event counter is equal to zero, the **Lower threshold** output will be at high for the duration of the logic execution time. Subsequently, the value of the internal event counter will be set at the configured **Start value**.

If the **Set to start value after lower threshold** parameter is configured as **Manual** and the lower limit, i.e. zero, has been reached, the **Lower threshold** output is set at high and remains on high until the event counter value changes, either due to a count upwards or due to a valid pulse sequence at the **Set to start value** input. Until then, all of the additional "Downward" count pulses will be ignored.

### Truth table for the Up, Down, and Up/Down counter function blocks

The following applies to the truth table in this section:

- 0 means logical low
- 1 means logical high
- "↑" means that a rising edge has been detected at the signal input
- "↓" means that a falling edge has been detected at the signal input
- "n-1" relates to the previous value
- "n" relates to the current value
- "Y" relates to the value of the internal event counter
- "X" means "any" For example, the Reset to zero and Set to start value inputs have priority over the Up and Down inputs.

## Logic programming – Function blocks

Table 44: Truth table for the Up, Down, and Up/Down counter function blocks

Up	Down	Reset to zero	Set to start value	Event counter value $n-1$	Event counter value $n$	Upper threshold $n$	Lower threshold $n$
↑	0, 1, or ↓	0	0	Y	Y+1	0	0
↑	0, 1, or ↓	0	0	Y	Y+1 = upper threshold value	1	0
↑	0, 1, or ↓	0	0	Y = upper threshold value	Y = upper threshold value	1	0
0, 1, or ↓	↑	0	0	Y	Y-1	0	0
0, 1, or ↓	↑	0	0	Y	Y-1 = 0	0	1
0, 1, or ↓	↑	0	0	Y = 0	Y = 0	0	1
↑	↑	0	0	Y	Y	0	0
X	X	1	0	Y	Reset to zero	0	0
X	X	0	1	Y	Set to start value	0	0
X	X	1	1	Y	Reset to zero	0	0

## 7.6.10 Fast shut-off and fast shut-off with bypass

### Function block diagram

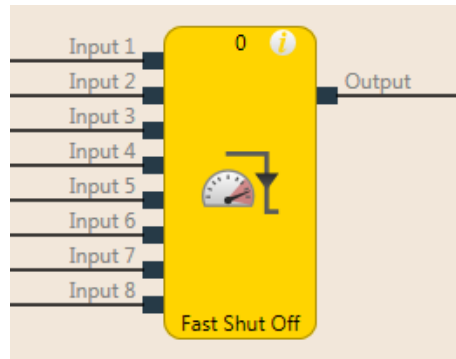


Illustration 71: Logical connections for the fast shut-off function block

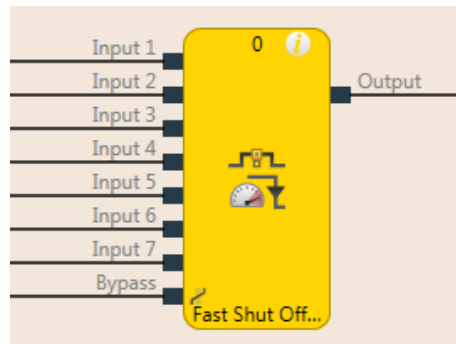


Illustration 72: Logical connections for the fast shut-off with bypass function block

### 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 samosPRO system. To this end both the inputs and the outputs of the switching path are connected to the same input/output module (i.e. SP-SDIO or SD-COP).

The fast shut-off function blocks, regardless of the logic execution time and the logic actually provided for the output switching, cause an immediate shutdown of an output.

The fast shut-off with bypass makes it possible to temporarily bypass the fast shut-off function using the bypass input.

**NOTICE** The fast shut-off with bypass function block is available in SP-SDIO modules with module versions B-01 or higher.

**NOTICE** 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 constructed such that a switch-off of the output of the fast shut-off function block always causes a direct switch-off of the physical output. Typically, the AND, Restart, or EDM function blocks can be used in the signal string. An OR function block on the other hand will not comply with this rule.



**ATTENTION**

### Consider the overall response time of the entire safety function.

The response time of the fast shut-off function block is not the same as the overall response time of the entire safety function. The overall response time includes multiple parameters outside of this function block. You can find a description of how you can calculate the overall response time of the samosPRO system in the "samosPRO Hardware" manual (BA000966).

## Parameters of the function block

Table 45: Parameters of the fast shut-off function block

Parameters	Possible values
Number of inputs	Fast shut-off: 1 to 8 Fast shut-off with bypass: 1 to 7
Fast shut-off output	All outputs of that I/O module whose inputs are also used for the input signals.

### This is how you configure the fast shut-off function block:

The following example shows the function with three light barriers connected to a fast shut-off function block.

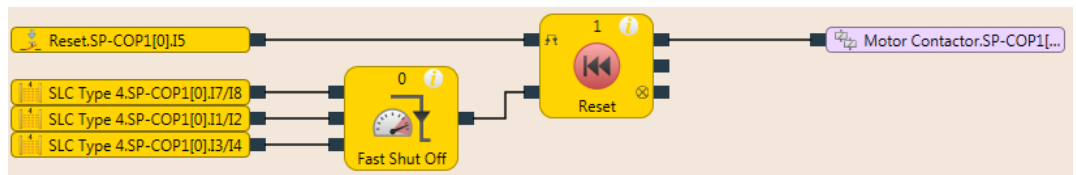


Illustration 73: Configuration example for a fast shut-off with three light barriers

### NOTICE

#### Type and source of danger

The input signals that are intended to trigger the fast shut-off function (i.e. the input signals of the FB) must be wired in the logic such that they alone could also shut down the fast shut-off output.



**ATTENTION**

#### Type and source of danger

The output on which the fast shut-off function block acts must only be configured using the parameters of the function block. The state output of the function block is only used to indicate the function of the FB or to process it further in the logic.

In order to configure the fast shut-off function block, proceed as follows:

- ➔ Connect the input elements to the function block. Double-click on the function block in order to open the configuration dialog, and then click on the **I/O Settings** tab.

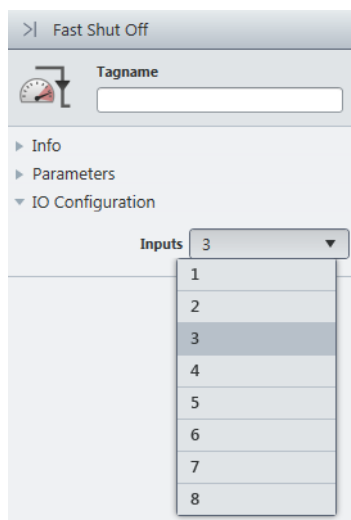


Illustration 74: I/O Settings for the fast shut-off function block

- ➔ Select the number of inputs you wish to connect to the function block.

- ➔ Then click on the **Parameters** tab and select the zone in which you want to activate the corresponding checkbox.

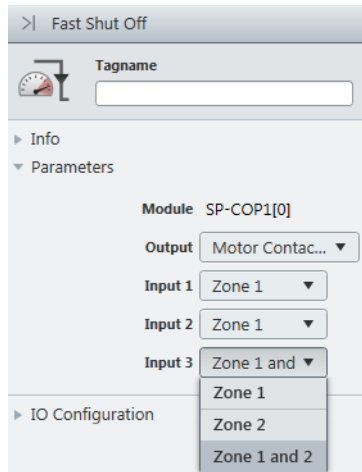


Illustration 75: Parameter settings for the fast shut-off function block

### NOTICE

Inputs for the fast shut-off function block can act upon one or even two different zones. The inputs within a zone always involve an AND link. The results of AND links of the inputs of the two zones are OR linked.

If inputs are only supposed to act on zone 1, then you do not need the zone 2 or zone 1 and zone 2 parameters.

If inputs of an application are supposed to act upon both protection zones, then the zone 1, zone 2, or zone 1 and zone 2 parameters are parameterized according to the function of the inputs.

- ➔ Finally, select the output for fast shut-off.

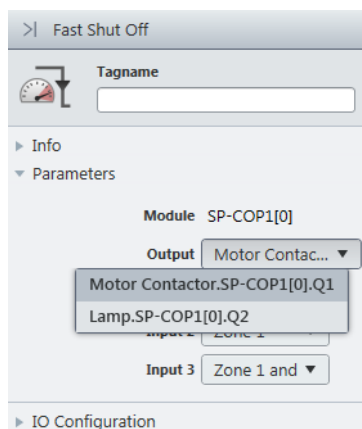


Illustration 76: Select output for fast shut-off

The selected inputs and outputs will then be connected with one another such that the outputs in the hardware configuration can no longer be moved to a different position and the inputs must remain connected at the same module. The elements connected in this manner will be displayed in the hardware configuration in orange.

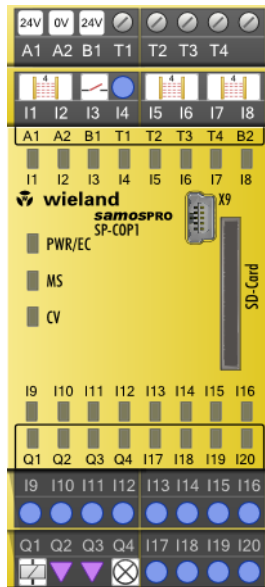


Illustration 77: View of the inputs and outputs connected with fast shut-off in the hardware configuration

These connections are canceled 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, for example in a safe setup mode for a machine, in which the machine can only be operated in jog mode. To do this, the fast shut-off with bypass function block is available. It is used and configured precisely 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. When the **Bypass** input is high, the fast shut-off with bypass function block is bypassed.



**Make sure that the system or the machine is in a safe state when using the bypass function.**

As long as the bypass function is active, a stop condition, such as the violation of a protective field, will **not** lead to a switch-off of the machine. You must ensure that other protective measures are absolutely effective during the bypass, such as the safe setup mode of the machine, so that the machine cannot endanger people or parts of the system during the bypass.



**Note the longer response time when deactivating the bypass.**

When the **Bypass** input is deactivated while a switch-off condition exists, then the outputs will not be switched-off until after the normal response time of the application. The minimum response time for fast shut-off does not apply to the **Bypass** input. Consider this for your risk analysis and avoidance strategy. Otherwise, there is a hazard for the operator of the machine.

### Info

- Contrary to 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 that also can be moved to a different module in the hardware configuration.
- The **Bypass** input has a switch-on delay of three logic cycles in order to compensate for delays due to the processing time of the logic and the transmission time of the SBUS+. This delay ensures that the corresponding module has received the bypass signal before it is used for further logic processing in the fast shut-off function block. As a result of this delay, the **Bypass** input must be at high for three logic cycles beforehand in order to bypass the fast shut-off successfully. Once this condition is fulfilled, then the fast shut-off function block output and the physical output at the I/O module will remain at high.
- The fast shut-off directly switches off the output connected to it of the corresponding module while the following logic programming is ignored. Therefore, it is not possible to program additional bypass conditions in the logic editor between the output of the fast shut-off function block and the module output connected to it.

## Logic programming – Function blocks

- Note that the value of the connected module output in the online monitor can deviate from the actual value of the physical output of the corresponding module. For example, the connected output may be low due to the following logic, while the output of the fast shut-off function block and the physical output of the module are high because the **Bypass** input is high.
- If your application requires that it be possible to switch off the output of the module independently of an existing bypass condition (e.g. emergency stop), then the underlying logic must be implemented such that the respective switch-off signal (e.g. emergency stop) also switches off the **Bypass** input of the function block, as shown in the following example:

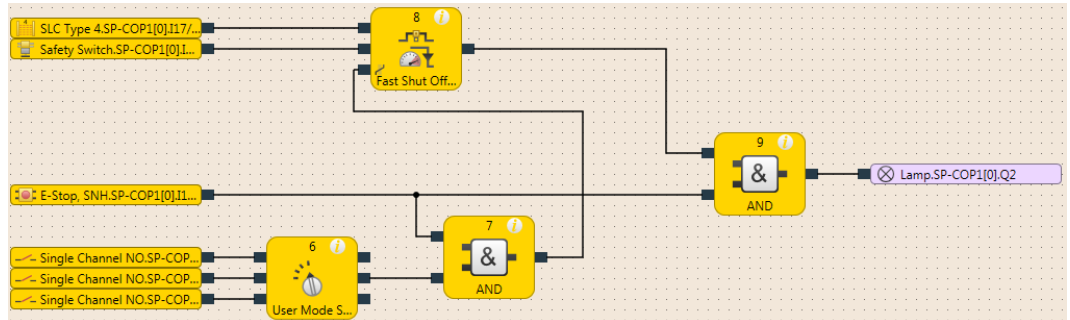


Illustration 78: Example of a fast shut-off with bypass with more than one condition for bypass

## 7.6.11 Edge detection

### Function block diagram

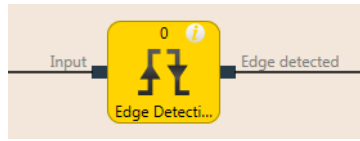


Illustration 79: Logical connections for the edge detection function block

### General description

The edge detection function block makes it possible to detect a positive (rising) or negative (falling) edge of the input signal. The function block can then be configured to detect a positive edge, a negative edge, or both. When an edge is detected according to the parameter settings, the **Edge detected** output will be at high for the duration of the logic execution time.

### Parameters of the function block

Table 46: Parameters of the edge detection function block

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

### Sequence/timing diagram

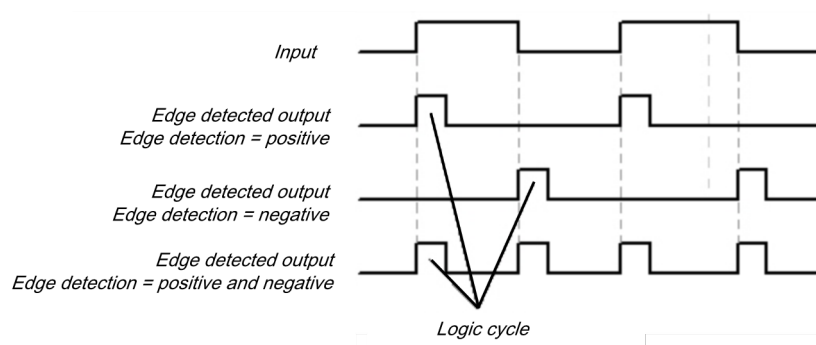


Illustration 80: Timing diagram for the edge detection function block



## 7.6.12 Binary encoder

### Function block diagram

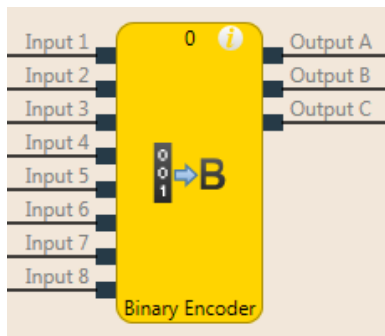


Illustration 81: Logical connections for the binary encoder function block

### General description

The binary encoder function block converts one-hot coding or greatest value code into binary code, as a function of the current configuration (output A =  $2^0$ , output B =  $2^1$ , output C =  $2^2$ ). 2 to 8 inputs can be configured. The number of outputs is determined by the number of inputs. An **Error flag** output is available as an option.

### Parameters of the function block

Table 47: Parameters of the binary encoder function block

Parameters	Possible values
Number of inputs	2 to 8
Coding mode	<ul style="list-style-type: none"> <li>• Peak</li> <li>• Greatest value</li> <li>• Greatest value with input 1 dominant</li> </ul>
Use error flag	<ul style="list-style-type: none"> <li>• With</li> <li>• Without</li> </ul>

### Peak

In **Peak** mode, only one input can be high at any time. The outputs are set as a function of the index (input 1 = 1, input 2 = 2, etc.) of this input. Once all of the inputs are at low or once more than one input is simultaneously set at high, all of the outputs are set at low and the **Error flag** output is set at high.

### Greatest value

In **Greatest value** mode, multiple inputs can be set at high simultaneously. The outputs are set as a function of the respective input thereof with the highest index (input 1 = 1, input 2 = 2, etc.). Once all of the inputs are simultaneously at low, all of the outputs are set at low and the **Error flag** output is set at high.

### Greatest value with input 1 dominant

In this mode, all the outputs are low when input 1 is high. The other inputs are not considered here. If input 1 is low, the function block will behave as in **Greatest value** mode. Once all of the inputs are simultaneously at low, all of the outputs are set at low and the **Error flag** output is set at high.

## Truth tables for the binary encoder function block

The following applies to the truth tables in this section:

- 0 means logical low
- 1 means logical high
- x means "any" = 0 or 1

Table 48: Truth table for the binary encoder function block with 2 inputs in peak mode

Input 2	Input 1	Output A	Error flag
0	0	0	1
0	1	0	0
1	0	1	0
1	1	0	1

Table 49: Truth table for the binary encoder function block with 8 inputs in peak mode

Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1	Out-put C	Out-put B	Out-put A	Error flag
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	1	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	1	0	1	0
0	1	0	0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	0	0	1	1	1	0
More than one input = 1								0	0	0	1

Table 50: Truth table for the binary encoder function block with 2 inputs in greatest value mode

Input 2	Input 1	Output A	Error flag
0	0	0	1
0	1	0	0
1	x	1	0

Table 51: Truth table for the binary encoder function block with 8 inputs in greatest value mode

Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1	Out-put C	Out-put B	Out-put A	Error flag
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

## Logic programming – Function blocks

Table 52: Truth table for the binary encoder function block with 2 inputs in greatest value mode with input 1 dominant.

Input 2	Input 1	Output A	Error flag
0	0	0	1
x	1	0	0
1	0	1	0

Table 53: Truth table for the binary encoder function block with 8 inputs in greatest value mode with input 1 dominant.

Input 8	Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1	Out-put C	Out-put B	Out-put A	Error flag
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



**Evaluate the error flag when the binary encoder function block is set for security purposes.**

When you use the binary encoder function block for safety-relevant logic, then you may have to evaluate the **Error flag** output. This is the only way to determine whether only input 1 is high or whether an invalid input state exists. In both cases, all outputs are low.

## 7.6.13 Binary decoder

### Function block diagram

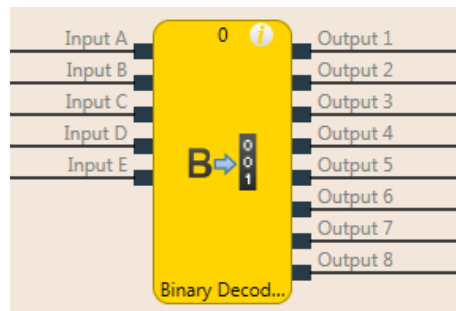


Illustration 82: Logical connections for the binary decoder function block

### General description

The binary decoder function block converts binary code into one-hot code or into greatest value code as a function of its current configuration. Up to five inputs can be configured. The number of outputs is determined by the number of inputs. The evaluation of inputs A, B, and C enables the decoding of binary codes with decimal values of 0 to 7 with a single binary decoder function block (input A =  $2^0$ , input B =  $2^1$ , input C =  $2^2$ ). With optional inputs D and E, it is possible to combine up to four binary decoders in order to decode binary codes with decimal values of 0 to 31.

### Parameters of the function block

Table 54: Parameters of the binary decoder function block

Parameters	Possible values
Coding mode	<ul style="list-style-type: none"> <li>• Peak</li> <li>• Level</li> </ul>
Inputs	<ul style="list-style-type: none"> <li>• Not inverted</li> <li>• Inverted</li> </ul>
Number of inputs	1 to 5
Value range	<ul style="list-style-type: none"> <li>• 0-7</li> <li>• 8-15 (only available when more than 3 inputs are used)</li> <li>• 16-23 (only available when 5 inputs are used)</li> <li>• 24-31 (only available when 5 inputs are used)</li> </ul>

#### Peak

In **Peak** mode, only the output whose number corresponds to the current input values is high.

#### Level

In **Level** mode, the output is high whose number corresponds to the current input values as well as all outputs with low numbers.

#### Inputs inverted/not inverted

With the assistance of this parameter, it is possible to invert all inputs.

## Truth tables for the binary decoder function block

The following applies to the truth tables in this section:

- 0 means logical low
- 1 means logical high

Table 55: Truth table for the binary decoder function block with 1 input in peak mode

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

Table 56: Truth table for the binary decoder function block with 2 inputs in peak mode

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

Table 57: Truth table for the binary decoder function block with 3 inputs in peak mode

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

Table 58: Truth table for the binary decoder function block with 1 input in level mode

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

Table 59: Truth table for the binary decoder function block with 2 inputs in level mode

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

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

## Logic programming – Function blocks

Input C	Input B	Input A	Out-put 8	Out-put 7	Out-put 6	Out-put 5	Out-put 4	Out-put 3	Out-put 2	Out-put 1
0	1	0	0	0	0	0	0	1	1	1
0	1	1	0	0	0	0	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1
1	0	1	0	0	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

### Evaluating more than three inputs

If 4 or 5 inputs are used, up to four binary decoders can be combined in order to decode binary codes with values of 0 to 31.

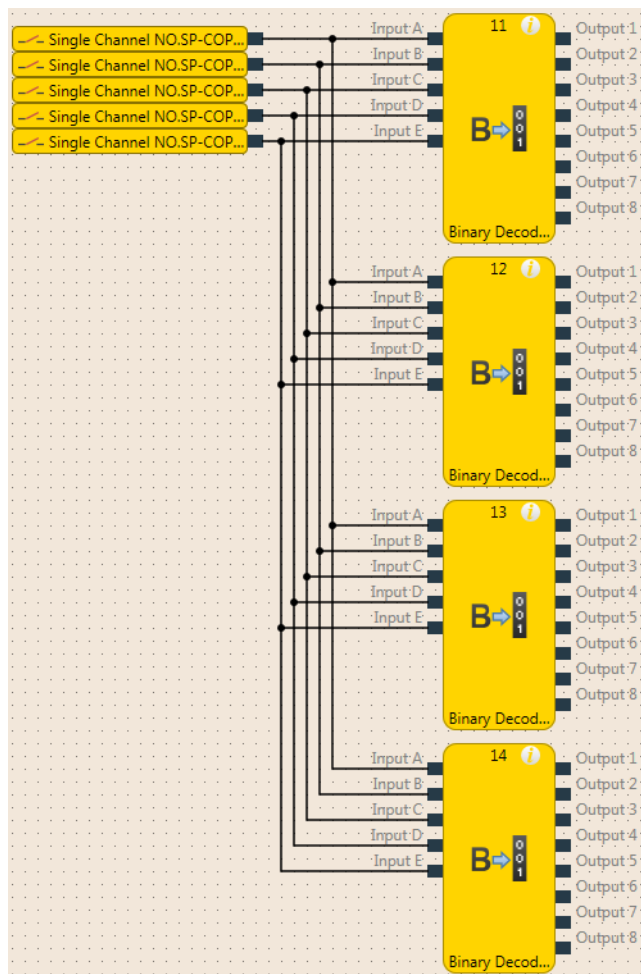


Illustration 83: Combination of four binary decoders

If you combine multiple binary decoders, you must configure the value range that should be covered for each of these function blocks with the assistance of the **Value range** option. This range is determined by the value of inputs D and E.

Table 61: The value range of the binary decoder function block depends on input D.

Input D	Outputs
0	0-7
1	8-15

Table 62: The value range of the binary decoder function block depends 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

- When **Input D** and **Input E** have the same value as the **Value range** parameter (e.g. when **Input E** = 1, **Input D** = 0, and the **Value range** is set at 16-23), the function block behaves as shown in the truth tables above, depending on the value of inputs A, B, and C and on the configured coding mode (peak or level).
- When **Input D** and **Input E** have a **lower** value than the **Value range** parameter (e.g. when **Input E** = 0, **Input D** = 1, and the **Value range** = 16-23), then all of the outputs are low, regardless of the configured coding mode (peak or level).
- When **Input D** and **Input E** have a **higher** value than the **Value range** parameter (e.g. **Input E** = 1, **Input D** = 1, and **Value range** = 16-23) ...
  - all outputs are low in peak mode,
  - all outputs are high in level mode.

## 7.6.14 Log generator

### Function block diagram

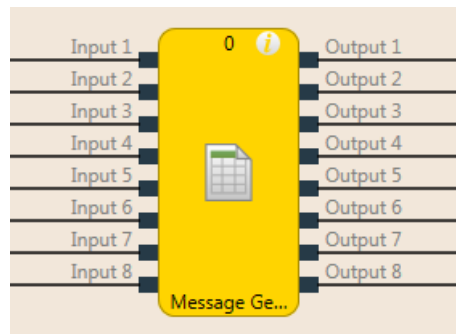


Illustration 84: Function block diagram for the log generator function block

### General description

The log generator function block evaluates up to eight inputs. If an edge is detected at one of these inputs 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 diagnostic history. This text message can be read in online mode with the assistance of the samosPLAN5+ software diagnostic function.

Further information: "Diagnostics" view [ch. 5.9, p. 65]

### NOTICE

You can generate no more than 10 messages within 3 seconds with the log generator function components. Use text that is as brief as possible.

### Parameters of the function block

Table 63: Parameters of the log generator function block

Parameters	Possible values
Number of inputs	1 to 8
Messages	Up to 64 user-definable messages per project.
Input condition	<ul style="list-style-type: none"> <li>Rising edge</li> <li>Falling edge</li> </ul>

### This is how you can configure the log generator function block:

The following example shows the log generator function block with an emergency stop button, a safety light curtain, and a safety switch connected.

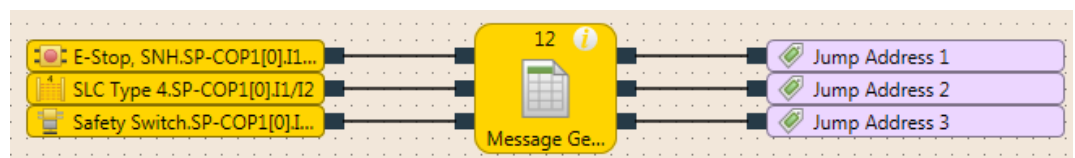


Illustration 85: Configuration example of a log generator with two emergency stop buttons and one safety switch



## Logic programming – Function blocks

In order to configure the log generator function block, proceed as follows:

- ➔ Connect the input elements to the function block. Double-click on the function block in order to open the configuration dialog, and then click on the **I/O Settings** tab.

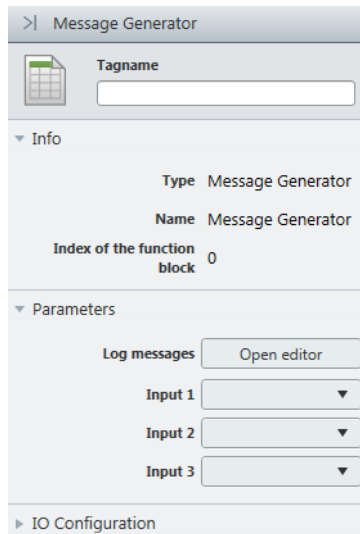


Illustration 86: I/O settings for the log generator function block

- ➔ Select the number of inputs you wish to connect to the function block.
- ➔ Then click on the **Parameters** tab, open the **Editor** under the **Log messages**, and enter the messages that should be generated in the diagnostics.

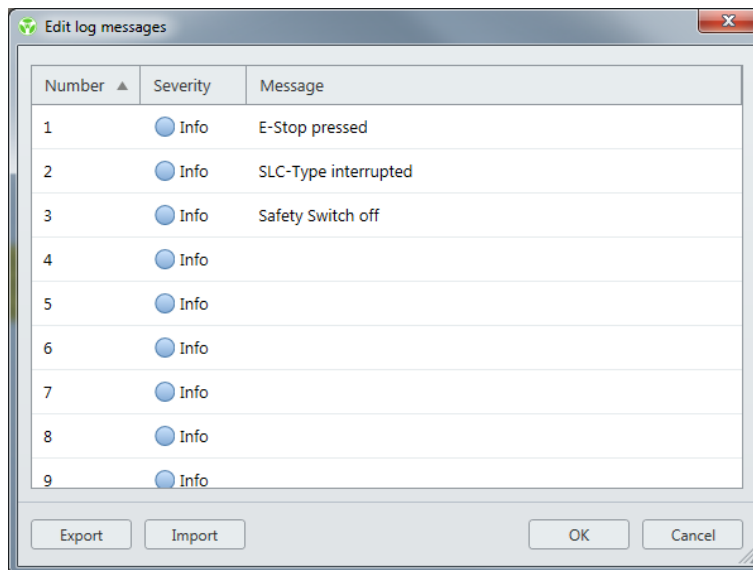


Illustration 87: Available messages in the log generator function block

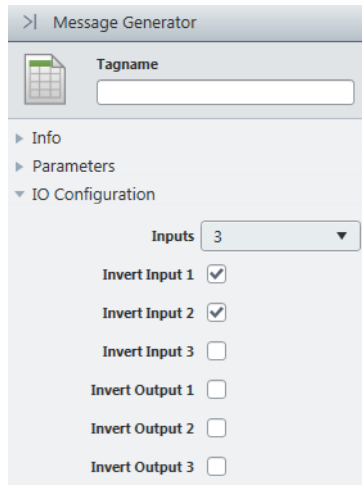
### NOTICE

- The messages entered will apply to all log generator function blocks used throughout a project.
- You can enter up to 64 different messages with a length of up to 110 characters each per project. To internally save umlauts and special characters, they need two or three characters in UTF-8 format so that the displayed character string is correspondingly shorter.
- Using the **Export** and **Import** button to the lower left in this window, you can save messages as a text file in CSV format (Comma Separated Values) or import messages from a CSV file.

- ➔ Click on the **I/O Configuration** tab.  
Assign the desired message to each input used here and select the input condition for each input (by activating/deactivating the input inverting) that when fulfilled will trigger the generation of the respective message (rising edge or falling edge).

### NOTICE

The message assignment cannot be exported or imported.



*Illustration 88: Assignment of the input conditions for the log generator function block*

### Priority of messages

If more than one condition is met simultaneously, the following priorities will apply:

- When there is a single log generator function block, the input with the lowest number has priority, i.e. the message generated from this input is logged first.
- If there are multiple log generator function blocks being used, the function block with the lowest function block index has priority, i.e. the messages generated from this function block are logged first.

## 7.6.15 Residual memory

### Function block diagram

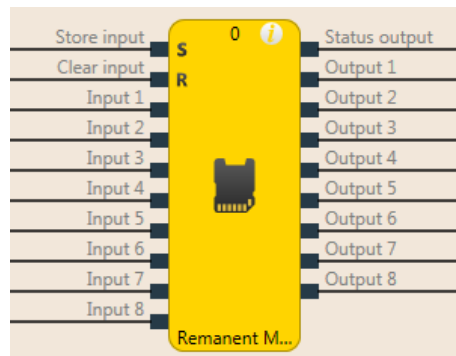


Illustration 89: Function block diagram for the retentive memory function block

### General description

The retentive memory function block can store up to 8 bits in non-volatile memory in the SP-COP module. The value present at the inputs is stored by means of a high signal at the S memory input in the function block.

### Saving and deleting data

If **Memory input S** is high, after it was previously low, the data present at the inputs is stored retentively and is present at the outputs. If **Memory input S** is once again low, no new data can be saved and the most recently saved data will be present at the outputs and will thus also be once again available after a power failure and restoration of the power.

If **Memory input S** is at high up to the point of failure of the power supply, the data present at the inputs will be permanently stored in the retentive memory. After power is restored, the outputs will be updated with the most recently saved data, even if memory input S is at high. There will be no saving of new data until **Memory input S** is once again at low and then switches back to high. This functionality can be used to construct, for example, a retentive error memory in that an error bit can only be deleted through manual acknowledgment (interruption of a high signal at memory input S).

The content of the retentive memory and its outputs can be reset by means of a high signal at **Reset input R** if memory input S is simultaneously at high (low-high edge after power-up or stop → run).

This function block can be used a maximum of 2 times in a project.

### NOTICE

- In order to save the desired data in the event of a power failure, it must be ensured that the store signal reaches the low state before the data to be saved.
- The pre-processed signal is relevant as an input signal for all inputs. If an input signal has the logical level "1" and there is a test pulse error, a "0" is saved. The same applies to the memory and reset input.

### Parameters of the function block

Table 64: Parameters of the retentive memory function block

Parameters	Possible values
Number of inputs	3 to 10
Number of outputs	2 to 9

## 7.7 Application-specific function blocks

### 7.7.1 Reset

#### Function block diagram

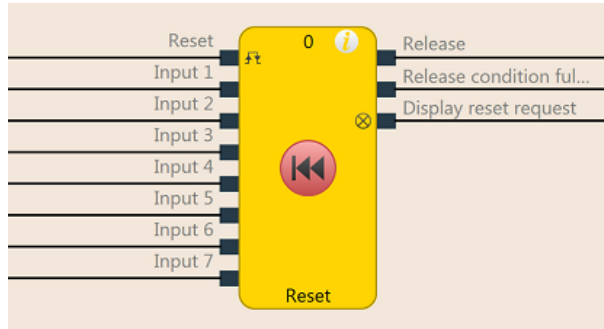


Illustration 90: Function block diagram for the reset function block

#### General description

The reset function block can be used to fulfill the standard requirements of safety applications for acknowledging a manual safety stop and the subsequent prompt for restart of the application. Typically, each safety logic function of a samosPRO flexible safety control contains a reset function block.

#### Parameters of the function block

Table 65: Parameters of the reset function block

Parameters	Possible values
Min. reset pulse time	<ul style="list-style-type: none"> <li>• 100 ms</li> <li>• 350 ms</li> </ul>
Number of inputs	2 to 8 (= 1 to 7 release inputs activated)

#### Release condition fulfilled output

The **Release condition fulfilled** output indicates the result of an AND link of all activated **Release** inputs. It is high when all activated **Release** inputs are high.

#### Reset required output

The **Reset required** output indicates, by pulsing at 1 Hz, that the function block is expecting a valid reset pulse at the **Reset** input, so that the **Release** output can go to high. This is the case when the **Release condition fulfilled** output is at high, i.e. all activated **Release** inputs are high, but the **Release** output is still low. Typically, this output is used to actuate an indicator lamp.

## Release output

The **Release** output is at high when the **Release condition fulfilled** output is high and a valid reset pulse has been detected at the **Reset** input, provided all activated **Release** inputs remain high.

The **Min. reset pulse time** determines the minimum duration of the pulse at the **Reset** input. Valid values are 100 ms and 350 ms. If the pulse duration is less than the configured minimum pulse time or longer than 30 seconds, then the pulse is ignored.

The **Release** output is low when one or more **Release** inputs go to low.



**ATTENTION**

### Make sure that the transitions of the signals for reset meet the requirements!

When there is 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 as a result of short-circuit detection. If this type of pulse can lead to a hazard-inducing state in the machine, then the following points should be noted:

- Ensure that the cable routing is protected for the signal lines (due to cross-connection with other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

## Sequence/timing diagram

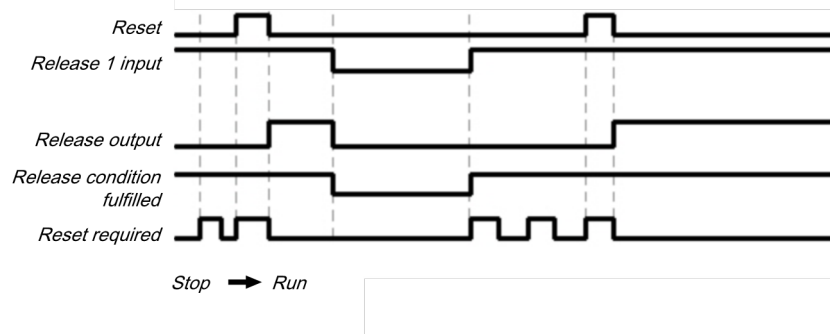


Illustration 91: Sequence/timing diagram for the reset function block

## 7.7.2 Restart

### Function block diagram

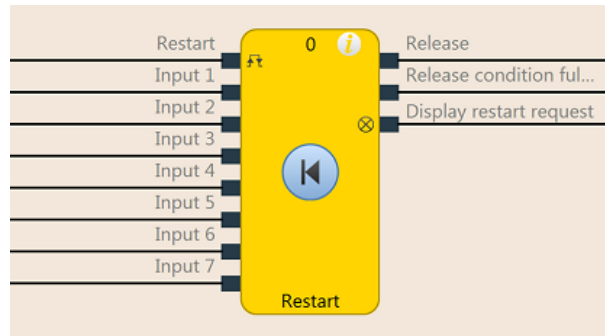


Illustration 92: Function block diagram for the restart function block

### General description

The internal logic of the restart function block has the same function as that of the reset function block. The restart function block enables graphical differentiation of the function blocks while adhering to application standards for acknowledging a manual restart request.

### Parameters of the function block

Table 66: Parameters of the restart function block

Parameters	Possible values
Min. restart pulse time	<ul style="list-style-type: none"> <li>• 100 ms</li> <li>• 350 ms</li> </ul>
Number of inputs	2 to 8 (= 1 to 7 release inputs activated)

### Release condition fulfilled output

The **Release condition fulfilled** output indicates the result of an AND link of all activated **Release** inputs. It is high when all activated **Release** inputs are high.

### Restart required output

The **Restart required** output indicates, by pulsing at 1 Hz, that the function block is expecting a valid restart pulse at the **Restart** input, so that the **Release** output can go to high. This is the case when the **Release condition fulfilled** output is at high, i.e. all activated **Release** inputs are high, but the **Release** output is still low. Typically, this output is used to actuate an indicator lamp.

### Release output

The **Release** output is at high when the **Release condition fulfilled** output is high and a valid restart pulse has been detected at the **Restart** input, provided all activated **Release** inputs remain high.

The **Min. restart pulse time** determines the minimum duration of the pulse at the **Restart** input. Valid values are 100 ms and 350 ms. If the pulse duration is less than the configured minimum pulse time or longer than 30 seconds, then the pulse is ignored.

The **Release** output is low when one or more **Release** inputs go to low.



**ATTENTION**

### Make sure that the transitions of the signals for restart meet the requirements!

When there is 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 as a result of short-circuit detection. If this type of pulse can lead to a hazard-inducing state in the machine, then the following points should be noted:

- Ensure that the cable routing is protected for the signal lines (due to cross-connection with other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

### Sequence/timing diagram

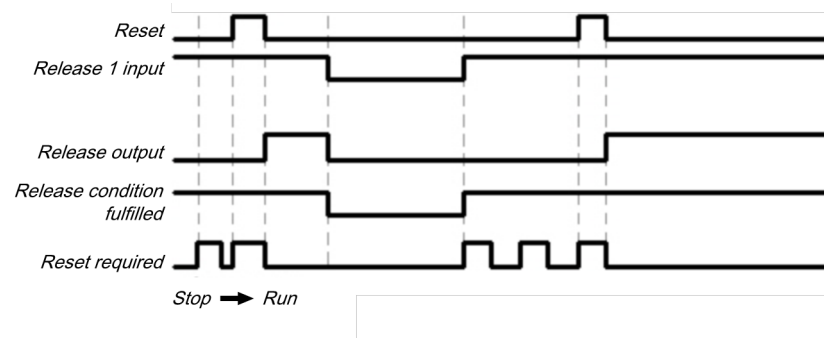


Illustration 93: Sequence/timing diagram for the restart function block

## 7.7.3 Switch-off delay timer

### Function block diagram



Illustration 94: Function block diagram for the "switch-off delay timer" function block

### General description

The "switch-off delay timer" function block delays the switch-off of the **Release** output for a configurable length of time.

### Parameters of the function block

Table 67: Parameters of the "switch-off delay timer" function block

Parameters	Possible values
Switch-off delay time	0 to 300 seconds in 10 ms increments. If the value is not 0, it must be greater than the logic execution time.

The timer starts with the delay sequence upon a transition of the input from high to low. If the timer has expired after the configured time, the **Release** output is likewise low, provided the input continues to be low. If the input is high, the **Release** output immediately goes to high and the timer is reset.

### Sequence/timing diagram

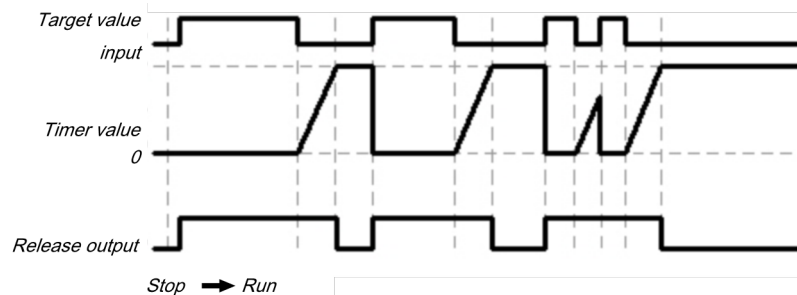


Illustration 95: Sequence/timing diagram for the "switch-off delay timer" function block



## 7.7.4 Adjustable switching-off delay

### Function block diagram

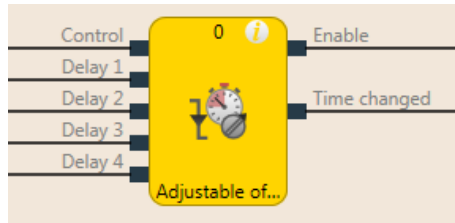


Illustration 96: Function block diagram for the adjustable switching-off delay function block

### General description

The adjustable switching-off delay function block delays the switch-off of the **Release** output for an adjustable length of time. Four individual switch-off delay times, each of which can be activated using a corresponding **Delay** input, can be configured. The total delay is equal to the total of all activated delay times.

### Parameters of the function block

Table 68: Parameters of the adjustable switching-off delay function block

Parameters	Possible values
Delay 1	0 to 600 seconds in 10 ms increments.
Delay 2	If the value is not 0, the corresponding input is activated. In this case, the value must be greater than the logic execution time.
Delay 3	
Delay 4	The total delay (total of all switch-off delay times) is limited to 600 seconds.

The timer starts with the delay sequence when there is a falling edge (high to low) at the **Control input**. If the timer has expired after the selected total delay time, the **Release** output is likewise low, provided the **Control input** continues to be low. If the **Control input** is high, the **Release** output immediately goes to high and the timer is reset.

If one of the **Delay** inputs assumes a different value during an ongoing delay sequence, then the **Time change** output goes to high and remains high until the **Control input** goes back to high.

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

If the **Control input** is low during the first logic cycle after a transition from the stop state to the run state, the **Release** output likewise remains low.

### Sequence/timing diagram

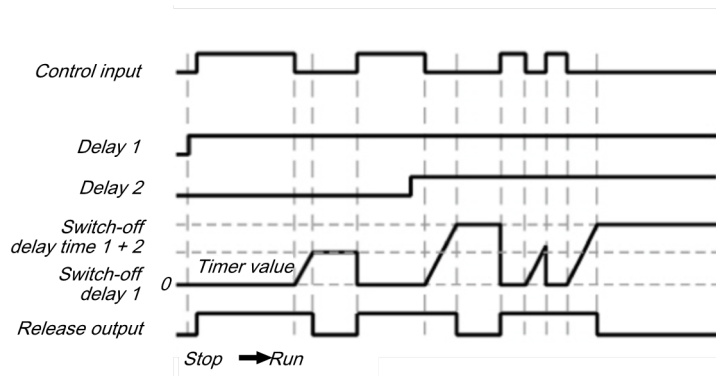


Illustration 97: Sequence/timing diagram for the adjustable switching-off delay function block with switch-off delay time 1 and switch-off delay time 2

## 7.7.5 Switch-on delay timer

### Function block diagram



Illustration 98: Function block diagram for the switch-on delay timer function block

### General description

The switch-on delay timer function block delays the switch-on of the **Release** output for a configurable length of time.

### Parameters of the function block

Table 69: Parameters of the switch-on delay timer function block

Parameters	Possible values
Switch-on delay time	0 to 300 seconds in 10 ms increments. If the value is not 0, it must be greater than the logic execution time.

The timer starts with the delay sequence upon a transition of the input from low to high. If the timer has expired after the configured time, the **Release** output is likewise high, provided the input continues to be high. If the input is low, the **Release** output immediately goes to low and the timer is reset.

### Sequence/timing diagram

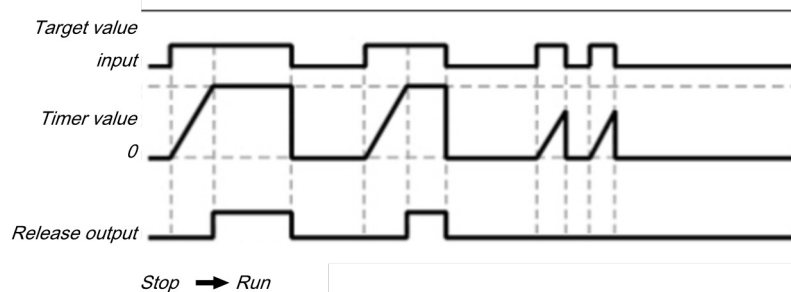


Illustration 99: Sequence/timing diagram for the switch-on delay timer function block

## 7.7.6 Adjustable switching-on delay

### Function block diagram

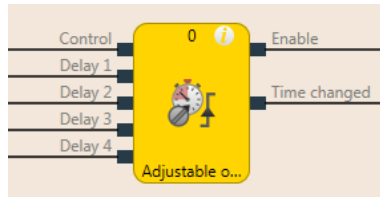


Illustration 100: Function block diagram for the "adjustable switching-on delay" function block

### General description

The "adjustable switching-on delay" function block delays the switch-on of the **Release** output for an adjustable length of time. Four individual delay times, each of which can be activated using a corresponding **Delay** input, can be configured. The total delay is equal to the total of all activated delay times.

### Parameters of the function block

Table 70: Parameters of the "adjustable switching-on delay" function block

Parameters	Possible values
Delay 1	0 to 300 seconds in 10 ms increments. If the value is not 0, it must be greater than the logic execution time.
Delay 2	
Delay 3	
Delay 4	

The timer starts with the delay sequence when there is a rising edge (low to high) at the **Control input**. If the timer has expired after the selected total delay time, the **Release** output is likewise high, provided the **Control input** continues to be high. If the Control input is low, the **Release** output immediately goes to low and the timer is reset.

If one of the **Delay** inputs assumes a different value during an ongoing delay sequence, then the **Time change** output goes to high and remains high until the **Control input** goes back to low.

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

If the **Control input** is high during the first logic cycle after a transition from the stop state to the run state, the **Release** output immediately goes to high without delay.

### Sequence/timing diagram

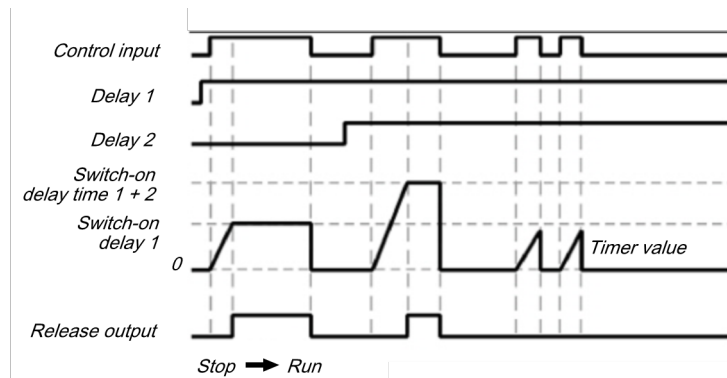


Illustration 101: Sequence/timing diagram for the "adjustable switching-on delay" function block with switch-on delay time 1 and switch-on delay time 2

## 7.7.7 EDM (Contact monitor)

### Function block diagram

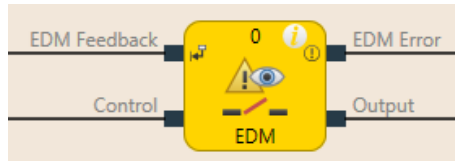


Illustration 102: Function block diagram for the EDM function block

### General description

The EDM (contactor monitor) function block makes it possible to actuate an external device (e.g. a contractor) and to check whether it has switched as expected by means of its feedback signal. The external device is connected to the **Output** for this. The feedback signal is connected to the **EDM read-back signal** input. The **Control input** is connected to the logic signal, which represents the desired state for the external device, e.g. the **Release** output of a reset function block.

After the maximum feedback delay has elapsed, a de-bounce time of 12 ms takes effect. If the EDM read-back signal switches into the wrong state for more than 12 ms, an EDM error occurs.

### Parameters of the function block

Table 71: Parameters of the EDM function block

Parameters	Possible values
Max. feedback delay	100 to 1000 seconds in 10 ms increments. The value must be greater than the logic execution time.

### Output

The output is high when the **EDM read-back signal** is high and then the control input switches from low to high.

The **Output** is low when the **Control input** is **Low** or when an EDM error is pending (**EDM error** output is high).

### EDM errors and error flag

In general, it is expected that the **EDM read-back signal** will always assume the inverted value of the **Control input** within the configured max. feedback delay ( $T_{EDM}$ ).

The **EDM error** outputs and **Error flag** are high when ...

- The **Control input** switches from low to high and the **EDM read-back signal** is low (independently of  $T_{EDM}$ ), or
- The **Control input** switches from low to high and the **EDM read-back signal** does not switch from high to low within  $T_{EDM}$ , or
- The **Control input** switches from high to low and the **EDM read-back signal** does not switch from low to high within  $T_{EDM}$ , or
- The **Control input** is low and the **EDM read-back signal** switches to low for more than 12 ms
- The **Control input** is high and the **EDM read-back signal** switches to high for more than 12 ms

The **EDM error** outputs and **Error flag** are low when a signal sequence has been detected that sets the **Output** to high.

**NOTICE** If you require a delay of signals from the **Output**, then you must implement the output delay with another function block before the EDM function block and not after. Otherwise, this could lead to an EDM error.

## Sequence/timing diagram

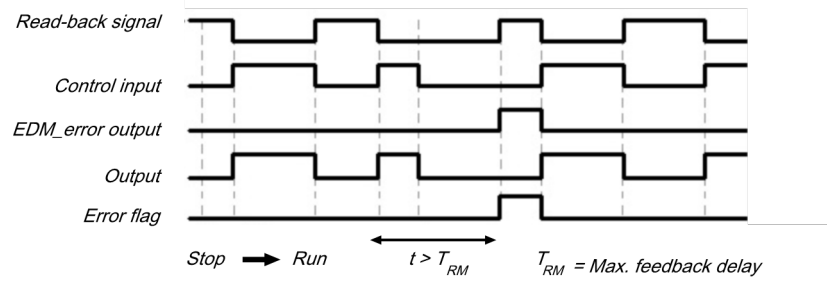


Illustration 103: Sequence/timing diagram for the EDM function block

## 7.7.8 Valve monitoring

### Function block diagram

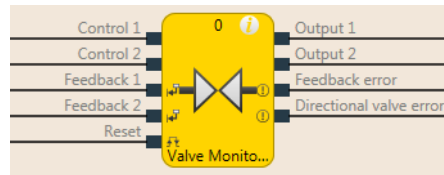


Illustration 104: Logical connections for the valve monitoring function block configured for a directional valve

### General description

The valve monitoring function block makes it possible to actuate valves and to check, using their feedback signals, whether they have switched as expected.

To this end, the valves are connected to **Output 1** to **Output 2**. The feedback signals are connected to the **Read-back 1** and **Read-back 2** inputs. **Control input 1** and **Control input 2** are connected to the logic signal that represents the desired state for the valve, e.g. the **Release** output of a reset function block. Depending on the valve type, some signals will not be required.

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

### Parameters of the function block

Table 72: Parameters of the valve monitoring function block

Parameters	Possible values
Reset condition	<ul style="list-style-type: none"> <li>Manual reset</li> <li>Automatic reset</li> </ul>
Continuous monitoring with active valve	<ul style="list-style-type: none"> <li>Active</li> <li>Inactive</li> </ul>
Valve type	<ul style="list-style-type: none"> <li>Single valve (control output 1, output 1, read-back 1 activated)</li> <li>Double valve (control input 1, output 1, read-back-1, output 2, read-back 2 activated)</li> <li>Directional valve (control input 1, output 1, read-back-1, control input 2, read-back 2, directional error activated)</li> </ul>
Max. switch-on feedback delay	50 ms to 10 s in 10 ms increments (0 = inactive, only with CPU module version B-01 or higher). If this parameter is deactivated, then the continuous monitoring option with active valve must also be deactivated. If the value is not 0, it must be greater than the logic execution time.
Max. switch-off feedback delay	50 ms to 10 s in 10 ms increments (0 = inactive, only with CPU module version B-01 or higher). If this parameter is activated, the value must be greater than the logic execution time.
Min. reset pulse time	<ul style="list-style-type: none"> <li>100 ms</li> <li>350 ms</li> </ul>
Use error flag	<ul style="list-style-type: none"> <li>With</li> <li>Without</li> </ul>



### Connect the read-back signals correctly!

The signals for **Read-back 1** and **Read-back 2** must be protected against short-circuits at the signals for outputs (e.g. **Output 1** and **2**) and against short-circuits among each other (e.g. through protective wiring or wiring these signals exclusively within the switchbox).

### Output 1 to output 2

**Output 1** and/or **Output 2** is high when the corresponding **Read-back 1** and/or **Read-back 2** input is high and then the corresponding **Control input** switches from low to high.

**Output 1** and/or **Output 2** is low when the corresponding **Control input** is low or when an error is pending (**Read-back error** output is high or **Directional error** output is high).

The corresponding control input for **Output 1** is always **Control input 1**.

The corresponding control input for **Output 2** depends upon the configured valve type.

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

### Read-back error, directional error, and error flag

In general, it is expected that the **Read-back 1/2** input will always assume the inverted value of the corresponding **Control input** within the configured max. switch-on feedback delay ( $T_{ON}$ ) or max. switch-off feedback delay ( $T_{OFF}$ ).

The **Read-back error** output is high when ...

- The **Control input** switches from low to high and the corresponding **Read-back signal** is low (independently of  $T_{ON}$  and  $T_{OFF}$ ), or
- $T_{ON}$  is greater than zero and the **Control input** switches from low to high and the corresponding **Read-back signal** does not switch from high to low within  $T_{ON}$ , or
- $T_{OFF}$  is greater than zero and the **Control input** switches from high to low and the corresponding **Read-back signal** does not switch from low to high within  $T_{OFF}$ , or
- **Continuous monitoring with active valve** is active and the **Control input** is high and the corresponding **Read-back signal** is switched to high.

The **Directional error** output is high when the **Valve type** parameter is = **Directional valve** and **Control input 1** and **Control input 2** are high at the same time.

The **Error flag** output is high when the **Read-back error** and/or **Directional error** is high.

The **Read-back error**, **Directional error**, and **Error flag** outputs are low when all activated control inputs are low and all activated read-back inputs are high. When **Manual reset** is configured as a reset condition, then a valid reset pulse must additionally be implemented at the **Reset** input.

The **Min. reset pulse time** determines the minimum duration of the pulse at the **Reset** input. Valid values are 100 ms and 350 ms. If the pulse duration is less than the configured minimum pulse time or longer than 30 seconds, then the pulse is ignored.



### Make sure that the transitions of the signals for reset meet the requirements of the safety standards and regulations!

When there is 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 as a result of short-circuit detection. If this type of pulse can lead to a hazard-inducing state in the machine, then the following points should be noted:

- Ensure that the cable routing is protected for the signal lines (due to cross-connection with other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

## Sequence/timing diagrams

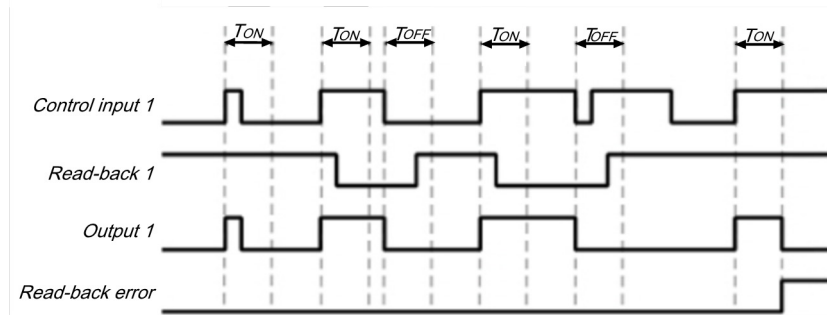


Illustration 105: Sequence/timing diagram for single valve in manual reset mode

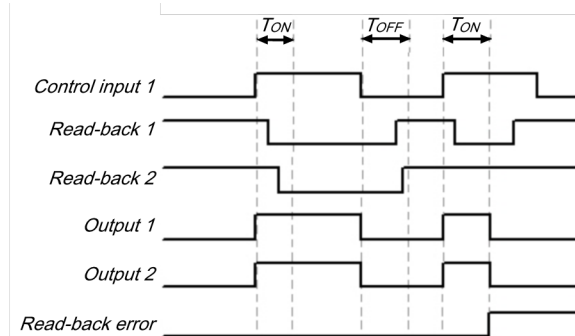


Illustration 106: Sequence/timing diagram for double valve in manual reset mode

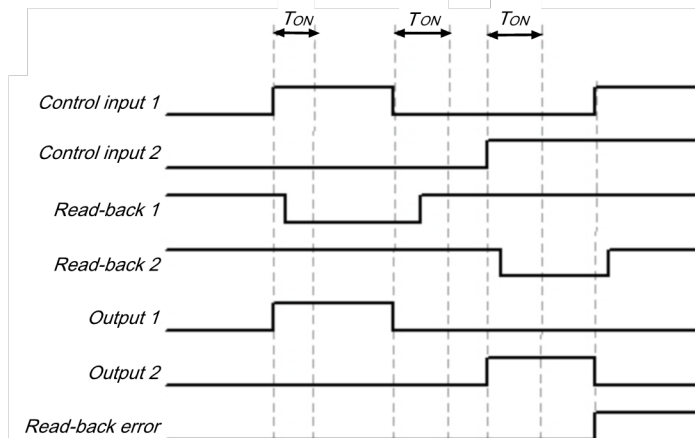


Illustration 107: Sequence/timing diagram for directional valve



## 7.7.9 Operation mode selection switch

### Function block diagram

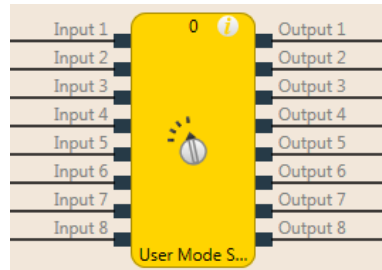


Illustration 108: Function block diagram for the mode selection switch function block

### General description

The mode selection switch function block selects an output as a function of an input value. Output x is high when input x is high.

The function block supports two to eight inputs and the corresponding outputs.

Exactly only one input can be high at any one time. If no input or more than one input is high, then the output that was most recently high remains high for the duration of the set high synchronous time. After the synchronous time has elapsed, the outputs are set to the values defined in the error output combination and the **Error flag** output goes to high.

If there is no valid input combination present during the first logic cycle after the transition from the stop state to the run state, then the outputs are immediately set to the values defined in the error output combination and the **Error flag** output goes to high.

### Parameters of the function block

Table 73: Parameters of the mode selection switch function block

Parameters	Possible values
Synchronous time	0 to 10 seconds in 10 ms increments
Error output combination	Marked outputs are high and unmarked outputs are low when the error flag is high.
Number of inputs and number of outputs	2 to 8
Use error flag	<ul style="list-style-type: none"> <li>• With</li> <li>• Without</li> </ul>

### Truth table for the mode selection switch function block

The truth table uses the following designations:

0 means logical low

1 means logical high

Table 74: Truth table for the mode selection switch function block

Inputs								Outputs								
1	2	3	4	5	6	7	8	Error flag	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

Inputs								Outputs								
1	2	3	4	5	6	7	8	Error flag	1	2	3	4	5	6	7	8
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 configured synchronous time								0	= last output combination							
More than one input high or no input high for longer than the configured synchronous time								1	= error output combination							

### Sequence/timing diagram

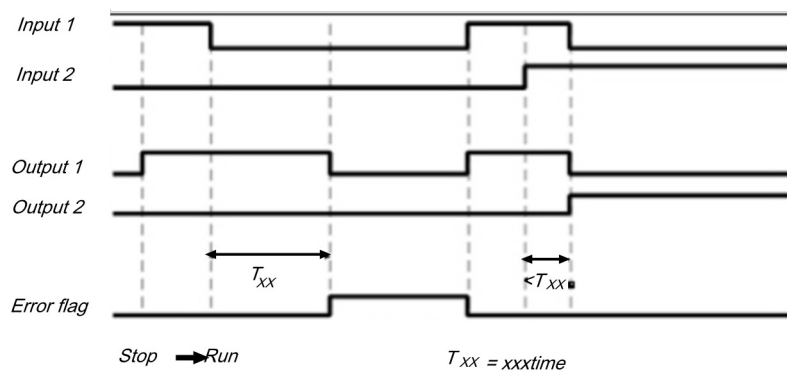


Illustration 109: Sequence/timing diagram for the mode selection switch function block

### Info

- When the inputs of the function block are connected at the inputs of an expansion module that are connected to test outputs, and the faulty input combination is the result of a test pulse error (short-circuit to high) that will lead to a low input value, then the test pulse error must first be reset, e.g. by briefly disconnecting the particular line at the input are at the test output.
- When the inputs of the function block are connected at inputs of an expansion module that are connected to test outputs, then a cross-connection can only be detected between the inputs used when an operating mode is selected that activates one of these inputs.

## 7.7.10 Ramp down detection

### Function block diagram

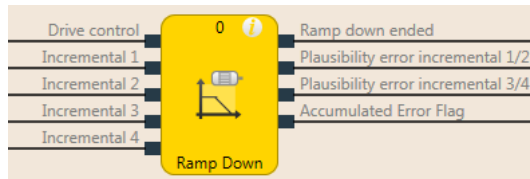


Illustration 110: Logical connections for the ramp down detection function block

### General description

The ramp down detection function block checks whether a connected drive has stopped, i.e. that no pulses have been detected from the pulse encoder system for the duration of a configurable time span (e.g. from an HTL encoder or from proximity switches). A safety door lock can be released, for example, depending on the results of this check.

The ramp down detection is started by a falling edge of the **Drive released** input signal. A stop of the drive is detected when no signal change (rising or falling edge) has taken place for at least the duration of the configured **Min. time between the signal changes** at any **Pulse encoder** input. In this case, the **Post-run ended** output is at high. When the **Drive released** input is high, this sets the **Post-run ended** output immediately to low and ends any currently running ramp down detection as well.

In the drive running state (**Drive released** input is high) and in the stop detected state (**Post-run ended** output is high), the **Pulse encoder** inputs are not monitored for signal changes (see *Figure [ch. 7.7.10, p. 127]*).

The function block enables an optional plausibility check of the **Pulse encoder** inputs in order to detect disconnections in the cabling provided that the pulse encoder provides suitable signals such as complementary outputs or proximity switches and a gearwheel with 270° span and a phase offset of 180°. When the possibility check is active, there must be at least one signal of each signal pair at high at all times. The **Pulse encoder plausibility error** output goes to high when this condition is not fulfilled for the duration of two successive logic cycles. This means that both inputs of a pair may be at low for the duration of the logic execution time without this being evaluated as an error (see *Figure [ch. 7.7.10, p. 127]*).

The **Pulse encoder plausibility error** output is reset at low when at least one signal of a signal pair is high and the **Drive released** input is low.

The **Group error flag** is high when any **Pulse encoder plausibility error** output is high. The **Error flag** output is low when all error outputs are low.

### Parameters of the function block

Table 75: Parameters of the ramp down detection function block

Parameters	Possible values
Number of pulse encoder inputs	<ul style="list-style-type: none"> <li>• Single-channel pulse encoder input</li> <li>• One pair of pulse encoder inputs</li> <li>• Two pairs of pulse encoder inputs</li> </ul>
Input plausibility check	<ul style="list-style-type: none"> <li>• Inactive</li> <li>• Active</li> </ul> <p>If active, then the number of pulse encoder inputs must either be 1 pair or 2 pairs.</p>
Min. time between signal changes	100 ms to 10 seconds in 10 ms increments. The value must be greater than the logic execution time.
Use error flags (input plausibility check active)	<ul style="list-style-type: none"> <li>• With</li> <li>• Without</li> </ul>

**NOTICE**

**Make sure that your application meets the following requirements!**  
 The duration of the pulse encoder signals must be at least as long as the logic execution time (see step 1 below).  
 Connect the signal that controls the physical output for the drive to the **Drive released** input. Make sure that the torque for the drive is switched off under all circumstances when this input is low.  
 The pulse encoder must be locally connected to an SP-COP, SP-SDIO, or SP-SDI module.

**Configuration steps**

- Check the minimum duration of the pulse encoder signals (see step 1 below).
- Determine the time between the signal changes for the speed limit (see step 2 below).

**Step 1: Check the maximum signal frequency of the pulse encoder signals.**

The minimum duration of signals  $t_{high}$  and  $t_{low}$  of the pulse encoder must be greater than the logic execution time. This limits the permissible signal frequency and the pulse encoder speed as a function of the type of pulse encoder. The following figures show typical signal patterns for various types of pulse encoders:

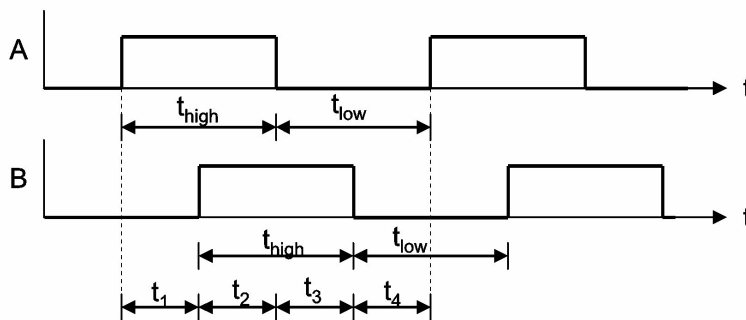


Illustration 111: Signal pattern for A/B pulse encoder with 90° phase offset

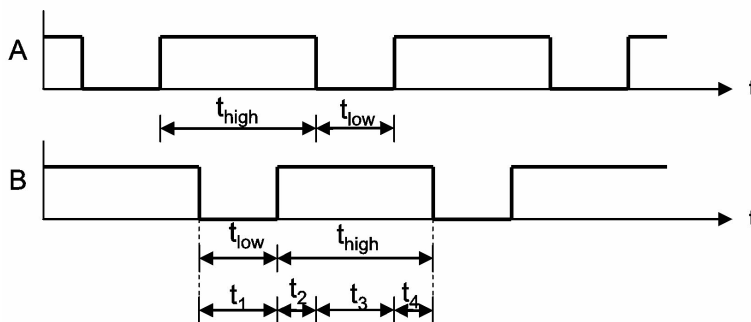


Illustration 112: Signal pattern for 1/3 gap pulse encoder with 180° phase offset

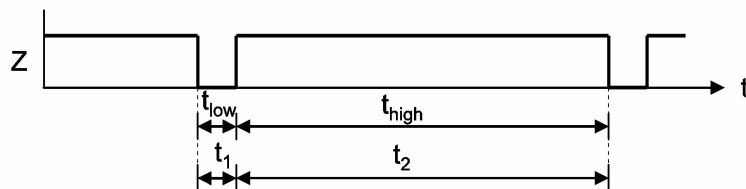


Illustration 113: Signal pattern for individual pulse encoder signal

Based on the design of your system, you must ensure that the minimum duration of pulse encoder signals  $t_{high}$  and  $t_{low}$  are each always greater than the logic execution time. In doing so, consider all of the potential tolerance values, such as switching tolerances, gearwheel tolerances, etc. The following table shows typical values for various types of pulse encoders:

Table 76: Maximum permissible signal frequency and speed (RPM) of pulse encoders, as a function of the type and the logic execution time

Pulse encoder type	Max. permissible pulse encoder signal frequency (Hz) for logic execution time									
	4 ms	8 ms	12 ms	16 ms	20 ms	24 ms	28 ms	32 ms	36 ms	40 ms
A/B, 90° phase offset	120	60	40	30	24	20	17.1	15	13.3	12
1/3 gap <sup>1)</sup>	80	40	26.6	20	16	13.3	11.4	10	8.8	8
1/4 gap <sup>1)</sup>	60	30	20	15	12	10	8.5	7.5	6.6	6
Pulse 180°	120	60	40	30	24	20	17.1	15	13.3	12

1) 180° Phase offset, at least 1 signal always high.

## Step 2: Determine the time between signal changes for the speed limitation

1. Determine the speed at which the **Post-run ended** output should be activated, for example in order to release a safety door.
2. Determine the maximum time between two signal changes at this speed (maximum values of  $t_1$  to  $t_4$ ). In doing so, consider all of the potential tolerance values, such as switching tolerances, gearwheel tolerances, etc.

**Min. time between signal changes = maximum values of  $t_1$  to  $t_4$  + 10 ms**

The **Min. time between single changes** must absolutely be greater than the logic execution time and must be rounded up to the next multiple of 10 ms.



### Note the increased logic execution times!

Any time the logic program is changed, the logic execution time can be increased. In this case, it may be necessary to recalculate the maximum signal frequency of the pulse encoder. Otherwise, there is a hazard for the operator of the machine.

### Example 1: A/B 90° phase offset

- 4 teeth per rotation
- Switching tolerances  $\pm 5^\circ$  → teeth  $175^\circ$  to  $185^\circ$  (corresponds to  $t_{low}$ ,  $t_{high}$ ); signal change  $85^\circ$  to  $95^\circ$  (corresponds to  $t_1$  to  $t_4$ )
- Maximum drive speed = 750 RPM = 12.5 Hz
- Drive speed for release = 15 RPM = 0.25 Hz
- Logic execution time = 8 ms

### Procedure

- ➔ Check the maximum signal frequency of the pulse encoder signals:  
 Max. signal frequency = 12.5 Hz × 4 teeth/rotation = 50 Hz  
 Lowest  $t_{low}$  =  $1/50 \text{ Hz} \times 175^\circ/360^\circ = 9.7 \text{ ms}$   
 → Higher than the logic execution time  
 Lowest  $t_{high}$  =  $1/50 \text{ Hz} \times 175^\circ/360^\circ = 9.7 \text{ ms}$   
 → Higher than the logic execution time
- ➔ Determine the time between signal changes for the speed limitation.  
 Signal frequency for release = 0.25 Hz × 4 teeth/rotation = 1 Hz  
 Max. duration input pattern =  $1/1 \text{ Hz} \times 185^\circ/360^\circ = 514 \text{ ms}$   
 Time between signal changes = 514 ms + 10 ms = 524 ms  
 → Min. time between signal changes = 530 ms (rounded up to the next multiple of 10 ms)

## Example 2: 1/3 gap 180° phase offset

- 8 teeth per rotation
- Switching tolerances  $\pm 2^\circ \rightarrow$  teeth  $118^\circ$  to  $122^\circ$  (corresponds to  $t_{low}$ ,  $t_{high}$ ); signal change  $118^\circ$  to  $122^\circ$  (corresponds to  $t_1$  to  $t_4$ )
- Maximum drive speed = 120 RPM = 2 Hz
- Drive speed for release = 12 RPM = 0.2 Hz
- Logic execution time = 16 ms

### Procedure

- ➔ Check the maximum signal frequency of the pulse encoder signals:  
 Max. signal frequency = 2 Hz  $\times$  8 teeth/rotation = 16 Hz  
 Lowest  $t_{low} = 1/16 \text{ Hz} \times 118^\circ/360^\circ = 20.5 \text{ ms}$   
 $\rightarrow$  Higher than the logic execution time  
 Lowest  $t_{high} = 1/16 \text{ Hz} \times 238^\circ/360^\circ = 41.3 \text{ ms}$   
 $\rightarrow$  Higher than the logic execution time
- ➔ Determine the time between signal changes for the speed limitation.  
 Signal frequency for release = 0.2 Hz  $\times$  8 teeth/rotation = 1.6 Hz  
 Max. duration input pattern =  $1/1.6 \text{ Hz} \times 122^\circ/360^\circ = 212 \text{ ms}$   
 Time between signal changes = 212 ms + 10 ms = 222 ms  
 $\rightarrow$  Min. time between signal changes = 230 ms (rounded up to the next multiple of 10 ms)

## Example 3: Zero pulse 10°

- 1 tooth per rotation
- Switching tolerances  $\pm 1^\circ \rightarrow$  tooth  $9^\circ$  to  $11^\circ$  (corresponds to  $t_{low}$ ,  $t_{high}$ ); signal change  $349^\circ$  to  $351^\circ$  (corresponds to  $t_1$  to  $t_4$ )
- Maximum drive speed = 300 RPM = 5 Hz
- Drive speed for release = 3 RPM = 0.05 Hz
- Logic execution time = 4 ms

### Procedure

- ➔ Check the maximum signal frequency of the pulse encoder signals:  
 Max. signal frequency = 5 Hz  $\times$  1 tooth/rotation = 5 Hz  
 Lowest  $t_{low} = 1/5 \text{ Hz} \times 9^\circ/360^\circ = 5 \text{ ms}$   
 $\rightarrow$  Higher than the logic execution time  
 Lowest  $t_{high} = 1/5 \text{ Hz} \times 351^\circ/360^\circ = 195 \text{ ms}$   
 $\rightarrow$  Higher than the logic execution time
- ➔ Determine the time between signal changes for the speed limitation.  
 Signal frequency for release = 0.05 Hz  $\times$  1 tooth/rotation = 0.05 Hz  
 Max. duration input pattern =  $1/0.05 \text{ Hz} \times 11^\circ/360^\circ = 611 \text{ ms}$   
 Time between signal changes = 611 ms + 10 ms = 621 ms  
 $\rightarrow$  Min. time between signal changes = 630 ms (rounded up to the next multiple of 10 ms)

## Logic example

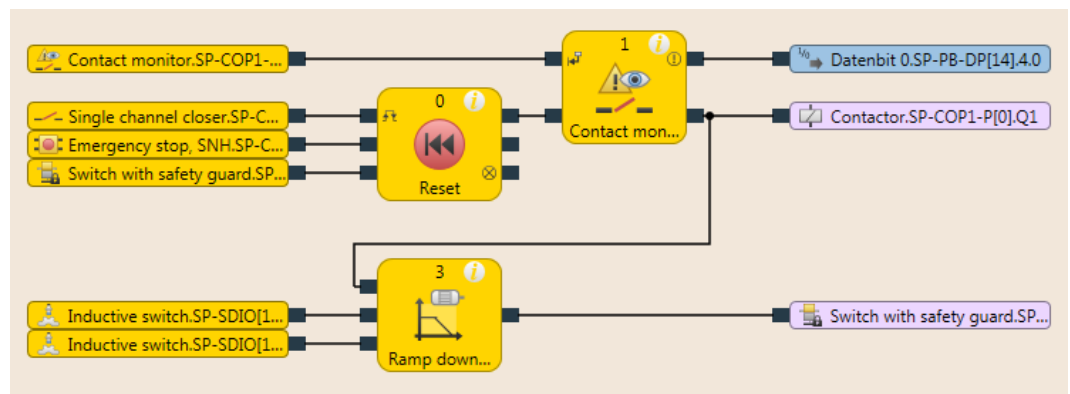


Illustration 114: Logic example of the "ramp down detection" function block

## Sequence/timing diagrams

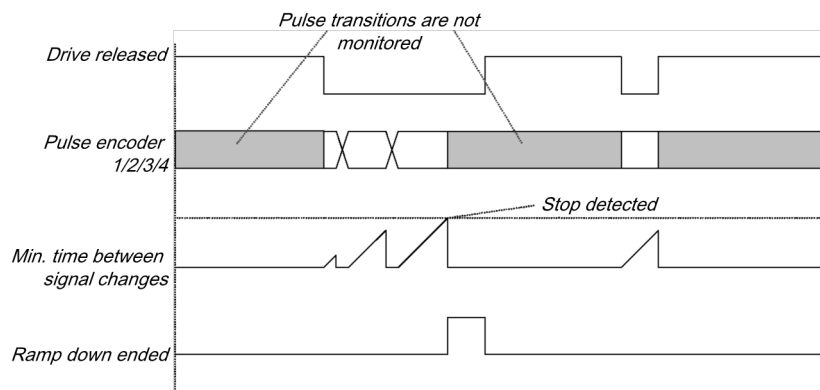


Illustration 115: Sequence/timing diagram for the "Ramp down detection" function block

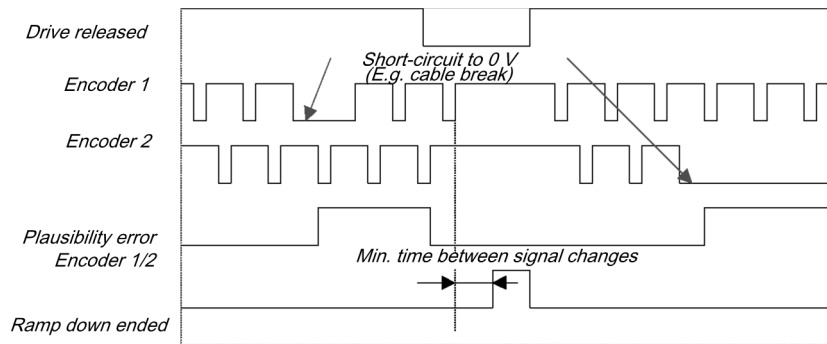


Illustration 116: Sequence/timing diagram for the ramp down detection with plausibility check function block

### 7.8 Function blocks for two-channel evaluation

The samosPRO system supports applications up to SIL3 (as per EN 62061) and Performance Level PL e (as per EN ISO 138491). Potential sources for function block inputs are one or two safety signals locally connected to the samosPRO safety control. You can choose between the following input evaluations (depending on the function block):

- Single-channel
- Two-channel
  - Two-channel equivalent (1 pair)
  - Two-channel discrepant (1 pair)
  - Two-channel equivalent (2 pairs)
  - Two-channel discrepant (2 pairs)

The following truth tables contain the internal evaluation for the individual types of input signal evaluations of the samosPRO safety control.

#### Truth table

The following applies to the truth tables in this section:

0 means logical low

1 means logical high

x means "any" = 0 or 1

**NOTICE** The error flag is high when the logic processing of the samosPRO safety control detects an error in the combination or in the sequence of input signals.



## 7.8.1 Emergency stop

### Function block diagram



Illustration 117: Function block diagram for the emergency stop function block

### General description

The emergency stop function block enables implementation of an emergency stop function with an emergency stop button.

If the corresponding two-channel input element is configured in the hardware configuration of the samosPLAN5+, then this function block is no longer required in the logic, because the preliminary evaluation then takes place directly at the module (e.g. SP-COP, SP-SDI, or SP-SDIO). However, if the **Error flag** output is required for further processing, this function block can be used. To this end, the two input signals are to be configured as single-channel signals and routed to the inputs of the function blocks.

With emergency stop buttons, a reset and/or restart function block must take over the processing of the reset/restart conditions for the security string when the **Release** output is low. This may also be necessary for emergency stop buttons with a combined push/pull release.

### Parameters of the function block

Table 77: Parameters of the emergency stop function block

Parameters	Possible values
Inputs	<ul style="list-style-type: none"> <li>• Single-channel</li> <li>• Two-channel equivalent</li> <li>• Two-channel discrepant</li> </ul>
Synchronous time	0 = inactive, 10 to 30000 ms in 10 ms increments. When active, the value must be greater than the logic execution time.
Number of outputs	• 3 (release output, synchronous time error, and sequence error)

### Sequence/timing diagrams

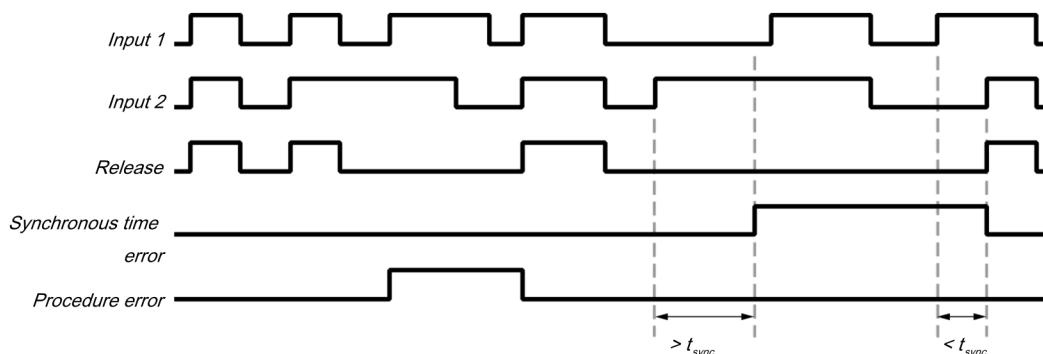


Illustration 118: Sequence/timing diagram for the emergency stop function block

### Further information

You can find additional information on the behavior of this function block here: *Two-channel evaluation and synchronous time [ch. 10, p. 232]*

## 7.8.2 Solenoid switch

### Function block diagram

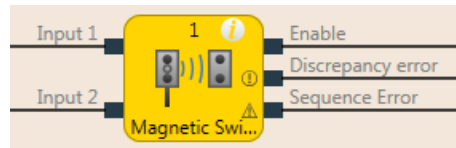


Illustration 119: Logical connections for the solenoid switch function block

### General description

The internal logic of the solenoid switch function block has the same function as that of the emergency stop function block, only with limited parameter selection. The function block enables graphic differentiation according to use.

The solenoid switch function block is a predefined function block for reed switches or other sensors for which synchronous time monitoring is required. When the evaluation of the discrepant inputs is high, the **Release** output is high.

Further information: *Two-channel evaluation and synchronous time [ch. 10, p. 232]*

### Parameters of the function block

Table 78: Parameters of the solenoid switch function block

Parameters	Possible values
Inputs	<ul style="list-style-type: none"> <li>Two-channel equivalent</li> <li>Two-channel discrepant</li> </ul>
Synchronous time	10 to 3000 seconds in 10 ms increments. The value must be greater than the logic execution time.
Number of outputs	<ul style="list-style-type: none"> <li>3 (release output, synchronous time error, and sequence error)</li> </ul>

## 7.8.3 Light curtain evaluation

### Function block diagram

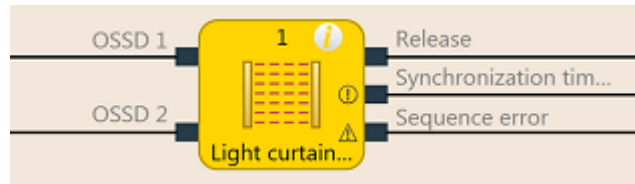


Illustration 120: Function block diagram for the light grid evaluation function block

### General description

The light grid evaluation function block enables the implementation of a semi-conductor protective device functionality with BWS.

The internal logic of the light grid evaluation function block corresponds to the function of the emergency stop function block, but with limited parameter selection. The single-channel input type is not selectable in the light grid evaluation function block. When the evaluation of the discrepant inputs is high, the **Release** output is high.

Further information: *Two-channel evaluation and synchronous time [ch. 10, p. 232]*

### NOTICE

If the corresponding two-channel input element is configured in the hardware configuration of the samosPLAN5+s, then this function block is no longer required in the logic, because the preliminary evaluation then takes place directly at the module (e.g. SP-COP, SP-SDI, or SP-SDIO). However, if the **Error flag** output is required for further processing, this function block can be used. To this end, the two input signals are to be configured as single-channel signals and routed to the inputs of the function blocks.

### Parameters of the function block

Table 79: Parameters of the light grid evaluation function block

Parameters	Possible values
Input type	Two-channel equivalent
Synchronous time	0 = inactive, 10 to 500 ms in 10 ms increments. When active, the value must be greater than the logic execution time.
Number of outputs	3 (release output, synchronous time error, and sequence error)

## 7.8.4 Switch evaluation

### Function block diagram

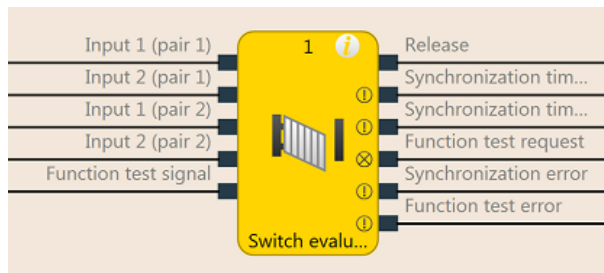


Illustration 121: Function block diagram for the switch evaluation function block

### General description

The function block can be used to evaluate two-channel switches. 1 pair or 2 pairs can be selected.

See the following for the behavior of the two-channel evaluation: *Two-channel evaluation and synchronous time* [ch. 10, p. 232]

Furthermore, the function block enables function test monitoring as an option.

### Parameters of the function block

Table 80: Parameters of the switch evaluation function block

Parameters	Possible values
Inputs / Mode	<ul style="list-style-type: none"> <li>• Single-channel</li> <li>• Two-channel equivalent (1 pair)</li> <li>• Two-channel discrepant (1 pair)</li> <li>• Two-channel equivalent (2 pairs)</li> <li>• Two-channel discrepant (2 pairs)</li> </ul>
With function test	<ul style="list-style-type: none"> <li>• yes: With function test</li> <li>• no: without function test</li> </ul>
Synchronous time pair 1	For inputs 1 and 2 of pair 1 and inputs 1 and 2 of pair 2 separately adjustable.
Synchronous time pair 2	Values: 0 = inactive, 10 to 30000 ms in 10 ms increments. When active, the value must be greater than the logic execution time.
Synchronization time	0 = inactive, 10 to 30000 ms in 10 ms increments. When active, the value must be greater than the logic execution time.
Number of outputs	1 to 6

### Function test

In some applications, safety devices require a cyclic physical check in order to ensure that the safety device is still functioning correctly.

When the switch evaluation function block is configured with the **Function test** parameter so that the **Require function test** input must be present, the input signal of the safety inputs must change once per machine cycle such that the enable condition is no longer met and back again (e.g. in the sequence of opening and closing a safety door).

The **Require function test** input is typically connected to the machine cycle contact.

If a function test is required according to the configuration, then it must be implemented in the following cases:

- after the samosPRO system has switched from the stop state into the run state and
- after each rising edge (low to high) at the **Require function test** input.

This is indicated by the **Function test required** output going to high. The **Function test required** output goes back to low when, before the next rising edge at the **Require function test** input, a signal sequence has been detected at the inputs, causing the **Release** output to switch from low to high.

The **Function test error** output goes to high and the **Release** output goes to low when the next machine cycle starts, before a function test has been conducted, i.e. when the **Function test required** output is still high and an additional rising edge (low to high) occurs at the **Require function test** input.

The **Function test error** output returns to low when a signal sequence has been detected causing the **Release** output to switch from low to high.

## Sequence/timing diagrams

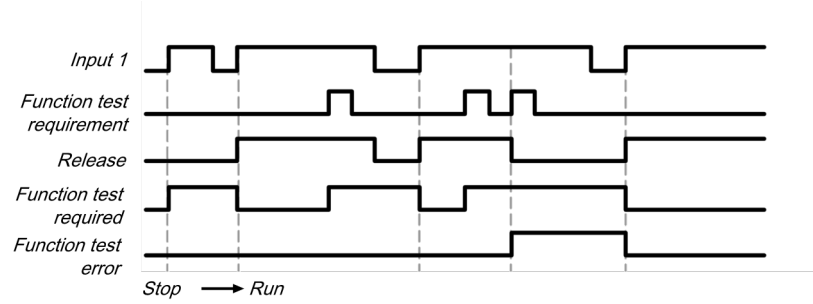


Illustration 122: Sequence/timing diagram for switch evaluation function block, category 2, single-channel with function test

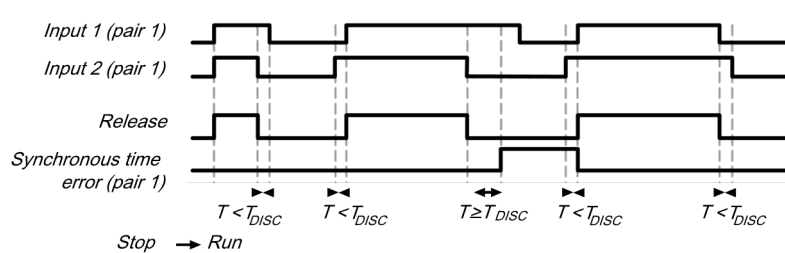


Illustration 123: Sequence/timing diagram for switch evaluation function block, category 4, two-channel without function test

## 7.8.5 Two-hand control, type IIIA

### Function block diagram



Illustration 124: Function block diagram for two-hand type IIIA function block

### General description

The two-hand type IIIA function block is a predefined function block for two-hand controls for which synchronous time monitoring of equivalent inputs is required. The internal logic of the two-hand type IIIA function block corresponds to the function of the emergency stop function block, but with limited parameter selection. The function block enables graphic differentiation according to the application.

**Input 1** and **Input 2** form a two-channel evaluation and must be equivalent. When the evaluation of the inputs is high, the **Release** output is high.

Further information: *Two-channel evaluation and synchronous time [ch. 10, p. 232]*

The synchronous time is 500 ms (the synchronous time is permanently set and cannot be changed).

### Parameters of the function block

Table 81: Parameters of two-hand type IIIA function block

Parameters	Possible values
Inputs	Permanently defined value: Two-channel equivalent
Start interlock	<ul style="list-style-type: none"> <li>without start interlock</li> <li>with start interlock</li> </ul>
Number of outputs	<ul style="list-style-type: none"> <li>3 (release output, synchronous time error, and sequence error)</li> </ul>

## 7.8.6 Two-hand control, type IIIC

### Function block diagram



Illustration 125: Function block diagram for two-hand type IIIC function block

### General description

The two-hand type IIIC function block provides the logic for monitoring the inputs of a two-hand control according to EN 574.



**Only use the two-hand type IIIC function block together with an SP-COP, SP-SDIO, or an SP-SDI module!**

The two-hand type IIIC function block requires the use of an SP-COP, SP-SDIO, or an SP-SDI module. Otherwise, the requirements of EN 574 will not be fulfilled.

The inputs used as single-channel input signals must be configured in the hardware configuration, i.e. no two-channel input evaluation at the expansion module.

### Parameters of the function block

Table 82: Parameters of two-hand type IIIC function block

Parameters	Possible values
Synchronous time (pair 1) ( $T_{SYN1}$ )	0 = inactive, 10 to 500 ms in 10 ms increments. When active, the value must be greater than the logic execution time.
Synchronous time (pair 2) ( $T_{SYN2}$ )	0 = inactive, 10 to 500 ms in 10 ms increments. When active, the value must be greater than the logic execution time.
Synchronization time $T_{SZ}$	Permanently defined value: 500 ms
Number of outputs	3 (release output, synchronous time error pair 1 output, and synchronous time error pair 2 output)

The function block evaluates its input signals. **Input 1 and Input 2 of pair 1** form a two-channel evaluation and must be discrepant. **Input 1 and Input 2 of pair 2** form a two-channel evaluation and must also be discrepant. A synchronous time can be specified for each of the two 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 two-hand circuit evaluation may not exceed 500 ms (the synchronization time is permanently defined and cannot be changed).

See the following for the behavior of the double two-channel evaluation: *Two-channel evaluation and synchronous time [ch. 10, p. 232]*

The synchronization evaluation with the two-hand type IIIC function block differs from the switch evaluation function block with respect to the condition for synchronization state inactive. With the two-hand type IIIC function block, both two-channel evaluations must be inactive, i.e. the 1/2 inputs of the two input pairs must be low/high simultaneously.

Furthermore, with the two-hand type IIIC function blocks, there is no **Synchronization error** output, because with a two-hand control, it is not evaluated as an error when the two manual switches are not activated simultaneously within the specified 500 ms. That said, this synchronization time may not be exceeded, because if it is, the **Release** output will not go to high.

## Sequence/timing diagram

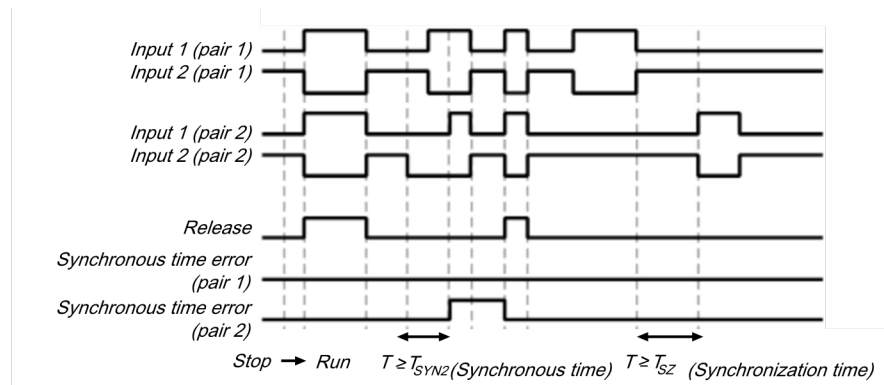


Illustration 126: Sequence/timing diagram for the two-hand type IIC function block



## 7.8.7 Multi-operator start

### Function block diagram

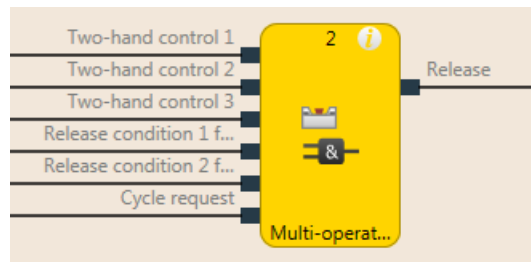


Illustration 127: Logical connections for the multi-two-hand function block

### General description

The multi-two-hand function block makes it possible to monitor the simultaneous operation of up to three two-hand controls. For example, for a press application with more than one operator, multiple two-hand controls or foot switches may be required in order to jointly trigger the downward movement of the press. Typically, each **Two-hand** input is connected to a two-hand function block.

Alternatively, **Release** inputs (e.g. safety light curtains) can be connected to ensure that the assigned devices are at high before the **Release** output can go to high. Reset and restart must be treated independently of this function block.

The **Cycle request** input can be used to force each connected two-hand control to have to be released at least once before a restart is possible. Typically, this input is connected to a signal that generates a pulse with each machine cycle. This can prevent one or more of the two-hand controls to remain active permanently.



**ATTENTION**

#### The two-hand and the release inputs must be pre-evaluated signals!

- Only connect safely pre-evaluated signals to the **Two-hand** inputs, e.g. the **Release** output of a two-hand type IIIA or two-hand type IIIC function block. A safety-relevant evaluation of the inputs of a two-hand control must occur either through another function block (e.g. two-hand control or light grid evaluation) or as a component of the configuration of the safety inputs (e.g. configuration of the inputs with two-channel evaluation).
- The **Cycle request** input must not be used for safety functions. This input is only used for automation control.

## Parameters of the function block

Table 83: Parameters for the multi-two-hand function block

Parameters	Possible values
Cycle request	<ul style="list-style-type: none"> <li>• Rising edge</li> <li>• Falling edge</li> </ul>
Number of operators	<ul style="list-style-type: none"> <li>• 2 operators</li> <li>• 3 operators</li> </ul>
Number of enable conditions	<ul style="list-style-type: none"> <li>• 0</li> <li>• 1</li> <li>• 2</li> </ul>

The **Release** output is high when ...

- all **Release** inputs are high and remain high; and
- every activated **Two-hand** input was at low at least once (including chronologically offset), after the samosPRO system has switched from the stopped state into the run state or after a rising or falling edge (depending on the configuration) has been detected at the **Cycle request** input; and
- all activated **Two-hand** inputs are then at high.

The **Release** output is low when ...

- one or more **Release** inputs are low; or
- one or more **Two-hand** inputs are low; or
- a rising or falling edge (depending on the configuration) has been detected at the **Cycle request** input.

## Sequence/timing diagram

All two-hand inputs completed one cycle after the falling edge at the input cycle request (or after the start) and the activated release input is High.

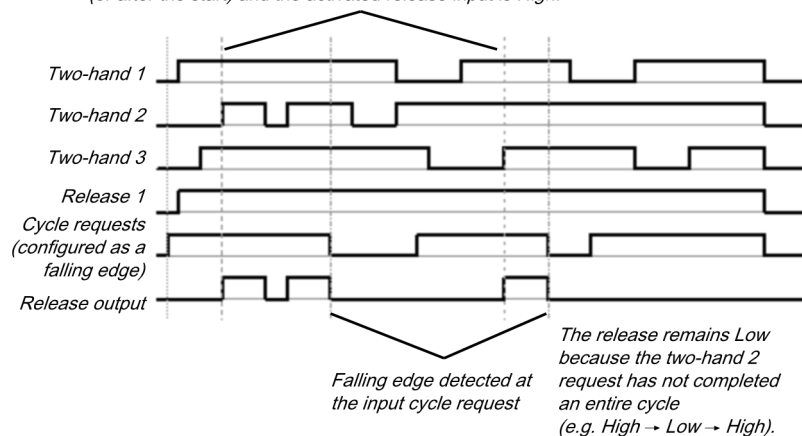


Illustration 128: Sequence/timing diagram for the multi-two-hand function block

## 7.9 Function blocks for parallel muting, sequential muting, and cross muting

### 7.9.1 Overview and general description

Muting is the automatic temporary suppression of safety-based monitoring of an access area with the assistance of a contactless-acting safety device (BWS), while certain objects, e.g. pallets of material, are being moved into the hazardous area.

Muting sensors monitor the presence of the material while it is being transported. By carefully selecting the type and arrangement of the sensors, it is possible to differentiate between objects and people.

Working together with the muting sensors and the contactless safety device, the object being conveyed generates a precisely defined signal sequence while it is moving in the hazardous area. The muting sensors must ensure that any potential hazard is excluded if a person enters the area being protected by the contactless safety device (i.e. a hazard-inducing state must be ended immediately). It must not be possible for a person to generate the same signal sequence as the object being conveyed.

The placement of muting sensors is determined by the shape of the object to be detected. To this end, the following options, among other things, are available with differing numbers of sensor input signals:

- Two sensors
- Two sensors and one C1 auxiliary signal
- Four sensors (two sensor pairs)
- Four sensors (two sensor pairs) and one C1 auxiliary signal

Muting sensor signals can be generated by the following external sensors:

- Optical sensors
- Inductive sensors
- Mechanical switches
- Signals from the control

If you use optical sensors for muting applications, use sensors with background suppression in order to ensure that only the material being conveyed meets the muting conditions. The sensors detect material only up to a certain distance. Objects that are further away therefore cannot fulfill the input conditions of the muting sensors.

There are three different function blocks available for muting:

- Parallel muting (muting with two parallel sensor pairs)
- Sequential muting (muting with two sequential sensor pairs)
- Cross muting (muting with a crossed sensor pair)

#### Info

- The muting cycle is the defined sequence of all processes that occur during muting.
- The muting cycle starts 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 reactivate muting until the previous muting cycle has ended.
- Material can be transported multiple times within a muting cycle if the muting condition is continuously maintained during this time, i.e. at least one sensor pair remains activated continuously.

### Safety information

Because the safety functions of a safety device are bypassed by the muting, multiple requirements must be fulfilled, as shown below, in order to ensure the safety of the application.



#### The general safety regulations and safety measures must be followed!

If you use muting, make absolutely sure that the following information is followed for correct use of muting.

- Access to the hazardous area must be reliably detected by the contactless safety device or be prevented by other measures. It must not be possible for a person to bypass, climb over, crawl under, or cross the contactless safety device undetected. Note the operating instructions for the contactless safety device for correct installation and use of the device.
- Always follow the valid local, regional, and national regulations and standards that apply to your application. Make sure that your application corresponds to an appropriate risk analysis and avoidance strategy.
- Muting must never be used to convey a person into the hazardous area.
- Install the command devices for reset and override outside of the hazardous area, so that they can never be operated by someone who is inside of the hazardous area. In addition, the operator must have a complete overview of the hazardous area when operating a command device.
- The muting sensors must be arranged such that, after access into the protective field, the hazardous area can only be reached once the hazard-inducing state has been ended. A condition for this is that the safety distances required and defined in EN ISO 13855 are maintained. At least two muting signals, which are independent from one another, are required.
- Muting may only be activated for the time span in which the object that is triggering the muting condition is blocking access to the hazardous area.
- The area between the contactless safety device and the muting sensors must be secured against someone accessing the area from the rear:
  - With parallel muting between the contactless safety device and A1/A2 sensors and between the contactless safety device and B1/B2 sensors (see *Figure [ch. 7.9.6, p. 151]*).
  - With sequential muting between the contactless safety device and A2 sensor and between the contactless safety device and B1 sensor (see *Figure [ch. 7.9.7, p. 153]*).
  - With cross muting between the contactless safety device and A1 sensor and between the contactless safety device and A2 sensor (see *Figure [ch. 7.9.8, p. 155]*).
- Muting must take place automatically but not depend on a single electric signal.
- The material to be transported must be detected over the entire length, i.e. there must not be any interruption in the output signals.
- Muting must be triggered by at least two independently wired signals (e.g. by muting sensors) and must not depend completely on software signals (e.g. on PLC).
- The muting condition must be ended immediately after the object has traveled through so that the safety device returns to its normal state, which is the state not bypassed by muting (i.e. that it is again effective).
- The muting sensors must be arranged such that muting cannot be unintentionally triggered by a person.

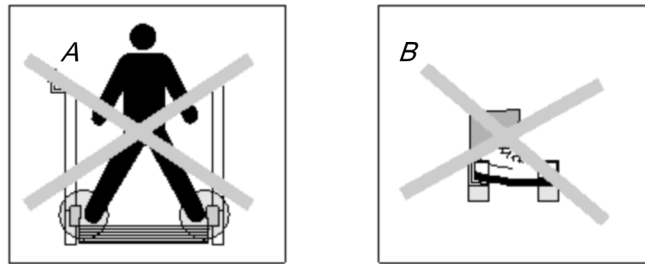


Illustration 129: Safety when installing the muting sensors

- Always arrange the muting sensors such that only the material is detected and not the transport means (pallet or vehicle).

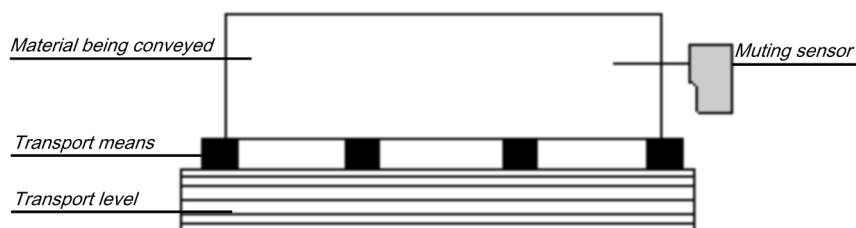


Illustration 130: Detection of material during muting

- Always arrange the muting sensors such that the material can travel through without hindrance but that people are always reliably detected.
- Always arrange the muting sensors such that when the material is detected a minimum distance is always maintained with respect to the detection area of the contactless safety device (e.g. with respect to the light beam of a light curtain).
- Before and during activation of override, ensure that there is no one within the hazardous area.
- Before you activate override, make sure that the device is in proper working condition, particularly the muting sensors (visual inspection).
- If it was necessary to activate override, afterward check the function of the device and the arrangement of the muting sensors.
- During longer muting cycles (i.e. longer than 24 hours) or during longer shutdowns of the machine, the correct function of the muting sensors must be checked.
- In order to indicate that muting or override is active, a muting and/or override lamp must be used. An external muting/override lamp or one integrated into the contactless safety device can be used.
- Depending on the local, regional, and national regulations and standards, it may be necessary to monitor the muting/override lamp(s). If this is the case, additional measures will be necessary for this. The SP-SDIO and SP-SDI modules do not support the monitoring of lamps.
- Always attach the muting or override lamp so it is highly visible. The muting or override lamp must be clearly visible from all sides around the hazardous area and to the operator of the system.
- If safety-relevant information (i.e. decentralized safety input values and/or decentralized safety output values) are transmitted via a safety fieldbus network, you must always consider the associated delay times. These delay times can influence both the system behavior and the requirements for the minimum safety distances associated with the response times.
- Once an override input has been configured, no test pulse outputs can be used when configuring the safety inputs.
- For sensor signals A1 and A2 (B1 and B2), separate lines must be used.

## Logic programming – Function blocks

- For the signals for reset and reset required, a line that is independent from the other input signals must be used in order to prevent any unintentional resetting of the system. The line must also be routed in a protected manner.
- The total muting time cannot be set to endless (inactive) without additional precautions being made. If the total muting time is deactivated, additional measures must be undertaken in order to ensure that no one can reach the hazardous area while muting is active.

### 7.9.2 Parameters of the function blocks

The following table shows the potential configuration parameters of the function blocks for muting.

Table 84: Mode of the function blocks for muting

Mode	Possible values
Directional detection	<ul style="list-style-type: none"> <li>• Inactive</li> </ul> Only with parallel muting and sequential muting: <ul style="list-style-type: none"> <li>• Forwards (A1/A2 first)</li> <li>• Backwards (B1/B2 first)</li> </ul>
Condition for muting start	<ul style="list-style-type: none"> <li>• All sensors are free</li> </ul> Only with parallel muting and sequential muting: <ul style="list-style-type: none"> <li>• At least one sensor is free</li> </ul>
Condition for muting end	<ul style="list-style-type: none"> <li>• With muting sensor pair</li> <li>• With contactless safety device</li> </ul>
C1 input	<ul style="list-style-type: none"> <li>• With</li> <li>• Without</li> </ul>
Belt signal	<ul style="list-style-type: none"> <li>• With</li> <li>• Without</li> </ul>
Override input	<ul style="list-style-type: none"> <li>• With</li> <li>• Without</li> </ul>
Sequence monitoring	Not selectable This is specified by selecting the muting function block. <ul style="list-style-type: none"> <li>• Active: With sequential muting</li> <li>• Inactive: With parallel muting and cross muting</li> </ul>

Table 85: Parameters of the function blocks for muting

Parameters	Possible values
Total muting time	0 = inactive, 5 s to 3600 s, adjustable in 1 s increments
Simultaneity monitoring time	0 = inactive, 10 to 3000 ms, adjustable in 10 ms increments. When active, the value must be greater than the logic execution time.
Dropout time (suppression of sensor signal gaps)	0 = inactive, 10 to 1000 ms, adjustable in 10 ms increments. When active, the value must be greater than the logic execution time.
Additional muting time after the contactless safety device is free	0 ms, 200 ms, 500 ms, 1000 ms
Min. override pulse time	<ul style="list-style-type: none"> <li>• 100 ms</li> <li>• 350 ms</li> </ul>

### 7.9.2.1 Directional detection

Directional detection is used when material being conveyed has to be moved in a certain direction. The direction depends on the sequence in which the muting sensors are activated.

When directional detection is inactive, the material being conveyed can be moved in both directions in order to fulfill the muting conditions. In this case, it does not matter which sensor pair is activated first.

If **Forwards (A1/A2 first)** was selected as the direction, the muting sensor pairs must be activated in the sequence (A1/A2) before (B1/B2). In the opposite direction, muting is not possible. A transition from four active sensors to an inactive "B" sensor pair (0 or 1 sensor active) ends muting.

If **Backwards (B1/B2 first)** was selected as the direction, the muting sensor pairs must be activated in the sequence (B1/B2) before (A1/A2). Muting is not possible in the forward direction. A transition from four active sensors to an inactive "A" sensor pair (0 or 1 sensor active) ends muting.

### 7.9.2.2 Condition for muting start

The **Condition for muting start** parameter determines when a valid muting sequence can begin. The **Condition for muting start** can be defined as follows:

- Both sensors are free: All muting sensors have jointly or individually gone to low and the OSSDs of the safety device (e.g. safety light curtain) are high (i.e. the protective field is free);
- or
- At least one sensor is free: All muting sensors except for the last muting sensor are low and the OSSDs of the safety device (e.g. safety light curtain) are high (i.e. the protective field is free).

If a higher throughput rate is required, it can be advantageous to enable the start of the next muting sequence as soon as the material being conveyed has passed the protective device and all of the muting sensors with the exception of the last one (i.e. **At least one sensor is free**).

### 7.9.2.3 Condition for muting end

Contrary to the **Condition of the other sensor pair for muting start** parameter, the **Condition for muting end** parameter determines when a valid muting state is over. You can select when the **Condition for muting end occurs**.

- **With muting sensor pair:** When a muting sensor of the last muting sensor pair goes to low (sensor free)
- or
- **With contactless safety device:** When the OSSDs of the safety device (e.g. safety light curtain) indicate that the protective field is no longer being violated, i.e. that the protective field is free and the OSSDs are back at high.

If the OSSD input of the contactless safety device is low after muting end (e.g. due to a violation of the safety device's protective field), before the next valid muting sequence has begun, the **Release** output of the function block goes to low. The next muting cycle cannot begin until the **Condition for muting end** has been fulfilled.

### 7.9.2.4 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, then the **Muting error** outputs and the **Error flag** go to high and the **Release** output goes to low.

The timer for the **Total muting time** starts upon activation of the muting function, and this is indicated by the transition of the **Muting status** output to high. The timer for the **Total muting time** is maintained and reset to zero when the muting function is again deactivated. If an optional **Belt signal** input is used, the timer pauses for the total muting time when the **Belt signal** input is high and thus indicates that the conveyor belt has stopped.

### 7.9.2.5 Additional muting time after the contactless safety device is free

The **Additional muting time after contactless safety device is free** parameter is used when the **Condition for muting end** parameter has been configured as **with contactless safety device**. When the contactless safety device is not always precisely detecting the muting end due to irregularities in the material or the transport means, then you can increase the machine availability by configuring an additional muting time of up to 1000 ms. The **Additional muting time after contactless safety device is free** parameter will only determine the additional muting time in this case after the OSSDs of the contactless safety device have gone back to high, i.e. the safety light curtain is no longer interrupted.

### 7.9.2.6 Simultaneity monitoring time

The simultaneity monitoring time is used to check whether the muting sensors are being activated simultaneously. This value indicates the maximum duration for which each of the two muting sensor inputs being evaluated as two channels can 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 adopt the equivalent values before the simultaneity monitoring time has elapsed.

The simultaneity monitoring time starts with the first change of an input value for a muting sensor. If the simultaneity monitoring time has elapsed and the two inputs for an input pair still have different values, an error occurs.

If the simultaneity monitoring time determines an error with at least one input pair, the function block indicates this error by setting the muting error output at high.

### 7.9.2.7 Suppressing sensor signal gaps

Occasionally, malfunctions occur in the output signals of muting sensors that have no significance for muting. The **Suppression of sensor signal gaps** function makes it possible to filter out brief malfunctions without muting being interrupted.

If **Suppression of sensor signal gaps** is active, a low signal from a muting sensor input is ignored for the duration of the set value for the **Suppression of sensor signal gaps**. The function block continues to interpret this signal as an uninterrupted high as long as only one sensor per A1/A2 or B1/B2 pair has a signal gap. If a signal gap has been detected at a sensor, the simultaneous occurrence of an additional signal gap at another sensor will lead to the termination of muting.

### 7.9.2.8 Sequence monitoring

**Sequence monitoring** makes it possible to define a special forcibly defined sequence in which the muting sensors must be active. The following table shows the valid sequence of muting sensor input signals. This parameter is only available for configurations with four muting sensors, e.g. for parallel or sequential muting.

Table 86: Requirements for sequence monitoring

Directional detection	Requirements for muting sensor signal inputs for sequence monitoring
Inactive	A1 before A2 before B1 before B2 or B2 before B1 before A2 before A1
Forwards	A1 before A2 before B1 before B2
Backwards	B2 before B1 before A2 before A1

This parameter depends on the function block. Deviations from the previously shown sequence will cause a muting error that is displayed at the **Muting error** output. In order to prevent machine stoppages, the configured time for the **Suppression of sensor signal gaps** should additionally be less than the timeframe that the object being conveyed requires in order to pass a muting sensor pair (e.g. A1/A2 or B1/B2).



## 7.9.2.9 C1 input

The **C1** input is used as additional safeguarding against manipulations. If the **C1** input is used, a transition from low to high must occur before the first muting sensor pair goes to high. The **C1** input must then remain at high until both sensors of the muting sensor pair are at high so that a valid muting condition can result. If this condition is not fulfilled, this will lead to a muting error, which is indicated at the **Muting error** output. The **C1** input must then go back to low before the next muting cycle is enabled.

## 7.9.2.10 Override input

An **Override** input signal makes it possible to remove objects being conveyed that remain lying in the protective field of the safety device (e.g. safety light curtain) after power failures, triggering of an emergency stop, muting errors, or other similar circumstances.

The **Override required** output pulses at 2 Hz when the following conditions are fulfilled:

- Muting is currently low (i.e. **Muting status** is low).
- At least one muting sensor is high.
- The OSSDs of the contactless safety device are low (e.g. safety light curtain has been interrupted).
- The **Release** output is low.

Once the conditions for the **Override required** output have been fulfilled and a valid override sequence with a low-high-low transition (at least 100 ms or 350 ms and a maximum 3 seconds; longer or shorter pulses are ignored) has occurred at the **Override** input, the **Release** output goes to high as if the muting conditions had been fulfilled. Once all of the muting sensors have gone back to low and the OSSD input of the contactless safety device is at high (e.g. indicates that the protective field of a safety light curtain is now free), the next valid muting cycle will be expected. If the next object does not fulfill the conditions for a muting cycle, but does meet the conditions for the **Override required** voutput, an additional override cycle can be used to remove the material being conveyed. The number of override cycles is limited (see table titled *Number of permissible override cycles* below).

**NOTICE** A reset button can also be suitable for the override function. Check the requirements of your application to ensure that the safety-relevant logic meets the requirements of the local, regional, national, and international regulations.

You can find information on the **Override required** output and when override is possible under the conditions shown and when not in the following table:

Table 87: Conditions for override required and override possible

Muting status	At least one muting sensor is high	Contactless safety device OSSDs are high	Override required output	Override possible
0	No	0	No	No
0	No	1	No	No
0	Yes	0	Pulses (2 Hz)	Yes when the maximum permissible number of override cycles has not been 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

The following figure shows an example sequence for **Override** and **Override required**.

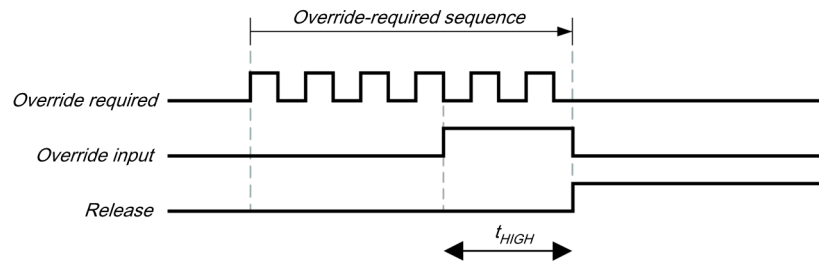


Illustration 131: Logic diagram for override and override requirement

**NOTICE**

$t_{HIGH}$  must be equal to or greater than the minimum override pulse time (100 ms or 350 ms) but less than or equal to 3 s. When  $t_{HIGH}$  is less than the minimum override pulse time or greater than 3 s, the **Override** input is ignored.



**ATTENTION**

**When you use override, check whether the system is in a safe state!**

The override function makes it possible for you to activate the **Release** output of the muting function block even though the safety device (e.g. safety light curtain) is indicating that a hazard-inducing state could be present. The **Override** input should only be used when the hazardous area has been visually checked and there is no one in the hazardous area and no one has had access to the hazardous area while the **Override** input is being used.



**ATTENTION**

**Make sure that the transitions of the signals for restart meet the requirements of the safety standards and regulations!**

When there is 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 as a result of short-circuit detection. If this type of pulse can lead to a hazard-inducing state in the machine, then the following points should be noted:

- Ensure that the cable routing is protected for the signal lines (due to cross-connection with other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

During an override cycle, the **Release** output is set at high, just as during a valid muting sequence. To prevent excessive use of the override function, the number of permissible override cycles is limited. The number of permissible override cycles depends on the value for the total muting time. The following table has a compilation of the number of permissible override cycles:

Table 88: 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	Minimum number of override cycles = 5
15 min	5	
30 min	5	
60 min	5	
Inactive (unlimited)	5	

The number of override cycles is stored in the function block. This value is increased each time the **Override required** output starts to pulse. The value is reset to 0 when a valid muting cycle has taken place after a system reset (e.g. with the assistance of the samosPLAN5+) or after a transition from the stop state to the run state.

After the **Override required** output has started to pulse at 2 Hz and a subsequent **Override** signal has been set at high, muting starts again and the **Release** output is set at high.

When the muting cycle is stopped due to a faulty input signal of a muting sensor, **Override required** is at high for the duration of the logic execution time when the remaining conditions are fulfilled for **Override required**. When the faulty input of the muting sensor goes back to high and subsequently back to low, the muting cycle is again maintained and **Override required** goes to high when the remaining conditions for **Override required** are met.

During a valid override state, directional detection, sequence monitoring (depending on the function block), and simultaneity monitoring time are not executed for the duration of an override cycle.

### 7.9.2.11 Belt signal

If the movement of the transported material is stopped during the muting cycle, it is possible that the total muting time and other parameters that could lead to a muting error are being exceeded. This can be avoided with the assistance of the **Belt signal** inputs. This input makes it possible to stop time-dependent functions associated with muting when the material to be conveyed does not continue to move.

The **Belt signal** input must correspond to EN 61131 and have 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 **Belt signal** input:

Table 89: Effects of conveyor belt monitoring on timer functions

<b>Monitoring of total muting time</b>	<ul style="list-style-type: none"> <li>• When a belt stop is detected, the timer functions will pause.</li> <li>• When the conveyor belt starts back up, the timer will continue with the value stored before detection of the belt stop. When this occurs for the first time, the total muting time is increased once by 5 seconds.</li> </ul>
<b>Simultaneity monitoring time</b>	

**NOTICE** The **Suppression of sensor signal gaps** is not influenced by a belt stop.

### 7.9.2.12 Min. override pulse time

The **Min. override pulse time** determines how long the **Override** input must be at least high so that the override signal is valid.

### 7.9.2.13 Muting status output

The **Muting status** output indicates the status of the muting function according to the following table:

Table 90: Output values for muting status

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

### 7.9.2.14 Muting lamp output

The **Muting lamp** output is used to indicate an active muting cycle. The value for the **Muting lamp** output depends directly on the value for the **Muting status**, as shown in the following table:

Table 91: Output values for the muting lamp output

Status of the function block for muting	Value of the muting lamp output
Muting status output is low	Low
Muting status output is high	High
Override cycle active	High
Override requirement	Pulses at 2 Hz

### 7.9.2.15 Muting error output

The **Muting error** output is used to indicate that an error associated with the muting function block has been detected. The value of the **Muting error** output is high when any muting error is detected. In order to reset a muting error, it is necessary for all of the muting sensors to go back to low and for the OSSD signal of the contactless safety device to be at high.

### 7.9.2.16 Release output

When a valid muting condition is present, a valid override cycle takes place or if the OSSD input of the contactless safety device is free and no error/error state is active, then the **Release** output is high.

### 7.9.3 Information regarding cabling

When muting functions are supposed to be implemented, potential errors in the cabling must be considered. If certain signal combinations are to be transferred in a common cable, additional precautions must be implemented in order to ensure that the respective signals are correct. Suitable measures must be undertaken (e.g. protected cabling) in order to ensure that no errors can occur as a result of the cabling.

*Table 92: Cabling combinations for muting and requirements*

Signal description	A1	A2	B1	B2	C1	Belt signal	Contactless safety device	Override input	Release 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
Belt signal	A	A	A	A	A	–	C	A	A	C	C	C
Contactless safety device	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 indicated signals must not be installed in a common cable if a protected cable is not being used.

**B**—The indicated signals must not be installed in a common cable if a protected cable or sequence monitoring is not being used.

**C**—The indicated signals must not be installed in a common cable.

– – Not applicable

## 7.9.4 State transition from stop to run

When the samosPRO safety control transitions from the stop state to the run state, the following behaviors may occur depending on the state of the muting sensors and the OSSDs of the sensors (e.g. safety outputs of a safety light curtain). The following table shows details regarding the system behavior during the transition from stop to run.

Table 93: Stop-to-run transition behavior for muting functions

State after the transition from stop to run		System behavior	
Contactless safety device input	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 must return to low before the sensor OSSDs go to low. Once the OSSDs of the sensors are low, the override must be used before all of the muting sensors are at low.
	The muting condition is fulfilled.		
Low (e.g. object detected)	All muting sensors are low.	Muting is blocked.	The sensor OSSDs must be high before muting can take place.
	The muting condition is partially fulfilled.	Override is required if configured.	Either transition to normal behavior (upon cyclically correct sequence of the sensor states) or the total override time will be exceeded.
	The muting condition is fulfilled.		

## 7.9.5 Error states and information regarding reset

Table 94: Error states and information regarding reset for muting function blocks

Diagnostic outputs	Error state reset	Remarks
Muting error: <ul style="list-style-type: none"> <li>• Error in simultaneity monitoring time</li> <li>• Error in total muting time monitoring</li> <li>• Error in directional detection</li> <li>• Sequence error detected</li> <li>• Error in sensor gap monitoring</li> </ul>	Before any muting error can be reset, a complete valid muting cycle must take place. To this end, either override must be used or all of the muting sensors and the OSSDs of the contactless safety device must be free and subsequently a valid muting sequence must take place.  Once one of these two conditions is fulfilled, the muting error output goes back to low, provided that no other error is pending.	The release output goes to low and the error flag goes to high when the muting error output is high.

## 7.9.6 Parallel muting

### Function block diagram

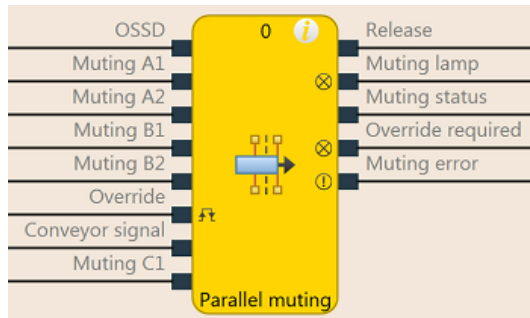


Illustration 132: Logical connections for the parallel muting function block

### Representation of use

The following figure shows an example of the placement of sensors for parallel muting:

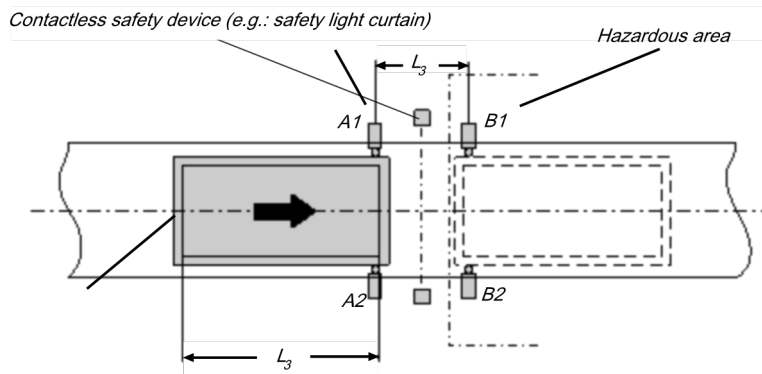


Illustration 133: Muting with two parallel sensor pairs

The material in this example is moving from left to right. As soon as the first muting sensor pair, A1 & A2, is activated, the protective effect of the safety device is bypassed. The protective effect remains bypassed until muting sensor pair B1 & B2 is once again free.

You will find a description of the parameters here: *Parameters of the function blocks [ch. 7.9.2, p. 142]*

### Input conditions for muting sensors

Table 95: Conditions for parallel muting

Condition	Description
A1 & A2 (or B1 & B2)	Starts the muting cycle. The first sensor pair is activated depending on the transport direction 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 transport direction of the material.

Formulas and requirements for calculating the distance:

- $L_1 \geq v \times 2 \times T_{IN \text{ Muting sensor}}$
- $v \times t > L_1 + L_3$
- $L_1 < L_3$
- $T_{IN \text{ Light curtain}} < T_{IN \text{ Muting sensor}}$

The following is valid ...

$L_1$	Distance between the sensors (symmetrical arrangement with respect to the detection area of the contactless safety device)
$L_3$	Length of the material in the conveying direction
$v$	Speed of the material (e.g. of the conveyor belt)
$t$	Set total muting time [s]
$T_{IN}$ Light curtain	Response time of the light curtain or the muting sensors in the samosPRO system (see "samosPRO hardware" manual (BA000966) in the section "Response times of the samosPRO system")
$T_{IN}$ Muting sensor	

### Info

- The material can be moved in both directions or a defined transport direction can be defined for it as follows:
  - Using the optional **C1** input If used, the **C1** input must always be activated before both muting sensors of the first sensor pair (e.g. A1 and A2) go to high.
  - Using the **Directional detection** configuration parameter
- With a parallel arrangement, the width of the permissible object is additionally checked through the position of the muting sensors. The objects must always pass the muting sensors with an identical width.
- For this application, optical scanners and all types of non-optical sensors can be used. Use sensors and scanners with background suppression.
- Avoid interaction between sensors.
- You can increase the protection against manipulation and the security with the assistance of the following configurable functions:
  - Simultaneity monitoring time
  - Monitoring of total muting time
  - Muting end through contactless safety device
- Further information: *Information regarding cabling [ch. 7.9.3, p. 149]*

### Sequence/timing diagram

The function block requires that a valid muting sequence take place. The following figure shows an example of a valid muting sequence based on the basic parameter setting for this function block:

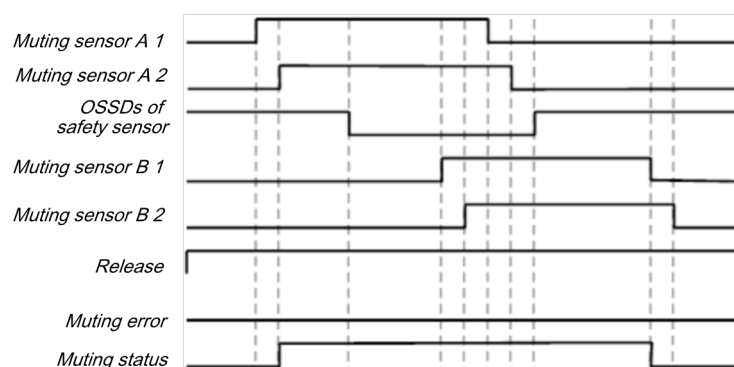


Illustration 134: Valid muting sequence when using the basic configuration setting



## 7.9.7 Sequential muting

### Function block diagram

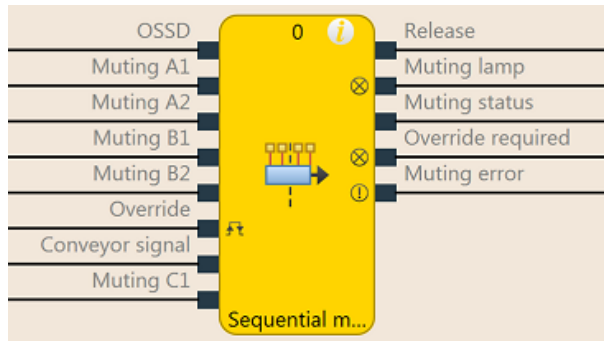


Illustration 135: Logical connections for the sequential muting function block

### Representation of use

The following figure shows an example of the arrangement of sensors together with the sequential muting function block.

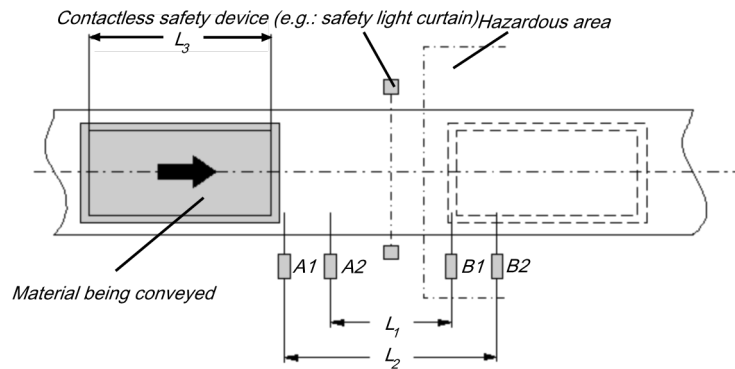


Illustration 136: Example of the sequential arrangement of muting sensors

The material in this example is moving from left to right. As soon as muting sensors A1 & A2 are activated, the protective effect of the safety device is bypassed. The protective effect remains bypassed until a sensor of muting sensor pair B1 & B2 is once again free.

You will find a description of the parameters here: *Parameters of the function blocks [ch. 7.9.2, p. 142]*

### Input conditions for muting sensors

Table 96: Conditions for muting with four sensors with sequential arrangement

Condition	Description
A1 & A2 (or B1 & B2)	Starts the muting cycle. The first sensor pair is activated depending on the transport direction 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 transport direction of the material.

Formulas and requirements for calculating the distance:

$$L_1 \geq v \times 2 \times T_{IN \text{ Muting sensor}}$$

$$v \times t > L_1 + L_3$$

$$L_2 < L_3$$

$$T_{IN \text{ Light curtain}} < T_{IN \text{ Muting sensor}}$$

The following is valid ...

$L_1$	Distance between the inner sensors (symmetrical arrangement with respect to the detection area of the contactless safety device)
$L_2$	Distance between the outer sensors (symmetrical arrangement with respect to the detection area of the contactless safety device)
$L_3$	Length of the material in the conveying direction
$v$	Speed 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 samosPRO system (see "Response times of the samosPRO system" in the "samosPRO hardware" manual (BA000966)).

### Info

- In this example, the material can either be moved in both directions or a specified transport direction can be defined as follows:
  - Using the optional **C1** input If used, the **C1** input must always be activated before both muting sensors of the first sensor pair (e.g. A1 and A2) go to high.
  - Using the **Directional detection** configuration parameter
- The arrangement of sensors shown in this example is suitable for all types of sensors.
- Avoid interaction between sensors.
- You can increase the protection against manipulation and the security with the assistance of the following configurable functions:
  - Simultaneity monitoring time
  - Monitoring of total muting time
  - Muting end through contactless safety device
  - Sequence monitoring
- You can find information on cabling here: *Information regarding cabling [ch. 7.9.3, p. 149]*

### Sequence/timing diagram

The function block requires that a valid muting sequence take place. The following figure shows an example of a valid muting sequence based on the basic parameter setting for this function block.

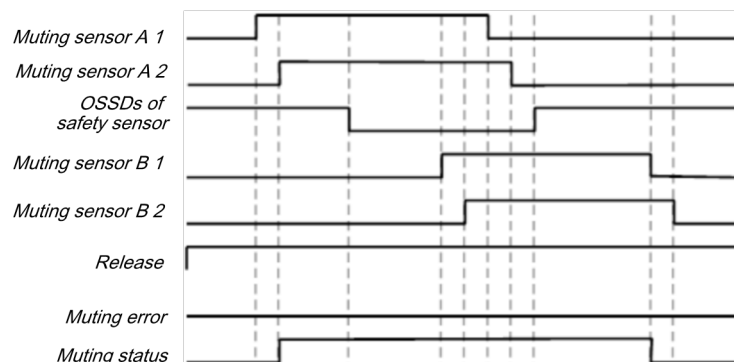


Illustration 137: Valid muting sequence when using the basic configuration setting

## 7.9.8 Cross muting (one-way) – Movement direction only forwards or only backwards

### Function block diagram

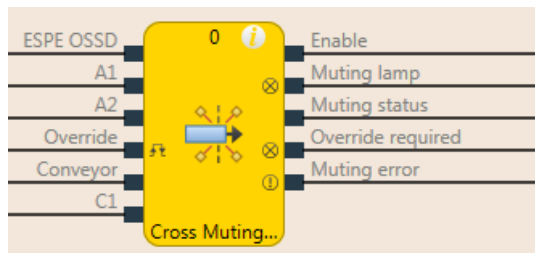


Illustration 138: Logical connections for the cross muting function block with the C1 input

### Representation of use

The figure below shows an example of the arrangement of the sensors for the cross muting function block. The **C1** input is used as additional manipulation protection for the muting system.

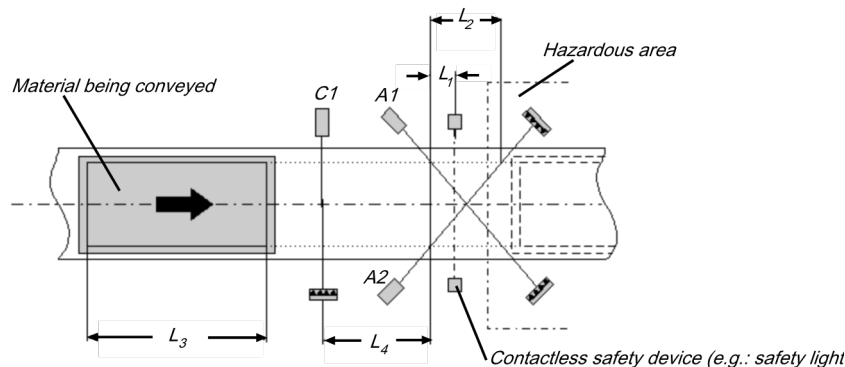


Illustration 139: Example of cross muting with C1 input

The protective effect of the safety device is bypassed when the muting sensors are activated in a defined sequence. The **C1** input must always be activated before **both** muting sensors of the first sensor pair (e.g. A1 and A2) go to high.

You will find a description of the parameters here: *Parameters of the function blocks [ch. 7.9.2, p. 142]*

### Input conditions for muting sensors

Table 97: Conditions for cross muting with C1 input

Condition	Description
C1 & A1 & A2	C1 must always be activated before both muting sensors of the sensor pair (e.g. A1 and A2) go to high.
A1 & A2	Muting applies as long as this condition is fulfilled and the previously shown requirement is also met.

Formulas and requirements for calculating the distance:

$$L_1 \geq v \times T_{IN \text{ Muting sensor}}$$

$$v \times t > L_2 + L_3$$

$$L_3 > L_4$$

$$T_{IN \text{ Light curtain}} < T_{IN \text{ Muting sensor}}$$

The following is valid ...

$L_1$	Minimum distance between the detection line of the contactless safety device and detection through A1, A2
$L_2$	Distance between the two detection lines of the sensors (sensors activated/sensors free)
$L_3$	Length of the material in the conveying direction
$L_4$	Maximum distance between C1 and the detection line of A1, A2
$v$	Speed of the material (e.g. of the conveyor belt)
$t$	Set total muting time [s]
$T_{IN}$ Light curtain'	Response time of the light curtain or the muting sensors <b>in the samosPRO system</b> (see "Response times of the samosPRO system" in the "samosPRO hardware" manual (BA000966)).
$T_{IN}$ Muting sensor	

### Info

- In this example, the flow of material is only possible in one direction.
- In order to move material in both directions (i.e. bidirectionally), place the crossing point directly in the light beam of the contactless safety device.

Further information: *Cross muting (two-way) – Material transport in both directions [ch. 7.9.9, p. 157]*

- The arrangement of the sensors shown in this example is suitable for both through-beam light barriers as well as for reflection light barriers.
- Avoid interaction between sensors.
- You can increase the protection against manipulation and the security with the assistance of the following configurable functions:
  - Simultaneity monitoring time
  - Monitoring of total muting time
  - Muting end through contactless safety device
- Further information: *Information regarding cabling [ch. 7.9.3, p. 149]*

### Sequence/timing diagram

The function block requires that a valid muting sequence take place. The following figure shows an example of a valid muting sequence based on the basic parameter setting for this function block. The **C1** input is not included in the sequence shown below.

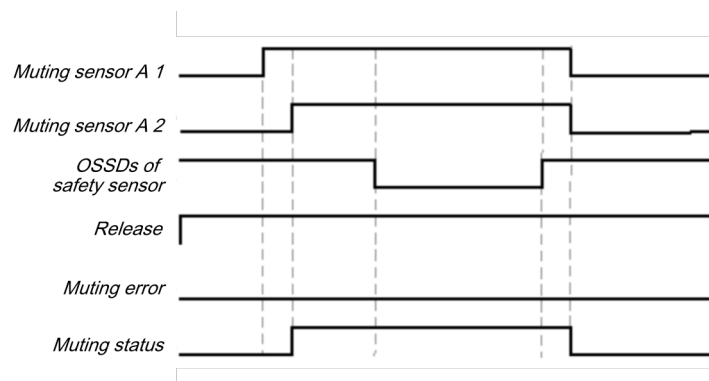


Illustration 140: Valid muting sequence when using the basic configuration setting

## 7.9.9 Cross muting (two-way) – Material transport in both directions

### Function block diagram

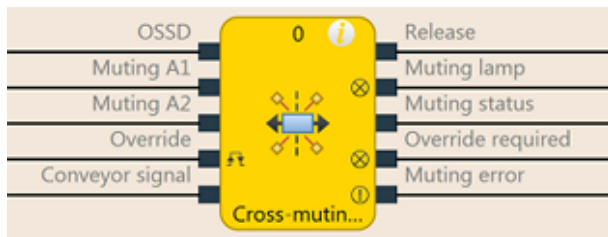


Illustration 141: Logical connections for the cross muting function block (two-side)

### Representation of use

For muting applications with a crossed sensor pair with which material must be moved in both directions, the sensors can be arranged in the following manner.



#### Make sure that the muting sensors only detect the material being moved!

You must ensure that the muting sensors are arranged such that no one can enter the hazardous area by fulfilling the muting conditions (i.e. activate both muting sensors and thus create the prerequisites for muting).

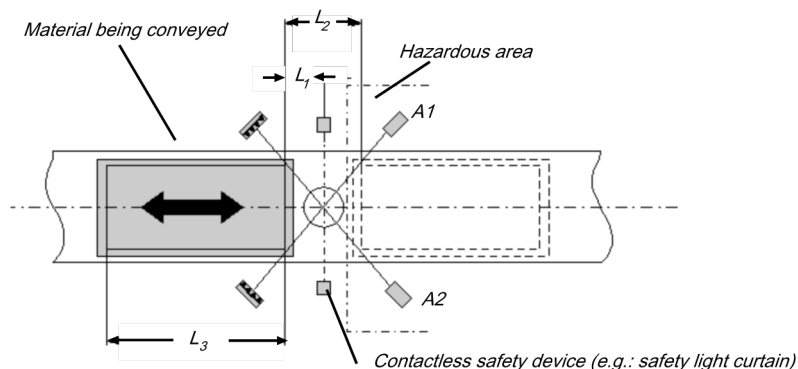


Illustration 142: Cross muting with bidirectional movement of material

You will find a description of the parameters here: *Parameters of the function blocks [ch. 7.9.2, p. 142]*

### Input conditions for muting sensors

Table 98: Conditions for cross muting without the optional C1 input

Condition	Description
A1 & A2	Muting applies as long as this condition is met.

Formulas and requirements for calculating the distance:

$$L_1 \geq v \times T_{\text{IN Muting sensor}}$$

$$v \times t > L_2 + L_3$$

$$T_{\text{IN Light curtain}} < T_{\text{IN Muting sensor}}$$

The following is valid ...

- $L_1$  Minimum distance between the detection line of the contactless safety device and detection through A1, A2
- $L_2$  Distance between the two detection lines of the sensors (sensors activated/sensors free)
- $L_3$  Length of the material in the conveying direction

v	Speed of the material (e.g. of the conveyor belt)
t	Set total muting time [s]
$T_{IN}$ Light curtain	Response time of the light curtain or the muting sensors in the samosPRO system (see "Response times of the samosPRO system" in the "samosPRO hardware" manual (BA000966)).
$T_{IN}$ Muting sensor	

### Info

- In this example, a flow of material is possible in both directions.  
In order to move material in both directions, place the crossing point of the muting sensors precisely in the light beam of the contactless safety device.
- In order to move material only in one direction, place the crossing point in the conveying direction behind the light beam of the contactless safety device (see *Cross muting (one-way) – Movement direction only forwards or only backwards* [ch. 7.9.8, p. 155]).
- The arrangement of the sensors shown in this example is suitable for both through-beam light barriers as well as for reflection light barriers.
- Avoid interaction between sensors.
- You can increase the protection against manipulation and the security with the assistance of the following configurable functions:
  - Simultaneity monitoring time
  - Monitoring of total muting time
  - Muting end through contactless safety device
- You can find information on cabling here: *Information regarding cabling* [ch. 7.9.3, p. 149]

### Sequence/timing diagram

The function block requires that a valid muting sequence take place. The following figure shows an example of a valid muting sequence based on the basic parameter setting for this function block.

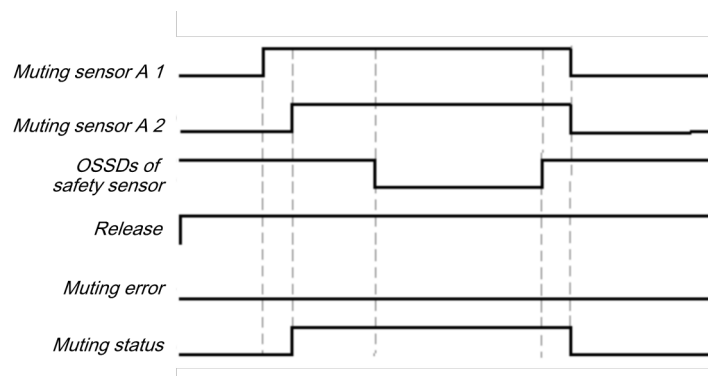


Illustration 143: Valid muting sequence when using the basic configuration setting

## 7.10 Function blocks for presses

### 7.10.1 Function blocks for press contact monitoring

#### 7.10.1.1 Overview and general description

There are two complementary types of function blocks for press applications. This chapter describes the function blocks for contact monitoring which provide signals for the function blocks that control the press cycles of mechanical presses (e.g. eccentric presses) and universal presses.

There are two different function blocks for press contact monitoring that can be used to monitor the correct signal sequence of the contacts and the correct stoppage of the press (after-run). The outputs of these function blocks indicate in which phase of the press cycle the press currently is (e.g. ramp-up or top dead center). Typically the **Enable** output, the **Top** (top dead center) output and the **ramp-up** output of a function block for press contact monitoring are connected to the corresponding inputs of one or more function blocks for press cycle control.

Table 99: Overview of the function blocks for press contact monitoring

	Eccentric press	Universal press
<b>Typical press types</b>	Eccentric press	Eccentric press Mechanical press
<b>Direction of movement of the press</b>	Forwards	Forwards and backwards
<b>Contacts</b>	TDC contact (TopDeadCenter) BDC contact (BottomDeadCenter) Dynamic contact	TDC contact BDC contact After-run
<b>Condition for TDC</b>	If TDC contact = high	If TDC contact = low
<b>Ramp-up condition</b>	If BDC contact = high	If BDC contact = high
<b>After-run monitoring</b>	Optional	Optional
<b>Enable</b>	Mandatory	Mandatory

## 7.10.1.2 Eccentric press contact monitor

### 7.10.1.2.1 Function block diagram

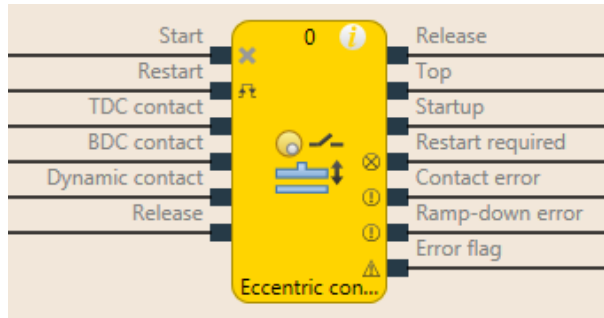


Illustration 144: Logical connections for the eccentric press contact monitor function block

Inputs		
Description	Type	Short description
<b>Start</b>	Mandatory	Connection of a signal that controls the physical output of the press drive, e.g. the enable output from the FB single stroke or automatic.
<b>Restart</b>	Optional	Reset the state of the press block
<b>TDC contact</b>	Mandatory	Contact for the detection of top dead center
<b>BDC contact</b>	Mandatory	Contact for detection of the press ramp-up stroke
<b>Dynamic contact</b>	Optional	Contact for bringing forward of detection of top dead center
<b>Enable</b>	Mandatory	Connection of a signal which stops the press motion, e.g. from shaft breakage detection

Parameters	Possible values
<b>Dynamic contact</b>	<ul style="list-style-type: none"> <li>On: <b>Dynamic contact</b> input is activated.</li> <li>Off: <b>Dynamic contact</b> input is deactivated.</li> </ul>
<b>Restart input</b>	<ul style="list-style-type: none"> <li>On: <b>Restart</b> input is activated.</li> <li>Off: <b>Restart</b> input is deactivated.</li> </ul>
<b>Min. restart pulse time</b>	<ul style="list-style-type: none"> <li>100 ms : <b>Restart</b> input must be at high for at least 100 ms.</li> <li>350 ms : <b>Restart</b> input must be at high for at least 350 ms.</li> </ul>
<b>Use error flag</b>	<ul style="list-style-type: none"> <li>On: <b>Error flag</b> output is present.</li> <li>Off: <b>Error flag</b> output is not present.</li> </ul>

Outputs		
Description	Type	Short description
<b>Enable</b>	Mandatory	Enable of the press process.
<b>Top</b>	Mandatory	The press is in the TDC area.
<b>Ramp-up</b>	Mandatory	The press is in the ramp-up area.
<b>Restart required</b>	Mandatory	The press must be reset because of an error.



Outputs		
Description	Type	Short description
<b>Contact error</b>	Mandatory	Invalid sequence of the contact signals.
<b>After-run error</b>	Mandatory	An after-run error was detected.
<b>Error flag</b>	Optional	A contact error or after-run error is present.

### 7.10.1.2.2 General description

The **Eccentric Press Contact Monitor** function block (FB) can be used for certain types of mechanical presses (e.g. eccentric presses). Besides the **Start** input, the minimum configuration requires the inputs **TDC contact**, **BDC contact** and **enable**. The remaining inputs are optional.

The function block has the special feature that the **Enable** input is set to High from the outset when the inputs do not feature a configuration that results in an error. The signal sequence at the **Start** input is evaluated if a falling edge at the **TDC contact** input is detected and the **BDC contact** input is low (the press is leaving the top dead center).

Another special feature is that the **Restart required** output can be High even if the **Restart** input is *not* present. A reset of this output is then possible only by a stop/run transition of the samos® PRO COMPACT.

A typical sequence of the function block is that the press is at top dead center (**TDC contact** input High) and in the first step therefore the **Enable** and **Top** outputs are High. In this state the **Start** input must either remain High or go through a High-Low sequence. The press now goes through its lifting motion so that the next **TDC contact** input is Low because the press is leaving the top dead center. If the **Start** input has changed its state not as previously specified, the FB enters the **after-run error** and the associated output is high, the **Enable** output is low and **Restart required** high. If there is no error, then the press continues to operate and eventually reaches the bottom dead center to then start the ramp-up stroke. This is reported by the **BDC contact** input, which is high at this point. The **ramp-up** output is also high. If the press approaches the top dead center, the **TDC contact** again becomes high, the **Top** output is high, the **Ramp-up** output is low. Shortly afterwards the **BDC contact** is low and the sequence can begin again. If the contacts do not respond as specified, then the **Contact error** is detected and the associated output is high, **Enable** is low and **Restart required** high.

### 7.10.1.2.3 Parameters of the function block

Table 100: Parameters of the eccentric press contact monitor function block

Parameters	Possible values
<b>Dynamic contact</b>	<ul style="list-style-type: none"> <li>• On: <b>Dynamic contact</b> input is activated.</li> <li>• Off: <b>Dynamic contact</b> input is deactivated.</li> </ul>
<b>Restart input</b>	<ul style="list-style-type: none"> <li>• On: <b>Restart</b> input is activated.</li> <li>• Off: <b>Restart</b> input is deactivated.</li> </ul>
<b>Min. restart pulse time</b>	<ul style="list-style-type: none"> <li>• 100 ms: <b>Restart</b> input must be at high for at least 100 ms.</li> <li>• 350 ms: <b>Restart</b> input must be at high for at least 350 ms.</li> </ul>
<b>Use error flag</b>	<ul style="list-style-type: none"> <li>• On: <b>Error flag</b> output is present.</li> <li>• Off: <b>Error flag</b> output is not present.</li> </ul>

#### Dynamic contact parameter

The **Dynamic Contact** parameter specifies whether the start of the TDC phase can be advanced by a falling edge at the **Dynamic Contact** input.

If the **Dynamic Contact** parameter is set to **On** then the start of the TDC phase is advanced by a falling edge at the **Dynamic Contact** input. In this case the **BDC contact** and **TDC contact** inputs must behave as described in the **General description** section. The falling edge at the **Dynamic contact** input ends the ramp-up phase, the **Ramp-up** output is high, the **Top** output is high.

#### Restart input parameter

This parameter activates the **Restart** input. If the **Restart** input is present, then errors can be reset by a valid restart sequence. A valid restart sequence consists of a low-high-low sequence on the **Restart** input with a high time of 100 ms or 350 ms, depending on the configuration. Shorter high times or high-times longer than 30 seconds are ignored.

#### Min. restart pulse time parameter

A restart sequence (low-high-low **Restart** input) is only valid if the **Restart** input was set to high for at least 100 ms or 350 ms. This value is set with the **Min. restart pulse time** parameter.

#### Use error flag parameter

This parameter makes an additional output available that is set to high when the function block enters an error state.

## 7.10.1.2.4 Function block inputs

Table 101: Inputs of the eccentric press contact monitor function block

Description	Type	Short description
<b>Start</b>	Mandatory	Connection of a signal that controls the physical output of the press drive, e.g. the enable output from the FB single stroke or automatic.
<b>Restart</b>	Optional	Reset the state of the press block
<b>TDC contact</b>	Mandatory	Contact for the detection of top dead center
<b>BDC contact</b>	Mandatory	Contact for detection of the press ramp-up stroke
<b>Dynamic contact</b>	Optional	Contact for bringing forward of detection of top dead center
<b>Enable</b>	Mandatory	Connection of a signal which stops the press motion, e.g. from shaft breakage detection

### Start input

The **Start** input is used to monitor the press after-run. The input must be connected to the signal that controls the physical output of the press drive so that the FB can detect whether the press is currently running or has been stopped.

#### NOTICE

Do *not* connect any physical input signals to the Start input. Use a CPU flag to connect the signal that controls the physical output for the press drive.

### Restart input

Using the Restart input it is possible to reset errors by a valid restart sequence.

A valid restart sequence at the Restart input corresponds to the low-high-low transition with a pulse duration of at least 100 ms or 350 ms and a maximum of 30 s. Shorter or longer pulses are ignored.

If the Restart input is deactivated, then an error can only be reset by stopping execution of the logic program, e.g. by briefly switching off and back on again or by switching the system from the run state to the stop state and then back to the run state using the samos® PLAN5+.



**ATTENTION**

#### **Make sure that the transitions of the signals for reset meet the requirements!**

When there is 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 as a result of short-circuit detection. If this type of pulse can lead to a hazard-inducing state in the machine, then the following points should be noted:

- Ensure that the cable routing is protected for the signal lines (due to cross-connection with other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.



**ATTENTION**

#### **Safety signals must conform to the applicable standards and regulations.**

Always take into account the valid national, regional and local regulations and standards for your application. Type C standards such as EN 692 and EN 693 contain requirements as to how safety signals must be used. For example, in the case of after-run errors it may be necessary that the restart signal is suitably protected (e.g. by a key switch or in a locked cabinet).

### TDC contact and BDC contact inputs

These inputs are used for contact monitoring. The input signals for the **TDC contact**, **BDC contact** and **Start** inputs must match the illustration below and the rules described therein.

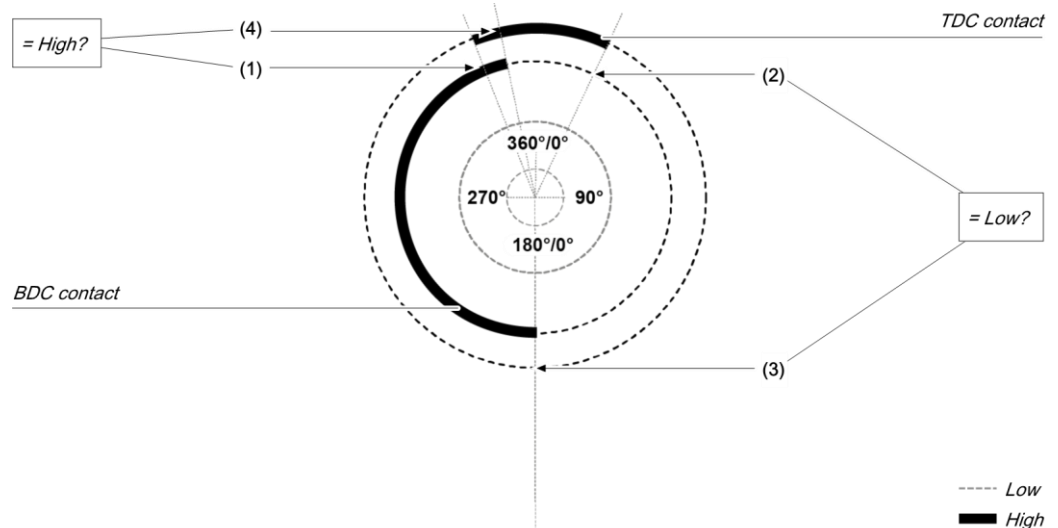


Illustration 145: Contact monitoring with the eccentric press contact monitor function block

Item	Explanation
(1)	The after-run must start during the ramp-up phase: The rising edge at the <b>TDC contact</b> input (low-high transition) must occur while the <b>BDC contact</b> input is high.
(2)	The after-run must end after the end of the ramp-up phase: The falling edge at the <b>TDC contact</b> input (high-low transition) must occur if the <b>BDC contact</b> input is low.
(3)	The ramp-up phase must begin after the after-run has ended: The rising edge at the <b>BDC contact</b> input (low-high transition) must occur while the <b>TDC contact</b> input is low.
(4)	The ramp-up phase must end during the after-run: The falling edge at the <b>BDC contact</b> input (high-low transition) must occur while the <b>TDC contact</b> input is high.

If during operation only one of these conditions is not met, the **Enable** output is low and the **Contact error** is high.

A valid sequence that satisfies these conditions looks as follows:

Step	System behavior
0)	Start condition: <b>TDC contact</b> input = high, <b>BDC contact</b> input = low, Start = high (or sequence low → high (→ Low))
1)	<b>TDC contact</b> input: High → Low
2)	<b>BDC contact</b> input: Low → High
3)	<b>TDC contact</b> input: Low → High
4)	<b>BDC contact</b> input: High → Low



### Be aware of the corresponding standards and safety regulations!

All of the safety-related parts of the system (wiring, connected sensors and control devices, configuration) must meet the respective standards (e.g. EN 62061 or EN ISO 13849-1 or type C standards such as EN 692 and EN 693) and safety regulations. Only safety signals may be used for safety applications. Make sure that the application meets all of the applicable standards and regulations.

This must be noted in particular for the **BDC contact** input if the **Ramp-up** output is being used for ramp-up muting, e.g. in connection with a function block for press cycle control.

In order to satisfy the safety regulations, it may be necessary to use tested switches each with different test sources for the contact input signals. In order to use different test sources for the contact signals, the **TDC contact**, **BDC contact** and **Dynamic contact** inputs must be connected to the different SP-SDI or SP-SDIO modules.

**NOTICE** An SP-SDI module has only two test sources although it has eight test output terminals.

### Description of after-run monitoring

The eccentric press contact monitor function block monitors the press after-run. If the **TDC contact** is left, although the press would actually have to have stopped, then the function block detects an after-run error, and the **After-run error** is set to high.

The **Start** input must then be equal to the following illustration and rule:

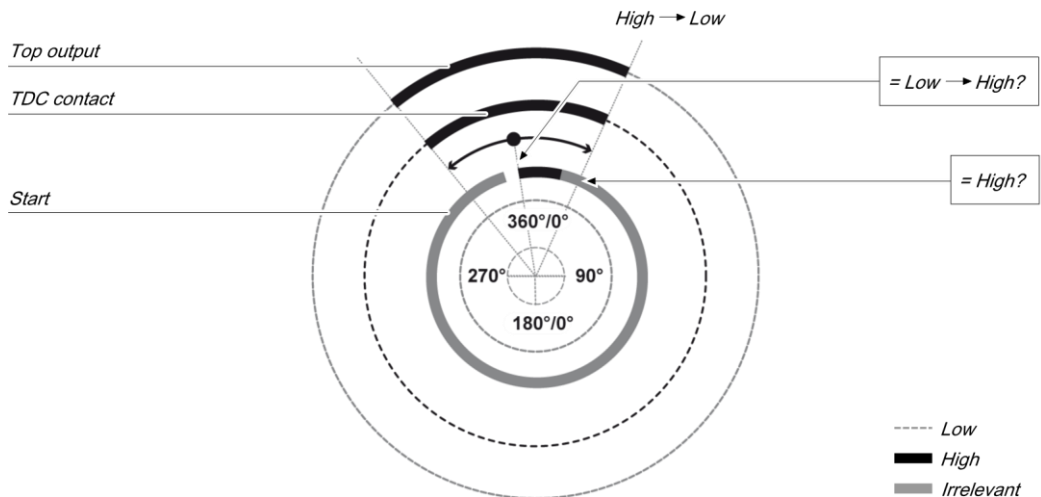


Illustration 146: After-run monitoring with the eccentric press contact monitor function block

Either at least one rising edge must occur at the **Start** input while the **Top** output is high or the **Start** input must be high at the end of the after-run area (falling edge at **Top** output). If neither of these two conditions is met, then the **Enable** output is low and the **After-run error** and **Restart required** outputs are high.

The **Start** input must be connected to the signal that controls the physical output of the press drive so that the function block can detect whether the press is currently running or has been stopped. Typically this is the **Enable** output of a downstream press set-up or press single stroke function block.

**NOTICE** Do *not* connect any physical input signals to the **Enable drive** input. Use a CPU flag to connect the signal that controls the physical output for the press drive.

## Dynamic contact input

Using the dynamic contact the start of the top phase can be advanced (falling edge at the **Dynamic contact** input).

The Ramp-up output moves to high in the case of a rising edge (transition from low to high) at the **BDC contact** input. It moves to low either in the event of a rising edge of the **TDC contact** input or a falling edge at the **Dynamic contact** input, depending on which of these occurs first.

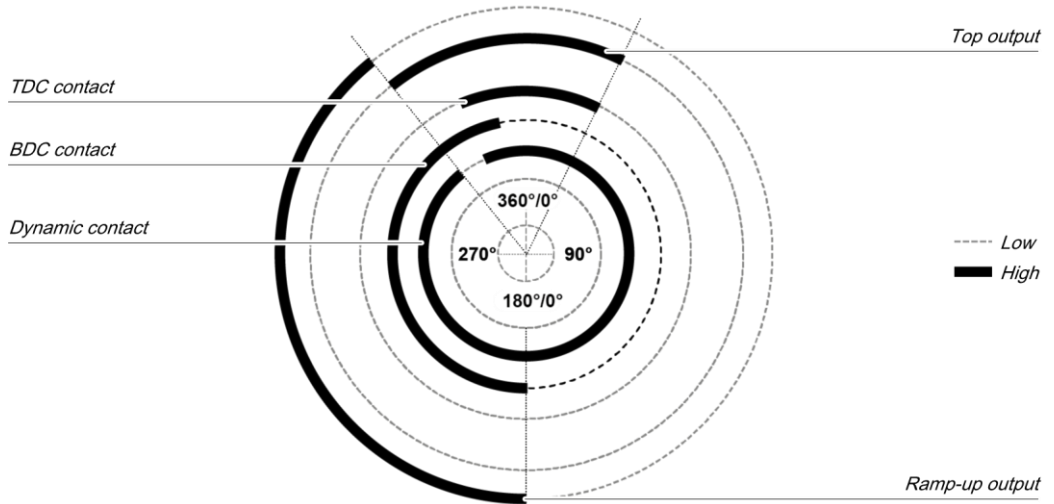


Illustration 147: Press cycle for the eccentric press contact monitor function block with dynamic contact on the upstroke

If a falling edge occurs at the **Dynamic contact** input while the **BDC contact** input is low, i.e. during the downward phase of the press cycle, then the **Top** output moves to high until a rising edge is detected at the **BDC contact** input. The Ramp-up output remains low during the remaining press cycle.

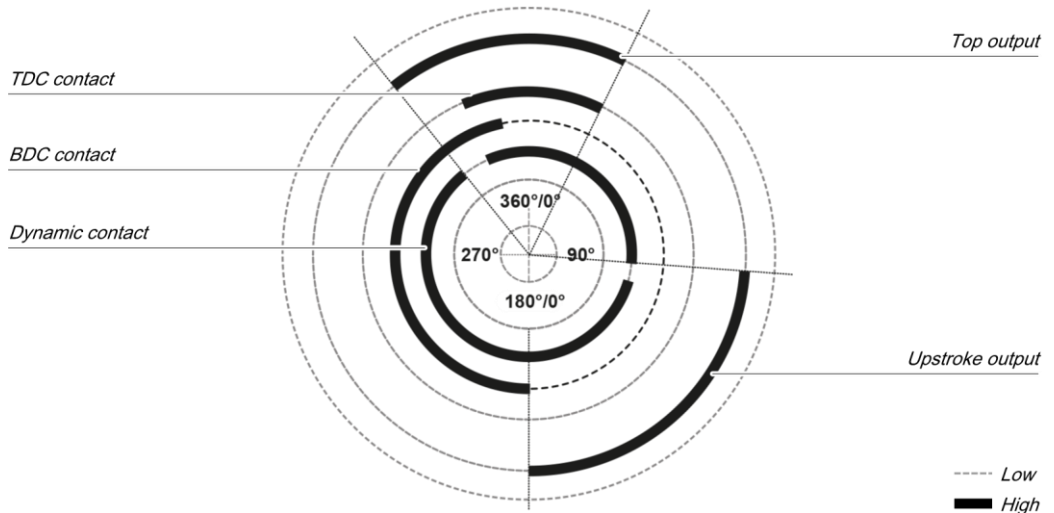


Illustration 148: Press cycle for the eccentric press contact monitor function block with dynamic contact on ramp-up and the downward movement

## Enable input

This input is used to connect a shaft breakage detection.

If the **Enable** input is low, then the **Enable** output of the function block is low and monitoring of the contact signal sequence and the after-run is deactivated, assuming that there is no error. The error outputs are not affected by this.

If the **Enable** input changes from low to high, then depending on the configuration of the inputs the suitable state is assumed and the outputs are actuated in accordance with the state.

## 7.10.1.2.5 Function block outputs

Table 102: Outputs of the eccentric press contact monitor function block

Description	Type	Short description
<b>Enable</b>	Mandatory	Enable of the press process.
<b>Top</b>	Mandatory	The press is in the TDC area.
<b>Ramp-up</b>	Mandatory	The press is in the ramp-up area.
<b>Restart required</b>	Mandatory	The press must be reset because of an error.
<b>Contact error</b>	Mandatory	Invalid sequence of the contact signals.
<b>After-run error</b>	Mandatory	An after-run error was detected.
<b>Error flag</b>	Optional	A contact error or after-run error is present.

### Release output

The **Enable** output is used to stop the press and is connected to another supplementary press function block such as press set-up or press single stroke. If no error has been detected, the **Enable** output of the function block is high.

If an error is detected in the sequence of contact signals, the **Enable** output moves to low, the affected error output moves to high and the **Reset required** output moves to high. A valid restart sequence at the **Restart** input is then required.

The **Enable** output moves to low when the **Enable** input is deactivated.

### Top output

The **Top** output moves to high in the event of a rising edge at the **TDC contact** input or a falling edge at the **Dynamic contact** input (the **Enable** input cannot have a falling edge), depending on which of them occurs first. The **Top** output moves to low in the event of a falling edge at the **TDC contact** input.

The **Top** output is typically used to stop the press and is connected to another supplementary press function block such as press set-up or press single stroke.

### Ramp-up output

The **Ramp-up** output is typically connected to another supplementary press function block such as press set-up or press single stroke. It can also be used to trigger ramp-up muting.

This function block sets the Ramp-up and Top outputs based on the state changes at the contact inputs. If the function block does not detect an error, both outputs are set to low.

### Restart required output

The **Restart required** output is high if a valid restart sequence is expected at the **Restart** input. This output is set back to low only after a valid restart sequence and the **Enable** output cannot be high while this output is high. A valid restart sequence is described in the **Restart input** and **Min. restart pulse time parameter** sections.

### Contact error output

This output is set to high when the predetermined sequence of the contact states is not respected. The valid sequences were described in the **TDC contact input**, **BDC contact input** and **Dynamic contact input** sections. Above all, activation of the **Dynamic contact** parameter changes the contact sequence. All variants of an invalid sequence lead to errors and the Contact error output is set to high.

## After-run error output

This output is set to high when the after-run monitoring detects an unexpected movement of the press. If the TDC contact is left, although the press should have been stopped, this input is set to high.

## Error flag output

This output is set to high if any error is present or if at least one of the contact error or after-run error outputs is set to high and the output is configured as active.

### 7.10.1.2.6 Error states and information regarding reset

Table 103: Error states and information regarding reset for the universal press function block

Outputs	Error state reset	Remarks
<b>Contact error</b>	The contact error output is high if an unexpected signal sequence has been detected. Resetting requires a valid restart sequence.	The Enable output is low, the Restart required output is high. If the Error flag output is present, this is high.
<b>After-run error</b>	The <b>After-run error</b> output is high if the TDC contact is left although the press should actually have been stopped. Resetting requires a valid restart sequence.	

### 7.10.1.2.7 Example sequence of a press cycle

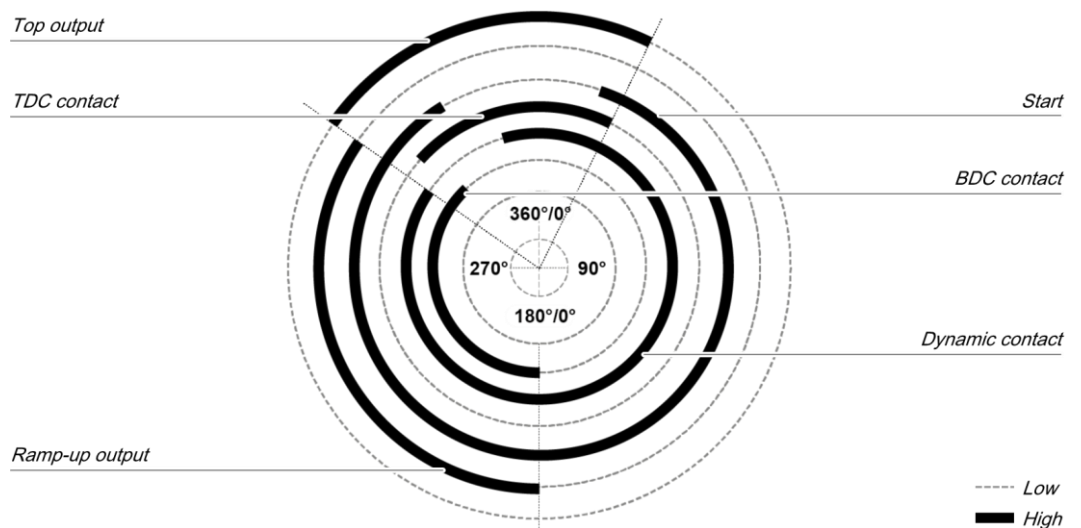


Illustration 149: Contact and output sequence of an eccentric press during an error-free cycle (example)



## 7.10.1.3 Universal press contact monitor

### 7.10.1.3.1 Function block diagram

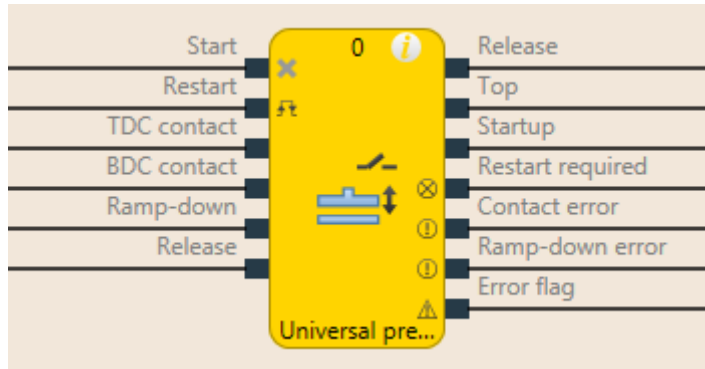


Illustration 150: Logical connections for the universal press contact monitor function block

Inputs		
Description	Type	Short description
<b>Start</b>	Mandatory	Connection of a signal that controls the physical output of the press drive, e.g. the enable output from the FB single stroke or automatic.
<b>Restart</b>	Optional	Reset the state of the press.
<b>TDC contact</b>	Mandatory	Contact for the detection of top dead center.
<b>BDC contact</b>	Mandatory	Contact for ramp-up area.
<b>After-run contact</b>	Optional	Contact for after-run area.
<b>Enable</b>	Mandatory	Connection of a signal which stops the press motion, e.g. from shaft breakage detection

Parameters	Possible values
<b>Restart input</b>	<ul style="list-style-type: none"> <li>On: <b>Restart</b> input activated</li> <li>Off: <b>Restart</b> input deactivated</li> </ul>
<b>Ramp-up signals per cycle</b>	<ul style="list-style-type: none"> <li>0-2 (e.g. universal press)</li> <li>1 (e.g. eccentric press)</li> </ul>
<b>Min. restart pulse time</b>	<ul style="list-style-type: none"> <li>100 ms</li> <li>350 ms</li> </ul>
<b>Use error flag</b>	<ul style="list-style-type: none"> <li>On: <b>Error flag</b> output activated</li> <li>Off: <b>Error flag</b> output deactivated</li> </ul>

Outputs		
Description	Type	Short description
<b>Enable</b>	Mandatory	Enable of the press process.
<b>Top</b>	Mandatory	The press is in the TDC area.
<b>Ramp-up</b>	Mandatory	The press moves upwards.
<b>Restart required</b>	Mandatory	The press must be reset because of an error.

Outputs		
Description	Type	Short description
Contact error	Mandatory	Invalid sequence of the contact signals.
After-run error	Optional	An after-run error was detected.
Error flag	Optional	A contact error or after-run error is present.

### 7.10.1.3.2 General description

The Universal Press Contact Monitor function block can be used for different types of presses (e.g. mechanical presses). The minimum configuration requires the inputs **Start**, **TDC contact**, **BDC contact**, **After-run contact**, **Enable**. Optionally, the **Restart** input can be connected.

#### Description of after-run

If the **After-run contact** input is activated, then the input signals for **After-run contact** must match the following illustration and the following rules:

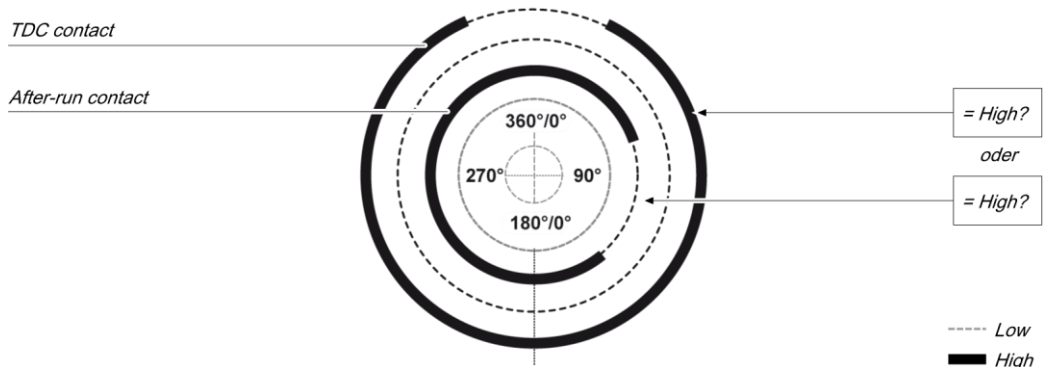


Illustration 151: Contact monitoring with the universal press contact monitor function block with activated after-run

Precisely one pulse must occur at the **After-run** input per cycle. The rising edge at the **After-run** input (transition low-high) must happen before the falling edge at the **TDC contact** input. The falling edge at the **After-run** input (transition high-low) must happen after the rising edge at the **TDC contact** input. This means that at any time at least one of the two inputs must be high.

### 7.10.1.3.3 Parameters of the function block

Table 104: Parameters of the universal press contact monitor function block

Parameters	Possible values
Restart input	<ul style="list-style-type: none"> <li>On: <b>Restart</b> input activated</li> <li>Off: <b>Restart</b> input deactivated</li> </ul>
Ramp-up signals per cycle	<ul style="list-style-type: none"> <li>0-2 (e.g. universal press)</li> <li>1 (e.g. eccentric press)</li> </ul>
Min. restart pulse time	<ul style="list-style-type: none"> <li>100 ms</li> <li>350 ms</li> </ul>
Use error flag	<ul style="list-style-type: none"> <li>On: <b>Error flag</b> output activated</li> <li>Off: <b>Error flag</b> output deactivated</li> </ul>
After-run contact	<ul style="list-style-type: none"> <li>On: <b>After-run contact</b> input activated</li> <li>Off: <b>After-run contact</b> input deactivated</li> </ul>

## Restart input parameter

This parameter activates the **Restart** input. If the **Restart** input is present, then errors can be reset by a valid restart sequence. A valid restart sequence consists of a low-high-low sequence on the Restart input with a high time of 100 ms or 350 ms, depending on the configuration. Shorter high times or high-times longer than 30 seconds are ignored.

## Ramp-up signals per cycle parameter

The **BDC contact** input signals that the press has reached the BDC area (bottom dead center). This happens when a rising edge is detected at the **BDC contact** input while the **TDC contact** input is high. If the parameter is set to **1 (e.g. eccentric press)** then this signal must occur exactly once during the press cycle. The **TDC contact** input cannot therefore immediately change to low without the **BDC contact** input having had a rising edge at least once. A deviation from these processes would result in a contact error.

If the parameter is set to **0-2 (e.g. universal press)**, it is possible to deviate from this cycle in the framework of the configuration. In other words 0 rising edges at BDC, 1 rising edge at BDC or two rising edges at BDC are allowed. During the first rising edge at BDC the **Ramp-up** output is set to high. If two rising edges occur, then **Ramp-up** is low with the first falling edge from BDC and no longer high with the second rising edge.

## Description of ramp-up monitoring

If the **Ramp-up signals per cycle** parameter is set to 1, then the input signals for **BDC contact** must match the following illustration and the following rules:

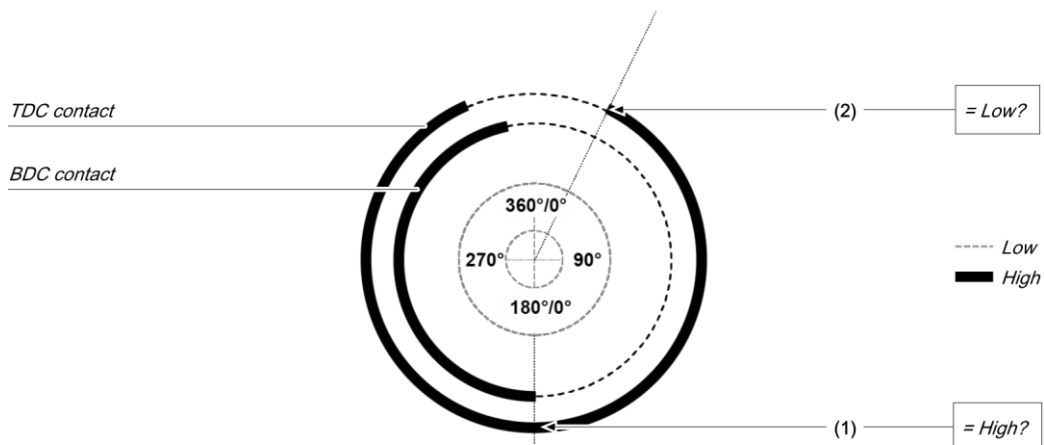


Illustration 152: Contact monitoring with the universal press contact monitor function block with activated BDC contact

Item	Description
(1)	The start of the signal at the <b>BDC contact</b> (rising edge) must be close to 180° and must occur while the <b>TDC contact</b> is high. The rising edge at the <b>BDC contact</b> switches the ramp-up output to high.
(2)	The falling edge at the <b>BDC contact</b> (transition high-low) must happen before the rising edge (transition low-high) at the <b>TDC contact</b> input. This means that the <b>BDC contact</b> input must be low if a rising edge (transition low-high) occurs at the <b>TDC contact</b> input.

This cycle can be changed with the **Ramp-up signals per cycle** parameter if it is configured to 0-2 (see previous section and BDC contact input section).

### Min. restart pulse time parameter

A restart sequence (low-high-low **Restart** input) is only valid if the **Restart** input was set to high for at least 100 ms or 350 ms. This value is set with the **Min. restart pulse time** parameter.

### Use error flag parameter

This parameter makes an additional output available that is set to high when the function block enters an error state.

### Input signals of the universal press contact monitor function block

Description	Type	Short description
<b>Start</b>	Mandatory	Connection of a signal that controls the physical output of the press drive, e.g. the enable output from the FB single stroke or automatic.
<b>Restart</b>	Optional	Reset the state of the press.
<b>TDC contact</b>	Mandatory	Contact for the detection of top dead center.
<b>BDC contact</b>	Mandatory	Contact for ramp-up area.
<b>After-run contact</b>	Optional	Contact for after-run area.
<b>Enable</b>	Mandatory	Connection of a signal which stops the press motion, e.g. from shaft breakage detection

### Start input

The **Start** input is used to monitor the press after-run. The input must be connected to the signal that controls the physical output of the press drive so that the FB can detect whether the press is currently running or has been stopped. Typically this is the **Enable** output of a downstream press set-up, press auto mode or press single stroke function block.

**NOTICE** Do not connect any physical input signals to the **Start** input. Use a CPU flag to connect the signal that controls the physical output for the press drive.

## Restart input

Using the Restart input it is possible to reset errors by a valid restart sequence.

A valid restart sequence at the **Reset** input corresponds to a low-high-low transition with a pulse duration of at least 100 ms or 350 ms and a maximum of 30 s. Shorter or longer pulses are ignored.

If the Restart input is deactivated, then an error can only be reset by stopping execution of the logic program, e.g. by briefly switching off and back on again or by switching the system from the run state to the stop state and then back to the run state using the samos® PLAN5+.



**ATTENTION**

### Make sure that the transitions of the signals for reset meet the requirements!

When there is 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 as a result of short-circuit detection. If this type of pulse can lead to a hazard-inducing state in the machine, then the following points should be noted:

- Ensure that the cable routing is protected for the signal lines (due to cross-connection with other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.



**ATTENTION**

### Safety signals must conform to the applicable standards and regulations.

Always take into account the valid national, regional and local regulations and standards for your application. Type C standards such as EN 692 and EN 693 contain requirements as to how safety signals must be used. For example, in the case of after-run errors it may be necessary that the restart signal is suitably protected (e.g. by a key switch or in a locked cabinet).

## TDC contact input

Precisely one pulse must occur at the **TDC contact** input per cycle.

## BDC contact input

If the **BDC contact** input is high at the start of the function block (switch-on), then the **Ramp-up** output remains low during the first press cycle.

The diagrams below show the press cycle with differently designed switching gates for the BDC contact (falling edge of BDC after falling edge of TDC contact or falling edge BDC leading the falling edge of the TDC contact)

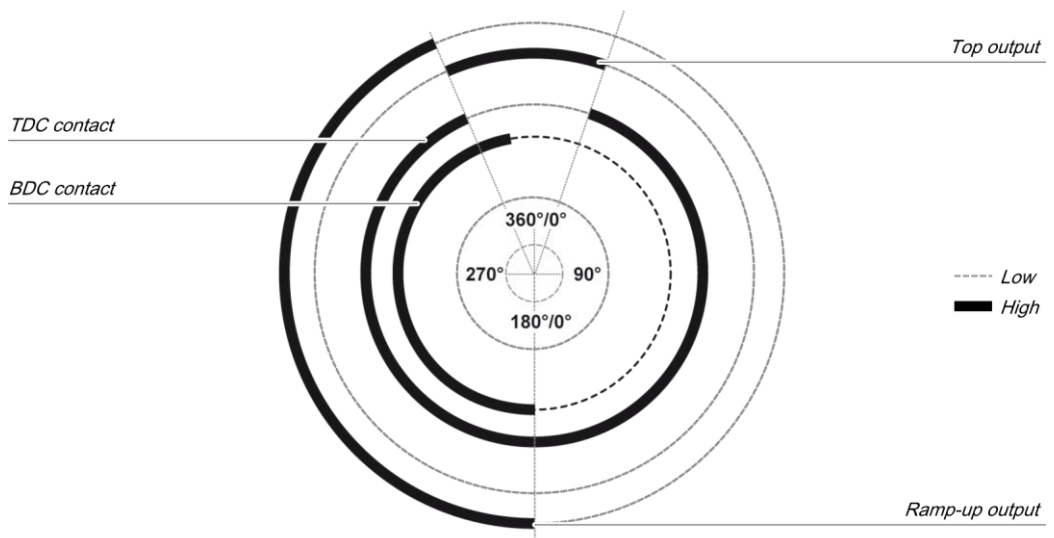


Illustration 153: Press cycle for the universal press contact monitor function block with falling edge of **TDC contact** before **BDC contact**

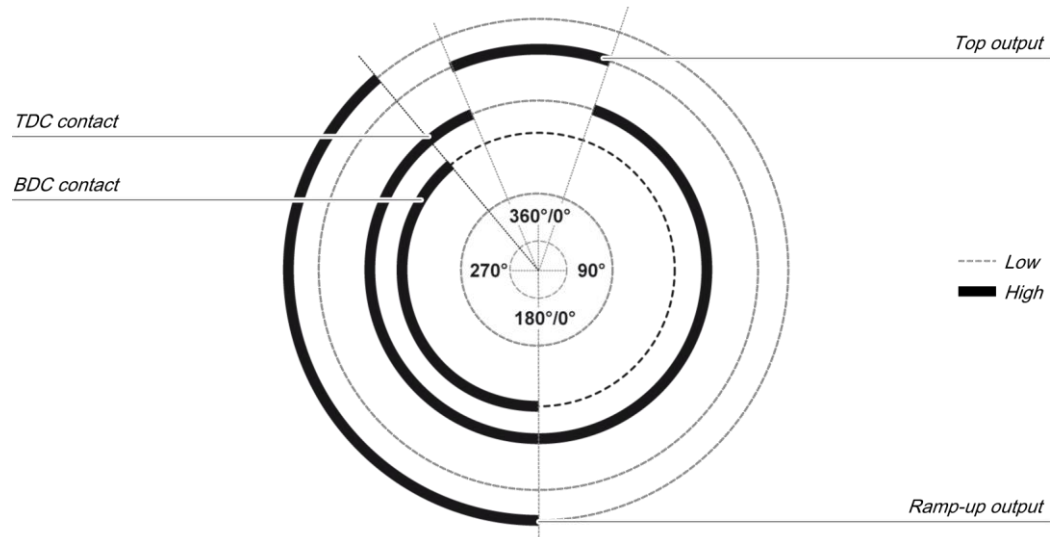


Illustration 154: Press cycle for the universal press contact monitor function block with falling edge of **BDC contact** before **TDC contact**

A second rising edge at the **BDC contact** input does not re-start the upstroke phase. This is the case if the **Number of BDC signals per cycle** parameter is configured to 0-2 (e.g. universal press) and the press is moved forwards and backwards in the lower section.

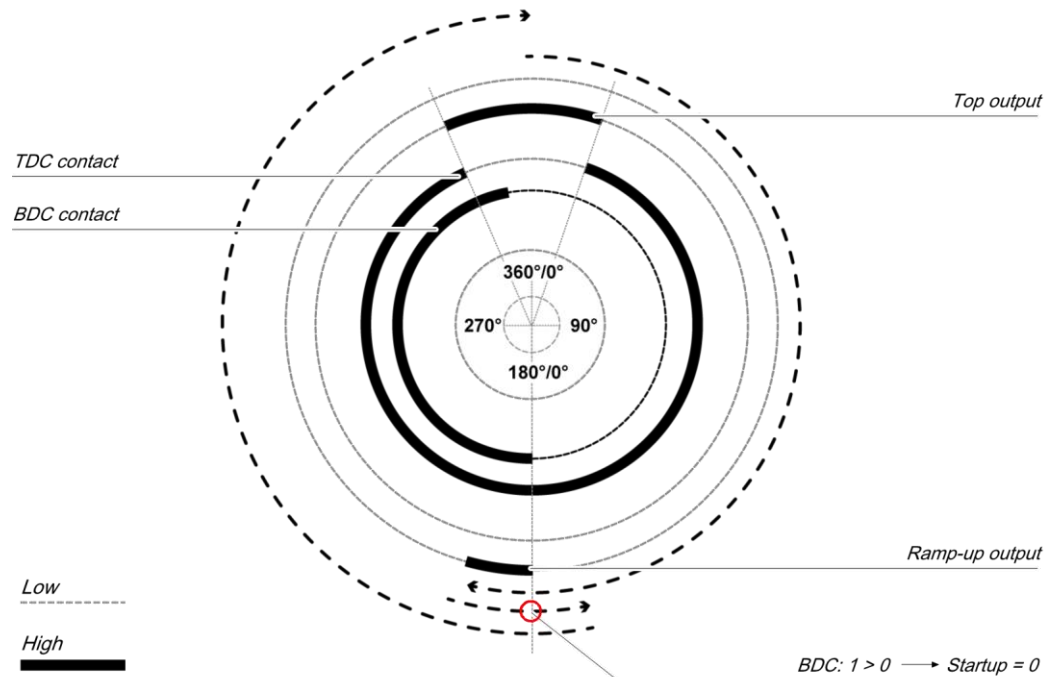


Illustration 155: Press cycle for the universal press contact monitor function block with 2 BDC transitions

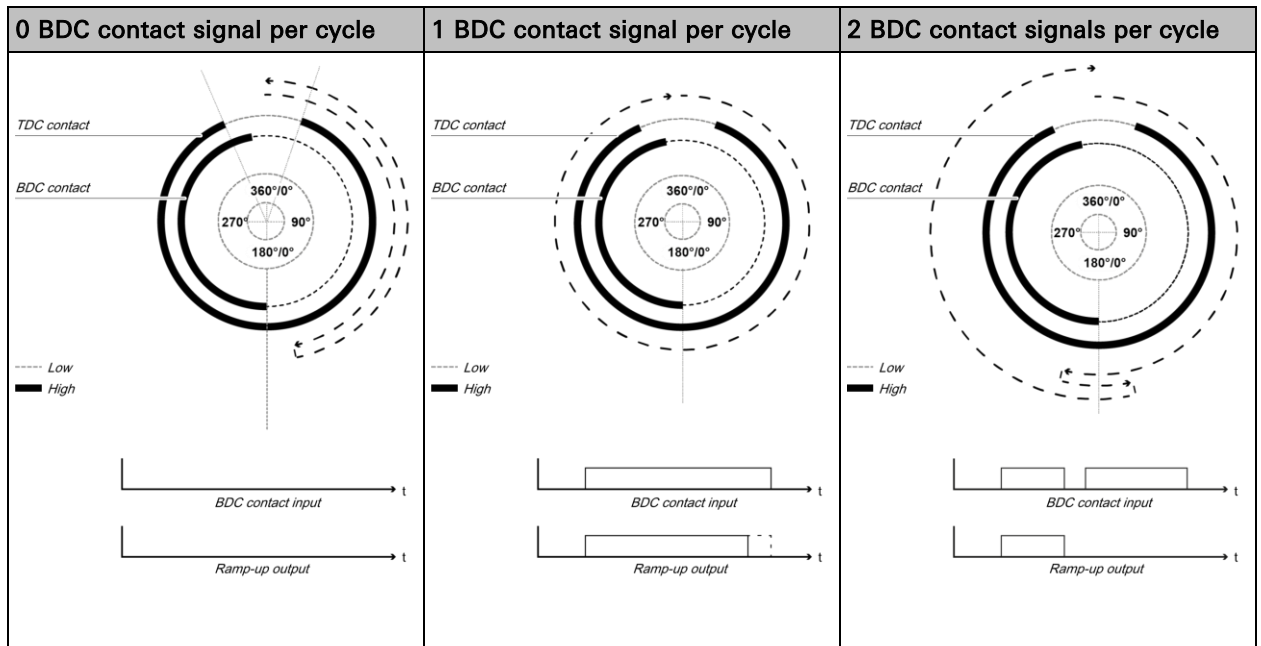
If no pulse occurs at the **BDC contact** input during the cycle with this setting, then the **Ramp-up** output remains low during the cycle.

**NOTICE** If the **BDC contact** input is already high when monitoring of the contact inputs begins (e.g. during the first logic cycle after resetting an error) then the **Ramp-up** output remains low during the first logic cycle. The next transition from low to high at the **BDC contact** input is only accepted if previously a transition from high to low has occurred at the **Top** output.

# Logic programming – Function blocks

The following illustrations show the different press operations with 0, 1 and 2 BDC contact sequences.

Table 105: Timing diagrams for 0, 1 and 2 BDC contact signals per cycle



The **BDC contact** input can be combined with the **After-run** input. In this case the processes described in the **BDC contact** and **After-run inputs** section apply.

### After-run contact input

The universal press contact monitor function block monitors the press after-run. If the **After-run contact** is left although the press should actually have been stopped, then the function block detects an after-run error.

The **Start** input must then be equal to the following illustration and following rules:

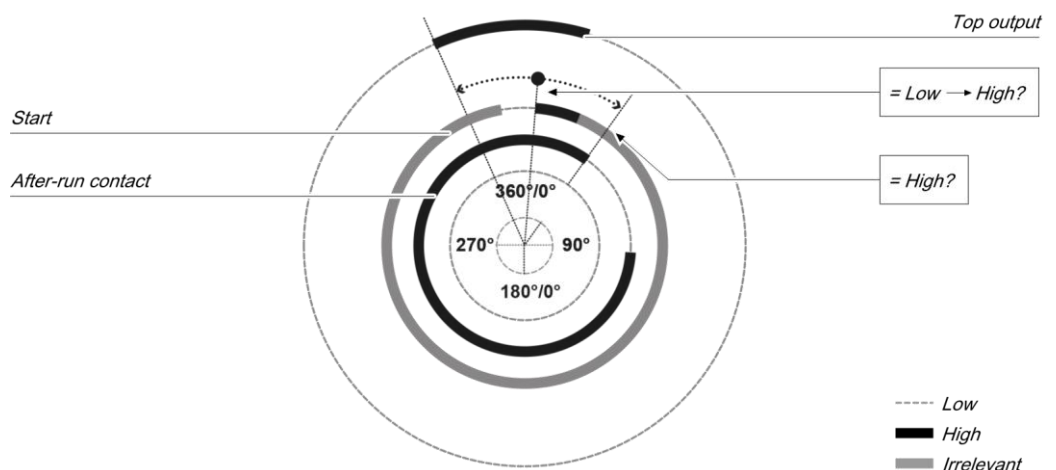


Illustration 156: After-run monitoring with the universal press contact monitor function block

Either a rising edge must occur at **Start** between the rising edge of the **Top** output and the end of the after-run area (falling edge at the **After-run contact** input) or the **Start** input must be high at the end of the after-run area (falling edge at the **After-run contact** input). If neither of these two conditions is met, then the **Enable** output is low and the **After-run error** and **Restart required** outputs are high.

The **After-run contact** input is also combined with the **BDC contact** input. The processes described in the **BDC contact** and **After-run inputs** section apply.

### Enable input

This input is used to connect a shaft breakage detection.

If the **Enable** input is low, then the **Enable** output of the function block is low and monitoring of the contact signal sequence and the after-run is deactivated, assuming that there is no error. The error outputs are not affected by this.

If the **Enable** input changes from low to high, then depending on the configuration of the inputs the suitable state is assumed and the outputs are actuated in accordance with the state.

If the **Enable** input changes from high to low, then the **Enable**, **Top** and **Ramp-up** outputs are inactive. The error outputs are not affected by this. If the **Enable** input is active again, then the outputs assume their corresponding state. Monitoring is not deactivated during an active **Enable** input.

#### 7.10.1.3.4 Output signals of the universal press contact monitor function block

Description	Type	Short description
<b>Enable</b>	Mandatory	Enable of the press process.
<b>Top</b>	Mandatory	The press is in the TDC area.
<b>Ramp-up</b>	Mandatory	The press moves upwards.
<b>Restart required</b>	Mandatory	The press must be reset because of an error.
<b>Contact error</b>	Mandatory	Invalid sequence of the contact signals.
<b>After-run error</b>	Optional	An after-run error was detected.
<b>Error flag</b>	Optional	A contact error or after-run error is present.

### Release output

The **Enable** output is used to stop the press and is connected to another supplementary press function block such as press set-up or press single stroke. If no error has been detected, the **Enable** output of the function block is high.

If an error is detected in the sequence of contact signals, the **Enable** output moves to low, the affected error output moves to high and the **Restart required** output moves to high. A valid restart sequence at the **Restart** input is then required.

The **Enable** output moves to low when the **Enable** input is low.

### Top output

The **Top** output is typically used to stop the press and is connected to another supplementary press function block such as press set-up or press single stroke.

This function block sets the **Top** output based on the changes to the values at the contact inputs. If the function block detects an error, the output is set to low. The **Top** output moves to high when the **TDC contact** input is low.

The **Top** output moves to low when the **Enable** input is low.

### Ramp-up output

The **Ramp-up** output is typically connected to another supplementary press function block such as press set-up or press single stroke. It can also be used to trigger ramp-up muting.

This function block sets the **Ramp-up** output based on the changes to the values at the contact inputs. If the function block detects an error, the output is set to low.

The **Ramp-up** output moves to high in the case of a rising edge (transition from low to high) at the **BDC contact** input. It moves to low in the event of a rising edge at the **TDC contact** input or a falling edge at the **BDC contact** input, depending on which of these occurs first.

The **Ramp-up** output moves to low when the **Enable** input is low.



## Restart required output

The **Restart required** output is high if a valid restart sequence is expected at the **Restart** input. This output is set back to low only after a valid restart sequence and the **Enable** output cannot be high while this output is high. A valid restart sequence is described in the **Restart, Restart input** and **Min. restart pulse time parameter** sections.

## Contact error output

This output is set to high when the predetermined sequence of the contact states is not respected. The valid sequences were described in the **After-run input, TDC contact input, BDC contact input** section and the **BDC contact and After-run inputs** section. All variants of an invalid sequence lead to errors and the Contact error output is set to high.

## After-run error output

This output is set to high when the after-run contact detects an unwanted movement of the press. If the After-run contact is left, although the press should actually have been stopped, this input is set to high.

## BDC contact and After-run contact inputs

The signals at the **TDC contact, BDC contact** and **After-run contact** inputs must match the illustration below and the following rules:

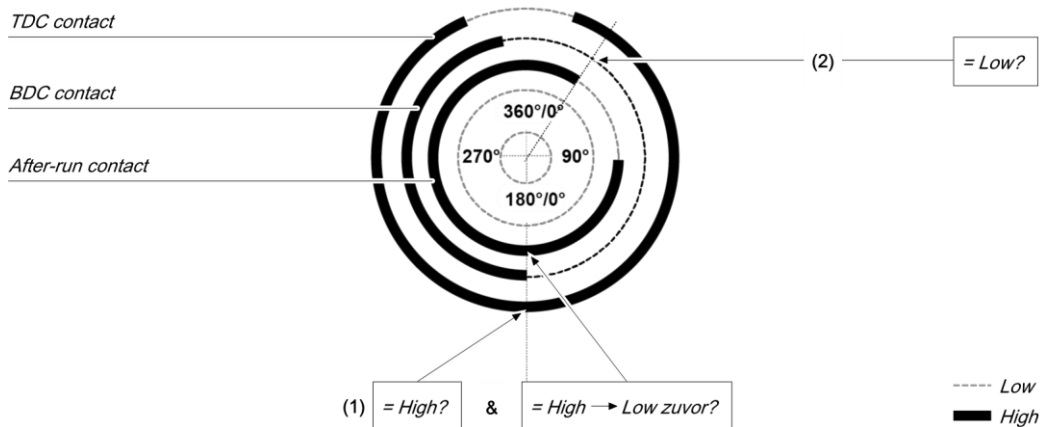


Illustration 157: Contact monitoring with the universal press contact monitor function block with activated BDC contact and after-run

Item	Description
(1)	The beginning of the signal at the <b>BDC contact</b> (transition low-high) must be close to 180° and must occur while the <b>TDC contact</b> input is high and after the falling edge (transition high-low) of the <b>After-run contact</b> input (the <b>After-run contact</b> input may have moved back to high in the meantime).
(2)	The end of the signal at the <b>BDC contact</b> (transition high-low) must happen before the falling edge (transition high-low) at the <b>After-run contact</b> input. This means that the <b>BDC contact</b> input must be low if a falling edge (transition high-low) occurs at the <b>After-run contact</b> input.
(3)	The rules for the <b>Start</b> input (see <b>Start input</b> section) must be respected.

A valid sequence that satisfies the conditions for **BDC contact** and **After-run** looks like this:

Step	System behavior
1.	Start condition: <b>TDC contact</b> = low, <b>BDC contact</b> = low, <b>After-run contact</b> = high
2.	<b>TDC contact</b> : Low → High
3.	<b>Start</b> = High (satisfies the condition for after-run monitoring)
4.	<b>After-run contact</b> : High → Low

Step	System behavior
5.	<b>BDC contact:</b> Low → High ( <b>Ramp-up</b> output is high only for the first rising edge)
6.	<b>After-run contact:</b> Low → High
7.	<b>TDC contact:</b> High → Low and <b>BDC contact:</b> High → Low (sequence irrelevant, ramp-up output is low)

Depending on the type of press it can happen that the start of the **BDC contact** signal (step 5 above) does not just occur once, but twice or even not at all. In order to prevent this leading to a contact error, the **Number of BDC signals per cycle** parameter must be configured to 0-2 (e.g. universal press). With this setting the conditions for the **BDC contact** still apply for every pulse at the **BDC contact** input with the exception of the falling edge at the **After-run contact** input (step 4 above).

If during operation only one of these conditions is not met, the **Enable** output is low and the **Contact error** is high.



**ATTENTION**

### Be aware of the corresponding standards and safety regulations!

All of the safety-related parts of the system (wiring, connected sensors and control devices, configuration) must meet the respective standards (e.g. EN 62061 or EN ISO 13849-1 or type C standards such as EN 692 and EN 693) and safety regulations. Only safety signals may be used for safety applications. Make sure that the application meets all of the applicable standards and regulations.

This must be noted in particular for the **BDC contact** input if the **Ramp-up** output is being used for upstroke muting, e.g. in connection with a function block for press cycle control.

If the **Number of BDC signals per cycle** parameter is configured to 0-2 (e.g. universal press) then the possibilities of the function block for error detection are reduced and not all input errors can be detected (e.g. short circuit after 0 V at the **BDC contact** input).

In order to satisfy the safety regulations, it may be necessary to use tested switches each with different test sources for the contact input signals. In order to use different test sources for the contact signals, the **TDC contact**, **BDC contact** and **After-run** inputs must be connected to the different SP-SDI or SP-SDIO modules.

**NOTICE** An SP-SDI module has only two test sources although it has eight test output terminals.

### 7.10.1.3.5 Error states and information regarding reset

Table 106: Error states and information regarding reset for the universal press function block

Outputs	Error state reset	Remarks
<b>Contact error</b>	The <b>contact error</b> output is high if an unexpected signal sequence has been detected. Resetting requires a valid restart sequence.	The <b>Enable</b> output is low, the <b>Restart required</b> output is high. If the <b>Error flag</b> output is present, this is high.
<b>After-run error</b>	The <b>After-run error</b> output is high if the After-run contact is left although the press should actually have been stopped. Resetting requires a valid restart sequence.	

## 7.10.1.3.6 Example sequence of a press cycle

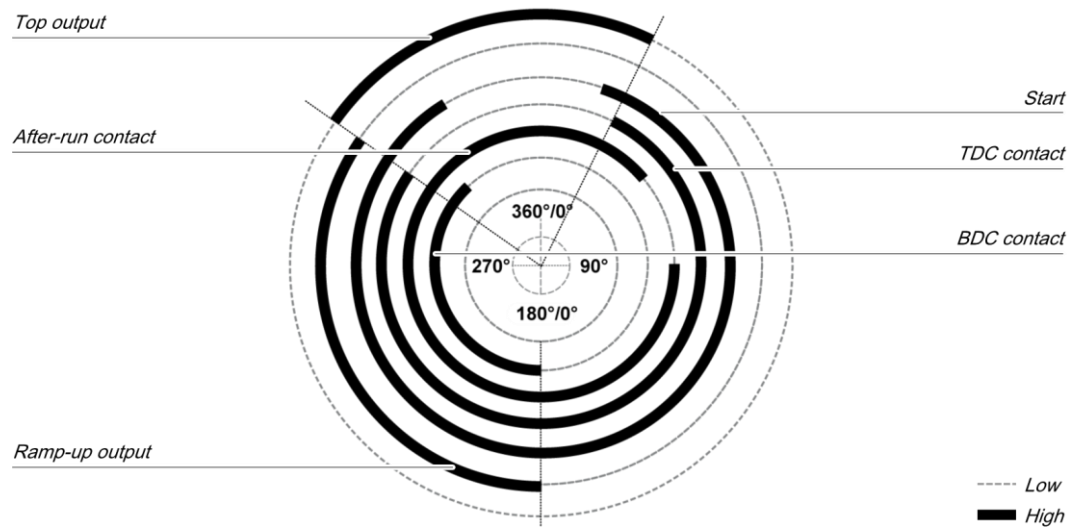


Illustration 158: Contact and output sequence of a universal press during an error-free cycle (example)

## 7.10.2 Function blocks for press cycle control

### 7.10.2.1 Press set-up

#### Function block diagram

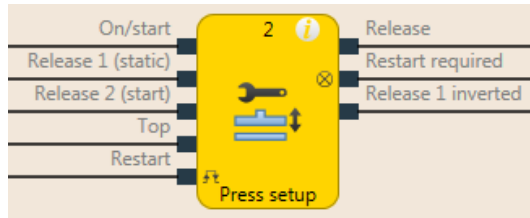


Illustration 159: Logical connections for the press set-up function block

Inputs		
Description	Type	Short description
<b>On/Start</b>	Mandatory	Start a press operation with the rising edge or stop with the falling edge
<b>Enable 1 (static)</b>	Mandatory	Main enable of the press
<b>Enable 2 (start)</b>	Optional	Additional enable input is only evaluated when the Enable output is low.
<b>Top</b>	Optional	Top dead center for single stroke monitoring
<b>Restart</b>	Mandatory	Reset the state of the press

Parameters	Possible values
<b>Restart interlock</b>	<ul style="list-style-type: none"> <li>1- without</li> <li>2 - always</li> <li>3 - if <b>On/Start</b> or <b>Enable 1</b> inactive</li> <li>4 - if <b>Top</b> active or <b>Enable 1</b> inactive</li> </ul>
<b>Enable 2</b>	<ul style="list-style-type: none"> <li>On: The <b>Enable 2</b> input is available</li> <li>Off: The <b>Enable 2</b> input is not available</li> </ul>
<b>Single-stroke press</b>	<ul style="list-style-type: none"> <li>On: Single stroke monitoring active, <b>Top</b> input available</li> <li>Off: Single stroke monitoring deactivated, <b>Top</b> input not available</li> </ul>
<b>Min. restart pulse time</b>	<ul style="list-style-type: none"> <li>100 ms</li> <li>350 ms</li> </ul>

Outputs		
Description	Type	Short description
<b>Enable</b>	Mandatory	Enable of the press process.
<b>Restart required</b>	Mandatory	The press must be reset because of an active restart interlock.
<b>Enable 1 inverted</b>	Mandatory	Output with inverted signal and the Enable 1 input.

## 7.10.2.1.1 General description

The press set-up function block is generally used together with the universal press contact monitor function block or the eccentric press contact monitor function block in order to set up the press. The **Top** output of the contact monitor FB is used as input for the Press set-up function block. If the **Press single stroke** parameter is active (single stroke monitoring active) then the **Enable** output is low once the press has reached the top dead center (rising edge at the **Top** input). Depending on the setting of the **Restart interlock** parameter it is possible to implement a step operation using the **On/Start** input. Depending on this setting a restart sequence is required after a press stop in order to reset or not reset the restart interlock. An activated restart interlock is signaled by a high at the **Restart required** output.

The module has an **Enable 1** input. If this is low then the **Enable** output is immediately low. The **Enable 2** input (if configured as active) is only required during the start sequence. As soon as the **Enable** output is high, **Enable 2** is no longer monitored. The **On/Start** input starts the press movement with a rising edge, in the event of a falling edge the press movement is stopped (**Enable** output is low). A restart sequence can be triggered with the **Restart** input; this sequence resets the restart interlock. The **Enable 1 inverted** output always shows the inverted state of the **Enable 1** input.



**ATTENTION**

**Make sure that the transitions of the signals for restart meet the requirements of the safety standards and regulations!**

When there is 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 as a result of short-circuit detection. If this type of pulse can lead to a hazard-inducing state in the machine, then the following points should be noted:

- Ensure that the cable routing is protected for the signal lines (due to cross-connection with other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

### Sequence/timing diagram

The diagram below shows a typical progression of the input and output states of the function block. The sequence shows three cycles of the press with the press operation interrupted twice by the falling edge at the **On/Start** input. A restart interlock was not activated.

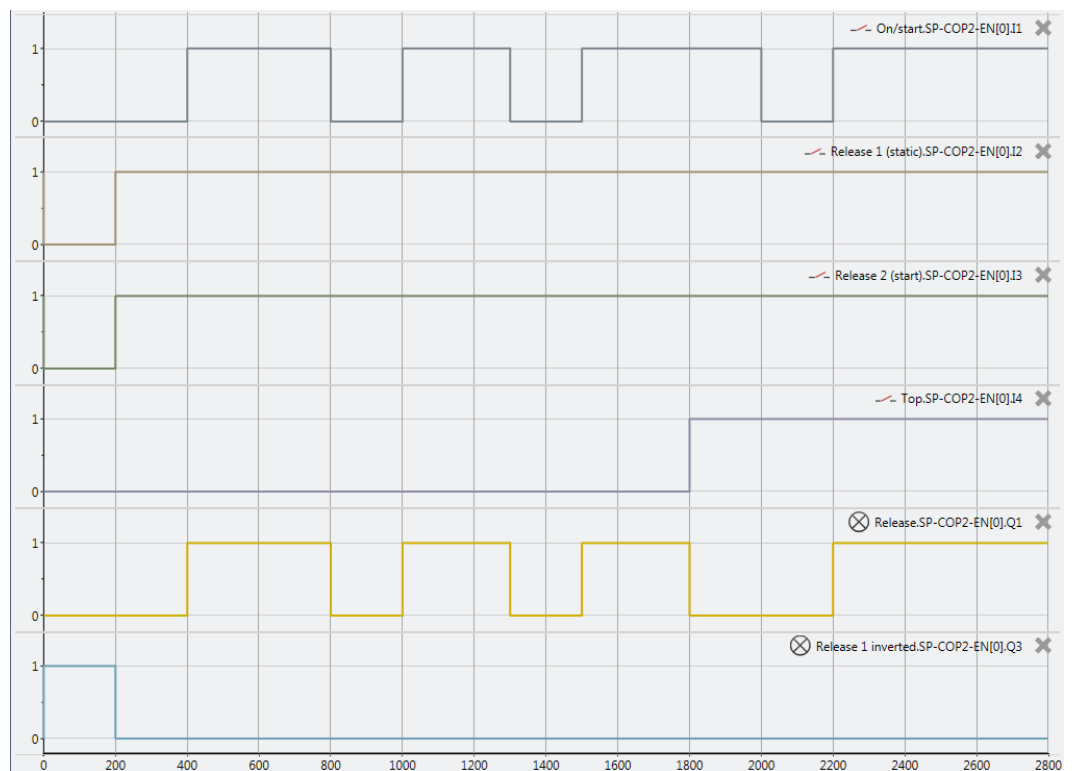


Illustration 160: Typical sequence/timing diagram for the press set-up function block

## 7.10.2.1.2 Parameters of the function block

Table 107: Parameters of the press set-up function block

Parameters	Possible values
<b>Restart interlock</b>	<ul style="list-style-type: none"> <li>• 1- without</li> <li>• 2 - always</li> <li>• 3 - if <b>On/Start</b> or <b>Enable 1</b> inactive</li> <li>• 4 - if <b>Top</b> active or <b>Enable 1</b> inactive</li> </ul>
<b>Enable 2</b>	<ul style="list-style-type: none"> <li>• On: The <b>Enable 2</b> input is available</li> <li>• Off: The <b>Enable 2</b> input is not available</li> </ul>
<b>Single-stroke press</b>	<ul style="list-style-type: none"> <li>• On: Single stroke monitoring active, <b>Top</b> input available</li> <li>• Off: Single stroke monitoring deactivated, <b>Top</b> input not available</li> </ul>
<b>Min. restart pulse time</b>	<ul style="list-style-type: none"> <li>• 100 ms</li> <li>• 350 ms</li> </ul>

### Restart interlock parameter

The **Restart interlock parameter** can be used to configure the behavior of the module after a stop. An active restart interlock (triggered by a previous stop or a restart of the module) is signaled by a high signal at the **Restart required** output. The **Restart required** output only shows the activated restart interlock if the preconditions for a valid restart sequence are given. These are: **Enable 1** high and, if configured as active, **Enable 2** also high.

In the case of configuration with **1 - without** no restart interlock is active (and the **Restart required** output is not available) and the press operation can be advanced without a valid restart sequence. In other words in the event of interruption of the press operation by a rising edge at the **Top** input (with active single stroke monitoring) the press operation can be continued by a rising edge at the **On/Start** input. In the event of interruption of the press operation by a falling edge at the **On/Start** input the press operation can be continued by a rising edge at the **On/Start** input.

In the case of configuration with **2 - always** the restart interlock is always active, i.e. the module starts with an activated restart interlock and each stop results in an active restart interlock that has to be reset by a valid restart sequence. Stops due to a falling edge at **On/Start**, a rising edge of **Top** or a falling edge of **Enable 1**.

In the case of configuration with **3 - if On/Start or Enable 1 inactive** the restart interlock is active, i.e. the module starts with an activated restart interlock and every stop due to a falling edge at **On/Start** or a falling edge at **Enable 1** results in an active restart interlock that has to be reset by a valid restart sequence.

In the case of configuration with **4 - if Top active or Enable 1 inactive** the restart interlock is active, i.e. the module starts with an activated restart interlock and every stop due to a rising edge at **Top** or a falling edge at **Enable 1** results in an active restart interlock that has to be reset by a valid restart sequence.

### Enable 2 parameter

This parameter activates the **Enable 2** input, if the parameter is active

### Press single stroke parameter

This parameter activates the single stroke monitoring, if the parameter is active. This means the **Top** input is available and a rising edge at the **Top** input ends the press operation (i.e. a complete single stroke has been fully performed).

### Min. restart pulse time parameter

A restart sequence (low-high-low **Restart** input) is only valid if the **Restart** input was set to high for at least 100 ms or 350 ms. This value is set with the **Min. restart pulse time** parameter.

## 7.10.2.1.3 Inputs of the press set-up function block

Table 108: Inputs of the press set-up function block

Description	Type	Short description
<b>On/Start</b>	Mandatory	Start a press operation with the rising edge or stop with the falling edge
<b>Enable 1 (static)</b>	Mandatory	Main enable of the press
<b>Enable 2 (start)</b>	Optional	Additional enable input is only evaluated when the <b>Enable</b> output is low.
<b>Top</b>	Optional	Top dead center for single stroke monitoring
<b>Restart</b>	Optional	Reset the state of the press

The press set-up function block supports the following input signals:

### On/Start input

The **On/Start** input is used to indicate the beginning and end of the press movement. A rising edge (low to high) at the **On/Start** input signals a start of the press. A falling edge at the **On/Start** input (**high to low**) signals a stop of the press. If the **Restart interlock** parameter is set to **2 - always** or **3 - if On/Start or Enable 1 inactive** then a valid **Restart** sequence is required after a stop caused by a low at the **On/Start** input.

### Enable 1 (static) input

The **Enable 1 (static)** input signal is mandatory. The **Enable** output is always immediately low when **Enable 1 (static)** is low.

If this function block is used together with a press contact function block (e.g. eccentric press contact monitor or universal press contact monitor), its **Enable** output is connected to the **Enable 1 (static)** input of the press set-up function block.

### Enable 2 (start)

The **Enable 2 (start)** input signal is optional. If **Enable 2 (start)** is configured, the **Enable** output can only be high (e.g. during switch on) if **Enable 2 (start)** is high. If the **Enable** output is high, **Enable 2 (start)** is no longer monitored.



ATTENTION

#### Do not use the Enable 2 (start) input for safety purposes.

Do not use the **Enable 2 (start)** input to initiate an emergency stop because this input is only evaluated temporarily during the start sequence. If not, you are placing the press operator in danger.

### Top input

The **Top** input signal is optional. It is used to determine the end of the press cycle (i.e. the press has reached the top reversal point). This signal is available at the eccentric press contact monitor or universal press contact monitor function blocks. The **Top** input signal is used for the single stroke monitoring. If the **single stroke monitoring** configuration parameter is set to **active**, the **Enable** output is low if the **Top** input changes from low to high.



ATTENTION

#### Do not use the Top input for safety purposes.

Only connect the **Top** input to a **Top** output of the universal press contact monitor or eccentric press contact monitor function blocks or to an equivalent signal source. Do not use the **Top** input to initiate an emergency stop. If not, you are placing the press operator in danger.

## Restart input

If the **Restart interlock parameter** is set to **1 - without**, no **restart** signal is required to restart the press after a stop. The **Restart interlock parameter** can be set to the following values:

1	without
2	always
3	if On/Start or Enable 1 inactive
4	if Top active or Enable 1 inactive

This parameter determines when a **restart** sequence is expected as input signal for the function block.

If the **Enable** output is low and, due to the aforementioned setting of the **Restart interlock parameter** a restart interlock is configured, the **Enable** output can only be reset after a valid **restart** sequence has been completed with a low-high-low transition (minimum 100 ms or 350 ms; shorter pulses and pulses over 30 s are ignored).

### 7.10.2.1.4 Outputs of the press set-up function block

Table 109: Outputs of the press set-up function block

Description	Type	Short description
<b>Enable</b>	Mandatory	Enable of the press process.
<b>Restart required</b>	Optional	The press must be reset because of an active restart interlock.
<b>Enable 1 inverted</b>	Mandatory	Output with inverted signal and the Enable 1 input.

## Release output

The **Enable** output is high if **Restart required** is low (i.e. no restart is required) and the following conditions are satisfied:

If the **Single stroke parameter** is set to **inactive**, **Enable 1 (static)** is high and **Enable 2 (start)** (if configured) is high; and a rising edge (low to high) is detected at the **On/Start** input. (In this case the **Enable** output is low if the **On/Start** input changes from high to low or the Enable 1 input is low); or

If the **Single stroke parameter** is set to **active**, **Enable 1 (static)** is high and **Enable 2 (start)** (if configured) is high; and a rising edge (low to high) is detected at the **On/Start** input. (In this case the **Enable** output is low if the **Top** input changes from low to high or the **On/Start** input changes from high to low or the **Enable 1** input is low)

## Restart required output

The **Restart required** output is high if a valid restart sequence is expected at the **Restart** input. This output is set back to low only after a valid restart sequence and the **Enable** output cannot be high while this output is high. A valid restart sequence is described in the **Restart interlock**, **Min. restart pulse time parameter** and **Restart input** sections.



## Enable 1 inverted output

The **Enable 1 inverted** output indicates whether an enable signal is pending at the press set-up function block. If the **Enable 1** input is high, the **Enable 1 inverted** output is low and vice versa.

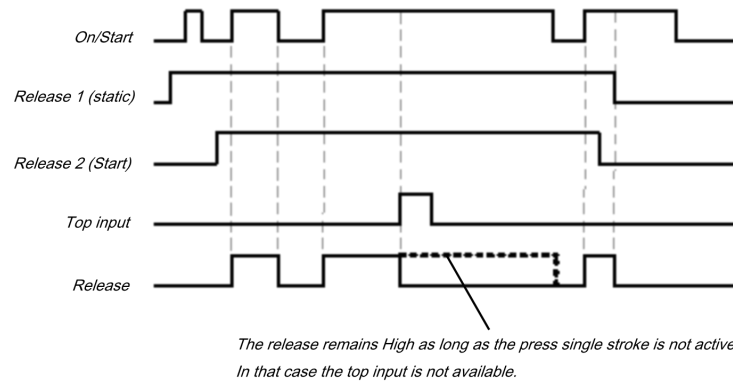


Illustration 161: Sequence/timing diagram for the press set-up function block

**NOTICE** If no immediate change from set-up to production operation is needed, a corresponding wait time should be programmed in the logic editor.

### 7.10.2.1.5 Error states and information regarding reset

Table 110: Error states and information regarding reset for the set-up function block

Outputs	Error state reset	Remarks
<b>Restart required</b>	The <b>Restart required</b> output is high if a valid restart sequence is expected at the <b>Restart</b> input. This output is set back to low only after a valid restart sequence and the <b>Enable</b> output cannot be high while this output is high. A valid restart sequence means a change in the restart input from low to high to low where the time of the high signal must be at least 100 or 350 ms (depending on the configuration of min. restart pulse time). Times less than the configured min. restart pulse time or larger than 30 seconds are ignored.	This occurs when the enable is switched off and a restart interlock is active depending on the configuration.

## 7.10.2.2 Single-stroke press

### Function block diagram

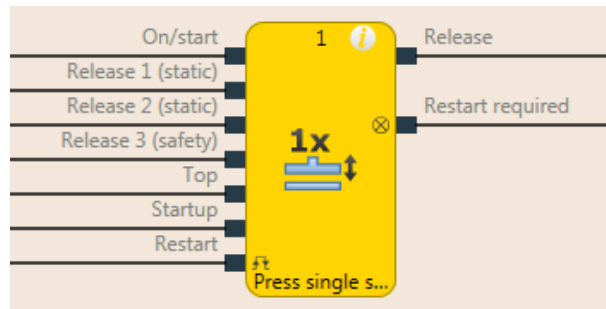


Illustration 162: Logical connections for the press single stroke function block

Inputs		
Description	Type	Short description
<b>On/Start</b>	Mandatory	Start a press operation with the rising edge or stop with the falling edge
<b>Enable 1 (static)</b>	Mandatory	Main enable of the press
<b>Enable 2 (start)</b>	Optional	Additional enable input
<b>Enable 3 (safety)</b>	Optional	Additional enable input
<b>Top</b>	Mandatory	Top dead center
<b>Ramp-up</b>	Optional	Contact for signaling of the press ramp-up
<b>Restart</b>	Optional	Reset the state of the press

Parameters	Possible values
<b>Restart interlock</b>	<ul style="list-style-type: none"> <li>1- without</li> <li>2 - always</li> <li>3 - if On/Start, Enable 1 or Enable 3 inactive</li> <li>4 - if Top active or Enable 1 or Enable 3 inactive</li> <li>5 - if Enable 1 or Enable 3 inactive</li> </ul>
<b>Enable 2</b>	<ul style="list-style-type: none"> <li>On: The <b>Enable 2</b> input is available</li> <li>Off: The <b>Enable 2</b> input is not available</li> </ul>
<b>Enable 3</b>	<ul style="list-style-type: none"> <li>On: The <b>Enable 3</b> input is available</li> <li>Off: The <b>Enable 3</b> input is not available</li> </ul>
<b>On/Start mode</b>	<ul style="list-style-type: none"> <li>1 - Inching</li> <li>2 - Single start</li> </ul>
<b>Mode for upstroke muting</b>	<ul style="list-style-type: none"> <li>1- without</li> <li>2 - always (only for Enable 3)</li> <li>3 - for Enable 3 and On/Start</li> </ul>
<b>Max. time for ramp-up muting</b>	1 to 7200 s.
<b>Min. restart pulse time</b>	<ul style="list-style-type: none"> <li>100 ms</li> <li>350 ms</li> </ul>
<b>Deselect restart interlock (for Enable 3) in TDC</b>	<ul style="list-style-type: none"> <li>On</li> <li>Off</li> </ul>

Outputs		
Description	Type	Short description
<b>Enable</b>	Mandatory	Enable of the press process.
<b>Restart required</b>	Optional	The press must be reset because of an active restart interlock.

## 7.10.2.2.1 General description

The press single stroke function block is generally used together with the universal press contact monitor function block or the eccentric press contact monitor function block in order to provide information of the **Top** and **Ramp-up these modules** outputs as input for this function block. The **Top** output is required for single-stroke operation. The press can be controlled for example using a two-hand control or by means of a clock mode function block in conjunction with a safety light curtain.

The single stroke monitoring is always active and cannot be configured. This means: If the rising edge is detected at the **Top** input then the **Enable** output is always low. The preconditions for a restart depend on the configuration of the **Restart interlock** parameter.

The **Enable 2** and **Enable 3**, **Ramp-up**, **Restart** inputs and the **Restart required** output are optional. These are available or not depending on the configuration.

The FB has

- a configurable restart interlock,
- the option to choose between inching mode and a complete single stroke cycle,
- a time configurable ramp-up muting and
- the configurable option during a regular stop at top dead center that falling edges of the Enable 3 input do not lead to a restart interlock.



**ATTENTION**

### Make sure that the transitions of the signals for restart meet the requirements of the safety standards and regulations!

When there is 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 as a result of short-circuit detection. If this type of pulse can lead to a hazard-inducing state in the machine, then the following points should be noted:

- Ensure that the cable routing is protected for the signal lines (due to cross-connection with other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

### Sequence/timing diagram

A typical sequence in minimum configuration (see the figure below) starts with a press cycle from the top dead center (**Top** input high). If then the **Enable 1** input is high and then a rising edge follows at the **On/Start** input, the Enable output is high. Now the press stroke begins and the top dead center is left (**Top** input is low). If subsequently in the press cycle the top dead center is reached again (rising edge at the **Top** input) then the **Enable** output is low again. A rising edge at the **On/Start** input would start this sequence from the beginning.

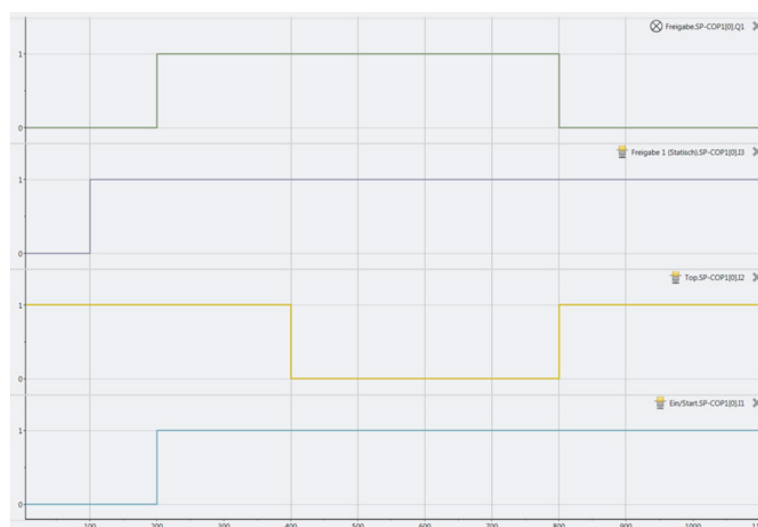


Illustration 163: Minimalist sequence press set-up module

## 7.10.2.2.2 Parameters of the function block

Table 111: Parameters of the press single stroke function block

Parameters	Possible values
<b>Restart interlock</b>	<ul style="list-style-type: none"> <li>• 1- without</li> <li>• 2 - always</li> <li>• 3 - if <b>On/Start</b>, <b>Enable 1</b> or <b>Enable 3</b> inactive</li> <li>• 4 - if <b>Top</b> active or <b>Enable 1</b> or <b>Enable 3</b> inactive</li> <li>• 5 - if <b>Enable 1</b> or <b>Enable 3</b> inactive</li> </ul>
<b>Enable 2</b>	<ul style="list-style-type: none"> <li>• On: The <b>Enable 2</b> input is available</li> <li>• Off: The <b>Enable 2</b> input is not available</li> </ul>
<b>Enable 3</b>	<ul style="list-style-type: none"> <li>• On: The <b>Enable 3</b> input is available</li> <li>• Off: The <b>Enable 3</b> input is not available</li> </ul>
<b>On/Start mode</b>	<ul style="list-style-type: none"> <li>• 1 - Inching</li> <li>• 2 - Single start</li> </ul>
<b>Mode for upstroke muting</b>	<ul style="list-style-type: none"> <li>• 1- without</li> <li>• 2 - always (only for <b>Enable 3</b>)</li> <li>• 3 - for <b>Enable 3</b> and <b>On/Start</b></li> </ul>
<b>Max. time for ramp-up muting</b>	<ul style="list-style-type: none"> <li>• 1 to 7200 s.</li> </ul>
<b>Min. restart pulse time</b>	<ul style="list-style-type: none"> <li>• 100 ms</li> <li>• 350 ms</li> </ul>
<b>Deselect restart interlock (for Enable 3) in TDC</b>	<ul style="list-style-type: none"> <li>• On</li> <li>• Off</li> </ul>

### Restart interlock parameter

The **Restart interlock** parameter can be used to configure the behavior of the module after a stop. An active restart interlock (triggered by a previous stop or a restart of the module) is signaled by a high signal at the **Restart required** output. The **Restart required** output only shows the activated restart interlock if the preconditions for a valid restart sequence are given. These are: **Enable 1** high and, if configured as active, **Enable 2** also high and **Enable 3** high. In the case of configuration of the restart interlock from 2 to 5 the following applies: If the **Restart interlock deselect** parameter is active, then a low signal from **Enable 3** in the top dead center (**Top** input is high) does not result in a restart interlock.

In the case of configuration with **1 - without** no restart interlock is active (and the **Restart required** output is not available) and the press operation can be advanced without a valid restart sequence. In other words in the event of interruption of the press operation by a rising edge at the **Top** input the press operation can be continued by a rising edge at the **On/Start** input. In the event of interruption of the press operation by a falling edge at the **On/Start** input the press operation can be continued by a rising edge at the **On/Start** input.

In the case of configuration with **2 - always** the restart interlock is always active, i.e. the module starts with an activated restart interlock and each stop results in an active restart interlock that has to be reset by a valid restart sequence. Stops due to a falling edge at **On/Start**, a rising edge of **Top**, a falling edge of **Enable 1** or a falling edge of **Enable 3**.

In the case of configuration with **3 - if On/Start, Enable 1 or Enable 3 inactive** the restart interlock is active, i.e. the module starts with an activated restart interlock and every stop due to a falling edge at **On/Start**, a falling edge at **Enable 1** or a falling edge at **Enable 3** results in a restart interlock that has to be reset by a valid restart sequence.

In the case of configuration with **4 - if Top active or Enable 1 or Enable 3 inactive** the restart interlock is active, i.e. the module starts with an activated restart interlock and every stop due to a rising edge at **Top** or a falling edge at **Enable 1** or **Enable 3** results in a restart interlock that has to be reset by a valid restart sequence.

In the case of configuration with **5 - if Enable 1 or Enable 3 inactive** the restart interlock is active, i.e. the module starts with an activated restart interlock and every stop due to a falling edge at **Enable 1** or **Enable 3** results in a restart interlock that has to be reset by a valid restart sequence.

### Enable 2 parameter

This parameter activates the **Enable 2** input, if the parameter is active.

### Enable 3 parameter

This parameter activates the **Enable 3** input, if the parameter is active.

### On/Start mode parameter

If the parameter is configured to **1 - Inching** then it is possible to stop the press cycle by a falling edge at the On/Start input and (depending on the configured restart interlock) to restart it with a rising edge. If the parameter is configured to **2 - Single start**, then the started press cycle cannot be stopped by a falling edge at the On/Start input.

### Mode for upstroke muting parameter

If the parameter is configured to **1 - without**, then the **Ramp-up** input is not available and ramp-up muting cannot be performed.

If the parameter is configured to **2 - for Enable 3**, then it is possible during the time defined under **Max. ramp-up muting** to let the **Enable 3** input become low without the restart interlock being activated.

If the parameter is configured to **3 - for Enable 3 and On/Start**, then after the rising edge at the ramp-up input and while this input remains high within the time set under **Max. ramp-up muting time**, it is possible to let the **Enable 3** input or the **On/Start** input become low without the restart interlock being activated.

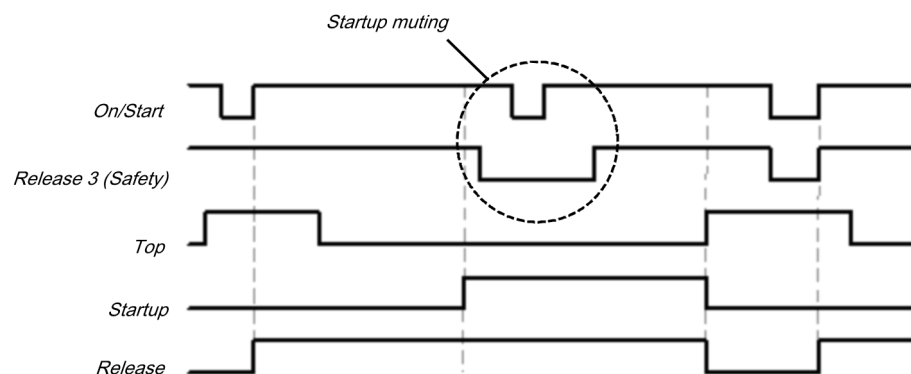


Illustration 164: Cycle/timing diagram for the press single stroke function block ramp-up muting of on/start and enable 3 (safety)



### Rule out any hazard during the upward stroke of the press!

If you use ramp-up muting, you must ensure that there are no hazards during ramp-up, e.g. due to the ramp-up movement itself.

### Max. ramp-up muting time parameter

The maximum time for bridging **Enable 3** input and **On/Start** input during the ramp-up phase can be configured (1-7200s). The time starts with the rising edge at the **Ramp-up** input. If the time expires before a falling edge is detected at the **Ramp-up** input, then the FB ends the muting of the configured **Enable 3** and **On/Start** inputs. If at this time or afterwards one of these inputs (dependent on the configuration) is or becomes low, then the Enable output is also low.

### **Min. restart pulse time parameter**

A restart sequence (low-high-low **Restart** input) is only valid if the **Restart** input was set to high for at least 100 ms or 350 ms. This value is set with the **Min. restart pulse time** parameter.

### **Deselect restart interlock (for Enable 3) in TDC parameter**

The activated parameter prevents the restart interlock being activated if the Enable 3 input is low during a regular stop at top dead center (Top input high).

## 7.10.2.2.3 Input signals of the single stroke function block

Description	Type	Short description
<b>On/Start</b>	Mandatory	Start a press operation with the rising edge or stop with the falling edge
<b>Enable 1 (static)</b>	Mandatory	Main enable of the press
<b>Enable 2 (start)</b>	Optional	Additional enable input
<b>Enable 3 (safety)</b>	Optional	Additional enable input
<b>Top</b>	Mandatory	Top dead center
<b>Ramp-up</b>	Optional	Contact for signaling of the press ramp-up
<b>Restart</b>	Optional	Reset the state of the press

### On/Start input

The **On/Start** input is used to indicate the beginning and end of the press movement. A rising edge (low to high) at the **On/Start** input signals a start of the press. A low at the **On/Start** input signals a stop of the press. If the **On/Start mode** parameter is set to **2 - Single start**, the press cannot be stopped by a low at the **On/Start** input.



**ATTENTION**

**Take additional precautions when the mode for the On/Start input is set to "2 - Single start".**

If the **On/Start mode** parameter is set to **2 - Single start**, you must take additional security measures (e.g. secure the hazard zone by a light curtain). If not, you are placing the press operator in danger.

If the **On/Start mode** parameter is set to **1 - Inching** and the **Condition for restart interlock** is set to **2 - always** or **3 - if On/Start, Enable 1 or Enable 3 inactive** then a valid **Restart** sequence is required after a stop caused by a low at the **On/Start** input.

The enable signal of a two-hand control or a function block for clock mode is particularly suitable for connection to the **On/Start** input.

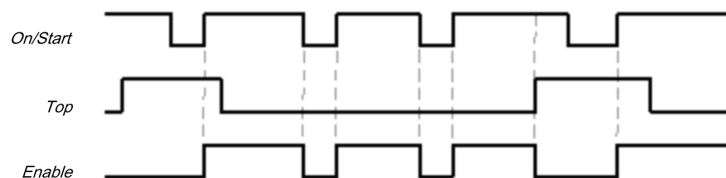


Illustration 165: Cycle/timing diagram for the press single stroke function block with configuration of on/start to 1 - inching

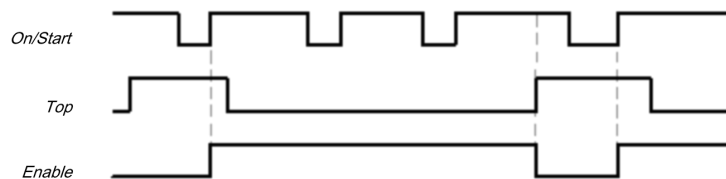


Illustration 166: Cycle/timing diagram for the press single stroke function block with configuration of on/start to 2 - single start

### Enable 1 (static) input

The **Enable 1 (static)** input signal is mandatory. The **Enable** output is always immediately low when **Enable 1 (static)** is low.

If this function block is used together with a press contact function block (e.g. eccentric press contact monitor or universal press contact monitor), its enable signal is connected to the **Enable 1 (static)** input of this function block.

### Enable 2 (start) input

The **Enable 2 (start)** input signal is optional. If **Enable 2 (start)** is configured, the **Enable** output can only be high (e.g. during switch on) if **Enable 2 (start)** is high. If the **Enable** output is high, **Enable 2 (start)** is no longer monitored.





### Do not use the Enable 2 (start) input for safety purposes.

Do not use the **Enable 2 (start)** input to initiate an emergency stop because this input is only evaluated temporarily during the start sequence. If not, you are placing the press operator in danger.

### Enable 3 (safety) input

The **Enable 3 (safety)** input signal is an optional signal. If **Enable 3 (safety)** is configured, the **Enable** output can only be high if **Enable 3 (safety)** is high. If **Enable 3 (safety)** is low and **Ramp-up muting is not active**, then the **Enable** output is set to low and a **restart** sequence must be done in accordance with the configuration.

If **Enable 1 (static)** and **Ramp-up** are high and the configured ramp-up time has not yet expired, then the **Enable 3 (safety)** signal is bypassed (ramp-up muting).

### Top input

The **Top** input signal is used to determine the end of the press cycle (i.e. the press has reached the top dead center). This signal is available at the eccentric press contact monitor or universal press contact monitor function blocks. The **Top** input signal is used for single stroke monitoring. The **Enable** output is low if the **Top** input passes from low to high.



### Do not use the Top input for safety purposes.

Only connect the **Top** input to a **Top** output of the universal press contact monitor or eccentric press contact monitor function blocks or to an equivalent signal source. Do not use the **Top** input to initiate an emergency stop. If not, you are placing the press operator in danger.

### Ramp-up input

If the **Ramp-up** input is available because of the configuration, then this must be connected.

### NOTICE

Only connect the **Ramp-up** input to the **Ramp-up** output of an eccentric press contact monitor or universal press contact monitor function block.

If the input is available, then the **Enable 3 (safety)** and **On/Start** input signals are bypassed (muting of the **On/Start** input depends on the parameter settings) if the **Enable** output is high and the **Ramp-up** input is high. This function block does not perform a plausibility check of the **Ramp-up** input signal. If the **Ramp-up** input is high several times during a single press cycle, then it is possible to bypass the corresponding input of the function block several times. If a signal should not be bypassed, then together with other signals that must be connected to the **Enable 1 (static)** input it should be connected to the **Enable 1 (static)** input by means of an AND function block.



### Rule out any hazard during the ramp-up of the press!

If you use ramp-up muting, you must ensure that there are no hazards during ramp-up, e.g. due to the ramp-up movement itself.

## Restart input

If the **Restart interlock** parameter has been set to **1 - without**, no **restart** signal is required to restart the press after a stop. The **Restart interlock** parameter can be set to the following values:

- 1 - without
- 2 - always
- 3 - if On/Start, Enable 1 or Enable 3 inactive
- 4 - if Top active or Enable 1 or Enable 3 inactive
- 5 - if Enable 1 or Enable 3 inactive

This parameter determines when a valid **restart** sequence is expected as input signal for the function block.

If the **Enable** output is low, the **Enable** output can only be reset in the case of settings 2 to 5 after a valid **restart** sequence has been completed at the **Restart** input with a low-high-low transition (minimum 100 ms or 350 ms; shorter pulses and pulses over 30 s are ignored).

### 7.10.2.2.4 Output signals of the function block

Description	Type	Short description
<b>Enable</b>	Mandatory	Enable of the press process.
<b>Restart required</b>	Optional	The press must be reset because of an active restart interlock.

## Release output

The **Enable** output is high if **Restart required** is low (i.e. no restart is required) and the following conditions are satisfied:

- The **Enable 1** input is high;
- if activated, Enable 2 must also be high;
- if activated, Enable 3 must also be high;
- and a rising edge at the On/Start input is detected;

## Restart required output

The **Restart required** output is high if a valid **restart** sequence is expected at the **Restart** input.

### 7.10.2.2.5 Error states and information regarding reset

Table 112: Error states and information regarding reset for the single stroke function block

Outputs	Error state reset	Remarks
<b>Restart required</b>	The <b>Restart required</b> output is high if a valid restart sequence is expected at the <b>Restart</b> input. This output is set back to low only after a valid restart sequence and the <b>Enable</b> output cannot be high while this output is high. A valid restart sequence means a change in the restart input from low to high to low where the time of the high signal must be at least 100 ms or 350 ms (depending on the configuration of min. restart pulse time). Times less than the configured min. restart pulse time or larger than 30 seconds are ignored.	This occurs when stopped depending on the configuration of the restart interlock.

## 7.10.2.3 Press automatic mode

### Function block diagram

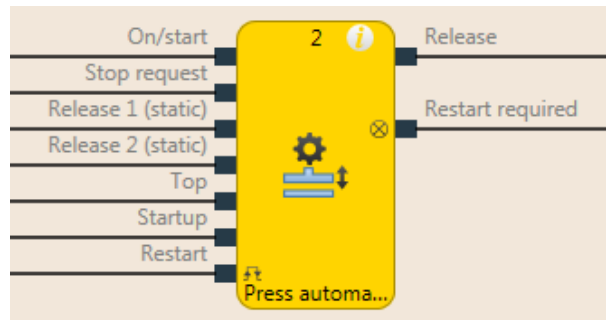


Illustration 167: Logical connections for the press auto mode function block

Inputs		
Description	Type	Short description
<b>On/Start</b>	Mandatory	Start a press operation (and trigger a stop request if Off/Stop is not available)
<b>Stop request</b>	Optional	Trigger a stop request if the input is available
<b>Enable 1 (static)</b>	Mandatory	Main enable of the press
<b>Enable 2 (start)</b>	Optional	Additional enable input
<b>Top</b>	Mandatory	Top dead center
<b>Ramp-up</b>	Optional	Press is in ramp-up (upstroke)
<b>Restart</b>	Optional	Reset the state of the press

Parameters	Possible values
<b>Restart interlock after stop condition</b>	On: After an error the RB must be reset with a valid restart sequence. Off: No restart required, <b>Restart</b> input not available.
<b>Stop request</b>	On: Stop request is triggered via the <b>Stop request</b> input Off: Stop request is triggered via a low signal at the On/Start input
<b>Ramp-up input</b>	On: The <b>Ramp-up</b> input is available Off: The <b>Ramp-up</b> input is not available
<b>Enable 2</b>	On: The <b>Enable 2</b> input is available Off: The <b>Enable 2</b> input is not available
<b>Min. restart pulse time</b>	For a valid reset: 100 ms: The <b>Restart</b> input must be at high for at least 100 ms. 350 ms: The <b>Restart</b> input must be at high for at least 350 ms.

Outputs		
Description	Type	Short description
<b>Enable</b>	Mandatory	Enable of the press process.
<b>Restart required</b>	Optional	The press must be reset because of an active restart interlock.

## 7.10.2.3.1 General description

The **press auto mode** function block (FB) is used in connection with press applications where the workpieces are moved automatically to and from the press, whereby access to the press is **required** occasionally, e.g. to change a die.

The function block can generate a stop signal for the press for this purpose at top dead center (**Top** input high) (i.e. the **Enable** output is low) if a stop has previously been requested. The request for a stop is triggered by a rising edge at the **Stop request** input or by a falling edge from **On/Start** (depending on the configuration of the stop request).

If the **Stop request** input is not configured, then a falling edge at the **On/Start** input leads to shutting down of the **Enable** output. If the **Run-up** input is active, then the shutting down of the **Enable** output is delayed until the **Top** input is active. If the **Ramp-up** input is or becomes inactive, the **Enable** output is shut down immediately. If the **Ramp-up** input is configured as active, then a rising edge at the **Stop request** input (or a falling edge at **On/Start**) causes an immediate change of the **Enable** output to low while the **Ramp-up** input is low. If the press is in ramp-up (**Ramp-up** input high) but not yet reached the top dead center (**Top** input low) and a rising edge is detected at **Stop request** (or a falling edge at **On/Start**), then the **Enable** output is only low if subsequently a rising edge is detected at the **Top** input or a falling edge is detected at the **Ramp-up** input.

If the **Ramp-up** input is not configured as active, then a falling edge at the **On/Start** input only leads to shut down of the **Enable** output if the top dead center has been reached (**Top** input high)

In addition the FB has its own **Restart** input that can be activated with the **Restart interlock after stop condition** parameter.



### Make sure that the transitions of the signals for restart meet the requirements of the safety standards and regulations!

When there is 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 as a result of short-circuit detection. If this type of pulse can lead to a hazard-inducing state in the machine, then the following points should be noted:

- Ensure that the cable routing is protected for the signal lines (due to cross-connection with other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

### Sequence/timing diagram

The diagram below shows a typical progression of the input and output states of the function block. The sequence shows five cycles of the press with the press operation interrupted twice by the **Stop request** input.

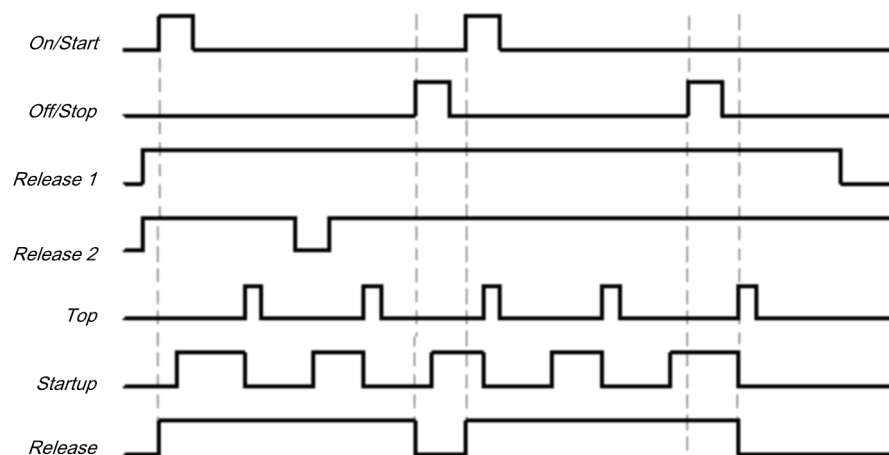


Illustration 168: Typical sequence/timing diagram for the press auto mode function block

### 7.10.2.3.2 Parameters of the function block

Table 113: Parameters of the press auto mode function block

Parameters	Possible values
<b>Restart interlock after stop condition</b>	<ul style="list-style-type: none"> <li>• On: After an error the RB must be reset with a valid restart sequence.</li> <li>• Off: No restart required, <b>Restart</b> input not available.</li> </ul>
<b>Stop request</b>	<ul style="list-style-type: none"> <li>• On: Stop request is triggered via the <b>Stop request</b> input</li> <li>• Off: Stop request is triggered via a low signal at the On/Start input</li> </ul>
<b>Ramp-up input</b>	<ul style="list-style-type: none"> <li>• On: The <b>Ramp-up</b> input is available</li> <li>• Off: The <b>Ramp-up</b> input is not available</li> </ul>
<b>Enable 2</b>	<ul style="list-style-type: none"> <li>• On: The <b>Enable 2</b> input is available</li> <li>• Off: The <b>Enable 2</b> input is not available</li> </ul>
<b>Min. restart pulse time</b>	For a valid reset: <ul style="list-style-type: none"> <li>• 100 ms: The <b>Restart</b> input must be at high for at least 100 ms.</li> <li>• 350 ms: The <b>Restart</b> input must be at high for at least 350 ms.</li> </ul>

#### Restart interlock after stop condition parameter

If the **Restart interlock after stop condition** parameter is not active, then no **restart** sequence is required to restart the press after a stop.

If the **Restart interlock after stop condition** parameter is active, the FB requests a valid restart sequence after the function block start or if the stop request was triggered. If this is the case, the **Enable** output is low and the **Restart required** output is in the high state. The Enable output cannot be set to high without a valid restart sequence. If during the stop request or during the first start of the FB, the inputs that are necessary for starting the FB are not in the expected state (**Enable 1** and **Enable 2** high) then the **Restart required** output is not set to high. This only changes to the high state after reaching the start states (**Enable 1** and **Enable 2** high). The **Restart required** output is only reset after a valid restart sequence has occurred with a low-high-low transition (at least 100 ms or 350 ms; shorter pulses and pulses over 30 s are ignored). Then the **Enable** output can be reset to the high state by a rising edge at the **On/Start** input.

### Stop request parameter

The **Stop request** parameter determines the stop mode of the press auto mode function block. If this parameter is not active, then the **On/Start** input (falling edge) is used to trigger a stop request. If the **Stop request** parameter is active then a rising edge at the **Stop request** input triggers a stop request.

In both cases the **Enable** output is only high if the following conditions are met:

A transition from low to high takes place at the **On/Start** input; and

the **Stop request** input is low, if used; and

there is no other reason that would normally trigger a stop signal; e.g. if **Enable 1 (static)** is low; and

- if the restart interlock after stop condition parameter is active, a valid restart sequence must occur beforehand.

If the **Ramp-up** input is configured as active, then a stop request causes an immediate change of the **Enable** output to low while the **Ramp-up** input is low. If the press is in ramp-up (**Ramp-up** input high) but not yet reached the top dead center (**Top** input low) and a stop request is detected, then the **Enable** output is only low if subsequently a rising edge is detected at the **Top** input or a falling edge is detected at the **Ramp-up** input.

If the **Ramp-up input** parameter is not active, then a stop request causes a change of the **Enable** output to low, immediately after a rising edge is detected at the **Top** input.



#### Do not use the On/Start and Stop request inputs for safety stops!

Regardless of the configuration of the **Condition for stop request**, the **On/Start** and **Stop request** inputs may not be used to initiate a safety stop. These inputs may only be used to trigger stop requests of the automation control system. Signals to initiate a safety stop (e.g. emergency stop) must be connected to the **Enable 1 (static)** input of the function block.

### Ramp-up input parameter

If the **Ramp-up input** parameter is active, a high signal at the **Ramp-up** input (the press moves upwards) makes it possible to stop the press both during the upwards movement and in the top dead center (**Top** input high) (the **Restart required** output is high, if available).

After the **Ramp-up** input has been set to high, a stop request (a rising edge at the **Stop request** input or a falling edge at **On/Start**, depending on configuration) can be triggered without the **Enable** output immediately switching to low.

If this parameter is not active, then regular stops are only possible at top dead center (**Top** input high).

#### NOTICE

Only connect the **Ramp-up** input to the **Ramp-up** output of a universal press contact monitor or eccentric press contact monitor function block.

### Enable 2 parameter

This parameter activates the **Enable 2** input, if the parameter is active

### Min. restart pulse time parameter

A restart sequence (low-high-low **Restart** input) is only valid if the **Restart** input was set to high for at least 100 ms or 350 ms. This value is set with the **Min. restart pulse time** parameter.

## 7.10.2.3.3 Input signals of the press auto mode function block

Table 114: Parameters of the press auto mode function block

Description	Type	Short description
<b>On/Start</b>	Mandatory	Start a press operation (and trigger a stop request if Off/Stop is not available)
<b>Stop request</b>	Optional	Trigger a stop request if the input is available
<b>Enable 1 (static)</b>	Mandatory	Main enable of the press
<b>Enable 2 (start)</b>	Optional	Additional enable input
<b>Top</b>	Mandatory	Top dead center
<b>Ramp-up</b>	Optional	Press is in ramp-up (upstroke)
<b>Restart</b>	Optional	Reset the state of the press

### On/Start input

The **On/Start** input signal is used to give a signal for the start of the press operation. If the **Stop request** parameter is not active, the falling edge of the **On/Start** input is evaluated as a stop request. If a rising edge (low to high) is detected at the **On/Start** input, then the **Enable** output is high, provided that the **Stop request** input is low and there is no other reason that would normally trigger a stop. These are:

- The Enable 1 input must be high,
- if Enable 2 is available, then this input must also be high,

if the **Stop request** input is available, then this input must be low.

Before the signal transition of **On/Start** from low to high a valid restart sequence is required if the **Restart interlock after stop condition** parameter is active. If you connect command devices (e.g. a two-hand controller) to the **On/Start** input, you must ensure that no unintended restart is possible.

### Off/Stop input

If the **Stop request** parameter is active, then the **Stop request** input is used to signal a stop request (rising edge at Off/Stop) to the press. If the **Ramp-up** input is configured as active, then a stop request causes an immediate change of the **Enable** output to low while the **Ramp-up** input is low. If the press is in ramp-up (**Ramp-up** input high) but not yet reached the top dead center (**Top** input low) and a stop request is detected, then the **Enable** output is only low if subsequently a rising edge is detected at the **Top** input or a falling edge is detected at the **Ramp-up** input. If the **Enable** output is low then in this case the **Restart required** output is set to high if the **Restart interlock after stop condition** parameter is active.

If the **Ramp-up input** parameter is not active, then a stop request causes a change of the **Enable** output to low, immediately after a rising edge is detected at the **Top** input.

This input can be used only if the Stop request parameter is active.

The **Stop request** input is not intended for the connection of non-safety related signals (e.g. from a programmable logic controller (PLC)). Safety-related signals may only be connected to the **Enable 1 (static)** input, not to the **Stop request** input.

### Enable 1 (static) input

The **Enable 1** input signal is mandatory. The **Enable** output is always immediately low when **Enable 1** is low.

If this function block is used together with a press contact function block (e.g. eccentric press contact monitor or universal press contact monitor), its **Enable** output must be connected to the **Enable 1** input of this function block.

### Enable 2 (start) input

The **Enable 2** input is optional. If the **Enable 2** parameter is active, the **Enable** output can only be high (e.g. during switch on) if the **Enable 2** input is high. As soon as the **Enable** output is high, the **Enable 2** input is no longer monitored.



#### **Do not use the Enable 2 (start) input for safety purposes.**

Do not use the **Enable 2 (start)** input to initiate an emergency stop because this input is only evaluated temporarily during the start sequence. If not, you are placing the press operator in danger.

### Top input

The **Top** input is used to determine the end of the press cycle (i.e. the press has reached the top dead center). If the **Restart interlock after stop condition** parameter is active and a stop request was present, then the **Restart required** output is high and the **Enable** output low due to the rising edge of the **Top** input.

If the **Restart interlock after stop condition** parameter is not active and a stop request was present, then only the **Enable** output is low due to the rising edge of the **Top** input. This signal is available at the eccentric press contact monitor or universal press contact monitor function blocks.



#### **Do not use the Top input for safety purposes.**

Do not use the **Top** input to initiate an emergency stop. If not, you are placing the press operator in danger.

### Ramp-up input

The Ramp-up input allows the press to stop both during the upwards movement and in the top dead center (**Top** input high). If the parameter is not active, then regular stops are only possible at top dead center.

#### **NOTICE**

#### **Do not use the Ramp-up input for safety purposes!**

Only connect the **Ramp-up** input to the **Ramp-up** output of a universal press contact monitor or eccentric press contact monitor function block.

### Restart input

If the **Restart interlock after stop condition** parameter is active, then the **Restart** input is available. In this case the **Restart required** output must be reset in the event of the first start of the module or after a successful stop (**Enable** output is low). Reset is done by a valid restart sequence. This means a change in the **Restart** input from low to high to low where the time of the high signal must be at least 100 ms or 350 ms (depending on the configuration of **Min. restart pulse time**). Times less than the configured **Min. restart pulse time** or larger than 30 seconds are ignored.



### 7.10.2.3.4 Outputs of the press auto mode function block

*Table 115: Parameters of the press auto mode function block*

Description	Type	Short description
<b>Enable</b>	Mandatory	Enable of the press operation
<b>Restart required</b>	Optional	The press must be reset because of an error.

#### Release output

The press operation can only be performed if the **Enable** output is high. The output is controlled depending on the parameters and input states of the block. If the **Enable** output is high, the **Restart required** output is always low. Both outputs can be low at the same time.

#### Restart required output

The **Restart required** output is high if a valid restart sequence is expected at the **Restart** input. This output is set back to low only after a valid restart sequence and the **Enable** output cannot be high while this output is high. A valid restart sequence is described in the **Restart interlock after stop condition parameter**, **Min. restart pulse time parameter** and **Restart input sections**.

### 7.10.2.3.5 Error states and information regarding reset

*Table 116: Error states and information regarding reset for the auto mode function block*

Outputs	Error state reset	Remarks
<b>Restart required</b>	The <b>Restart required</b> output is high if a valid restart sequence is expected at the <b>Restart</b> input. This output is set back to low only after a valid restart sequence and the <b>Enable</b> output cannot be high while this output is high. A valid restart sequence means a change in the <b>Restart required</b> input from low to high to low where the time of the high signal must be at least 100 ms or 350 ms (depending on the configuration of min. restart pulse time). Times less than the configured min. restart pulse time or larger than 30 seconds are ignored.	This occurs when stopped depending on the configuration of the restart interlock.

## 7.10.2.4 Clock mode

### Function block diagram

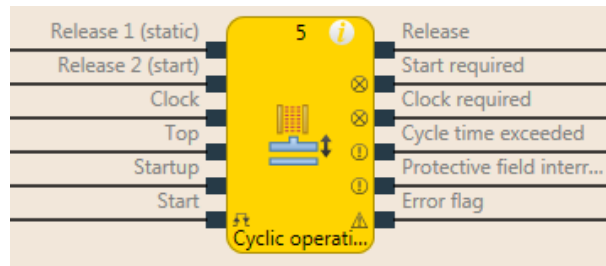


Illustration 169: Logical connections for the clock mode function block

Inputs		
Description	Type	Short description
Enable 1 (static)	Mandatory	Main enable of the press
Enable 2 (start)	Optional	Additional enable input
Clock	Mandatory	Clock input e.g. from a light barrier (contactless safety device)
Top	Mandatory	Top dead center
Ramp-up	Optional	Contact for signaling of the press ramp-up
Start	Optional	Reset the state of the press
Drive enable	Optional	Connection of a signal that controls the physical output of the press drive, e.g. the enable output from the FB single stroke or automatic.

Parameters	Possible values
Number of clocks	1 to 8 (to EN692 and EN693 max. 2)
Mode	<ul style="list-style-type: none"> <li>1 - Standard</li> <li>2 - Sweden</li> </ul>
Max. ramp-up muting time	0 = inactive, 1 to 7200 s. The Ramp-up input is only available if the value is not set to 0.
Clock time monitoring	0 = inactive, 1 to 500 s
Enable 2	<ul style="list-style-type: none"> <li>1- Without</li> <li>2 - Required for every start</li> <li>3 - Required for first start</li> </ul>
Start of the first clock	<ul style="list-style-type: none"> <li>1 - After reaching the TDC</li> <li>2 - After start of the ramp-up</li> </ul>
Restart interlock	<ul style="list-style-type: none"> <li>1- without</li> <li>2 - always</li> <li>3 - after shutdown in downstroke or initial position</li> </ul>
Min. restart pulse time	<ul style="list-style-type: none"> <li>100 ms</li> <li>350 ms</li> </ul>
Starting position	<ul style="list-style-type: none"> <li>1 - everywhere</li> <li>2 - only in TDC</li> </ul>
Min. clock pulse time	<ul style="list-style-type: none"> <li>100 ms</li> <li>350 ms</li> </ul>
Use error flag	<ul style="list-style-type: none"> <li>On: The error flag output is available</li> <li>Off: The error flag output is not available</li> </ul>

## Logic programming – Function blocks

Outputs		
Description	Type	Short description
<b>Enable</b>	Mandatory	Enable of the press process.
<b>Start required</b>	Optional	The press must be reset because in the event of an active restart interlock.
<b>Clock required</b>	Mandatory	One (or several) intervention(s) is (are) required to continue the press cycle.
<b>Clock timeout</b>	Mandatory	Exceeding the set clock time monitoring is signaled.
<b>Safety field intervention</b>	Mandatory	Signals an unauthorized intervention.
<b>Start not possible here</b>	Optional	A stop outside of the top dead center has been performed. The press must first be moved to top dead center.
<b>Error flag</b>	Optional	A clock timeout or an unexpected clock is present

## 7.10.2.4.1 General description

The clock mode function block is used for press applications with clock mode (PSDI = Press Sensing Device Initiation).



ATTENTION

### Ensure the safety rules for clock mode are satisfied!

The requirements for clock mode (PSDI) are described in local, regional, national and international standards. Always bring the clock mode applications in line with these standards and regulations as well as with your risk analysis and avoidance strategy.

If the contactless safety device (e.g. safety light curtain) is not used in an operating mode, the contactless safety device must be switched off in this mode so that it is clear that the contactless safety device is not currently active in safety mode.

If more than one contactless safety device (e.g. safety light curtain) is used in an application that uses the N-clock functions (PSDI), then only one contactless safety device may be used to meet the requirements for the N-clock mode (PSDI).

In accordance with EN 692 and EN 693 for press applications, the number of interventions is limited to 1 or 2. Other applications are subject to the applicable standards.



ATTENTION

### Prevent access to dangerous movement!

Press systems with a configuration that would allow a person to penetrate the safety zone of a contactless safety device, cross it and leave it, are not permitted for clock mode.

This function block defines a specific sequence of interventions that trigger a press cycle. **Interventions** are defined as the transition from high to low to high of the **Clock** input signal. In clock mode of a press there is an indirect manual triggering of a press cycle based on a predefined number of "interventions" into the contactless safety device. If the contactless safety device (e.g. safety light curtain) detects that the operator's work movements in connection with the loading or unloading of parts have ended and that the operator has removed all body parts from the safety zone of the contactless safety device, the press may trigger automatically.

The clock mode function block can be used in conjunction with the universal press contact monitor or press single stroke function blocks and an input for a safety light curtain. The **Enable** output of this function block controls for example the **On/Start** input of a press single stroke function block.

The clock mode function block checks whether the start sequence is valid and when the intervention counter or the function block have to be reset.

In the minimum configuration shown below the following sequence must be performed to switch the **Enable** output to high. The **Enable 1** input and the **Clock** input must be high. The **Clock required** output then signals with a high that an intervention is required. A subsequent intervention at the **Clock** input (high-low-high sequence) switches the **Enable** output to high.

### Complete start sequence

A complete start sequence is required to be able to reset errors or stops if restart interlock is active.

If the **Enable** output is low due to one of the following conditions, a complete start sequence may be required:

**Enable 1 (static)** is low

The **Intervention safety zone** output is high while the **Clock** input is low and there is no active ramp-up muting and no stop at the top dead center (Top input low).

- In the event of a clock timeout (Clock timeout output high)
- After switching on the controller

If the **Intervention safety field** output is high, the **Enable** output is low, the **Clock** input is also low and **Restart interlock** is configured to **1 - without**, then a restart is possible without a complete **restart** sequence. This can also apply during the ramp-up of the press if **Restart interlock** is configured to **3 - After shutdown in downstroke or initial position**.

## Cycle start sequence

A cycle start sequence is required in order to start the next press cycle in the event of a regular stop at top dead center. A cycle start sequence consists of the programmed number of clocks.

## Sequence/timing diagram

The following diagram shows a typical profile in mode **1 - Standard** with a programmed number of interventions of 2 and without restart interlock.

Intervention in this case means that the safety outputs of the contactless safety device are switched off by an intervention. A high-low-high transition at the Clock input (a consecutive falling and then rising edge) is evaluated as 1 intervention (1 clock).

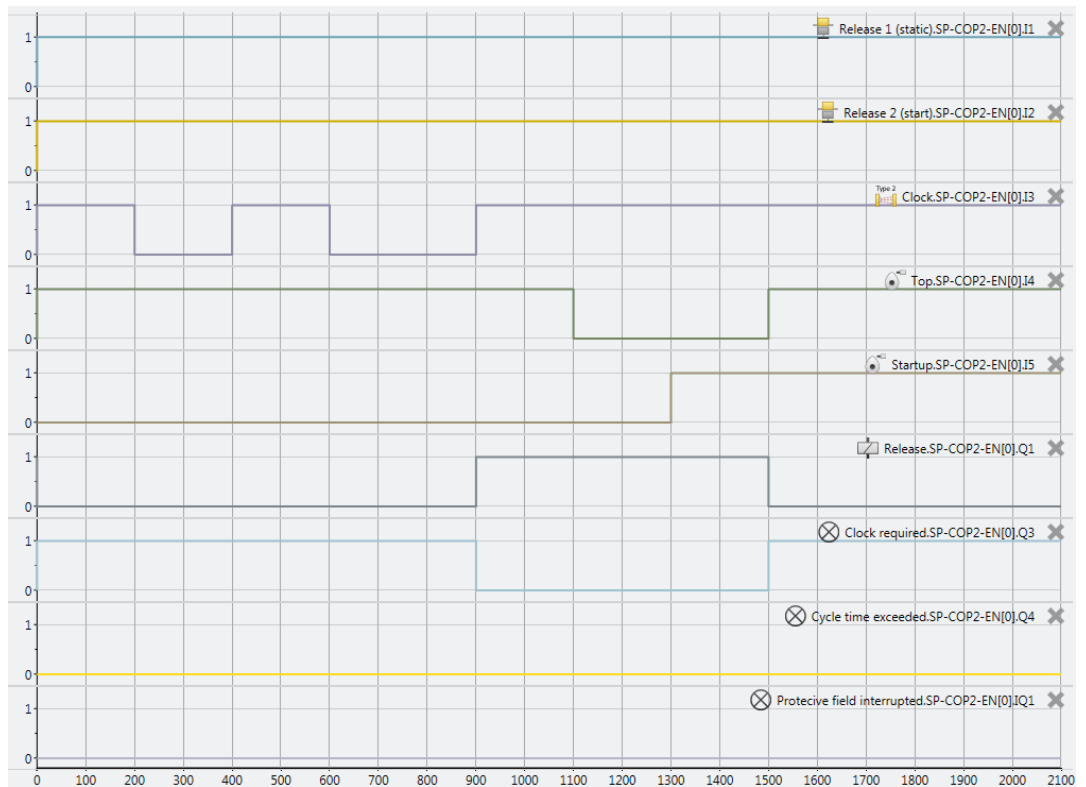


Illustration 170: Cycle/timing diagram for a complete sequence in standard mode with 2 programmed interventions without restart interlock.



**ATTENTION**

### Make sure that the transitions of the signals for restart meet the requirements!

When there is 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 as a result of short-circuit detection. If this type of pulse can lead to a hazard-inducing state in the machine, then the following points should be noted:

- Ensure that the cable routing is protected for the signal lines (due to cross-connection with other signal lines).
- No short-circuit detection, i.e. do not reference to test outputs.

### 7.10.2.4.2 Parameters of the clock mode function block

Table 117: Parameters of the clock mode function block

Parameters	Possible values
<b>Number of clocks</b>	<ul style="list-style-type: none"> <li>• 1 to 8 (to EN692 and EN693 max. 2)</li> </ul>
<b>Mode</b>	<ul style="list-style-type: none"> <li>• 1 - Standard</li> <li>• 2 - Sweden</li> </ul>
<b>Max. ramp-up muting time</b>	0 = inactive, 1 to 7200 s. The Ramp-up input is only available if the value is not set to 0.
<b>Clock time monitoring</b>	0 = inactive, 1 to 500 s
<b>Enable 2</b>	<ul style="list-style-type: none"> <li>• 1 - Without</li> <li>• 2 - Required for every start</li> <li>• 3 - Required for first start</li> </ul>
<b>Start of the first clock</b>	<ul style="list-style-type: none"> <li>• 1 - After reaching the TDC</li> <li>• 2 - After start of the ramp-up</li> </ul>
<b>Restart interlock</b>	<ul style="list-style-type: none"> <li>• 1 - without</li> <li>• 2 - always</li> <li>• 3 - after shutdown in downstroke or initial position</li> </ul>
<b>Min. restart pulse time</b>	<ul style="list-style-type: none"> <li>• 100 ms</li> <li>• 350 ms</li> </ul>
<b>Starting position</b>	<ul style="list-style-type: none"> <li>• 1 - everywhere</li> <li>• 2 - only in TDC</li> </ul>
<b>Min. clock pulse time</b>	<ul style="list-style-type: none"> <li>• 100 ms</li> <li>• 350 ms</li> </ul>
<b>Use error flag</b>	<ul style="list-style-type: none"> <li>• On: The error flag output is available</li> <li>• Off: The error flag output is not available</li> </ul>

#### Number of clocks parameter

This parameter is used to set after how many clocks (interventions) the press restarts a new cycle. The press only starts if the configured number of clocks was detected. If more clocks are detected than are configured, then the press stops immediately (**Enable** output low and **Intervention safety zone** output high) If the clock time monitoring is configured not equal to 0, then both a complete start sequence (depending on **Standard/Sweden** configuration first the clocks and then a valid restart sequence or vice versa) and a cycle start sequence (without restart sequence) occur with the configured time.

#### Mode parameter

The **Mode** parameter can be used to influence the order of a complete start sequence. A complete start sequence consists of the programmed number of clocks and a valid restart sequence.

In **Standard** mode first the configured number of clocks must occur, followed by a valid restart sequence (see illustration below).

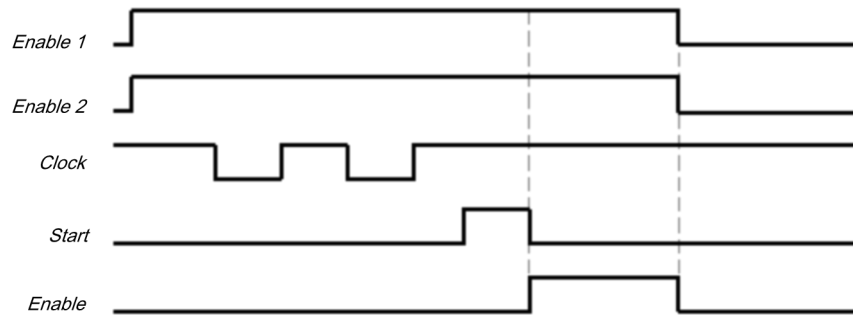


Illustration 171: Cycle/timing diagram for a complete start sequence in standard mode in two-clock mode

In **Sweden** mode first the valid restart sequence must occur followed by the configured number of clocks (see illustration below).

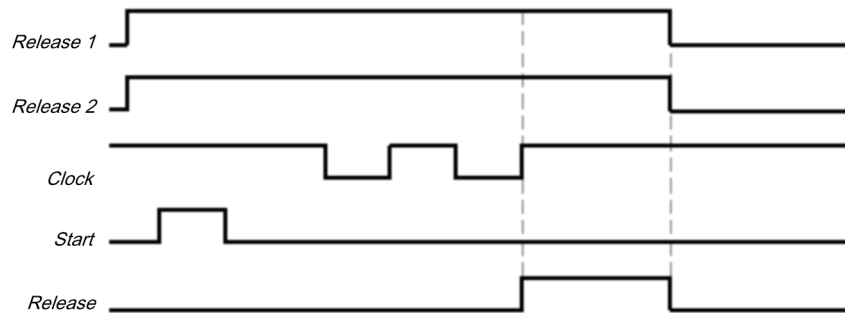


Illustration 172: Cycle/timing diagram for a complete start sequence in Sweden mode in two-clock mode

The minimum intervention time at the **Clock** input is 100 ms or 350 ms (see **Min. clock pulse time** parameter). Shorter interventions are not evaluated as valid, i.e. are ignored. If the parameter for the **Enable 2** input is configured as **3 - Required for first start** or as **2 - Required for every start**, the **Enable 2 (start)** input must also be high if a complete start sequence is required.

After the initial complete start sequence has been completed and the press has completed a press cycle, the **Top** input has to show that the press has reached the top dead center. This is indicated by a rising edge (low to high) of the **Top** input. If this happens, the internal intervention counter is reset. The **Enable** output is low and the **Clock required** output high.

In order to trigger a subsequent cycle, a cycle start sequence is required. In this case the **Enable** output is high if the configured number of interventions has occurred and the other configured conditions are met (e.g. the **Enable 2** parameter can be configured as **2 - Required for every start**).

### Max. ramp-up muting time parameter

Ramp-up muting allows bypassing of the **Clock** input (e.g. of the OSSD of a safety light curtain) during the press ramp-up. Ramp-up muting is activated if the **Max. ramp-up muting time** parameter is set to a value greater than 0. Ramp-up muting is inactivate if the **Max. ramp-up muting time** parameter is set to 0.

If ramp-up muting is activated...

it is mandatory that the **Ramp-up** input is connected to a suitable signal. This can be the **Ramp-up** output e.g. of the eccentric press contact monitor function block or the universal press contact monitor function block.

the **Clock** input of the function block is bypassed if the **Ramp-up** input is high and the **Top** input remains low.

The function block does not check the **Ramp-up** input for plausibility. This means that it is possible to bypass the **Clock** input several times if the **Ramp-up** input is activated several times during a single press cycle.

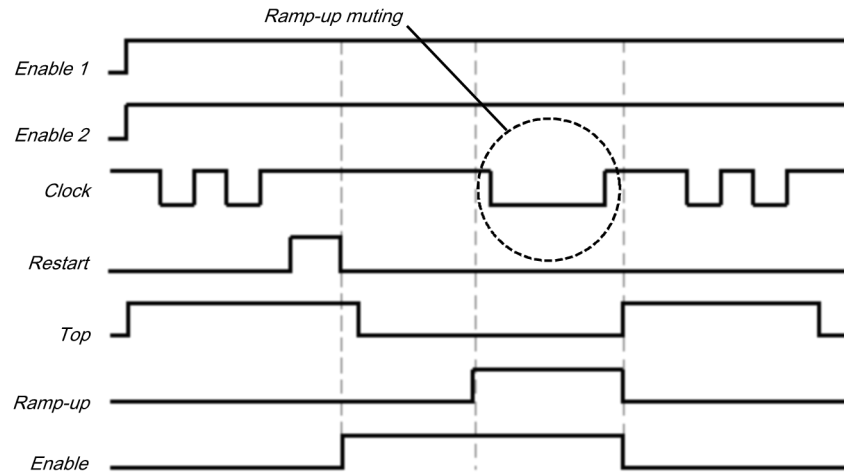


Illustration 173: Cycle/timing diagram for ramp-up muting in standard mode in two-clock mode

The **Max. time for ramp-up muting** can be configured. The timer for ramp-up muting starts when there is a rising edge (low to high) at the **Ramp-up** input. If the timer reaches the configured **Max. ramp-up muting time** or the **Ramp-up** input is low, the ramp-up muting ends and, if the **Clock** input is low, then the **Enable** output is set to low and the **Intervention safety zone** output is high. If a second rising edge at the **Ramp-up** input occurs and the **Enable** output is high, the ramp-muting begins anew.

### Clock time monitoring parameter

The **Clock time monitoring** parameter defines the necessary time both for a complete start sequence and for a cycle start sequence. If the **Clock time monitoring** is exceeded, the **Clock timeout** output is high. In this case a complete start sequence is required so that the **Enable** output can be high again (e.g. to start the press). The clock time timer starts when the press is stopped at top dead center (i.e. the **TDC** input changes from low to high). After all other stops, the **Clock time monitoring** starts in standard mode with the rising edge of the first valid clock pulse (rising edge at the **Clock** input) and in Sweden mode at the end of the restart sequence (falling edge at the **Start** input).

The basic setting for **Clock time monitoring** is 30 s in accordance with the maximum permitted clock time for eccentric presses (defined in EN 692). If **Clock time monitoring** is set to 0, the clock time monitoring is inactive.

### Enable 2 parameter

The use of the **Enable 2** input can be configured with this parameter. With the setting **1 - Without** the input is not available.

With the setting **2 - Required for every start** the input must be high for a complete start sequence or a cycle start sequence.

With the setting **3 - Required for first start** the input only needs to be high for a complete start sequence.

### Start of the first clock parameter

The **Start of the first clock** parameter determines from what point of the press cycle an intervention is deemed as valid.

If the **Start of the first clock** parameter is set to **2 - After start of the ramp-up**, then an intervention is valid if the start of the intervention (i.e. falling edge (high to low) at the **Clock** input) occurs after the rising edge at the **Ramp-up** input. It is immaterial whether the **Top** input has already moved to high.

If the **Start of the first clock** parameter is set to **1 - After reaching TDC**, then an intervention is only valid if the start of the intervention (i.e. falling edge (high to low) at the **Clock** input) first occurs after the rising edge at the **Top** input.

In both cases the end of the intervention (i.e. rising edge (low to high) at the **Clock** input) must occur after the rising edge at the **Top** input. It is immaterial whether the **Top** input is still high or has already moved back to low.



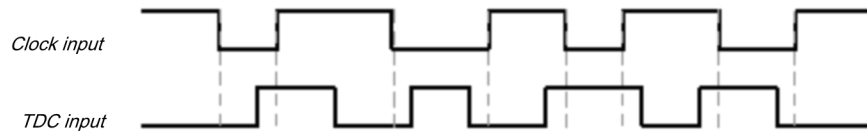


Illustration 174: Valid interventions if the Start of the first clock parameter is set to 2 - After start of the ramp-up.

**NOTICE** If the Start of the first clock parameter is set to **2 - After start of the ramp-up**, then ramp-up muting must be activated. Otherwise the Enable output moves to low as soon as the the Clock input moves to low (i.e. at the beginning of the intervention).

### Restart interlock parameter

Using the restart interlock parameter it is possible to configure the reset behavior of the FB after a stop. If the restart interlock is configured to **1 - without**, then no valid restart sequence is required if the module stops. The **Start** input and the **Start required** output are not available in this case.

If the restart interlock was triggered by a stop (**Enable** output is low) that is not part of the configured press cycle, then the restart interlock must be reset by a complete start sequence. The requirement of a complete start sequence is signaled, depending on the standard or Sweden mode, by a high at the **Start required** output (Sweden) or **Clock required** (standard). If the first part of the start sequence has been performed (valid restart sequence for Sweden or configured number of clocks for standard) then the necessary second part of the start sequence is signaled by a high at the **Clock required** output (Sweden) or **Start required** (standard). The Enable output is only high again after the start sequence for standard or Sweden has been completed in full.

If the restart interlock is configured to **2 - always** and the **Max. ramp-up muting time** is set to 0, then an intervention (**Clock** input is low) results in the **Enable** output being low, the **Intervention safety zone** output high and an activated restart interlock. A complete start sequence is required.

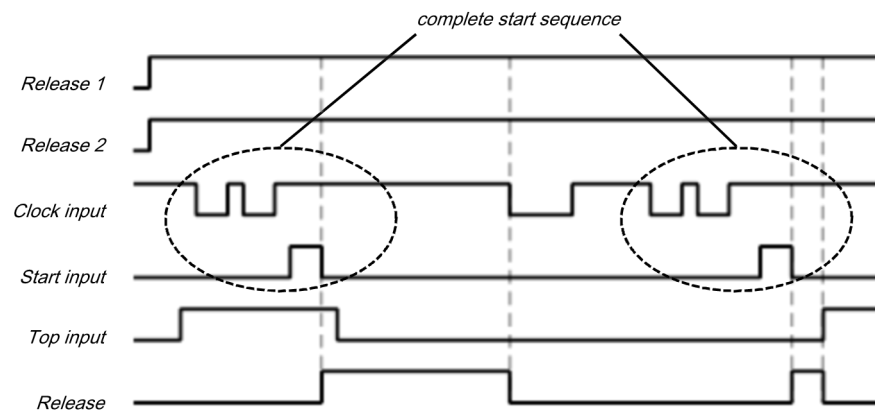


Illustration 175: Cycle/timing diagram if the Clock input is low, ramp-up muting inactive and restart interlock set to **2 - always**.

If the restart interlock is configured to **2 - always** and the **Max. ramp-up muting time** is set as not equal to 0, then an intervention with the **Ramp-up** input low results in the **Enable** output being low, the **Intervention safety zone** output high and an activated restart interlock. A complete start sequence is required.

If the restart interlock is configured to **2 - always** and the **Max. ramp-up muting time** is set as not equal to 0 and the **Start of the first clock** parameter is configured to **2 - After start of the ramp-up**, then an intervention with **Ramp-up** input high (the intervention may not be ended) does not result in an activated restart interlock due to the ramp-up muting and the **Start of the first clock** parameter setting. If the intervention is ended after reaching the top dead center (**Top** input is high) the intervention is counted as a valid intervention and then the subsequent interventions must be performed as configured in order to trigger a new cycle start.

If the restart interlock is configured to **2 - always** and the **Max. ramp-up muting time** is set as not equal to 0 and the **Start of the first clock** parameter is configured to **2 - After start of the ramp-up**, then an intervention with Ramp-up input high does not immediately result in an activated restart interlock due to the ramp-up muting and the **Start of the first clock** parameter setting. If however, the intervention is ended (**Clock** input is high again) before reaching the

top dead center (**Top** input is high) then when top dead center is reached the **Intervention safety zone** output is high, the **Enable** output low and the restart interlock is activated. A complete start sequence is required.

If the restart interlock is configured to **2 - always** and the **Max. ramp-up muting time** is set as not equal to 0 and the **Start of the first clock** parameter is configured to **1 - After reaching TDC**, then an intervention with **Ramp-up** input high (regardless of whether the intervention remains or is ended) does not immediately result in an activated restart interlock due to the ramp-up muting. When the top dead center is reached (**Top** input is high) the **Intervention safety zone** output is high, the **Enable** output low and the restart interlock is activated. A complete start sequence is required.

If the restart interlock is configured to **3 - After shutdown in downstroke or initial position** and an intervention occurs in the downstroke (**Top** input and **Ramp-up** input low) or in the top dead center (**Top** input high) after the configured number of interventions, then the **Intervention safety zone** output is high, the **Enable** output low and the restart interlock is activated. A complete start sequence is required.

If the restart interlock is configured to **3 - After shutdown in downstroke or initial position** and the **Ramp-up** input is high, then the **Enable** output remains high even if several interventions are made as long as ramp-up muting is running. If the **Clock** input is low, if the ramp-up muting time elapses, then the **Enable** output is low and the **Intervention safety zone** output is high. However, an end of the intervention sufficient to switch the **Enable** output back to high even if the top dead center was reached in the meantime. No restart interlock is activated or cycle start required.

If the restart interlock is configured to **3 - After shutdown in downstroke or initial position** and the **Ramp-up** input is high, then the **Enable** output remains high even if several interventions are made as long as ramp-up muting is running. If the **Clock** input is low once, if the ramp-up muting time has elapsed, then the **Enable** output is low and the **Intervention safety zone** output is high. However, an end of the intervention sufficient to switch the **Enable** output back to high even if the top dead center was reached in the meantime. No restart interlock is activated or cycle start required.

If the restart interlock is configured to **3 - After shutdown in downstroke or initial position** and the **Ramp-up** input is high, then the **Enable** output remains high even if several interventions are made as long as ramp-up muting is running. If the **Clock** input is high when the ramp-up muting time has elapsed, then the **Enable** output is actuated in the same way by a signal change at **Clock** from high to low to high. (If an intervention is performed again after completion of this signal change, the **Intervention safety zone** output is high, the **Enable** output low and the restart interlock is activated). If then the top dead center is reached (**Top** input is high), then the **Enable** output is low and a cycle start sequence is required (configured number of interventions), see also figure below.

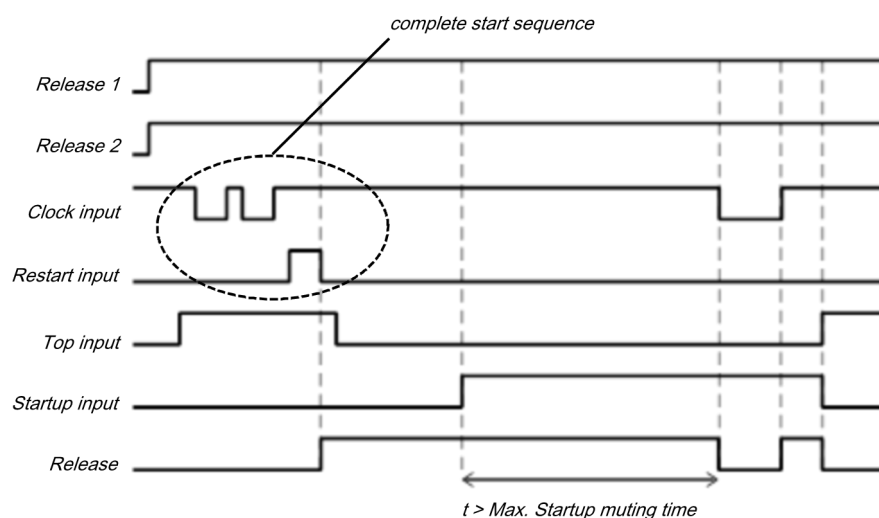


Illustration 176: Cycle/timing diagram if the Clock input is low, maximum ramp-up muting time > 0 and restart interlock set to 3 - After shutdown in downstroke or initial position.

### Min. restart pulse time parameter

A restart sequence (low-high-low **Restart** input) is only valid if the **Restart** input was set to high for at least 100 ms or 350 ms. This value is set with the **Min. restart pulse time** parameter.

### Start position parameter

If the **Start position** parameter is set to **2 - only at TDC**, a restart of the press is only possible at the top reverse point. In any other position a restart is prevented. If the press e.g. during the downward movement was stopped by an intervention into the safety zone of the light curtain, you must switch to another operating mode (e.g. in connection with the press set-up function block) in order to move the press back into the position at the top reverse point, because the clock mode function block uses this parameter setting to prevent a restart.

If the **Start position** parameter is set to **2 - only at TDC**, the optional **Drive enable** input must be connected in order to determine whether the press is running or has been stopped. This must be the same signal that controls the press directly. Usually the **Drive enable** input is connected to the output signal of the logic editor by a CPU flag and this output signal is connected to the physical output for the press.

**NOTICE** Do not connect any physical input signals to the **Enable drive** input. Use a CPU flag to connect the signal that controls the physical output for the press drive.

If the **Enable** output is low, because either the **Enable 1 (static)** input or the **Clock** input are low, then the **Start not possible here** diagnostic output is high. A restart of the press is then prevented until the **Top** input is high again and now new start in another operating mode has occurred.

### Min. clock pulse time parameter

A clock sequence (high-low-high **Clock** input) is only valid if the **Clock** input was set to low for at least 100 ms or 350 ms. This value is set with the **Min. clock pulse time** parameter.

## 7.10.2.4.3 Input signals of the clock mode function block

Description	Type	Short description
<b>Enable 1 (static)</b>	Mandatory	Main enable of the press
<b>Enable 2 (start)</b>	Optional	Additional enable input
<b>Clock</b>	Mandatory	Clock input e.g. from a light barrier (contactless safety device)
<b>Top</b>	Mandatory	Top dead center
<b>Ramp-up</b>	Optional	Contact for signaling of the press ramp-up
<b>Start</b>	Optional	Reset the state of the press
<b>Drive enable</b>	Optional	Connection of a signal that controls the physical output of the press drive, e.g. the enable output from the FB single stroke or automatic.

### Enable 1 (static) input

The **Enable 1 (static)** input signal is mandatory. The **Enable** output is always immediately low when **Enable 1 (static)** is low.

If this function block is used together with a press contact function block (e.g. eccentric press contact monitor or universal press contact monitor), its **Enable** output must be connected to the **Enable 1 (static)** input of this function block.

### Enable 2 (start) input

The **Enable 2 (start)** input signal is optional. If **Enable 2 (start)** is configured, the **Enable** output can only be high (e.g. during switch-on) if **Enable 2 (start)** is high (this only applies for complete start sequences when the **Enable 2** parameter is set to **3 - Required for first start**). If the **Enable** output is high, **Enable 2 (start)** is no longer monitored.



#### Do not use the Enable 2 (start) input for safety purposes.

Do not use the Enable 2 (start) input to initiate an emergency stop because this input is only evaluated temporarily during the start sequence. If not, you are placing the press operator in danger.

### Clock input

At the Clock input a contactless safety device is connected e.g. the output of a safety light curtain. Every completed intervention causes a high-low-high transition at the **Clock** input. The **Clock** input remains low for as long as the intervention lasts.

### Top input

The **Top** input signal is used to determine the end of the press cycle (i.e. the press has reached the top dead center). This signal is available at the eccentric press contact monitor or universal press contact monitor function blocks.



#### Do not use the Top input for safety purposes.

Only connect the Top input to a Top output of the universal press contact monitor or eccentric press contact monitor function blocks or to an equivalent signal source. Do not use the Top input to initiate an emergency stop. If not, you are placing the press operator in danger.

## Ramp-up input

The ramp-up input is optional and only available if **Max. ramp-up muting time** is greater than 0. If ramp-up muting is active (i.e. if the **Max. ramp-up muting time** is greater than 0) then the **Clock** input of the function block is bypassed when the **Ramp-up** input is high, the **Top** input remains low and the **Max. ramp-up muting time** has not yet elapsed. If the restart interlock is configured to **1 - always** then unauthorized interventions during ramp-up muting, after reaching top dead center, result in an activated restart interlock (**Intervention safety zone** output is high, **Enable** low). A complete start sequence is then necessary.

## Start input

If the **Restart interlock parameter** is set to **1 - without**, no **Start** signal is required to restart the press after a stop. The **Restart interlock parameter** can be set to the following values:

1	without
2	always
3	after shutdown in downstroke or initial position

This parameter determines when a **start** sequence is expected as input signal for the function block.

If the **Start required** output is high due to an activated restart interlock during a required complete **start** sequence, then this is only rest after a valid start sequence has been completed at the **Start** input with a low-high-low transition (minimum 100 ms or 350 ms; shorter pulses and pulses over 30 s are ignored).

## Drive enabled input

If the **Start position** parameter is set to **2 - only at TDC**, the optional **Drive enable** input must be connected in order to determine whether the press is running or has been stopped. This must be the same signal that controls the press directly. Usually the **Drive enable** input is connected to the output signal of the logic editor by a CPU flag and this output signal is connected to the physical output for the press.

**NOTICE** Do not connect any physical input signals to the **Enable drive** input. Use a CPU flag to connect the signal that controls the physical output for the press drive.

### 7.10.2.4.4 Output signals of the clock mode function block

Description	Type	Short description
<b>Enable</b>	Mandatory	Enable of the press process.
<b>Start required</b>	Optional	The press must be reset because in the event of an active restart interlock.
<b>Clock required</b>	Mandatory	One (or several) intervention(s) is (are) required to continue the press cycle.
<b>Clock timeout</b>	Mandatory	Exceeding the set clock time monitoring is signaled.
<b>Safety field intervention</b>	Mandatory	Signals an unauthorized intervention.
<b>Start not possible here</b>	Optional	A stop outside of the top dead center has been performed. The press must first be moved to top dead center.
<b>Error flag</b>	Optional	A clock timeout or an unexpected clock is present

#### Release output

The press operation can only be performed if the **Enable** output is high. The output is controlled depending on the parameters and input states of the block. If the **Enable** output is high, the **Start required** output is always low. Both outputs can be low at the same time.

#### Start required output

The **Start required** output is high if a valid start sequence is expected at the **Start** input. This output is set back to low only after a valid start sequence and the **Enable** output cannot be high while this output is high. A valid start sequence is described in the **Restart interlock parameter**, **Min. restart pulse time parameter** and **Start input** sections.

#### Clock required output

The **Clock required** output is high if an intervention is expected at the **Clock** input.

#### Clock timeout output

The **Clock timeout** output is high when a complete start sequence or a cycle start sequence has lasted longer than the set clock time monitoring. A complete start sequence is then necessary.

#### Intervention safety zone output

The **Intervention safety zone** output is high if an intervention has occurred at an unauthorized time. Depending on the configuration of the restart interlock and the ramp-up muting, only a rising edge at the **Clock** input, the configured number of clocks or a complete start sequence is required to acknowledge this error. Details can be found in the **Restart interlock parameters** chapter.

#### Start not possible here output

If the **Enable** output is low, because either the **Enable 1 (static)** input or the **Clock** input are low, then the **Start not possible here** output is high. A restart of the press is then prevented until the **Top** input is high again and now new start in another operating mode has occurred.

#### Error flag output

This output is set to high if at least one of the **Clock timeout** or **Intervention safety zone** outputs is set to high and the output is configured as active.

### 7.10.2.4.5 Error states and information regarding reset

*Table 118: Error states and information regarding reset for the clock mode function block*

Outputs	Error state reset	Remarks
<b>Safety field intervention</b>	The <b>Intervention safety zone</b> output is high if an intervention has occurred at an unauthorized time. Depending on the configuration of the restart interlock and the ramp-up muting, only a rising edge at the <b>Clock</b> input, the configured number of clocks or a complete start sequence is required to acknowledge this error. Details can be found in the <b>Restart interlock parameters</b> chapter.	The <b>Enable</b> output goes to low and the <b>Error flag</b> goes to high when <b>Intervention safety zone</b> or <b>Clock timeout</b> is high.
<b>Clock time-out</b>	In the event of a clock timeout the error is reset by a complete start sequence.	

## 7.11 User-defined function blocks

### 7.11.1 Grouped function block

#### Function block diagram

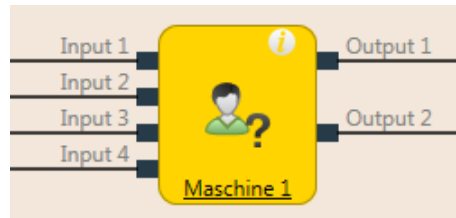


Illustration 177: Logical connections for grouped function block

You can select groups of function blocks in order to convert them into a single grouped function block. This is useful for simplifying repeated use of logic groups and for reducing the number of function blocks on one page.

A grouped function block has the following properties:

- It can have a maximum of 8 inputs and 8 outputs.
- It cannot contain the fast shut-off function block or a different grouped or user-defined function block.
- The icon for a grouped function block can be selected from a permanent library in samosPLAN5+.
- It is created in the logic editor but is not displayed in the list of function blocks.
- It is stored with the project. If the project is opened on a different PC, the grouped function block is displayed.
- It can be stored as a user-defined function block.

**NOTICE** When calculating the total number of function blocks in a project, a grouped function block is not counted as one block but instead with the number of function blocks used within it.

#### This is how you create a grouped function block:

- ➔ Select the function blocks you wish to group.
- ➔ Activate the context menu by right clicking on one of the selected function blocks.

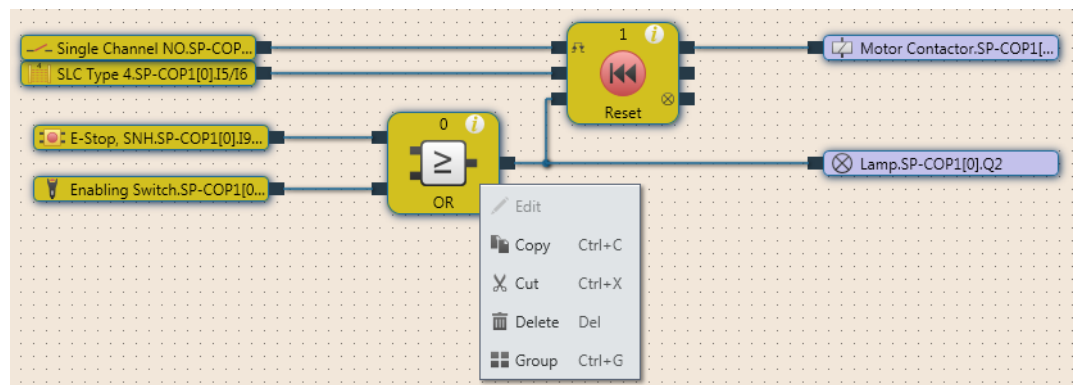


Illustration 178: Creating a grouped function block



## Logic programming – Function blocks

➔ Click on **Group...** The **Grouping** dialog will open.

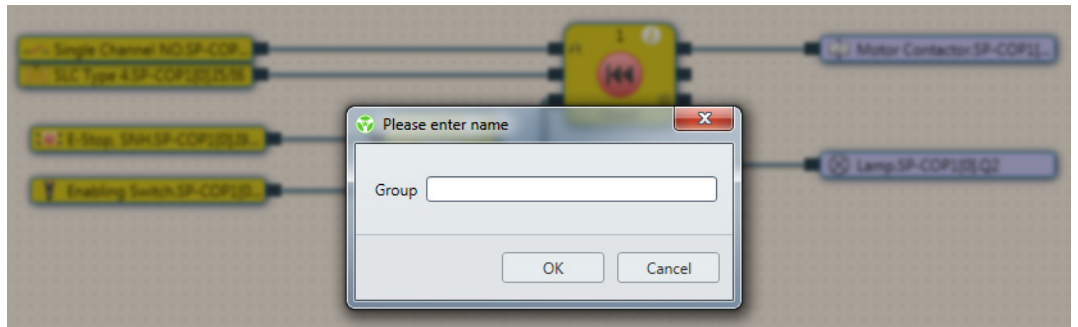


Illustration 179: Dialog for editing function block details for the grouped function block

- ➔ Enter a name for the new grouped function block.
- ➔ Within the dialog, click on **OK** to confirm your changes and close the dialog. The selected function blocks will be reduced to a single grouped function block on the worksheet for the main program.

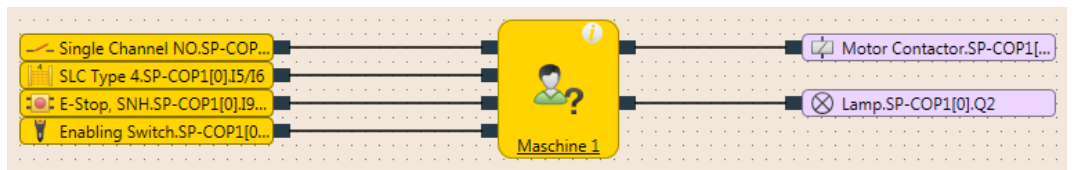


Illustration 180: New grouped function block in the work area

### NOTICE

The name of the grouped function block can be edited here by clicking on the icon of the function block and then **Edit**.  
When you click on the corresponding tab (in this case: **Machine 1**), you can edit the grouped function block.

The content of the new grouped function block will be saved on a new page. In the example, the name of the new grouped function block is **Machine 1**. The work area of the grouped function block is displayed in orange.

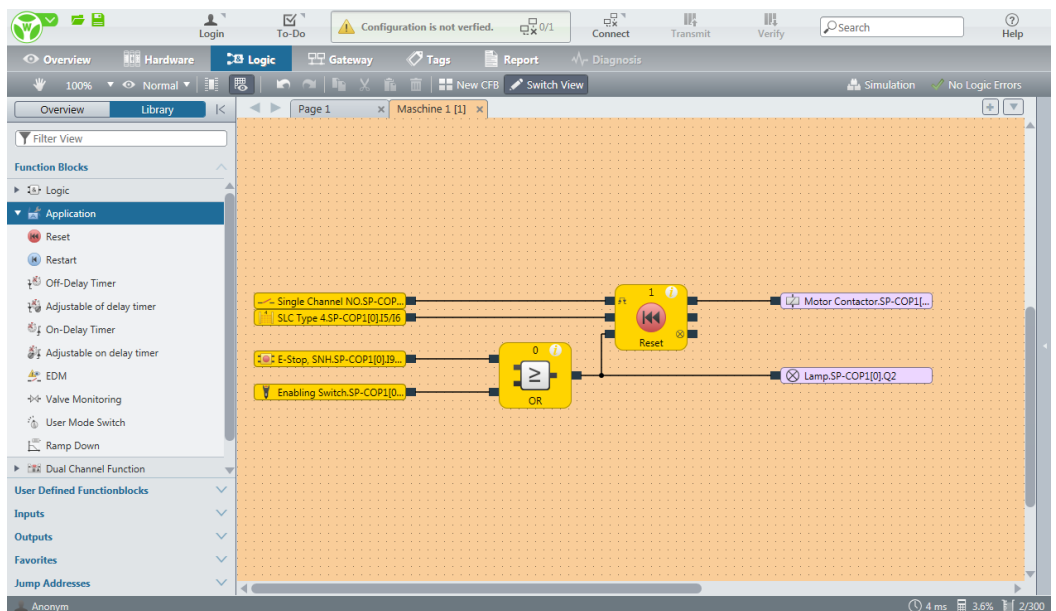


Illustration 181: New page in the logic editor for the new grouped function block

This is how you can add inputs and outputs to a grouped function block.

- ➔ Click on the tab for the grouped function block.
- ➔ In that area to the left, switch to the **Inputs** or **Outputs** view.
- ➔ Using drag-and-drop, move the inputs and outputs to the work area of the grouped function block and connect them as required in the logic.
- ➔ Double-click on an input or output in order to change its tag name.

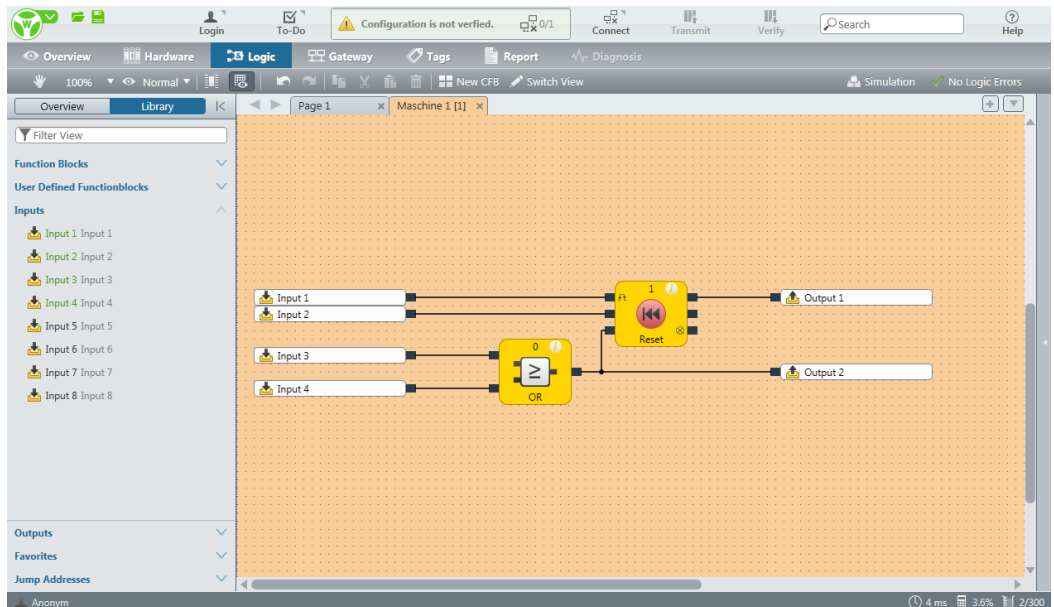


Illustration 182: Adding inputs and outputs to a grouped function block

Inputs and outputs added to the grouped function block are displayed in the main program for the function block itself and devices can be connected to it. As soon as the device has been connected, it is displayed in the logic of the grouped function block when the **External view** is displayed.

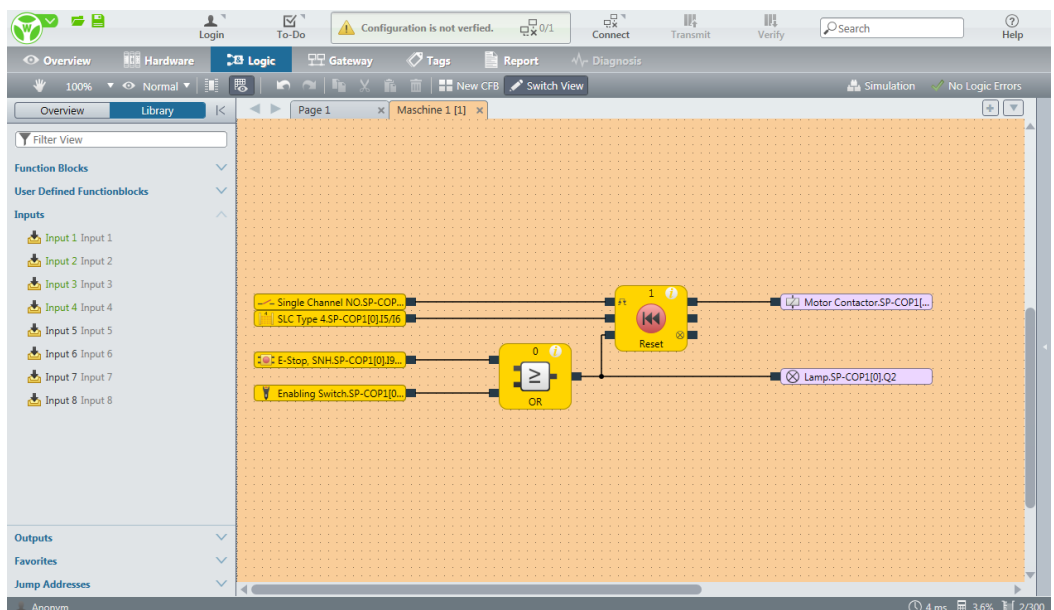


Illustration 183: Grouped function block with connected devices

In order to toggle between the internal tag name of the grouped function block (internal view) and the external I/O descriptions (external view), click on **Change view** in the toolbar.

- The **Internal view** indicates the tag name of the grouped function block for its inputs and outputs.
- The **External view** indicates what is connected to the grouped function block.

### This is how you can transfer a grouped function block to another PC:

- ➔ Save the project and open it on the other PC. The grouped function blocks contained in the project will automatically be imported.

### 7.11.2 User-defined function block

Once you have created a grouped function block, you can protect it against changes and import it into the selection list of function blocks to use it in future projects. The resulting function block is called a user-defined function block.

#### Function block diagram

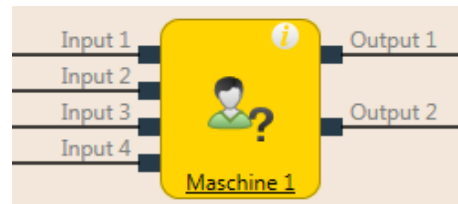


Illustration 184: Logical connections for a user-defined function block

A user-defined function block has the following properties:

- It can have a maximum of 8 inputs and 8 outputs.
- It cannot contain the fast shut-off function block or a different grouped or user-defined function block.
- The icon for a user-defined function block can either be user-defined or can be selected from a permanent library in samosPLAN5+.
- The user-defined function block is created in the logic editor, displayed in the selection list of the function blocks, and is available for all new projects on the same PC.
- When you open a project containing the user-defined function blocks on another PC, you have the following options:
  - You can import the user-defined function blocks into the selection list of function blocks on the new PC in order to use them in other projects.
  - Or you can import the user-defined function blocks for this project only. In this case, they will not be displayed in the selection list of function blocks.

**NOTICE** When calculating the total number of function blocks in a project, a user-defined function block is not counted as one block but instead with the number of function blocks used within it.

## This is how you create a user-defined function block:

In order to create a user-defined function block, you must have previously created it as a grouped function block (see *Grouped function block [ch. 7.11.1, p. 216]*).

- ➔ Open the view of the grouped function block by clicking on its tab.
- ➔ In the toolbar, click on **New CFB....** The **Function block details** dialog will open.

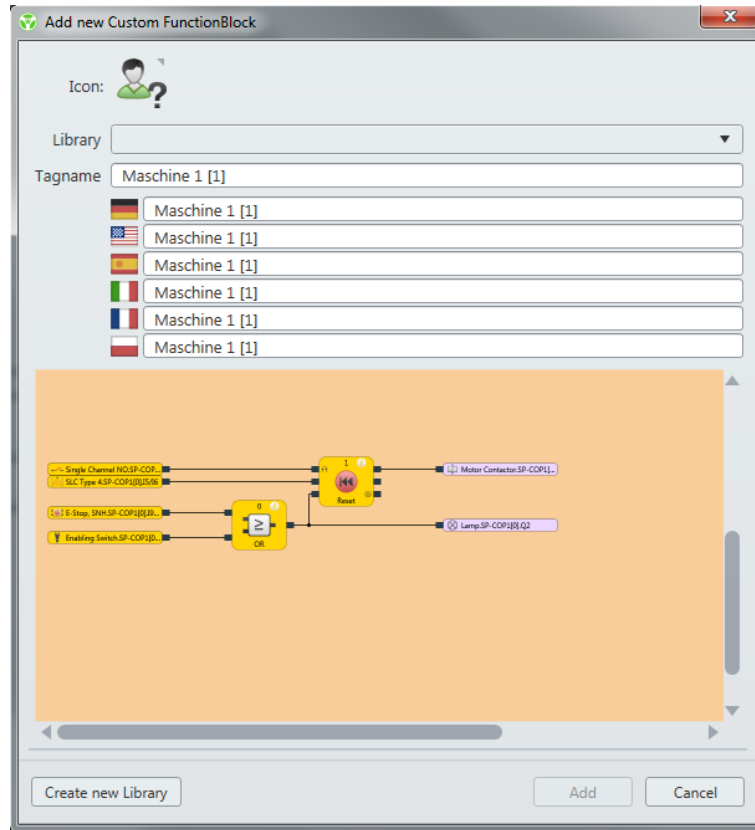


Illustration 185: Dialog for editing function block details for the user-defined function block

- ➔ Enter a name for the new user-defined function block.
- ➔ If you have not yet created a library for user-defined function blocks, press the **Create new library** button at the lower left. The following dialog will open:

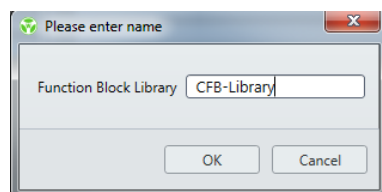


Illustration 186: Dialog for library name

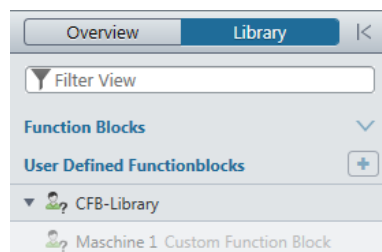


Illustration 187: New user-defined function block in the selection list of a library

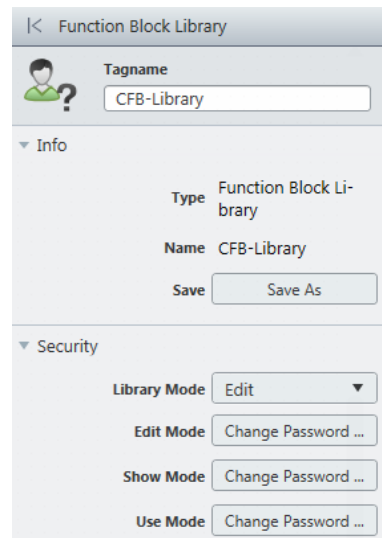
### This is how you can edit the properties of a function block library:

After you have given the library a name, you can set the library mode. This mode determines the properties of all of the function blocks contained in this library.

- Editable
- Use and display
- Use
- Protected

In addition, user rights can be assigned for access to the various properties of the function blocks.

Function block libraries can be saved with the **Save as** command.



The screenshot shows a configuration window for a 'Function Block Library'. At the top, there is a 'Tagname' field with the value 'CFB-Library'. Below this, the 'Info' section displays 'Type' as 'Function Block Library' and 'Name' as 'CFB-Library', with a 'Save' button labeled 'Save As'. The 'Security' section includes a 'Library Mode' dropdown menu set to 'Edit', and three other modes ('Edit Mode', 'Show Mode', 'Use Mode') each with a 'Change Password ...' button.

Illustration 188: Editing a library for user-defined function blocks

When a user-defined function block is placed in the work area, its content is displayed on a new page. In the example, the name of the user-defined function block is **Machine 1**. The work area of the user-defined function block is displayed in orange. The options of use, display, and edit for a user-defined function block depend on the parameters of the library from which the function block comes.

### This is how you edit a user-defined function block:

- The user-defined function block can be edited the same as a grouped function block, provided the user rights for the respective library allow this.
- A subsequently edited user-defined function block can be re-saved by clicking on **New CFB** in the toolbar.

### This is how you import a user-defined function block in a new project

With the import function you can simply apply a user-defined function block, which is stored on your PC's hard drive, into a new project.

- ➔ Switch to the **Logic | Library** view.
- ➔ Click on **User-defined elements** with the right mouse key.
  - ⇒ The context menu will open.

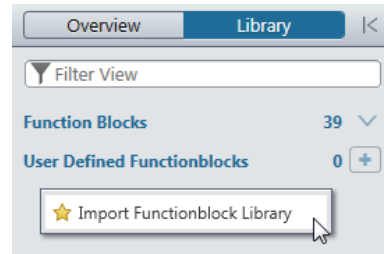


Illustration 189: Import of a function block library

- ➔ In the context menu, click on **Import Functionblock Library**.
  - ⇒ Windows Explorer will open.

You can search through Windows Explorer and select the library (file in \*.SPI format), in which the function block is.

### This is how you can transfer user-defined function blocks to another computer:

- ➔ Using drag-and-drop, move the desired user-defined function blocks to the logic editor and save the project.
- ➔ Open the project in the other computer.
  - You will be asked whether you wish to import the user-defined function blocks that were used in the project.
- ➔ Click on **Yes** in order to import the user-defined function blocks.
  - They will be displayed in the selection list of function blocks and are available in all new projects on the same PC.
- Or:**
- ➔ Click on **No** in order to import the user-defined function blocks as grouped function blocks only.
  - In this case, they will not be displayed in the selection list of function blocks and will only be available in the current project.

**NOTICE** The importing of function blocks must be activated in the hardware configuration view (**Settings, General** tab; deactivate the **Do not import user-defined function blocks when loading a project** option). When opening the project file, you will be prompted to confirm the import.

### This is how you can permanently delete a user-defined function block on your PC:

- ➔ Delete all occurrences of the user-defined function block on your computer or convert each individual one into a grouped function block by clicking on **Edit....** in the toolbar.
- ➔ Using the right mouse key, click on the user-defined function block that you wish to delete in the selection list of function blocks. The context menu will open.
- ➔ Select the **Delete user-defined function block** command.

**NOTICE** **You cannot undo this command.**  
You will still be able to use other projects containing the deleted user-defined function blocks. When you open an older project containing the user-defined function blocks that have been deleted from your PC, it will be treated like a project that was transferred from another PC. You will be asked whether you wish to import the user-defined function blocks in the project permanently as user-defined function blocks or as grouped function blocks for use in the current project only.

## 8 Simulating configuration

It is possible to simulate the programmed logic off-line in the logic editor. Inputs can be set to high or low and the resulting switching of outputs can be monitored. In addition, the timer and counter values of the function blocks used will be displayed on the function blocks during the simulation. Only valid logic configurations (i.e. without logic errors) can be simulated.

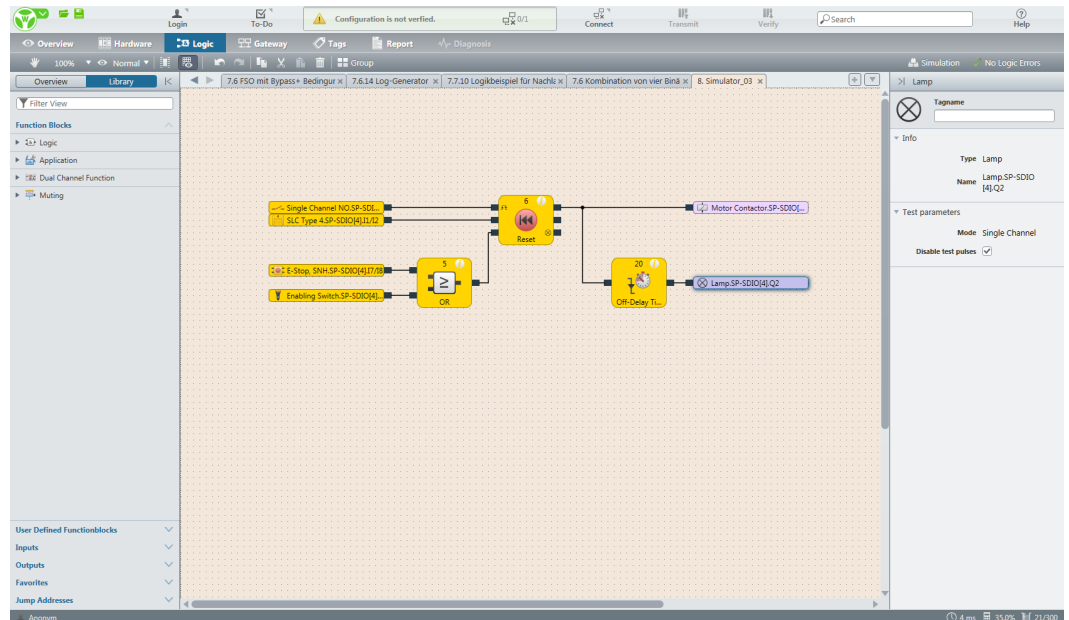


Illustration 190: Simulation of a valid logic configuration

### Perform simulation (overview)

- ➔ Click on the **Simulation** icon (🏠) in the toolbar in order to activate simulation mode. The background of the logic editor will become green and the simulation toolbar will appear.



Illustration 191: Simulation toolbar

- ➔ In order to start a simulation of the logic, click on the **Start/Pause** (A) button for a simulation at full speed (almost real-time).
  - ⇒ The **Timer** (B) will show the elapsed time.
  - ⇒ The timer can be reset using the **Reset** (C) button.
- ➔ In order to stop a simulation, click again on the **Start/Pause** (A) button.

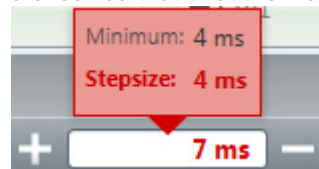
## Simulation time control

For logic processes that proceed too quickly in real time for them to be followed, there are two options:

- The simulation is executed continuously when you click on **Start/Pause** (A).  
Use the pulldown menu under **Real-time** (D) in order to slow down the sequence of the simulation.
- It is possible to execute a simulation incrementally. To do this, stop the simulation using the **Start/Pause** (A) button and then click on the **Increment** (E) button for incremental execution.

The time increments can be adjusted with the plus/minus button in the **Time window** (F). When you click on the **Increment** (E) button, the simulation will jump forward by the corresponding timeframe. In this input method, samosPLAN5+ rounds the entered time to the next possible permitted logic execution time.

Alternatively you can enter the time increments via the keyboard. To do this, click in the **time window** (F) and enter the desired time value. samosPLAN5+ automatically performs a check so that unauthorized values cannot be entered.



## Action possibilities

While the simulation is running, you can set an input to high by clicking on it. Inputs on high are displayed in green. You can set the input back to low by clicking again.

Once the simulation is stopped, it is possible to select inputs that are supposed to switch at the next-possible time. When you click on an input with the simulation stopped, initially only the input will appear (not the connected connection) in green to indicate that it will switch during the next cycle of simulation. This makes it possible to simultaneously switch multiple elements with the simulation stopped or in increment mode and to then observe the effect on the logic.

After the desired inputs have been set, the simulation can continue so that the logic and the outputs will switch accordingly. To this end, click on the **Start/Pause** button for continuous execution or use the **Increment** button for incremental execution.

**NOTICE** If the **EDM** or **Valve monitoring** function blocks are used, the recommendation is to execute the simulation in increments. These function blocks expect a high signal at their read-back input within 300 ms when their corresponding output was at high. This cannot be simulated in real time but rather only with the assistance of the corresponding small time increments.



## 8.1 Logic analyzer

### 8.1.1 Starting the logic analyzer

In order to open the logic analyzer, first simulation must be activated. Then the "Logic analyzer" button will be visible. After this has been clicked, the logic analyzer will open in a new window.

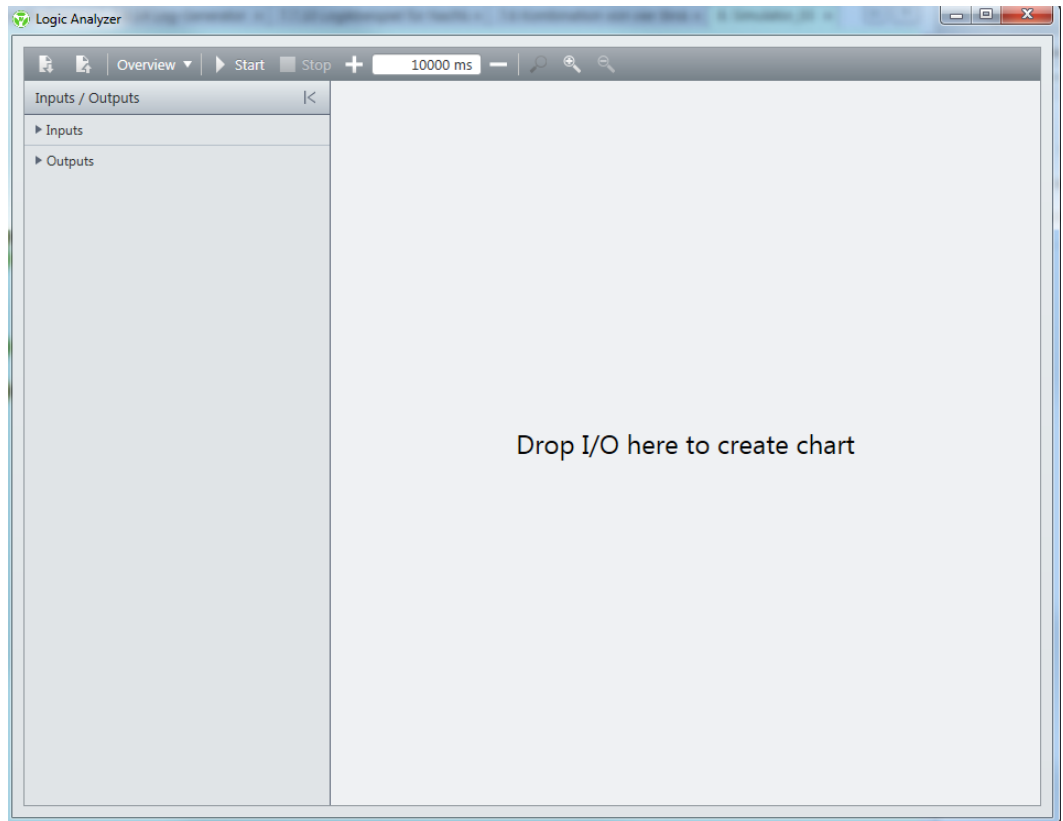


Illustration 192: Logic analyzer after open

## 8.1.2 Adding inputs and outputs

In order to see data in the analysis as well, inputs and outputs must be placed on the diagram section of the logic analyzer (drag-and-drop).

The inputs and outputs can be freely replaced in the sequence within the diagram view using the mouse.

In addition, the visible time window can be selected and the view can be enlarged.

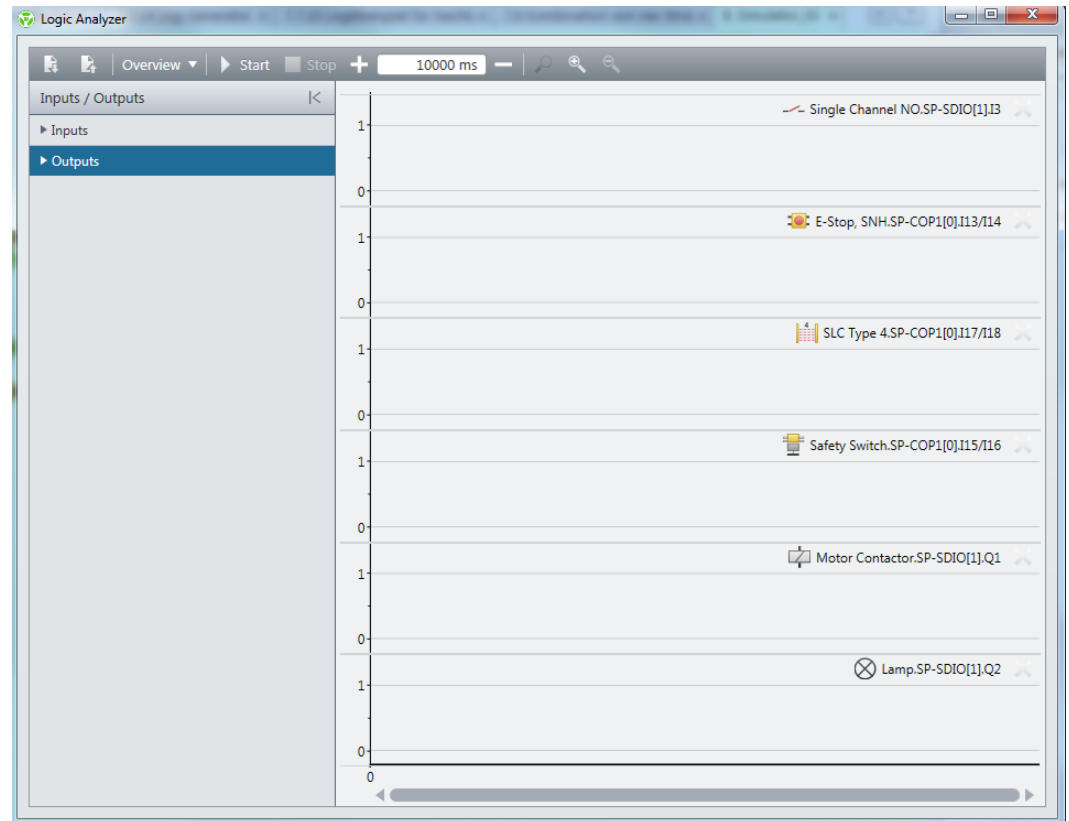


Illustration 193: Logic analyzer with inputs and outputs

## 8.1.3 Recording data

In order to analyze the logic, the "Start" button must be pressed in the logic analyzer dialog. Subsequently, the simulation can be started and the I/O data will be recorded.

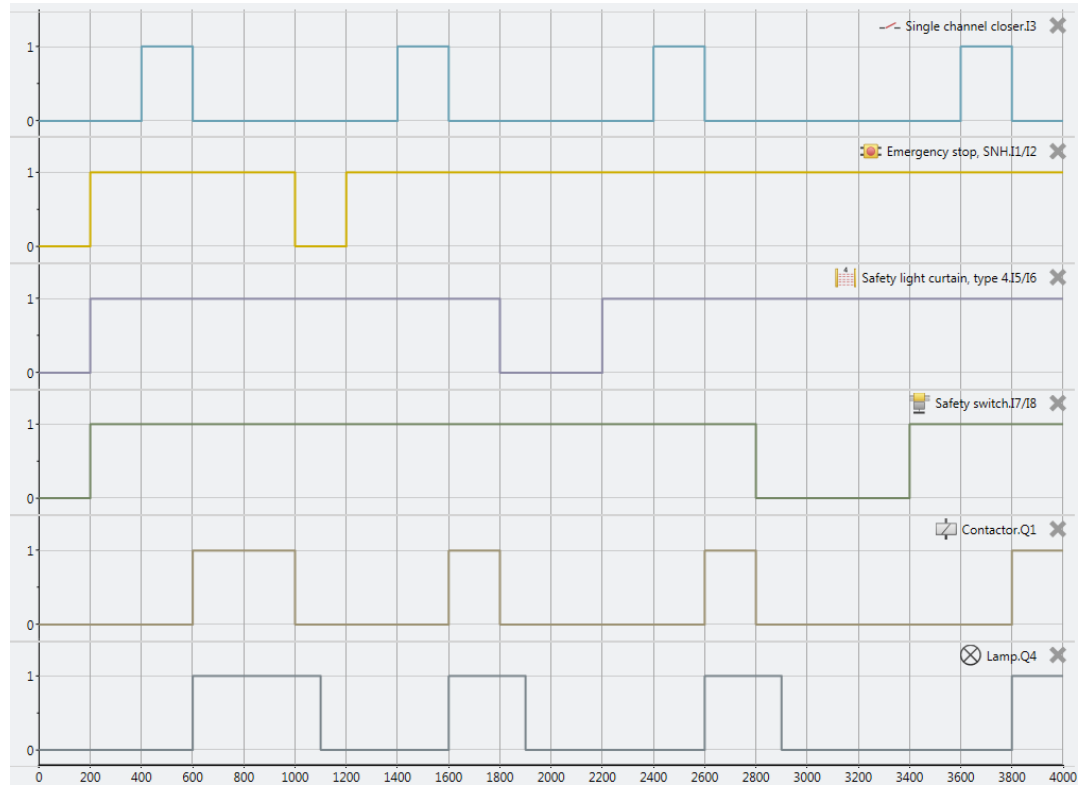


Illustration 194: Logic analyzer with value changes

In order to end the recording, click on the "End" button.




The "Real-time view" button can be used to switch to "Overview" which makes it possible to review the recorded data again in its entirety.

## 8.1.4 Exporting, importing and printing recorded data

In addition to the control functions for recording, the following functions are also available in the command bar of the logic analyzer.

These functions are only enabled if no recording is running.

Table 119: Key

Element	Function
	Imports a previously exported signal sequence in samosPLAN5+. You can view the signal sequence again in the logic analyzer.
	Exports the recorded data. You can import the export file with the recorded signal sequence to samosPLAN5+ again at a later time.
	Prints the recording most recently completed.

## 9 Force mode

In force mode, you can set the inputs in the samosPRO logic program to high or low using software control, regardless of the actual value of the physical inputs, while the system is in the run state. In this case, the samosPRO system, including the programmed logic, will behave precisely as if the physical inputs had actually assumed the respective value.

This will make it possible for you to test the wiring of your system in online mode and to check the function of your logic program, for example, during commissioning or maintenance.

### Technical information

With force mode, you can only directly influence the inputs in the logic of a samosPRO system but not the outputs or logic events, such as function blocks or jump labels.

### Safety information



ATTENTION

**Ensure that there can be no danger to people or damage to equipment!**  
**In force mode, you can freely influence the value of the safety inputs. By doing so, you can cancel the protective function of your safety equipment and induce a hazardous situation.**

- Make sure that no one is in the hazardous area of your machine or system before you activate force mode.
- Make sure that no one can enter the hazardous area of the machine or system during the time that force mode is active.
- Additional safety measures may be necessary when force mode is used.



ATTENTION

**Do not use force mode simultaneously on multiple PCs.**

When using force mode:

- Make sure that no one can also activate force mode from a second PC. This could result in a hazard-inducing condition.

## How to activate force mode

The following requirements must be met in order to use force mode:

- The module version of your samosPRO-COMPACT modules must be A-03 or higher.
- You must be logged on as a user with the authorization **F (use force mode)** (see *User administration [ch. 5.3.4, p. 35]*).
- The configuration of your samosPRO project must not be verified (CV LED on the COMPACT module is flashing **Yellow** at 1 Hz).
- The recommendation is to connect your PC to the samosPRO system via the USB interface when you wish to use force mode.

**NOTICE** If you attempt to activate force mode even though the configuration has already been verified (CV LED on the COMPACT module lit **Yellow**), a dialog will appear that will enable you to reset the status to **Not verified**.

- ➔ Click on **Connect** to establish a connection to your samosPRO system.
- ➔ Change to the **Logic** view and click on the **Forcing** button.
  - ⇒ A dialog for entering the time opens. Once this time entry elapses, force mode is automatically ended if no other actions have been initiated.

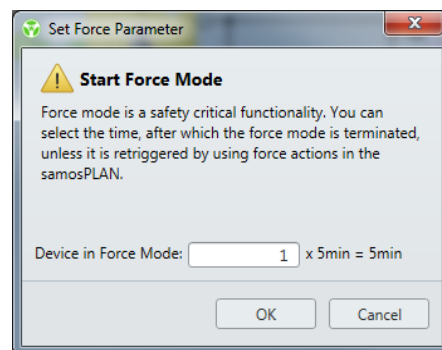


Illustration 195: Dialog window when starting force mode

- ➔ Click on the desired time in the selection list and click on **OK**.
  - ⇒ Force mode will be started and the background color of the logic editor will change to orange.

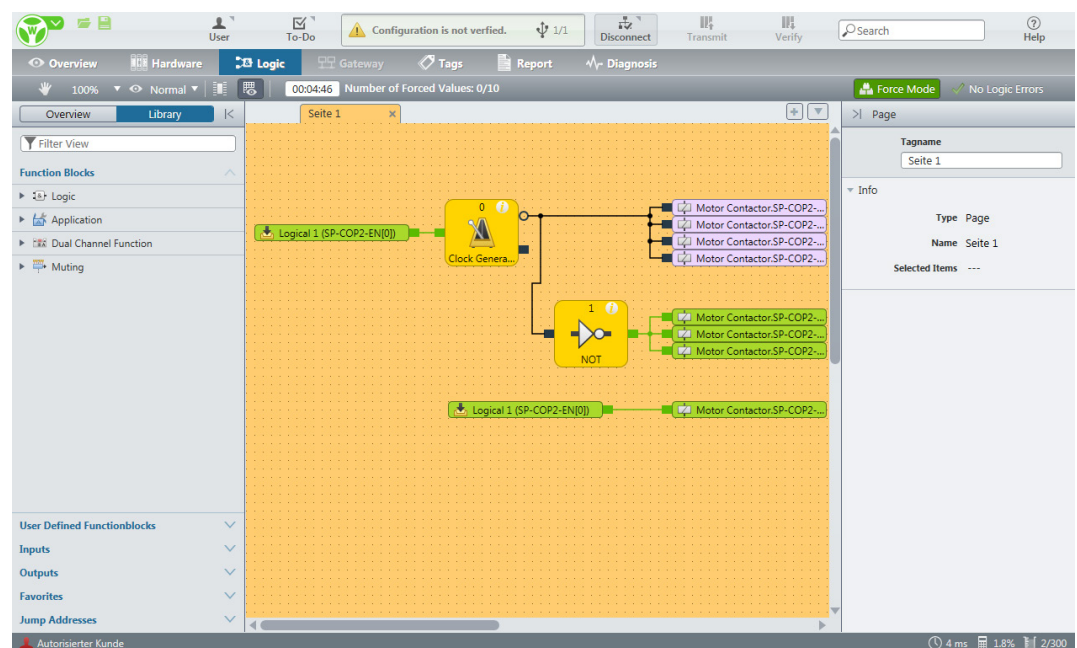


Illustration 196: Logic editor view with activated force mode

**NOTICE** While force mode is active, it is not possible to log out, to receive and compare a configuration, or to stop the device.

## How to force one or more inputs

➔ Select one or more inputs that you would like to force.

- Force individual input: Click on the input.
- Simultaneously force several inputs: Left-click on the empty background of the work area and drag a capture frame (marked in blue) around the desired inputs.

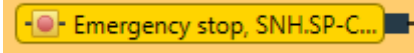
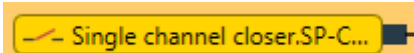
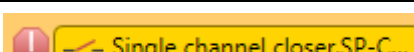

➔ In the context menu, select the desired forcing option.

Table 120: Forcing options in the context menu

Option	Function
Forcing	Starts the force mode.
Switch over all force values	Switches over the input value. <b>Note:</b> You can also switch over an individual input by clicking it.
Deactivate forcing	Stops the force mode. The input will be evaluated with its actual physical value by the samosPRO system.

**Representation:** A forced input is indicated with an exclamation point. An active input (high) is shown in green, while an inactive input (low) is white. Inputs whose forced value is different than their actual physical value are shown in light blue.

Table 121: Display in the force mode

State	Display
Physically low input, non-forced	
Physically high input, non-forced	
Physically high input, forced to low	
Physically high input, forced to high	

## Info

- While an input is being forced in the logic, the real value of the physical input is not displayed in the logic editor, but only in the **Hardware configuration** view.
- Forcing only influences the inputs in the logic program but not the physical inputs of the expansion modules. Examples:
  - Forcing does not affect the inputs of a module that are being used for fast shut-off. Thus, the output in the hardware may stay at low even though the inputs in the logic are forced to high, because fast shut-off in the module is being controlled directly by the physical inputs.
  - Forcing does not affect inputs whose values are not being controlled by the logic program but instead are being directly transferred to a PLC via a gateway.
- Force mode always applies to the entire project. For logic programs comprising multiple pages in the logic editor, this means that a forced input is not only set to the same value on the currently displayed page in the logic editor but everywhere that it is used.
- Contrary to simulation mode, you can use the **EDM** or **Valve monitoring** function blocks as well in force mode if corresponding devices are actually connected that will transmit the required read-back signal when the outputs are activated.
- A maximum of 10 inputs can be forced simultaneously.
- When using a samosPRO gateway, note that the process image of the samosPRO gateway **always** reflects the actual physical value of the inputs and outputs of the connected devices and not the (virtual only) forced value of an input in the logic program. If the value of

an **Output** is changed (e.g. from high to low) i.e. by forcing an **Input** in the logic program (e.g. from high to low), the (actually changed) physical value of the **Output** (low in the example) in the process image is transmitted to the PLC, however **not** the forced low value of the input in the logic program, but again the actual physical value of the input on the device (high in the example). Consider this when evaluating the data transmitted in the PLC.

### Stopping the force mode

Force mode can be ended in the following ways:

- Manually by the user
- Automatically after the time defined at the start has elapsed when no further actions have been initiated.
- Automatically after 3 seconds if the samosPRO system finds an error (e.g. when there is a disconnection in the connection to the PC).

When ending force mode, all of the outputs for the samosPRO system are set to low and the active application is stopped.



#### **Make sure that exiting force mode will not cause a hazardous situation!**

- Make sure that your machine or system is in a safe state and cannot be damaged when ending force mode.
- During force mode, the real value of an input may have changed (e.g. switches pressed, safety doors open, etc.). Before restarting your machine or system, make sure that this will not result in any hazard.

- ➔ Click on the green **Forcing** button.
- ➔ After the time defined at the start has expired, force mode will automatically end if no action (e.g. force input) has been executed.  
In force mode, a timer is displayed in the command bar until force mode is automatically ended.

# 10 Two-channel evaluation and synchronous time

The modules, e.g. SP-COP, SP-SDIO, or SP-SDI, can carry out a two-channel evaluation if predefined input sensor elements from the element window (e.g. solenoid switch SMA, safety light curtains SLC, etc.) are connected to them in the hardware menu. When such type of input sensor element is selected, then you will not require any separate function block for the two-channel evaluation (e.g. light curtain evaluation, switch evaluation, or solenoid switch).

The two-channel evaluation checks the correct sequence of the two input signals. When one of the two signals has effected a switch-off, then it is expected that the other signal will follow accordingly. The question as to what values the two signals must have depends on the type of two-channel evaluation. There are two options:

- Equivalent evaluation
- Discrepant evaluation

A **Synchronous time** can be configured as an option. Synchronous time determines how long the two inputs will be allowed to have discrepant values after one of the two input signals has changed its value without this being evaluated as an error.

**NOTICE**

**The following must be noted during the configuration:**

- The synchronous time must be a multiple of 4 seconds.
- When signals of tested sensors are connected to SP-COP, SP-SDIO, or SP-SDI modules, then the synchronous time should be at least the set **Test gap [ms]** plus the **Max. Off-On delay [ms]**, because the signal change at the input of the module can be delayed by this time. Both values are displayed in the samosPLAN5+ report for the test output used.

The following truth table describes the synchronous time conditions for the two-channel equivalent and the two-channel discrepant input evaluation:

*Table 122: Two-channel evaluation*

Evaluation type	Input A	Input B	Synchronous time timer <sup>2)</sup>	Status of the two-channel evaluation	Input of the I/O module in the logic editor	Synchronous time error
Equivalent	0	0	0	Inactive	0	Unchanged <sup>3)</sup>
	0	1	< Synchronous time	Discrepant	0	Unchanged
	1	0	< Synchronous time	Discrepant	0	Unchanged
	1	1	0	Active	1	0
	x	x	≥ Synchronous time (timeout)	Error	0	1
Discrepant	0	1	0	Inactive	0	Unchanged
	0	0	< Synchronous time	Discrepant	0	Unchanged
	1	1	< Synchronous time	Discrepant	0	0
	1	0	0	Active <sup>4)</sup>	1	Unchanged
	x	x	≥ Synchronous time (timeout)	Error	0	1

<sup>2)</sup> When synchronous time is active (> 0), the synchronous time timer is restarted with the first signal change that leads to a discrepant state. When synchronous time is deactivated (= 0), the synchronous time timer does not start, i.e. a timeout will never occur.

<sup>3)</sup>Unchanged = The last state is retained.

<sup>4)</sup>If the correct sequence was retained.



## Two-channel evaluation and synchronous time

The following rules apply to the transitions between the various states of the two-channel evaluation:

A two-channel evaluation can only be active (the input of the I/O module in the logic editor changes from low to high) in the following cases ...

- The state since the last active was inactive at least once; i.e. it is not possible to switch from active to discrepant and back to active; and
- Synchronous time has not expired or is deactivated.

### NOTICE

If the correct sequence for reaching the active state was not maintained (i.e. when the state has switched from active to discrepant to active), then SP-SDIO and SP-SDI modules with module versions B-01 and higher will indicate this sequence error within 100 ms at the latest if the synchronous time has not elapsed before this (i.e. if the synchronous time is set at 0 or to a value > 100 ms). Older modules will not show this sequence error; however, their input in the logic editor likewise remains at low.

In the event of a synchronous time error or a sequence error, the module will behave as follows:

- The MS LED for the respective module will flash **red** (1 Hz).
- The LEDs for the respective inputs will flash **green** (1 Hz).
- The **Input data state** for the module will be at low in the logic editor.

### Resetting the error:

A synchronous time error (timeout) or sequence error is reset when the inactive state was reached.

# 11 Transferring the system configuration

The configuration of the safety control initially only exists as a project, i.e. only as a samosPRO project file. This project file must be transferred into the samosPRO program removable storage SP-COP-CARD1 via the COMPACT module.

**NOTICE** The samosPRO program removable storage SP-COP-CARD1 and the COMPACT modules SP-COP communicate via an internal interface. It is also possible to directly connect a PC to the program removable storage SP-COP-CARD1.

The configuration data are checked for compatibility during the transfer to the program removable storage and can subsequently be verified (through read and compare) and then optionally provided with the desired protective rights from the user control (e.g. write protection).

Using the program removable storage and a PC, the verified project data, for example, can be copied without additional editing with samosPLAN5+ and transferred to any desired number of samosPRO safety controls; in doing so, the configuration data are copied exactly, including the verification and any write protection information that was set during the configuration of the first safety control with this data.

## 11.1 Transferring project data into the safety control

After the transfer, the configuration data are read back from the program removable storage if the verification has been activated in samosPLAN5+ (see *Verifying the configuration [ch. 11.3, p. 235]*).

**NOTICE** The reading back of the configuration data from the program removable storage requires a bit of time; the program removable storage must not be removed during this time. samosPLAN5+ shows a corresponding warning as long as this process is ongoing.

## 11.2 Compatibility check

The configuration data contain an electronic type code and a version code for each module that is supposed to be configured. During transfer, each module checks whether it is compatible with the configuration data. The compatibility check is solely based on the functional part of the respective module, not on the hardware version; the design of the terminals, for example, is not included.

If the compatibility check has a negative result, a corresponding error message is generated in the respective module and in the COMPACT module.

**NOTICE** Some modules in samosPLAN5+ have various version numbers stored, which means that a compatible module can be selected from a list underneath the module.

## 11.3 Verifying the configuration

After the configuration has been downloaded into the control, the samosPRO system can be verified. To do this, the downloaded configuration data are read back from the samosPRO system and compared to the project data. If they match, the data are displayed in a report. When the user confirms the correctness, the system is considered to be verified.

**NOTICE** Once the configuration is verified, then samosPRO automatically switches into the run state after the supply voltage is switched on.  
If the configuration is not verified, then the system must be manually placed into the run state using samosPLAN5+ (see *Changing the device state [ch. 12.3, p. 242]*).  
It is only possible to verify a samosPLAN5+ project in connection with a samosPRO system having COMPACT modules.

This is how you can verify a configuration:

- ➔ Click on the Verify button.

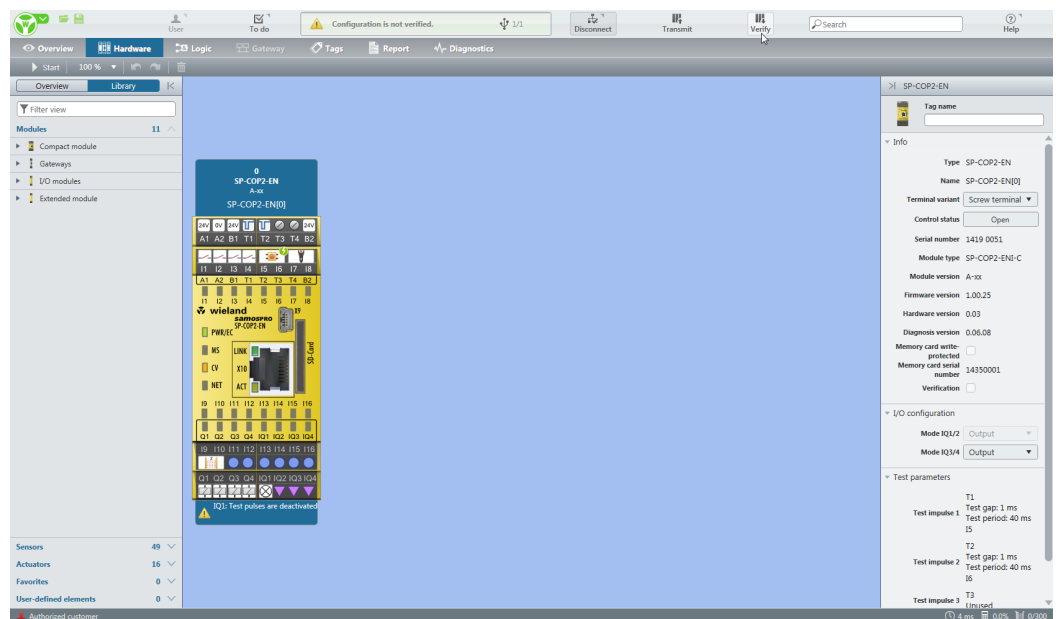


Illustration 197: Receiving and comparing the configuration

# Transferring the system configuration

- ➔ The **Verification report** window will open. Click on **Verify** below when the displayed configuration is the expected configuration. The system will then be considered verified.

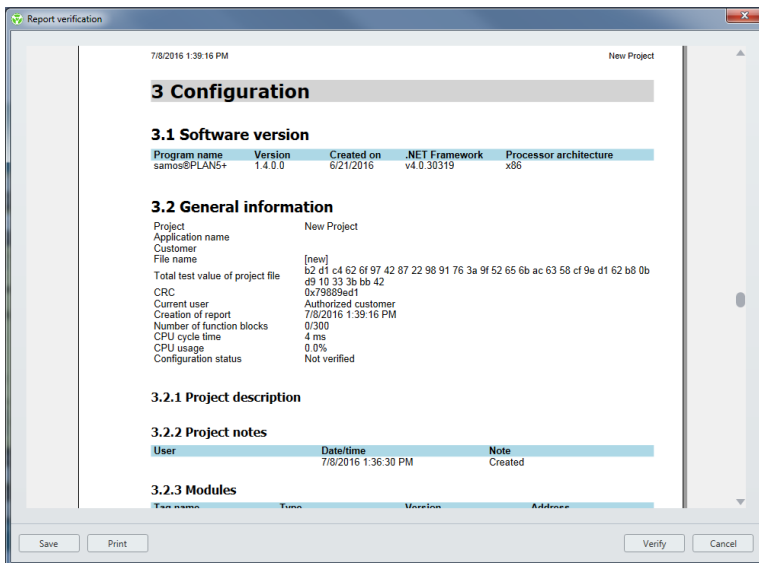


Illustration 198: Mark the device as verified.

**NOTICE** The configuration of connected elements, e.g. safety light curtains, is not included in this process. Their verification takes place according to the operating instructions for these devices.

- ➔ If differences are found between the project data and the configuration data read back, a corresponding message is displayed, including information as to possible actions to take. The verification of configuration cannot take place. In that case, note the information in the error message as to how to proceed. End the dialog by clicking on **Cancel**.

Once verification has been successfully completed, a project report will subsequently be created, which you can then print out or save.

The status verified/not verified will be displayed in the lower right corner of samosPLAN5+ and indicated by lighting up of the CV LED on the samosPRO-COMPACT module.

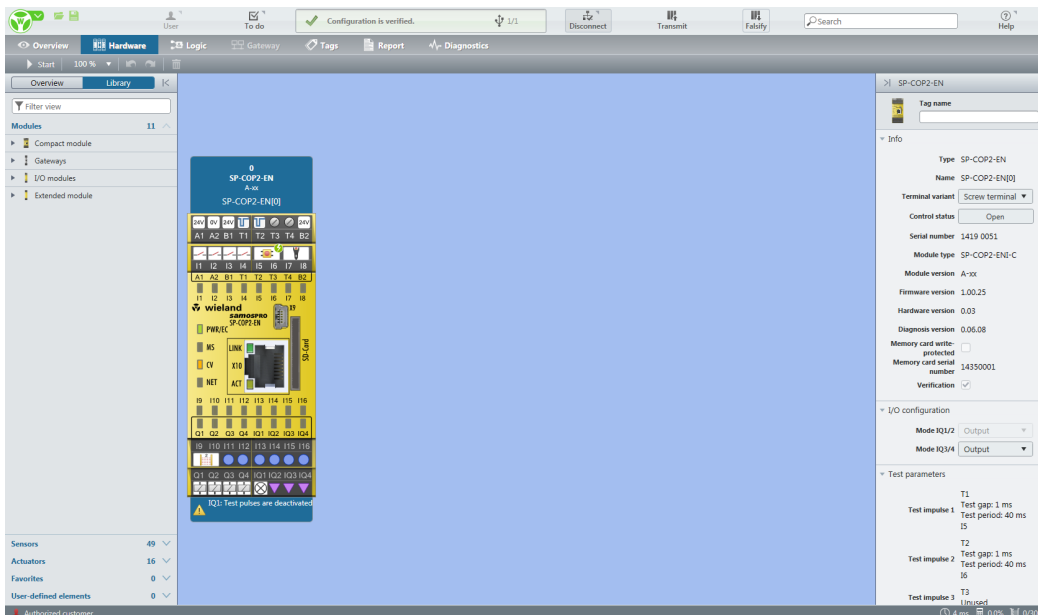


Illustration 199: Verification successful

## Transferring the system configuration

The verification flag is copied during read back of the data into the program removable storage and automatically also transferred to each safety control onto which the configuration data are being duplicated.

The safety control is also ready for use when the configuration is only validated and has not been verified. The automatic transition of the samosPRO system into the run state however is not possible after switch-on of the power supply in this case.

**NOTICE** The dialog for verification will only appear at the request of the user so that the verification process does not have to be carried out with each change in the configuration or each time new project data is uploaded.

- In order to validate the samosPRO system, the safety functions must be completely tested on the machine or system and must fully function. The validation is identical with respect to content to the technical check during commissioning of the samosPRO system.

# 12 Device states in the samosPRO system

The samosPRO system recognizes various device states during operation. Some device states require user involvement, e.g. change of the state from **Stop** to **Run** using samosPLAN5+. Other states are based on the internal self-test of the samosPRO system, e.g. **Internal errors**. The following tables summarize the device states in the samosPRO system.

## 12.1 Device state and LED displays in the COMPACT modules (SP-COP1, SP-COP2-ENx)

### Flash code meaning

Table 123: Key


















Symbol	Meaning
	LED off
	LED flashing
	LED lights up




Table 124: Device state and LED displays in the COMPACT modules

PWR/EC LED	Meaning	Additional info
 Red flashing	An error has occurred in the control. All 24V outputs have been switched off. The control must be restarted with a power ON reset after the cause of the error has been eliminated.  The number of flash pulses indicates the error class to which the occurring error belongs.	Number of flashing pulses = error class 2: Configuration data 3: Application 4: Self-test 5: Voltage/current monitoring 6: I/O modules 7: Cross-communication 8: Internal
 Green flashing (1 Hz)	The supply voltage at A1, B1, or B2 is outside of the range of 16.8 V to 30 V.	
 Green	The supply voltage at A1, B1, and B2 is within the range of 16.8 V to 30 V.	
MS LED module state	State	Additional info
 Red flashing (1 Hz)	There is no project at the control or the project data is faulty (because, e.g., the number of inserted I/O modules does not match the project)	No module or faulty module configuration
 Green flashing (1 Hz)	Project data adopted from control and I/O modules; waiting for start command	
 Green	Control has started.	
 Red/green flashing	One or more inputs have a cable break or short-circuit to 24V.  Or there is a sequence/synchronous time error at a two-channel input.  Or an output has a test error (e.g. short-circuit).	

## Device states in the samosPRO system

Code-verified CV LED	Control behavior
 Yellow flashing (1 Hz)	The project at the control has not been verified. The control will not start automatically after power ON reset.
 Yellow	The project at the control has been verified. The control will start automatically after power ON reset.
NET	Meaning
 Flashing green (for 3 s)	Connection setup with control

Input LED	Meaning	Additional info
 Green flashing (1 Hz)	A single-channel input has a test error (cable break or short-circuit at 24 V) or the input was not configured in the project and 24 V is pending.	Applies to I1 to I16 and IQ1 to IQ4 if single-channel has been configured. Flashes synchronously with MS LED in red.
 Green flashing, alternating (1 Hz)	Two-channel input has synchronous time error or a sequencing error or at least one of the two inputs has a test error (cable break or short-circuit at 24 V)	Applies to I1 to I16 and IQ1 to IQ4 if two-channel has been configured. Input pair flashing on and off
 Off	Signal level at the input terminal is 0 V.	
 Green	Signal level at the input terminal is 24V.	

Output LED	Meaning	Additional info
 Green flashing (1 Hz)	Output has a test error.	Applies to Q1..Q4 and IQ1..IQ4
 Off	Output is switched off.	
 Green	Output is switched off.	



### 12.2 Device state and LED displays in the expansion modules (SP-SDIO, SP-SDI)

**NOTICE** The displays of the MS LED and the input LEDs I1 to I8 are identical to those for the SP-SDIO and SP-SDI expansion modules.

#### Flash code meaning

Table 125: Key











Symbol	Meaning
	LED off
	LED flashing
	LED lights up

Table 126: Displays of the MS LED

MS LED	Meaning	Info
	Supply voltage outside of operating range	Check supply voltage at terminals A1 and A2.
 Red/green flashing (1 Hz)	Repairable external error	Check cable of flashing inputs and outputs. If all output LEDs are flashing, check the supply voltage of terminal A1 and A2 for this module.
 Green flashing (1 Hz)	System is in the stop state or the supply voltage at A1 is outside of the range of 16.8 V to 30 V.	Start the application in samosPLAN5+. Check supply voltage at A1.
 Green	System is in the run state and the supply voltage at A1 is within the range of 16.8 V to 30 V.	
 Red flashing (1 Hz)	Invalid configuration	
 Red flashing (2 Hz)	Critical error in the system; suspected in this module. Application has been stopped. All outputs are switched off.	Switch supply voltage off and back on. If the error has not been eliminated after this has been done multiple times, then replace module. In order to contain the module affected, use the diagnostics display in samosPLAN5+.
 Red	Critical error in the system; suspected in a different module. Application has been stopped. All outputs are switched off.	Switch supply voltage off and back on. If the error has not been eliminated after this has been done multiple times, then replace module in which the red LED is flashing (2 Hz). In order to contain the module affected, use the diagnostics display in samosPLAN5+.

## Device states in the samosPRO system

Table 127: Displays of input LEDs








Input LEDs (I1–I8)	Meaning
	Signal level at the input terminal is 0 V. Safety mat: Both inputs activated.
 Green	Signal level at the input terminal is 24V.
 Green (1 Hz) Synchronously alternating with the red MS LED	Signal level at the input terminal is 0 V and a repairable error at the two-channel input is pending.
 Green (1 Hz) Synchronous with the red MS LED	Signal level at the input terminal is 24V and a repairable error is pending.

Table 128: Displays of output LEDs

Output LEDs (Q1–Q4)	Meaning
	Output has a test error.
	Output is switched off.
 Green	Output is switched on.



### 12.3 Changing the device state

You have to implement certain state changes in the samosPRO system manually in samosPLAN5+. These changes in the device state are as follows:

- Change from **Stop** to **Run**
- Change from **Run** to **Stop**

In order to change the device state, click on the **Stop application** or **Start application** icon respectively, in the hardware configuration next to the display of modules.

Table 129: Start and Stop buttons

Button	Function	Description
	Start	Places the samosPRO system in the <b>Run</b> state
	Stop	Places the samosPRO system in the <b>Stop</b> state

**NOTICE** Once the configuration is verified, then samosPRO automatically switches into the run state after the supply voltage is switched on. If the configuration is not verified, then the system must be manually placed into the run state using samosPLAN5+.

### 12.4 Behavior at system start

When the samosPRO safety control transitions from the stop state to the run state:

- The **First logic cycle** status bit of the COMPACT module is at high for the duration of the logic execution time. This status bit is available in the logic editor as an input element of the COMPACT module.
- All timers and states, including the error states of the function blocks, will be reset.

## 13 Technical commissioning

Before you start technical commissioning, the configuration of the samosPRO system must be complete.

### 13.1 Wiring and supply voltage



**Note the technical data in the "samosPRO hardware" manual when connecting the samosPRO system!**

- Connect the individual field devices to the corresponding signal connections and for each safety input, test/signal output, and safety output, check whether they are behaving as required for the application. You will support the diagnostic information of the samosPRO LEDs when validating the individual field signals. Check whether the external wiring, the design of the wiring, the selection of the command encoder, and their arrangement on the machine meet the required safety level.
- Eliminate any malfunctions (e.g. incorrect wiring or crossed signals) at all safety inputs, test/signal outputs, or safety outputs before you proceed with the next step.
- Switch on the voltage supply. As soon as the voltage supply is present at the A1 and A2 connections of the COMPACT+ SP-COPx modules or the SP-SDIO modules, the samosPRO system will automatically implement the following steps:
  - Internal self-test
  - Loading of the stored configuration
  - Test of the loaded configuration for validity

The system will not transition into operation if the aforementioned steps cannot be completed successfully. If there are errors, a corresponding LED display will appear (see "samosPRO hardware" manual, BA000966) and the samosPRO will set all of the transmitted values to low.

### 13.2 Transferring the configuration

After you have configured the hardware and the logic in the samosPRO system and checked it for correctness, transfer the configuration to the samosPRO system via samosPLAN5+.

### 13.3 Technical check and commissioning

The machine or system that is being protected by a samosPRO safety control may only be commissioned after a successful technical check of all of the safety functions. The technical check may only be completed by a qualified person.

The technical check comprises the following test points:

- ➔ Mark all of the connection lines and plug connectors on the samosPRO system clearly to prevent mixups. Because the samosPRO system has multiple connections with the same shape, make sure that connection lines that are disconnected are not connected back to the wrong connection.
- ➔ Verify the configuration of the samosPRO system.
- ➔ Check the signal paths and the correct integration into higher-level controls.
- ➔ Check the correct data transmission from and to the samosPRO safety control.
- ➔ Check the logic program of the safety control.
- ➔ Fully document the configuration of the entire system, the individual devices, and the results of the safety check.
- ➔ Check the safety functions of the machine or system completely and make sure that the safety functions are fully functioning.
- ➔ To prevent unintentional overwriting of the configuration, activate the write protection of the configuration parameters in the samosPRO system. Changes will now no longer be possible until the write protection has been canceled.

## 14 Troubleshooting

If an error occurs, you can find further information at:

- *Device states in the samosPRO system [ch. 12, p. 238]* (list of the LED error displays)
- *Complete list of error messages [ch. 15.1, p. 245]* (error codes, error causes, and measures for error elimination)
- "samosPRO hardware" manual (BA000966)

### "Diagnostics" view

Error codes and error messages can be displayed as well in the **Diagnostics** view if you have established a connection with the samosPRO system.

For more information as to how you can carry out diagnostics, please see the following: *"Diagnostics" view [ch. 5.9, p. 65]*

# 15 Appendix

## 15.1 Complete list of error messages

Table 130: Fehlermeldungen SP-COP

Error No.	Error type	Logbook message	System behavior	Remedy
00000001	Info	Function block log generator info	System continues to run	
00000002	Warning	Function block log generator warning	System continues to run	
00000003	Error	Function block log generator error	System continues to run	
10100001	Error	An unknown error occurred.	Connection terminated	Contact support
10100002	Error	An internal error occurred.	Connection terminated	Contact support
10100003	Error	Time exceeded when preparing a message for the control.	No connection	
10100004	Error	The value cannot be forced because force mode is inactive.	Connection kept	Activate force mode
10100005	Error	The control does not support the message type.	Connection terminated	
10100006	Error	The hash value of a read file is not valid.	Connection terminated	Contact support
10100007	Error	The header size in the message from the control is not plausible.	Connection terminated	Contact support
10100008	Error	The user data size in the message from the control is not plausible.	Connection terminated	Contact support
10100009	Error	The total data size does not match the number of received data.	Connection terminated	Retry Contact support
1010000A	Error	A data flow error in a segmented read message occurred.	No connection	Retry Contact support
1010000B	Error	The checksum in a message from the control is not valid.	Connection terminated	Contact support
1010000C	Error	Time exceeded when sending a message to the control. Possible reasons: Communication to SP-COP already exists; Ethernet or USB connection is interrupted.	Connection terminated	Check connections Contact support
1010000D	Error	A timeout occurred during receiving a message from the control. Possible reasons: Communication to SP-COP already exists; Ethernet or USB connection is interrupted.	Connection terminated	Check connections Contact support
1010000E	Error	Unexpected message received.	Connection terminated	Contact support
1010000F	Error	The message from the control is corrupt.	Connection terminated	Contact support
10100010	Error	The message from the control is corrupt.	Connection terminated	Contact support

Error No.	Error type	Logbook message	System behavior	Remedy
10100011	Error	The message to the control could not be processed.	Connection terminated	Retry Contact support
10100012	Error	The control could not positively respond to the request.	Connection kept	Retry Contact support
10100013	Error	The number of request retries has been exceeded.	Connection terminated	Retry Contact support
10100015	Error	Communication to the control could not be established.	No connection	Check connections Contact support
10100016	Error	The password is not valid for the user to be logged on.	Connection kept	Verify correct password
10100017	Error	The control could not accept the desired state.	Connection kept	Retry Contact support
10100018	Error	The memory card of the station is not plugged in.	Connection terminated	Insert valid SD-Card
10200002	Error	The project on the control is invalid.	No connection	Download new valid Project
10200003	Error	The verification status of project and control is not the same.	No connection	Re-verify project
10200004	Error	The PC project and project on the control could not be synched.	No connection	Reconnect Contact support
10200005	Error	Current user does not have permission to communicate with the control Connection has been closed.	No connection	Redefine user rights
10200006	Warning	The project on the target does not match the module configuration.	Connection kept	Update Project or Hardware
10200007	Error	An error is reported by the control		
10200008	Error	The checksum in samos®PLAN5+ and control deviates.		Contact support
10200009	Error	The waiting time permitted for the project has been exceeded.		Retry Contact support
1020000A	Info	The verification was cancelled.		Retry Contact support
1020000B	Warning	The incorrect project file is still running on the station and must be replaced with the updated project file. Please update the station by connecting and downloading the updated project again		Update the station with the fixed project
10300001	Error	The logic analyzer data could not be saved.		Check Windows file access
10300002	Error	The logic analyzer data could not be loaded.		Retry Contact support
10300003	Error	I/O could not be found.		Contact support
10400001	Error	The log messages could not be saved.		Check Windows file access
10400002	Error	The file contained more than 64 messages. Only the first 64 have been imported.		Reduce log message count
10400003	Error	The log messages could not be imported.		Contact support

Error No.	Error type	Logbook message	System behavior	Remedy
10500001	Error	Login to control failed.		Retry Contact support
10600001	Error	This user already exists. Please choose another user name.		Use other name
10600002	Error	Could not import user list.		Retry Contact support
10600003	Warning	The following users have not been imported because they already exist.		
10700001	Error	Project File could not be opened. File format is incorrect.		Update samosPLAN5+ and retry Contact support
10700002	Error	Creating project from module configuration failed!		Update samosPLAN5+ and retry Contact support
10700003	Error	Project file could not be saved!		Check Windows file access
10700004	Error	Project File could not be opened. File format is incorrect.		Update samosPLAN5+ and retry Contact support
10700005	Error	Library file could not be opened. File format is not correct.		Update samosPLAN5+ and retry Contact support
10700006	Error	Project structure is incorrect.		Update samosPLAN5+ and retry Contact support
10700008	Error	Could not load settings file. File is corrupt.		Update samosPLAN5+ and retry Contact support
10700009	Error	Failed to import library. Elements are already exist.		
1070000A	Error	File cannot be opened. Signature is not correct.		Update samosPLAN5+ and retry Contact support
1070000B	Error	Cannot load gateway configuration because the configuration is for another gateway type.		
1070000C	Error	Version of the project file is not supported by this version of the program. Please try it with the current version of samosPLAN5+.		Update samosPLAN5+ and retry Contact support
1070000D	Error	The configuration data for a module can't be loaded correctly.		Update samosPLAN5+ and retry Contact support
10800001	Warning	Forcing more than 10 values is not permitted.		

Error No.	Error type	Logbook message	System behavior	Remedy
11000000	Error	The HTML help could not be found. Please check if it is installed correctly.		Repair or reinstall samosPLAN5+ Contact support
12000000	Error	The version information was incorrect. Please contact Support.		Contact support
12000001	Error	Could not reach update server. Please check internet access		Check internet access
13000000	Error	Test gap exceeds half the maximum period duration.		Verify test parameters
13000001	Error	Test period exceeds maximum test period of input.		Verify test parameters
13000002	Error	A test period with these values for minimum and maximum cannot be configured.		Verify test parameters
13000003	Error	Test gap exceeds half the period duration.		Verify test parameters
13000004	Error	Required test parameters are not possible for at least one element on the module.		Verify test parameters
14000001	Error	Not enough space to insert elements on logic page.		Insert new logic sheet and reorganize the Functionblocks
14000002	Warning	Items could not be grouped.		
14000003	Error	Item is only allowed for grouping.		
14000004	Error	Maximum number of function blocks have already been created.		Simplify logic
14000005	Error	Failed to create residual memory.		Contact support
14000006	Error	Item is not allowed for grouping.		
14000007	Error	Function blocks are not compatible with the choosed CPU module.		If you want to use this CPU module, all related function blocks will be deleted.
14000008	Error	Selection cannot be grouped because there are more than 8 connections to inputs.		
14000009	Error	Selection cannot be grouped because there are more than 8 connections to outputs.		
1400000A	Error	No function blocks have been selected to group.		
15000001	Error	CRC calculation failed		Retry Contact support
15000002	Error	Report generation failed		Retry Contact support
22010140	Warning	Error in system configuration	System continues to run	Reload system configuration
220101F5	Warning	Error in system configuration	Configuration required	Reload system configuration
220101F6	Warning	Error in system configuration	Configuration required	Reload system configuration



Error No.	Error type	Logbook message	System behavior	Remedy
220101F7	Warning	Error in system configuration	Configuration required	Reload system configuration
220101F8	Warning	Error in system configuration	Configuration required	Reload system configuration
220101F9	Warning	Error in system configuration	Configuration required	Reload system configuration
220101FA	Warning	Error in system configuration	Configuration required	Reload system configuration
220101FC	Warning	Error in system configuration	Configuration required	Reload system configuration
22010226	Warning	Error in system configuration	Configuration required	Reload system configuration
22010227	Warning	Error in system configuration	Configuration required	Reload system configuration
22010228	Warning	Error in system configuration	Configuration required	Reload system configuration
22010231	Warning	Pulse period 0 must have pulse length 0.	Configuration required	Change system configuration and reload
22010232	Warning	Pulse length must be $\leq$ pulse period/2.	Configuration required	Change system configuration and reload
22010233	Warning	Impermissible test period (permissible: 0,40,200,400,600,800,1000).	Configuration required	Change system configuration and reload
22010234	Warning	Pulse length must be multiple of 4 and in range from 4 to 100ms	Configuration required	Change system configuration and reload
22010240	Warning	Maximum count of function blocks or mapping exceeded	Configuration required	Change system configuration and reload
22010241	Warning	The number of IO modules is different in the configuration	Configuration required	Change system configuration and reload
22010242	Warning	The number of Gateway modules is different in the configuration	Configuration required	Change system configuration and reload
22010243	Warning	The major version of the CPU module and project file differs.	Configuration required	Change system configuration and reload
22010244	Warning	The major version of the IO module and project file differs.	Configuration required	Change system configuration and reload
22010245	Warning	The major version of the gateway module and project file differs.	Configuration required	Change system configuration and reload
22010250	Warning	function blocks for presses are not supported by this device	Configuration required	Change system configuration and reload
2201xxxx	Warning	Error in the configuration	Configuration required	Change system configuration and reload
23010001	Warning	Processing error at I1/I2	System continues to run	
23010003	Warning	Processing error at I3/I4	System continues to run	
23010005	Warning	Processing error at I5/I6	System continues to run	
23010007	Warning	Processing error at I7/I8	System continues to run	
23010009	Warning	Processing error at I9/I10	System continues to run	

Error No.	Error type	Logbook message	System behavior	Remedy
2301000B	Warning	Processing error at I11/I12	System continues to run	
2301000D	Warning	Processing error at I13/I14	System continues to run	
2301000F	Warning	Processing error at I15/I16	System continues to run	
23010011	Warning	Processing error at IQ1/IQ2	System continues to run	
23010013	Warning	Processing error at IQ3/IQ4	System continues to run	
2301xxxx	Warning	Processing error at 2-channel input	System continues to run	
23020001	Warning	Synchronous time error I1/I2	System continues to run	
23020003	Warning	Synchronous time error I3/I4	System continues to run	
23020005	Warning	Synchronous time error I5/I6	System continues to run	
23020007	Warning	Synchronous time error I7/I8	System continues to run	
23020009	Warning	Synchronous time error I9/I10	System continues to run	
2302000B	Warning	Synchronous time error I11/I12	System continues to run	
2302000D	Warning	Synchronous time error I13/I14	System continues to run	
2302000F	Warning	Synchronous time error I15/I16	System continues to run	
23020011	Warning	Synchronous time error IQ1/IQ2	System continues to run	
23020013	Warning	Synchronous time error IQ3/IQ4	System continues to run	
2302xxxx	Warning	Synchronous time error at 2-channel input	System continues to run	
240A0000	Warning	Output error on Q1	System continues to run; affected outputs switch off	Check outputs
240A0001	Warning	Output error on Q2	System continues to run; affected outputs switch off	Check outputs
240A0002	Warning	Output error on Q3	System continues to run; affected outputs switch off	Check outputs
240A0003	Warning	Output error on Q4	System continues to run; affected outputs switch off	Check outputs
240A0004	Warning	Output error on IQ1	System continues to run; affected outputs switch off	Check outputs
240A0005	Warning	Output error on IQ2	System continues to run; affected outputs switch off	Check outputs

Error No.	Error type	Logbook message	System behavior	Remedy
240A0006	Warning	Output error on IQ3	System continues to run; affected outputs switch off	Check outputs
240A0007	Warning	Output error on IQ4	System continues to run; affected outputs switch off	Check outputs
240A0008	Warning	Output error on group Q1/Q2	System continues to run; affected outputs switch off	Check outputs
240A0009	Warning	Output error on group Q3/Q4	System continues to run; affected outputs switch off	Check outputs
240A000A	Warning	Output error on group IQ1/IQ2	System continues to run; affected outputs switch off	Check outputs
240A000B	Warning	Output error on group IQ3/IQ4	System continues to run; affected outputs switch off	Check outputs
240Axxxx	Error	Output error	System stop; voltage OFF-ON required	Check outputs
240B0001	Info	Output error at Q1/Q2 rectified	System continues to run	
240B0002	Info	Output error at Q3/Q4 rectified	System continues to run	
240B0003	Info	Output error at IQ1/IQ2 rectified	System continues to run	
240B0004	Info	Output error at IQ3/IQ4 rectified	System continues to run	
240Bxxxx	Info	Output error rectified	System continues to run	
240Dxxxx	Error	Error in system configuration	System stop; voltage OFF-ON required	Reload system configuration and restart
240Exxxx	Warning	Problem with forcing	System continues to run	Restart forcing
240Fxxxx	Warning	Problem with forcing	System continues to run	Restart forcing
2410xxxx	Warning	Problem with forcing	System continues to run	Restart forcing
2411xxxx	Warning	Problem with forcing	System continues to run	Restart forcing
2412xxxx	Warning	Problem with forcing	System continues to run	Restart forcing
2413xxxx	Warning	Problem with forcing	System continues to run	Restart forcing
2414xxxx	Warning	Problem with forcing	System continues to run	Restart forcing
2415xxxx	Warning	Problem with forcing	System continues to run	Restart forcing
2416xxxx	Warning	Connection problem	System stop	Restart
2417xxxx	Warning	Forcing time expired	System continues to run	-

Error No.	Error type	Logbook message	System behavior	Remedy
2418xxxx	Error	Internal error	System stop; voltage OFF-ON required	Restart or make complaint
2419xxxx	Warning	Error in system configuration.	System continues to run	Reload system configuration
241Axxxx	Warning	Output error	System continues to run	Check outputs
241B0001	Warning	Stuck-at-high at Q1	System continues to run	Check outputs
241B0002	Warning	Stuck-at-high at Q2	System continues to run	Check outputs
241B0003	Warning	Stuck-at-high at Q3	System continues to run	Check outputs
241B0004	Warning	Stuck-at-high at Q4	System continues to run	Check outputs
241B0005	Warning	Stuck-at-high at IQ1	System continues to run	Check outputs
241B0006	Warning	Stuck-at-high at IQ2	System continues to run	Check outputs
241B0007	Warning	Stuck-at-high at IQ3	System continues to run	Check outputs
241B0008	Warning	Stuck-at-high at IQ4	System continues to run	Check outputs
241Bxxxx	Warning	Output error	System continues to run	Check outputs
241D0001	Warning	Test pulse error at I1	System continues to run	Check cabling
241D0002	Warning	Test pulse error at I2	System continues to run	Check cabling
241D0003	Warning	Test pulse error at I3	System continues to run	Check cabling
241D0004	Warning	Test pulse error at I4	System continues to run	Check cabling
241D0005	Warning	Test pulse error at I5	System continues to run	Check cabling
241D0006	Warning	Test pulse error at I6	System continues to run	Check cabling
241D0007	Warning	Test pulse error at I7	System continues to run	Check cabling
241D0008	Warning	Test pulse error at I8	System continues to run	Check cabling
241D0009	Warning	Test pulse error at I9	System continues to run	Check cabling
241D000A	Warning	Test pulse error at I10	System continues to run	Check cabling
241D000B	Warning	Test pulse error at I11	System continues to run	Check cabling
241D000C	Warning	Test pulse error at I12	System continues to run	Check cabling
241D000D	Warning	Test pulse error at I13	System continues to run	Check cabling

Error No.	Error type	Logbook message	System behavior	Remedy
241D000E	Warning	Test pulse error at I14	System continues to run	Check cabling
241D000F	Warning	Test pulse error at I15	System continues to run	Check cabling
241D0010	Warning	Test pulse error at I16	System continues to run	Check cabling
241D0011	Warning	Test pulse error at IQ1	System continues to run	Check cabling
241D0012	Warning	Test pulse error at IQ2	System continues to run	Check cabling
241D0013	Warning	Test pulse error at IQ3	System continues to run	Check cabling
241D0014	Warning	Test pulse error at IQ4	System continues to run	Check cabling
241Dxxxx	Warning	Check of test pulse resulted in error	System continues to run	Check cabling
241Exxxx	Warning	Verification of project failed	System continues to run	Re-verification
241Fxxxx	Warning	Verification of project failed	System continues to run	Re-verification
2420xxxx	Warning	Verification of project failed	System continues to run	Re-verification
2421xxxx	Warning	Verification of project failed	System continues to run	Re-verification
2422xxxx	Warning	Verification of project failed	System continues to run	Re-verification
2423xxxx	Info	The verified project on the SD card has changed	System continues to run	
2433xxxx	Warning	Problem during fast shutoff	System continues to run	
2435xxxx	Warning	Safety mat cable break	System continues to run	Check cabling
2436xxxx	Warning	Check of a safety feature	System continues to run	Unverification of project
2437xxxx	Warning	Check of a safety feature	System continues to run	Reduce the number of forced inputs to less than or equal to 10
2438xxxx	Warning	Configuration data faulty	System continues to run	Modify project data or make complaint
2439xxxx	Error	The configuration changed during execution of the application	System stop; voltage OFF-ON required	Restart or make complaint
243Bxxxx	Warning	Configuration data faulty	System continues to run	Modify project data or make complaint
25010001	Warning	Supply voltage A1 too low	System continues to run	Supply voltage must be set correctly
25010002	Warning	Supply voltage B1 too low	System continues to run	Supply voltage must be set correctly
25010003	Warning	Supply voltage B2 too low	System continues to run	Supply voltage must be set correctly
2501xxxx	Warning	Supply voltage too low	System continues to run	Supply voltage must be set correctly

Error No.	Error type	Logbook message	System behavior	Remedy
25020001	Warning	Supply voltage A1 too high	System continues to run	Supply voltage must be set correctly
25020002	Warning	Supply voltage B1 too high	System continues to run	Supply voltage must be set correctly
25020003	Warning	Supply voltage B2 too high	System continues to run	Supply voltage must be set correctly
2502xxxx	Warning	Supply voltage too high	System continues to run	Supply voltage must be set correctly
2503xxx1	Error	Supply voltage A1 too low	System stop; voltage OFF-ON required	Supply voltage must be set correctly
2504xxx1	Error	Supply voltage A1 too high	System stop; voltage OFF-ON required	Supply voltage must be set correctly
2504xxx2	Error	Supply voltage B1 too high	System stop; voltage OFF-ON required	Supply voltage must be set correctly
2504xxx3	Error	Supply voltage B2 too high	System stop; voltage OFF-ON required	Supply voltage must be set correctly
2504xxxx	Error	Supply voltage too high	System stop; voltage OFF-ON required	Supply voltage must be set correctly
25050001	Info	Supply voltage A1 within normal range	System continues to run	
25050002	Info	Supply voltage B1 within normal range	System continues to run	
25050003	Info	Supply voltage B2 within normal range	System continues to run	
2505xxxx	Info	Supply voltage within normal range	System continues to run	
2509xxxx	Warning	External error	System continues to run	Check load current
2604xxxx	Warning	Internal/external error SBus	System continues to run	Reduce count of extension modules
2609xxxx	Warning	Error in system configuration.	System continues to run	Reload system configuration
260Axxxx	Warning	Error in system configuration.	System continues to run	Reload system configuration
260Bxxxx	Error	Too many I/O modules used	System stop; voltage OFF-ON required	Check number of modules
2733xxxx	Warning	Input discrepancy rectified	System continues to run	
28020000	Info	Values were changed	System continues to run	
2805xxxx	Warning	Communication interrupted	System continues to run	
2808xxxx	Warning	No program memory	Configuration required	Insert SD card
2809xxxx	Warning	Action not permitted	System continues to run	Execute correct action
280Axxxx	Warning	Ethernet connection too slow	System continues to run	
3409xxxx	Warning	Invalid force request	System continues to run	

Error No.	Error type	Logbook message	System behavior	Remedy
340Axxxx	Warning	Invalid force request	System continues to run	
34290003	Warning	Synchronous time error I1/I2	System continues to run	
3429000C	Warning	Synchronous time error I3/I4	System continues to run	
34290030	Warning	Synchronous time error I5/I6	System continues to run	
342900C0	Warning	Synchronous time error I7/I8	System continues to run	
3429xxxx	Warning	Dualchannel synchronous time error	System continues to run	
342A0003	Warning	Processing error at I1/I2	System continues to run	
342A000C	Warning	Processing error at I3/I4	System continues to run	
342A0030	Warning	Processing error at I5/I6	System continues to run	
342A00C0	Warning	Processing error at I7/I8	System continues to run	
342Axxxx	Warning	Processing error at 2-channel input	System continues to run	
36010001	Warning	external testpuls error at I1	System continues to run	
36010002	Warning	external testpuls error at I2	System continues to run	
36010004	Warning	external testpuls error at I3	System continues to run	
36010008	Warning	external testpuls error at I4	System continues to run	
36010010	Warning	external testpuls error at I5	System continues to run	
36010020	Warning	external testpuls error at I6	System continues to run	
36010040	Warning	external testpuls error at I7	System continues to run	
36010080	Warning	external testpuls error at I8	System continues to run	
3601xxxx	Warning	Error at external input test	System continues to run	
3602xxxx	Warning	Safety mat cable break	System continues to run	
3702xxxx	Warning	Short circuit, stuck-at-low, VCC or GND break	System continues to run	
37040003	Warning	Cross-comparison error at Q1/Q2	System continues to run	
3704000C	Warning	Cross-comparison error at Q3/Q4	System continues to run	
3704xxxx	Warning	Cross-reference error at the output	System continues to run	

Error No.	Error type	Logbook message	System behavior	Remedy
37050001	Warning	Stuck-at-high at Q1	System continues to run	
37050002	Warning	Stuck-at-high at Q2	System continues to run	
37050004	Warning	Stuck-at-high at Q3	System continues to run	
37050008	Warning	Stuck-at-high at Q4	System continues to run	
3705xxxx	Warning	Stuck-at-high at the output	System continues to run	
3801xxxx	Error	Power supply voltage error (logic voltage)	System stop; voltage OFF-ON required	
3802xxxx	Error	Power supply unit watchdog	System stop; voltage OFF-ON required	
3803xxxx	Error	Output voltage error	System stop; voltage OFF-ON required	
3806xxxx	Warning	GND break at A1 and A2	System continues to run	
3807xxxx	Warning	Supply voltage A1 too low	System continues to run	
3902xxxx	Warning	Error in system configuration	System continues to run	
3903xxxx	Warning	Error in system configuration	System continues to run	
3904xxxx	Warning	Error in system configuration	System continues to run	
3905xxxx	Warning	Error in system configuration	System continues to run	
3906xxxx	Warning	Error in system configuration	System continues to run	
3907xxxx	Warning	Error in system configuration	System continues to run	
3908xxxx	Warning	Error in system configuration	System continues to run	
3909xxxx	Warning	Error in system configuration	System continues to run	
390Axxxx	Warning	Error in system configuration	System continues to run	
390Bxxxx	Warning	Error in system configuration	System continues to run	
390Cxxxx	Warning	Error in system configuration	System continues to run	
390Dxxxx	Warning	Error in system configuration	System continues to run	
390Exxxx	Warning	Error in system configuration	System continues to run	
390Fxxxx	Warning	Error in system configuration	System continues to run	
3910xxxx	Warning	Error in system configuration	System continues to run	



Error No.	Error type	Logbook message	System behavior	Remedy
3911xxxx	Warning	Error in system configuration	System continues to run	
3945xxxx	Warning	Fast shutoff control signal faulty	System continues to run	
4102xxxx	Warning	CRC error in the configuration	System continues to run	
4103xxxx	Warning	Module type deviates	System continues to run	
4104xxxx	Warning	Module version deviates	System continues to run	
4106xxxx	Warning	Service data project not processed	System continues to run	
4302xxxx	Info	Service data project not processed	System continues to run	
4303xxxx	Info	Service data project not processed	System continues to run	
4304xxxx	Info	Service data project not processed	System continues to run	
4305xxxx	Info	Service data project not processed	System continues to run	
4306xxxx	Info	Service data project not processed	System continues to run	
4307xxxx	Info	Service data project not processed	System continues to run	
4309xxxx	Info	Service data project not processed	System continues to run	
430Bxxxx	Error	Gateway address outside permissible range	System stop; voltage OFF-ON required	
4501xxxx	Warning	Data loss in receive buffer due to very high bus load	System continues to run	
4502xxxx	Warning	CAN controller TEC or REC $\geq$ 96	System continues to run	
4503xxxx	Warning	CAN controller TEC or REC $>$ 127	System continues to run	
4504xxxx	Warning	CAN controller TEC $>$ 255	System continues to run	
4505xxxx	Warning	Transmission of a message was faulty	System continues to run	
4506xxxx	Warning	Data loss in transmit buffer due to overload	System continues to run	
4507xxxx	Error	Initialization was faulty	System stop; voltage OFF-ON required	
4508xxxx	Warning	Lifeguarding faulty	System continues to run	
4601xxxx	Error		System stop; voltage OFF-ON required	
4602xxxx	Error		System stop; voltage OFF-ON required	
4603xxxx	Error		System stop; voltage OFF-ON required	

Error No.	Error type	Logbook message	System behavior	Remedy
4604xxxx	Warning		System continues to run	
4605xxxx	Warning		System continues to run	
50xxxxxx	Warning	Modbus/TCP related error	System continues to run	
51xxxxxx	Warning	PROFINET IO related error	System continues to run	
5201xxxx	Error	Too many EtherNet/IP connections	System continues to run	
5202xxxx	Warning	Wrong EtherNet/IP data format	System continues to run	
5203xxxx	Warning	Wrong EtherNet/IP data format	System continues to run	
5204xxxx	Warning	Wrong EtherNet/IP data size	System continues to run	
5205xxxx	Warning	Wrong EtherNet/IP command	System continues to run	
5206xxxx	Warning	EtherNet/IP read error	System continues to run	
5209xxxx	Warning	Wrong EtherNet/IP data index	System continues to run	
520C00xx	Error	Wrong EtherNet/IP connection configuration	System continues to run	
520Fxxxx	Warning	EtherNet/IP timeout	System continues to run	
52xxxxxx	Warning	EtherNet/IP related error	System continues to run	
60000000	Info	Log file cleared	System continues to run	
60000001	Info	Base module firmware	System continues to run	
60000002	Info	Base module serial number	System continues to run	
60000003	Info	Project file read	System continues to run	
63xxxxxx	Warning	USB error	System continues to run	
640A0001	Warning	SD card cannot be read	Configuration required	
64xxxxxx	Warning	File system error on SD card	Configuration required	
65xxxxxx	Warning	Ethernet error	System continues to run	
680A0001	Warning	Supply voltage A1 too low	Configuration required	
690Fxxxx	Warning	Communication interrupted	-	
6A020001	Warning	Communication (Ethernet/USB) disrupted	System continues to run	
6A04xxxx	Warning	Communication (Ethernet/USB) disrupted	System continues to run	

Error No.	Error type	Logbook message	System behavior	Remedy
6Axxxxxx	Warning	Communication error (Ethernet/USB)	System continues to run	
6B03xxxx	Warning	Project file faulty	Configuration required	
6B04xxxx	Warning	Project file faulty	Configuration required	
6Bxxxxxx	Warning	File error	Configuration required	