

# Axioline F SBT V3 configuration on a KV-8000/7000 series KEYENCE controller with KV STUDIO

Quick start guide



RSPSupply - 1-888-532-2706 - <https://www.RSPSupply.com>  
See the product details here

## Quick start guide

# Axioline F SBT V3 configuration on a KV-8000/7000 series KEYENCE controller with KV STUDIO

UM QS EN AXL F SBT V3 KEYENCE, Revision 01

2019-09-30

---

This user manual is valid for:

Designation	From HW/FW/FW version	Order No.
AXL F LPSDO8/3 1F	00/100	2702171
AXL F SSDI8/4 1F	01/200	2702263
AXL F SSDO8/3 1F	01/200	2702264
IB IL 24 LPSDO 8 V3-PAC	00/100/100	2701625
IB IL 24 PSDI 8-PAC	00/201	2985688
IB IL 24 PSDI 16-PAC	00/100	2700994
IB IL 24 PSDO 8-PAC	01/200/100	2985631
IB IL 24 PSDOR 4-PAC	01/200/100	2985864
IB IL 24 PSDO 4/4-PAC	01/201/100	2916493

108292\_en\_01

## Table of contents

1	For your safety .....	5
	1.1	Marking of warning notes.....5
	1.2	Qualification of users .....
		5
2	Introduction .....	6
	2.1	Purpose of this user manual .....
		6
	2.2	Prerequisites.....
		6
	2.3	Additional documentation .....
		7
	2.4	Safety hotline.....
		7
3	Overview of the integration of the SafetyBridge Technology V3 system .....	8
4	Example project: 2-channel emergency stop monitoring .....	10
	4.1	Download and installation of the Phoenix Contact software .....
		10
	4.2	Hardware installation .....
		11
	4.2.1	Setting the device DIP switches .....
		11
	4.2.2	Mounting and wiring the bus configuration .....
		12
	4.3	Configuring the safety logic in SAFECONF .....
		13
	4.3.1	Creating a new project .....
		13
	4.3.2	Configuring and parameterizing the hardware structure .....
		15
	4.3.3	Configuring the safety function .....
		18
	4.3.4	Exporting the configuration and parameter data record .....
		20
	4.4	KV STUDIO project configuration .....
		21
	4.4.1	Creating a project and importing the device description file .....
		21
	4.4.2	Configuring the hardware structure .....
		21
	4.4.3	Integrating function blocks for SafetyBridge Technology V3 .....
		23
	4.4.4	Creating the SBT program in KV STUDIO .....
		24
	4.4.5	Importing the configuration and parameter data record as a BIN file ....
		28
	4.4.6	Complete example project in KV STUDIO .....
		29
	4.4.7	Compile project and load it into the controller .....
		29
	4.5	Startup.....
		30
	4.6	Online configuration and connection establishment .....
		31
	4.6.1	Configuration of the KV-XLE02 module as a Modbus TCP server .....
		31
	4.6.2	Configuration of a Modbus TCP connection in SAFECONF .....
		35
	4.6.3	Displaying online values in SAFECONF .....
		37

A	Flowchart for starting up and testing the application .....	39
B	Description of the function blocks for SafetyBridge Technology V3 .....	40
	B 1 SBT_V3_Operate function block .....	42
	B 2 SBT_V3_ReadDev function block .....	47
	B 3 SBT_V3_WriteDev function block .....	48
	B 4 SBT_V3_ProjHeader function block .....	49
	B 5 SBT_V3_BinFile function block .....	50
	B 6 SBT_V3_TransTime function block .....	52
	B 7 SBT_V3_CrossComm function block .....	54
C	Revision history .....	55

# 1 For your safety

Read this user manual carefully and keep it for future reference.

## 1.1 Marking of warning notes



This symbol indicates hazards that could lead to personal injury.

There are three signal words indicating the severity of a potential injury.

### **DANGER**

Indicates a hazard with a high risk level. If this hazardous situation is not avoided, it will result in death or serious injury.

### **WARNING**

Indicates a hazard with a medium risk level. If this hazardous situation is not avoided, it could result in death or serious injury.

### **CAUTION**

Indicates a hazard with a low risk level. If this hazardous situation is not avoided, it could result in minor or moderate injury.



This symbol together with the **NOTE** signal word warns the reader of actions that might cause property damage or a malfunction.



Here you will find additional information or detailed sources of information.

## 1.2 Qualification of users

The use of products described in this user manual is oriented exclusively to qualified application programmers and software engineers. The users must be familiar with the relevant safety concepts of automation technology as well as applicable standards and other regulations.

## 2 Introduction

### 2.1 Purpose of this user manual

This quick start guide uses an example project to describe how to integrate SafetyBridge Technology V3 modules into a KV-8000/7000 series KEYENCE controller with EtherNet/IP™ .

### 2.2 Prerequisites

#### Knowledge

Knowledge of the following is required:

- Components used in the application
- KV STUDIO software used
- Microsoft Windows operating system

#### Hardware

The following hardware is required in order to start up the example system:

Designation	As of HW/FW	Order No.
AXL F LPSDO8/3 1F (logic module)	00/100	2702171
AXL F SSDI8/4 1F (input module)	00/200	2702263
AXL F DI8/1 DO8/1 1H (I/O module)	-	2701916
AXL F BK EIP (EtherNet/IP™ bus coupler)	-	2688394
KEYENCE KV-7500 controller with KV-XLE02 module	-	-

- Programming device/PC
- Other components: emergency stop button, external reset button, signal lamp, contactor (optional)

#### Software



The Phoenix Contact software can be found in the download area for the specified product at [phoenixcontact.net/products](http://phoenixcontact.net/products).

The following software is required in order to start up the example system:

Designation	Order No.
SAFECONF V2.92 or later	2986119
Integration package for SafetyBridge Technology V3	2702171
KEYENCE KV STUDIO V9.41 or later	-

- Other software: Microsoft Windows

### 2.3 Additional documentation

Please refer to the documentation for the software used, the components used in the application, and the function blocks used.

The documentation for the SafetyBridge Technology V3 modules used must be strictly observed.

Description	Type	Order No.
User manual: Axioline F module with integrated safety logic and safe digital outputs	UM EN AXL F LPSDO8/3 1F	2702171
User manual: Axioline F module with safe digital inputs	UM EN AXL F SSDI8/4 1F	2702263



The documentation for Phoenix Contact devices can be found in the download area for the specified product at [phoenixcontact.net/products](https://phoenixcontact.net/products).

### 2.4 Safety hotline

Should you have any technical questions, please contact our 24-hour hotline.

- Phone: +49 5281 9462777
- E-mail: [safety-service@phoenixcontact.com](mailto:safety-service@phoenixcontact.com)

### 3 Overview of the integration of the SafetyBridge Technology V3 system

**Safety with the SafetyBridge Technology V3 system**

Within a SafetyBridge Technology V3 system, safety can only be ensured by using the modules of this system (AXL F LPSDO8/3 1F and 1 to 16 satellites). None of the other components in the overall system are safety-related components. Errors at non-safety-related components or errors during integration of the SafetyBridge Technology V3 system are reliably detected by the SafetyBridge Technology V3 system components. These errors only reduce the system availability but not the system safety.



**No safety controllers are required for the implementation of safety functions.**

Table 3-1 Integration of a SafetyBridge Technology V3 island

Step	Process	See...
<b>1</b>	<b>Download and installation of the Phoenix Contact software</b> (not safety-related)	
	<ul style="list-style-type: none"> <li>- Download and install SAFECONF configuration software</li> <li>- Download device description file for bus coupler</li> <li>- Download and install integration package for SafetyBridge Technology V3</li> </ul>	<p><a href="#">page 10</a></p> <p><a href="#">page 10</a></p> <p><a href="#">page 10</a></p>
<b>2</b>	<b>Hardware installation</b> (not safety-related)	
	<ul style="list-style-type: none"> <li>- Set device DIP switches</li> <li>- Mount and wire bus configuration</li> </ul>	<p><a href="#">page 11</a></p> <p><a href="#">page 12</a></p> <p>User documentation for the devices</p>
<b>3</b>	<b>Configure safety logic in SAFECONF</b> (safety-related)	
	<ul style="list-style-type: none"> <li>- Create new project, assign safety island number</li> <li>- Configure the hardware structure</li> <li>- Parameterize I/O channels</li> <li>- Configure safety function</li> <li>- Export configuration and parameter data record</li> </ul>	<p><a href="#">page 13, page 15</a></p> <p><a href="#">page 15</a></p> <p><a href="#">page 16</a></p> <p><a href="#">page 18</a></p> <p><a href="#">page 20</a></p> <p>SAFECONF online help</p>
<b>4</b>	<b>Configuring a KV STUDIO project for the KV-7500 controller</b> (not safety-related)	
	<ul style="list-style-type: none"> <li>- Create project and import device description file</li> <li>- Configure the hardware structure</li> <li>- Integrate function blocks for SafetyBridge Technology V3</li> <li>- Create the SBT program in KV STUDIO</li> </ul>	<p><a href="#">page 21</a></p> <p><a href="#">page 21</a></p> <p><a href="#">page 23</a></p> <p><a href="#">page 24</a></p>



Table 3-1 Integration of a SafetyBridge Technology V3 island

Step	Process	See...
	<ul style="list-style-type: none"> <li>– Import configuration and parameter data record</li> <li>– Compile project and load it into the controller</li> </ul>	<a href="#">page 28</a> <a href="#">page 29</a>
5	<b>Startup and overall safety validation</b> (safety-related)	<a href="#">page 30</a>

System overview:

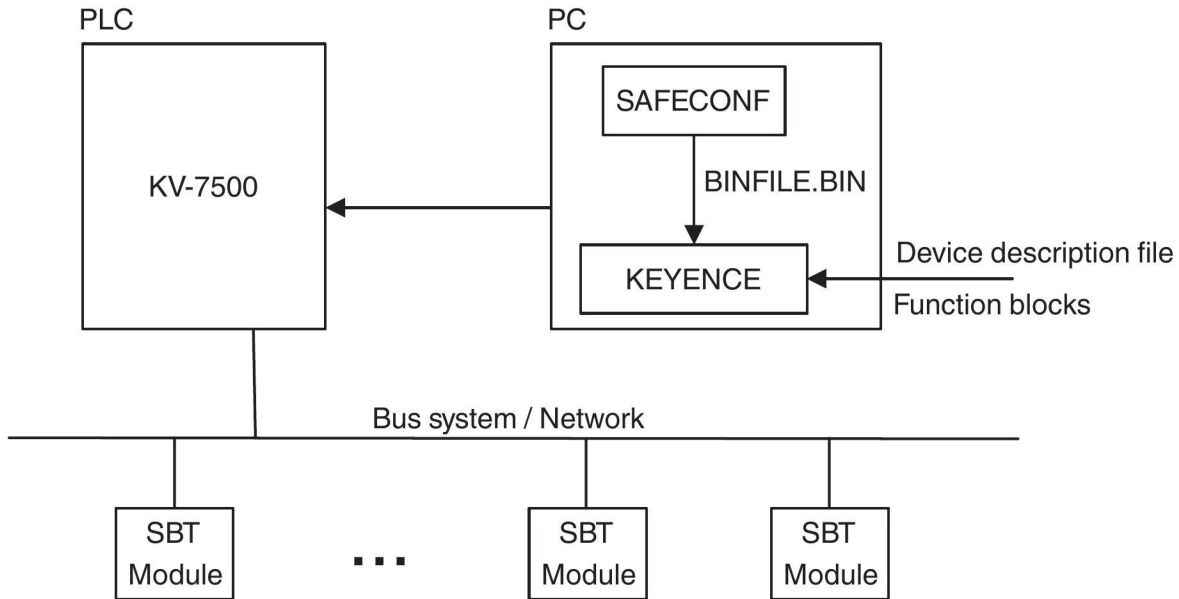


Figure 3-1 System overview of SafetyBridge Technology V3

## 4 Example project: 2-channel emergency stop monitoring

### 4.1 Download and installation of the Phoenix Contact software



The Phoenix Contact software can be found in the download area for the specified product at [phoenixcontact.net/products](http://phoenixcontact.net/products).  
 Make sure that you always use the latest version of the integration package and the function blocks for the KV STUDIO software version.  
 See “Software” on page 6.

**SAFECONF**

1. Download the SAFECONF configuration software and install the software (Order No. 2986119).

**Device description file (EDS)**

2. Download the EDS file for the bus coupler (Order No. 2688394).

**SBT V3 integration package**

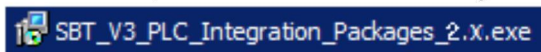
3. Download the integration package for SafetyBridge Technology V3 (Order No. 2702171).

Software

	Description	Language	Revision
<input checked="" type="checkbox"/>	[exe, 123 MB] <b>Software</b> SafetyBridge technology integration package for controllers from Phoenix Contact, Rockwell and Siemens (S7-1200 from CPU 1214C, S7-1500, S7-300), Schneider as well as CODESYS-based controllers. SBT_V3_PLC_Integration_Packages_2.0.exe	International	2.0

Figure 4-1 Integration package in the download area for Order No. 2702171

4. Install the integration package as follows:
  - Run setup and select “KEYENCE” during installation.



Please make a note of where the example projects are installed, as you will need this information later when you open the example projects in KV STUDIO.

## 4.2 Hardware installation

### 4.2.1 Setting the device DIP switches

To implement the example project, make the following settings at the DIP switches of the SafetyBridge Technology V3 modules before installing the bus configuration.

Table 4-1 Setting the DIP switches

	CM		Island number					Satellite number				
	Operat- ing mode	Re- served	SafetyBridge Technology V3 address: 32 <sub>dec</sub> (20 <sub>hex</sub> )									
DIP switch	11	10	9	8	7	6	5	4	3	2	1	0
Setting for AXL F LPSD08/3 1F	off	on	0	0	0	0	1	0	0	0	0	0
			1 <sub>dec</sub>					0 <sub>dec</sub>				

	CM		Island number					Satellite number				
	Operat- ing mode	Re- served	SafetyBridge Technology V3 address: 33 <sub>dec</sub> (21 <sub>hex</sub> )									
DIP switch	11	10	9	8	7	6	5	4	3	2	1	0
Setting for AXL F SSDI8/4 1F	off	on	0	0	0	0	1	0	0	0	0	1
			1 <sub>dec</sub>					1 <sub>dec</sub>				



You can display the complete DIP switch setting in the SAFECONF configuration software by right-clicking on the module and selecting the "Display address switch" command. See Figure 4-10 on page 16.

### 4.2.2 Mounting and wiring the bus configuration



Refer to the user documentation for the devices.  
See "Additional documentation" on page 7.

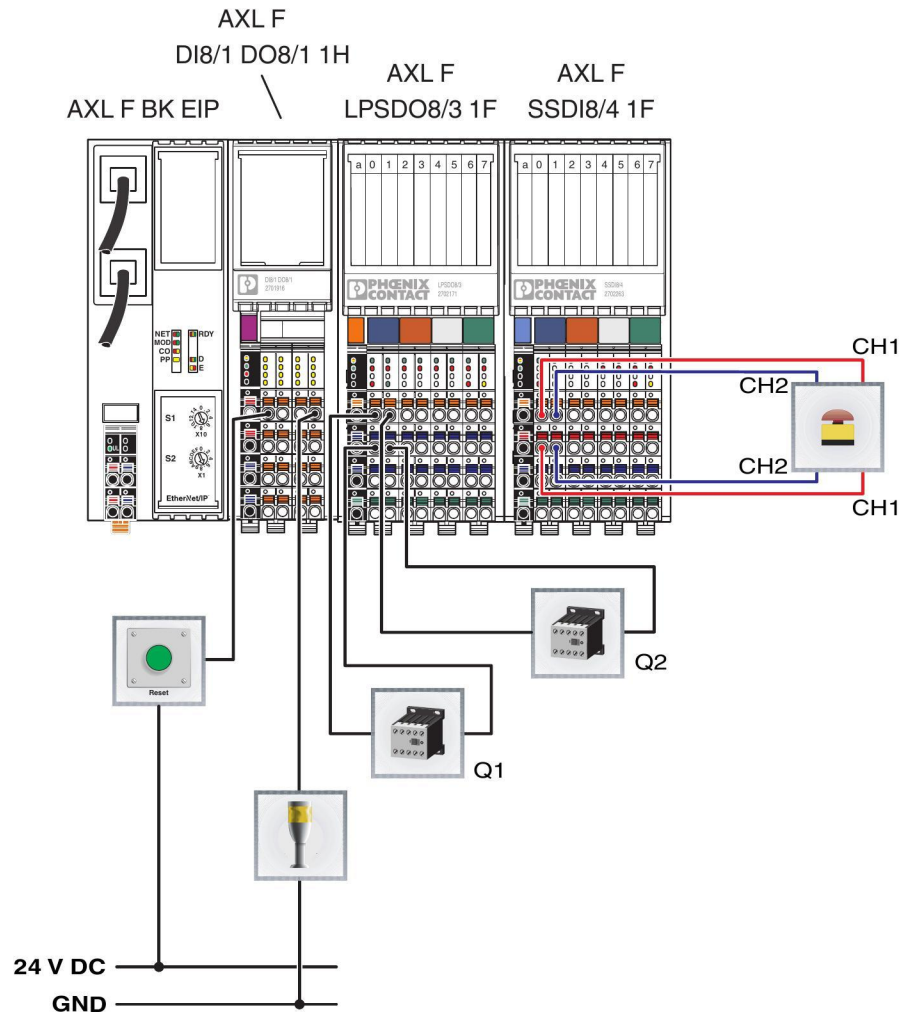


Figure 4-2 Bus configuration for example project

1. After setting the DIP switches on the modules, mount the bus configuration as illustrated.
2. Connect the power supply for the bus coupler and I/O modules in accordance with the corresponding user documentation.
3. Connect channel 1 of the emergency stop button to terminal points 00 and 10 and channel 2 to terminal points 01 and 11 of the SSDI8/4 module.
4. Connect the external reset button to the AXL F DI8/1 DO8/1 1H module.
5. Connect a signal lamp to the AXL F DI8/1 DO8/1 1H module.
6. Optional: connect contactors to terminal points 00 and 10, as well as to 01 and 11 of the LPSDO8/3 module.

### 4.3 Configuring the safety logic in SAFECONF



If you have any questions about SAFECONF, please refer to the online help for the software.

#### 4.3.1 Creating a new project

- Open the SAFECONF software.
- Create a new project with the Project Wizard. To do this, select “File, New Project”.
- Specify the name and storage location for the project.



Do not use spaces, dashes or special characters. Note the name and storage location for the project as you will need this information later.

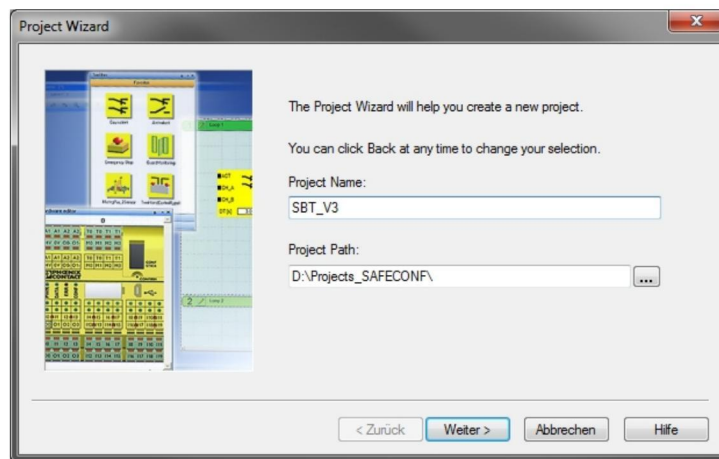


Figure 4-3 Creating the project name and path

#### Select master device

- Select the AXL F LPSDO8/3 1F master device.

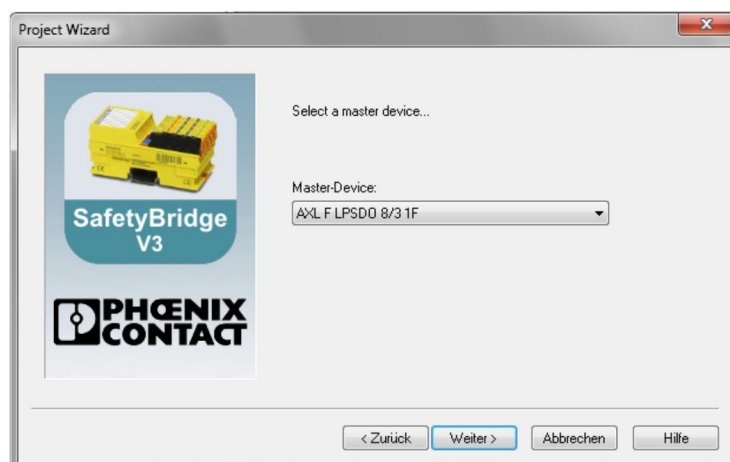


Figure 4-4 Selecting AXL F LPSDO8/3 1F

Select file format

- Select “Binary file” as the file format in which the configuration and parameter data record is to be output.

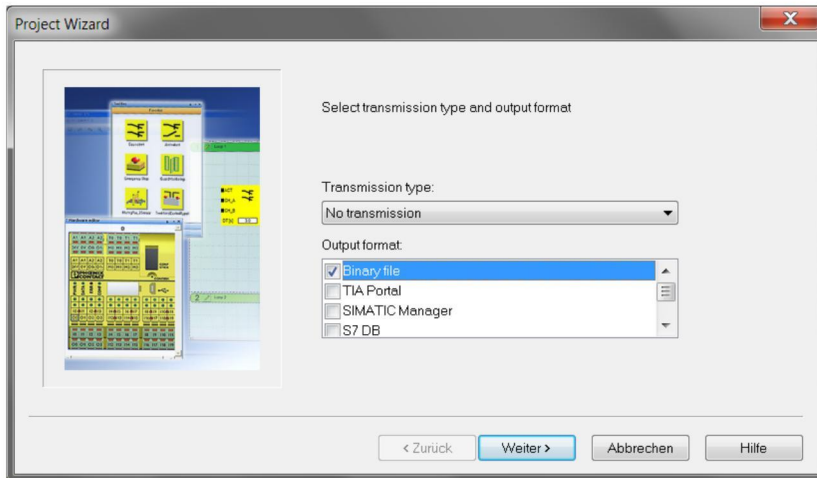


Figure 4-5 Selecting the output format

Enter project description

- Enter a description of the project.



Use a maximum of four characters for the description and version.

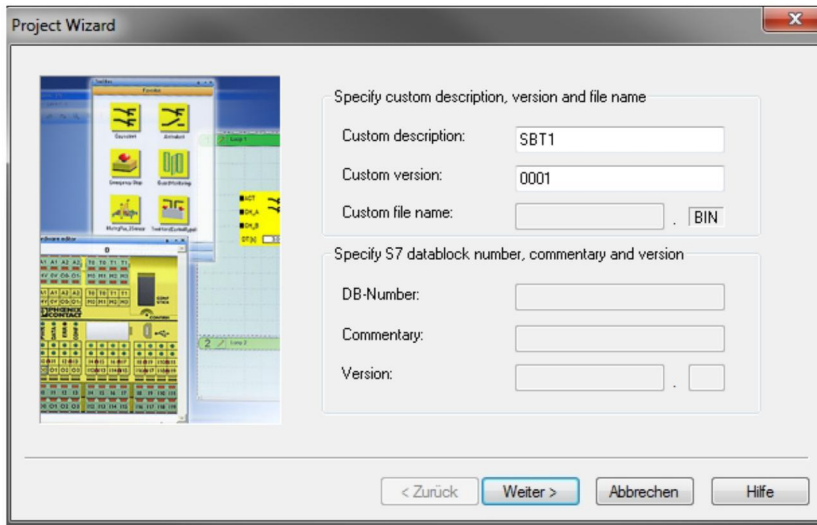


Figure 4-6 Describing the project

- Click on “Finish” to complete the project creation process.

**Assign safety island number**

When the project is completed, a window opens prompting you to enter the number for the safety island you are configuring.

- Enter an island number (1 in the example).

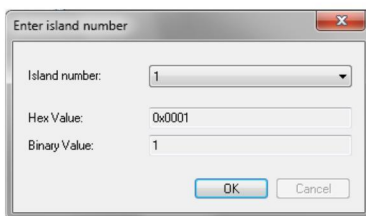


Figure 4-7 Specifying the island number

**Specify password**

- Specify a password of at least six characters for the project (123456 in the example).



Figure 4-8 Specifying a password

**4.3.2 Configuring and parameterizing the hardware structure**

- Configure the hardware structure.  
To do this, use drag and drop to move the AXL F SSDI8/4 1F module from the “Hardware” toolbox to the hardware editor.

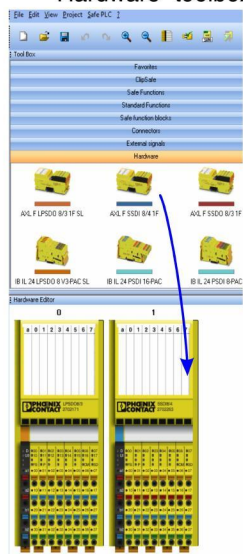


Figure 4-9 Hardware configuration



The corresponding satellite number is displayed via the module. You can display the complete DIP switch setting by right-clicking on the module and selecting "Display address switch".

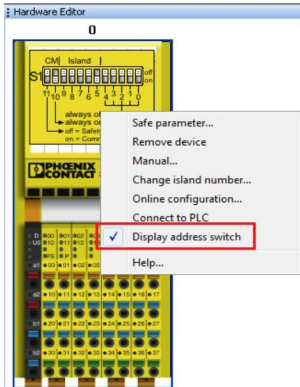


Figure 4-10 Displaying the DIP switch

**Parameterize I/O channels**

There are two options for parameterizing the input and output channels of the modules:

**Option 1:**

In the hardware editor, double-click on the module. This opens the window for parameterizing the entire module.

**Option 2:**

In the hardware editor, double-click on a terminal point. This opens the window for parameterizing the selected terminal point.

- Parameterize the output channels of the LPSDO8/3 module as illustrated (double-click on the module to parameterize).

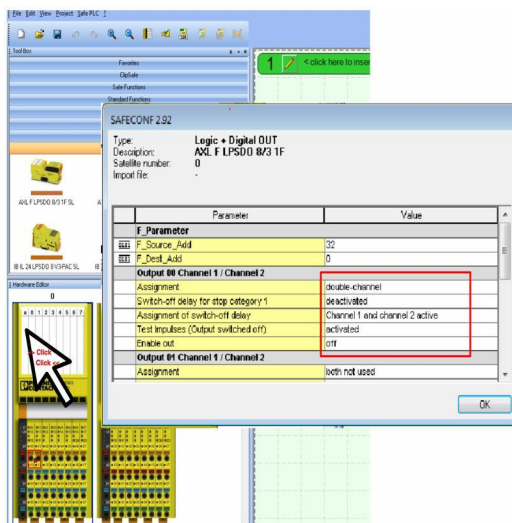


Figure 4-11 Parameterization of the LPSDO8/3



- Parameterize the input channels of the SSDI8/4 module (double-click on the module to parameterize).

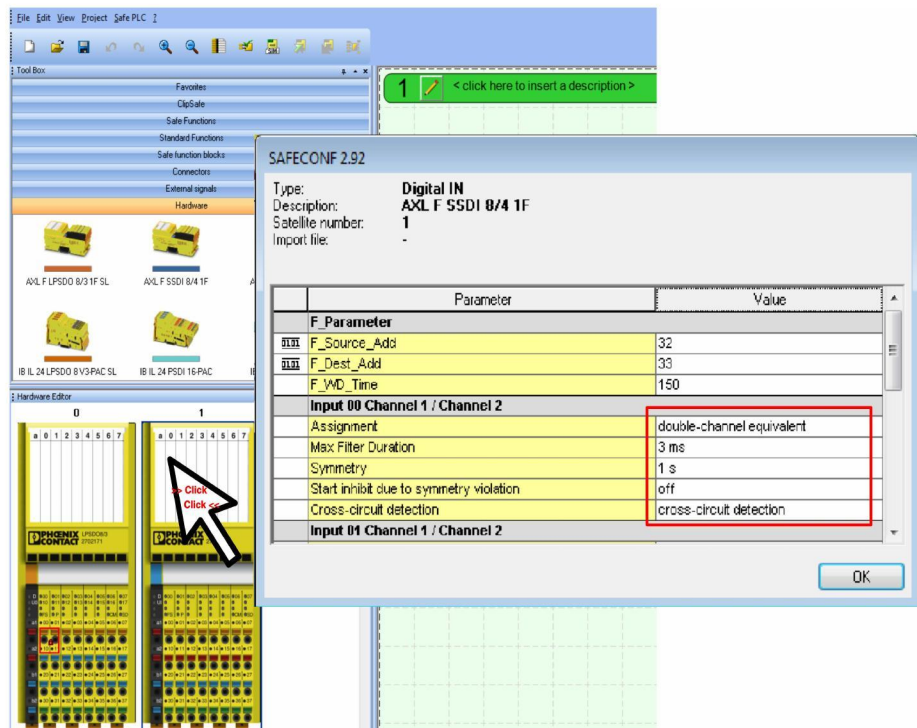


Figure 4-12 Parameterization of the SSDI8/4



Inputs or outputs parameterized for two-channel operation are indicated by a padlock symbol on the module in the hardware editor.

However, the input and output signals are only displayed in single-channel form in the connection editor, even if they are parameterized for two-channel operation. See Figure 4-14 on page 18.

### 4.3.3 Configuring the safety function

#### Comment function



You can add comments to both function blocks and signals in SAFECNF. Please refer to the online help for the software.

#### Insert function blocks

- Configure the safety function. To do this, use drag and drop to move the blocks and signals from the corresponding toolboxes to the connection editor.

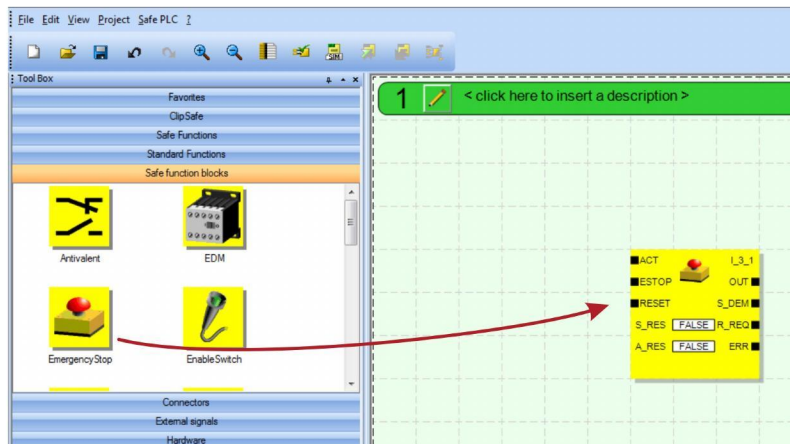


Figure 4-13 Inserting a function block from the “Safe function blocks” toolbox

#### Insert safe inputs and outputs

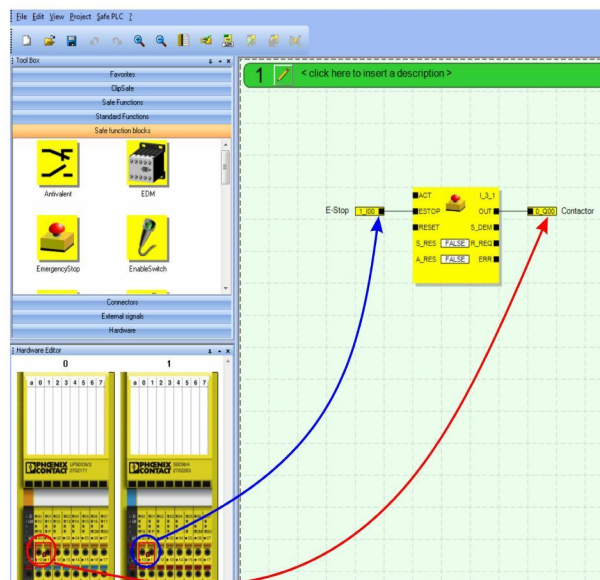


Figure 4-14 Inserting safe inputs and outputs from the hardware editor



When you use drag and drop to place the safety module terminal point directly onto a function block input or output, the connecting line is created automatically.

Insert external signals

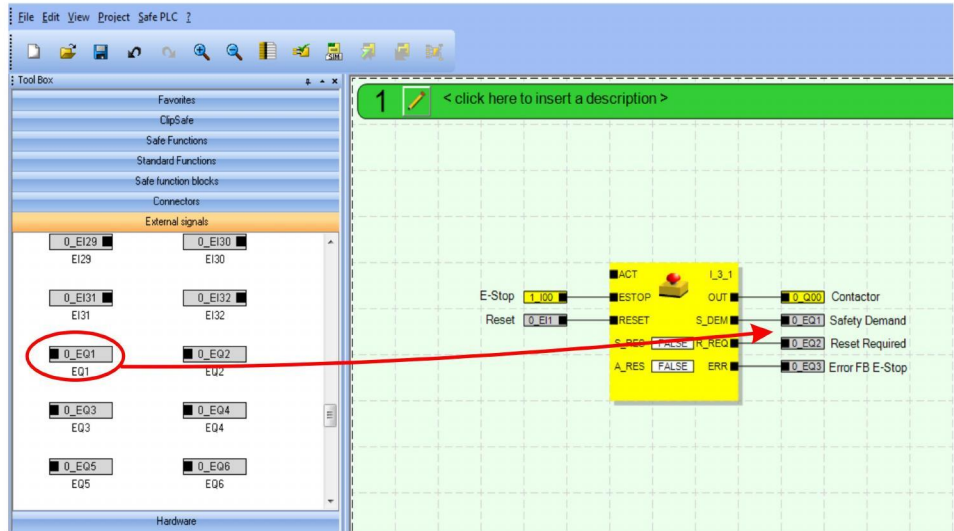
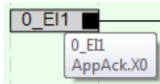


Figure 4-15 Inserting external signals from the “External signals” toolbox



Move your mouse over an external signal to display the corresponding tool tip.



Insert safe functions

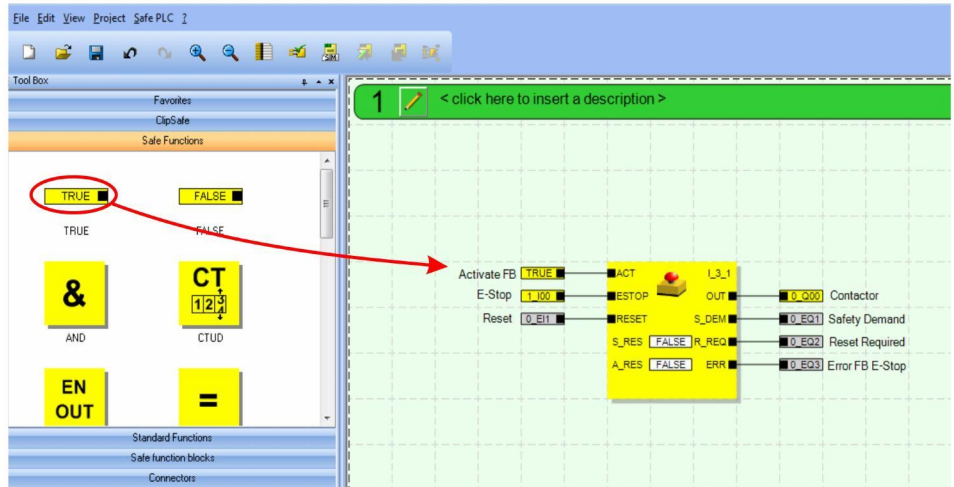


Figure 4-16 Inserting a safe function from the “Safe Functions” toolbox

### 4.3.4 Exporting the configuration and parameter data record

#### Check project

- Check the project.  
To do this, select the “Project, Check Project” command or confirm by clicking on the corresponding button.



A message window opens displaying the progress of the check. Once the check is complete, the amount of program memory used by the program is displayed.

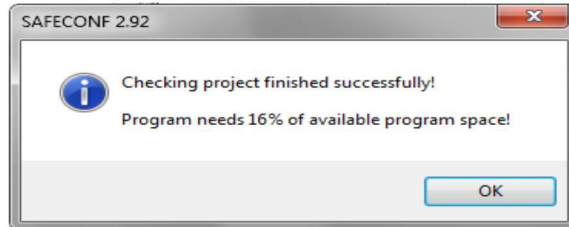


Figure 4-17 Program memory used

#### BIN file for KV STUDIO

If the check is completed without errors, the configuration and parameter data record is created as a BIN file. This is saved in the path that you have entered for the project (see [Figure 4-3 on page 13](#)) in the “FileOutput” folder.

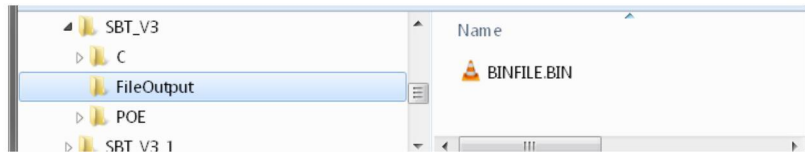


Figure 4-18 BIN file in the “FileOutput” folder

## 4.4 KV STUDIO project configuration



If you have any questions about KV STUDIO, please refer to the online help for the software.

### 4.4.1 Creating a project and importing the device description file

#### Import an EDS file

- Create a new project in KV STUDIO or use one of the provided example projects.
- Import the device description file for the bus coupler (AXL\_F\_BK\_EIP.eds in the example) into the device repository of KV STUDIO.

### 4.4.2 Configuring the hardware structure

#### Insert bus coupler

- Select the AXL F BK EIP bus coupler from the “Unit list” and use drag-and-drop to add it under the EIP master of the KV-7500 controller.

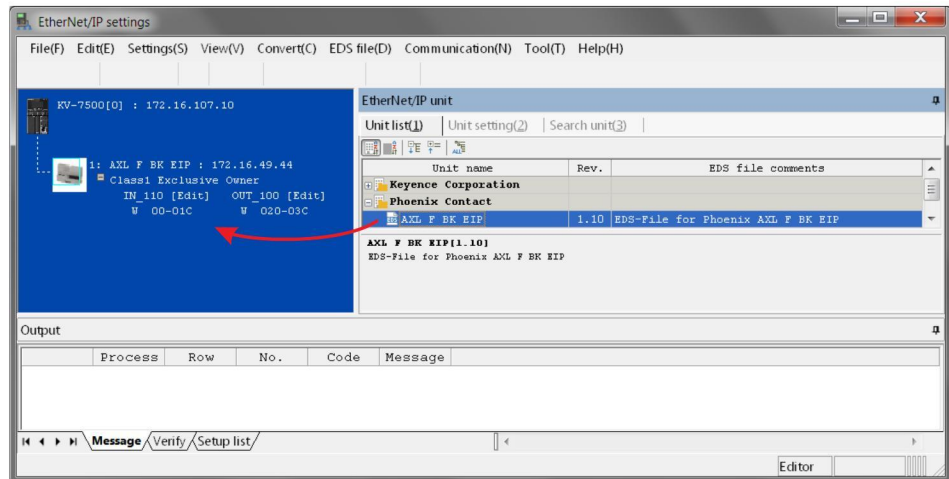


Figure 4-19 Add bus coupler AXL F BK EIP in “Unit configuration”



The KV-7500 port can be used, as in this example, but for maximum performance KEYENCE recommends connecting the SBT modules to the KV-XLE02 module.



Always use the most recent device description file for the bus coupler.

**EIP connection parameters**

- Set the EIP connection parameters for the bus coupler AXL F BK EIP according to Figure 4-20. Reserve 29 words for both the input area and the output area.

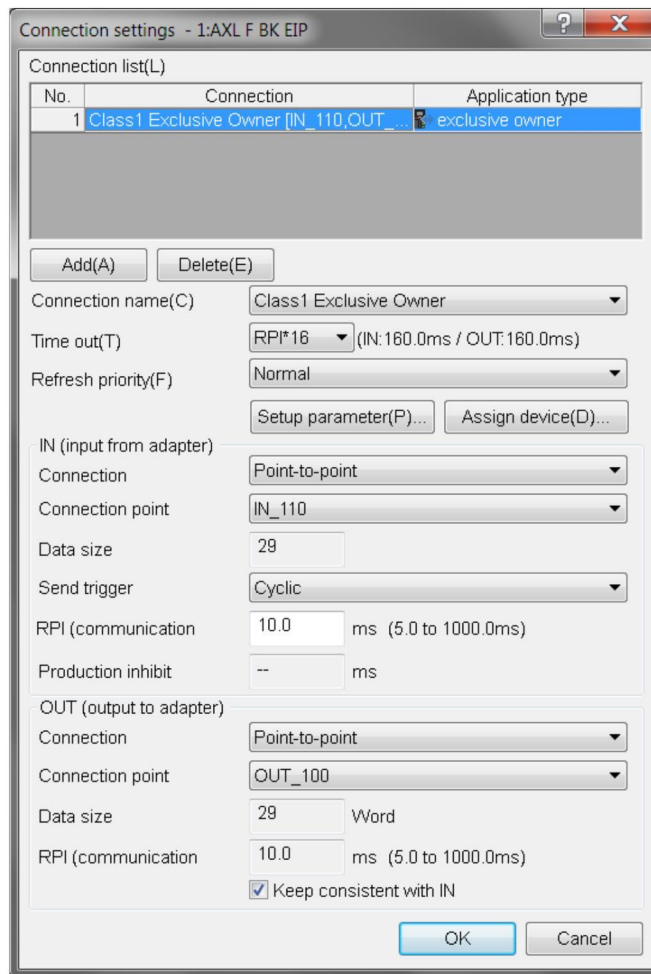


Figure 4-20 Set connection parameters for bus coupler AXL F BK EIP



For further use in the KV STUDIO project, you will require the addresses and offsets created during configuration for the individual SBT modules within the EtherNet/IP™ station. See Table 4-2.

Table 4-2 Addresses and offsets within the EtherNet/IP™ station

Module	IN Data size (number of I/O words)	IN Base addr. (offset value)	OUT Data size (number of I/O words)	OUT Base addr (offset value)
AXL F BK EIP	0	0	0	0
DI8/DO8	1	0	1	0
LPSD08/3	24	1	24	1
SSDI8/4	4	25	4	25
Total	29	-	29	-

### 4.4.3 Integrating function blocks for SafetyBridge Technology V3

#### Integrate the library from the integration package

The example project contains all of the function blocks necessary for handling the SafetyBridge Technology V3.

- Open the example project “SBT\_V3\_Lib\_V\_1\_0\_Demo\_AXL”. See [Figure 4-21](#).
- Copy the function blocks into your project or directly or use the appropriate example project.

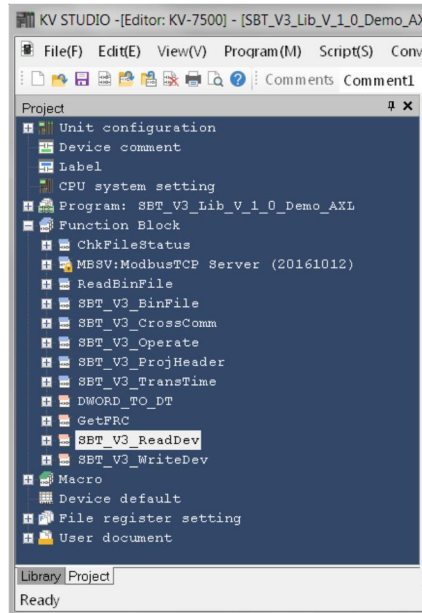


Figure 4-21 Integrating the library

#### Function blocks

#### Corresponding function blocks for SafetyBridge Technology V3:

- SBT\_V3\_BinFile
- SBT\_V3\_CrossComm
- SBT\_V3\_Operate
- SBT\_V3\_ProjectHeader
- SBT\_V3\_TransTime
- SBT\_V3\_ReadDev
- SBT\_V3\_WriteDev



A description of the function blocks and data types is provided in Appendix B “Description of the function blocks for SafetyBridge Technology V3”. Please also use the online help in KV STUDIO.

### 4.4.4 Creating the SBT program in KV STUDIO

To insert and connect the required SafetyBridge function blocks in your project, proceed as follows:

**Function block**  
**SBT\_V3\_Operate**

- Insert the “SBT\_V3\_Operate” function block in the program.
- Connect the function block as shown in [Figure 4-22](#).

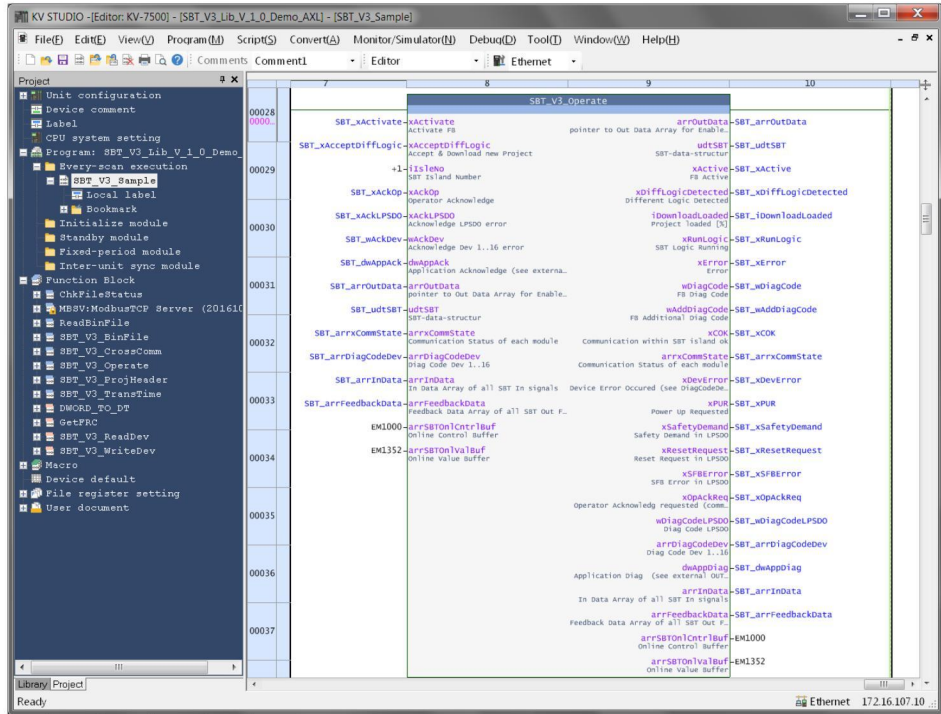


Figure 4-22 Function block SBT\_V3\_Operate



Function block  
SBT\_V3\_ReadDev



An instance of the “SBT\_V3\_ReadDev” function block is required for each SafetyBridge module.

Call the instances of the “SBT\_V3\_ReadDev” function block **before** the “SBT\_V3\_Operate” function block.

- Insert an instance of the “SBT\_V3\_ReadDev” function block in the program for the LPSDO8/3 module.
- Insert an instance of the “SBT\_V3\_ReadDev” function block in the program for the SSDI8/4 module.
- Connect the function blocks as shown in Figure 4-23.



Make sure the start addresses (arrInBuffDevX) are correctly assigned to the corresponding SBT components, see Table 4-2.

In the example: W01 for the LPSDO8/3 and W019 for the SSDI8/4.

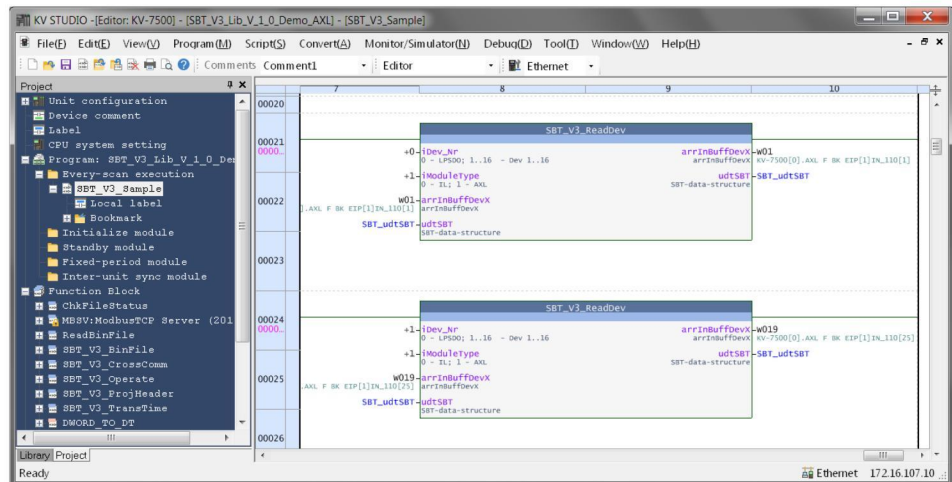


Figure 4-23 Function block SBT\_V3\_ReadDev

Function block  
SBT\_V3\_WriteDev



An instance of the “SBT\_V3\_WriteDev” function block is required for each SafetyBridge module.  
Call the instances of the “SBT\_V3\_WriteDev” function block **after** the “SBT\_V3\_Operate” function block.

- Insert an instance of the “SBT\_V3\_WriteDev” function block in the program for the LPSDO8/3 module.
- Insert an instance of the “SBT\_V3\_WriteDev” function block in the program for the SSDI8/4 module.
- Connect the function blocks as shown in Figure 4-24.



Make sure the start addresses (arrInBuffDevX) are correctly assigned to the corresponding SBT components, see Table 4-2.  
In the example: W021 for the LPSDO8/3 and W039 for the SSDI8/4.

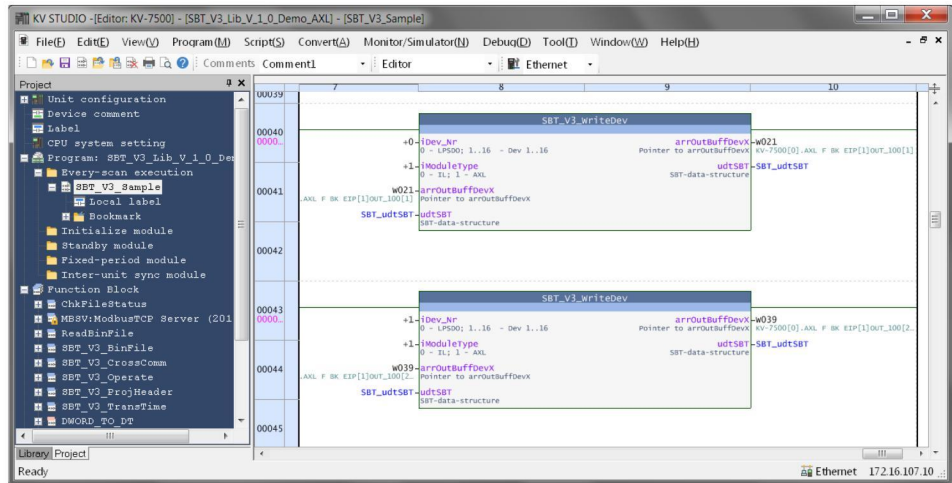


Figure 4-24 SBT\_V3\_WriteDev function blocks

**SBT\_V3\_ProjHeader function block**

- Insert the “SBT\_V3\_ProjHeader” function block in the program.
- Connect the function blocks as shown in Figure 4-25.
- Link the “udtSBT” structure variable to the same variable as with the “SBT\_V3\_Operate” function block.

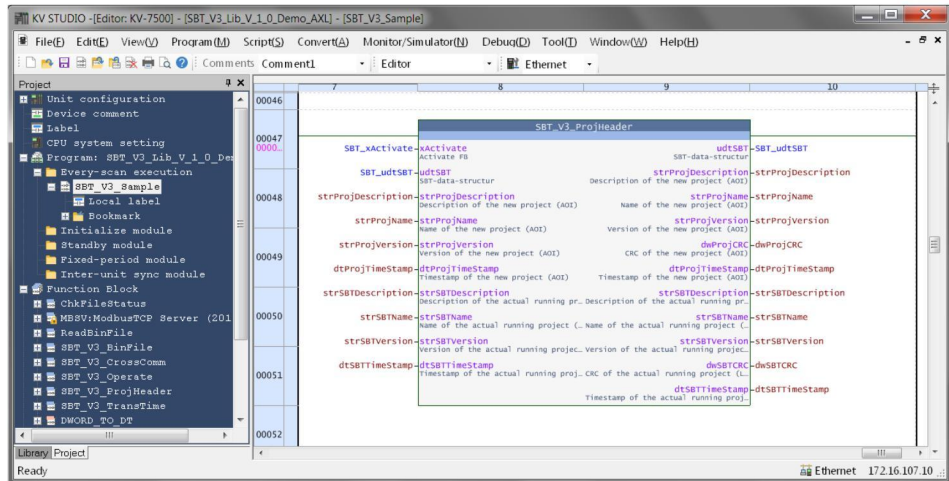


Figure 4-25 SBT\_V3\_ProjHeader function blocks

**SBT\_V3\_TransTime function block**

- Insert the “SBT\_V3\_TransTime” function block in the program.
- Connect the function blocks as shown in Figure 4-26.
- Link the “udtSBT” structure variable to the same variable as with the “SBT\_V3\_Operate” function block.

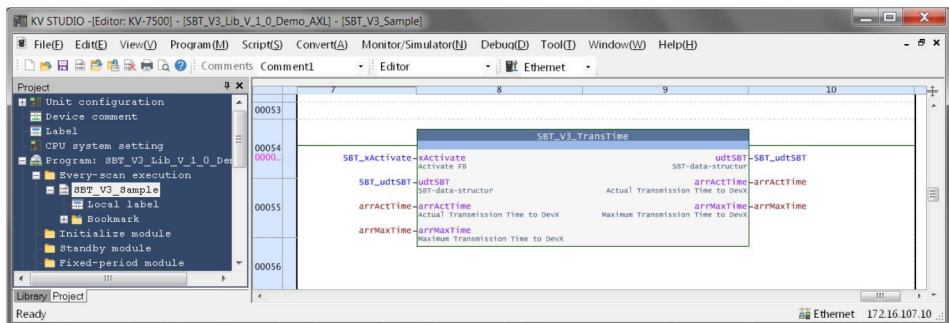


Figure 4-26 SBT\_V3\_TransTime function blocks

### 4.4.5 Importing the configuration and parameter data record as a BIN file

**Save BIN files on KV controller**

In “Exporting the configuration and parameter data record” on page 20, the **BIN file** was created and saved in the “FileOutput” folder under the previously specified project path (see Figure 4-18 on page 20).

- Open the KV STUDIO project.
- Copy the BIN file into the “userdoc directory” of the CPUMEM area of your specific project.

In the example: .../SBT\_V3\_Lib\_V\_1\_0\_Demo\_AXL/1\_CPUMEM/userdoc

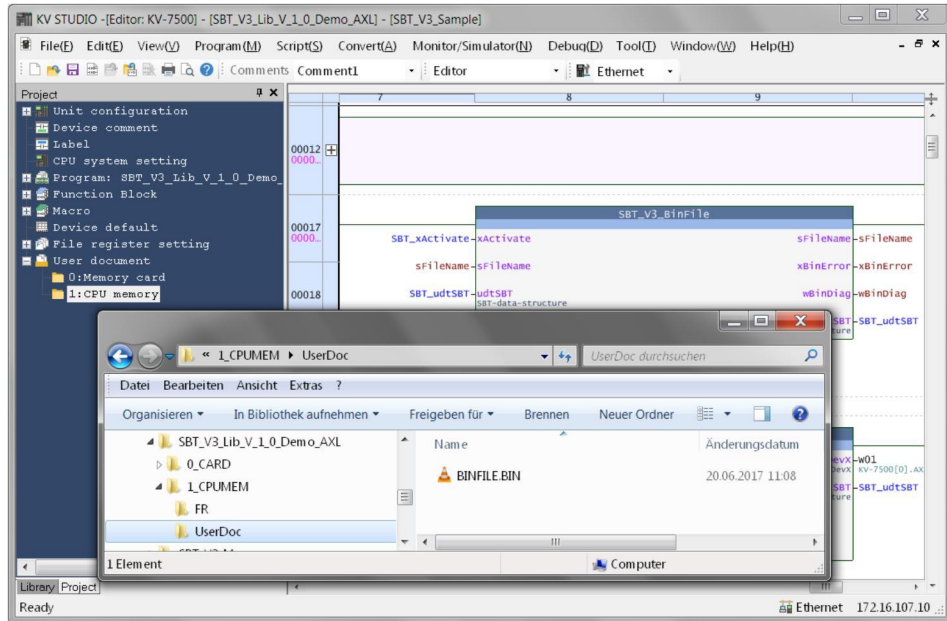


Figure 4-27 Copying the BIN file to the KV controller

**Function block SBT\_V3\_BinFile**

- Insert the “SBT\_V3\_BinFile” function block in the program.
- Connect the function block as shown in Fig. 3-28.
- Link the “udtSBT” structure variable to the same variable as for the “SBT\_V3\_Operate” function block.

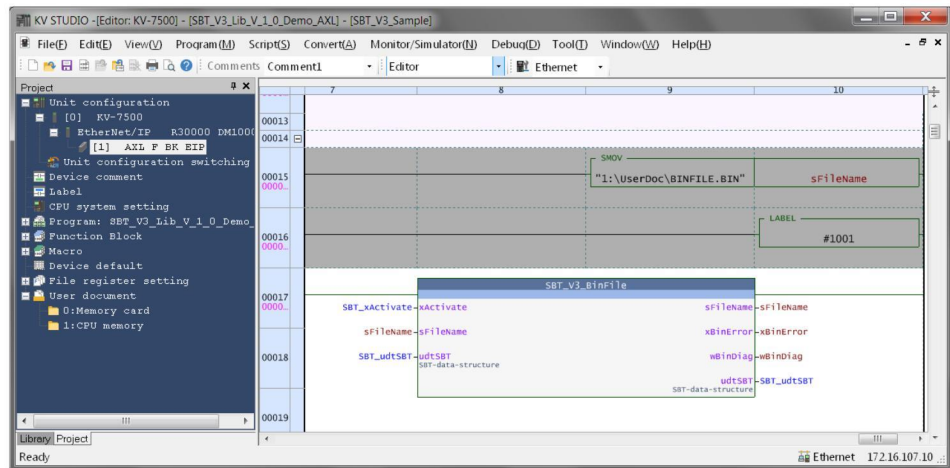


Figure 4-28 SBT\_V3\_BinFile function block

#### 4.4.6 Complete example project in KV STUDIO

Once all the necessary function blocks have been imported and connected, the KV STUDIO project has the following structure:

- Function block SBT\_V3\_BinFile
- 2 instances of the SBT\_V3\_ReadDev function block
- Function block SBT\_V3\_Operate
- 2 instances of the SBT\_V3\_WriteDev function block
- SBT\_V3\_ProjHeader function block
- SBT\_V3\_TransTime function block

See example project.

#### 4.4.7 Compile project and load it into the controller

- Compile the created project and load it into the controller.

The controller switches to the “Run” state.

The LPSDO8/3 module indicates that it has not been parameterized yet by flashing the FS LED.

## 4.5 Startup



A flowchart for starting up and testing the application can be found in Appendix A on page 39.

1. Switch to KV STUDIO.
2. Activate the online values in KV STUDIO.
3. Set the "xActivate" input parameter to "1".
4. Check the "xDiffLogicDetected" output parameter.

If the value is set to "1", a new SAFECONF project has been detected.

5. Set the "xAcceptDiffLogic" input parameter to "1".

You can check the download progress at the "iDownloadLoaded" output parameter. Download time: approximately 40 seconds (depending on the project size, CPU, and bus speed).

6. Check whether the "xRunLogic" output parameter outputs "TRUE" and the "wDiagCode" output parameter outputs the value 16#8000.



If a different code is issued by the "wDiagCode" output parameter, see "Function block diagnostics" on page 46.

Once successfully downloaded, the safe application is ready. The diagnostics LEDs of the modules now have the following status:

LEDs off: FS, CM, SD

LEDs on: P, U<sub>0</sub>, D

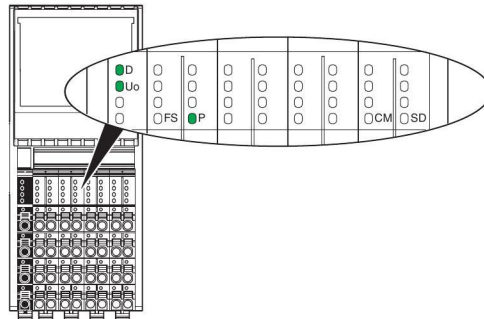


Figure 4-29 Diagnostics LEDs



Perform an overall safety validation after you start up your system.

## 4.6 Online configuration and connection establishment

In order to monitor the online status of the safety logic of the LPSDO8/3 module during startup or maintenance, two configuration steps are required:

1. Configuration of the KV-XLE02 module as a Modbus TCP server.
2. Configuration of a Modbus TCP connection in SAFECONF.

### 4.6.1 Configuration of the KV-XLE02 module as a Modbus TCP server

- Launch the Unit Editor by double-clicking on „[0] KV-7500“.

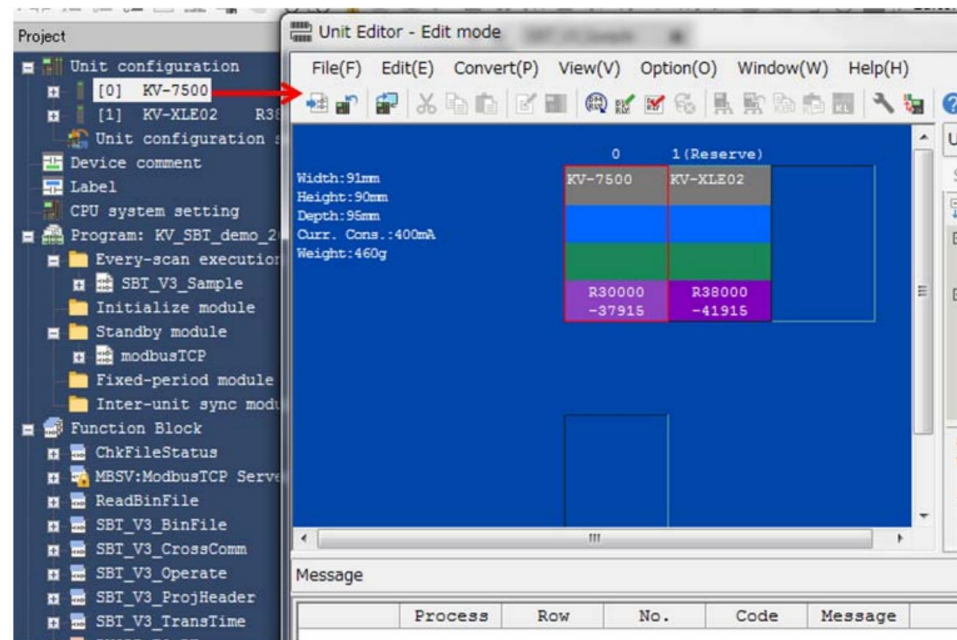


Bild 4-30 Launch Unit Editor

- In the Unit-Editor, click once on the KV-XLE02 module to open the Setup unit dialog.

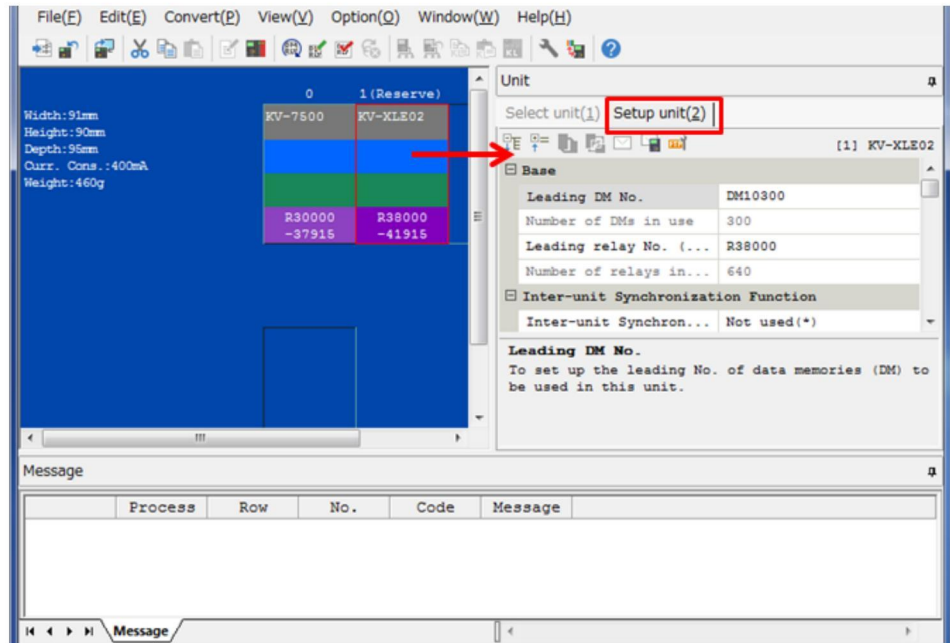


Bild 4-31 Open Setup unit dialog for KV-XLE02 module

#### Activate Modbus TCP server

- Activate the Modbus TCP server on Port 1 with the following network settings.

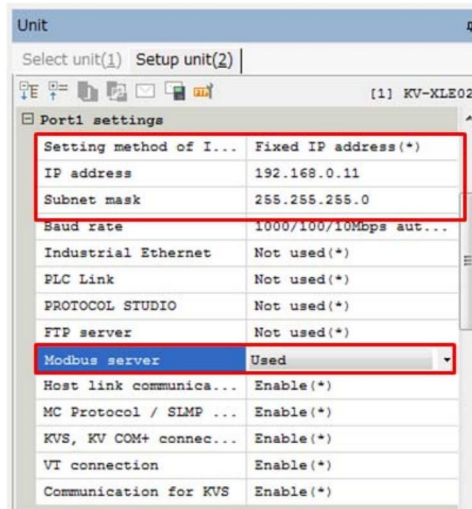


Bild 4-32 Activate Modbus TCP server on Port 1



- Activate the Modbus TCP server on Port 2.

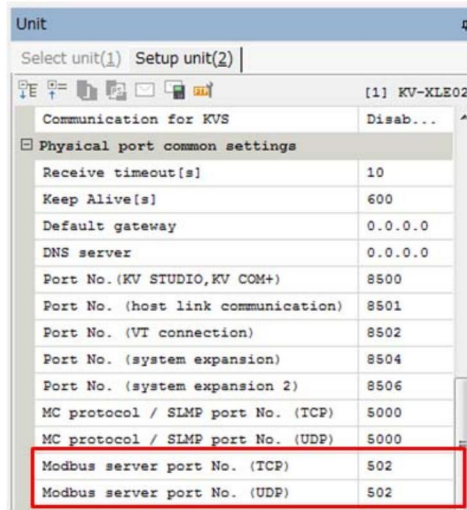


Bild 4-33 Activate Modbus TCP server on Port 2

#### Assignment of the Modbus address range

- Take the address range from the Holding registers out of the Setup unit dialog.
- Write the address range in the parameter „arrSBTONCtrlBuf“ of the „SBT\_V3\_Operate“ function block.
- Use an offset of 352 for the parameter „arrSBTONValBuf“.

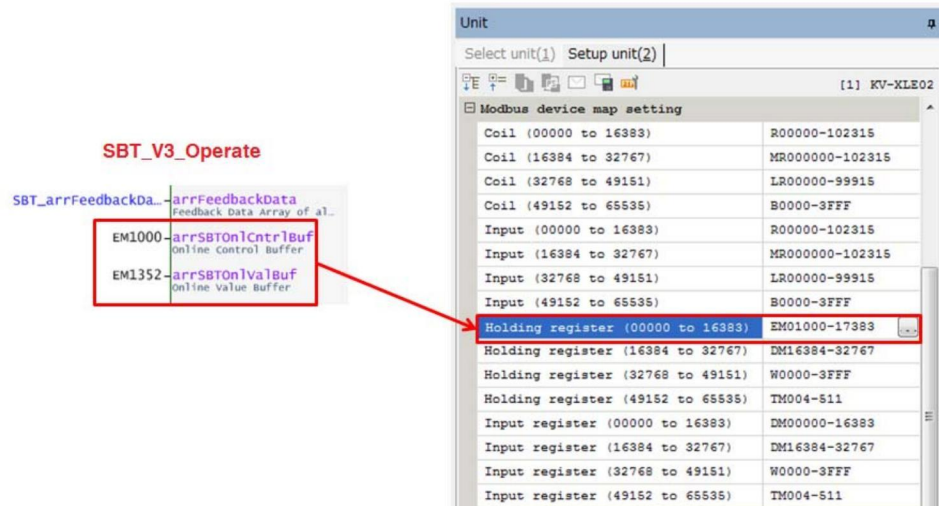


Bild 4-34 Assignment of the Modbus address range at the operate function block

**Complete settings**

- Confirm the settings bei clicking on „Apply“.

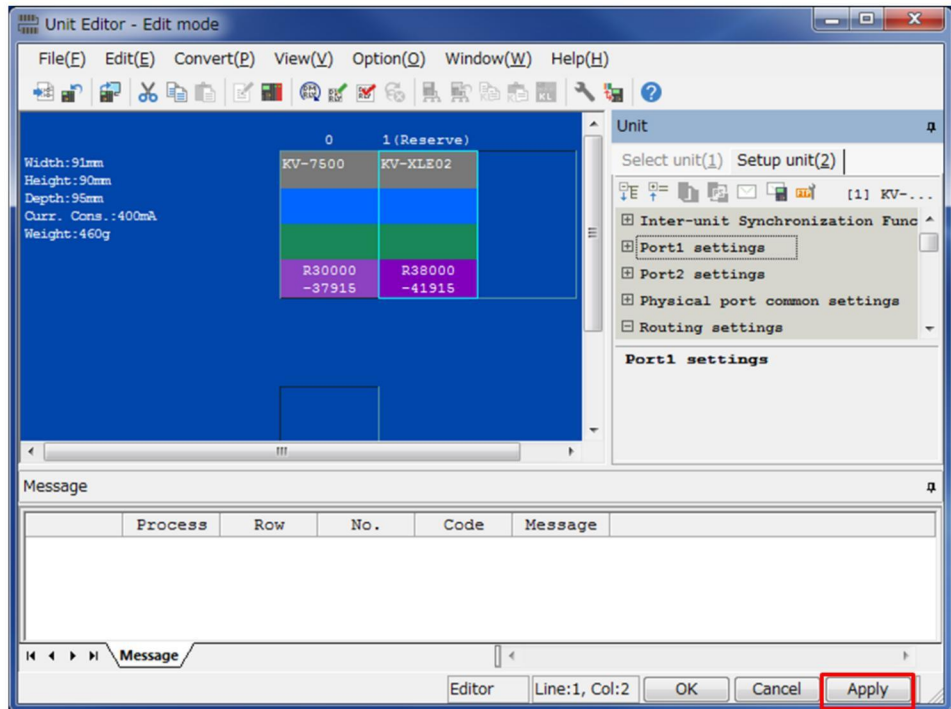


Figure 4-35 Complete settings

## 4.6.2 Configuration of a Modbus TCP connection in SAFECONF

- Open SAFECONF.
- In the hardware editor, right-click on the LPSDO8/3 module and select “Online configuration...”.

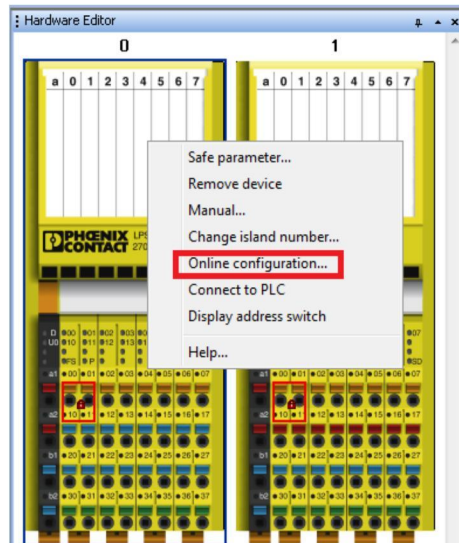


Figure 4-36 Selecting online configuration

- Select the “Generic Modbus/TCP Device” interface and click on “Next”.

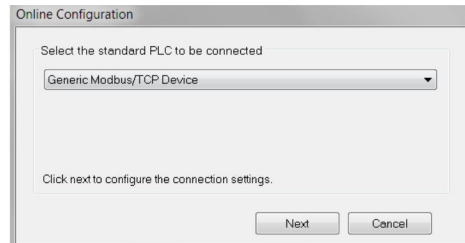


Figure 4-37 Selecting the interface

- Enter the IP address of the Modbus TCP server on the KV-XLE02 module.
- Set the “Expert Settings” as shown in Figure 4-38.
- Once this test has been completed successfully, click on “Finish”.

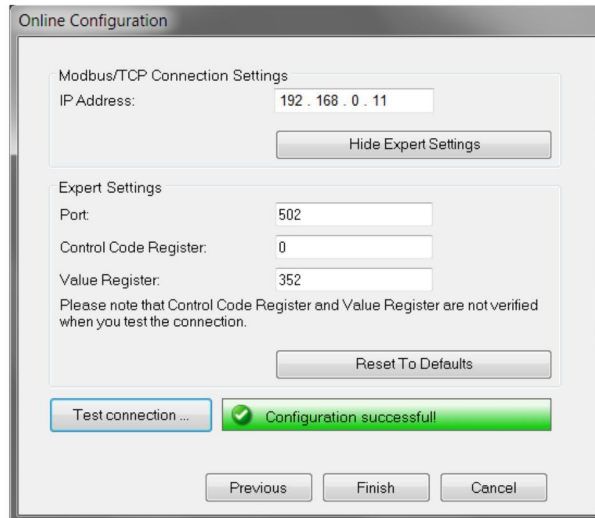


Figure 4-38 Making the connection settings

- After a few seconds, the following message appears in the SAFECONF status bar.

**Project: Read/Write** **PLC: Logged off** **PLC: Connected**

Figure 4-39 SAFECONF status message

### 4.6.3 Displaying online values in SAFECONF

Communication between SAFECONF and the “SBT\_V3\_Operate” function block is achieved by the “arrSBTONIcntrlBuf” and “arrSBTONIValBuf” I/O parameters.

To view the online values in SAFECONF, proceed as follows:

- Open SAFECONF.
- In the hardware editor, right-click on the LPSDO8/3 module.
- If another online connection is active, disconnect the connection by selecting “Disconnect from PLC”.

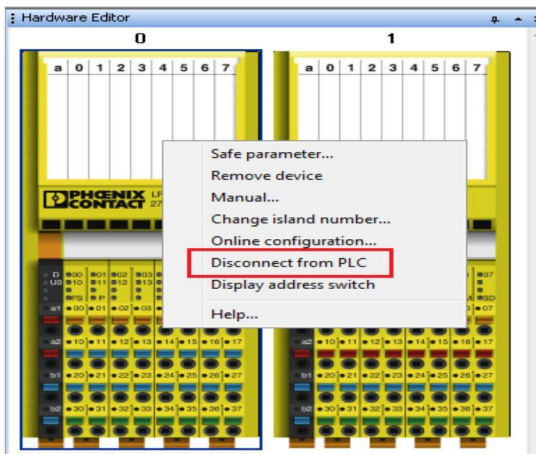


Figure 4-40 Disconnecting an online connection

- In the hardware editor, right-click on the LPSDO8/3 module.
- Establish an online connection by selecting “Connect to PLC”.

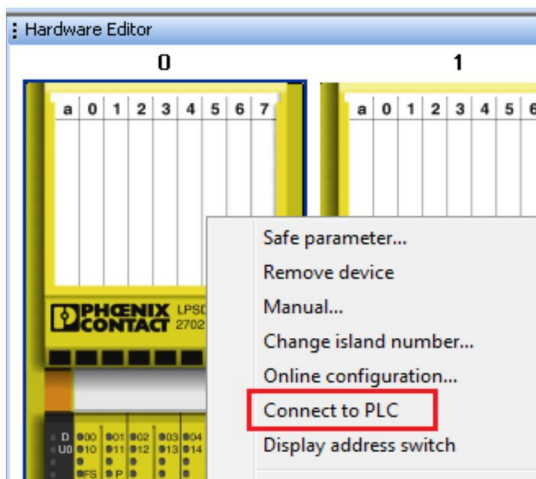


Figure 4-41 Establishing an online connection

- In the SAFECNF tool bar, click on the “Show Online Values” button.

The online values are now displayed in the SAFECNF project.

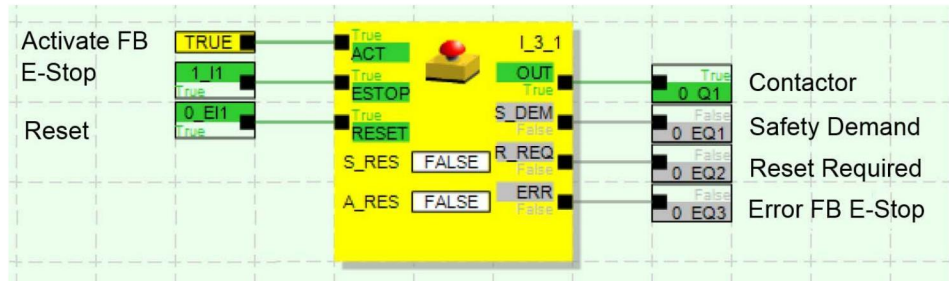


Figure 4-42 Online values in the SAFECNF project

# A Flowchart for starting up and testing the application

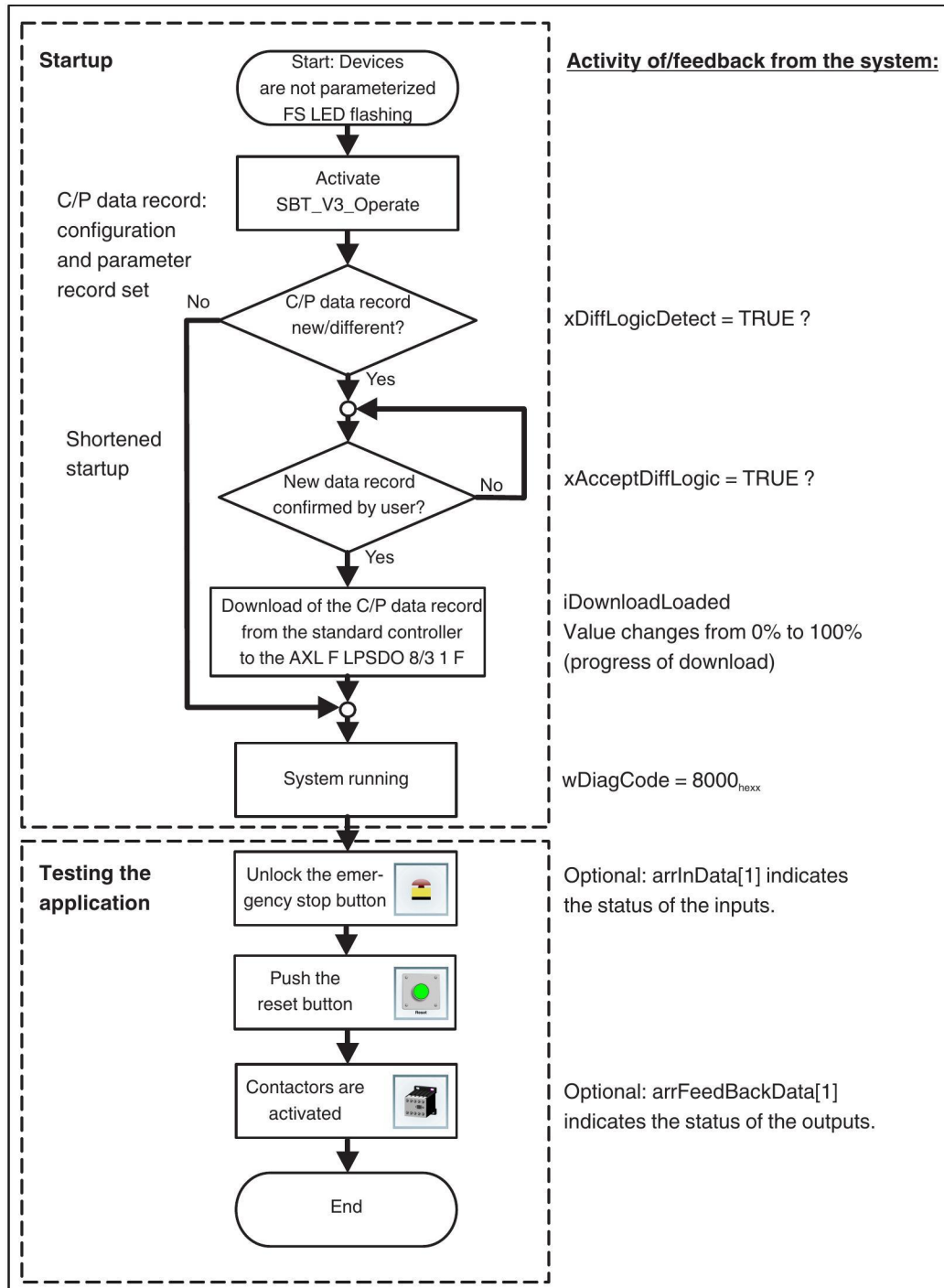


Figure A-1 Flowchart for starting up and testing the application

## B Description of the function blocks for SafetyBridge Technology V3

**Validity** The function blocks described are valid for the following controller/network combinations.

Table B-1 Validity of the function blocks

<b>Controller</b>	KEYENCE KV-7500
<b>Bus system</b>	EtherNet/IP™
<b>Software</b>	KV STUDIO from V9.41 onward

**Function block overview** Table B-2 Function block overview

Function block	Description	Programming language
SBT_V3_Operate	Function block used to operate an SBT island with a maximum of 16 satellites: <ul style="list-style-type: none"> <li>– Download the SAFECONF project</li> <li>– Read the input, diagnostic, and feedback signals of all connected SBT modules</li> <li>– Makes the “enableOut” principle available</li> </ul> See <a href="#">B 1 on page 42</a>	KV Script
SBT_V3_ReadDev	Reads input data from the I/O (process image) to the SBT_V3_Operate function block via udtSBT See <a href="#">B 2 on page 47</a>	KV Script
SBT_V3_WriteDev	Writes output data from the SBT_V3_Operate function block to the I/O (process image) via udtSBT See <a href="#">B 3 on page 48</a>	KV Script
SBT_V3_ProjHeader	Displays the project information for the connected LPSDO module and the imported SAFECONF project See <a href="#">B 4 on page 49</a>	KV Script
SBT_V3_BinFile	Read the contents of the SAFECONF project to be imported from the CPUMEM/userdoc area of the controller See <a href="#">B 5 on page 50</a>	KV Script



**Description of the function blocks for SafetyBridge Technology V3**

Function block	Description	Programming language
SBT_V3_TransTime	Measures the runtime of the current and maximum transmission times between the LPSDO module and each of the connected satellites  See B 6 on page 52	KV Script
SBT_V3_CrossComm	Cross communication between SBT islands within a controller  See B 7 on page 54	KV Script
SBT_V3_CrossCommMaster or SBT_V3_CrossCommSlave	Cross communication between two SBT islands, if they are connected to different controllers	KV Script

**Data types**

Table B-3 Data types

Designation	Description
udtSBT_V3	<ul style="list-style-type: none"> <li>- Internal user-defined data type</li> <li>- Structures of this data type connect the function blocks of an island</li> <li>- No access required to these structures</li> <li>- Access to all SBT data via the input and output parameters of the function blocks</li> </ul>

## B 1 SBT\_V3\_Operate function block

The function block performs the following functions:

- Download of the configuration and parameter data record from the standard controller to the AXL F LPSDO8/3 1F
- Cyclical routing of the SafetyBridge data flow
- Display of diagnostic information for all the SafetyBridge modules of an island
- Acknowledgment of error messages
- Reading the status information for all inputs and outputs

Use the function block once per island.

The figure shows the division of the function block into its individual function areas. The parameters used are described in the tables below.

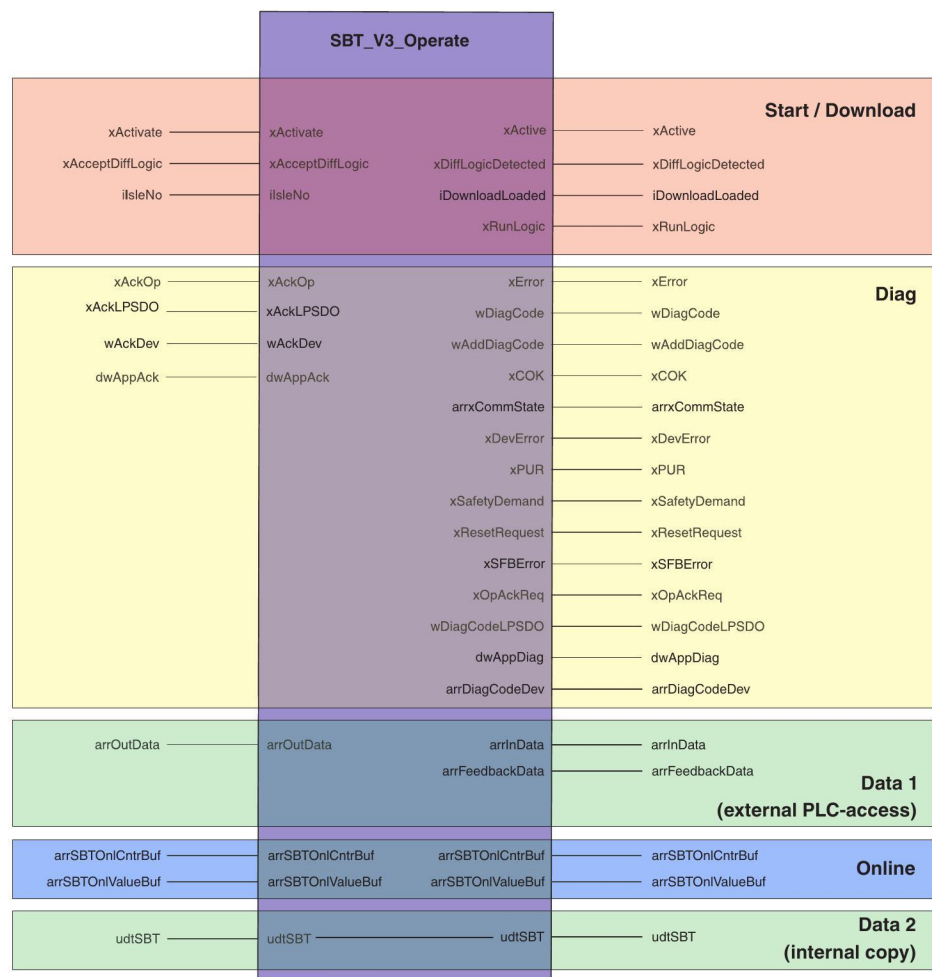


Figure B-1 SBT\_V3\_Operate function block

## B 1.1 Input parameters

Table B-4 Input parameters for the SBT\_V3\_Operate function block

Name	Type	Description
xActivate	Bit	Activation/Deactivation of the function block
xAcceptDiffLogic	Bit	User confirmation signal that the safety logic differs for the controller and the LPSDO module True: user has accepted and confirmed the difference (the download then starts)
ilsleNo	1W signed integer	SBT island number Valid range: 1... 31 (must be the same as the DIP switch position on the LPSDO module)
xAckOp	Bit	User confirmation of an "xOpAckReq" signal
xAckLPSDO	Bit	User confirmation of an LPSDO error
wAckDev	1W unsigned integer	User confirmation of module errors: wAckDev.0: for module 1 wAckDev.15: for module 16
dwAppAck	2W unsigned integer	32 freely configurable acknowledgment signals (application data) from the standard controller to the LPSDO module
arrOutData	1W unsigned integer [17]	Array of "enableOutputs" from all output modules (satellites) of the island; the "enableOut" principle is optional and must be parameterized in SAFECONF The index represents the module number (0 stands for the LPSDO module): arrOutData [0] - enableOut data LPSDO arrOutData [1] - enableOut data module 1 ... arrOutData [16] - enableOut data module 16

**B 1.2 Output parameters**

Table B-5 Output parameters for the SBT\_V3\_Operate function block

Name	Type	Description
xActive	Bit	True: FB initialized successfully False: FB not initialized
xDiffLogicDetected	Bit	True: detection of different safety logic for the controller and the LPSDO module
iDownloadLoaded	1-word signed integer	Percentage of SBT logic that has been downloaded
xRunLogic	Bit	Safety logic (SAFECONF logic) running on the LPSDO module
xError	Bit	Function block error
wDiagCode	1W signed integer	Diagnostic code for the function block (status or error)
wAddDiagCode	1W signed integer	Additional diagnostic code for the function block (status or error)
xCOK	Bit	True: communication status of the SBT island is OK False: loss of communication for one or more modules
arrxCommState	Bit [17]	Communication status of each module; each bit represents the status of a module  True: communication established False: communication interrupted  The index represents the module number (0 is not used): arrxCommState [1] - communication module 1 ... arrxCommState [16] - communication module 16
xDevError	Bit	Indicates an error in one or more SBT modules
xPUR	Bit	Error state that cannot be acknowledged; restart is required
xSafetyDemand	Bit	True: one or more safety demands in the SAFECONF project (LPSDO) False: no safety demand
xResetRequest	Bit	True: one or more safety function blocks indicate an error False: no error message from a safety function block
xSFBErro	Bit	True: one or more reset requests in the SAFECONF project (LPSDO) False: no reset request
xOpAckReq	Bit	Acknowledgment by the user is required (one or more communication errors have occurred)
wDiagCodeLPSDO	1W unsigned integer	Diagnostic code of the LPSDO module (device-specific; see module user documentation)
arrDiagCodeDev	1W unsigned integer [17]	Array of SBT module diagnostic codes (device-specific; see module user documentation)

Name	Type	Description
dwAppDiag	2W unsigned integer	32 freely configurable feedback signals from the LPSDO module to the standard controller
arrInData	1W unsigned integer [17]	Array of input data from all input modules (satellites) of the island The index represents the module number (0 stands for the LPSDO module): arrInData [0] - InData LPSDO (no input data) arrInData [1] - InData module 1 ... arrInData [16] - InData module 16
arrFeedbackData	1W unsigned integer [17]	Array of feedback output data from the LPSDO module and the satellites (automatic call) The index represents the module number (0 stands for the LPSDO module): arrFeedbackData [0] - feedback data LPSDO arrFeedbackData [1] - feedback data module 1 ... arrFeedbackData [16] - feedback data module 16

### B 1.3 I/O parameters

Table B-6 I/O parameters for the SBT\_V3\_Operate function block

Name	Type	Description
arrSBTONlCntrlBuf	EM address of the Modbus TCP slave See example.	Pointer to ONLINE-Control array
arrSBTONlValueBuf	EM address of the Modbus TCP slave See example.	Pointer to ONLINE-Value array
udtSBT	udtSBT_V3	Structure for data exchange between function blocks

**B 1.3.1 Function block diagnostics**

**Diagnostics**

Table B-7 Diagnostic codes

<b>DiagCode</b>	<b>Meaning</b>	
0000 <sub>hex</sub>	Function block is not active	
0100 <sub>hex</sub>	Initialization	
C100 <sub>hex</sub>	Initialization error	
	<b>AddDiagCode</b>	<b>Meaning</b>
	0004 <sub>hex</sub>	Wrong or invalid island number
8000 <sub>hex</sub>	Function block is active and operating without errors	
8100 <sub>hex</sub>	Reading data from data block	
8200 <sub>hex</sub>	Reading "Project Header" of LPSDO module	
	<b>AddDiagCode</b>	<b>Meaning</b>
	0001 <sub>hex</sub>	Initializing reading of "Project Header"
	0002 <sub>hex</sub>	Reading of "Project Header" is complete
8300 <sub>hex</sub>	Comparing "Header" of LPSDO module and the loaded program	
8400 <sub>hex</sub>	Downloading SAFECONF program to the LPSDO module	
	<b>AddDiagCode</b>	<b>Meaning</b>
	0000 <sub>hex</sub>	Removing old "Header" and writing a new one
	0001 <sub>hex</sub>	Downloading logic block
	0002 <sub>hex</sub>	Downloading address block
	0003 <sub>hex</sub>	Downloading new "Project Header"
8500 <sub>hex</sub>	Download completed successfully	
C400 <sub>hex</sub>	Error during download	
	<b>AddDiagCode</b>	<b>Meaning</b>
	xxyy <sub>hex</sub>	Diagnostic code from the LPSDO module (see module user documentation)

## B 2 SBT\_V3\_ReadDev function block

The function block reads the IN process data of a SafetyBridge module and writes the data to the transfer structure for the “SBT\_V3\_Operate” function block.

**1 instance per SBT module**

Use one instance of the function block per SafetyBridge module in the island.

**Observe the sequence**

Arrange all the "SBT\_V3\_ReadDev" instances **before** the “SBT\_V3\_Operate” function block in order to enable the best possible response time in the SafetyBridge system.

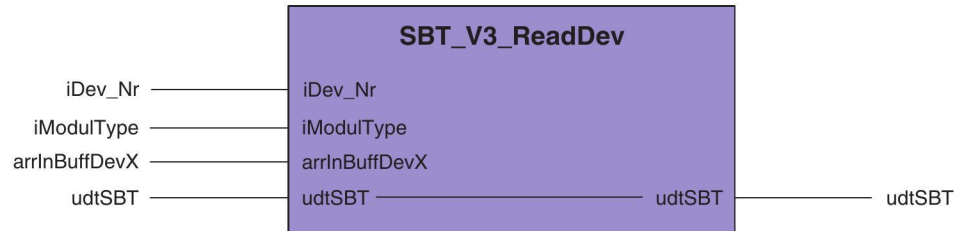


Figure B-2 SBT\_V3\_ReadDev function block

### B 2.1 Input parameters

Table B-8 Input parameters for the SBT\_V3\_ReadDev function block

Name	Type	Description
iDev_Nr	1W signed integer	Device number 0: LPSDO module 1 ... 16: module 1 ... 16
iModulType	1W signed integer	Format of connected “arrInBuffDevX” 0: Motorola 1: Intel Other: for further developments
arrInBuffDevX	WORD based address (e.g. W, DM, EM) See example.	Exchange data (inputs) from SBT module X

### B 2.2 I/O parameters

Table B-9 I/O parameters for the SBT\_V3\_ReadDev function block

Name	Type	Description
udtSBT	udtSBT_V3	Structure for data exchange between function blocks

### B 3 SBT\_V3\_WriteDev function block

The function block writes the OUT process data of a SafetyBridge module from the transfer structure of the “SBT\_V3\_Operate” function block to the output process image.

**1 instance per SBT module**

Use one instance of the function block per SafetyBridge module in the island.

**Observe the sequence**

Arrange all the SBT\_V3\_WriteDev” instances **after** the “SBT\_V3\_Operate” function block in order to enable the best possible response time in the SafetyBridge system.

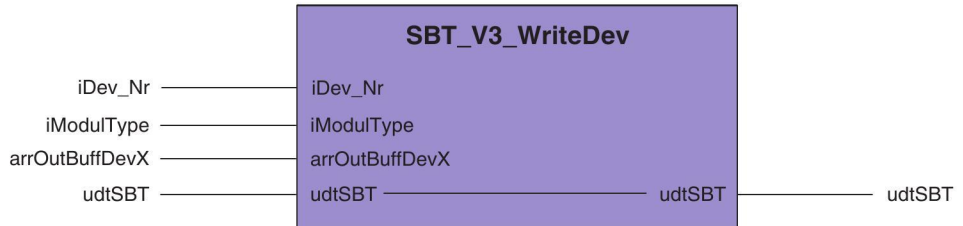


Figure B-3 SBT\_V3\_WriteDev function block

#### B 3.1 Input parameters

Table B-10 Input parameters for the SBT\_V3\_WriteDev function block

Name	Type	Description
iDev_Nr	1W signed integer	Device number 0: LPSDO module 1 ... 16: module 1 ... 16
iModulType	1W signed integer	Format of connected “arrInBuffDevX” 0: Motorola 1: Intel Other: for further developments
arrOutBuffDevX	WORD based address (e.g. W, DM, EM) See example.	Exchange data (outputs) from SBT module X

#### B 3.2 I/O parameters

Table B-11 I/O parameters for the SBT\_V3\_WriteDev function block

Name	Type	Description
udtSBT	udtSBT_V3	Structure for data exchange between function blocks



## B 4 SBT\_V3\_ProjHeader function block

### Optional function block



This function block is optional. It is not essential in order to operate a SafetyBridge island.

The function block contains all information about the SAFECONF project and the project stored on the LPSDO module.

### 1 function block per island

Use the function block once per island.

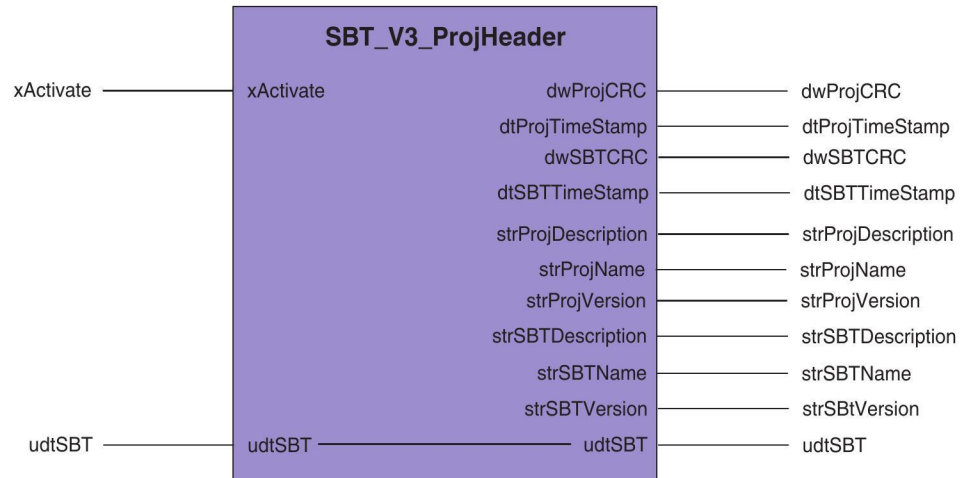


Figure B-4 SBT\_V3\_ProjHeader function block

### B 4.1 Input parameters

Table B-12 Input parameters for the SBT\_V3\_ProjHeader function block

Name	Type	Description
xActivate	Bit	Activation/Deactivation of the function block

### B 4.2 Output parameters

Table B-13 Output parameters for the SBT\_V3\_ProjHeader function block

Name	Type	Description
dwProjCRC	2W signed integer	Checksum of the new project (FB)
dtProjTimeStamp	Text string [32]	Time stamp of the new project (FB)
dwSBTCRC	2W signed integer	Checksum of the active project (LPSDO)
dtSBTimeStamp	Text string [32]	Time stamp of the active project (LPSDO)
strProjDescription	Text string [32]	Description of the new project (FB); see SAFECONF
strProjName	Text string [32]	Name of the new project (FB); see SAFECONF
strProjVersion	Text string [32]	Version of the new project (FB); see SAFECONF
strSBTDescription	Text string [32]	Description of the active project (LPSDO)
strSBTName	Text string [32]	Name of the active project (LPSDO)
strSBTVersion	Text string [32]	Version of the active project (LPSDO)

### B 4.3 I/O parameters

Table B-14 I/O parameters for the SBT\_V3\_ProjHeader function block

Name	Type	Description
udtSBT	udtSBT_V3	Structure for data exchange between function blocks

## B 5 SBT\_V3\_BinFile function block

The function block reads the project data from the BIN file and makes the data available to the „SBT\_V3\_Operate“ function block.

**1 function block per island** Use the function block once per island.

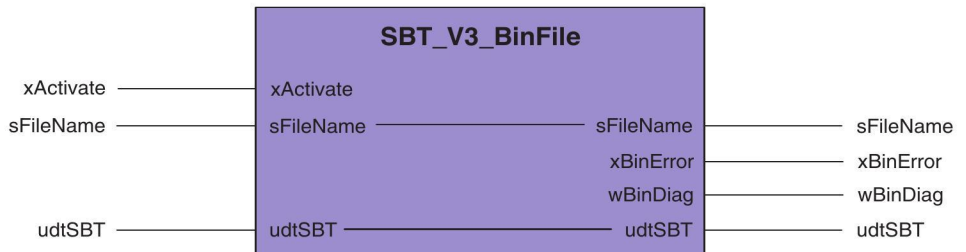


Bild B-5 SBT\_V3\_BinFile function block

### B 5.1 Input parameters

Tabelle B-15 Input parameters for the SBT\_V3\_BinFile function block

Name	Type	Description
xActivate	Bit	Activation/deactivation of the function block

### B 5.2 Output parameters

Tabelle B-16 Output parameters for the SBT\_V3\_BinFile function block

Name	Type	Description
xBinError	Bit	An error occurred while editing the file Error messages see "wBinDiag"
wBinDiag	1W unsigned integer	Detailed error messages: 0: Completed Successfully 1: Memory card not inserted 2: Abnormal file name 3: File does not exist 4: File already exists 5: File load error 6: File write error 7: Abnormal file read/write position 8: Not enough space on memory card 9: File access competition 10: CPU memory restricted error 1000: Other errors

### B 5.3 I/O parameters

Table B-17 I/O parameters for the SBT\_V3\_BinFile function block

Name	Type	Description
sFileName	STRING	BIN file from SAFECONF, which is stored on the controller, e. g. „BinFile.BIN“  <b>Important:</b> the name must not exceed 24 characters.
udtSBT	udtSBT_V3	Structure for data exchange between function blocks

## B 6 SBT\_V3\_TransTime function block

### Optional function block



This function block is optional. It is not essential in order to operate a SafetyBridge island.

The function block measures the data transmission times between the logic module and satellite modules 1 to 16, as well as the maximum transmission time of a module.

### 1 function block per island

Use the function block once per island.



If the data transmission time of a module is greater than the F\_WD\_Time, a communication error is issued at the LPSDO module. In this case, the data transmission time and F\_WD\_Time should be checked.

Please note that increasing the F\_WD\_Time has a direct effect on the safety function because it increases response times and therefore delay times and/or safety distances.

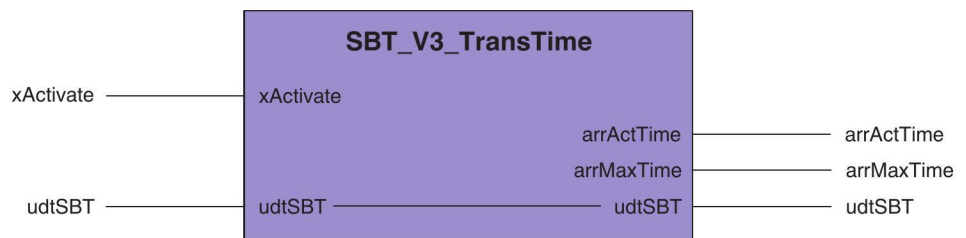


Figure B-6 SBT\_V3\_TransTime function block

### B 6.1 Input parameters

Table B-18 Input parameters for the SBT\_V3\_TransTime function block

Name	Type	Description
xActivate	Bit	Activation/Deactivation of the function block

## B 6.2 Output parameters

Table B-19 Output parameters for the SBT\_V3\_TransTime function block

Name	Type	Description
arrActTime	2W signed integer [16]	Current data transmission time between LPSDO module and satellite modules The index represents the module number (0 is not used): arrActTime [1] - current transmission time for module 1 ... arrActTime [16] - current transmission time for module 16
arrMaxTime	2W signed integer [16]	Maximum data transmission time between LPSDO module and satellite modules since the last FB activation The index represents the module number (0 is not used): arrMaxTime [1] - max. transmission time for module 1 ... arrMaxTime [16] - max. transmission time for module 16

## B 6.3 I/O parameters

Table B-20 I/O parameters for the SBT\_V3\_TransTime function block

Name	Type	Description
udtSBT	udtSBT_V3	Structure for data exchange between function blocks

## B 7 SBT\_V3\_CrossComm function block

### Optional function block



This function block is optional. It is not essential in order to operate a SafetyBridge island.

In SafetyBridge Technology V3, islands can communicate with one another. Cross communication takes place via a master/slave model, where one or more islands can act as slaves for other master modules. Each island has the “udtSBT” data structure. The “SBT\_V3\_CrossComm” function block combines the data structures of the individual islands in an array, thereby enabling cross communication between the islands.

### 1 function block per standard controller

Use the function block once per standard controller.

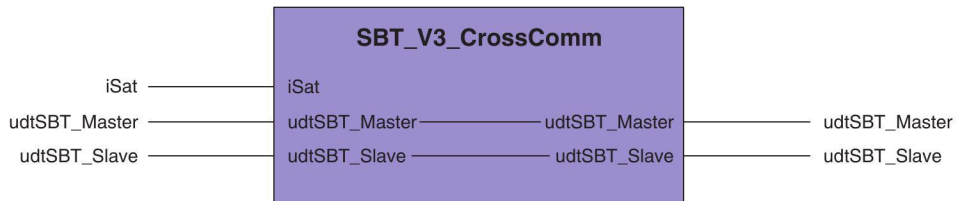


Figure B-7 SBT\_V3\_CrossComm function block

### B 7.1 Input parameters

Table B-21 Input parameters for the SBT\_V3\_CrossComm function block

Name	Type	Description
iSat	2W signed integer	Satellite number of the LPSDO slave module in the project of the superimposed LPSDO master module

### B 7.2 I/O parameters

Table B-22 I/O parameters for the SBT\_V3\_CrossComm function block

Name	Type	Description
udtSBT_Master	udtSBT_V3	Data structure of the superimposed LPSDO master project
udtSBT_Slave	udtSBT_V3	Data structure of the subordinate slave project

### Cross communication between control systems



For the cross communication between control systems use the function blocks “SBT\_V3\_CrossCommMaster” and “SBT\_V3\_CrossCommSlave”.  
For further information contact your local Phoenix Contact subsidiary.

## C Revision history

Revision	Date	Contents	Page
00	2018-01-17	First publication	
01	2019-09-30	New Cover	
		KV STUDIO Version updated	
		Infobox inserted in Chapter 4.4 “KV STUDIO project configuration”	21
		Chapter 4.6 “Online configuration and connection establishment” re-vised	31
		Data types of function blocks parameters revised in Appendix B	from 40
		Parameters of function blocks revised: B 1 “SBT_V3_Operate func-tion block”, B 2 “SBT_V3_ReadDev function block”,	42
		B 3 “SBT_V3_WriteDev function block”	47
		B 3 “SBT_V3_WriteDev function block”	48
		Appendix B 5 “SBT_V3_BinFile function block” inserted	50
		Infobox on cross communication between control systems added to	54
		Appendix B 7 “SBT_V3_CrossComm function block”	

---

## Please observe the following notes

### **General terms and conditions of use for technical documentation**

Phoenix Contact reserves the right to alter, correct, and/or improve the technical documentation and the products described in the technical documentation at its own discretion and without giving prior notice, insofar as this is reasonable for the user. The same applies to any technical changes that serve the purpose of technical progress.

The receipt of technical documentation (in particular user documentation) does not constitute any further duty on the part of Phoenix Contact to furnish information on modifications to products and/or technical documentation. You are responsible to verify the suitability and intended use of the products in your specific application, in particular with regard to observing the applicable standards and regulations. All information made available in the technical data is supplied without any accompanying guarantee, whether expressly mentioned, implied or tacitly assumed.

In general, the provisions of the current standard Terms and Conditions of Phoenix Contact apply exclusively, in particular as concerns any warranty liability.

This manual, including all illustrations contained herein, is copyright protected. Any changes to the contents or the publication of extracts of this document is prohibited.

Phoenix Contact reserves the right to register its own intellectual property rights for the product identifications of Phoenix Contact products that are used here. Registration of such intellectual property rights by third parties is prohibited.

Other product identifications may be afforded legal protection, even where they may not be indicated as such.



---

## How to contact us

### Internet

Up-to-date information on Phoenix Contact products and our Terms and Conditions can be found on the Internet at:

[phoenixcontact.com](http://phoenixcontact.com)

Make sure you always use the latest documentation.

It can be downloaded at:

[phoenixcontact.net/products](http://phoenixcontact.net/products)

### Subsidiaries

If there are any problems that cannot be solved using the documentation, please contact your Phoenix Contact subsidiary.

Subsidiary contact information is available at [phoenixcontact.com](http://phoenixcontact.com).

### Published by

PHOENIX CONTACT GmbH & Co. KG  
Flachsmarktstraße 8  
32825 Blomberg  
GERMANY

PHOENIX CONTACT Development and Manufacturing, Inc.  
586 Fulling Mill Road  
Middletown, PA 17057  
USA

Should you have any suggestions or recommendations for improvement of the contents and layout of our manuals, please send your comments to:

[tecdoc@phoenixcontact.com](mailto:tecdoc@phoenixcontact.com)