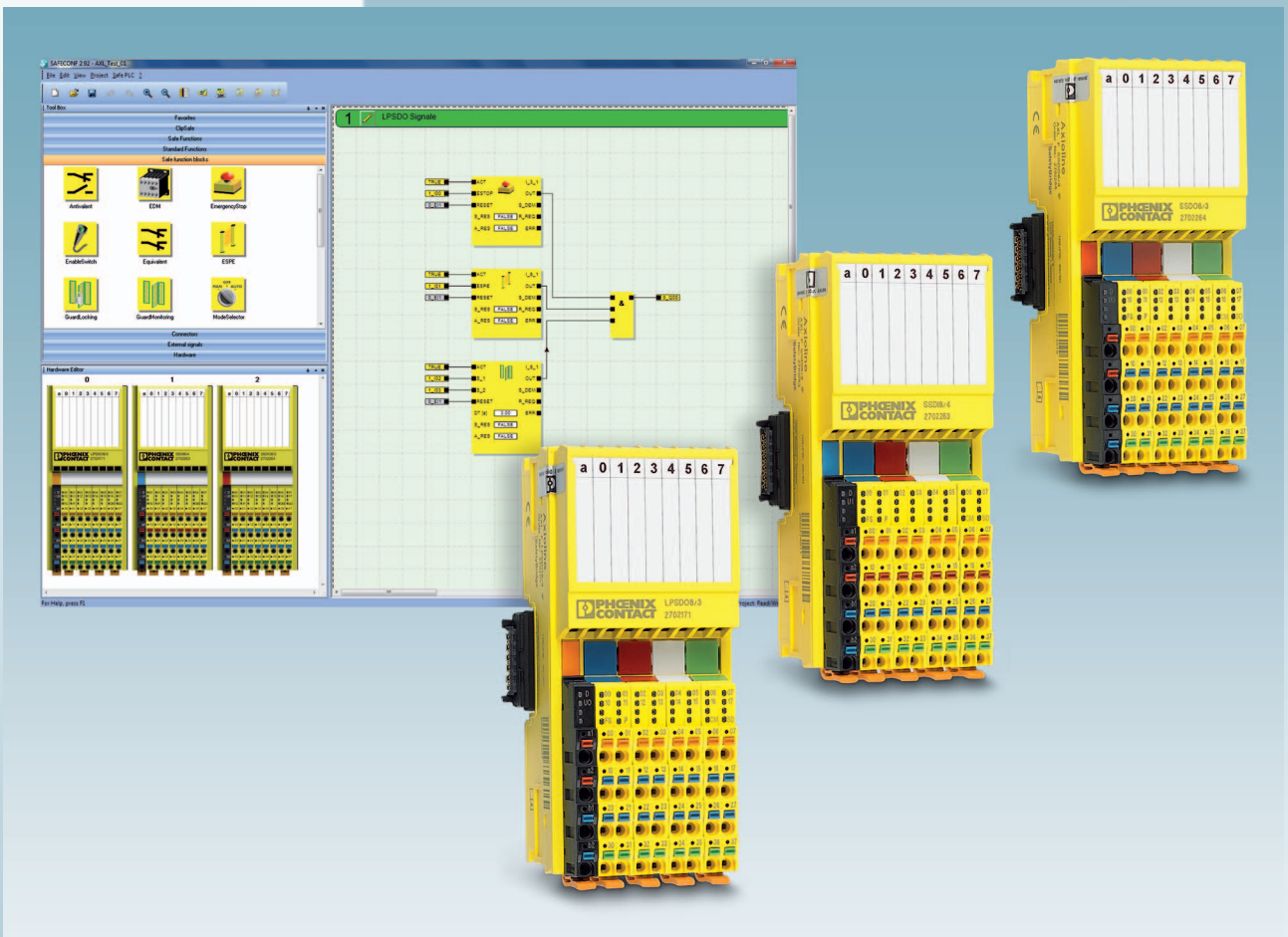



# Axioline F SBT V3 configuration on an AXC 1050 controller with PC Worx

## Quick start guide

PHOENIX CONTACT
INSPIRING INNOVATIONS

# Quick start guide

# Axioline F SBT V3 configuration on an AXC 1050 controller with PC Worx

2016-05-18

Designation: UM QS EN AXL F SBT V3 AXC 1050 PC WORX

Revision: 00

Order No.: —

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 |

# Please observe the following notes

### User group of this manual

The use of products described in this manual is oriented exclusively to qualified application programmers and software engineers, who are familiar with the safety concepts of automation technology and applicable standards.

### Explanation of symbols used and signal words

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety measures that follow this symbol to avoid possible injury or death.

There are three different categories of personal injury that are indicated with a signal word.

**DANGER** This indicates a hazardous situation which, if not avoided, will result in death or serious injury.

**WARNING** This indicates a hazardous situation which, if not avoided, could result in death or serious injury.

**CAUTION** This indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

This symbol together with the signal word **NOTE** and the accompanying text alert the reader to a situation which may cause damage or malfunction to the device, hardware/software, or surrounding property.

This symbol and the accompanying text provide the reader with additional information or refer to detailed sources of information.

### 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

Make sure you always use the latest documentation.
It can be downloaded at:
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.

**Published by**

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

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

### 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.



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.

# Table of contents

# 1 Introduction

## 1.1 Purpose of this user manual

This quick start guide uses an example project to describe how to integrate SafetyBridge Technology V3 modules into an AXC 1050 controller with PC Worx.

## 1.2 Requirements

**Knowledge**

Knowledge of the following is required:

– The components used in the application
– The PC Worx software used
– The 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 |
| AXC 1050 (controller) | - | 2700988 |

– 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.

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 |
| PC Worx V6.30 or later | - |

– Other software: Microsoft Windows

## 1.3 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:<br>Axioline F module with integrated safety logic and safe digital outputs | UM EN AXL F LPSDO8/3 1F | 2702171 |
| User manual:<br>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.

## 1.4 Safety hotline

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

Phone: + 49 5281 9-462777

E-mail: safety-service@phoenixcontact.com

# 2 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 2-1 Integration of a SafetyBridge Technology V3 island

| Step | Process | See... |
|---|---|---|
| **1** | **Download and installation of the Phoenix Contact software** (not safety-related) | |
| | – Download and install SAFECONF configuration software | page 11 |
| | – Download and install integration package for SafetyBridge Technology V3 | page 11 |
| **2** | **Hardware installation** (not safety-related) | |
| | – Set device DIP switches | page 12 |
| | – Mount and wire bus configuration | page 13 |
| | | User documentation for the devices |
| **3** | **Configure safety logic in SAFECONF** (safety-related) | |
| | – Create new project, assign safety island number | page 14, page 16 |
| | – Configure the hardware structure | page 16 |
| | – Parameterize I/O channels | page 17 |
| | – Configure safety function | page 14 |
| | – Export configuration and parameter data record | page 21 |
| | | SAFECONF online help |
| **4** | **Configure a PC Worx project for AXC 1050** (not safety-related) | |
| | – Create project and insert modules | page 22 |
| | – Integrate function blocks for SafetyBridge Technology V3 | page 23 |
| | – Create SBT program in PC Worx | page 24 |
| | | PC Worx online help |
| | – Process data assignment of the devices | page 36 |

Table 2-1 Integration of a SafetyBridge Technology V3 island

| Step | Process | See... |
|---|---|---|
| | – Compile project and download it to the controller | page 37 |
| | – Import configuration and parameter data record | page 37 |
| **5** | **Startup and overall safety validation** (safety-related) | page 40 |

**System overview:**

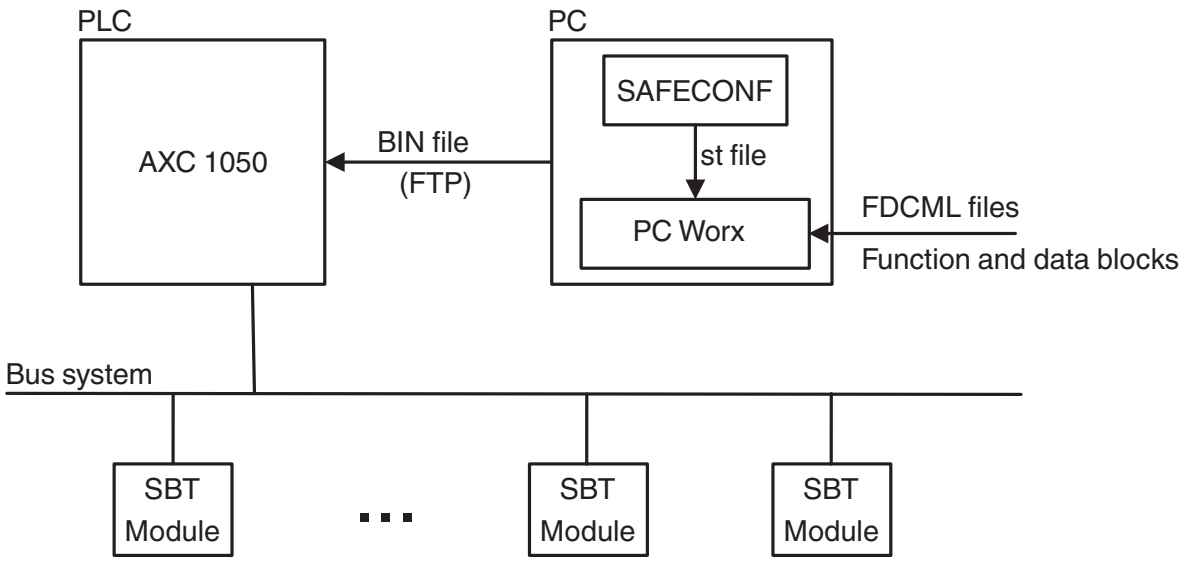



Figure 2-1 System overview of SafetyBridge Technology V3

# 3 Example project: two-channel emergency stop monitoring

## 3.1 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.

Make sure that you always use the latest version of the integration package as well as the correct version of the function blocks for the PC Worx software version. See “Software” on page 7.

**SAFECONF**

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

**SBT V3 integration package**

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

Software

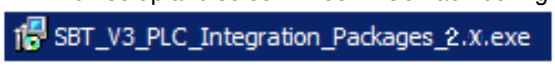
| | Description | Language | Revision |
|---|---|---|---|
| ☑ | [exe, 123 MB] **Software**<br>SafetyBridge technology integration package for controllers from Phoenix Contact, Rockwell and Siemens (S7-1200 from CPU 1214C, S7-1500, S7-300), Schneider as well as CODESYS-based controllers.<br>SBT_V3_PLC_Integration_Packages_2.0.exe | International | 2.0 |

Figure 3-1 Integration package in the download area for Order No. 2702171

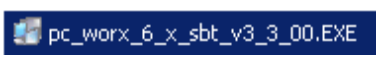
The integration package contains additional file formats so that the SAFECONF configuration can be saved as a BIN file for PC Worx. In addition, the integration package installs an exe file which creates the PC Worx function blocks for SafetyBridge.

3. Install the integration package as follows:

   - Run setup and select “Phoenix Contact” during installation.




   - Open “Windows Start menu, All Programs, Phoenix Contact, SBT V3 for SAFECONF 2.92, Phoenix Contact, SBT V3 Functionblocks”.
   - Install the Siemens function blocks.




Please make a note of where the library files are installed, as you will need this information later when you open the library in PC Worx.

## 3.2 Hardware installation

### 3.2.1 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 3-1 Setting the DIP switches

| | CM | | Island number | | | | | Satellite number | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Operating mode | Reserved | SafetyBridge Technology V3 address: $32_{dec}$ ($20_{hex}$) | | | | | | | | | |
| **DIP switch** | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **Setting for** | **off** | **on** | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 |
| **AXL F LPSDO8/3 1F** | | | **$1_{dec}$** | | | | | **$0_{dec}$** | | | | |

| | CM | | Island number | | | | | Satellite number | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Operating mode | Reserved | SafetyBridge Technology V3 address: $33_{dec}$ ($21_{hex}$) | | | | | | | | | |
| **DIP switch** | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **Setting for** | **off** | **on** | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | **1** |
| **AXL F SSDI8/4 1F** | | | **$1_{dec}$** | | | | | **$1_{dec}$** | | | | |

You can display the complete DIP switch setting in the SAFECONF configuration software by right-clicking on the module and selecting “Display address switch”. See Figure 3-10 on page 17.

### 3.2.2 Mounting and wiring the bus configuration

Refer to the user documentation for the devices.
See “Additional documentation” on page 8.

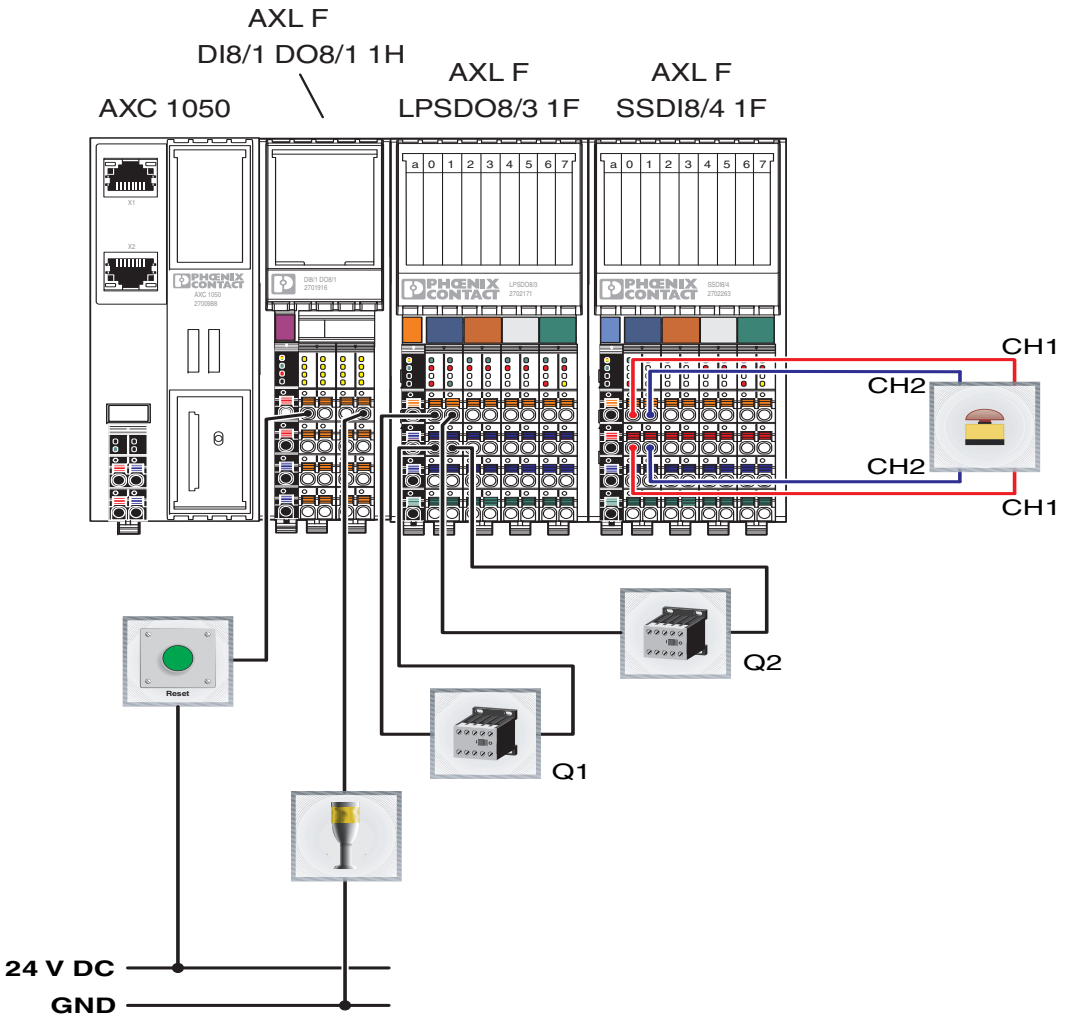



Figure 3-2 Bus configuration for example project

1. After setting the DIP switches on the modules, mount the bus configuration as illustrated.
2. Connect the power supply for the controller and I/O modules in accordance with the corresponding the 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.

## 3.3 Configuring the safety logic in SAFECONF

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

### 3.3.1 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.

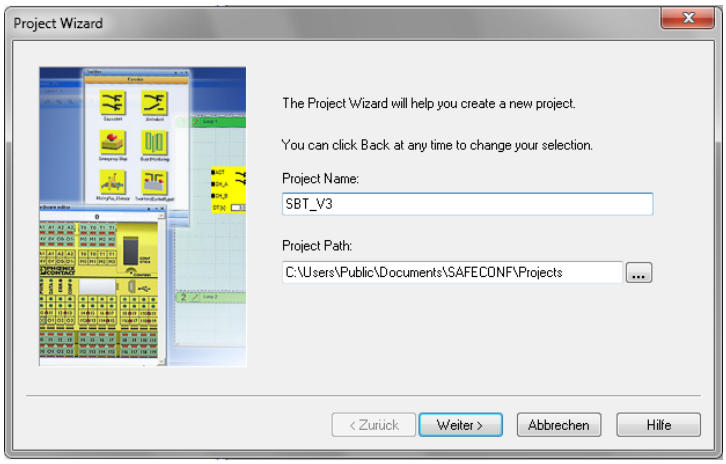



Figure 3-3 Creating the project name and path

**Select master device**

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

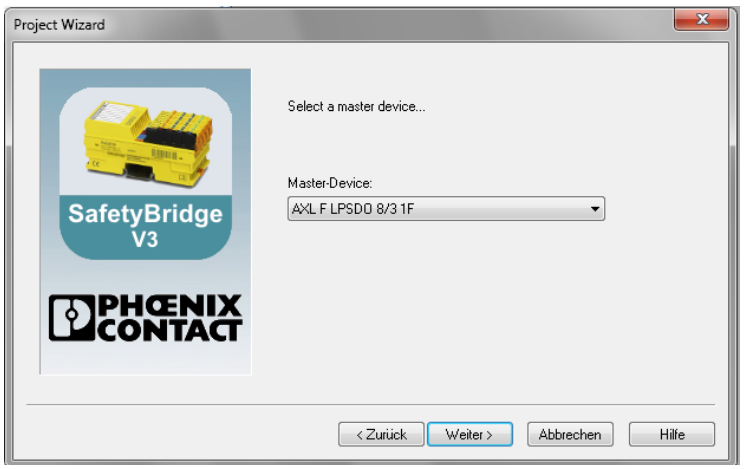



Figure 3-4 Selecting AXL F LPSDO8/3 1F

**Select file format**

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

Alternatively, select **“PC Worx”** to output the configuration and parameter data record as an **st file**. For additional information on this process, please refer to Appendix B on page 44.

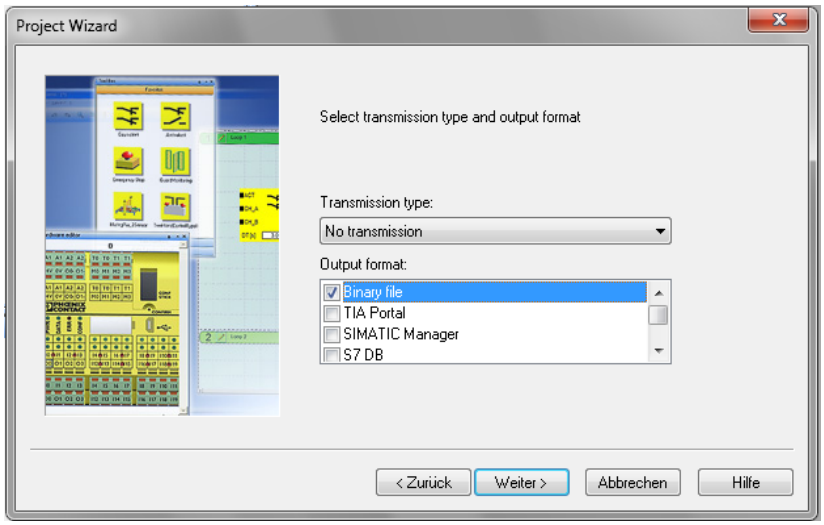



Figure 3-5 Selecting the output format

**Enter project description**

- Enter a description of the project.

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

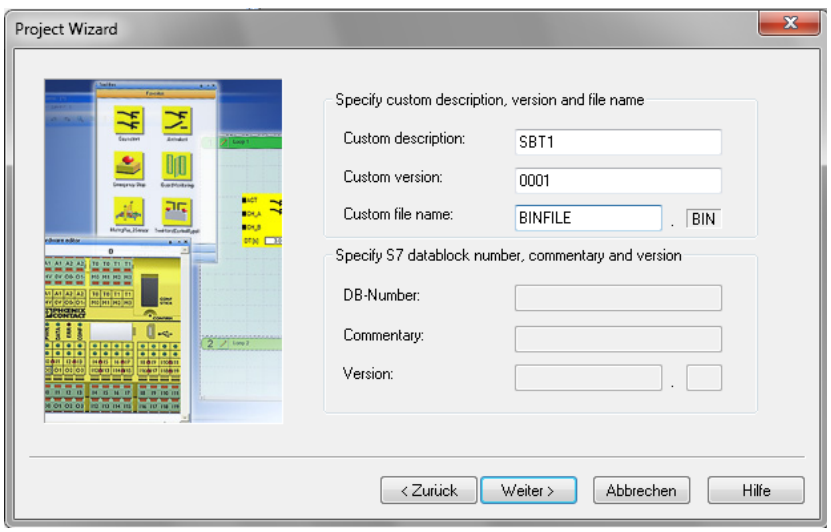



Figure 3-6 Describing the project

- 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).

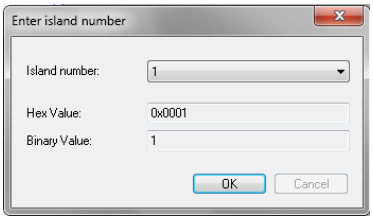



Figure 3-7 Specifying the island number

**Specify password**

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


Figure 3-8 Specifying a password

## 3.3.2 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.

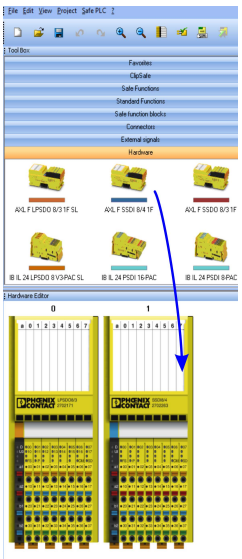


Figure 3-9 Hardware configuration

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”.

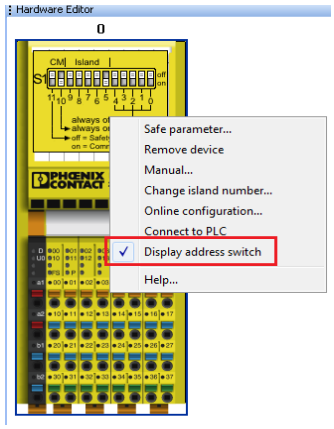



Figure 3-10 Displaying the DIP switch

## Parameterize I/O channels

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

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

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).

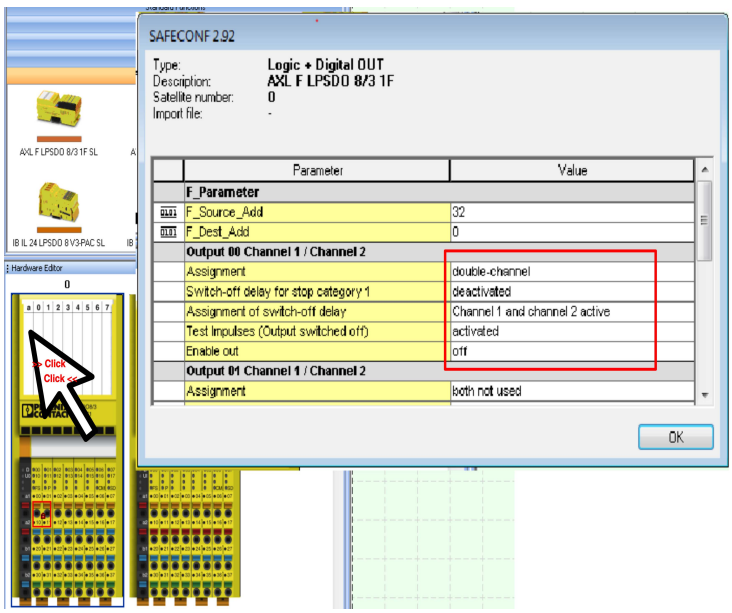



Figure 3-11 Parameterization of the LPSDO8/3

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

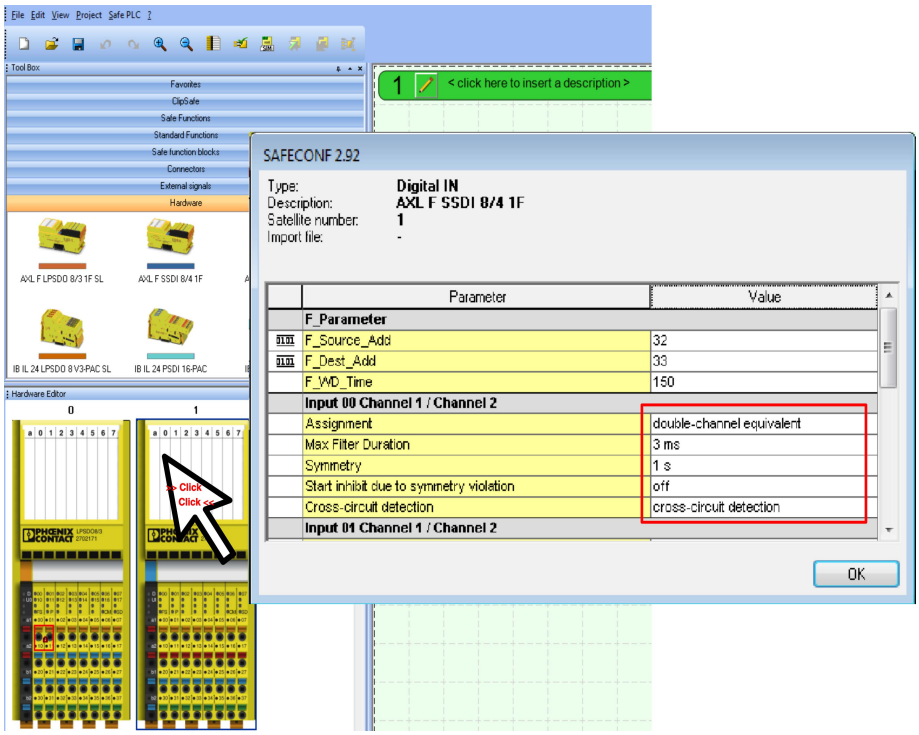



Figure 3-12 Parameterization of the SSDI8/4

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 3-14 on page 19.

### 3.3.3 Configuring the safety function

**Comment function**

You can add comments to both function blocks and signals in SAFECONF. 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.

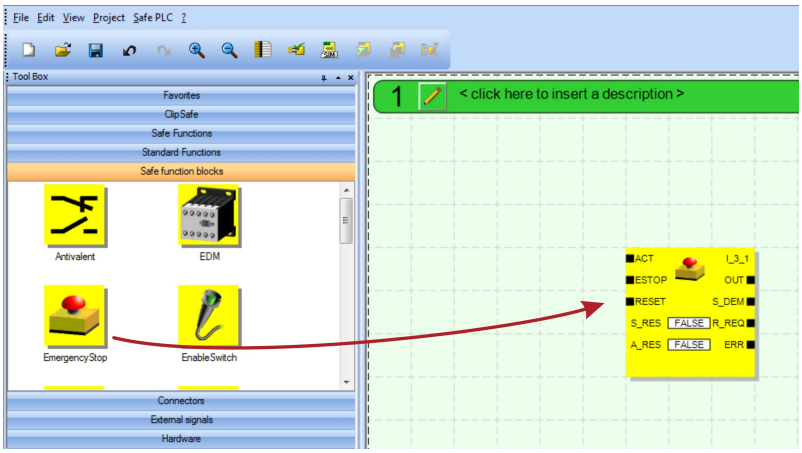



Figure 3-13 Inserting a function block from the “Safe function blocks” toolbox

**Insert safe inputs and outputs**

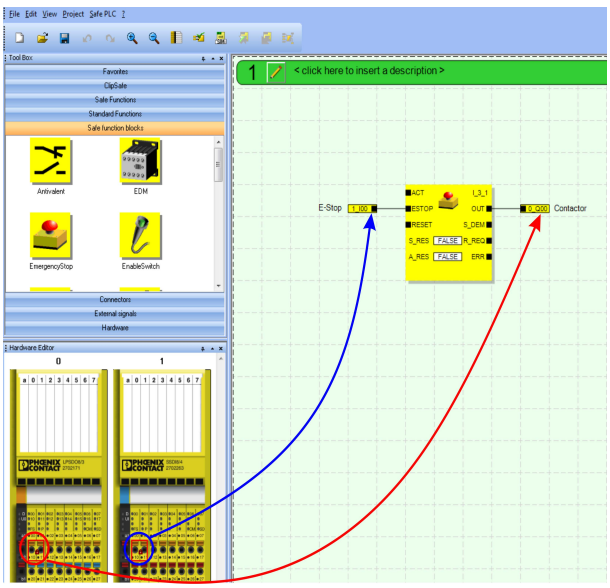



Figure 3-14 Inserting safe inputs and outputs from the hardware editor

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

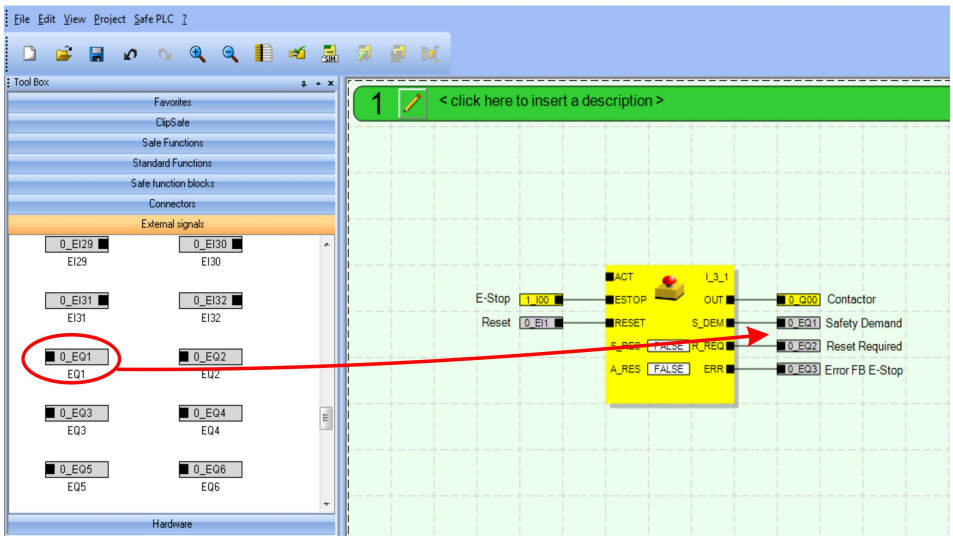
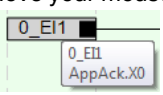



Figure 3-15 Inserting external signals from the “External signals” toolbox

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




### Insert safe functions

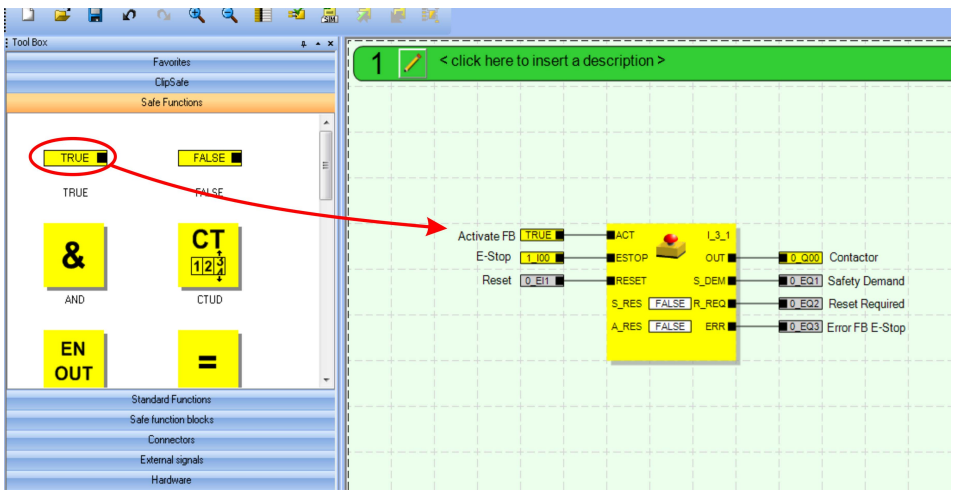



Figure 3-16 Inserting a safe function from the “Safe Functions” toolbox

### 3.3.4 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.

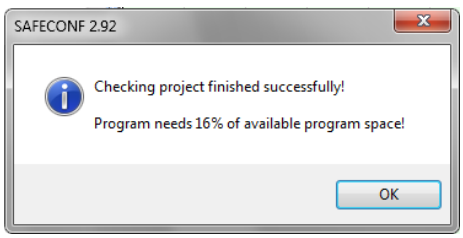



Figure 3-17 Program memory used

**BIN file for standard controller**

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 3-3 on page 14) in the “FileOutput” folder.


Figure 3-18 BIN file in the “FileOutput” folder

**st file for PC Worx (alternative)**

If you have selected **“PC Worx”** as the output format for the configuration and parameter data record (see Figure 3-5 on page 15), the **st file** is saved in the “FileOutput” folder. For additional information on this process, please refer to Appendix B on page 44.

## 3.4 Configuring a PC Worx project for AXC 1050

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

### 3.4.1 Creating a project and configuring the bus configuration

- Create a new project in PC Worx with the AXC 1050 controller.

**Insert modules**

- Select the following modules in the device catalog and use drag and drop to move them to the bus configuration under the INTERBUS node:
  - AXL F DI8/1 DO8/1 1H (Order No. 2701916)
  - AXL F LPSDO8/3 1F (Order No. 2702171)
  - AXL F SSDI8/4 1F (Order No. 2702263)

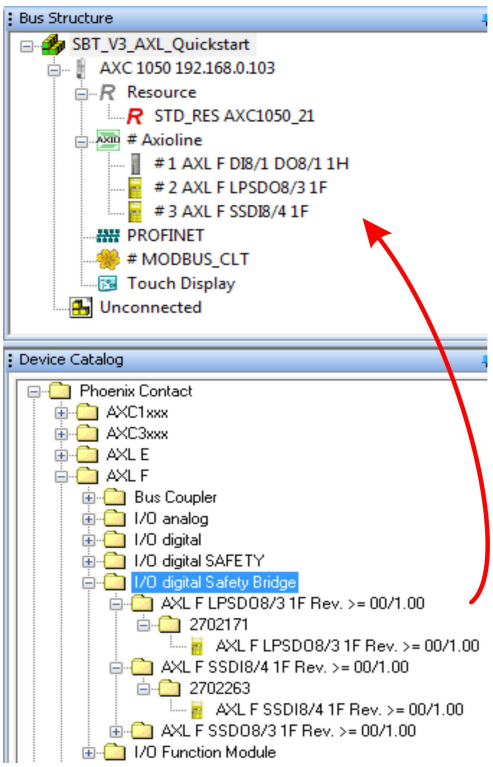



Figure 3-19 Inserting SBT modules in the device overview

**Specify IP address**

- Specify the device properties.

Table 3-2

| Node | Tab | Setting | Example |
|---|---|---|---|
| AXC 1050 | IP setting | IP address | 192.168.0.2 |

### 3.4.2 Integrating function blocks for SafetyBridge Technology V3

**Integrate library from integration package**

- Switch to “IEC Programming” view.
- In the project tree window, select “Libraries, Insert, User Library”.

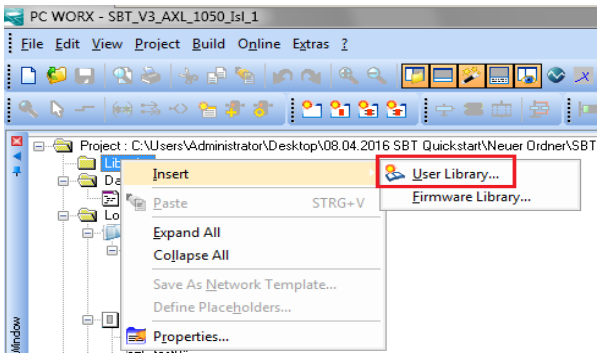



Figure 3-20 Integrating the library I

- Open the directory where the library files from the integration package for SafetyBridge Technology V3 were installed earlier.
  See page 11 in Section 3.1, “Download and installation of the Phoenix Contact software”.

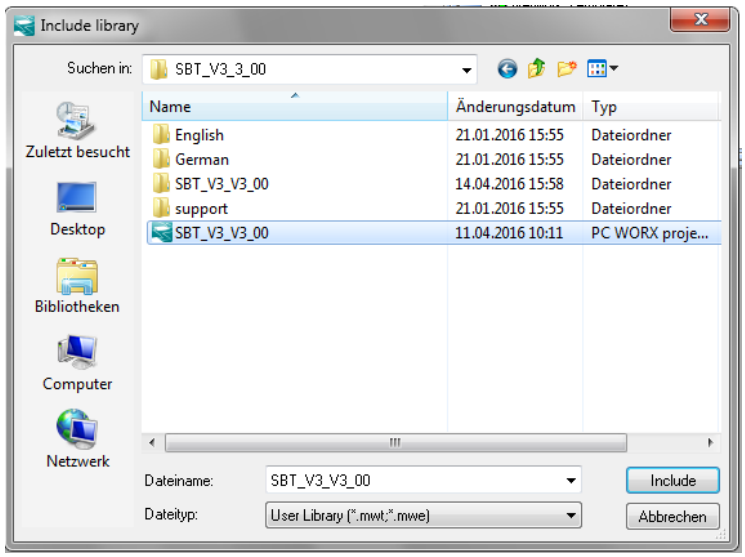



Figure 3-21 Integrating the library II

- Open file “SBT_V3_V3_00.mwt”.

The Edit Wizard window now contains the open “SBT_V3_V3_00” library with the corresponding SafetyBridge function blocks.

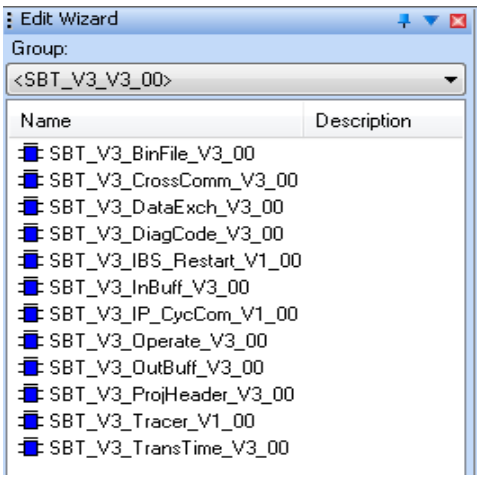



Figure 3-22 Library integrated

A description of the function blocks and data types is provided in Appendix C “Description of the function blocks for SafetyBridge Technology V3”.

Please also use the online help in PC Worx.

### 3.4.3 Creating the SBT program in PC Worx

Proceed as follows to extend your standard application program using the SafetyBridge function blocks from the “SBT_V3_V3_00” library:

**Insert logical POUs**

- In the project tree window, insert the “SBT_Island” program under “Logical POUs”.

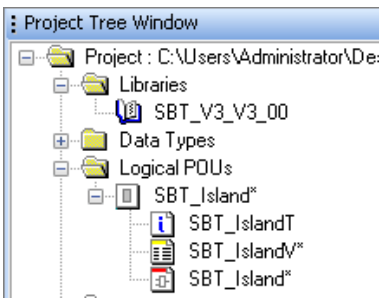



Figure 3-23 Inserting logical POUs

**Create cyclic task**

Proceed as follows to create a “cyclic task” in the hardware structure:

- Select the “Task” folder and then select “Task...” in the context menu.

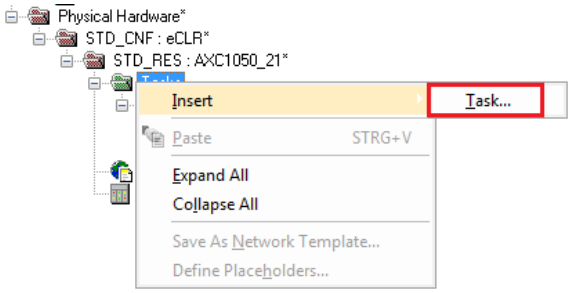



Figure 3-24 Inserting a cyclic task

- Specify “SBT_Run” for the task name and select “CYCLIC” as the task type.

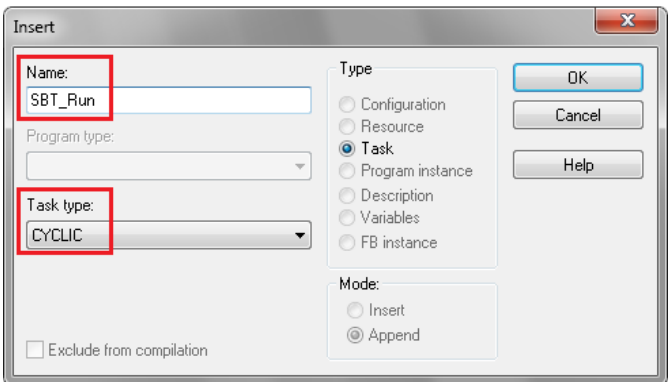



Figure 3-25 Assigning a name and selecting the task type

- In the following window for the task settings, set 10 ms in the “Interval” field.

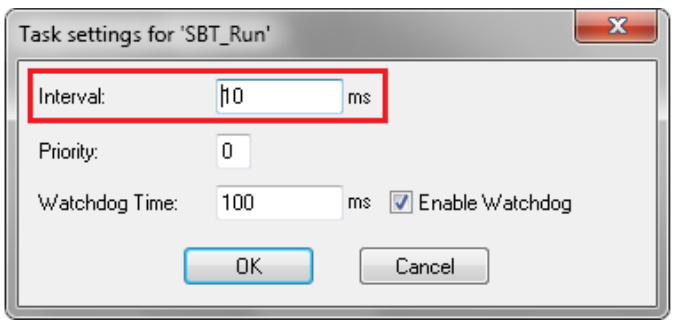



Figure 3-26 Making the task settings

The created task prevents potential communication interruptions by periodically calling the SafetyBridge function blocks at fixed intervals.

### Add “SBT_Island” program instance

Add the “SBT_Island” program instance to the task as follows:

- Select the “SBT_Run : CYCLIC” task and then select “Insert, Program instance...” in the context menu.

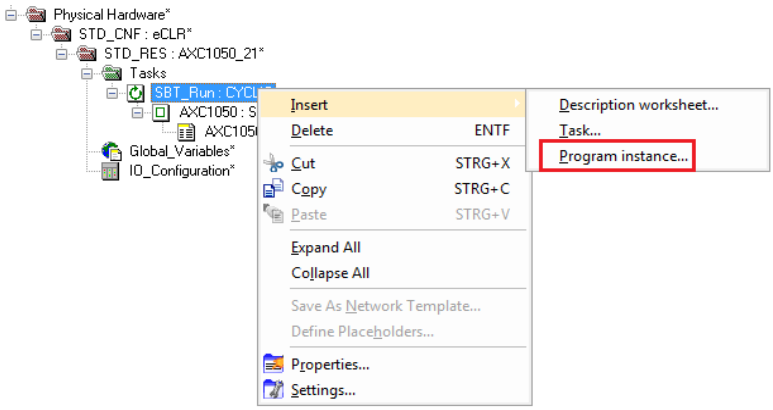



Figure 3-27 Inserting the program instance

- Enter “AXC 1050” in the “Program instance” field and select “SBT_Island” as the program type.

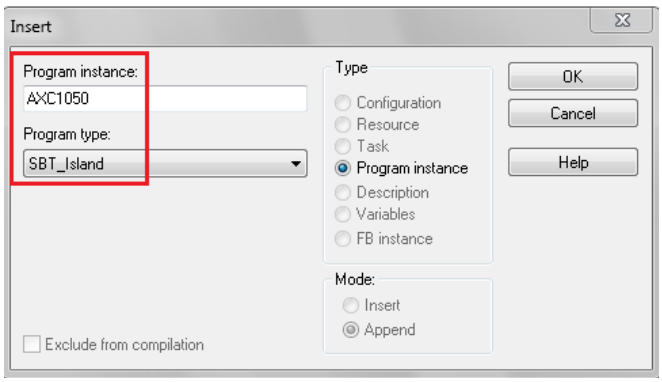



Figure 3-28 Assigning a name and selecting the program type

### Complete project tree

The complete project tree now has the following structure.

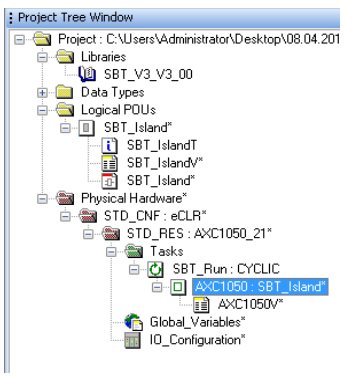



Figure 3-29 Complete project tree

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

**“SBT_V3_Operate_V3_00” function block**

- Select the “SBT_Island” program.
- Insert the “SBT_V3_Operate_V3_00” function block in the program using drag and drop.
- Connect the function block as shown in Figure 3-30.

Observe the notes regarding the connection of the “xActivate”, “iIslandNo”, “arrSBTOnlCntrlBuf”, and “arrSBTOnlValBuf” parameters (see page 28).

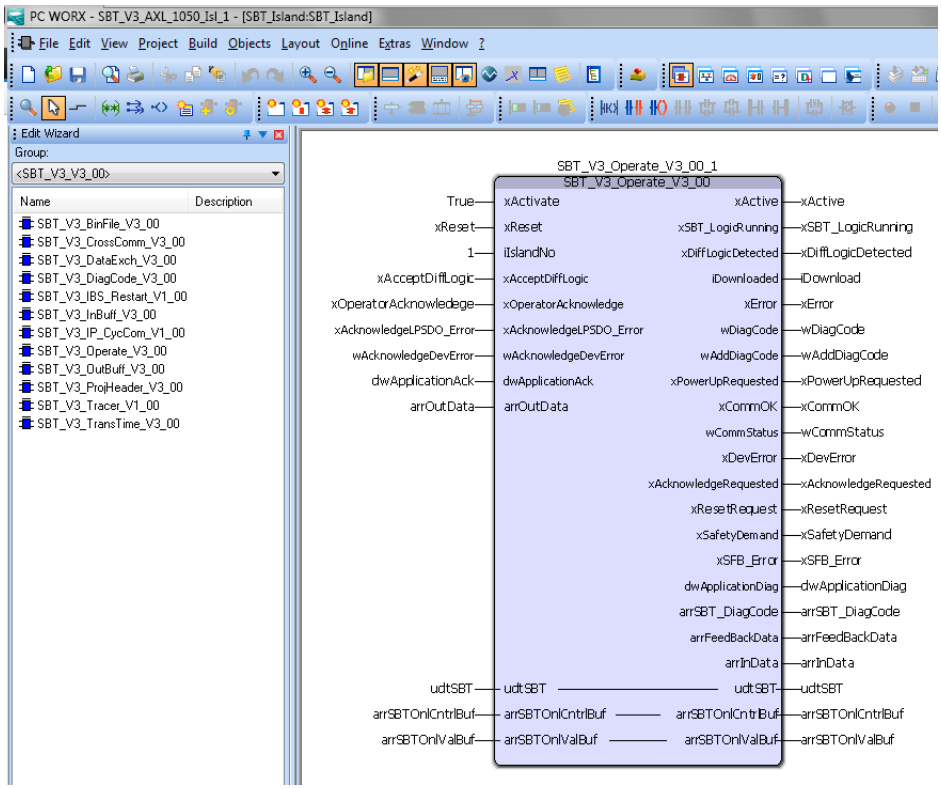



Figure 3-30 “SBT_V3_Operate_V3_00” function block

**Notes regarding the connection of I/O parameters:**

**xActivate**

Connect the “xActivate” input parameter to the “True” constant so that the “SBT_V3_Operate_V3_00” function block is activated automatically.

**iIslandNo**

Set the “iIslandNo” input parameter to “1” as per the island number selected in SAFECONF. See “Assign safety island number” on page 16.

**arrSBTOnlCntrlBuf**

For this input parameter, assign the exact same variable name (“arrSBTOnlCntrlBuf”) as follows:

1. Open the “Variable Properties” window.
2. In the “Global Variable Groups” field, select the “Default” folder.
3. Select “VAR_GLOBAL” as the usage.
4. Select the “PDD” function via the check box.

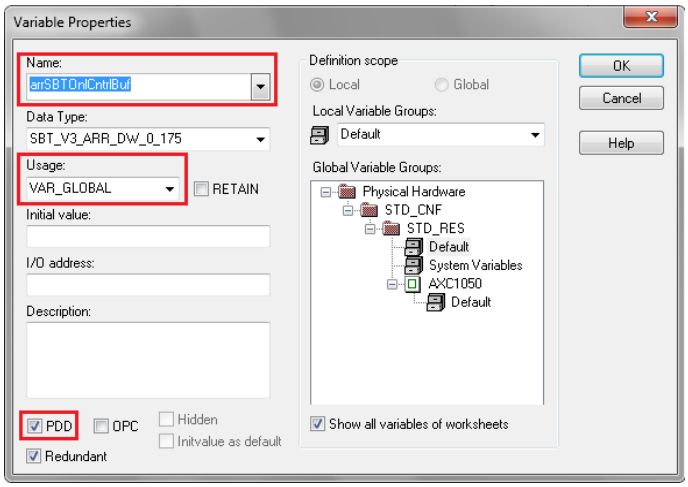



Figure 3-31 Settings for variable “arrSBTOnlCntrlBuf”

**arrSBTOnlValBuf**

Proceed as described above for the “arrSBTOnlCntrlBuf” input parameter.

The “arrSBTOnlCntrlBuf” and “arrSBTOnlValBuf” I/O parameters are required for the online mode function in SAFECONF.

**“SBT_V3_InBuff_V3_00” function block**

- Insert the “SBT_V3_InBuff_V3_00” function block in the program using drag and drop.
- Connect the function block as shown in Figure 3-32.

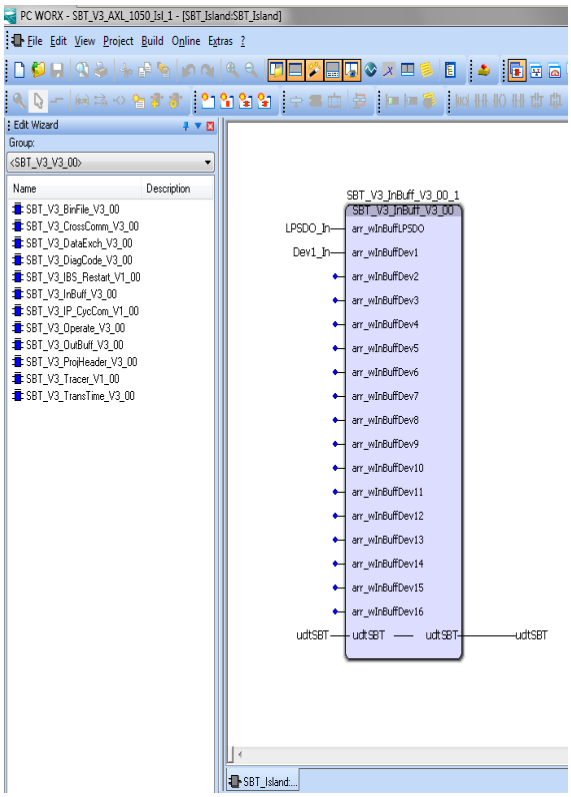



Figure 3-32 “SBT_V3_InBuff_V3_00” function block

- Proceed as follows for variables “LPSDO_In” und “Dev1_In”:

1. Open the “Variable Properties” window.
2. Select “VAR_GLOBAL_PG” as the usage.

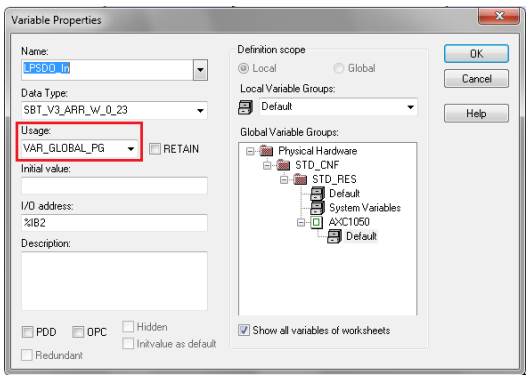



Figure 3-33 Settings for variables “LPSDO_In” and “Dev1_In”

- Link the “udtSBT” structure variable to the same variable as for the “SBT_V3_Operate_V3_00” function block.

**“SBT_V3_OutBuff_V3_00” function block**

- Insert the “SBT_V3_OutBuff_V3_00” function block in the program using drag and drop.
- Connect the function block as shown in Figure 3-34.

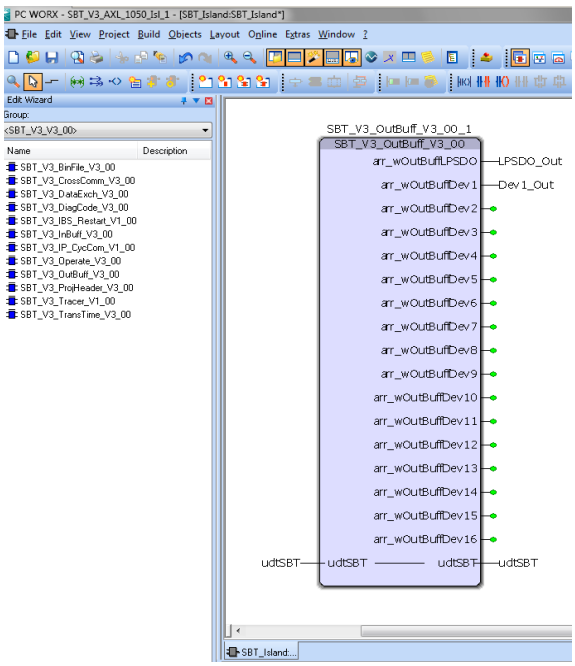



Figure 3-34 “SBT_V3_OutBuff_V3_00” function block

- Proceed as follows for variables “LPSDO_Out” und “Dev1_Out”:

1. Open the “Variable Properties” window.
2. Select “VAR_GLOBAL_PG” as the usage.

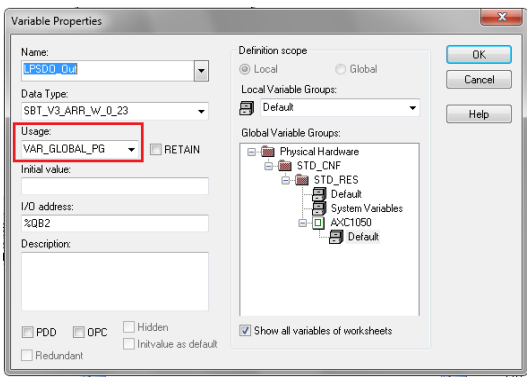



Figure 3-35 Settings for variables “LPSDO_Out” and “Dev1_Out”

- Link the “udtSBT” structure variable to the same variable as for the “SBT_V3_Operate_V3_00” function block.

### Observe the sequence of the function blocks

Now arrange the function blocks in the following sequence in the “SBT_Island” program:

1. “SBT_V3_InBuff_V3_00” function block
2. “SBT_V3_Operate_V3_00” function block
3. “SBT_V3_OutBuff_V3_00” function block

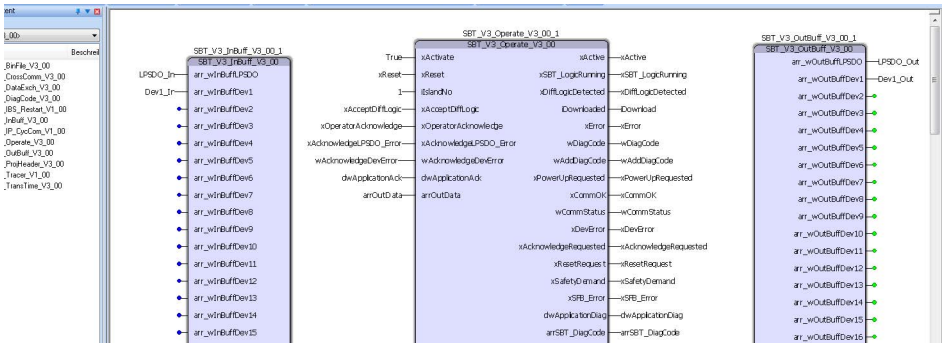



Figure 3-36 Sequence of the function blocks in the program

### “SBT_V3_BinFile_V3_00” function block

This function block is only required if you selected **“Binary file”** as the output format for the configuration and parameter data record from SAFECONF (see Figure 3-5 on page 15) and wish to download the BIN file to the controller via FTP.

If you selected **“PC Worx”** as the output format, this function block is not required. For additional information on this process, please refer to Appendix B on page 44.

- Insert the “SBT_V3_BinFile_V3_00” function block in the program using drag and drop.
- Connect the function block as shown in Figure 3-37.
- Link the “udtSBT” structure variable to the same variable as for the “SBT_V3_Operate_V3_00” function block.

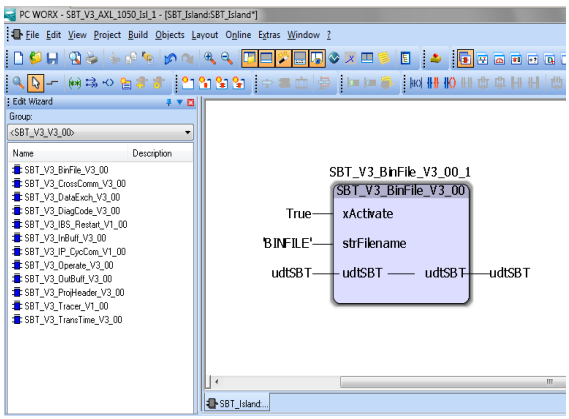



Figure 3-37 “SBT_V3_BinFile_V3_00” function block

**“SBT_V3_ProjHeader_V3_00” function block**

- Insert the “SBT_V3_ProjHeader_V3_00” function block in the program using drag and drop.
- Connect the function block as shown in Figure 3-38.
- Link the “udtSBT” structure variable to the same variable as for the “SBT_V3_Operate_V3_00” function block.

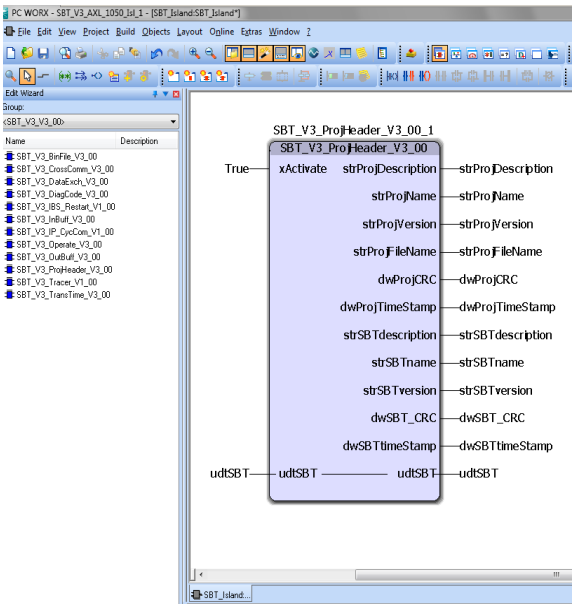



Figure 3-38 “SBT_V3_ProjHeader_V3_00” function block

**“SBT_V3_TransTime_V3_00” function block**

- Insert the “SBT_V3_TransTime_V3_00” function block in the program using drag and drop.
- Connect the function block as shown in Figure 3-39.
- Link the “udtSBT” structure variable to the same variable as for the “SBT_V3_Operate_V3_00” function block.

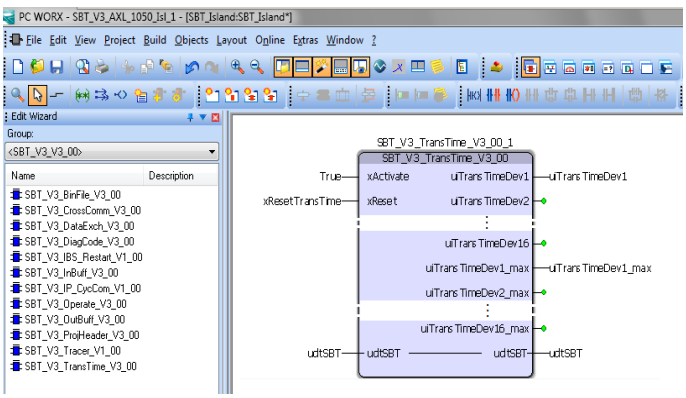



Figure 3-39 “SBT_V3_TransTime_V3_00” function block

### “SBT_V3_DiagCode_V3_00” function block

- Insert the “SBT_V3_DiagCode_V3_00” function block in the program using drag and drop.
- Connect the function block as shown in Figure 3-40.
- Link the “udtSBT” structure variable to the same variable as for the “SBT_V3_Operate_V3_00” function block.
- Link the “arr_wSBTDiagCode” array to the same variable as for the “SBT_V3_Operate_V3_00” function block.

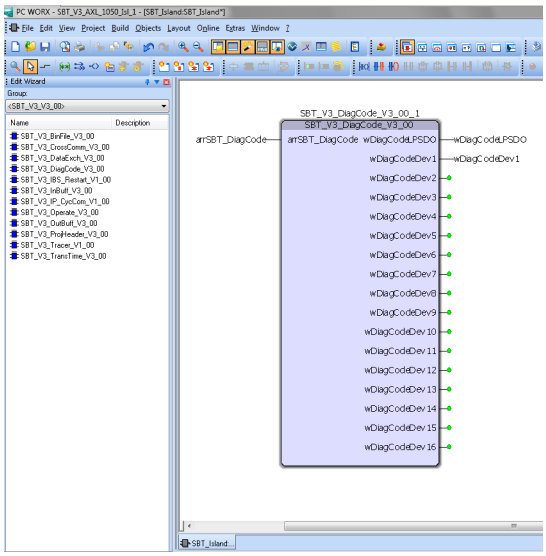



Figure 3-40 “SBT_V3_DiagCode_V3_00” function block

**Error acknowledgment**

- Create the program shown in Figure 3-41 in order to acknowledge communication and module errors for the AXL F LPSDO8/3 1F and AXL F SSDI8/4 1F modules.

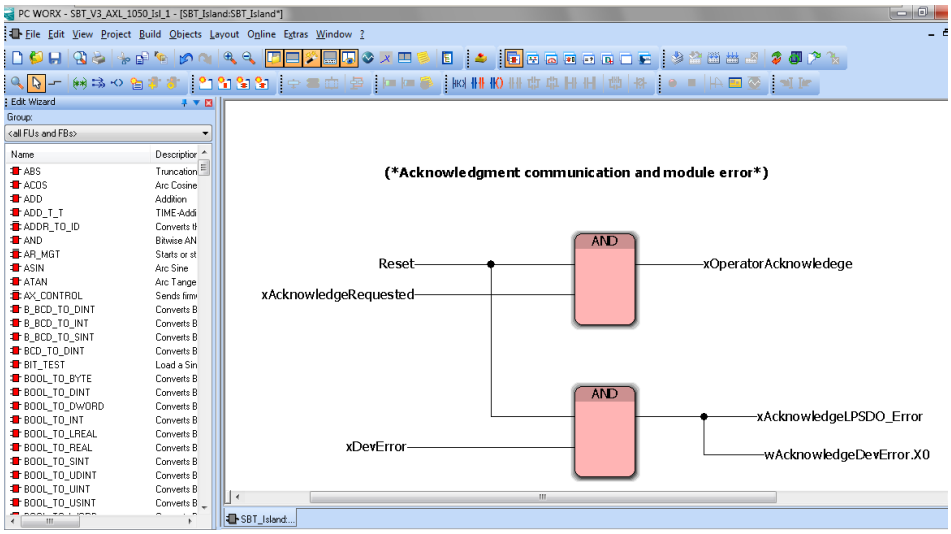



Figure 3-41 Program for error acknowledgment

Communication and module errors can be acknowledged via the “Reset” variable.

### Acknowledgment of the safety function

- Create the program shown in Figure 3-42 in order to acknowledge the safety function.

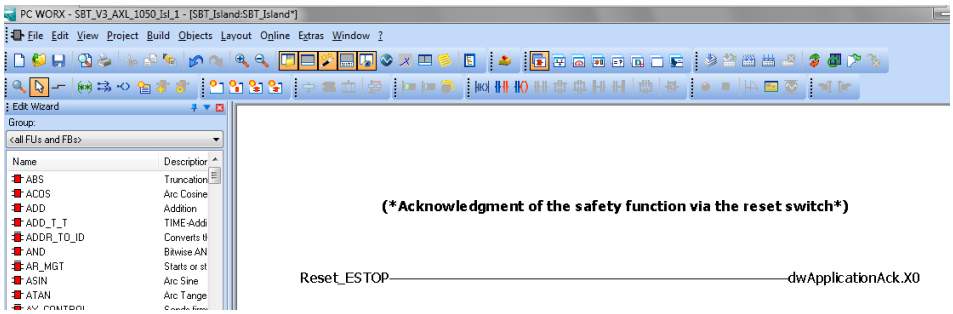
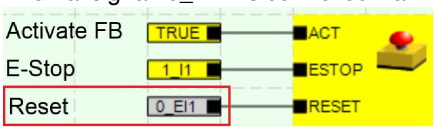



Figure 3-42 Program for acknowledging the safety function

External signal “0_EI1” is controlled via the “Reset_ESTOP” input in SAFECONF.




### Signal lamp for reset request

- Create the program shown in Figure 3-43 so that the reset request can be indicated via the signal lamp.

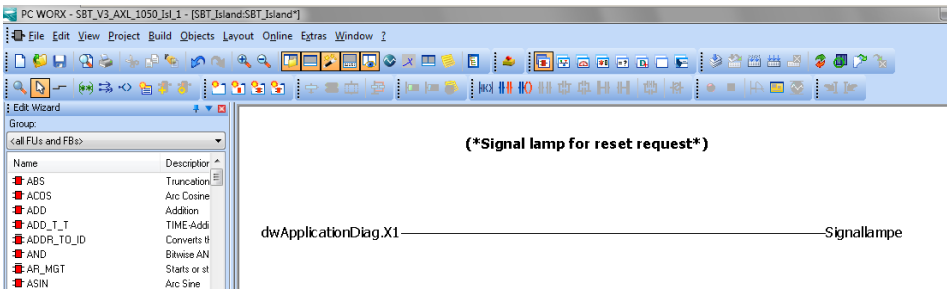
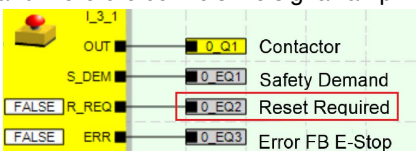



Figure 3-43 Program for the signal lamp for reset request

External signal “0_EQ2” from SAFECONF writes the “dwApplicationDiag.X1” variable and therefore controls the signal lamp.



### 3.4.4 Process data assignment of the devices

- Open the process data assignment window in PC Worx. 
- Select the “AXC 1050 : SBT_Island” program instance and the AXL F LPSDO8/3 1F module.

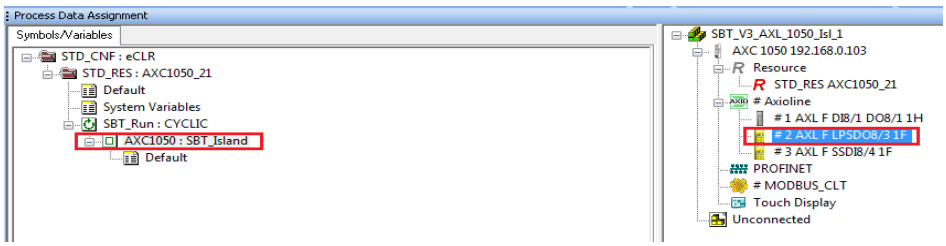



Figure 3-44 Process data assignment

- Connect the control system variables to the process data for the AXL F LPSDO8/3 1F in the bus configuration by carrying out the following steps:

1. Place the “LPSDO_Out” variable on the “arrOutBuff” process data item.
2. Place the “LPSDO_In” variable on the “arrInBuff” process data item.

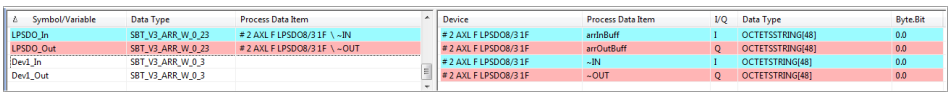



Figure 3-45 LPSDO8/3 – connecting variables and process data

- Select the AXL F SSDI8/4 1F module.
- Connect the control system variables to the process data for the AXL F SSDI8/4 1F in the bus configuration by carrying out the following steps:

1. Place the “Dev1_In” variable on the “arrInBuff” process data item.
2. Place the “Dev1_Out” variable on the “arrOutBuff” process data item.

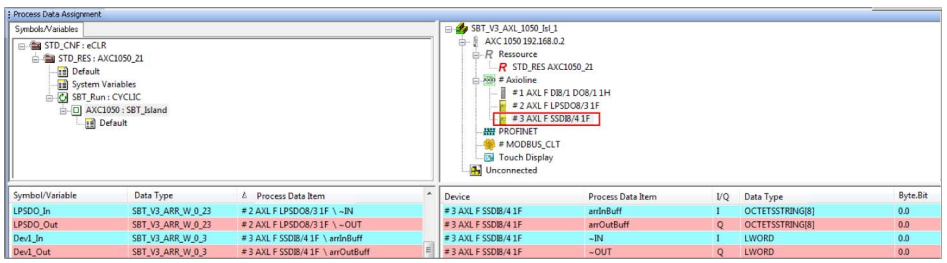



Figure 3-46 SSDI8/4 – connecting variables and process data

- Select the AXL F DI8/1 DO8/1 1H module.
- Connect the control system variables to the “Reset”, “Reset_Estop”, and “Signal lamp” process data for the AXL F DI8/1 DO8/1 1H in the bus configuration.

### 3.4.5 Compiling the project and downloading it to the controller

- Compile the created project and download it to the controller.

The controller switches to the “Run” state.
The LPSDO8/3 module indicates an error by flashing the FS LED.
Error message: device is not parameterized or parameterization was not applied.

### 3.4.6 Importing the configuration and parameter data record into the project

This step is only required if you selected **“Binary file”** as the output format for the configuration and parameter data record from SAFECONF (see Figure 3-5 on page 15).

If you selected **“PC Worx”** as the output format, this step is not required. For additional information on this process, please refer to Appendix B on page 44.

**Transmit BIN file**

Download the configuration and parameter record stored as a BIN file to the controller via SAFECONF. Proceed as follows:

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

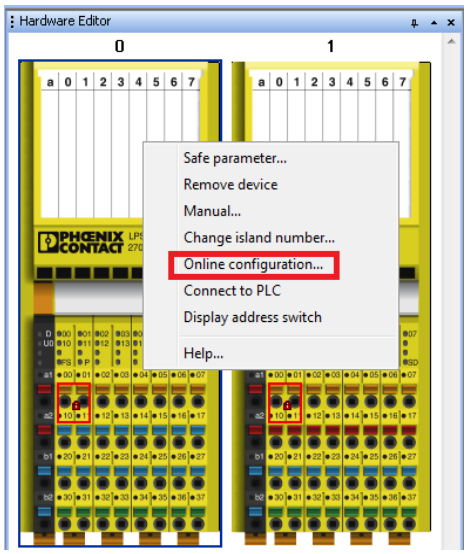



Figure 3-47 Selecting online configuration

- Select the “Phoenix Contact (ILC 1xx, ILC 3xx, RFC 4xx)” interface and click on “Next”.

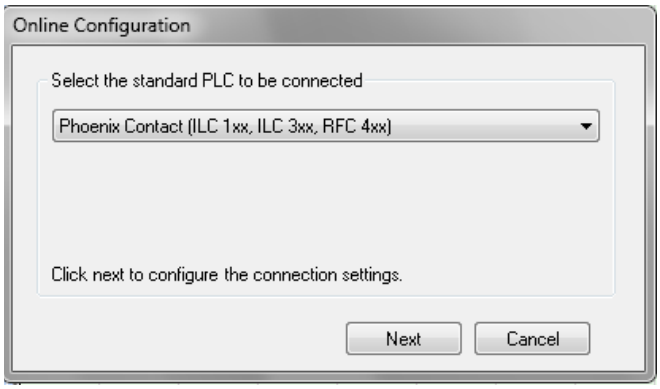



Figure 3-48 Selecting the interface

The “Online Configuration” window opens.

- Enter the IP address of the AXC 1050 controller and click on “Test connection...”.
- Once this test has been completed successfully, click on “Finish”.

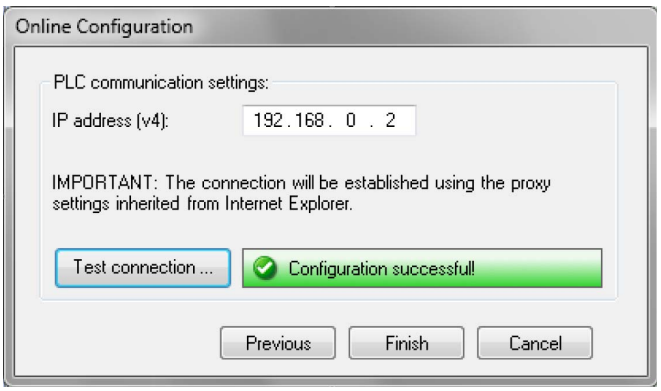



Figure 3-49 SAFECONF “Online Configuration” window

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

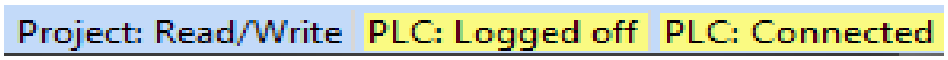



Figure 3-50 SAFECONF status message

- Click on the “Download” button. 

Following successful download, the following information window appears.

- Confirm with “OK”.

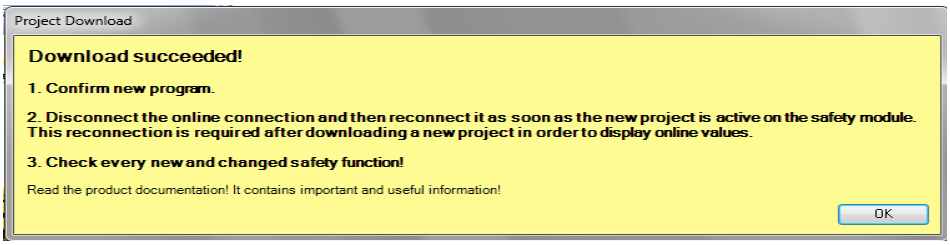



Figure 3-51 SAFECONF information window

See Section “Online configuration and connection establishment” on page 41.

**Important when making changes in SAFECONF**

Whenever a change is made in the SAFECONF project which affects the checksum (CRC), replace the BIN file in the root directory of the controller. Proceed as described above.

## 3.5 Startup

A flowchart for starting up and testing the application can be found in Appendix A on page 43.

1. Switch to PC Worx.
2. Open the “SBT_Island” program.
3. Activate the online values in PC Worx.
4. Set the “xReset” input parameter to “1” and 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 “iDownloaded” output parameter. Download time: approximately 40 seconds (depending on the project size, CPU, and bus speed).

6. Check whether the “xSBTlogicRunning” output parameter outputs “TRUE” and the “wDiagCode” output parameter outputs the value 16#8000.

If a different code is output by the “wDiagCode” output parameter, see “Function block diagnostics” on page 53.

Once successfully downloaded, the safe application is ready.

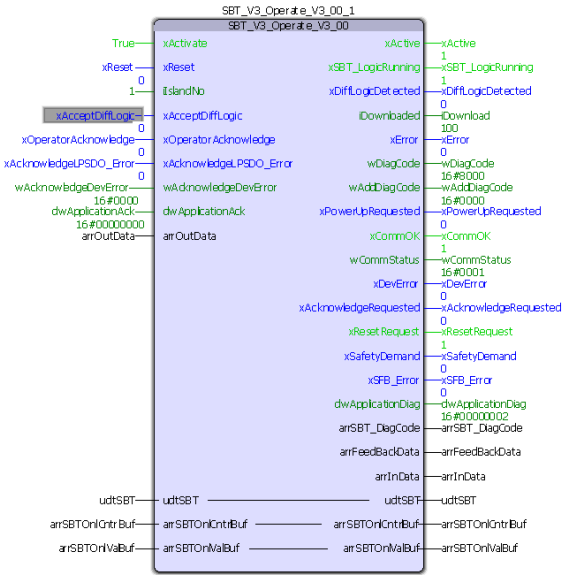



Figure 3-52 Finished project

The diagnostics LEDs of the modules now have the following status:

LEDs **off**: FS, CM, SD
LEDs **on**: P, $U_O$, D

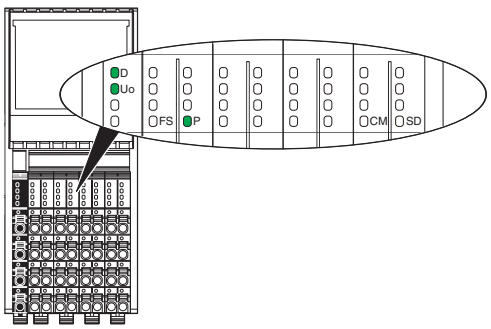



Figure 3-53 Diagnostics LEDs

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

## 3.6 Online configuration and connection establishment

Communication between SAFECONF and the “SBT_V3_Operate_V3_00” function block is achieved by the “arrSBTOnlCntrlBuf” and “arrSBTOnlValBuf” 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”.

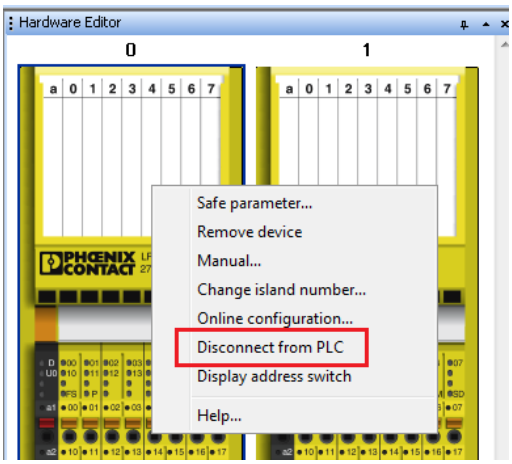



Figure 3-54 Disconnecting an online connection

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

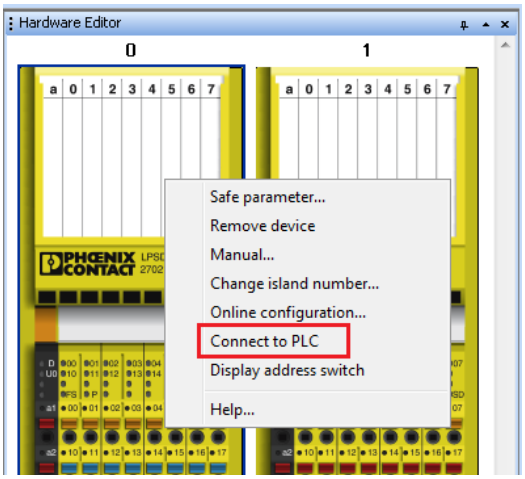



Figure 3-55 Establishing an online connection

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

The online values are now displayed in the SAFECONF project.

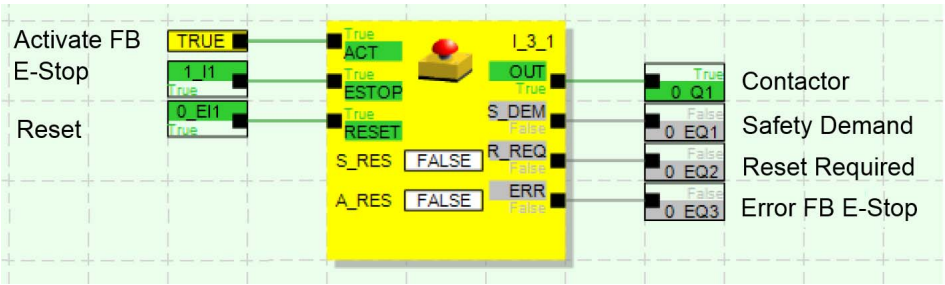



Figure 3-56 Online values in the SAFECONF project

# A Flowchart for starting up and testing the application

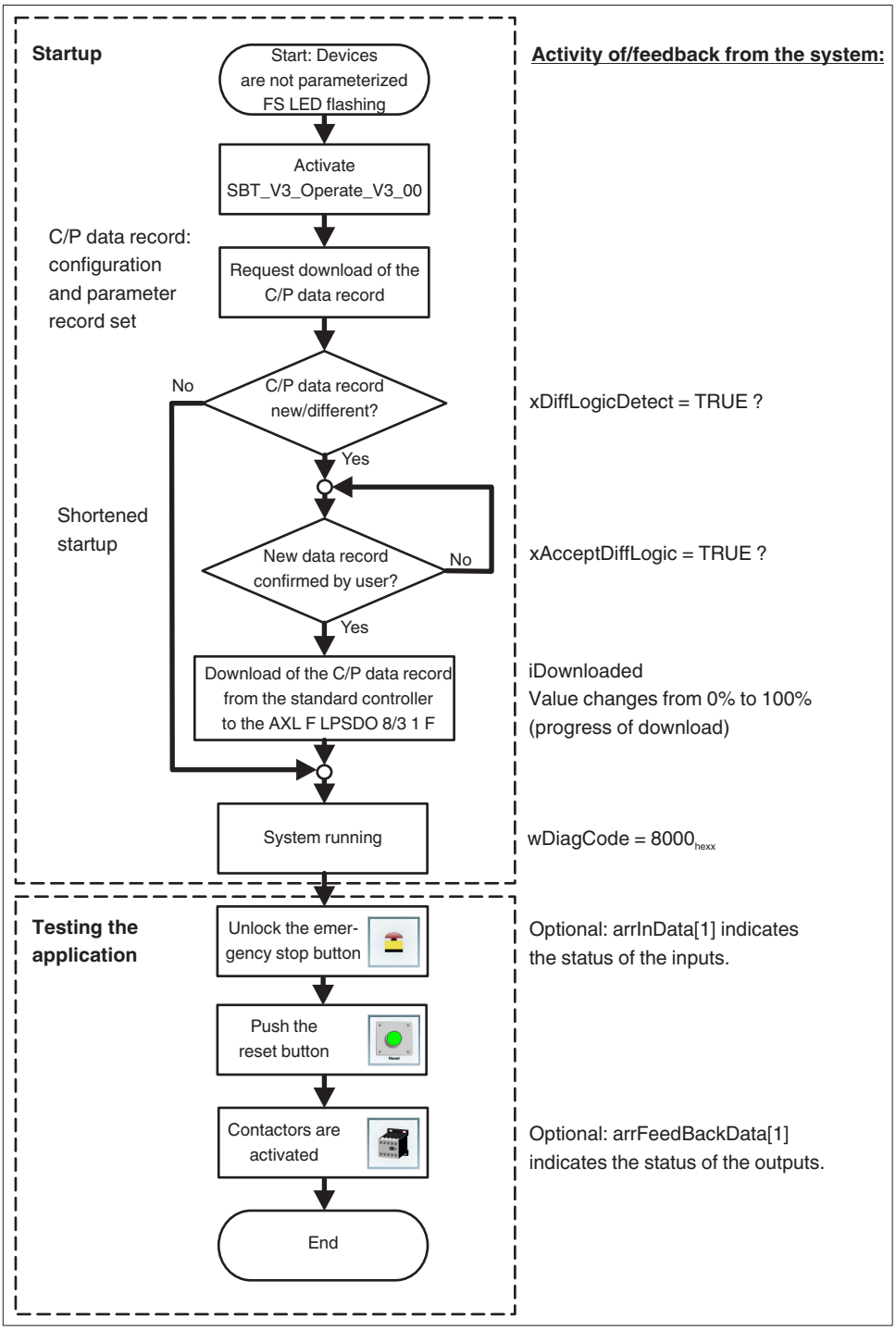



Figure A-1 Flowchart for starting up and testing the application

# B Alternatives for importing the configuration and parameter data record

## B 1 Copying the BIN file to the controller via FTP

**Requirement**

**“Binary file”** was selected as the output format for the configuration and parameter data record from SAFECONF (see Figure 3-5 on page 15).

In “Importing the configuration and parameter data record into the project” on page 37, the BIN file was created and saved in the “FileOutput” folder under the previously specified project path (see Figure 3-18 on page 21).

**Transmit BIN file**

- In Explorer, establish a connection to the controller via FTP (e.g., ftp://192.168.0.2).
- Copy the BIN file to the root directory of the controller.

**Important when making changes in SAFECONF**

Whenever a change is made in the SAFECONF project which affects the checksum (CRC), replace the BIN file in the root directory of the controller. Proceed as described above.

## B 2 Importing the st file in PC Worx

**Requirement**

**“PC Worx”** was selected as the output format for the configuration and parameter data record from SAFECONF (see Figure 3-5 on page 15).

In “Importing the configuration and parameter data record into the project” on page 37, the st file was created and saved in the “FileOutput” folder under the previously specified project path (see Figure 3-18 on page 21).

**Transmit st file**

- Open your project in PC Worx.
- Make sure that you are offline.
- Select the “Logical POU” folder.
- Select "File, Import".
- Select the “Extended IEC 61131-3 Import” entry and confirm your selection with “OK”.

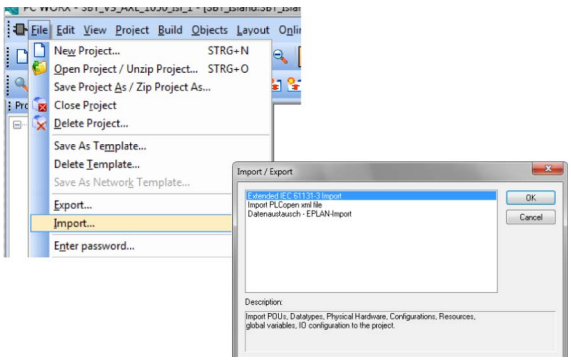



Figure B-1 Extended IEC 61131-3 import

- Select the POU object and confirm your selection with “OK”.

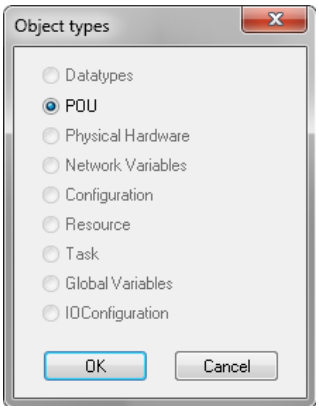



Figure B-2 POU object

- Select the st file and confirm your selection with “OK”.

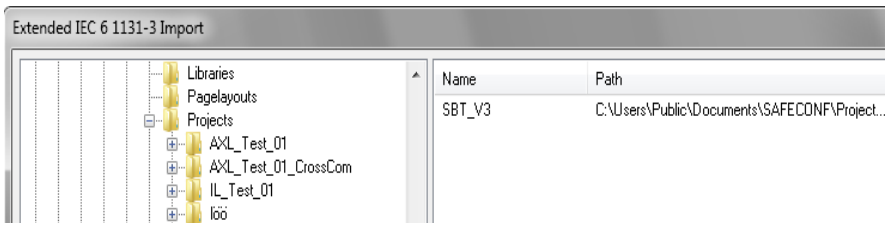



Figure B-3 Selecting the st file

If import is successful, the structure for the SafetyBridge islands (SBTIslands) is created under “Logical POUs”.

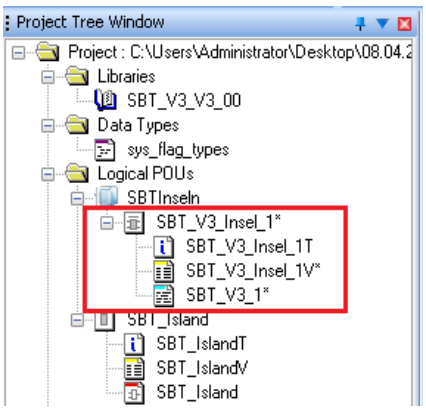



Figure B-4 POU for SafetyBridge island 1 inserted

If an error message appears and the structure is not created, this may because the name of the file you want to import is too long.

- If this is the case, give the SAFECONF project a shorter name.

- Select the “SBT_V3” function block from the library and move it to the programming view using drag and drop.
- Connect the function block as shown in Figure B-5.

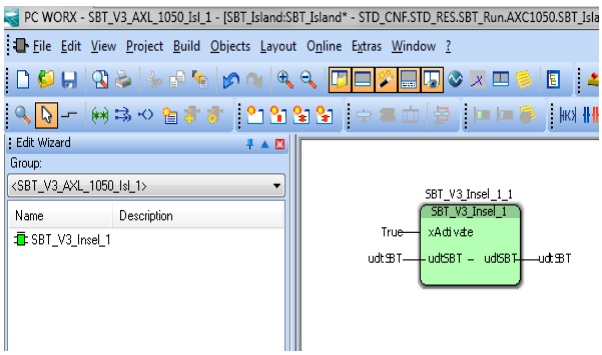



Figure B-5 Connecting the function block

**Important when making changes in SAFECONF**

Whenever a change is made in the SAFECONF project which affects the checksum (CRC), import a new st file in PC Worx. Proceed as described above.

# C Description of the function blocks for SafetyBridge Technology V3

For additional information, please refer to the online help for the PC Worx software.

The described function blocks are valid for the following controllers and software versions.

**Validity**

Table C-1 Validity of the function blocks

| | | | | |
|---|---|---|---|---|
| **Controller** | All controllers from Phoenix Contact | | | |
| **Bus system** | Inline | | Axioline | |
| | INTERBUS | PROFINET | AXIOBUS | PROFINET |
| **Software** | PC Worx V6.30 or later | | | |

**Function block overview**

Table C-2 Function block overview

| Function block | Description |
|---|---|
| SBT_V3_Operate_V3_00 | Function block used to operate an SBT island with a maximum of 16 satellites<br>– Download the SAFECONF project<br>– Monitoring of input and output modules<br>See C 1 on page 49 |
| SBT_V3_InBuff_V3_00 | Links the IN process data of the modules<br>See C 2 on page 54 |
| SBT_V3_OutBuff_V3_00 | Links the OUT process data of the modules<br>See C 3 on page 55 |
| SBT_V3_BinFile_V3_00 | The function block reads the project data of the BIN file generated from SAFECONF and makes the data available to the SBT_V3_Operate_V3_00 function block<br>See C 4 on page 56 |
| SBT_V3_CrossComm_V3_00 | Cross communication between two or more SBT islands<br>See C 5 on page 57 |
| SBT_V3_DataExch_V3_00 | Data exchange if two SBT islands are integrated in different controllers<br>See C 6 on page 58 |
| SBT_V3_ProjHeader_V3_00 | Displays information about the SAFECONF project and the project stored on the LPSDO module as output data<br>See C 8 on page 62 |

| Function block | Description |
|---|---|
| SBT_V3_TransTime_V3_00 | Indicates the data transmission times for each module<br>See C 7 on page 60 |
| SBT_V3_DiagCode_V3_00 | Displays the contents of the array of SBT module diagnostic codes (device-specific; see module user documentation)<br>Also indicates the value of the DIP switches if no diagnostic code is present and the modules are not parameterized<br>See C 9 on page 64 |
| SBT_V3_Tracer_V1_00 | If a communication error has occurred, the block displays the last four telegrams between the logic module and the satellites<br>See C 10 on page 66 |

**Data types**

Table C-3 Data types

| Designation | Description |
|---|---|
| SBT_V3_w_0_16 | Internal user-defined data type. For additional information, refer to the descriptions for the respective function block parameters. |
| SBT_V3_ARR_W_0_16 | |
| SBT_V3_ARR_W_0_175 | |
| SBT_V3_ARR_W_0_23 | |
| SBT_V3_ARR_W_0_3 | |
| SBT_V3_ARR_DW_0_175 | |
| SBT_V3_UDT_SBT_V2 | |
| SBT_V3_ARR_UDT_SBT_V2 | |
| SBT_V3_Data_2 | |
| SBT_V3_Tracer | |

# C 1 SBT_V3_Operate_V3_00 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

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.

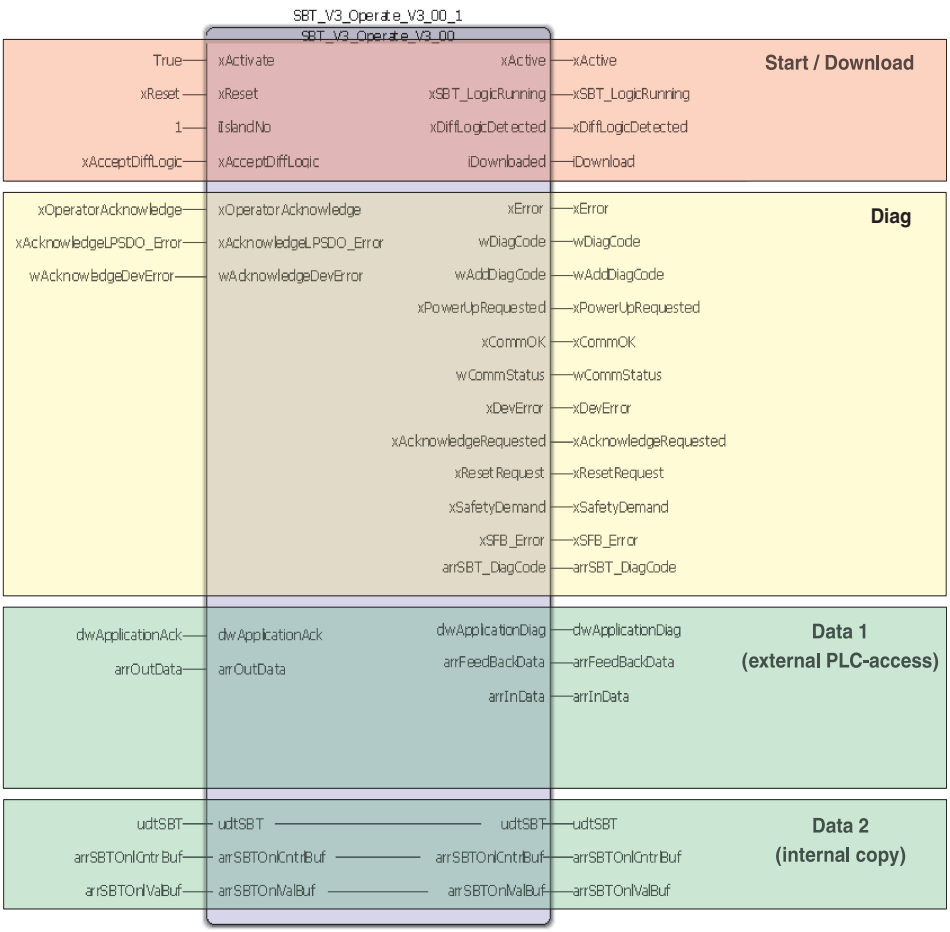



Figure C-1 “SBT_V3_Operate_V3_00” function block

## C 1.1 Input parameters

Table C-4 Input parameters for the “SBT_V3_Operate_V3_00” function block

| Name | Type | Description |
|---|---|---|
| xActivate | BOOL | Activation/deactivation of the function block |
| xReset | BOOL | On a rising edge, the function block is reset and re-initialized |
| iIslandNo | INT | SBT island number<br>Valid range: 1 ... 31 (must be the same as the DIP switch position on the LPSDO module and the setting in the SAFECONF project) |
| xAcceptDiffLogic | BOOL | User confirmation signal that the safety logic differs for the controller and the LPSDO module<br>Rising edge: user has accepted and confirmed the difference (the download then starts) |
| xOperatorAcknowledge | BOOL | Rising edge: user acknowledgement of a sporadic communication error |
| xAcknowledgeLPSDO_Error | BOOL | Rising edge: acknowledgment of an LPSDO module error |
| wAcknowledgeDevError | WORD | Each bit is associated with a module<br>If a bit is set, an error is acknowledged at the corresponding module |
| dwApplicationAck | DWORD | Freely configurable acknowledgment signals (application data) from the standard controller to the LPSDO module |
| arrOutData | SBT_V3_w_0_16 | Array of outputs from all output modules of the SBT island; the index represents the device number; index 0 stands for the LPSDO module<br>Info: depending on the parameterization of the safe outputs, the enable principle applies here. I.e., the safe outputs can be controlled via the standard controller as long as the safety function is not demanded (the safety logic is “enabled”). If there is a demand of the safety function, this overrides the standard controller and switches off the safe outputs. |

## C 1.2 Output parameters

Table C-5 Output parameters for the “SBT_V3_Operate_V3_00” function block

| Name | Type | Description |
|---|---|---|
| xActive | BOOL | True: the function block has been initialized successfully and is operating without errors<br>False: error or function block is not initialized |
| xSBT_LogicRunning | BOOL | Safety logic (SAFECONF logic) running on the LPSDO module |
| xDiffLogicDetected | BOOL | True: detection of different safety logic for the controller and the LPSDO module |
| iDownloaded | INT | Percentage of SBT logic that has been downloaded |
| xError | BOOL | True: the function block has detected an error and is no longer operating; re-activation is required |
| wDiagCode | WORD | Diagnostic code for function block errors |
| wAddDiagCode | WORD | Additional diagnostic code for function block errors |
| xPowerUpRequested | BOOL | Error state that cannot be acknowledged; restart is required (voltage reset) |
| xCommOK | BOOL | True: communication status of the SBT island is OK<br>False: loss of communication for one or more modules |
| wCommStatus | WORD | Communication status of the individual modules; each bit represents the status of a module (example: bit 1 corresponds to module 1) |
| xDevError | BOOL | Indicates an error in one or more SBT modules |
| xAcknowledgeRequested | BOOL | Acknowledgment by the user is required |
| xResetRequest | BOOL | Reset request signal from one or more safety function blocks is “True” in the logic module |
| xSafetyDemand | BOOL | Safety demand signal from one or more safety function blocks is “True” in the logic module |
| xSFB_Error | BOOL | Indicates that the error signal is set in a safety function block |
| dwApplicationDiag | DWORD | Freely configurable feedback signals from the LPSDO module to the standard controller |

| Name | Type | Description |
|---|---|---|
| arrSBT_DiagCode | SBT_V3_ARR_W_0_16 | Array of SBT module diagnostic codes (device-specific; see module user documentation)<br>Or value of the DIP switches if no diagnostic code is present and the modules are not parameterized<br>The diagnostic codes can be read using the “SBT_V3_DiagCode_V3_00” function block, see C 9 on page 64 |
| arrFeedBackData | Array [0...16] of WORD | Feedback data of all outputs of the LPSDO module and the satellites; the index represents the device number; index 0 stands for the LPSDO module |
| arrInData | Array [0...16] of WORD | Array of inputs from all input modules (satellites) of the SBT island; the index represents the device number; index 0 stands for the LPSDO module |

## C 1.3 I/O parameters

Table C-6 I/O parameters for the “SBT_V3_Operate_V3_00” function block

| Name | Type | Description |
|---|---|---|
| udtSBT | SBT_V3_UDT_SBT_V2 | Data structure for data exchange between function blocks |
| arrSBTOnlCntrlBuf | SBT_V3_ARR_DW_0_175 | Array for online mode in SAFECONF<br>**Important:** the parameter must have the same variable name (arrSBTOnlCntrlBuf) and be defined as a global variable. The PDD check box must be activated.<br>See Figure C-2. |
| arrSBTOnlValBuf | SBT_V3_ARR_W_0_175 | Array for online mode in SAFECONF<br>**Important:** the parameter must have the same variable name (arrSBTOnlValBuf) and be defined as a global variable. The PDD check box must be activated.<br>See Figure C-2. |

**Settings for “arrSBTOnlCntrlBuf” and “arrSBTOnlValBuf”:**


Figure C-2 Settings for “arrSBTOnlCntrlBuf” and “arrSBTOnlValBuf”

## C 1.4 Function block diagnostics

**Diagnostics**

Table C-7 Diagnostic codes

| DiagCode | Meaning | |
|---|---|---|
| $0000_{hex}$ | Function block is not active | |
| $8000_{hex}$ | Function block is active and operating without errors | |
| $8001_{hex}$ | Initialization | |
| $8100_{hex}$ | SAFECONF project and BIN file/POU are being read | |
| | **AddDiagCode** | **Meaning** |
| | $0000_{hex}$ | Read from source |
| | $0001_{hex}$ | Copy in function block |
| $8200_{hex}$ | Reading “ProjectHeader” of LPSDO module | |
| | **AddDiagCode** | **Meaning** |
| | $0000_{hex}$ | Initializing reading of “ProjectHeader” |
| | $0001_{hex}$ | Reading of “ProjectHeader” is complete |
| $8300_{hex}$ | Comparing “Header” of LPSDO module and the loaded program | |
| $8400_{hex}$ | Downloading SAFECONF program to the LPSDO module | |
| | **AddDiagCode** | **Meaning** |
| | $0000_{hex}$ | Removing old “Header” and writing a new one |
| | $0001_{hex}$ | Downloading logic block |
| | $0002_{hex}$ | Downloading address block |
| | $0003_{hex}$ | Downloading new “ProjectHeader” |
| | $0004_{hex}$ | Download successful; device restarting |
| $8500_{hex}$ | Download completed successfully | |
| $C001_{hex}$ | Initialization error | |
| | **AddDiagCode** | **Meaning** |
| | $0004_{hex}$ | Wrong or invalid island number |
| $C002_{hex}$ | LPSDO module unavailable<br>• Check communication and restart block | |
| $C100_{hex}$ | Error when reading/copying the SAFECONF project and BIN file/POU | |
| | **AddDiagCode** | **Meaning** |
| | $0001_{hex}$ | Error when opening the data |
| | $0002_{hex}$ | Error when reading the data (e.g., BIN file not found) |
| | $0003_{hex}$ | Error when closing the data |
| $C850_{hex}$ | Error during download | |
| | **AddDiagCode** | **Meaning** |
| | $0001_{hex}$ | Download error |

# C 2 SBT_V3_InBuff_V3_00 function block

The function block reads the IN process data of the SBT modules.

Use the function block once per island.


Figure C-3 “SBT_V3_InBuff_V3_00” function block

## C 2.1 Input parameters

**Input parameters**

Table C-8 Input parameters for the “SBT_V3_InBuff_V3_00” function block

| Name | Type | Description |
|---|---|---|
| arr_wInBuffLPSDO | SBT_V3_ARR_W_0_23 | Output buffer of the LPSDO module |
| arr_wInBuffDev1 | SBT_V3_ARR_W_0_3 | Output buffer of module 1 |
| arr_wInBuffDev2 | SBT_V3_ARR_W_0_3 | Output buffer of module 2 |
| ⋮ | ⋮ | ⋮ |
| arr_wInBuffDev16 | SBT_V3_ARR_W_0_3 | Output buffer of module 16 |

## C 2.2 I/O parameters

**I/O parameters**

Table C-9 I/O parameters for the “SBT_V3_InBuff_V3_00” function block

| Name | Type | Description |
|---|---|---|
| udtSBT | SBT_V3_UDT_SBT_V2 | Data exchange structure<br>The following applies when inside an island:<br>Same udtSBT parameter as the “SBT_V3_Operate_V3_00” function block |

# C 3 SBT_V3_OutBuff_V3_00 function block

The function block writes the OUT process data to the SBT modules.

Use the function block once per island.

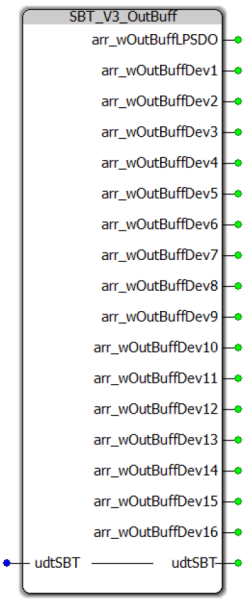



Figure C-4 “SBT_V3_OutBuff_V3_00” function block

## C 3.1 Output parameters

**Output parameters**

Table C-10 Output parameters for the “SBT_V3_OutBuff_V3_00” function block

| Name | Type | Description |
|---|---|---|
| arr_wOutBuffLPSDO | SBT_V3_ARR_W_0_23 | Input buffer of the LPSDO module |
| arr_wOutBuffDev1 | SBT_V3_ARR_W_0_3 | Input buffer of module 1 |
| arr_wOutBuffDev2 | SBT_V3_ARR_W_0_3 | Input buffer of module 2 |
| ⋮ | ⋮ | ⋮ |
| arr_wOutBuffDev16 | SBT_V3_ARR_W_0_3 | Input buffer of module 16 |

## C 3.2 I/O parameters

**I/O parameters**

Table C-11 I/O parameters for the “SBT_V3_OutBuff_V3_00” function block

| Name | Type | Description |
|---|---|---|
| udtSBT | SBT_V3_UDT_SBT_V2 | Data exchange structure<br>The following applies when inside an island:<br>Same udtSBT parameter as the “SBT_V3_Operate_V3_00” function block |

# C 4 SBT_V3_BinFile_V3_00 function block

This function block is only required if you selected **"Binary file"** as the output format for the configuration and parameter data record from SAFECONF (see Figure 3-5 on page 15).

If you selected **"PC Worx"** as the output format, this function block is not required. For additional information on this process, please refer to Appendix B on page 44.

The function block reads the project data from the BIN file and makes the data available to the "SBT_V3_Operate_V3_00" function block.

Use the function block once per island.

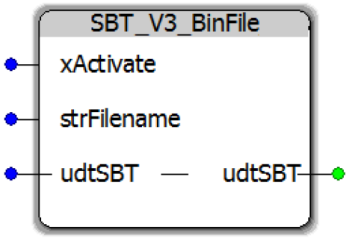



Figure C-5 "SBT_V3_BinFile_V3_00" function block

## C 4.1 Input parameters

**Input parameters**

Table C-12 Input parameters for the "SBT_V3_BinFile_V3_00" function block

| Name | Type | Description |
|---|---|---|
| xActivate | BOOL | Activation/deactivation of the function block |
| strFileName | STRING | BIN file from SAFECONF, which is stored on the FTP server of the controller, e.g., "BinFile.BIN"<br>**Important:** the name must not exceed 24 characters. |

## C 4.2 I/O parameters

**I/O parameters**

Table C-13 I/O parameters for the "SBT_V3_BinFile_V3_00" function block

| Name | Type | Description |
|---|---|---|
| udtSBT | SBT_V3_UDT_SBT_V2 | Data exchange structure<br>The following applies when inside an island:<br>Same udtSBT parameter as the "SBT_V3_Operate_V3_00" function block |

# C 5 SBT_V3_CrossComm_V3_00 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_V3_00” function block combines the data structures of the individual islands in an array, thereby enabling cross communication between the islands.

The “SBT_V3_CrossComm_V3_00” function block may only be called once per standard controller.

If another island is in another standard controller, use an additional “SBT_V3_DataExch_V3_00” function block. See Appendix C 6 on page 58.

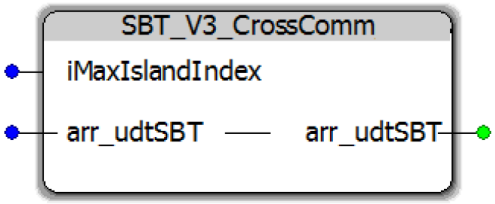



Figure C-6 “SBT_V3_CrossComm_V3_00” function block

## C 5.1 Input parameters

**Input parameters**

Table C-14 Input parameters for the “SBT_V3_CrossComm_V3_00” function block

| Name | Type | Description |
|---|---|---|
| iMaxIslandIndex | INT | Maximum index that is used in the “arr_udtSBT” array |

To increase performance, the “iMaxIslandIndex” parameter limits the cycle for cross communication. If the value is 0, all 31 possible islands are checked by the function block.

## C 5.2 I/O parameters

**I/O parameters**

Table C-15 I/O parameters for the “SBT_V3_CrossComm_V3_00” function block

| Name | Type | Description |
|---|---|---|
| arr_udtSBT | SBT_V3_ARR_UDT_SBT_V2 | Array of data structures for data exchange between function blocks |

The “arr_udtSBT” array should contain all “udtSBT” structures of the islands.
The index of the “arr_udtSBT” array is independent of the index of the island.

# C 6 SBT_V3_DataExch_V3_00 function block

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

If two SBT islands are connected to different controllers but cross communication is required between the two islands, the “SBT_V3_DataExch_V3_00” function block supports data exchange between the master and slave island. The function block uses input and output buffers, which buffer the data that is received from one controller and sent to the other controller. The data transmission method between the two controllers is not specified and can be any of various options, for example Modbus/TCP. However, the data must be transmitted consistently.

If the master island has more than one slave island, call this function block for each slave island.

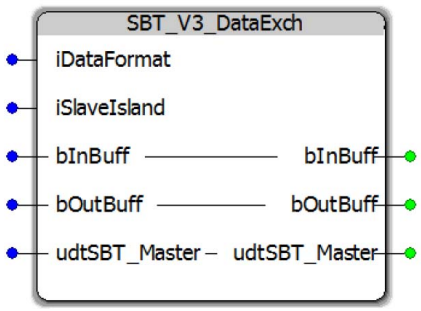



Figure C-7 “SBT_V3_DataExch_V3_00” function block

## C 6.1 Input parameters

**Input parameters** Table C-16 Input parameters for the “SBT_V3_DataExch_V3_00” function block

| Name | Type | Description |
|---|---|---|
| iDataFormat | INT | Determines the format for data exchange with the other controller<br>0: exchange data is not swapped<br>1: exchange data is swapped |
| iSlaveIsland | INT | In the **master** program: number of the slave island with which the master island is exchanging data<br>In the **slave** program: 0 because the island itself is a slave |

## C 6.2 I/O parameters

**I/O parameters** Table C-17 I/O parameters for the “SBT_V3_DataExch_V3_00” function block

| Name | Type | Description |
|---|---|---|
| arr_InBuff | BYTE | First byte of an array containing exchange data (input) from another island<br>**Important:** the array must be at least 16 bytes long.<br>Example:<br>If the input buffer from byte 20 to byte 35 is in the PROFINET exchange buffer, connect the PND_INPUTS[20] field. |
| arr_OutBuff | BYTE | First byte of an array containing exchange data (output) from another island<br>**Important:** the array must be at least 16 bytes long. |
| udtSBT_Master | SBT_V3_Data_2 | Data structure of the master island<br>In the **master** program: the same data structure of the “SBT_V3_Operate_V3_00” function block is connected<br>In the **slave** program: unassigned structure is connected; **not** the same data structure of the “SBT_V3_Operate_V3_00” function block<br>Example:<br>If slave data structure = udtSBT[3], then<br>master data structure = udtSBT[4] |

# C 7 SBT_V3_TransTime_V3_00 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.

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 output 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.

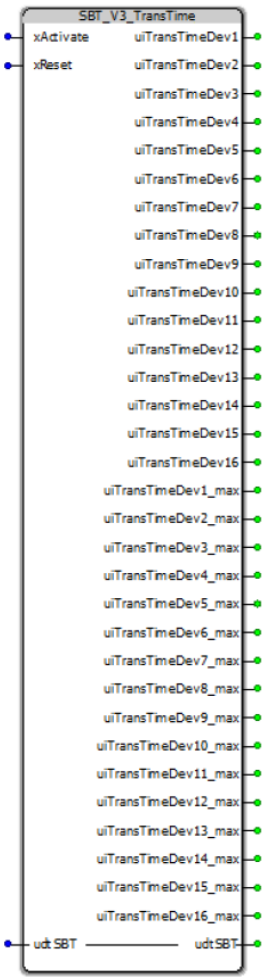



Figure C-8 “SBT_V3_TransTime_V3_00” function block

## C 7.1 Input parameters

**Input parameters** Table C-18 Input parameters for the “SBT_V3_TransTime_V3_00” function block

| Name | Type | Description |
|---|---|---|
| xActivate | BOOL | Activation/deactivation of the function block |
| xReset | BOOL | On a rising edge, the output data transmission times are reset.<br>**Important:** do **not** link the parameter to the same variable as the xReset parameter of the “SBT_V3_Operate_V3_00” function block. |

## C 7.2 Output parameters

**Output parameters** Table C-19 Output parameters for the “SBT_V3_TransTime_V3_00” function block

| Name | Type | Description |
|---|---|---|
| uiTransTimeDev[1...16] | UINT | Data transmission time between LPSDO and satellite 1 to 16. |
| uiTransTimeDev[1...16] _max | UINT | Maximum data transmission time between LPSDO and satellite 1 to 16. |

## C 7.3 I/O parameters

**I/O parameters** Table C-20 I/O parameters for the “SBT_V3_TransTime_V3_00” function block

| Name | Type | Description |
|---|---|---|
| udtSBT | SBT_V3_UDT_SBT_V2 | Data exchange structure<br>The following applies when inside an island:<br>Same udtSBT parameter as the “SBT_V3_Operate_V3_00” function block |

# C 8 SBT_V3_ProjHeader_V3_00 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.

Use the function block once per island.

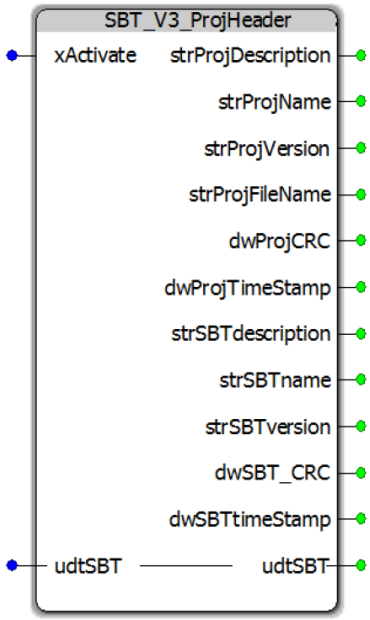



Figure C-9 “SBT_V3_ProjHeader_V3_00” function block

## C 8.1 Input parameters

**Input parameters**

Table C-21 Input parameters for the “SBT_V3_ProjHeader_V3_00” function block

| Name | Type | Description |
|---|---|---|
| xActivate | BOOL | Activation/deactivation of the function block |

## C 8.2 Output parameters

**Output parameters**

Table C-22 Output parameters for the “SBT_V3_ProjHeader_V3_00” function block

| Name | Type | Description |
|---|---|---|
| strProjDescription | STRING | Project description of the SAFECONF project (from BIN file/POU) |
| strProjName | STRING | Project name of the SAFECONF project (from BIN file/POU) |
| strProjVersion | STRING | Project version of the SAFECONF project (from BIN file/POU) |
| strProjFileName | STRING | File name of the SAFECONF project (name of BIN file/POU) |
| dwProjCRC | DWORD | Checksum of the SAFECONF project |
| dwProjTimeStamp | DWORD | Time stamp of the SAFECONF project (seconds from 1970-01-01) |
| strSBTdescription | STRING | Project description of the project stored on the LPSDO module |
| strSBTname | STRING | Name of the project stored on the LPSDO module |
| strSBTversion | STRING | Version of the project stored on the LPSDO module |
| dwSBT_CRC | DWORD | Checksum of the project stored on the LPSDO module |
| dwSBTtimeStamp | DWORD | Time stamp of the project stored on the LPSDO module (seconds from 1970-01-01) |

## C 8.3 I/O parameters

**I/O parameters**

Table C-23 I/O parameters for the “SBT_V3_ProjHeader_V3_00” function block

| Name | Type | Description |
|---|---|---|
| udtSBT | SBT_V3_UDT_SBT_V2 | Data exchange structure<br>The following applies when inside an island:<br>Same udtSBT parameter as the “SBT_V3_Operate_V3_00” function block |

# C 9 SBT_V3_DiagCode_V3_00 function block

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

The function block makes available the diagnostic data of all the SBT modules.

The diagnostic codes of the individual modules are combined in an array of the “SBT_V3_Operate_V3_00” function block and transferred to the “SBT_V3_DiagCode_V3_00” function block. This function block displays the contents of the array.

If no diagnostic code is present or the modules are not parameterized, the function block indicates the value of the DIP switches.

Use the function block once per island.


Figure C-10 “SBT_V3_DiagCode_V3_00” function block

## C 9.1 Input parameters

**Input parameters**

Table C-24 Input parameters for the “SBT_V3_DiagCode_V3_00” function block

| Name | Type | Description |
|---|---|---|
| arrSBT_DiagCode | SBT_V3_ARR_W_0_16 | Array of diagnostic codes from the LPSDO module and satellites 1 to 16.<br>**Important:** link this input to the arrSBT_DiagCode output of the “SBT_V3_Operate_V3_00” function block. |

## C 9.2 Output parameters

**Output parameters**

Table C-25 Output parameters for the “SBT_V3_DiagCode_V3_00” function block

| Name | Type | Description |
|---|---|---|
| wDiagCodeLPSDO | WORD | Diagnostic code of the LPSDO module |
| wDiagCodeDev[1...16] | WORD | Diagnostic code of satellites 1 to 16. |

## C 9.3 Diagnostics examples for the function block

**Diagnostics examples**

Table C-26 Examples for diagnostic codes

| Name | Code | Description |
|---|---|---|
| wDiagCodeLPSDO | $0500_{hex}$ | Communication with one or more modules interrupted |
| | $0D00_{hex}$ | Acknowledgment of a communication error requested |
| | $8000_{hex}$ | SBT island is operating properly |
| | $4xxx_{hex}$ | Configuration of the DIP switches is displayed |
| | Other | See diagnostics description in the user manual |
| wDiagCode[1...16] | $8000_{hex}$ | Module is operating properly |
| | $4xxx_{hex}$ | Configuration of the DIP switches is displayed |
| | Other | See diagnostics description in the user manual |

# C 10 SBT_V3_Tracer_V1_00 function block

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

If a communication error has occurred, the LPSDO module saves the last four telegrams between the module and other devices. The “SBT_V3_Tracer_V1_00” function block reads the telegrams and writes them to the “udtTracer” structure.

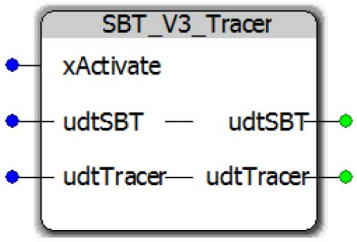



Figure C-11 “SBT_V3_Tracer_V1_00” function block

## C 10.1 Input parameters

**Input parameters**

Table C-27 Input parameters for the “SBT_V3_Tracer_V1_00” function block

| Name | Type | Description |
|---|---|---|
| xActivate | BOOL | Activation/deactivation of the function block |

## C 10.2 I/O parameters

**I/O parameters**

Table C-28 I/O parameters for the “SBT_V3_Tracer_V1_00” function block

| Name | Type | Description |
|---|---|---|
| udtSBT | SBT_V3_UDT_SBT_V2 | Data exchange structure<br>The following applies when inside an island:<br>Same udtSBT parameter as the “SBT_V3_Operate_V3_00” function block |
| udtTracer | SBT_V3_Tracer | Structure contains the last four telegrams of each module; it is filled when the LPSDO module sets the internal “Trace Trigger” signal to “True” |

# D Revision history

| Revision | Date | Contents |
|---|---|---|
| 00 | 2016-05-18 | First publication |

SCATTERGOOD & JOHNSON LTD
ELECTRICAL ENGINEERING & FLUID CONTROL DISTRIBUTORS
Est.1899

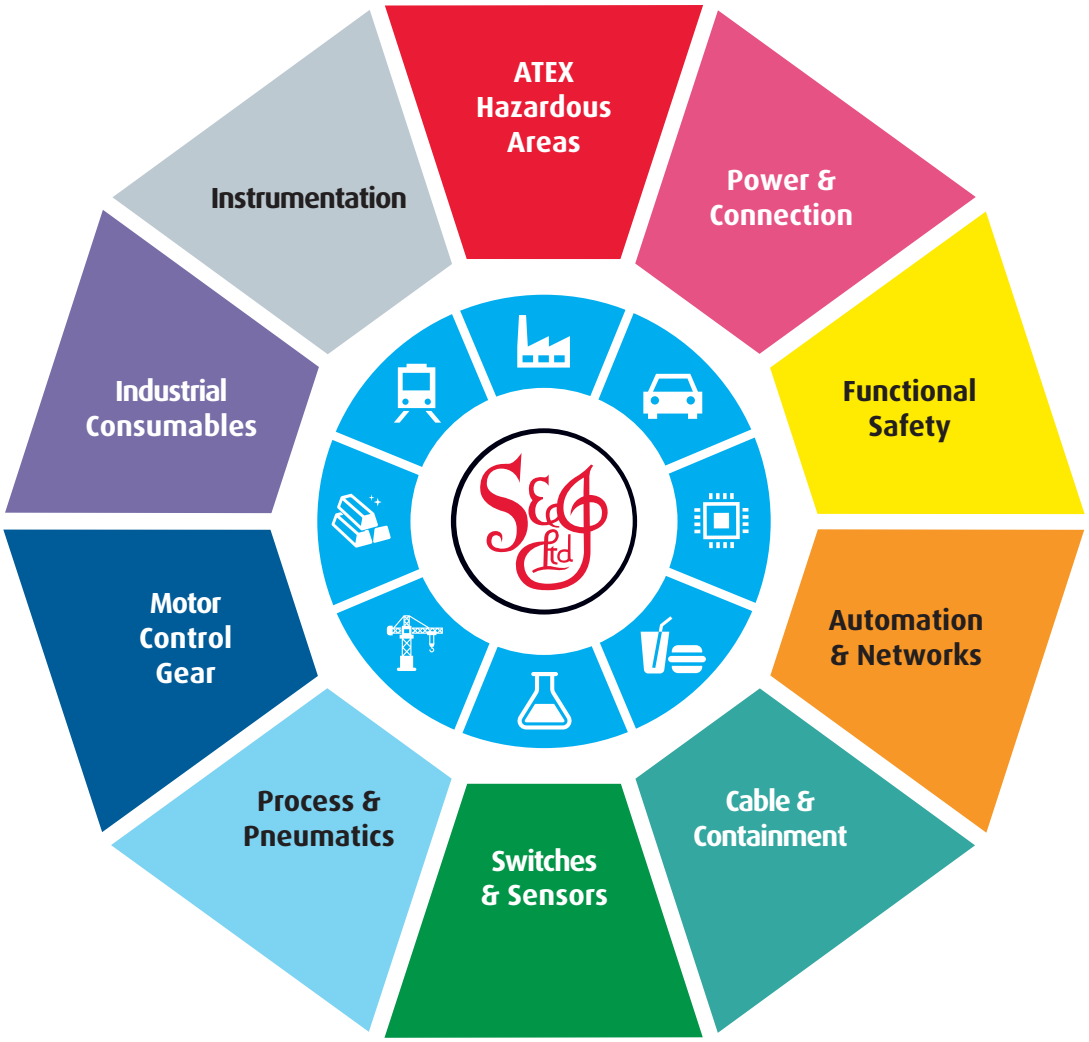
At Scattergood & Johnson Ltd, we pride ourselves on being a technical distributor to specialist industries.

Working with a range of quality product suppliers across a number of specialist markets, we are not your average 'box shifter' - we are your technical and supply chain partner.

We fully support every product we sell - for free! Our internal team and external sales engineers can answer any product or application question, no matter the complexity.

Backing up this technical ability is a range of 50,000+ products available from stock for nationwide next day delivery (same day if required!), or you can collect what you need from any of our trade counters around the UK.

Select your specialist interest below to learn more about how we can help.




Online, In Branch and On the Road - Scattergood & Johnson Ltd, there when you need us.

**www.scatts.co.uk**