rexroth
A Bosch Company

IndraWorks SafeLogic 15VRS

First Steps

Commissioning Manual
R911400170

Edition 02

Title IndraWorks
SafeLogic 15VRS
First Steps

Type of Documentation Commissioning Manual

Document Typecode DOK-IWORKS-SL*STEP*V15-CO02-EN-P

Internal File Reference RS-f0428d34854d08040a347e863c306e8e-2-en-US-4

Change Record Edition 02, 2021-04
Refer to tab. 1-1 "Change Record" on page 1

Editorial Department Engineering Automation Systems Solution Integration PLC, Safety controls, AnSt (MiNi/PiaSt)

Table of Contents

Page

1 About this documentation ... 1
1.1 Validity of the documentation ... 1
1.2 Documentation structure ... 2
1.3 Required and supplementing documentation ... 2
1.3.1 SafeLogic/Safe Motion ... 2
1.3.2 SafeLogic hardware ... 3
1.3.3 IndraWorks/WebAssistant ... 5
1.3.4 SafeLogic compact ... 6
1.4 Using safety instructions ... 8
1.4.1 Structure of the safety instructions ... 8
1.4.2 Explaining signal words and safety alert symbol ... 8
1.4.3 Symbols used ... 9
1.4.4 Explaining the signal alert symbol on the device ... 9
1.5 Names and abbreviations ... 9
1.6 Customer feedback ... 9

2 Project goal "First Steps - Safety" ... 11

3 Installation ... 13

4 Configuring hardware I/Os ... 15
4.1 PROFIsafe ... 15
4.1.1 Inline Profibus coupler R-IL PB BK DI8 DO4/CN ... 15
4.1.2 Inline Profibus coupler R-IL PN BK DI8 DO4 ... 17
4.1.3 Inline PSDI 8 and PSDI 16 (PROFIsafe data input) ... 19
4.1.4 Inline PSDO 8 and PSDOR 4 (PROFIsafe data output) ... 20
4.1.5 S20 Profibus coupler S20-PB-BK ... 21
4.1.6 S20 Profinet coupler S20-PN-BK+ ... 23
4.1.7 S20-PSDI 8/4 (PROFIsafe data input) ... 25
4.1.8 S20-PSDO 8/3 (PROFIsafe data output) ... 26
4.2 CSos ... 26
4.2.1 General information ... 26
4.2.2 CSos S20 I/Os ... 27
4.2.3 S20-SSDI 8/4 (CSos data input) ... 28
4.2.4 S20-SSDO 8/3 (CSos data output) ... 28
4.2.5 IndraDrive ... 29

5 Commissioning ... 31
5.1 Introduction and overview ... 31
5.2 Creating an IndraWorks project ... 31
5.3 PROFIsafe ... 32
5.3.1 Creating and configuring PROFIsafe Inline terminals at the Profibus ... 32
5.3.2 Creating and configuring PROFIsafe Inline terminals at the Profinet ... 37

Page

5.4 CSos ... 43
5.4.1 Creating and configuring CSos S20 terminals ... 43
5.4.2 IndraDrive with CSos ... 46
5.5 Exchanging variables with standard PLC ... 50
5.6 Safety network variables ... 51
5.7 Programming example ... 55
5.8 Loading and starting projects on the control ... 59
5.9 Modifying projects ... 60

6 Troubleshooting ... 61
6.1 SafeLogic interface ... 61
6.2 SafeLogic function module ... 62
6.3 PROFIsafe ... 62
6.4 CIP Safety on Sercos ... 63
6.5 Safety PLCopen function blocks ... 64
6.6 Converting variables (safe ⇔ not safe) ... 64

7 Diagnostics ... 67
7.1 Overview ... 67
7.2 Status of the Safety function module ... 67
7.3 Log of the Safety function module ... 68
7.4 Log of the standard control ... 69
7.5 Device diagnostics of the standard control ... 70
7.6 Diagnostic outputs of the ProfisafeHost and CSosOriginator function blocks ... 71
7.7 Diagnostics on the standard PLC control ... 71

8 Style guide Safety programming ... 75
8.1 Introduction ... 75
8.2 Application structure ... 75
8.3 Variable names ... 75

9 Service and support ... 77

Index ... 79

1 About this documentation

Editions of this documentation

Edition	Release date	Notes
01	2020-11	First edition for version 15VRS
02	2021-04	Revision for version 15V12 • "Prerequisites" on page 13 • chapter 7.1 "Overview" on page 67

Tab. 1-1: Change Record

1.1 Validity of the documentation

Overview on target groups and product phases

In the following illustration, the framed activities, product phases and target groups refer to the present documentation.

Example: In the product phase "Mounting (assembly/installation)", the "mechanic/electrician" can execute the activity "unpack, mount and install" using this documentation.

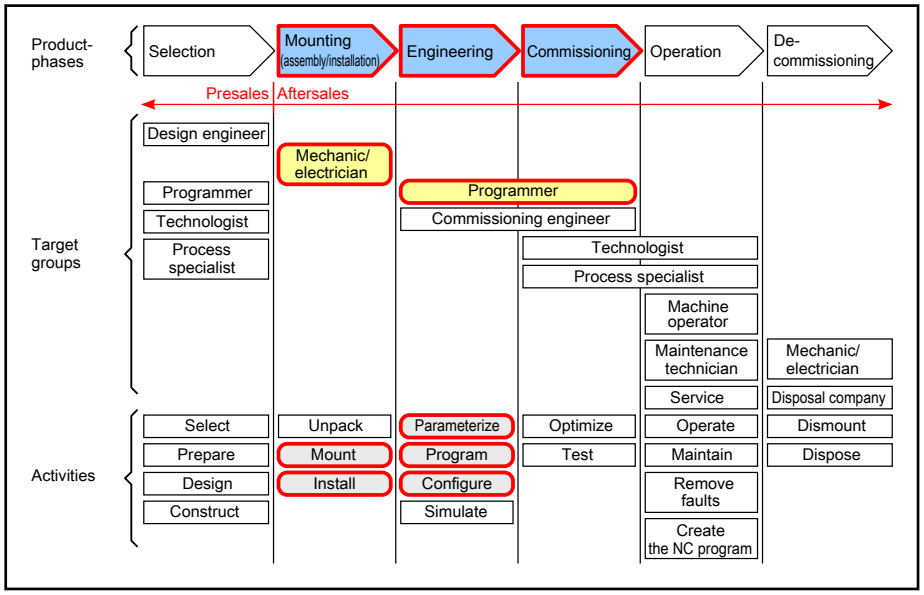

Fig. 1-1: Assigning the present documentation to the target groups, product phases and activities of the target group

Purpose

This documentation contains a description for the creation of a 15VRS control project to commission a SafeLogic control. Furthermore, this documentation includes information on diagnostics and troubleshooting.

In this documentation, the term "control" is used for the following control variants:

- XM21
- XM22
- XM42
- VPx

1.2 Documentation structure

For information on the required hardware and the description of the individual subtasks for the project, refer to chapter 2 "Project goal First Steps - Safety" on page 11.

For notes on hardware and software installation, refer to chapter 3 "Installation" on page 13.

For information on how to configure the hardware I/Os under ProfiSafe, refer to chapter 4 "Configuring hardware I/Os" on page 15 .

For the commissioning of the safety control and the connected periphery, refer to chapter 5 "Commissioning" on page 31.

For information on troubleshooting, refer to chapter 6 "Troubleshooting" on page 61.

For the diagnostic options, refer to chapter 7 "Diagnostics" on page 67.

For programming under SafeLogic, refer to chapter 8 "Style guide Safety programming" on page 75.

For information on the customer service helpdesk, refer to chapter 9 "Service and support" on page 77.

1.3 Required and supplementing documentation

1.3.1 SafeLogic/Safe Motion

Document	Part number, document type	Description
SafeLogic documentation		
IndraWorks SafeLogic 15VRS First Steps	R911400170 Commissioning Manual	This documentation contains a description for the creation of a 15VRS control project to commission a SafeLogic control.
IndraWorks SafeLogic 15VRS Project Configuration	R911398635 Application Description	This documentation describes the creation and the programming of SafeLogic projects in IndraWorks Engineering
ILC MLC 15VRS SafeLogic System Overview	R911400164 Project Planning Manual	This project planning manual describes the SafeLogic compact Safety control and the Safety extension module when using an IoT-compatible PLC system ILC SafeLogic and Motion Control system MLC SafeLogic
MTX 15VRS SafeLogic System Overview	R911398637 Project Planning Manual	This project planning manual describes the SafeLogic compact Safety control and the Safety extension module when using a CNC system MTX
Safe Motion documentations		

Document	Part number, document type	Description
Rexroth IndraDrive Integrated safety technology "Safe Torque Off" (from MPx-16)	R911332634 Application Description	This documentation describes the "Safe Torque Off" safety technology integrated in IndraDrive
IndraDrive Integrated safety technology "Safe Motion" (from MPx-18)	R911338920 Application Description	This documentation describes the "Safe Motion" safety technology integrated in IndraDrive

Tab. 1-2: SafeLogic/Safe Motion documentations

1.3.2 SafeLogic hardware

Document	Part number, document type	Description
Safety extension module		
IndraControl XFE01.1-SY-01 Safety extension module for XM devices	R911376654 Operating Instructions	This operating instructions provides information on the safe mounting and the electric installation of the Safety extension module for XM controls to the technical staff of the machine manufacturer or the machine operator
IndraControl PFC01.1-SY-01 Safety extension module for VPx devices	R911373152 Operating Instructions	This operating instructions provides information on the safe operation of the Safety extension module in the IndraMotion MLC VPx control to the technical staff of the machine manufacturer or the machine operator
Standard controls		
IndraControl XM21/XM22	R911340667 Operating Instructions	This documentation describes the IndraControl XM21/XM22 controls.
IndraControl XM42	R911345566 Operating Instructions	This documentation describes the IndraControl XM42 controls.
IndraControl VPx (based on VPB40.4)	R911383090 Operating Instructions	This documentation describes the MLC IndraControl VPx controls on the basis of the box PC VPB40.4
Bus coupler		
Rexroth Inline Bus Coupler for PROFIBUS-DP R-IL PB BK DI8 DO4/CN-PAC	R911324349 Application Description	This documentation describes the Rexroth Inline bus coupler R-IL PB BK DI8 DO4/CN-PAC
Rexroth Inline Bus Coupler for Profibus DP with Digital Inputs and Outputs R-IL PB BK DI8 DO4/CN-PAC	R911324351 Data Sheet	Contains the technical data of the Rexroth Inline bus coupler R-IL PB BK DI8 DO4/CN-PAC
Rexroth Inline Bus Coupler for PROFINET with digital Inputs and Outputs R-IL PN BK DI8 DO4-PAC	R911328682 Data Sheet	Contains the technical data of the Rexroth Inline bus coupler R-IL PN BK DI8 DO4-PAC

Document	Part number, document type	Description
Rexroth IndraControl S20 Bus Coupler for Profinet	R911342784 Data Sheet	This documentation describes the Rexroth S20 Profinet bus coupler S20-PN-BK+
IndraControl S20 bus coupler for Sercos	R911342782 Data Sheet	This documentation describes the Rexroth S20 Sercos bus coupler S20-S3-BK+
IndraControl S20 bus coupler for PROFIBUS-DP	R911343914 Application Description	This documentation describes den Rexroth S20 PROFIBUS-DP bus coupler S20-PB-BK
I/O modules with safe inputs/outputs		
Rexroth Inline Module with Safe Digital Inputs R-IB IL 24 PSDI 8-PAC	R911326026 Application Description	This documentation describes the Rexroth Inline module R-IB IL 24 PSDI 8-PAC
Rexroth Inline Module with Safe Digital Outputs R-IB IL 24 PSDO 8-PAC	R911326028 Application Description	This documentation describes the Rexroth Inline module R-IB IL 24 PSDO 8-PAC
Rexroth Inline Module with Safe Digital Outputs R-IB IL 24 PSDO 4/4-PAC	R911336653 Application Description	This documentation describes the Rexroth Inline module R-IB IL 24 PSDO 4/4-PAC
Rexroth Inline Module with Safe Digital Relay Outputs R-IB IL 24 PSDOR 4-PAC	R911336651 Application Description	This documentation describes the Rexroth Inline module R-IB IL 24 PSDOR 4-PAC
IndraControl S20 module with safe digital Outputs S20-PSDO-8/3	R911369164 Application Description	This documentation describes the Rexroth S20 module S20-PSDO-8/3 (PROFIsafe)
IndraControl S20 module with safe digital Inputs S20-PSDI-8/4	R911369168 Application Description	This documentation describes the Rexroth S20 module S20-PSDI 8/4 (PROFIsafe)
IndraControl S20 module with safe digital Outputs S20-SSDO-8/3	R911342482 Application Description	This documentation describes the Rexroth S20 module S20-SSDO-8/3 (CSoS)
IndraControl S20 module with safe digital Inputs S20-SSDI-8/4	R911342480 Application Description	This documentation describes the Rexroth S20 module S20-SSDI 8/4 (CSoS)

Tab. 1-3: SafeLogic hardware documentations

1.3.3 IndraWorks/WebAssistant

Document	Part number, document type	Description
IndraWorks 15VRS Software Installation	R911393450 Commissioning Manual	This documentation describes the IndraWorks installation.
IndraWorks 15VRS Engineering	R911393303 Application Description	This documentation describes the use of IndraWorks in which the Rexroth Engineering tools are integrated. It includes instructions on how to work with IndraWorks and how to operate the oscilloscope function.
IndraWorks 15VRS PLC Programming System IndraLogic 2G	R911396137 Application Description	This documentation describes the PLC programming tool IndraLogic 2G and its use. The documentation includes the basic use, first steps, visualization, menu items and editors.
IndraWorks 15VRS Basic Libraries, IndraLogic 2G	R911398633 Library	This documentation describes the system-comprehensive PLC libraries.
IndraWorks 15VRS Field buses	R911393284 Application Description	This documentation describes the field bus and local periphery connections supported by the MLC and MTX systems. The focus of this documentation lies in the configuration, parameterization, commissioning and diagnostics of different periphery connections. It is the basis for the online help.
IndraWorks 15VRS Field Bus Libraries	R911393275 Library	This documentation describes the field bus libraries for the IndraLogic ILC, IndraMotion MLC and IndraMotion MTX systems
IndraWorks 14VRS HMI	R911343569 Application Description	This documentation describes the functions, configuration and operation of the user interfaces IndraWorks HMI Engineering and IndraWorks HMI Operation.
WebAssistant	R911381469 Application Description	This documentation describes the WebAssistant. The WebAssistant is a web-based diagnostic tool used to access a control system via an Ethernet high-speed connection. The WebAssistant allows OEMs, end users and service engineers to access and to remotely diagnose a system.

Tab. 1-4: IndraWorks/WebAssistant documentations

1.3.4 SafeLogic compact

Document	Part number, document type	Description
IndraControl SafeLogic compact Designer software	R911332749 Operating Instructions	Instructs technical staff of the machine vendor on how to configure the software and on how to operate and diagnose a SafeLogic compact system with the SafeLogic Designer software
IndraControl SafeLogic compact Hardware	R911332746 Operating Instructions	Instructs the technical staff of the machine vendor and the machine operator on safe assembly, electrical installation, commissioning as well as maintenance of the SafeLogic compact Diagnostic Gateway
Rexroth IndraControl SafeLogic compact Diagnostic Gateways	R911332752 Operating Instructions	Describes the SafeLogic compact diagnostic gateways and their functions in detail
IndraControl SafeLogic compact Sercos Gateway	R911338436 Project Planning Manual	Instructs the technical staff of the machine vendor and the machine operator on safe assembly, configuration, electrical installation, commissioning as well as maintenance of the SafeLogic compact Sercos Gateway
Safety controls network solutions SafeLogic compact	R911332754 Safety instructions	These safety instructions provide information to the planner, developer and operator as well as persons installing the protective equipment in a machine/system, and initially commission and operate it

Document	Part number, document type	Description
SLC-3-GS3S00300 SafeLogic compact Sercos Gateway	R911339570 Assembly Instructions	This assembly instruction describes the assembly of the modules of the SafeLogic compact Safety control
SLC-0-GPNT00300 SafeLogic compact Ethernet Gateway Profinet I/O	R911334404 Assembly Instructions	
SLC-0-GPRO00300 SafeLogic compact Profibus Gateway	R911334403 Assembly Instructions	
SLC-3-MOC000300 SafeLogic compact Motion Control Module	R911343758 Assembly Instructions	
SLC-3-CPU000300/ SLC-3-CPU130302/ SLC-3-CPU320302 SafeLogic compact Main Modules	R911334402 Assembly Instructions	
SLC-3-XTDI80302/SLC-3-XTIO84302/ SLC-3-XTDS84302/SLC-0-STIO68302 SafeLogic compact Extension modules	R911334401 Assembly Instructions	
SLC-A-UE410-2RO4/ SLC-A-UE410-4RO4 SafeLogic compact Digital Output Modules	R911334400 Assembly Instructions	
SLC-A-MOC-MFSB-RX/ SLC-A-MOC-DECB-RX SafeLogic compact Encoder Junction Boxes	R911343761 Assembly Instructions	

Tab. 1-5: SafeLogic compact documentation

1.4 Using safety instructions

1.4.1 Structure of the safety instructions

The safety instructions are structured as follows:

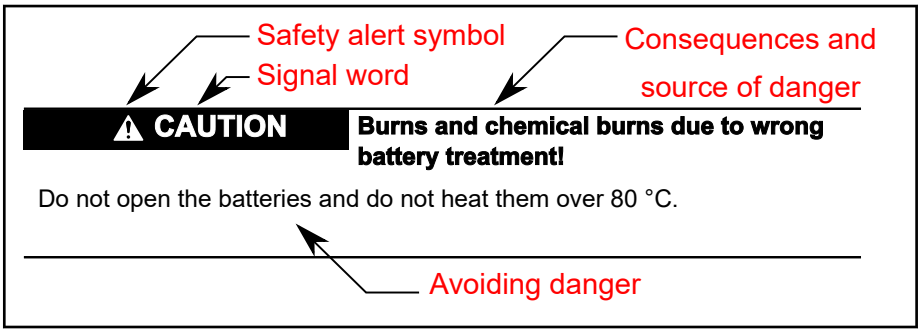

Fig. 1-2: Structure of the safety instructions

1.4.2 Explaining signal words and safety alert symbol

The safety instructions in this documentation contain specific signal words (danger, warning, caution, notice) and, if necessary, a safety alert symbol (according to ANSI Z535.6-2006).

The signal word draws attention to the safety instruction and indicates the risk potential.

The safety alert symbol (triangular safety reflector with exclamation marks), preceding the signal words Danger, Warning, Caution indicates hazards for persons.

⚠ DANGER

In case of non-compliance with this safety instruction, death or serious injury **will** occur.

⚠ WARNING

In case of non-compliance with this safety instruction, death or serious injury **can** occur.

⚠ CAUTION

In case of non-compliance with this safety instruction, minor or moderate injury can occur.

NOTICE

In case of non-compliance with this safety instruction, material damage can occur.

1.4.3 Symbols used

Pointers are displayed as follows:

This is a note.

Tips are displayed as follows:

This is a tip.

1.4.4 Explaining the signal alert symbol on the device

If this symbol is on your device, you have to observe the documentation on the device. The respective documentation informs on the type of hazard as well as the steps required to avoid this hazard.

1.5 Names and abbreviations

Term	Explanation
MLC	Compact Motion Logic systems with Motion, Robot and Logic Control functionalities
MTX	Universal CNC control
IndraWorks Engineering Framework	Project planning and commissioning tool of Bosch Rexroth
IndraDrive	Drive controller
Wizard	Wizard guiding through several dialogs for an ergonomic data input
Function block	Provides application-specific functionalities
IEC	International Electrotechnical Commission
Sercos	Sercos (serial real-time communication system) interface is a world-wide standardized interface for the communication between controls and drives
CSos	Also called CIP Safety on Sercos, is a safe field bus protocol for Sercos devices
PROFIsafe	Safe field bus protocol based on Profibus and Profinet
SDDML	Sercos Device Description Markup Language. Device description language for Sercos III devices. It is an XML description for Sercos III devices and follows the Sercos III device model

Tab. 1-6: Terms and abbreviations

1.6 Customer feedback

Customer requests, comments or suggestions for improvement are of great importance to us. Please email your feedback on the documentations to Feedback.Documentation@boschrexroth.de. Directly insert comments in the electronic PDF document and send the PDF file to Bosch Rexroth.

2 Project goal "First Steps - Safety"

The project goal is to facilitate the start into the safety control engineering of Bosch Rexroth.

One of the following listed control variants (including the Safety function module) is required for practical tasks:

MLC system

- **XM2 control**

  Equipped with the extension module SafeLogic **XFE01.1-SY-01**.
- **XM4 control**

  Equipped with the extension module SafeLogic **XFE01.1-SY-01**
- **VPx control**

  Equipped with the extension module SafeLogic **PFC01.1-SY-01**

MTX system

- **XM4 control**

  Equipped with the extension module SafeLogic **XFE01.1-SY-01**

Safe hardware I/Os based on "PROFIsafe" are connected via the Profibus/ Profinet standard interface of the control.

Subtasks

- Installing and creating a basis for a SafeLogic project
- Configuring hardware I/Os based on "PROFIsafe" and "CSos"
- *Commissioning a safety control and the connected periphery:*
  - Creating an IndraWorks project
  - Creating and configuring PROFIsafe Inline terminals
  - Creating and configuring IndraDrive with CSos
  - Concept of the exchange variables Standard PLC/Safety PLC
  - Programming example
  - Loading and the project to the control and starting it
  - Modifying the project
- Troubleshooting
- Diagnostic support
- Style guide “Safety programming”

3 Installation

Prerequisites

The following minimum system requirements apply to smaller IndraWorks Safety Manager projects with at most 100 blocks, 10 visualizations and 8 field bus devices:

- 2 GB RAM
- 1 GHz Pentium
- 5 GB available hard drive space

The IndraWorks Safety manager programming system is only released for:

- Operating system: Windows 10, 64 bit
- Screen resolution: 96 dpi

Installing the device description files

Drive firmware

To use the safety functions of the drive, the IndraDrive firmware MPx18V08 or higher is required. To use the safe switches of the drive, use the drive firmware 20V08 or higher.

SDDML file

To correctly control the safety-related functions of the IndraDrive, an SDDML file is required.

The SDDML files for Rexroth IndraDrive and CSos S20 I/Os are already installed in IndraWorks.

GSD and GSDML files

The GSD files for Rexroth Inline Profibus I/Os with PROFISafe are already installed in IndraWorks.

The GSDML files for Rexroth Inline Profibus I/Os with S20 Profinet I/Os with PROFIsafe are already installed in IndraWorks.

Installing the SafetyManager license

There are two types of SafetyManager licenses:

- IndraWorks SafetyManager (R911322463)

  This license is required to develop Safety programs.

  It allows the following:

  – Logging into the Safety control including all debugging actions
  – Setting passwords
  – Downloading Safety programs
  – Creating boot projects
  – Retrieving control log
  – Updating firmware

- IndraWorks SafetyManager Lite (R911383338)

  This license is required for the remote support and for the servicing of Safety applications.

  It allows the following:

  – Logging into the control via remote support access
  – Debugging without the option to change values
  – Retrieving control log

1. Open license overview

   For the license overview, go to **Tools** ▸ **Options...**, **General** ▸ **Software licenses**.

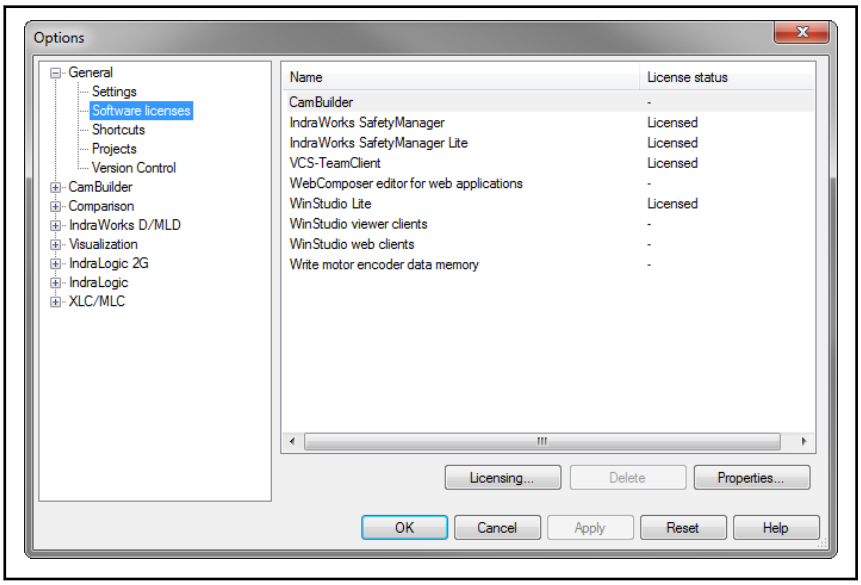

Fig. 3-1: License overview

2. Licensing

   For the license dialog, click on the **Licensing...** button. Enter the activation key.

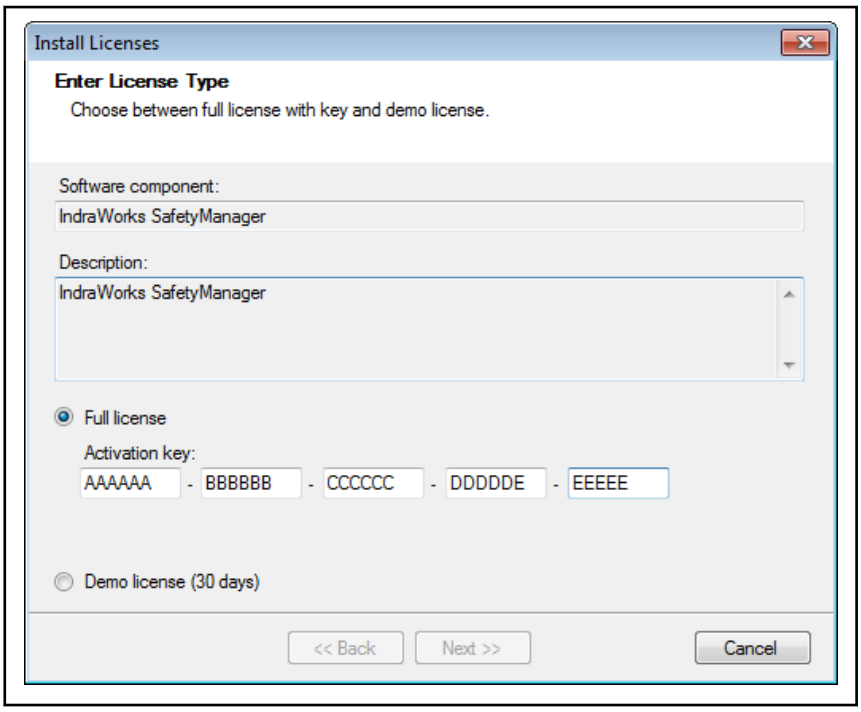

Fig. 3-2: License dialog

To verify the activation key and to apply the license, press **Next >>**.

The IndraWorks SafetyManager license (Lite) is now installed and activated.

4 Configuring hardware I/Os

4.1 PROFIsafe

4.1.1 Inline Profibus coupler R-IL PB BK DI8 DO4/CN

Maximum extension stage

Only a limited number of Safety Inline terminals can be operated at an Inline Profibus coupler of the type R-IL PB BK DI8 DO4/CN. Depending on the design, up to six safe Inline terminals can be connected.

The Inline Profibus coupler provides a Profibus parameter channel for up to 230 bytes to parameterize the Inline terminals.

The Safety Inline terminals are assigned to the following number of bytes in the Profibus parameter channel:

Safety Inline terminal	Process data in bits[1)]	Number of bytes in the Profibus parameter channel
R-IB IL 24 PSDI 8	8	54
R-IB IL 24 PSDI 16	16	36
R-IB IL 24 PSDO 8	8	60
R-IB IL 24 PSDO 4/4	8	48
R-IB IL 24 PSDOR 4	8	48

1) 4 bytes of process data are additionally assigned in the input and output range for each PSDx module (status/control byte, CRC)

Tab. 4-1: Assigning Profibus parameter channel using Safety Inline terminals

When equipping the Inline Profibus coupler, ensure that the maximum number of 230 bytes is not exceeded in the Profibus parameter channel.

Non-safe Inline terminals are also assigned to the Profibus parameter channel.

Use the Inline Builder from 1.3VRS to configure the PROFIsafe Inline terminals.

Example:

Equipping the Profibus coupler with one PSDI8 and two PSDO8 modules

The number of assigned bytes in the Profibus parameter channel is in total:

1 x 54 bytes + 2 x 60 bytes = 174 bytes

56 bytes are thus available.

Setting the DIP switches

The DIP switches have to be set as follows:

- Mode: Mode1
- Baud rate: 500 kBauds if grey or yellow Inline terminals are operated at a coupler; otherwise 2 MBauds
- F-address: Corresponds to the **F_Dest address** in the configuration dialog. PROFIsafe addresses between 1 and 1022 are permitted

⚠ CAUTION **This F-address has to be unique in the complete project**

Refer to the chapter "PROFISafe F-parameters" in the "IndraWorks 15VRS SafeLogic Project Configuration" manual.

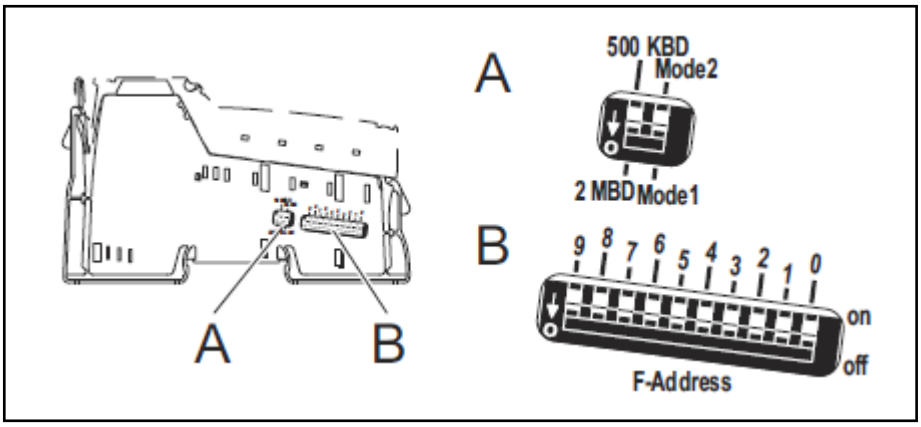

Fig. 4-1: DIP switch of a PROFIsafe Inline terminal

- The imprint on the housing is valid. The digits directly printed on the switch can be ignored
- Upon delivery, all switches for the F_Dest address are "on"

To obtain a uniform and unique numbering, set the F_Dest address as follows:

```
F_Dest = 10*ProfiBusAddr + SlotNo
```

ProfiBusAddr: Profibus slave address of the bus coupler

SlotNo: Slot of the Inline terminal at the bus coupler

The DIP switches of the Inline terminal have to be set to the Inline coupler before mounting them.

Note the set F_Dest addresses.

The DIP switches cannot be accessed anymore after integrating the Inline terminal.

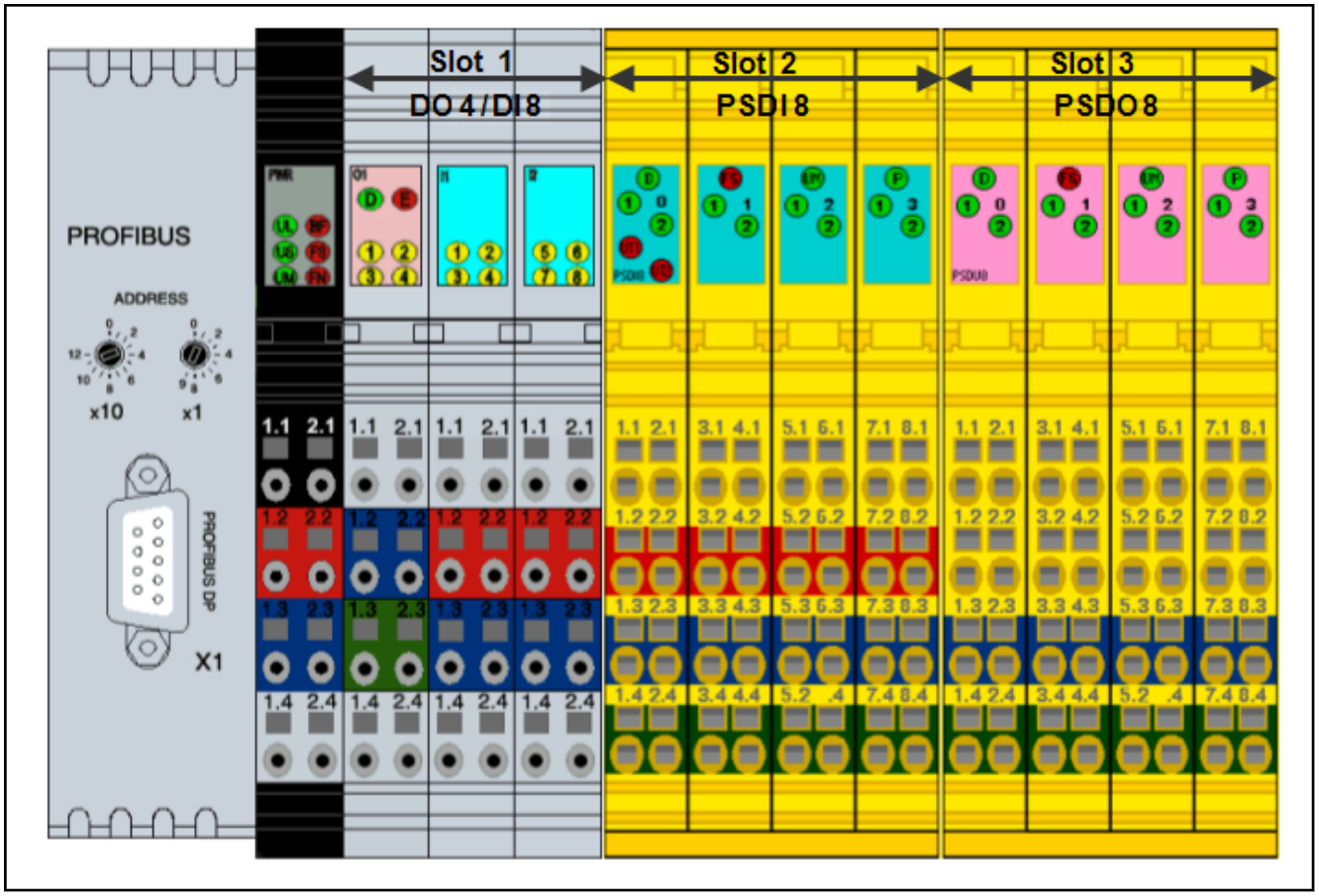

Fig. 4-2: Numbering the slots at the Profibus Inline coupler R-IL PB BK DI8 DO4/CN

4.1.2 Inline Profibus coupler R-IL PN BK DI8 DO4

Maximum extension stage

Up to 16 parameterizable Inline terminals (PCP devices) can be connected to one Inline Profinet coupler of the type R-IL PN BK DI8 DO 4. PROFIsafe Inline terminals can be parameterized. To ensure the voltage supply of the PROFIsafe Inline terminals, always provide a power feed of logic supply voltage after every five modules.

Use the Inline Builder from 1.3VRS to configure the PROFIsafe Inline terminals.

Setting the DIP switches

The DIP switches have to be set as follows:

- Mode: Mode1
- Baud rate: 500 kBauds if grey or yellow Inline terminals are operated at a coupler; otherwise 2 MBauds
- F-address: Corresponds to the **F_Dest address** in the configuration dialog. PROFIsafe addresses between 1 and 1022 are permitted

⚠ CAUTION **This F-address has to be unique in the complete project**

Refer to the chapter "PROFISafe F-parameters" in the "IndraWorks 15VRS SafeLogic Project Configuration" manual.

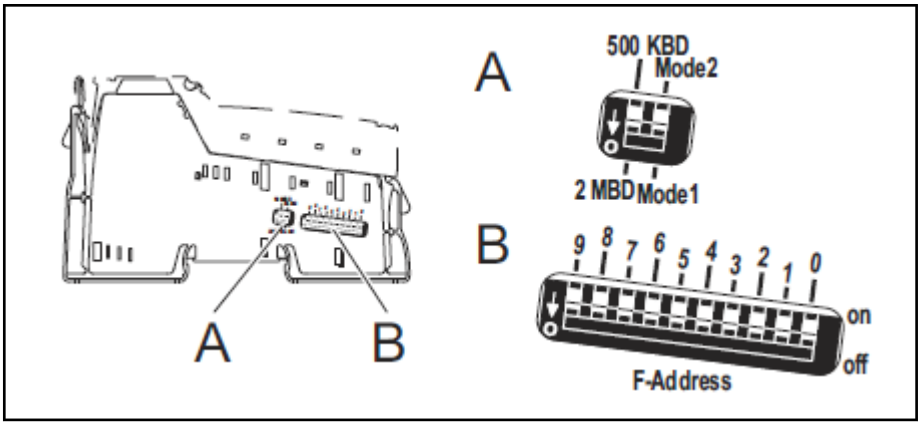

Fig. 4-3: DIP switch of a PROFIsafe Inline terminal

- The imprint on the housing is valid. The digits directly printed on the switch can be ignored
- Upon delivery, all switches for the F_Dest address are "on"

To obtain a uniform numbering, set the F_Dest address as follows:

```
F_Dest = 10*IPAddr + SlotNo
```

IPAddr: Last part of the 4-digit IP address separated by points (example: 192.168.2.4 ⇒ IPAddr = 4)

SlotNo: Slot of the Inline terminal at the bus coupler

The DIP switches of the Inline terminal have to be set to the Inline coupler **before mounting** them.

Note the set F_Dest addresses.

The DIP switches cannot be accessed anymore after integrating the Inline terminal.

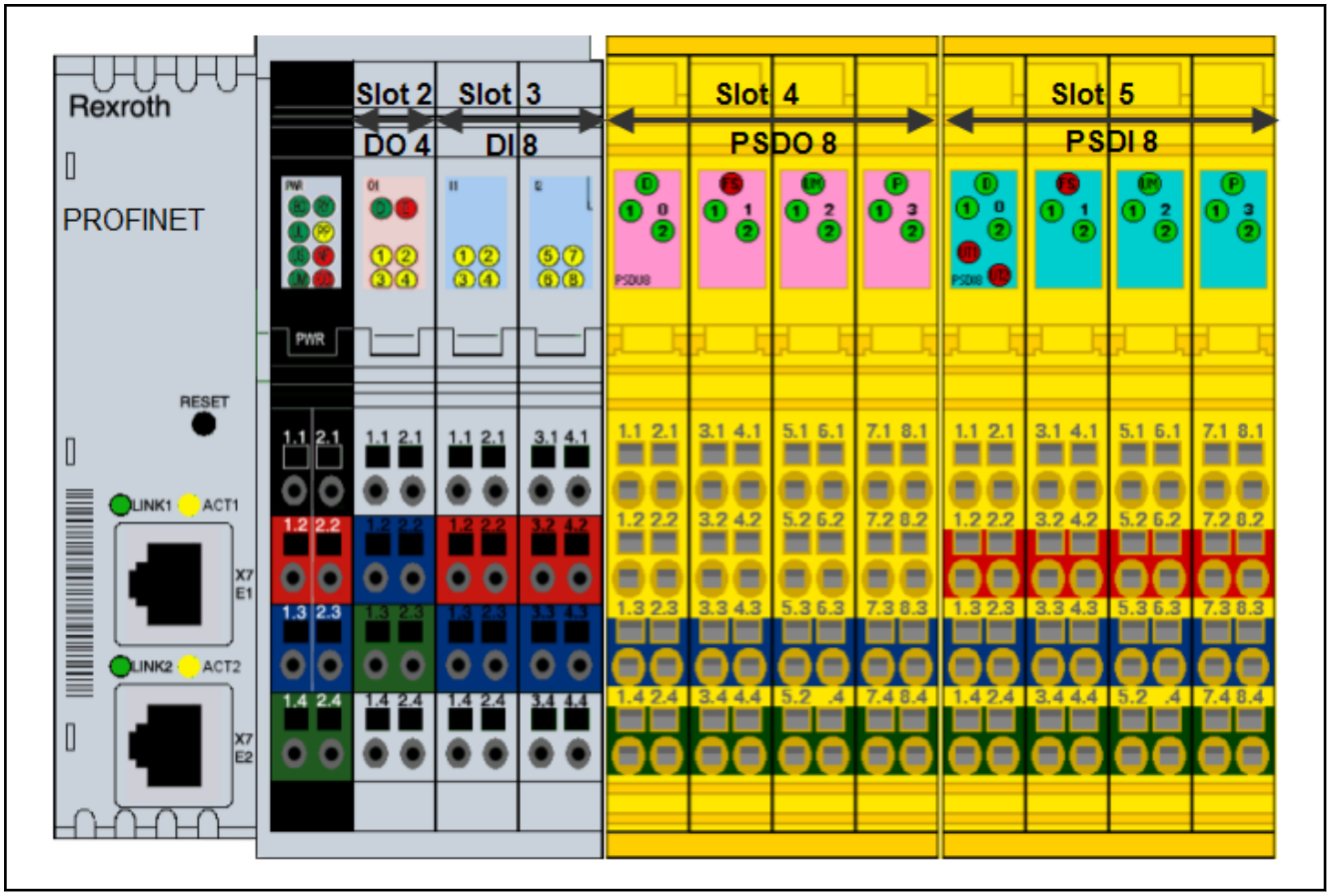

Fig. 4-4: Numbering the slots at the Profinet Inline coupler R-IL PN BK DI8 DO4

4.1.3 Inline PSDI 8 and PSDI 16 (PROFIsafe data input)

Parameterization

Single-channel

All channels can be operated individually. For the PSDI 8, both channels of one input have to be set to "single channel" and "used"/"not used".

Dual-channel

Both channels of one input are interconnected. For PSDI 8, both channels of an input (channel pair) have to be parameterized to "double channel" and "used".

	Channel 1			Channel 2		
	Assignment	**Evaluation**	**Symmetry**	**Assignment**	**Evaluation**	**Symmetry**
Single-channel: Both channels enabled	Used	Single channel	Disabled	Used	Single channel	Disabled
Single-channel: Channel 1 active	Used	Single channel	Disabled	Not used	Single channel	Disabled
Single-channel: Channel 2 active	Not used	Single channel	Disabled	Used	Single channel	Disabled
Channel 1 + channel 2 redundant	Used	Double channel	Disabled	Used	Double channel	Disabled
Channel 1 + channel 2 redundant with symmetry monitoring	Used	Double channel	Enabled	Used	Double channel	Enabled

Tab. 4-2: Parameterization for R-IB IL 24 PSDI 8

	SAFEBYTE							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Single-channel	IN3_Ch2	IN3_Ch1	IN2_Ch2	IN2_Ch1	IN1_Ch2	IN1_Ch1	IN0_Ch2	IN0_Ch1
Dual-channel		IN3_Ch1&2		IN2_Ch1&2		IN1_Ch1&2		IN0_Ch1&2
Terminal name	8.1	7.1	6.1	5.1	4.1	3.1	2.1	1.1

Tab. 4-3: I/O mapping of input data with regard to the channel configuration with R-IB IL 24 PSDI 8 as example

	SAFEBYTE							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Single-channel	IN3_Ch2	IN3_Ch1	IN2_Ch2	IN2_Ch1	IN1_Ch2	IN1_Ch1	IN0_Ch2	IN0_Ch1
Dual-channel	0	IN3_Ch1&2	0	IN2_Ch1&2	0	IN1_Ch1&2	0	IN0_Ch1&2
Terminal name	4.4	3.4	4.1	3.1	2.4	1.4	2.1	1.1

Tab. 4-4: I/O mapping of input data with regard to the channel configuration with R-IB IL 24 PSDI 16 (byte 0) as example

	SAFEBYTE							
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Single-channel	IN7_Ch2	IN7_Ch1	IN6_Ch2	IN6_Ch1	IN5_Ch2	IN5_Ch1	IN4_Ch2	IN4_Ch1
Dual-channel	0	IN7_Ch1&2	0	IN6_Ch1&2	0	IN5_Ch1&2	0	IN4_Ch1&2
Terminal name	8.4	7.4	8.1	7.1	6.4	5.4	6.1	5.1

Tab. 4-5: I/O mapping of input data with regard to the channel configuration with R-IB IL 24 PSDI 16 (byte 1) as example

4.1.4 Inline PSDO 8 and PSDOR 4 (PROFIsafe data output)

Parameterization

Single-channel

All channels can be operated individually. Both channels of an output have to be set to "single channel" and "used" / "not used".

Dual-channel

Both channels of one output are interconnected. Both channels of an output (channel pair) have to be parameterized to "double channel" and "used".

	Channel 1		Channel 2	
	Assignment	Output	Assignment	Output
Single-channel: Both channels enabled	Used	Single channel	Used	Single channel
Single-channel: Channel 1 active	Used	Single channel	Not used	Single channel
Single-channel: Channel 2 active	Not used	Single channel	Used	Single channel
Channel 1 + channel 2 redundant	Used	Double channel	Used	Double channel

Tab. 4-6: Parameterization for R-IB IL 24 PSDO 8

	SAFEBYTE							
Bit	7	6	5	4	3	2	1	0
Single-channel	OUT3_Ch2	OUT3_Ch1	OUT2_Ch2	OUT2_Ch1	OUT1_Ch2	OUT1_Ch1	OUT0_Ch2	OUT0_Ch1
Dual-channel		OUT3_Ch1 &2		OUT2_Ch1 &2		OUT1_Ch1 &2		OUT0_Ch1&2
Terminal name	8.1	7.1	6.1	5.1	4.1	3.1	2.1	1.1

Tab. 4-7: I/O mapping of output data with regard to the channel configuration with R-IB IL 24 PSDO 8 as example

	SAFEBYTE							
Bit	7	6	5	4	3	2	1	0
Single-channel					OUT1_Ch2	OUT1_Ch1	OUT0_Ch2	OUT0_Ch1
Dual-channel						OUT1_Ch1 &2		OUT0_Ch1&2
Terminal name					4.2	3.2	2.2	1.2

Tab. 4-8: I/O mapping of output data with regard to the channel configuration with R-IB IL 24 PSDOR 4 as example

Bit 4 to bit 7 may not be used. Otherwise, the module R-IB IL 24 PSDOR 4 goes into the safe state: Error message 0x1F4 "At least one reserved bit was set in the process data image".

4.1.5 S20 Profibus coupler S20-PB-BK

Maximum extension stage

Up to 63 S20 terminals can be connected to an S20 Profibus coupler of the type S20-PB-BK. The S20 Profibus coupler provides a Profibus parameter channel for up to 230 bytes to parameterize the S20 terminals. The limits the number of the PROFIsafe S20 terminals.

The Safety S20 terminals are assigned to the following number of bytes in the Profibus parameter channel:

Safety S20 terminal	Process data in bits[1)]	Number of bytes in the Profibus parameter channel
S20-PSDI 8/4	8	8
S20-PSDO 8/3	8	8

1) 4 bytes of process data are additionally assigned in the input and output range for each PSDx module (status/control byte, CRC)

Tab. 4-9: Assigning Profibus parameter channel using Safety S20 terminals

☞ When equipping the S20 Profibus coupler, ensure that the maximum number of 230 bytes is not exceeded in the Profibus parameter channel.

Non-safe S20 terminals are also assigned to the Profibus parameter channel.

It is recommended not to use more than 16 PROFIsafe S20 terminals per S20 Profibus coupler S20-PB-BK. To ensure the voltage supply of the terminals, always provide a power feed of logic supply voltage for all six PSDx modules.

☞ To operate S20-PSDx modules, the PROFIBUS S20 bus coupler S20-PB-BK (R911173247) from "AB1" is required.

☞ **Electronic damage due to overload**

When projecting an S20-PB-BK station, observe the logic current consumption of each device!

It is specified in each module-specific data sheet. It can differ module-specifically. Thus, the potential number of devices that can be connected depends on the specific setup of the station in question.

If required, use the logic voltage feed-in terminals!

Setting the DIP switches

The DIP switches have to be set as follows:

- F-address: Corresponds to the **F_Dest address** in the configuration dialog. PROFIsafe addresses between 1 and 1022 are permitted

⚠ CAUTION **This F-address has to be unique in the complete project**

Refer to the chapter "PROFISafe F-parameters" in the "IndraWorks 15VRS SafeLogic Project Configuration" manual.

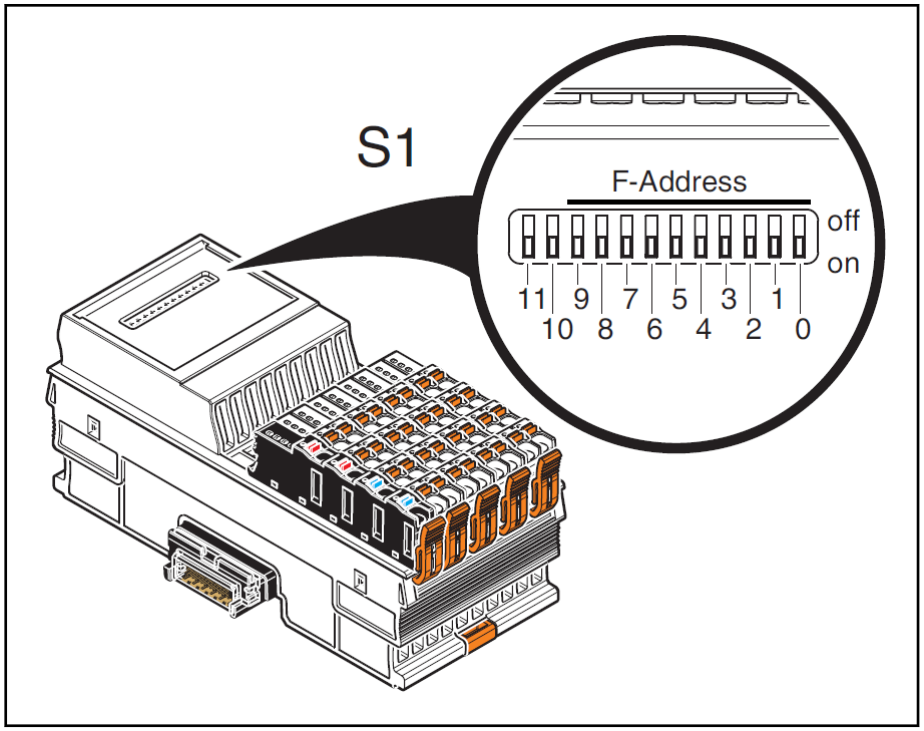

Fig. 4-5: DIP switch of a PROFIsafe S20 terminal

The imprint on the housing is valid. The digits directly printed on the switch can be ignored.

To obtain a uniform and unique numbering, set the **F_Dest address** as follows:

```
F_Dest = 10*ProfiBusAddr + SlotNo
```

ProfiBusAddr: Profibus slave address of the bus coupler

SlotNo: Slot of the S20 terminal at the bus coupler

The set address is only applied at a power-up.

If the address is changed at runtime, the module responds with FailureState.

The positions 10 and 11 of the 12-pin DIP switch are reserved for the operation mode of the module and preset on **off, off** upon delivery. The two positions may not be changed.

4.1.6 S20 Profinet coupler S20-PN-BK+

Maximum extension stage

Up to 63 S20 terminals can be connected to an S20 Profinet coupler of the type S20-PN-BK+. The number of the PROFIsafe S20 terminals is not limited. However, it is recommended not to use more than 16 PROFIsafe S20 terminals per S20 Profinet coupler S20-PN-BK+. To ensure the voltage supply of the terminals, always provide a power feed of logic supply voltage for all six PSDx modules.

Electronic damage due to overload

When projecting an S20-PN-BK+ station, observe the logic current consumption of each device!

It is specified in each module-specific data sheet. It can differ module-specifically. Thus, the potential number of devices that can be connected depends on the specific setup of the station in question.

If required, use the logic voltage feed-in terminals!

Setting the DIP switches

The DIP switches have to be set as follows:

- F-address: Corresponds to the **F_Dest address** in the configuration dialog. PROFIsafe addresses between 1 and 1022 are permitted

⚠ CAUTION **This F-address has to be unique in the complete project**

Refer to the chapter "PROFISafe F-parameters" in the "IndraWorks 15VRS SafeLogic Project Configuration" manual.

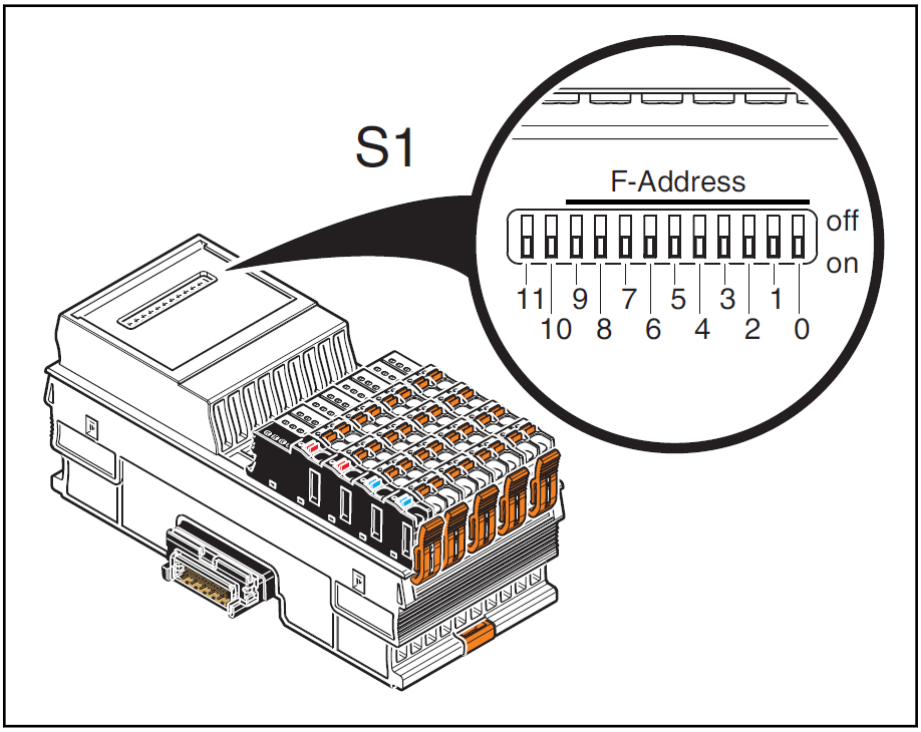

Fig. 4-6: DIP switch of a PROFIsafe S20 terminal

- The imprint on the housing is valid. The digits directly printed on the switch can be ignored.

To obtain a uniform numbering, set the F_Dest address as follows:

```
F_Dest = 10*IPAddr + SlotNo
```

IPAddr: Last part of the 4-digit IP address separated by points (example: 192.168.2.4 ⇒ IPAddr = 4)

SlotNo: Slot of the S20 terminal at the bus coupler

The set address is only applied at a power-up.

If the address is changed at runtime, the module responds with FailureState.

The positions 10 and 11 of the 12-pin DIP switch are reserved for the operation mode of the module and preset on **off, off** upon delivery. The two positions may not be changed.

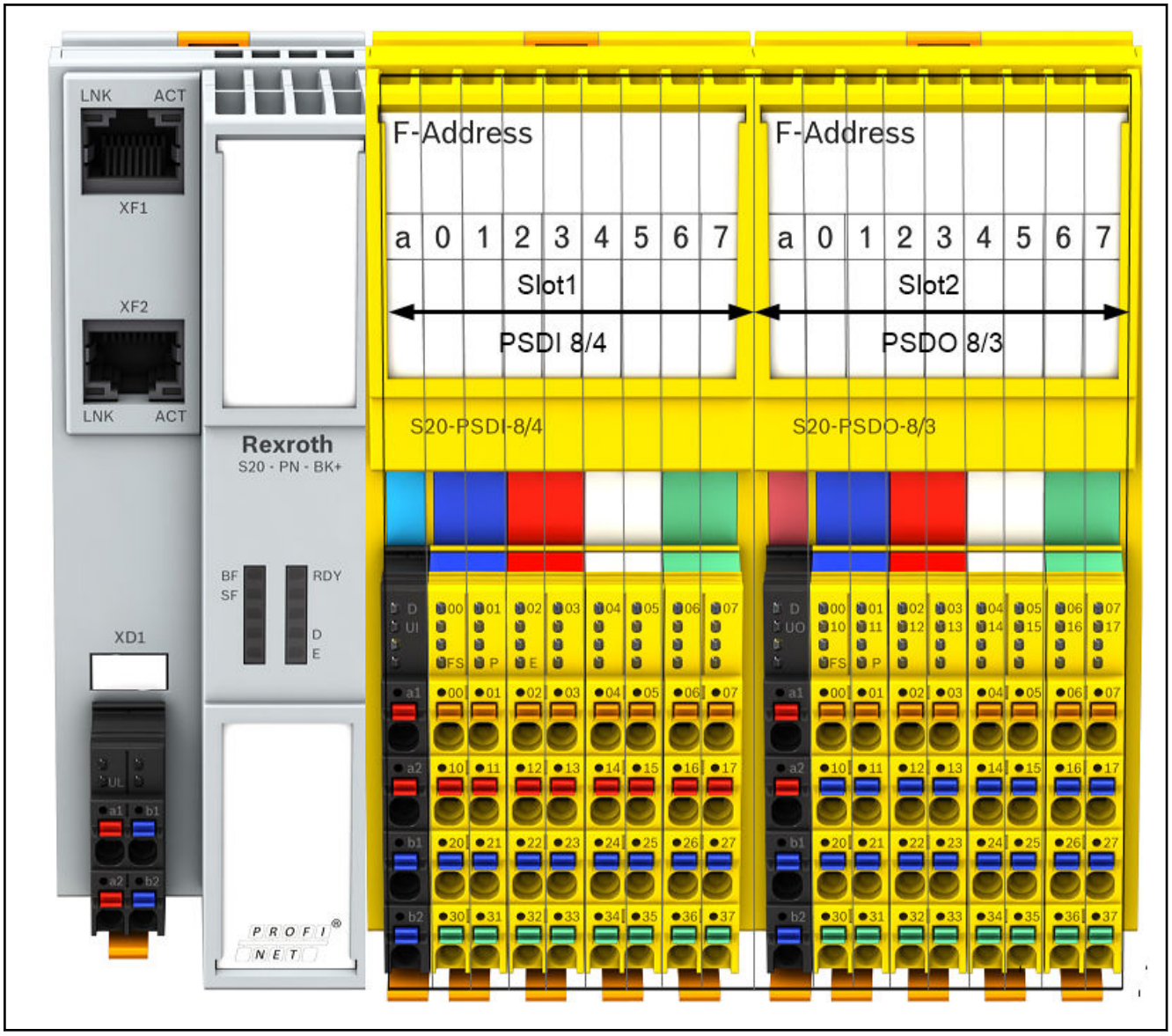

Fig. 4-7: Numbering the slots at the Profinet S20 coupler S20-PN-BK+

4.1.7 S20-PSDI 8/4 (PROFIsafe data input)

Parameterization

Single-channel

Both channels of an input can be operated individually. Parameterize in the safe parameters for the "Both single channel" input.

Dual-channel

Both channels of one input are interconnected. Parameterize in the safe parameters for the input "double channel- equivalent" or for the input "double channel- antivalent".

	SAFEBYTE							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Single-channel	IN3_Ch2	IN3_Ch1	IN2_Ch2	IN2_Ch1	IN1_Ch2	IN1_Ch1	IN0_Ch2	IN0_Ch1
Dual-channel		IN3_Ch1&2		IN2_Ch1&2		IN1_Ch1&2		IN0_Ch1&2
Terminal name	07	06	05	04	03	02	01	00

Tab. 4-10: I/O mapping of the input data with regard to the channel configuration with S20-PSDI 8/4 as example

4.1.8 S20-PSDO 8/3 (PROFIsafe data output)

Parameterization

Single-channel

Both channels of an output can be operated individually. Parameterize in the safe parameters for the "Both single channel" output.

Dual-channel

Both channels of one output are interconnected. Parameterize in the safe parameters for the "double channel" output.

	SAFEBYTE							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Single-channel	OUT3_Ch2	OUT3_Ch1	OUT2_Ch2	OUT2_Ch1	OUT1_Ch2	OUT1_Ch1	OUT0_Ch2	OUT0_Ch1
Dual-channel		OUT3_Ch1 &2		OUT2_Ch1 &2		OUT1_Ch1 &2		OUT0_Ch1& 2
Terminal name	07	06	05	04	03	02	01	00

Tab. 4-11: I/O mapping of the output data with regard to the channel configuration with S20-PSDO 8/3 as example

4.2 CSos

4.2.1 General information

CSos, also called CIP-Safety on Sercos, is a safe field bus protocol for Sercos devices.

One CSos connection is established between the safety control and each CSos device at the Sercos bus for each direction of communication. The safety control is the originator. The CSos device is called "target".

Addressing

Each CSos device in a Sercos bus as well as the safety control have to be provided with the same SNN (Safe Network Number). Each CSos device at the Sercos bus additionally requires a unique SDID (Safety Device ID). The SDID should have no reference to the Sercos address or the device position in the ring.

SNN and SDID result together in the TUNID (Target Unique Network Identifier) of the bus device. The safety control is also provided with an SNN and SDID together resulting in OUNIT (Originator Unique Network Identifier).

Sercos communication phases

A connection can only be established when the Sercos bus is in P4. If the bus is switched to P2 or P0, the CSos connections are canceled.

Signal propagation delay

The propagation time mainly depends on the Sercos cycle time and the cycle time of the SafeLogic PLC. As the Sercos cycle time is low compared to the SafeLogic cycle time, it is not considered for the calculation.

The following settings have to be made:

EPI = 0 (default)

TiMu = 1 (default)

NTE ≥ (TiMu+3) * Safety cycle time

Use the NTE (Network Time Expectation) to calculate the signal propagation delay.

4.2.2 CSos S20 I/Os

Maximum extension stage

To operate CSos I/Os, the Sercos S20 bus coupler of type S20-S3-BK+ is used. Up to 63 S20 terminals can be connected. The number of the Csos S20 terminals is not limited. However, it is recommended not to use more than 16 CSos S20 terminals per S20 Sercos coupler S20-PN-BK+. To ensure the voltage supply of the CSos terminals, always provide a power feed of logic supply voltage for all six SSDx modules.

Setting the DIP switches

The DIP switches have to be set as follows:

- SDID: Corresponds to the SDID in the configuration dialog. The SDIDs 1 - 127 are permitted
- SNN: The set value + 600000000_{hex} corresponds to the SNN in the “Configuration” dialog. The SNNs from 600000001_{hex} to $60000001F_{hex}$ are permitted
- The combination from SNN and SDID is TUNID (Target Unique Network Identifier). This address has to be unique in the complete project

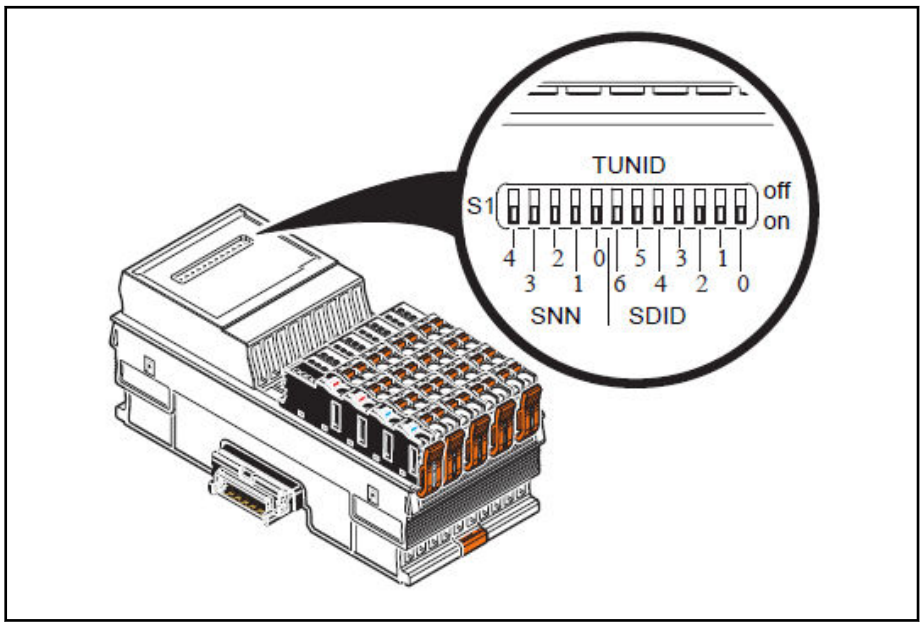

Fig. 4-8: DIP switch of a CSos S20 terminal

The imprint on the housing is valid. The digits directly printed on the switch can be ignored.

SNN = $1F_{hex}$ and SDID = 127 may not be set at the same time, as this combination is a test state.

The terminal always goes into a safe state when there is a change at the DIP switches at runtime.

After setting the DIP switches, the current supply has to be interrupted once for the terminal to apply the new settings.

4.2.3 S20-SSDI 8/4 (CSos data input)

Parameterization

Single-channel

Both channels of an input can be operated individually. Parameterize in the safe parameters for the "Both single channel" input.

Dual-channel

Both channels of one input are interconnected. Parameterize in the safe parameters for the input "double channel- equivalent" or for the input "double channel- antivalent".

	SAFEBYTE							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Single-channel	IN3_Ch2	IN3_Ch1	IN2_Ch2	IN2_Ch1	IN1_Ch2	IN1_Ch1	IN0_Ch2	IN0_Ch1
Dual-channel		IN3_Ch1&2		IN2_Ch1&2		IN1_Ch1&2		IN0_Ch1&2
Terminal name	07	06	05	04	03	02	01	00

Tab. 4-12: I/O mapping of the input data with regard to the channel configuration with S20-SSDI 8/4 as example

4.2.4 S20-SSDO 8/3 (CSos data output)

Parameterization

Single-channel

Both channels of an output can be operated individually. Parameterize in the safe parameters for the "Both single channel" output.

Dual-channel

Both channels of one output are interconnected. Parameterize in the safe parameters for the "double channel" output.

	SAFEBYTE							
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Single-channel	OUT3_Ch2	OUT3_Ch1	OUT2_Ch2	OUT2_Ch1	OUT1_Ch2	OUT1_Ch1	OUT0_Ch2	OUT0_Ch1

Dual-channel		OUT3_Ch1 &2		OUT2_Ch1 &2		OUT1_Ch1 &2		OUT0_Ch1& 2
Terminal name	07	06	05	04	03	02	01	00

Tab. 4-13: I/O mapping of the output data with regard to the channel configuration with S20-SSDO 8/3 as example

4.2.5 IndraDrive

IndraDrive control sections of the second generation (CSxx2.x) can be equipped with a safety module. To control safety-related functions, use the firmware MPx18V08 or higher.

Depending on the safety engineering option, different safety functions are available. The following table provides an overview on the available safety functions for each safety engineering option:

			Safety engineering option			
			L3	L4	S3	S4
SI functions EN 61800-5-2	STO	Safe Torque Off	SIL3	SIL3	SIL3	SIL3
	SBC	Safe Brake Control		SIL3	SIL3	SIL3
	SS1	SafeStop 1 (time monitored)			SIL3	SIL3
	SS1	SafeStop 1 (monitoring delayed)			SIL2/3	SIL2/3
	SS2	Safe Stop 2			SIL2/3	SIL2/3
	SLS	Safe Limited Speed			SIL2/3	SIL2/3
	SDI	Safe Direction			SIL2/3	SIL2/3
	SLI	Safely Limited Increment			SIL2/3	SIL2/3
	SLT	Safely Limited Torque				
	SLP	Safely Limited Position			SIL2	SIL2
	SCA	Safe CAM			SIL2	SIL2
SI functions Rexroth	SMS	Safe Maximum Speed			SIL2/3	SIL2/3
	SMD	Safely Monitored Deceleration			SIL2/3	SIL2/3
	SDL	Safe Door Locking			SIL3	SIL3
Selection and feedback		2 x 24 V input (OnBoard)	SIL3	SIL3		
		16 x 24 V input (zone module optional)			SIL3	
		4 x 2 x 24 V output (zone module optional)			SIL3	
		CIP Safety on Sercos			SIL3	SIL3

Tab. 4-14: Overview on the functional scope of IndraDrive Safety

The following table provides an overview on the available safety options for different drive control sections:

	L3	L4	S3	S4
Economy CSE02.1A	X	X		
Basic CSB02.1A	X	X		

	L3	L4	S3	S4
Basic CSB02.1B	X	X	X	
Advanced CSH02.1B	X	X	X	
Basic double-axis type CDB02.1B	X	X	X	
HCS 1.1			X	
KSM 02.1				X

Tab. 4-15: Availability of the safety options in the drive control sections

5 Commissioning

5.1 Introduction and overview

This chapter describes the commissioning of a safety control and the connected periphery.

5.2 Creating an IndraWorks project

The "IndraWorks SafetyManager" license is installed

1. Create a new IndraWorks project.
2. Create a control

   Open the context menu of the project node and select a control with **Add...** (an MLC XM2 in this example).
3. Control settings

   "Safety CPU" has to be selected as function module.

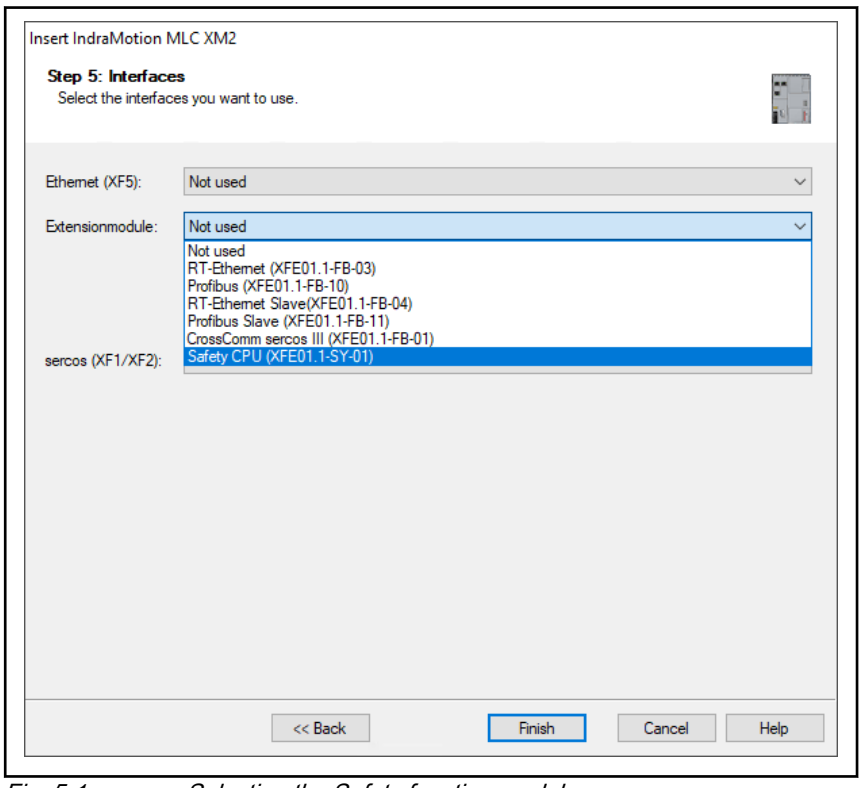

Fig. 5-1: Selecting the Safety function module

The standard control as well as the safety control were created.

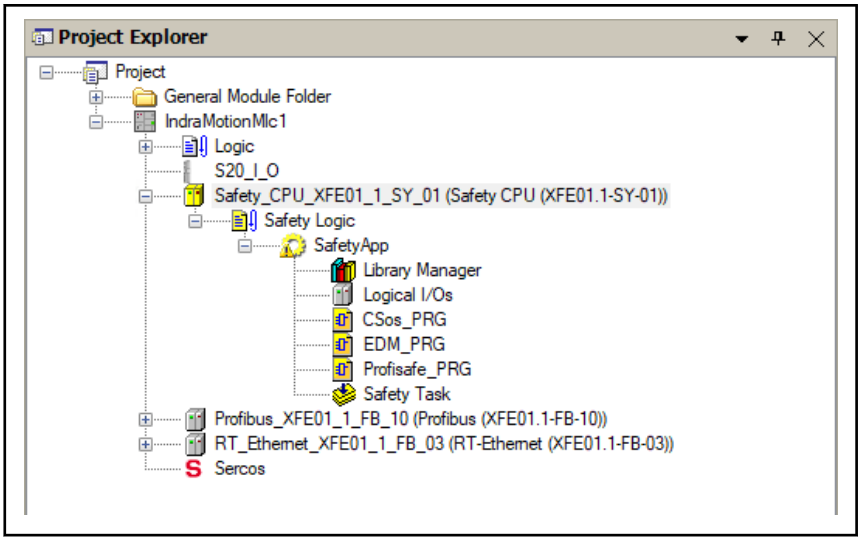

Fig. 5-2: Newly created Safety project

In the screen, the empty programs CSos_PRG (Sercos drives), EDM_PRG (a little program example) and Profisafe_PRG (PROFIsafe I/Os based on Profibus-DP) were created additionally.

5.3 PROFIsafe

5.3.1 Creating and configuring PROFIsafe Inline terminals at the Profibus

PROFISafe Inline terminals can only be operated at one Profibus coupler of type "R-IL PB BK DI8 DO4/CN-PAC" (ordering no. R911172194). This bus coupler has to be created below the "Profibus DP master" node in the project.

1. Inserting PROFIsafe Inline terminals

   The PROFIsafe Inline terminals are inserted at the slave via the context menu item "Add" , in this example below the bus coupler "2 - R_IL_PB_BK…".

   The logic I/O belonging to a PROFIsafe Inline terminal is automatically created below the "Logic I/Os" node.

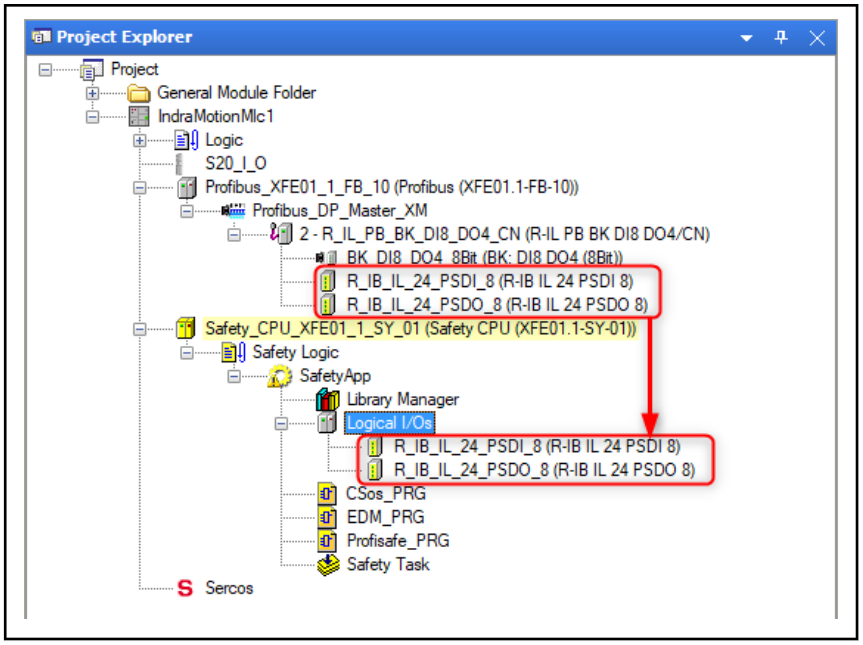

Fig. 5-3: PROFIsafe I/O on the standard control and on the safety control

To which logic I/O the physical I/O is linked, can be seen in the "I/O mapping" of the DP Inline terminal and changed if necessary.

There has to be at least one entry to use the Safety I/Os in the safety control.

The order of the PROFIsafe Inline terminals at the bus coupler has to correspond to the order in the project tree.

2. Parameterizing Safety Inline terminals

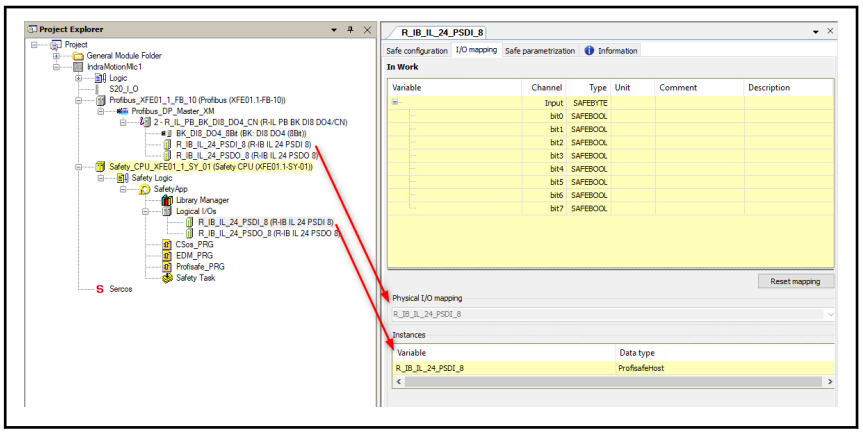

Fig. 5-4: Assigning logic and physical I/Os to a PROFIsafe Inline terminal

The PROFIsafe Inline terminals are parameterized in the dialogs "Safe configuration" and "Safe parameterization" below the "Logic I/O" node. The following parameters for the initial commissioning are important in the log parameters of the "Safe configuration":

- **F_Source_Add:** Address of the PROFIsafe master. This address has to be identical for all devices

- **F_Dest_Add:** PROFIsafe address of the PROFIsafe Inline terminal set at the DIP switches of the PROFIsafe Inline terminal (see "Setting the DIP switches" on page 15)

  The following addresses result from the example fig. 5-4 "Assigning logic and physical I/Os to a PROFIsafe Inline terminal" on page 33:

  R_IB_IL_24_PSDI_8: F_Dest_Add= 22

  R_IB_IL_24_PSDO_8: F_Dest_Add= 23

- **F_iPar_CRC:** Checksum of the "Safe parameterization"

  The checksum has to be applied always after changing the configuration of the safe I/Os

- **F_WD_Time:** This parameter is the monitoring period within a Safety telegram has to be answered

  If this time elapses, the Inline terminal goes into the "Safe state". A default value of 150 ms is generally sufficient for "smaller bus systems". The value should to be at least 2*SafetyTask time

The actual time for the response times can be determined at the ProfisafeHost instances of each connection in the Safety application.

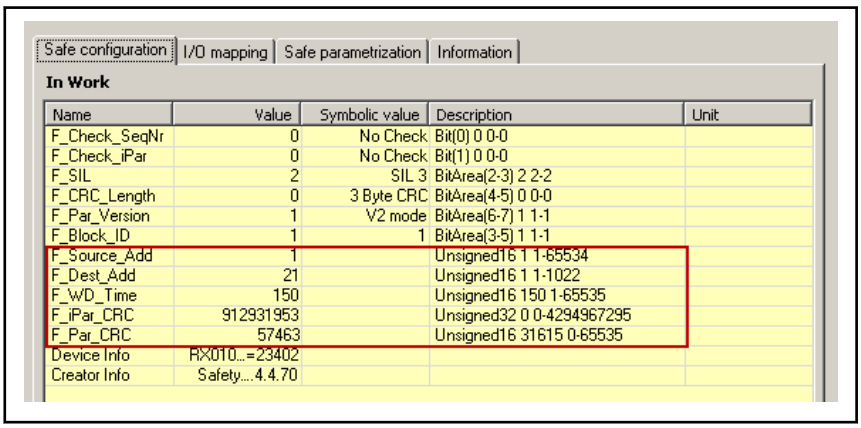

Fig. 5-5: Setting the safe configuration of a PROFIsafe Inline terminal

The parameters set in the "Safe parameterization" dialog are device-specific. For the settings, refer to the documentation of the corresponding I/O Inline terminal.

The following parameters are relevant for the initial commissioning:

- **F_Dest_Add:** PROFIsafe address of the PROFIsafe Inline terminal set at the DIP switches of the PROFIsafe Inline terminal (see "Setting the DIP switches" on page 15)

  The address has to be identical to the F_DEST_Add set in the "Safe configuration" tab

- **I_Par_CRC32:** It is automatically calculated and has to be updated manually in the "Safe configuration" after changing the safe parameters

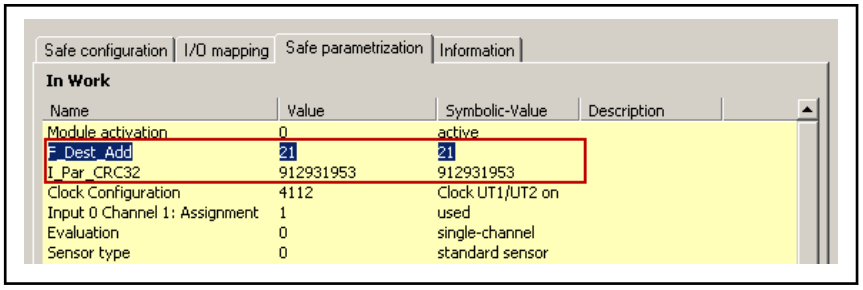

Fig. 5-6: Setting the safe parameterization of a PROFIsafe Inline terminal

The Inline terminals can be operated as single channel or double channel. That means that four to eight I/Os are available for each Inline terminal. Each channel has to be enabled separately.

For more information on the parameterization, refer to chapter 4.1.3 "Inline PSDI 8 and PSDI 16 (PROFIsafe data input)" on page 19 and chapter 4.1.4 "Inline PSDO 8 and PSDOR 4 (PROFIsafe data output)" on page 20.

After the parameterization, "I_Par_CRC32" has to be copied from the "Safe parameterization" dialog to the "Safe configuration" (F_iPar_CRC) dialog.

Note that **F_Dest_Add** and **F_Source_Add** may not have the same value.

The safe inputs and outputs are parameterized (offline).

These parameters are only loaded and thus applied to the control after logging into the standard control and into the safety control!

3. Creating a program

   Create the “Profisafe_PRG” program via the context menu item "Add" at the "SafetyApp". Decide whether to work on "Basic level" (Safety Basic-POU) or on "Extended level" (Safety ExtendedPOU).

We work on "Extended level".

"Basic level" does not permit writing on WORD variables for example. Thus, diagnostics ("DiagCode" output) cannot be issued at the Safety PLCopen function blocks.

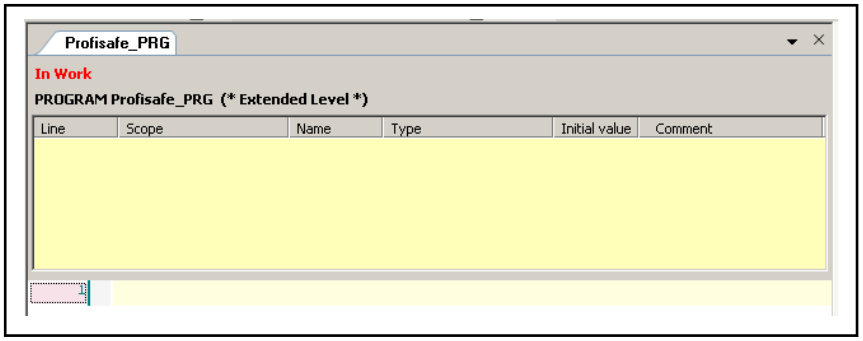

Fig. 5-7: "Profisafe_PRG" program with declaration part and a network

4. Inserting "ProfisafeHost" function blocks into the program

   The "ProfisafeHost" function block monitors the safe communication. The function block is automatically created for each PROFIsafe Inline terminal.

   To activate the function block, a reference has to be set to the instance of the function block in a Safety program or a Safety function block.

   A function block has to be inserted via the context menu item **Insert FB call**. After selecting the menu item, the input assistant is displayed. The "ProfisafeHost" function block can be selected in this input assistant.

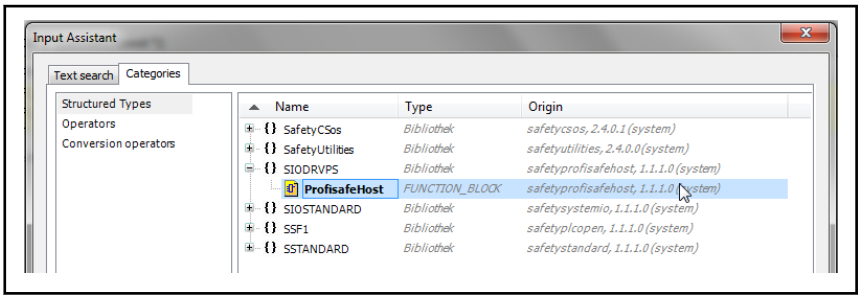

Fig. 5-8: Selecting the "ProfisafeHost" function block

After creating the function block, connect the function block with an instance.

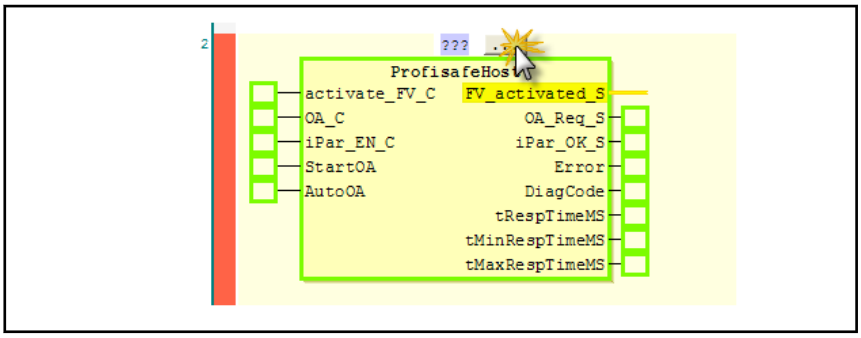

Fig. 5-9: Setting the "ProfisafeHost" instance

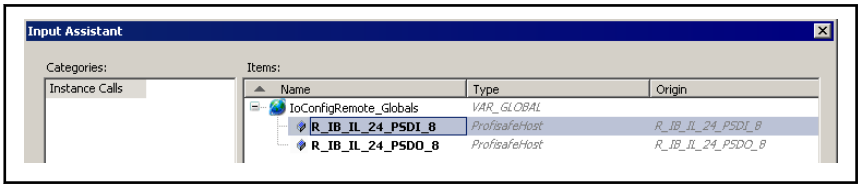

Fig. 5-10: Selecting the "ProfisafeHost" instance in the input assistant

Activate as shown in the following figure.

The inputs "activate_FV_C", "ipar_EN_C" and "AutoOA" are preassigned with "FALSE". The "StartOA" input is preassigned with "TRUE".

Attach the F_Dest address to variable names to obtain a unique assignment.

The outputs can be summarized to group status messages using OR operations.

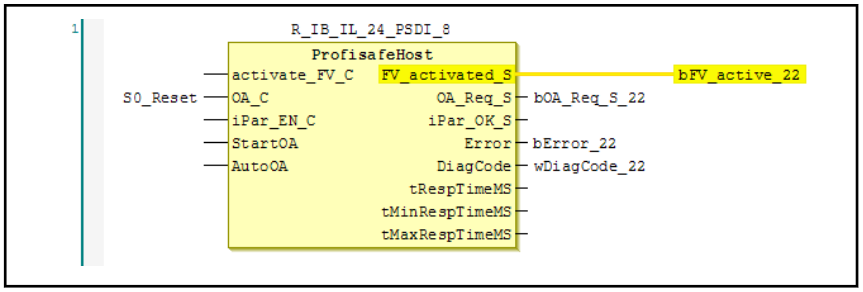

Fig. 5-11: Created and activated "ProfisafeHost" instance

5.3.2 Creating and configuring PROFIsafe Inline terminals at the Profinet

PROFISafe Inline terminals can only be operated at one Profinet coupler of type "R-IL PN BK DI8 DO4" (ordering no. R911171944). This bus coupler has to be created below the "Profinet I/O controller" node in the project.

The easiest method to apply the bus configuration to the project is to scan the Profinet bus and to apply the devices found to the project. This is also applicable to the S20-PROFIsafe and to any Profinet devices. The prerequisite is that the GSDML was imported in IndraWorks before. If there is no hardware when projecting, proceed as for the PROFIsafe terminals under Profibus.

1. Searching devices at the Profinet

   Starting the search via the menu item "Scan devices…" in the context menu of the node "PROFINET_IO_Controller".

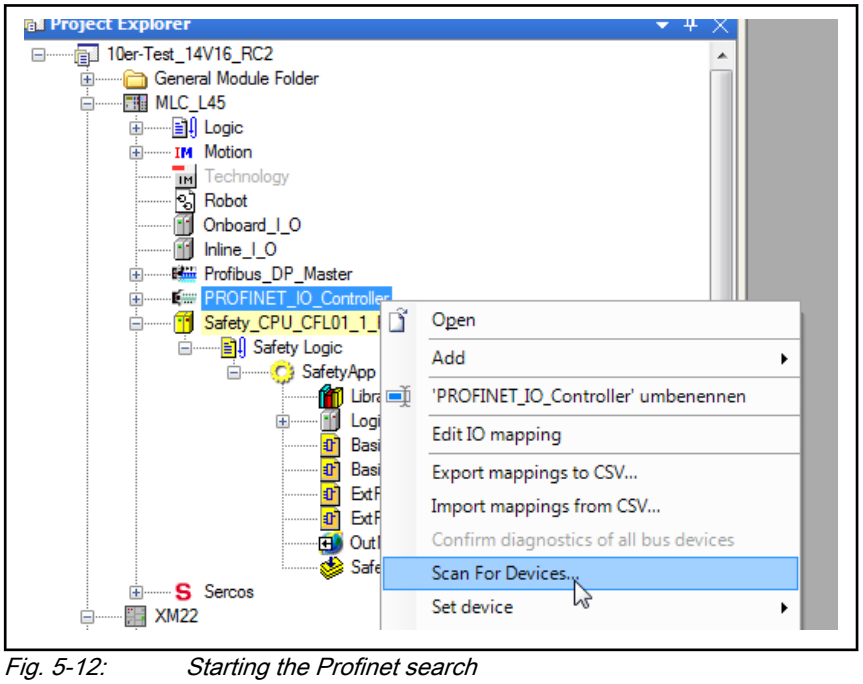

Fig. 5-12: Starting the Profinet search

   A list is displayed containing the Profinet devices found at the Profinet controller.

2. This step is only required for PROFIsafe Inline modules.

   The "R-IL PN BK DI8 DO4" bus couplers have to be searched and expanded in the list of the Profinet devices found. The found PROFIsafe modules are located below the bus coupler. As the ident number of all

PROFIsafe Inline modules is identical, assign the module types manually. Click on the selection in the "Device type" field.

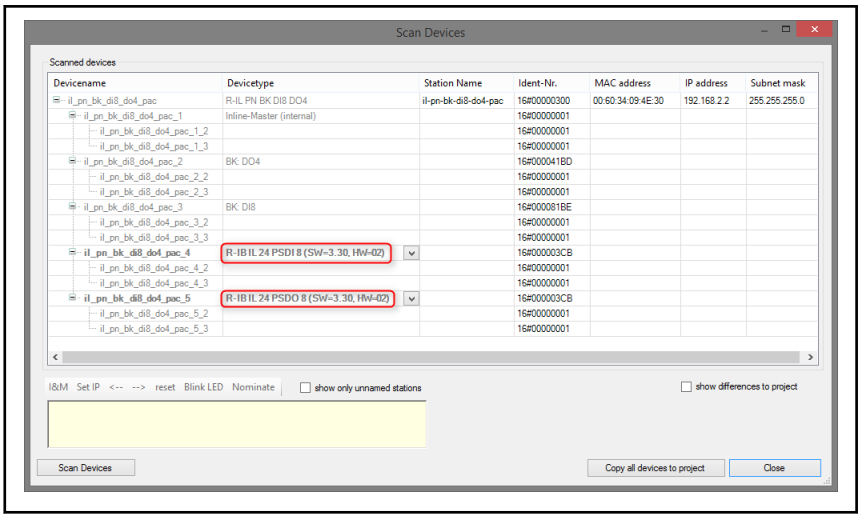

Fig. 5-13: Selection lists for PROFIsafe Inline device types in the Profinet search result

Select the correct device type in the window.

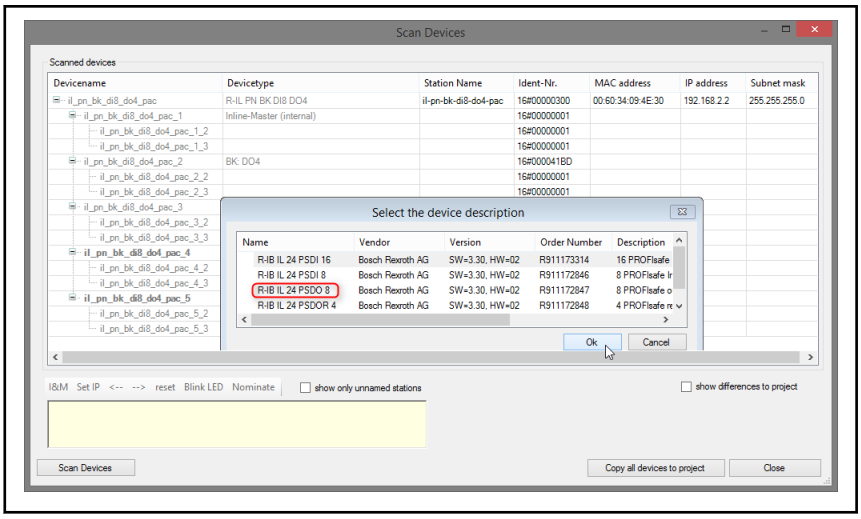

Fig. 5-14: Selecting the Inline device type for PROFIsafe modules

3. Click on “Copy all devices to project” to apply the found Profinet devices to the project.

   The logic I/O belonging to a PROFIsafe Inline terminal is automatically created below the "Logic I/Os" node.

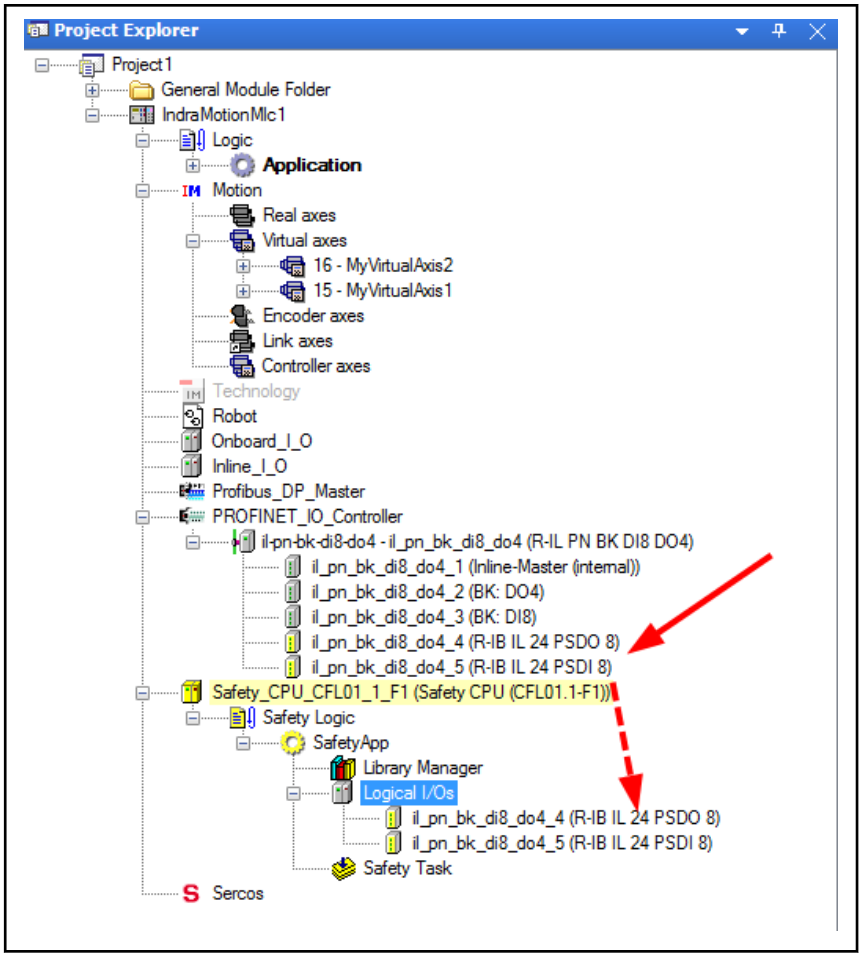

Fig. 5-15: Inserted Profinet devices including the PROFIsafe terminals

4. Parameterizing PROFIsafe terminals

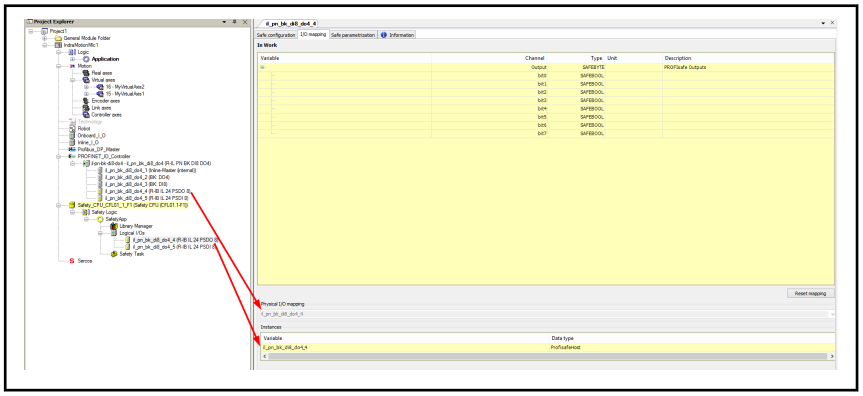

Fig. 5-16: Assigning logic and physical I/Os to a PROFIsafe terminal

The PROFIsafe terminals are parameterized in the dialogs "Safe configuration" and "Safe parameterization" below the "Logic I/O" node. The following parameters for the initial commissioning are important in the log parameters of the "Safe configuration":

- **F_Source_Add:** Address of the PROFIsafe master. This address has to be identical for all devices
- **F_Dest_Add:** PROFIsafe address of the PROFIsafe Inline terminal set at the DIP switches of the PROFIsafe Inline terminal (see "Setting the DIP switches" on page 15)

The following addresses result from the example fig. 5-4 "Assigning logic and physical I/Os to a PROFIsafe Inline terminal" on page 33:

R_IB_IL_24_PSDI_8: F_Dest_Add= 22

R_IB_IL_24_PSDO_8: F_Dest_Add= 23

- **F_iPar_CRC:** Checksum of the "Safe parameterization"

  The checksum has to be applied always after changing the configuration of the safe I/Os

- **F_WD_Time:** This parameter is the monitoring period within a Safety telegram has to be answered

  If this time elapses, the Inline terminal goes into the "Safe state". A default value of 150 ms is generally sufficient for "smaller bus systems". The value should to be at least 2*SafetyTask time

The actual time for the response times can be determined at the ProfisafeHost instances of each connection in the Safety application.

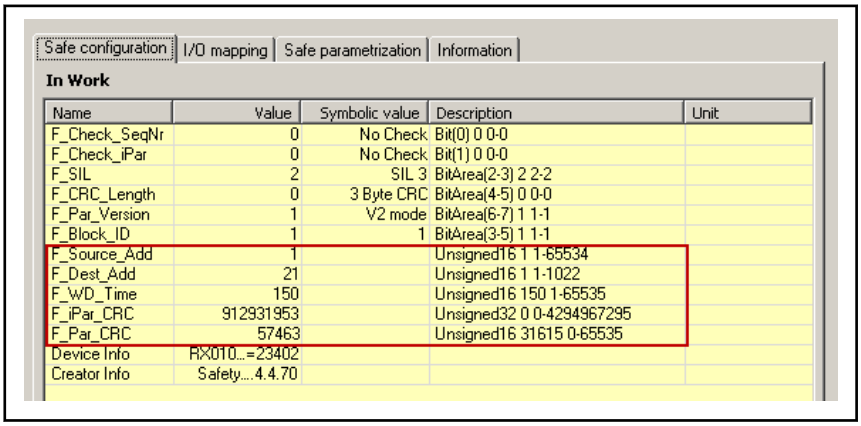

Fig. 5-17: Setting the safe configuration of a PROFIsafe Inline terminal

The parameters set in the "Safe parameterization" dialog are device-specific. For the settings, refer to the documentation of the corresponding I/O terminal.

The following parameters are relevant for the initial commissioning:

- **F_Dest_Add:** PROFIsafe address of the PROFIsafe Inline terminal set at the DIP switches of the PROFIsafe terminal (see "Setting the DIP switches" on page 15)

  The address has to be identical to the F_DEST_Add set in the "Safe configuration" tab

- **I_Par_CRC32:** It is automatically calculated and has to be updated manually in the "Safe configuration" after changing the safe parameters

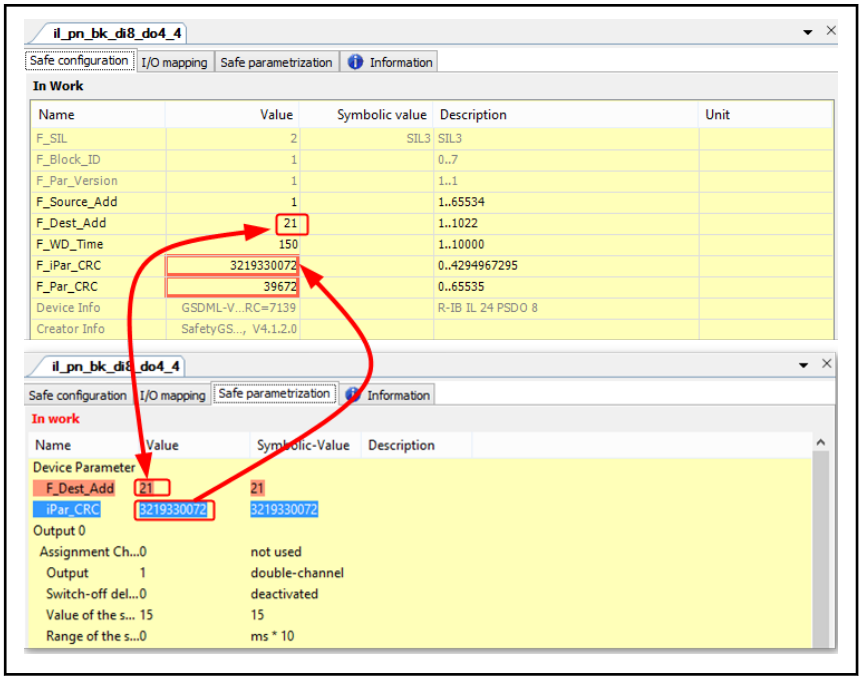

Fig. 5-18: Setting the safe parameterization of a PROFIsafe terminal

The PROFIsafe terminals can be operated as single channel or double channel. That means that four to eight I/Os are available for each Inline terminal. Each channel has to be enabled separately.

For more information on the parameterization, refer to chapter 4.1.3 "Inline PSDI 8 and PSDI 16 (PROFIsafe data input)" on page 19 and chapter 4.1.4 "Inline PSDO 8 and PSDOR 4 (PROFIsafe data output)" on page 20.

After the parameterization, "I_Par_CRC32" has to be copied from the "Safe parameterization" dialog to the "Safe configuration" (F_iPar_CRC) dialog.

Note that **F_Dest_Add** and **F_Source_Add** may not have the same value.

The safe inputs and outputs are parameterized (offline).

These parameters are only loaded and thus applied to the control after logging into the standard control and into the safety control!

5. Creating a program

   Create the “Profisafe_PRG” program via the context menu item "Add" at the "SafetyApp". Decide whether to work on "Basic level" (Safety Basic-POU) or on "Extended level" (Safety ExtendedPOU).

We work on "Extended level".

"Basic level" does not permit writing on WORD variables for example. Thus, diagnostics ("DiagCode" output) cannot be issued at the Safety PLCopen function blocks.

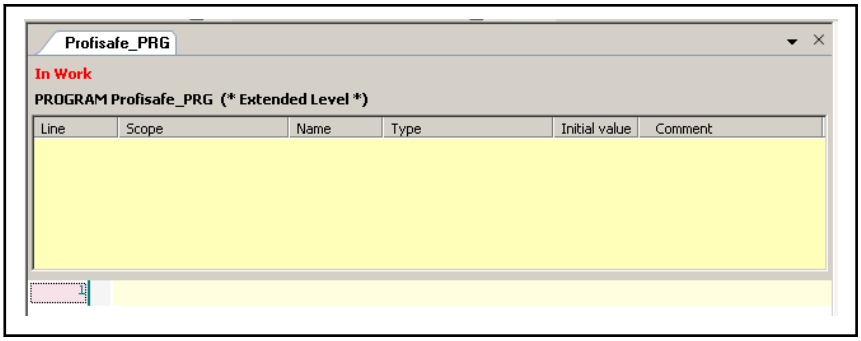

Fig. 5-19: "Profisafe_PRG" program with declaration part and a network

6. Inserting "ProfisafeHost" function blocks into the program

   The "ProfisafeHost" function block monitors the safe communication. The function block is automatically created for each PROFIsafe Inline terminal.

   To activate the function block, a reference has to be set to the instance of the function block in a Safety program or a Safety function block.

   A function block has to be inserted via the context menu item **Insert FB call**. After selecting the menu item, the input assistant is displayed. The "ProfisafeHost" function block can be selected in this input assistant.

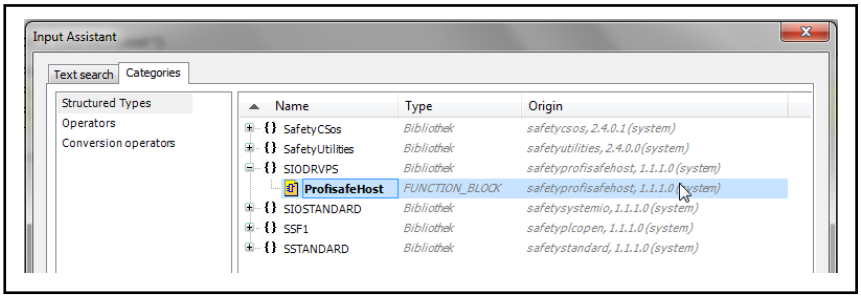

Fig. 5-20: Selecting the "ProfisafeHost" function block

After creating the function block, connect the function block with an instance.

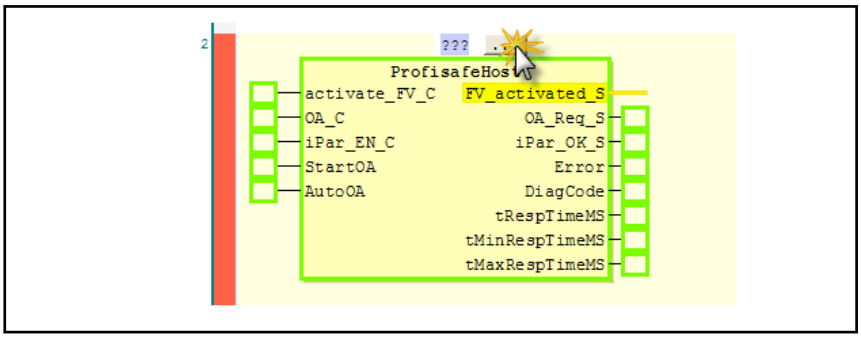

Fig. 5-21: Setting the "ProfisafeHost" instance

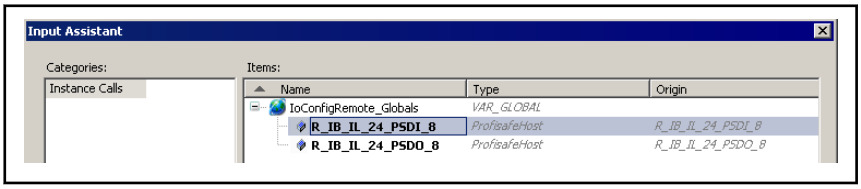

Fig. 5-22: Selecting the "ProfisafeHost" instance in the input assistant

Activate as shown in the following figure.

The inputs "activate_FV_C", "ipar_EN_C" and "AutoOA" are preassigned with "FALSE". The "StartOA" input is preassigned with "TRUE".

Attach the F_Dest address to variable names to obtain a unique assignment.

The outputs can be summarized to group status messages using OR operations.

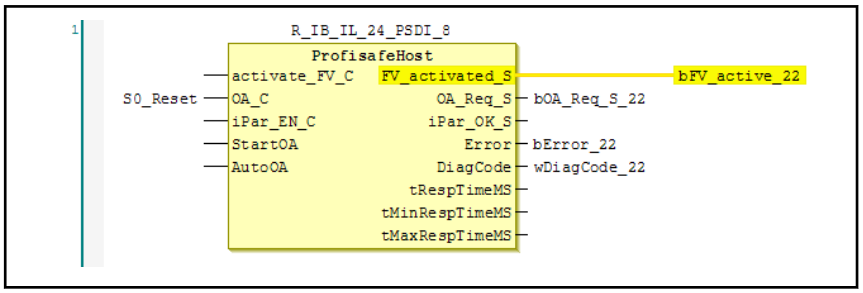

Fig. 5-23: Created and activated "ProfisafeHost" instance

5.4 CSos

5.4.1 Creating and configuring CSos S20 terminals

1. Inserting CSos S20 terminals

   CSos S20 terminals are inserted at the slave via the context menu item "Add" below the Sercos node, here "S20-S3-BK+".

   The CSos S20 terminals are automatically created below "Logic I/Os".

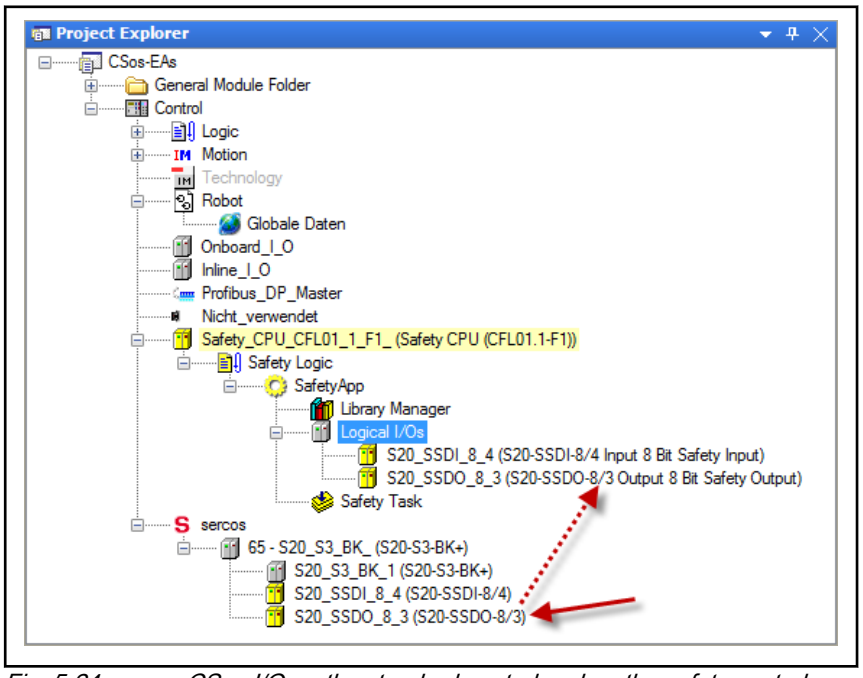

Fig. 5-24: CSos I/O on the standard control and on the safety control

To which logic I/O the physical I/O is linked, can be seen in the "I/O mapping" of the CSos S20 terminal and changed if necessary.

There has to be at least one entry to use the Safety I/Os in the safety control.

The order of the CSos S20 terminals at the bus coupler has to correspond to the order in the project tree.

2. Parameterizing CSos S20 terminals

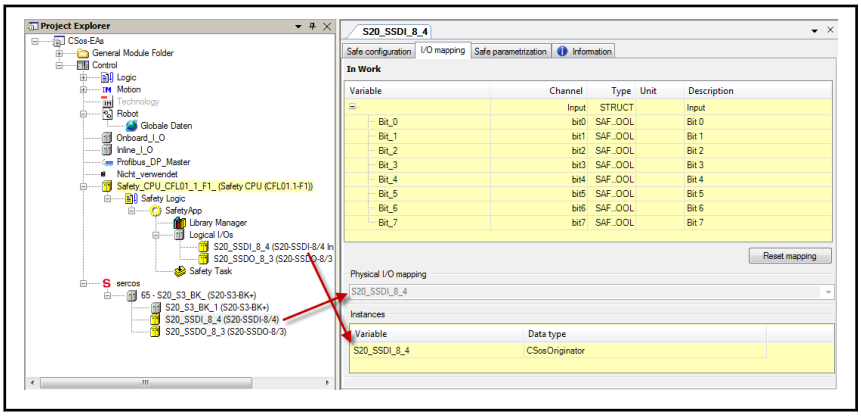

Fig. 5-25: Assigning logic and physical I/Os to a CSos S20 terminal

The CSos S20 terminals are parameterized in the dialogs "Safe configuration" and "Safe parameterization" below the "Logic I/Os" node. The following parameters for the initial commissioning are important in the log parameters of the "Safe configuration":

- **Safety Network Number (SNN):** Uniquely identifies the network via all safety systems. The default value is 0x600000001. The maximum value for CSos S20 I/Os is 0x6000001F (see "Setting the DIP switches" on page 27)
- **Safety Device ID (SDID):** Logical address of the CSos S20 I/Os. The default value is 1, the maximum value for CSos S20 I/Os is 127 (see "Setting the DIP switches" on page 27)
- **Safety Configuration CRC (SCCRC):** Checksum of the "Safe parameterization".

  The checksum always has to be applied after changing the parameterization of the safe I/Os
- **Network Time Expectation (NTE):** This parameter is the monitoring period within which a Safety telegram has to be answered.

  If this time elapses, the S20 terminal goes into the "Safe state". A default value of 50 ms is generally sufficient for "smaller bus systems".

  The required NTE can be calculated as follows: NTE ≥ (TiMu+3) * Safety cycle time [ms].

  The actual time for the response can be determined via the CSosOriginator function block of the connection in the Safety application. The signal propagation delay is only available for connections receiving data. No signal propagation time is available for sending connections.

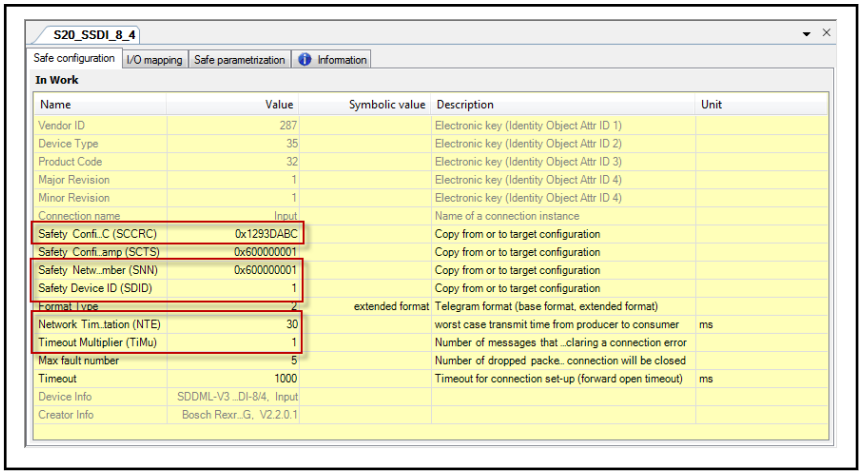

Fig. 5-26: Setting the safe configuration of a CSos S20 terminal

The parameters set in the "Safe parameterization" dialog are device-specific. For the settings, refer to the documentation of the corresponding CSos S20 terminal.

The following parameters are relevant for the initial commissioning:

- **SNN/SDID:** TUNID of the CSos S20 terminal set at the DIP switches of the CSos S20 terminal (see "Setting the DIP switches" on page 27)
- **SCCRC:** It is automatically calculated and has to be updated after changing the safe parameters manually in the "Safe configuration"

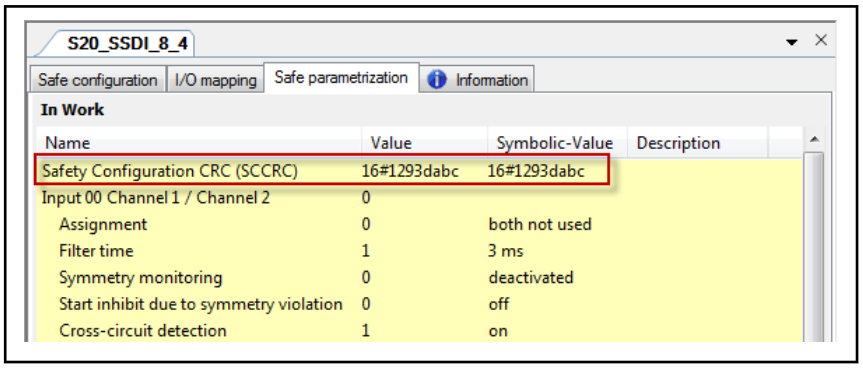

Fig. 5-27: Setting the safe parameterization of a CSos S20 terminal

The CSos S20 terminals can be operated as single channel or double channel. Thus, four to eight I/Os are available for each CSos S20 terminal.

After the parameterization, "SCCRC" has to be copied from the "Safe parameterization" dialog to the "Safe configuration" dialog.

The safe inputs and outputs are parameterized (offline).

These parameters are only loaded to the control by logging into the standard control and into the safety control!

3. Inserting "CSosOriginator" function blocks into the program

   The "CSosOriginator" function block monitors the safe communication to the CSos S20 terminals.

One CSosOriginator instance is created for each CSos terminal. To activate the function block, a reference has to be set to the instance of the function block in a Safety program or a Safety function block.

The procedure is analog to that of the "ProfisafeHost" function blocks (see fig. 5-9 "Setting the "ProfisafeHost" instance" on page 36).

The "CSosOriginator" is located under "SafetyCSos". The inputs "activate_FV_C" and "AutoOA" are preassigned with "FALSE". The "StartOA" input is preassigned with "TRUE".

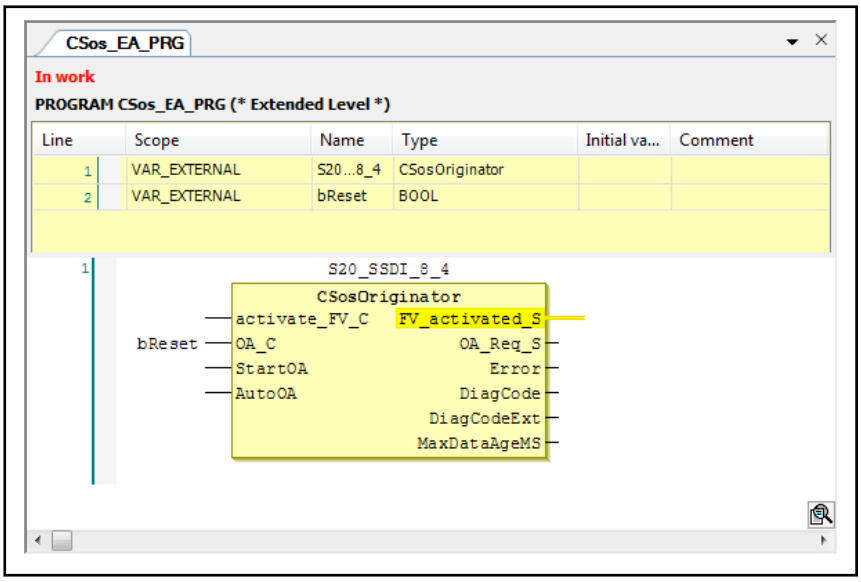

Fig. 5-28: Activating the CSos instances of a CSos S20 terminal

5.4.2 IndraDrive with CSos

The drive firmware supports CSos

1. Creating drives

   Drag IndraDrive on the "Sercos" node in the Project Explorer and drop it.

   The drive was inserted below the **Sercos** node and below **Motion ▸ Real axes**.
   Existing drives can also be scanned at the Sercos bus.

2. Configuring CIP Safety on Sercos

   After selecting the item **Configuration CIP Safety on Sercos...** in the context menu of the node **Sercos ▸ Drive name ▸ Axis**, the configuration dialog opens. The desired safety function of the drive is selected in the configuration dialog.

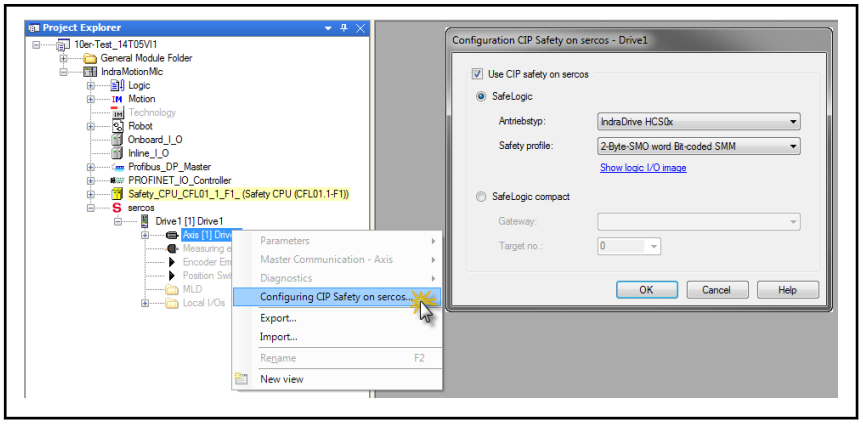

Fig. 5-29: CSos dialog

Both logic I/Os for the CSos functionality were created below the safety control.

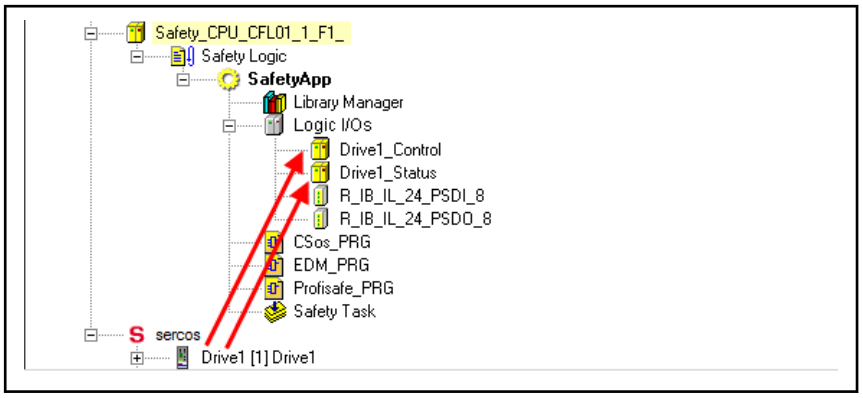

Fig. 5-30: Logic I/Os of the drive "Drive1"

"Control" stands for the information from the safety control to the drive. "Status" stands for the information from the drive to the safety control. The information can be accessed bit by bit (always "I/O mapping" tab).

3. Parameterizing the "Logic I/Os"

   See step 2 in chapter 5.4.1 "Creating and configuring CSos S20 terminals" on page 43.

4. Inserting "CSosOriginator" function blocks into the program

   The "CSosOriginator" function block monitors the safe communication to the drives.

   It is automatically created twice for each CSos drive. A CSos-capable drive is always provided with two channels: One status channel and one control channel. One instance of the "CSosOriginator" function block is created for each channel. To activate the function block, a reference has to be set to the instance of the function block in a Safety program or a Safety function block.

   The procedure is analog to that of the "ProfisafeHost" function blocks (see fig. 5-9 "Setting the "ProfisafeHost" instance" on page 36).

   The "CSosOriginator" is located under "SafetyCSos".

   Note on the activation: The inputs "activate_FV_C" and "AutoOA" are preassigned with "FALSE". The "StartOA" input is preassigned with "TRUE".

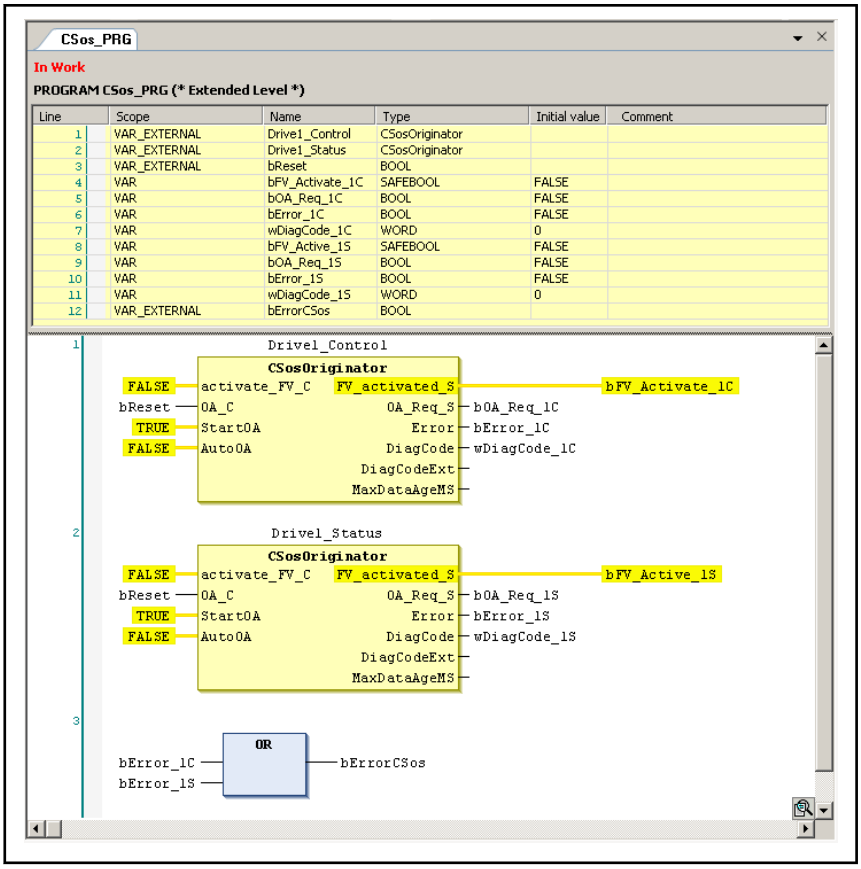

Fig. 5-31: Activating CSos instances of a drive

Controlling the SafeMotion via control bits

Different safe motion modes can be controlled via bits in the control word/ byte:

Variable	Channel	Type	Unit	Description
	Output	STRUCT		2B-Control Bit-SMM
ModeSelectionSignal	bit0	SAF...OL		ModeSelectionSignal
EmergencyStopSignal	bit1	SAF...OL		EmergencyStopSignal
EnablingControl	bit2	SAF...OL		EnablingControl
SMM1Signal	bit3	SAF...OL		SMM1Signal
SMM2Signal	bit4	SAF...OL		SMM2Signal
SMM3Signal	bit5	SAF...OL		SMM3Signal
SMM4Signal	bit6	SAF...OL		SMM4Signal
SMM5Signal	bit7	SAF...OL		SMM5Signal
SMM6Signal	bit8	SAF...OL		SMM6Signal
SMM7Signal	bit9	SAF...OL		SMM7Signal
SMM8Signal	bit10	SAF...OL		SMM8Signal
SMM9Signal	bit11	SAF...OL		SMM9Signal
SMM10Signal	bit12	SAF...OL		SMM10Signal
SMM11Signal	bit13	SAF...OL		SMM11Signal
SMM12Signal	bit14	SAF...OL		SMM12Signal
SafeOutput_local	bit15	SAF...OL		SafeOutput_local

Fig. 5-32: Control bits of a drive with Safe Motion

Input combination	Value	SafeMotion function	Function
EmergencyStopSignal ModeSelectionSignal	1 1	None	Normal mode
EmergencyStopSignal	0	SMES	Safe Motion Emergency Stop
EmergencyStopSignal Enabling Control ModeSelectionSignal	1 0 0	SMSS1 SMSS2	Safe Motion Safe Stop 1[1)] Safe Motion Safe Stop 2[1)]
EmergencyStopSignal Enabling Control ModeSelectionSignal SMM1	1 1 0 1	SMM1	Safe Motion Mode 1[2)]
...			
EmergencyStopSignal Enabling Control ModeSelectionSignal SMM16	1 1 0 1	SMM16	Safe Motion Mode 16[2)]

1) Depending on the configuration in the drive, either Safe Stop 1 or Safe Stop 2

2) The enabled operation modes as well as their parameters are specified in the drive configuration

Tab. 5-1: Enabling the Safe Motion mode in the drive

Controlling the SafeMotion via function blocks

Instead of using the bit of the SafeMotion control word, controlling via SafeMotion function blocks is also possible. Variables in the control word and in the status word do not have to be used at all. The drive is referenced via an "AxisID" corresponding to the SDID configured in the SafeMotion.

The SafeMotion function block is in the SafetyDriveMotion library. The following function blocks are available:

- SF_MotionEmergencyStop: Controlling the EmergencyStop functionality of the drive
- SF_MotionSafeStop: Controlling the SafeStop functionality of the drive
- SF_SMMSelector: Controlling an SMM (SafeMotionMode)
- SF_AxisGroup: Summarizes multiple axes to one group.

There are the following advantages when using the function blocks:

- Controlling the SafeMotion functionality is monitored. It is checked whether the drive actually reaches the required state.
- No variables have to be created and interconnected in the control word and in the status word.
- The functions can be modularized easier when using the “AxisID” as only reference.
- By grouping axes with the same safety function, multiple axes can be commanded simultaneously using SafeMotion function blocks.

All function blocks can be operated individually. However, the function block "SF_MotionEmergencyStop" always has to be available if the functionality is used.

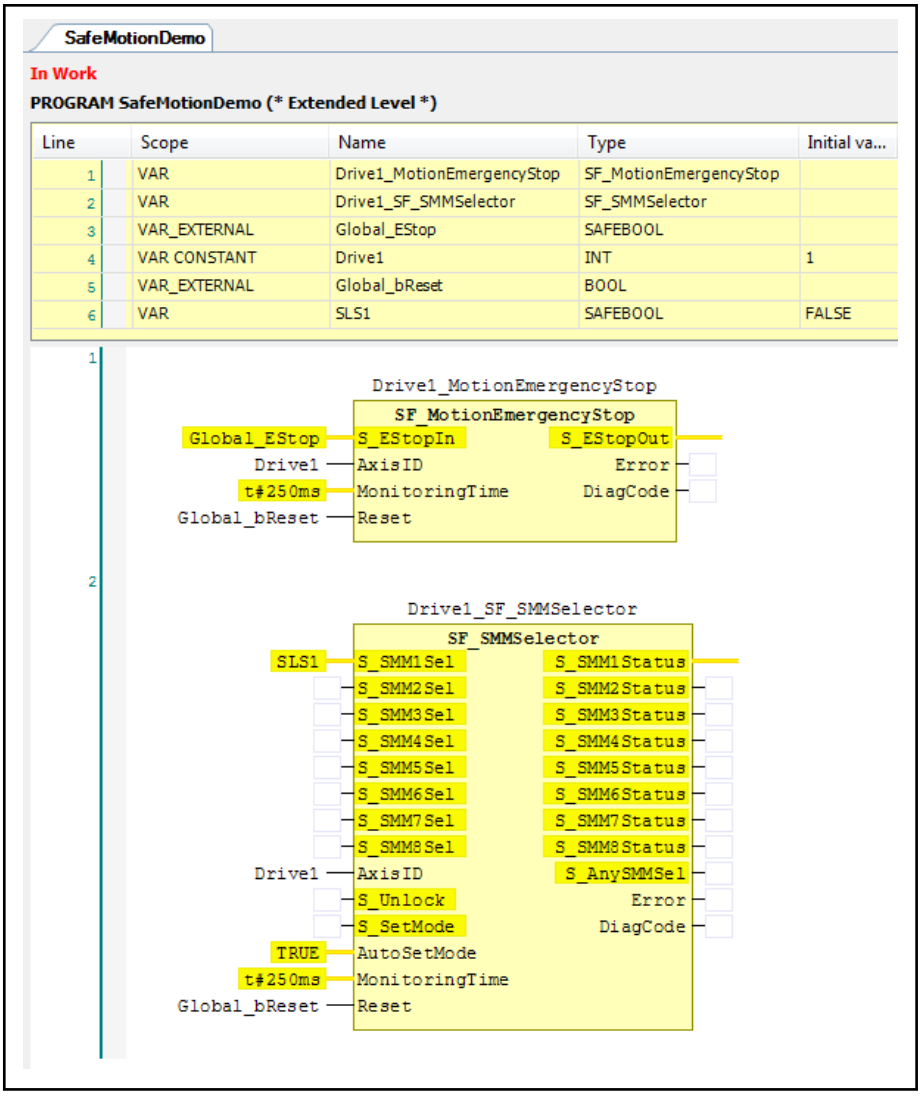

Fig. 5-33: Example program to control one axis with two SafeMotion function blocks

5.5 Exchanging variables with standard PLC

The SafeLogic safety control supports logic exchange variables in order not to exchange safety-relevant information between the standard and the safety control. These can be created with the following steps.

A lot of information on the safety control and on the connected safe field bus devices can be retrieved on the standard PLC also directly via the global variable **SafetyData** (see chapter 7.7 "Diagnostics on the standard PLC control" on page 71). Thus, only a small number of logic exchange variables has to be created.

1. Adding the data exchange with the standard application

   Use logic exchange devices to exchange variables between the standard control (grey PLC) and the safety control (yellow PLC).

   To create a logic exchange device, select the required data type and the direction of the data flow in the context menu of the node **SafetyApp** ▸ **Logic I/Os** below **Add** ▸ **Logic exchange devices**.

The variables can subsequently be created on the safe area in the logic exchange device.

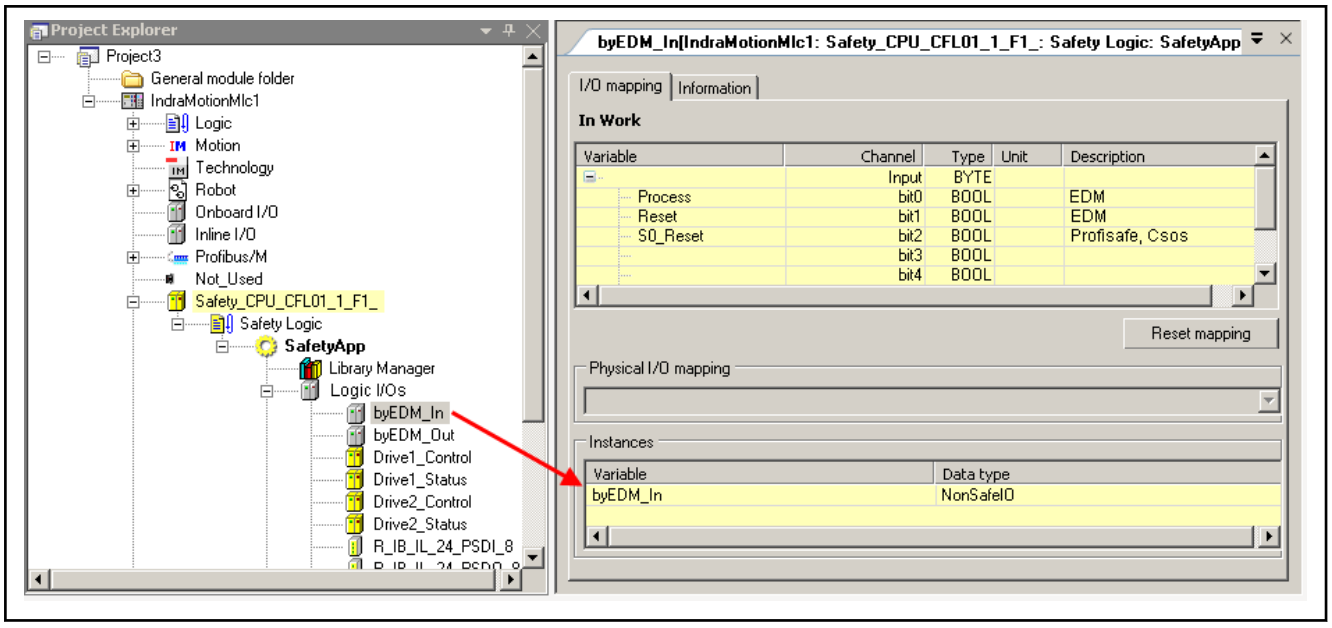

Fig. 5-34: Creating variables in the logic exchange device

Only non-safe variables can be exchanged (see chapter 6.6 "Converting variables (safe ⇔ not safe)" on page 64).

2. In the standard PLC, a Global Variable List has to be subsequently created for the logic data exchange. In the variable list, the corresponding list has to be selected from the safe PLC below "Logical exchange mapping". The variables entered into the list are applied automatically.

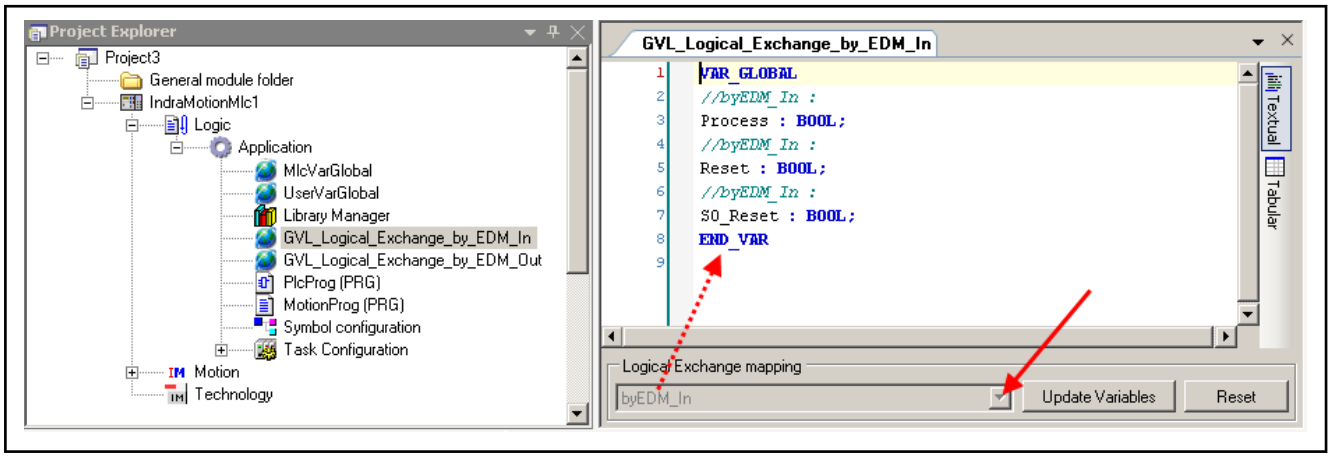

Fig. 5-35: GVL for logic I/Os

If the variable list changes on the safety control, refresh the list.

5.6 Safety network variables

The **NVL** (Safety **N**etwork**V**ariable **L**ist) contains variable declarations transferred to or from another safety control. The Safety network variables can be used in all safety application programs. The Safety network variables are accessed via the objects "Safety network variable list (sender)"" and "Safety network variable list (receiver)".

Safety network variables can be declared VAR_EXTERNAL like global variables in the safety application.

The following data types can be exchanged via network variables:

- SAFEBOOL
- SAFEWORD
- SAFEINT

At the sender, network variables are outputs of the application. That is, one value has to be assigned once in the cycle. At the receiver, network variables are inputs of the application. This means that they may be read, but it may not be written to them.

An NVL sending list can be transferred to multiple receivers simultaneously. The receivers have to log in at the sender to receive the variables. As an individual connection is created for each configured receiver, reduce the number of receivers to a minimum.

The connection monitoring of NVLs is performed via the function blocks "NetVarReceiver" and "NetVarSender". The required instances are created automatically. Function blocks cannot access network variables.

To use Safety network variables, at least two Safety controls have to be contained in the project. Both controls have to communicate to each other via Ethernet.

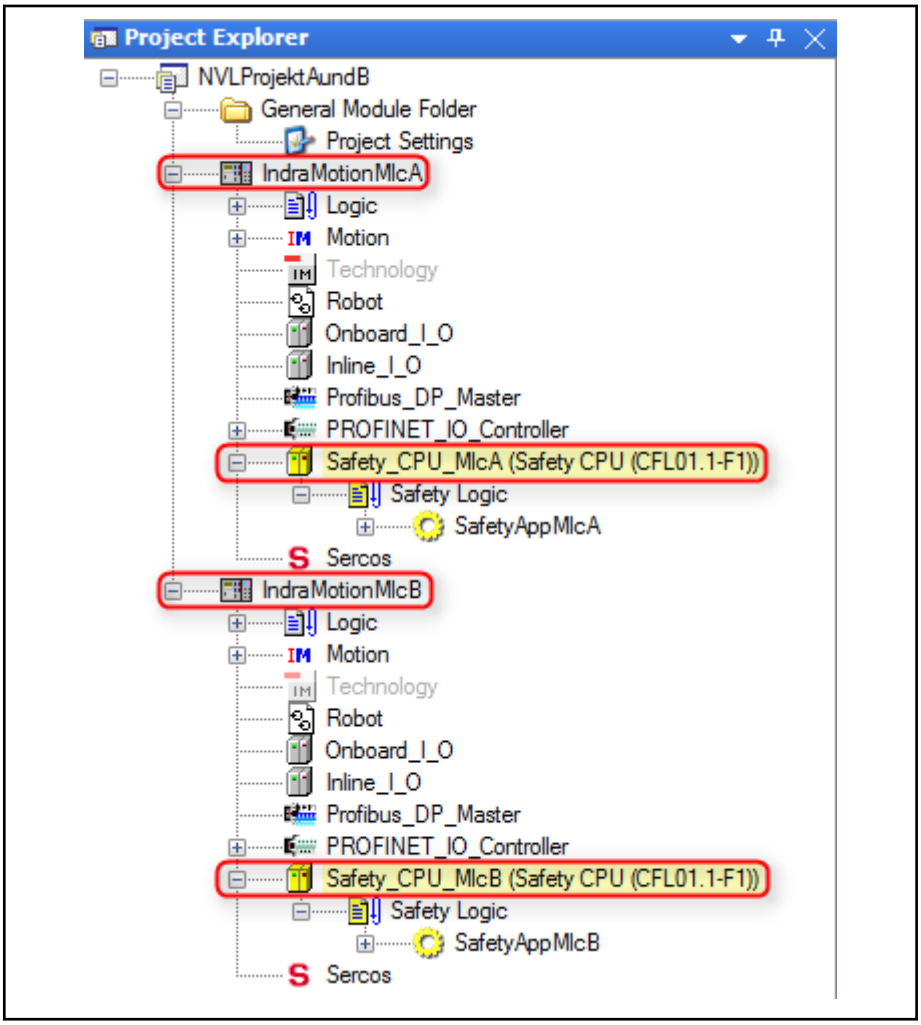

Fig. 5-36: Creating multiple safety controls in one project

The names of the two Safety applications have to be different.

Creating a safe sender GVL

1. Open the context menu of the Safety application on the safe sending control and select "Safety NVL sender…".

   A new Safety NVL sending list is inserted below the Safety application.

2. Set the Safety address of the sender and the required number of receivers in the "Safety configuration" tab (by default, the maximum number is set to 4 receiver).

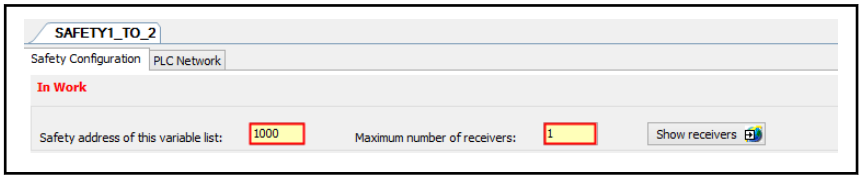

Fig. 5-37: Setting the Safety address and the maximum number of receivers

3. Set the IP address of the receiver controls in the "PLC network" tab (each receiver control has an individual entry).

   Default setting: The controls are addressed via the broadcast address.

However, the default setting generated computational load on all network devices as the broadcast packages are received by all devices and have to be processed individually.

Disable the default setting and directly enter the IP address of the target control.

If standard network variables and Safety network variables are used on the standard control simultaneously, double assignment of the "1202" port can occur as port "1202" is the standard configuration of both network variables.

In this case, the network variables do not work!

To prevent double assignment, it is recommended to set the port configuration of the Safety network variables for receiver and sender to a different port (e.g. "1209").

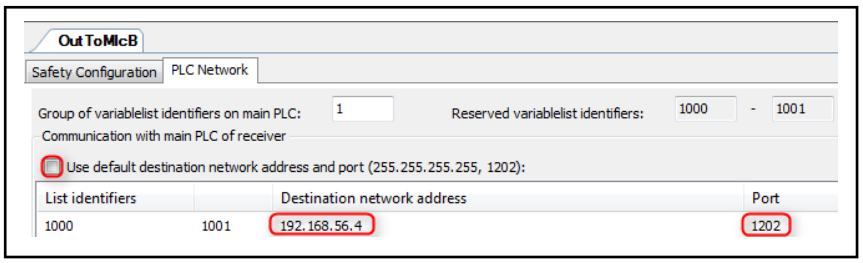

Fig. 5-38: Setting the IP address and the port of the receiver control

4. Add the required or desired variable declarations to the Safety NVL sender.

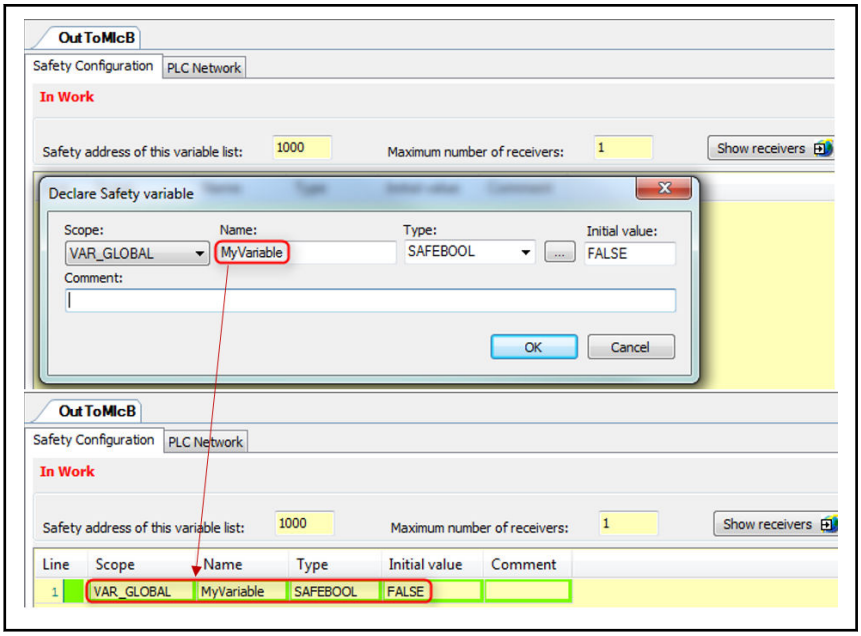

Fig. 5-39: Inserting a variable declaration into an NVL of the sender

Creating a safe receiving NVL

1. Select the safe receiver control in the Project Explorer and open the context menu of the Safety application.

   Select the context command **Add ▸ Safety-NVL receiver…**.

   A new Safety NVL receiving list is inserted below the Safety application.

2. Assign a sender to the receiver in the window of the NVL receiving list.

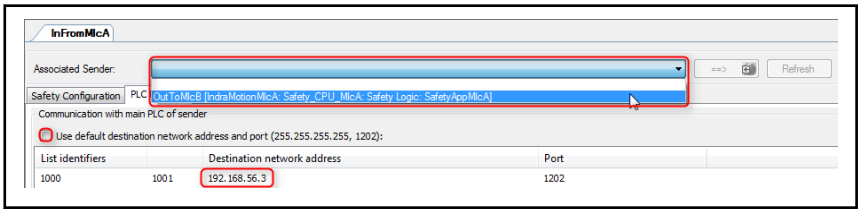

Fig. 5-40: Selecting the NVL sending list

The sending list is assigned to the receiving list.
The list identifier has to be entered manually in the "PLC network" tab.
Use the identifier of the reserved variable list identifiers of the sender.

Disable the default setting and directly enter the IP address of the sender control.
All declared variables of the NVL sending list are applied to the NVL receiving list.

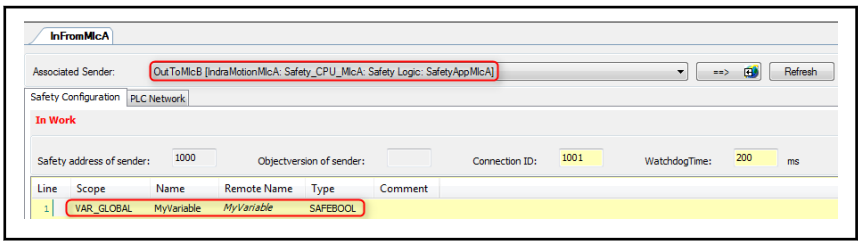

Fig. 5-41: Applying safe network variables of the sending NVL and the receiving NVL

To operate the safe NVL, one function block is necessary at the sender and one at the receiver. This is the function block "NetVarSender" for the sender NVL and the function block "NetVarReceiver" for the receiver NVL. These are created with NVLs and have to be implemented into the Safety program as VAR_EXTERNAL.

- The function block "`NetVarSender`" controls sending of the variable list.
- The function block "`NetVarReceiver`" establishes a connection to the network variable sender list. The connection is monitored by using the watchdog time.

Both function blocks have to be created on the respective Safety control.

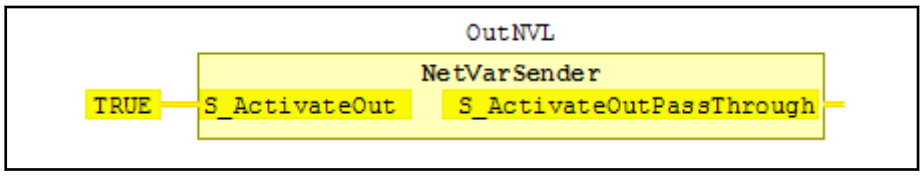

Fig. 5-42: Example of an interconnection of a “NetVarSender” function block

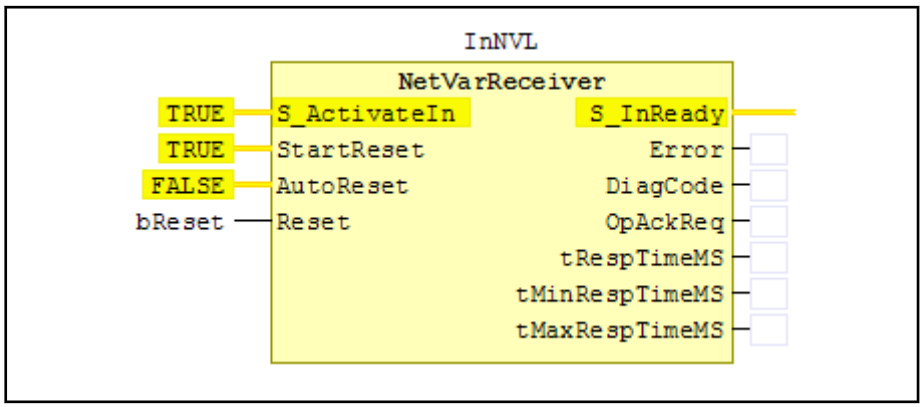

Fig. 5-43: Example of an interconnection of a “NetVarReceiver” function block

5.7 Programming example

The SafeLogic programming example for a program in the "Extended Level" is the **Two-Hand Control with EDM** of the document "PLCopen - Technical Committee 5 Safety Software Technical Specification Part 2: User Examples Version 1.01 - Official Release". It is shown as SafeLogic FBD implementation.

☞ It is programmed in the "Extended Level" to assign diagnostic information from the "DiagCode" outputs of the Safety PLCopen function blocks to WORD variables. This is not possible in the current implementation on "Basic Level".

Describing the safety functions

The following safety functions are used in this example:

- When pressing the E-STOP button, all dangerous motions have to be stopped (via SF_EmergencyStop).

  E-STOP has the highest priority.

  After releasing the E-STOP button, reset via **Reset**
- The safety output is enabled by pressing both the pushbuttons of the two-hand control.

Releasing one of the two-hand pushbuttons disables the safety output and stops the dangerous motion via the switching devices K1 and K2 (via SF_TwoHandControlTypeII)

- The initial state and the operating state of the connected switching devices are monitored. If an error is detected, the safety output cannot become ready for operation (via SF_EDM)
- After activating the safety application or the functional application or after an E-STOP condition, the two-hand control has to be released and pressed again to enable the safety output again (via SF_OutControl).

  To ensure this for the functional restart, the process signal of the functional application is connected to the "Activate" input of the two-hand control function block "THC_S2_S3".

  If the application process is started again while the two-hand control is enabled, the function block goes into the state 16#C003.

  This status signals an error in which both buttons are pressed during activation, thus preventing a restart.

Only one operating state exists in this example.

EDM_PRG

In Work

PROGRAM EDM_PRG (* Extended Level *)

Line	Scope	Name	Type	Initial value	Comment
1	VAR	EStop_S1	SF_EmergencyStop		SF_EmergencyStop function block instance
2	VAR	S1_S_EStopIn	SAFEBOOL	FALSE	E-Stop S1
3	VAR_EXTERNAL	Reset	BOOL		Reset comming from standard control
4	VAR	S_EStopOut	SAFEBOOL	FALSE	
5	VAR	Error_EStop_S1	BOOL	FALSE	EStop_S1 error flag
6	VAR	Diag_EStop_S1	WORD	0	Diagnostic code for EStop_S1: Regular operation,16#8xxx, in case of an error in EStop_S1: 16#Cxxx
7	VAR_EXTERNAL	Process	BOOL		Enabling the motion by the standard control process
8	VAR	S2_S_Switch1	SAFEBOOL	FALSE	Switch S2 connected to push button 1 of two-hand control
9	VAR	S3_S_Switch2	SAFEBOOL	FALSE	Switch S3 connected to push button 2 of two-hand control
10	VAR	S_TwoHandOut	SAFEBOOL	FALSE	
11	VAR	THC_S2_S3	SF_TwoHandControlTypeII		SF_TwoHandControlTypeII function block instance
12	VAR	Error_THC_S2_S3	BOOL	FALSE	TCH_S2_S3 error flag
13	VAR	Diag_THC_S2_S3	WORD	0	Diagnostic code for THC_S2_S3: Regular operation,16#8xxx, in case of an error in THC_S2_S3: 16#Cxxx
14	VAR	OC_K1_K2	SF_OutControl		FBs SF_OutControl function block instance
15	VAR	EDM_K1_K2	SF_EDM		SF_EDM function block instance
16	VAR	Error_OC_K1_K2	BOOL	FALSE	OC_K1_K2 error flag
17	VAR	Diag_OC_K1_K2	WORD	0	Diagnostic code for OC_K1_K2: Regular operation,16#8xxx, in case of an error in OC_K1_K2: 16#Cxxx
18	VAR	K1_S_EDM1	SAFEBOOL	FALSE	external device K1 feedback
19	VAR	K2_S_EDM2	SAFEBOOL	FALSE	external device K2 feedback
20	VAR	Error_EDM_K1_K2	BOOL	FALSE	EDM_K1_K2 error flag
21	VAR	Diag_EDM_K1_K2	WORD	0	Diagnostic code for EDM_K1_K2: Regular operation,16#8xxx, in case of an error in EDM_K1_K2: 16#Cxxx
22	VAR	S_EDM_Out_EDM_K1_K2	SAFEBOOL	FALSE	Enabling signal for safety control (safe drives...)
23	VAR_EXTERNAL	S_EDM_Out_EDM_K1_K2_unsafe	BOOL		Enabling signal for standard control (unsafe)

Fig. 5-44: Declaration for programming example: two-hand control with EDM

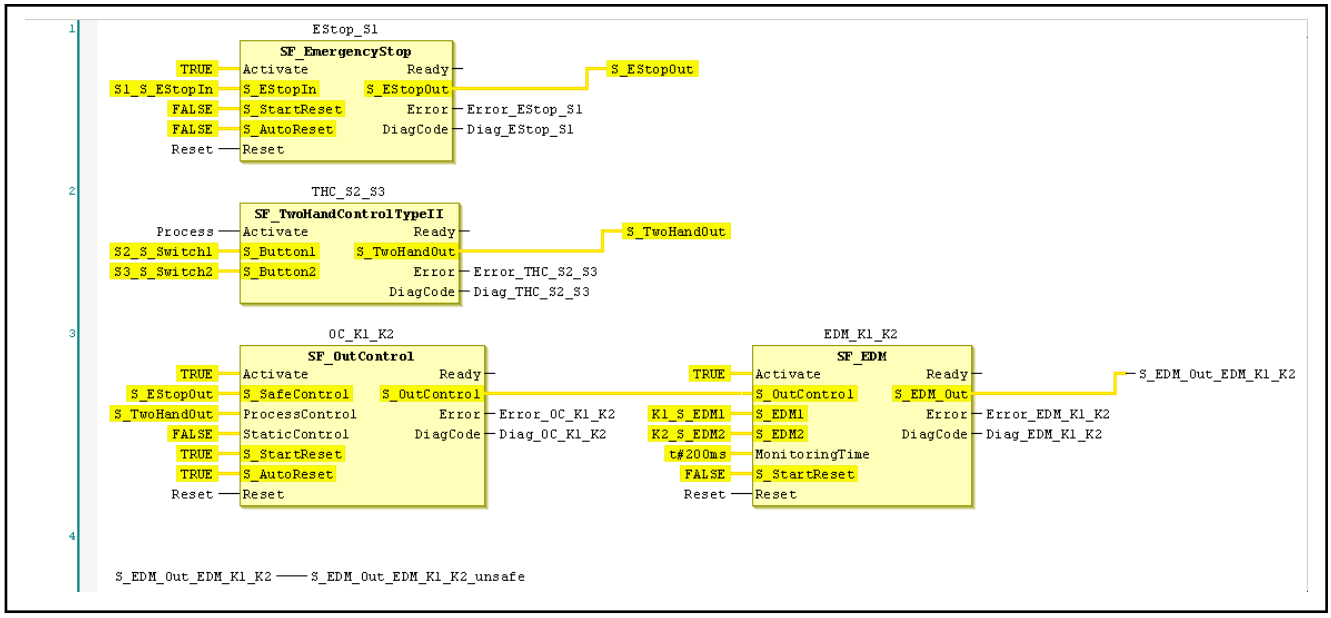

Fig. 5-45: Implementation on the example: Two-hand check with EDM

Name	Data type	Description
S1_S_EStopIn	SAFEBOOL	E-STOP button S1
S2_S_Switch1	SAFEBOOL	Switch S2 connected to pushbutton 1 of the two-hand control
S3_S_Switch2	SAFEBOOL	Switch S3 connected to pushbutton 2 of the two-hand control
K1_S_EDM1	SAFEBOOL	Feedback of external device K1
K2_S_EDM2	SAFEBOOL	Feedback of external device K2
Reset	BOOL	Reset of standard control
Process	BOOL	Enabling the motion by the process of the standard control

Tab. 5-2: Function block inputs

Name	Data type	Description
S_EDM_Out_EDM_K1_K2	SAFEBOOL	Enabling signal for the safety control (safe drives...)
S_EDM_Out_EDM_K1_K2_unsafe	BOOL	Enabling signal for the standard control (not safe)
Error_EStop_S1	BOOL	Error flag of EStop_S1
Error_THC_S2_S3	BOOL	Error flag of TCH_S2_S3
Error_OC_K1_K2	BOOL	Error flag of OC_K1_K2
Error_EDM_K1_K2	BOOL	Error flag of EDM_K1_K2
Diag_EStop_S1	WORD	Diagnostic code for EStop_S1 16#8xxx: Regular operation 16#Cxxx if error in EStop_S1
Diag_THC_S2_S3	WORD	Diagnostic code for THC_S2_S3 16#8xxx: Regular operation 16#Cxxx if error in THC_S2_S3

Name	Data type	Description
Diag_OC_K1_K2	WORD	Diagnostic code for OC_K1_K2 16#8xxx: Regular operation 16#Cxxx if error in OC_K1_K2
Diag_EDM_K1_K2	WORD	Diagnostic code for EDM_K1_K2 16#8xxx: Regular operation 16#Cxxx if error in EDM_K1_K2

Tab. 5-3: Function block outputs

Additional information

The example can also be used with SF_TwoHandControlTypeIII.

For simplicity reasons, the "Activate" input was set to TRUE. In the application, this can be replaced by a variable.

Function block	Input	Constant value	Description
EStop_S1	S_StartReset	FALSE	No automatic reset if the S-PLC is started
	S_AutoReset	FALSE	No automatic reset. Reset/acknowledgement required by operator
OC_K1_K2	S_StartReset	TRUE	Automatic reset permitted if the S-PLC is started
	S_AutoReset	TRUE	Automatic reset. No reset/acknowledgement required by operator
	Static Control	FALSE	A dynamic modification of the "Appl_Control" signal (rising edge) is requested after enabling the function block or a triggered safety function (S_SafeControl to FALSE)
EDM_K1_K2	S_StartReset	FALSE	No automatic reset if the S-PLC is started
	MonitoringTime	T#200 ms	The maximum response time of both feedback signals S_EDM1 and S_EDM2

Tab. 5-4: Information on the function block parameters used

5.8 Loading and starting projects on the control

1. Logging into the application on the standard control

   By logging in, the PLC program is loaded to the control. The data exchange to the safe control is still established.

2. Loading the Motion configuration from PC to control

   Create drives using the function "Synchronize -> Download Motion configuration to the device" in the context menu of the control. The Sercos bus has to be in P4 or BB.

3. Logging into "SafeApp" on the safety control

   When logging into the safety control, the Safety program is loaded to the RAM of the safety control.

   If a new project is loaded to the safety control, press the button on the Safety function module. It is prompted for the control password at each login, even if no password is set.

   If the Safety program is to remain on the safety control after logout, apply the boot project.

   Enable the "SafetyApp" via **Set active application** and generate the boot project via the menu item **Debug ▸ Generate boot application**.

4. Starting the PLC (standard application and safety application)

   Both controls are started.

   The PROFIsafe and the CSos communication start correctly. All yellow Inline couplers should display green in the status LEDs "D" and "UM". The "ProfisafeHost" and "CSosOriginator" function blocks do not display any errors and the "FV_activated_S" output is FALSE.
   The data exchange between standard control and safety control is active.

5.9 Modifying projects

The application on the safety control can only be modified if logged off from the safety control.

With each change at the Safety I/Os (PROFIsafe, CSos, data exchange devices), the program on the standard control as well as the program on the safety control have to be refreshed (login/download).

6 Troubleshooting

This chapter supports the troubleshooting in Safety projects. It lists all typical problems and their solutions.

6.1 SafeLogic interface

- **Problem:** When trying to download the SafeLogic program, the following message is displayed:

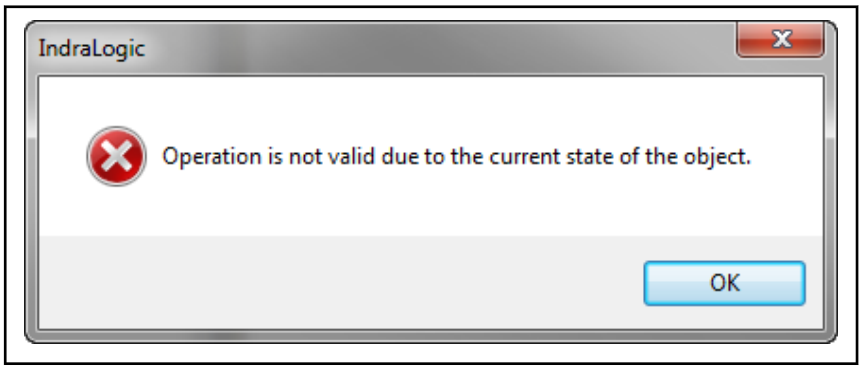

Fig. 6-1: SafeLogic compilation error

Cause: SafeLogic program contains syntactic errors.

Solution: Check the error output (in the message window) and debug the error in the SafeLogic program.

- **Problem:** When trying to log in to the Safety function module, the following message is displayed:

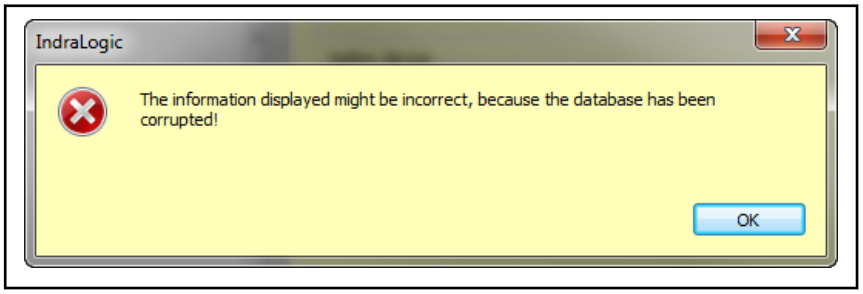

Fig. 6-2: SafeLogic connection error

Cause: Either the firmware of the Safety function module or the firmware of the standard control are outdated or there is no Safety module on the control.

Solution: Check firmware versions and update if required. For the standard control, the firmware version can be updated via the connection test in the "Properties" dialog of the control. For the Safety function module, the firmware version can be checked in the "Status" dialog of the Safety function module (double-click on the Safety function module, "Status" tab).

- **Problem:** While downloading the PLC program to the standard control, a message is displayed informing on the different config. IDs.

  Cause: The config ID of the standard control is not equal to the one of the Safety control (see fig. 7-1 "Status tab" on page 68)

  Solution: Reload the standard control and the Safety control.

6.2 SafeLogic function module

- **Problem:** After starting up the system, the FS-LED flashes red at the Safety function module. The Safety application is not running.

  Cause: There is no Safety boot application. Thus, no Safety application could be loaded as well.

  Solution: Store a boot application on the Safety function module. Go to the menu item **Create ▸ Generate boot application**.

- **Problem:** Three LEDs are red at the Safety function module. The Safety application is not running.

  Cause: The Safety control is in the safe state due to a fatal error.

  Solution: Restart the control and analyze the log of the safety control (see fig. 7-2 "Safety control logbook" on page 68).

6.3 PROFIsafe

- **Problem:** Individual PROFIsafe devices (Safety Inline terminals at the Profibus coupler) loose their connection sporadically. The "Profisafe-Host" function block of the corresponding devices displays the error code C104.

  Cause: Profibus is not running stable. Incorrectly set terminal resistances are frequently the cause.

  Solution: For Profibus, the terminal resistances are only to be activated at cable ends. All other terminal resistances have to be disabled. Otherwise, electrical waves reflect. This corrupts data.

- **Problem:** Individual PROFIsafe devices (Safety Inline terminals at the Profibus coupler) do not establish any connection. The "ProfisafeHost" function block of the corresponding devices displays the error code C102.

  Cause: Data is not exchanged between the Safety control and the PROFIsafe devices.

  Solution:

  – Ensure that the IOConfigIDs are identical on the standard control and on the safety control. Use `SafetyData.SafetyFM.State.IOConfigIDState`.

  – Use the global data structure “SafetyData” to check whether the inputs and the outputs of the respective PROFIsafe devices are transmitted.

  – Check whether the respective device was disabled via "UserDefs.cfg".

  – Check the I/O addresses of the Profibus/PROFINET device.

  – Ensure that the standard PLC does not have write access on the I/O addresses of the Safety device.

- **Problem:** The Profibus diagnostics of the coupler does not contain the slot number (slot of the Inline terminal).

  Cause: The IndraWorks Profibus diagnostics identifies only the Profibus coupler as device.

  Solution: The slot number (slot of the Inline terminal) is included in the "Extended diagnostics" (refer to the following figure). The slot can be counted. Position the cursor over the diagnostic text and a tooltip with additional information is provided if available.

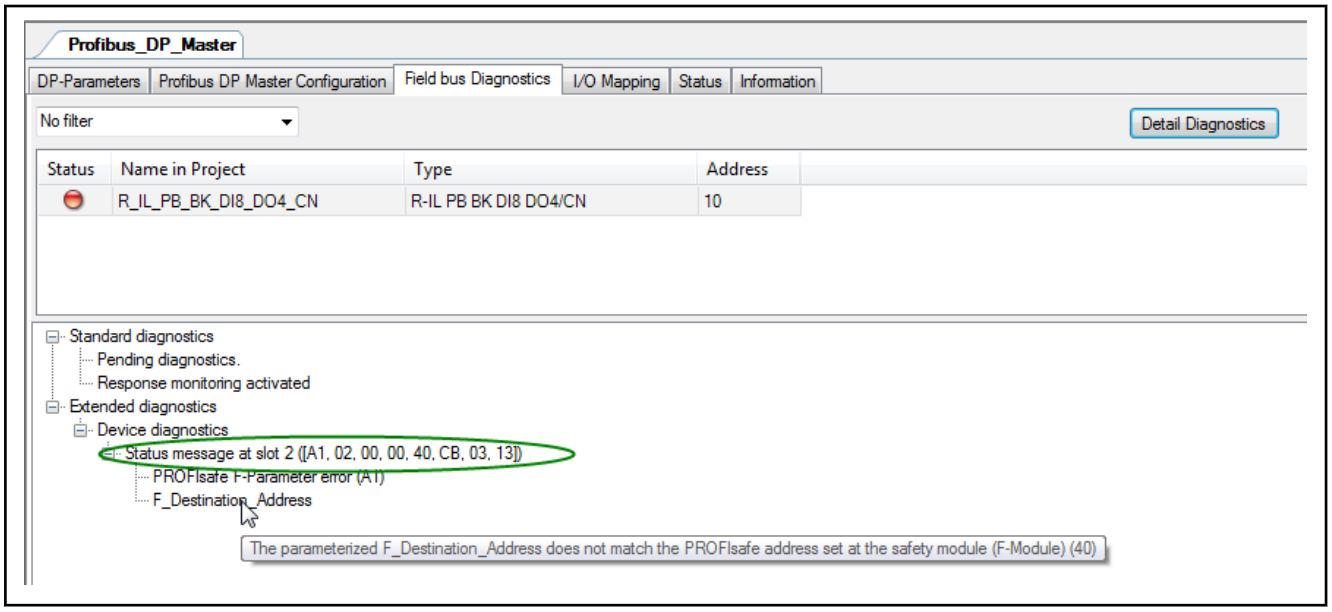

Fig. 6-3: Slot number of the extended Profibus diagnostics and additional information as tooltip

- **Problem:** UT1 and/or UT2 flash(es) at runtime.

  Cause: There is a problem with wiring. This problem can be caused as follows:

  1. The supply outputs are short-circuited against another cable.
  2. A test pulse clock is selected for the input or for the output. However, the connected device generates its own test pulse clock.
  3. The cables for the test pulses were interchanged with other inputs or outputs.

  Solution: Check wiring. Disable test pulses if required.

6.4 CIP Safety on Sercos

- **Problem:** If projects are applied from 13V08 or 14V02, drives with CSos do not longer operate. Projects can also not be edited correctly anymore.

  Cause: In 13V14 and 14V04, the logic CSos devices (below the "Logic I/Os" node) of the drives were changed. Thus, the logic CSos devices do not match the CSos devices in the project anymore.

  Solution: Create the logic devices of the drives again. Proceed as follows for each drive:

  1. Cancel the CSos connection: Open the context menu of the drive in the Project Explorer under **Sercos ▸ Drive name ▸ Drive** and select "CSos settings".

     Disable CSos and close the dialog.
  2. Rename the logic I/Os belonging to the drive below the "Logical devices" node. It is recommended to attach "_old".
  3. Create a new CSos connection: Open the CSos settings again and enable CSos for the drive.

     A new logic device for the drive is created below the "Logic I/Os" node of the safety control.

4. All settings and variables from the old logic device are applied to the new device.
5. Delete the old logic device.

- **Problem:** The CSos connections are interrupted again and again after timeouts.

  Cause: A cyclic data timeout is exceeded, i.e. insufficient number of produced data packages.

  Solution: Increase the configuration value "NTE" of 4*Safety cycle time in the CSos settings for the "safe communication".

6.5 Safety PLCopen function blocks

- **Problem:** The Safety PLCopen function block "SF_GuardMonitoring" remains in the "State 8002" after the start and cannot be reset via the "Reset" input.

  Cause: The inputs "S_GuardSwitch1" and "S_GuardSwitch2" are directly connected to a PROFIsafe Inline terminal. When starting the application, these inputs are sporadically FALSE for one cycle. Due to the state machine defined in the PLCopen, the function block remains in this state.

  Solution: The "Activate" input of the "SF_GuardMonitoring" function block may only be enabled after the PROFIsafe data exchange has started. OR operate the "FV_activated_S" outputs of the "ProfisafeHost" function block instances and use them as trigger for the "Activate" input of the PLCopen function blocks.

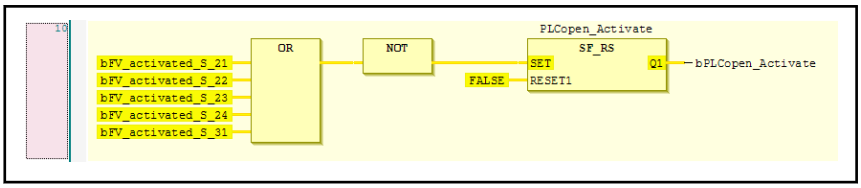

Fig. 6-4: OR operation of the "FV_activated_S" outputs

6.6 Converting variables (safe ⇔ not safe)

- **Problem:** A non-safe variable is to be converted into a safe variable.

  Solution: Conversion via an AND operation.

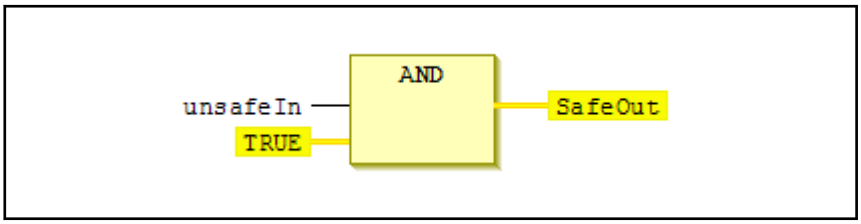

Fig. 6-5: Converting a non-safe variable into a safe variable

☞ A safe variable may only be converted into a non-safe variable for test purposes. For real machines, further safety actions are required.

- **Problem:** A safe variable is to be applied to the "grey" control via VAR_EXTERNAL.

  Solution: Assign a safe variable to a non-safe variable.

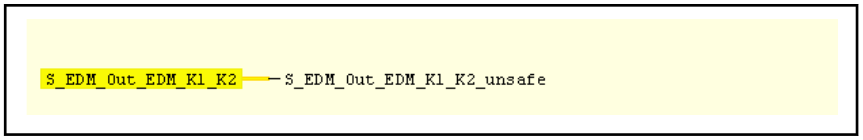

Fig. 6-6: Converting a safe variable into a non-safe variable

7 Diagnostics

7.1 Overview

IndraWorks:

- Safety Function module - Status (see chapter 7.2 "Status of the Safety function module" on page 67)
- Safety Function module - Logbook (see chapter 7.3 "Log of the Safety function module" on page 68)
- Standard control - Log (refer to chapter 7.4 "Log of the standard control" on page 69)
- Standard control - Device diagnostics (refer to chapter 7.5 "Device diagnostics of the standard control" on page 70)
- SafeLogic - Function blocks (refer to chapter 7.6 "Diagnostic outputs of the ProfisafeHost and CSosOriginator function blocks" on page 71)
- Diagnostics on the standard PLC control, refer to chapter 7.7 "Diagnostics on the standard PLC control" on page 71
- Sercos I/O diagnostics
- **Sercos ▸ HCS02.x ▸ Parameter editor**
- **Sercos ▸ HCS02.x ▸ Axis_x ▸ Error/diagnostic memory**
- MTX debug menu (show functions)

7.2 Status of the Safety function module

For the status of the Safety function module, go to the Project Explorer, double-click on the function module and select the "Status" tab.

No Safety license is required for this display. The Safety connection does not have to be confirmed via the button of the safety control.

To obtain the current module status, press the **Refresh** button.

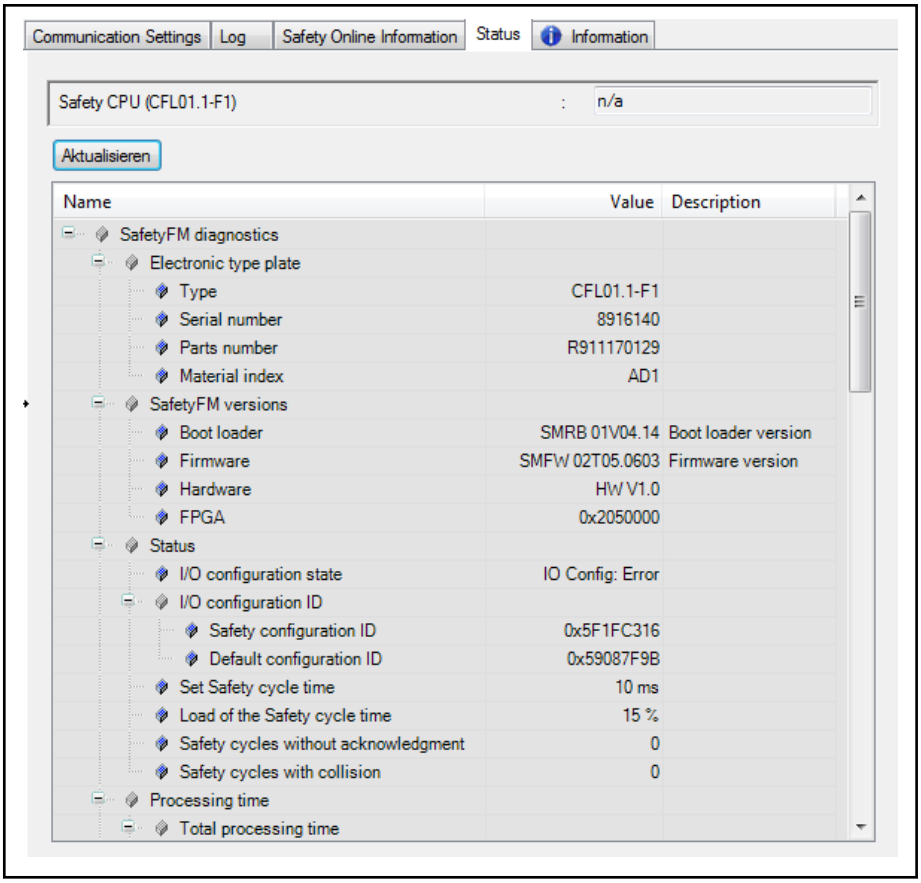

Fig. 7-1: "Status" tab

7.3 Log of the Safety function module

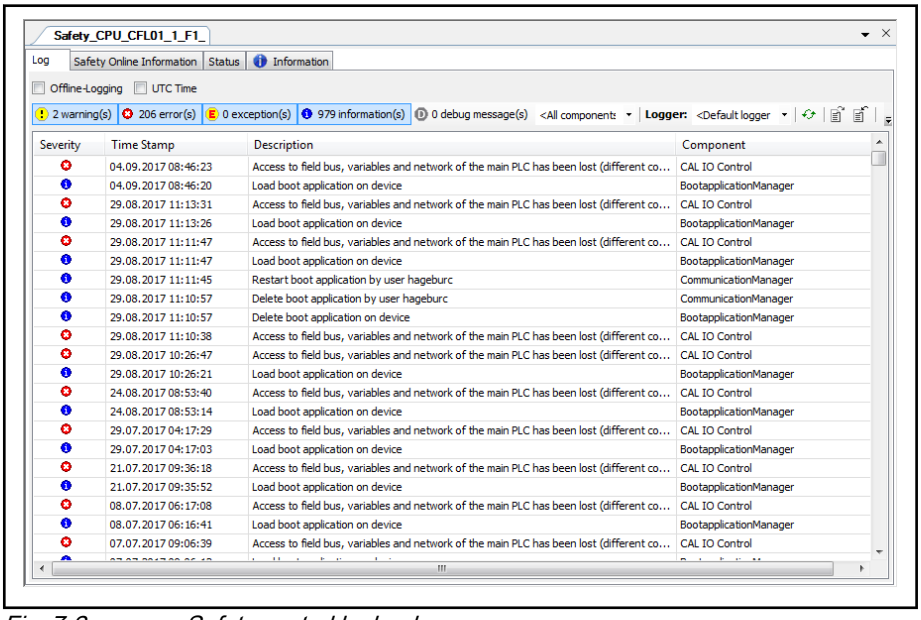

Fig. 7-2: Safety control logbook

The license "IndraWorks SafetyManager Lite" is at least required for this display. This connection is possible via "remote access".

The logbook is displayed in the **Log** tab of the Safety control and is used as protocol and to diagnose runtime errors of the application and system errors. The log can be used to find the error cause in the PLC or in the application.

Two logs are available in SafeLogic:

- Device logbook
- Application log

The device log is part of the device and is intended for entries concerning the device, such as system errors or generating new boot applications.

The application log is part of the IEC application and is intended for entries concerning the application, such as runtime errors, errors while loading the boot application and errors during online communication.

The application log is the default log of the Safety control. The device log is only available if the default log (application log) has been loaded.

Press the  button to load all available logs (device log and application log) cyclically from the control. They can then be subsequently selected in the **Logger** window.

The information shown in the logs is structured as follows:

- Weighting (information, warning, error, exception)
- Time stamp
- Error description
- Component (generator)

7.4 Log of the standard control

For example, the logbook of the standard control logs problems occurring when trying to establish a connection and during the data transfer from or to safe bus devices. To go to the log, double-click on the standard control and select the **Log** tab.

The log of the standard PLC is in the RAM of the control. The log is not lost when switching off and on the control.

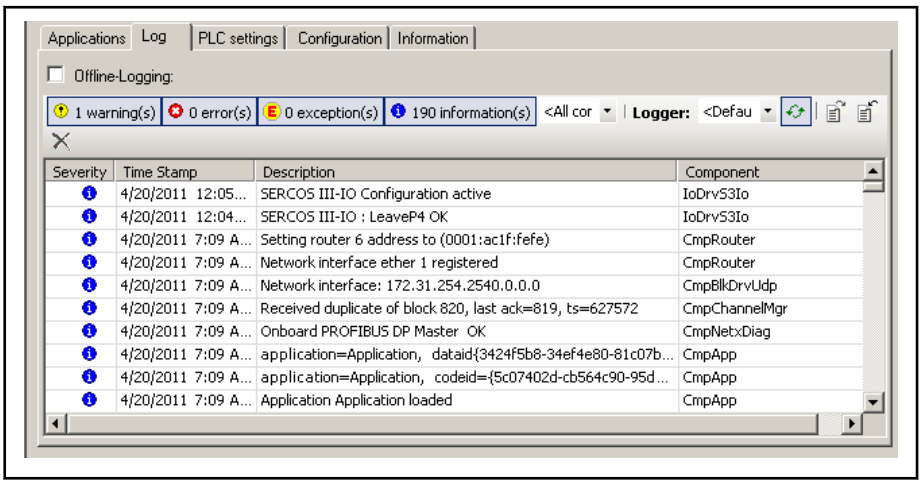

Fig. 7-3: Device editor – Log

To enable the log, go to **Tools** ▸ **Options...** and subsequently to **IL2G** ▸ **General settings**.

7.5 Device diagnostics of the standard control

MLC system

The "Error/diagnostic memory" dialog provides an overview on all diagnostic messages of the control and the connected devices. The diagnostic messages are sorted automatically. The latest messages are on top of the list. The messages of the standard PLC are also stored in a shortened version.

The log entries are remanently stored on the control and are thus still available after switching off and on the control. The memory of the device diagnostics is limited to 100,000 entries.

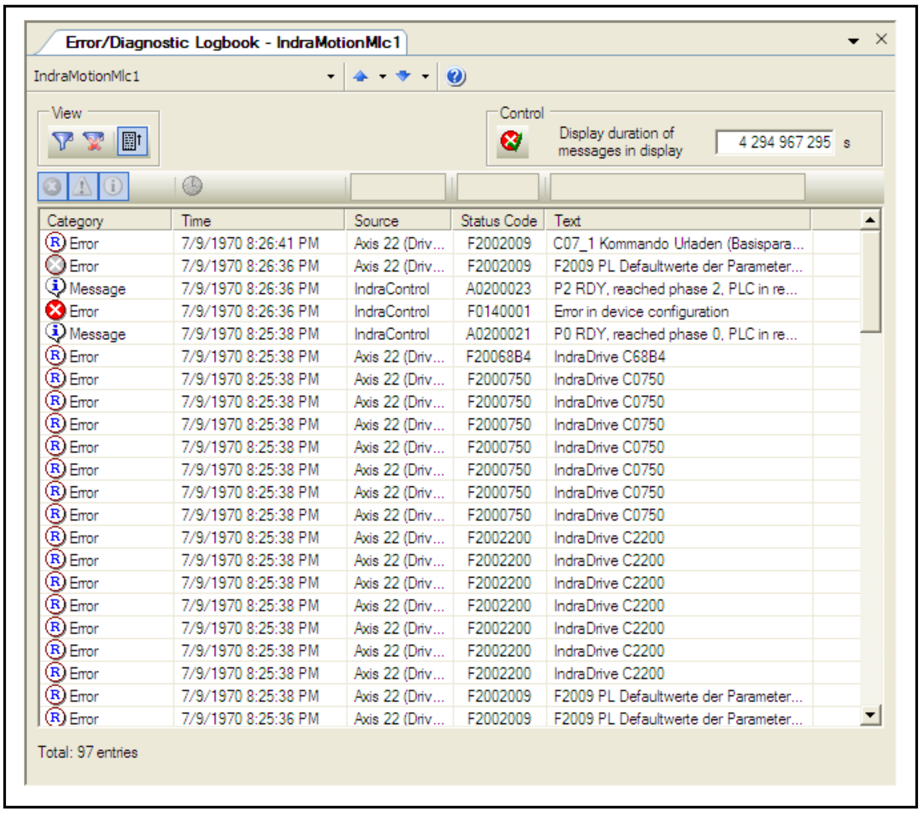

Fig. 7-4: Example of an MLC diagnostic logbook representation

This dialog can be opened online and offline. Some functions are disabled in offline mode.

MTX system

The "Device diagnostics" dialog shows messages, warnings and errors generated by the MTX system.

These messages can be applied to a log. The data is persistently stored in a database on the PC. Multiple logs can be created with 1,000 entries each.

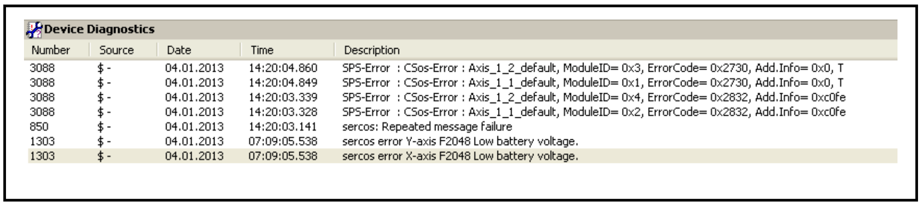

Fig. 7-5: Example of the MTX device diagnostics

7.6 Diagnostic outputs of the ProfisafeHost and CSosOriginator function blocks

To monitor the connection to safe field bus devices, the function blocks "ProfisafeHost" and "CSosOriginator" are provided. Both function blocks provide the "DiagCode" output. The "CSosOriginator" additionally provides the "DiagCodeExt" output. If there is a connection problem to the save field bus devices, these outputs point directly towards the error cause.

The license "IndraWorks SafetyManager Lite" is at least required for this display. This connection is possible via "remote access".

For the error codes, refer to the function blocks in the online helps.

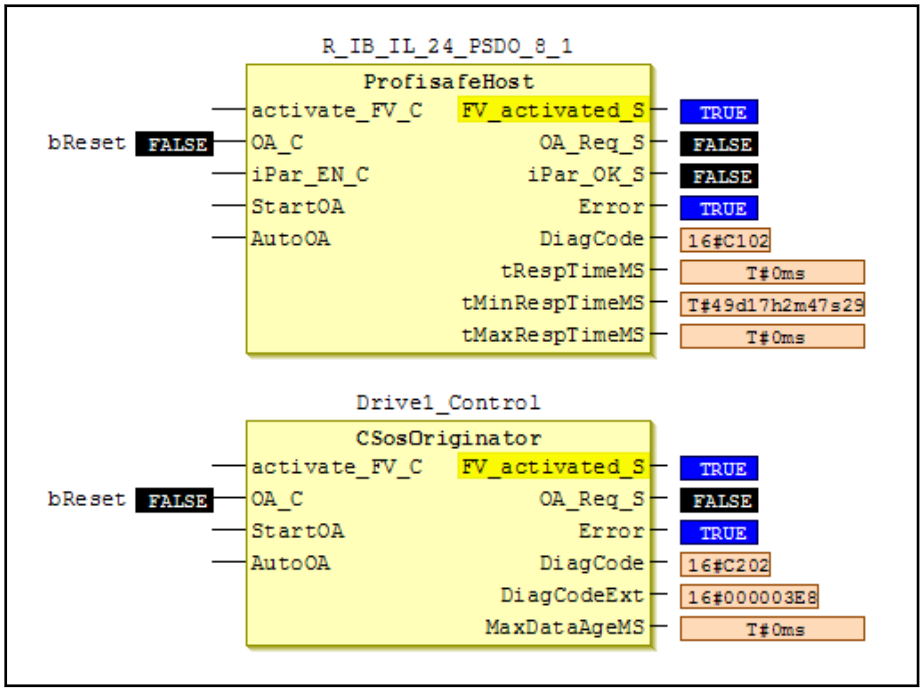

Fig. 7-6: Error information at the ProfisafeHost and CSosOriginator function blocks

7.7 Diagnostics on the standard PLC control

For many applications, it is required that the standard control is provided with a read-only access to the safe I/Os of the safety control. Additionally, more status information is required from the Safety control to operate the machine.

This information is provided via the global variable `SafetyData` on the standard PLC. This variable design is structured. It is automatically created and updated when compiling the standard PLC application.

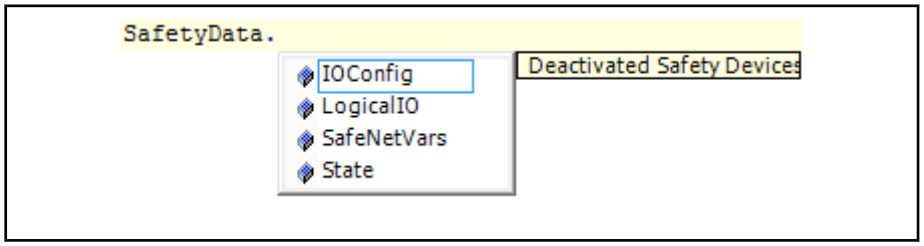

Fig. 7-7: Elements of the global variable “SafetyData”

The global variable `SafetyData` allows to access the following diagnostic information:

- `SafetyFM` - Status information of the Safety control:

  This information is mostly identical to the information given in the "status of the function module" (chapter 7.2 "Status of the Safety function module" on page 67)

- `LogicalIO` - Access to safe connections:

  This entry facilitates access to safe connections. It hierarchically maps the devices below the "Logic I/Os" node and the Safety control. Connection information is also available in addition to the access to the transmitted variables

- `SafeNetVars` - Access to safe network variables:

  This entry provides all safe network variable lists known to the control. Each configured connection is individually visible in case of sender lists. The connection status is displayed in addition to the access to the transmitted safe variables

- `IoConfig` - Disables safe devices:

  This entry maps the functionality to teach disabled safe devices. Apart from the checksum (CRC) of the current configuration, the access to deactivation bits is also possible

Information mapped in the global variable “SafetyData” is suitable for the following areas of application.

- Check the connection status of the safe devices
- Electrical commissioning of safe I/Os
- Status and diagnostic information for HMI devices
- Use safe signals to control the standard PLC program
- Access to diagnostic data without logging into the safety control

The global variable `SafetyData` does not generate any additional computing load in the standard PLC. Only when an entry is accessed, required data is provided and causes computational load. Entries that are not accessed, do not generate any computing load.

Entries below the global variable “SafetyData” are provided with the following properties:

- All entries are always read-only
- The information read has to be considered as unsafe variables
- Information is also updated if the standard PLC is in the "Stop" state
- Data can be recorded via the PLC trace
- The signals are sampled at the access point in time of the standard PLC program or the monitoring in the debugger. Thus, they do not have to be synchronous

Example of accessing input data of a safe I/O module:

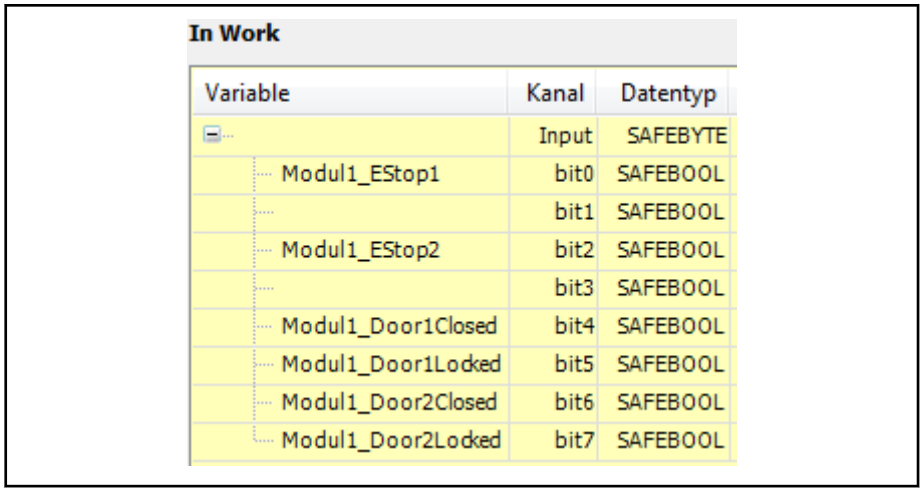

Fig. 7-8: Variable definition of a logic I/O (in this example" Modul1In")

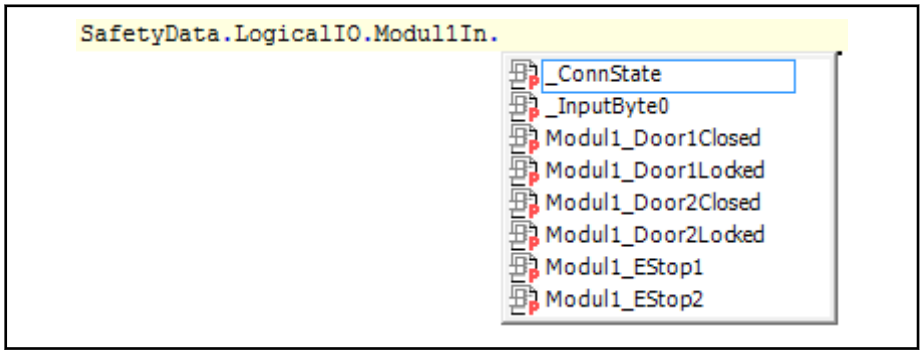

Fig. 7-9: Resulting image under SafetyData.LogicalIO.Modul1In

8 Style guide Safety programming

8.1 Introduction

Programming under SafeLogic differs from programming the standard control with regard to new data types, graphic editors, a changed color representation of safe variables and the option to verify program parts.

Data can be exchanged with the safe I/Os and the standard control.

For the Safety programming, the standard programming guidelines are always applied. Special guidelines for Safety are noted down in the following. They are based on the PLCopen Safety.

A uniform approach significantly facilitates the readability of the program code when specifying function block names, types, variables, etc. Thus, others become quicker familiar with the code and troubleshooting can be reduced.

Reference documents

- IEC 61131-3
- PLCopen - Technical Committee 5: Safety Software

  Part 1: Concepts and Function Blocks Version 1.0 - Official Release (2007-01-31)

  Part 2: User Examples Version 0.9 - Working Document (2007-06-14)
- CoDeSys Safety - Functional specification programming system

  Part 2: Language scope - Version 1.0 (2007-03-06)

8.2 Application structure

The Safety application has to be structured according to safety-relevant function groups. Status and error messages are to be output individually for each function group.

Basic Level

The Basic Level is suitable to interconnect I/Os and to create simple logic operations.

Only FBD is permitted as IEC language.

Additionally, the number of usable function blocks is strictly limited. Only AND and OR as well as certified function blocks are permitted.

Safety applications are to be created simply by instantiating and interconnecting existing function blocks. Thus, they can also be easily connected.

Extended Level

The Extended Level allows calling complex function blocks.

Only FBD is permitted as IEC language.

Certifying the application is more demanding due to the bigger instruction set.

Note further restrictions on the Safety programming. A POU in "Safety FBD" may not exceed 50 networks in the default setting for example. This value can be set via the properties of the "SafetyApp" node.

8.3 Variable names

- Label safe variables with the prefix "S_". This identifies variables easily. The graphic support (yellow marking) is no safety function. Continue using the standard prefixes for data types, such as "S_bSafetyActive"
- Label mapped variables (e.g. to PROFIsafe, Csos or logic exchange devices) either with the suffix "_IN" or "_OUT"

The direction is defined with reference to the safety control:

– IN: to the safety control (input)
– OUT: from the safety control (output)

9 Service and support

Our worldwide service network provides an optimized and efficient support. Our experts offer you advice and assistance should you have any queries. You can contact us **24/7**.

Service Germany

Our technology-oriented Competence Center in Lohr, Germany, is responsible for all your service-related queries for electric drive and controls.

Contact the **Service Hotline** and **Service Helpdesk** under:

Phone:	**+49 9352 40 5060**
Fax:	**+49 9352 18 4941**
E-mail:	service.svc@boschrexroth.de
Internet:	http://www.boschrexroth.com

Additional information on service, repair (e.g. delivery addresses) and training can be found on our internet sites.

Service worldwide

Outside Germany, please contact your local service office first. For hotline numbers, refer to the sales office addresses on the internet.

Preparing information

To be able to help you more quickly and efficiently, please have the following information ready:

- Detailed description of malfunction and circumstances
- Type plate specifications of the affected products, in particular type codes and serial numbers
- Your contact data (phone and fax number as well as your e-mail address)

Index

A
Abbreviations ... 9
ANSI Z535.6-2006 ... 8

C
Commissioning ... 31
  Create CSos S20 terminals ... 43
  Create IndraWorks project ... 31
  Create PROFIsafe Inline terminals at the Profibus ... 32
  Create PROFIsafe Inline terminals at the Profinet ... 37
  Exchange variables ... 50
  IndraDrive with CSos ... 46
  Load and start project ... 59
  Modify project ... 60
  Safety network variables ... 51
Complaints ... 9
Configuring hardware I/Os ... 15
  CSos ... 26
  PROFIsafe ... 15
Criticism ... 9
Customer Feedback ... 9

D
Diagnostics ... 67
  Device diagnostics of the standard control ... 70
  Log of the Safety function module ... 68
  Log of the standard control ... 69
  SafeLogic function blocks ... 71
  Standard control ... 71
  Status of the Safety function module ... 67
Documentation
  Change record ... 1
Documentation overview ... 1
Documentation structure ... 2
Documentation, overview ... 1
Documentation, purpose ... 1
Documentation, structure ... 2
Documentation, validity ... 1

F
Feedback ... 9

H
Hazard warnings ... 8
Helpdesk ... 77
Hotline ... 77

I
Installation ... 13
  Device description files ... 13
  Install the SafetyManager license ... 13
  System prerequisites ... 13

N
Names ... 9
Names and abbreviations ... 9

P
Product phases ... 1
Programming example
  Safety, two-hand control with EDM function block ... 55
Project goal First Steps - Safety ... 11
Purpose of the documentation ... 1

S
Safe network variables
  NVL ... 51
Safety
  Safe variables ... 51
Safety instructions ... 8
Service hotline ... 77
Signal alert symbol ... 8
Signal words ... 8
Style guide application structure ... 75
Style guide Safety programming ... 75
Style guide variable names ... 75
Suggestions ... 9
Support ... 77
Symbols ... 9

T
Target groups ... 1
Troubleshooting ... 61
  CIP Safety on Sercos ... 63
  Convert variables ... 64
  PROFIsafe ... 62
  SafeLogic function module ... 62
  SafeLogic interface ... 61
  Safety PLCopen function blocks ... 64

V
Validity of the documentation ... 1

W
Warnings ... 8

Notes

Notes

rexroth
A Bosch Company

Bosch Rexroth AG
P.O. Box 13 57
97803 Lohr a.Main, Germany
Bgm.-Dr.-Nebel-Str. 2
97816 Lohr a.Main, Germany
Phone +49 9352 18 0
Fax +49 9352 18 8400
www.boschrexroth.com/electrics

R911400170

DOK-IWORKS-SL*STEP*V15-CO02-EN-P