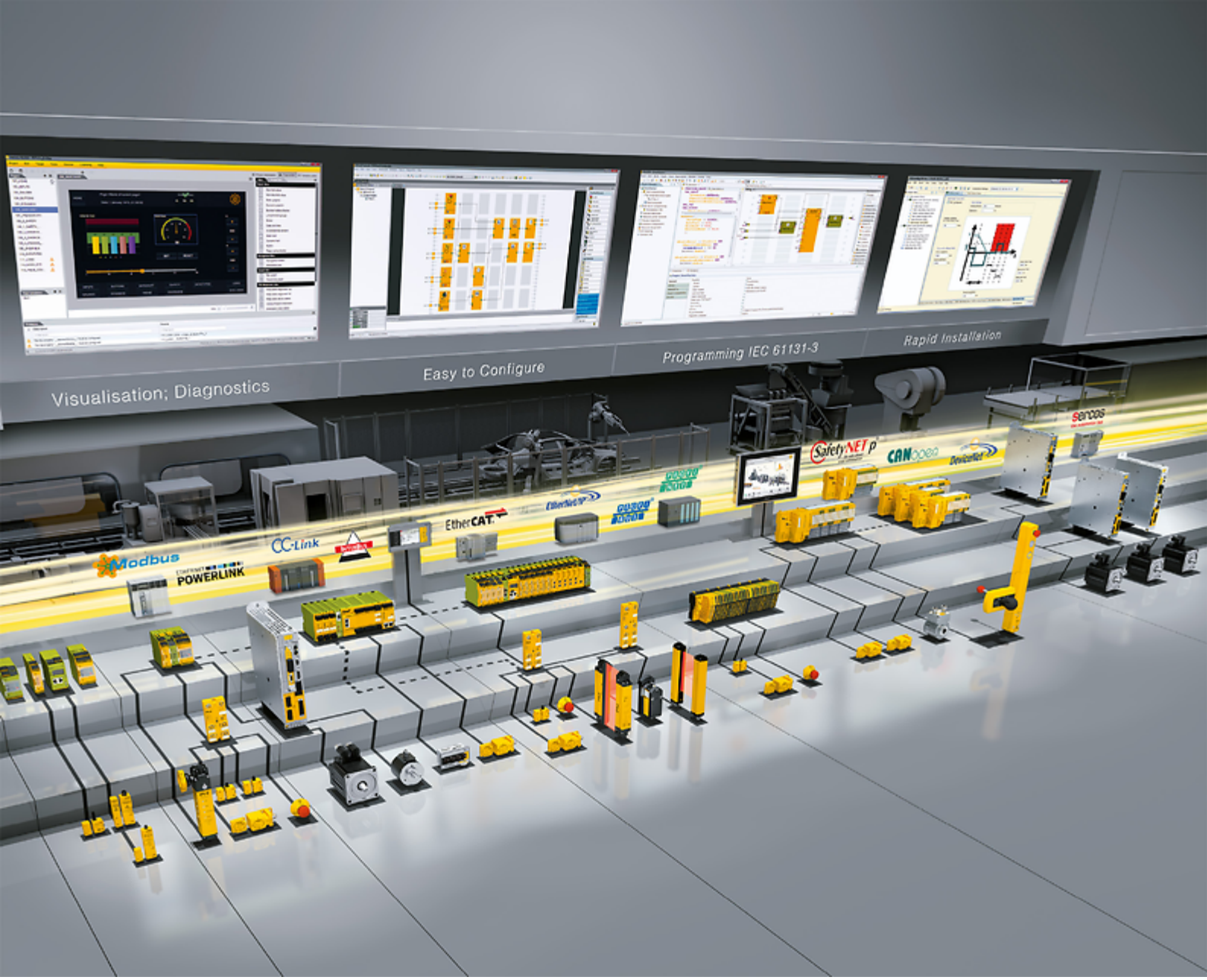


Safety Device Diagnostics

PILZ
THE SPIRIT OF SAFETY

- PSEN sensor technology

System Description-1003827-EN-08

1 Introduction 5
1.1 Definition of symbols 5

2 Overview 6
2.1 Safety Device Diagnostics 6
2.2 Communication with fieldbus 9

3 Safety 10
3.1 Intended use 10
3.2 Safety regulations 10
3.2.1 Additional documents that apply 10
3.2.2 Use of qualified personnel 10
3.2.3 Warranty and liability 11
3.2.4 Disposal 11

4 System structure 12
4.1 Structure of a system with SDD - example 12
4.2 Construction of safety circuits with the SDD 15
4.3 Examples for the connection of further devices 18
4.4 Wiring 19

5 List of data 23
5.1 Data types 23
5.2 Overview 24
5.2.1 Safety devices of all types 24
5.2.2 Table of events and status information per safety device 25
5.3 Writeable data (outputs) 26

6 Fieldbus Modules 27
6.1 SDD ES ETH 27
6.1.1 Modbus/TCP 27
6.1.1.1 General 27
6.1.1.2 Default settings that cannot be modified 28
6.1.1.3 Default settings that can be changed 28
6.1.1.4 Recommended settings of the client 28
6.1.1.5 Function codes 28
6.1.2 Input data 29
6.1.3 Output data 34
6.2 SDD ES PROFIBUS 35
6.2.1 Restriction when connecting PSENmlock 35
6.2.2 GSD file 35
6.2.3 Input data 36
6.2.4 Output data 38
6.3 SDD ES EIP 40
6.3.1 Class 01H - General device data - acyclical access 40
6.3.2 Class A2H - Device data of SDD ES EIP and all safety devices - acyclical access 40
6.3.3 Class A3H - Device data of a safety device - acyclical access 43
6.3.4 Class A0H - input data - cyclical access (implicit messaging) 44

6.3.5 Class A1H - output data - cyclical access (implicit messaging) ... 45
6.4 SDD ES PROFINET ... 46
6.4.1 GSDML file ... 46
6.4.2 Device data ... 46
6.4.2.1 Slot division ... 46
6.4.2.2 Submodule input data (slots 1- 16) ... 46
6.4.2.3 Submodule output data (slots 1- 16) ... 47
6.4.3 Global data (slot 17) ... 48
6.4.3.1 Submodule input data ... 48
6.4.3.2 Submodule output data ... 49
6.4.4 PROFINET Records ... 50
6.4.5 Diagnostics alarms ... 50

7 Events of Safety Device Diagnostics ... 52

8 Configuration guide ... 55
8.1 General implementation assistance ... 55
8.2 Program example ... 56
8.2.1 Variable list ... 56
8.2.2 Program section ... 60

1 Introduction

1.1 Definition of symbols

Information that is particularly important is identified as follows:

DANGER!

This warning must be heeded! It warns of a hazardous situation that poses an immediate threat of serious injury and death and indicates preventive measures that can be taken.

WARNING!

This warning must be heeded! It warns of a hazardous situation that could lead to serious injury and death and indicates preventive measures that can be taken.

CAUTION!

This refers to a hazard that can lead to a less serious or minor injury plus material damage, and also provides information on preventive measures that can be taken.

NOTICE

This describes a situation in which the product or devices could be damaged and also provides information on preventive measures that can be taken. It also highlights areas within the text that are of particular importance.

INFORMATION

This gives advice on applications and provides information on special features.

2 Overview

2.1 Safety Device Diagnostics

Safety Device Diagnostics (SDD) is used to expand a safety system by a diagnostic function and a control function with a fieldbus connection.

Components of SDD:

- Safety devices are Pilz devices that are suitable for the SDD.
- Safe evaluation devices suitable for evaluating a safety device.
- Pilz SDD ES Module: Fieldbus modules for communication between fieldbus and safety devices.

Diagnostic information is also output on the fieldbus module's display.

When using an SDD ES ETH, SDD ES EIP or SDD ES PROFINET the output is also made via a web server.

The output on the display and in the web server is described in the operating manuals for the fieldbus modules.

The following types of data are transferred to the fieldbus and read in.

- Process data
  - Information and commands on safety functions (OSSD, guard locking, …)
- Device data
  - Material number, serial number, product version, actuator ID, …
- Configuration data
  - Behaviour of the control of safety devices with guard locking (control of the guard locking via SDD can be switched on and off on PROFINET, EtherNET/IP and ETH using Autoinit)

The transfer of process data is performed cyclically, of device data it is acyclical.

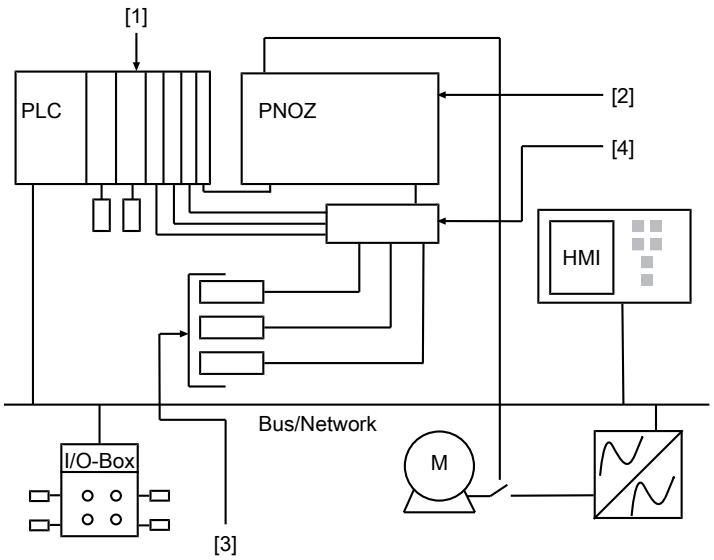
Communication is performed via ring protocol with telegrams.

A telegram is sent from the fieldbus module to the diagnostic input of the first safety device and transferred via the diagnostic output of the safety device to the next safety device. It is sent back to the fieldbus module from the last safety device.

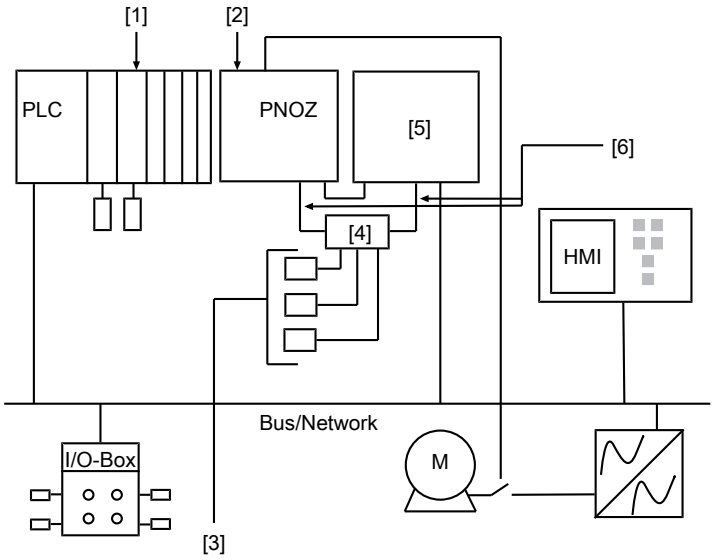
A maximum of 16 safety devices can be connected.

The following safety devices are suitable for SDD:

Product name	Device Version
PSEN cs1/2/3/4	From 2.0
PSEN cs5/6	From 2.0
PSEN ml b	From 2.0
PSEN ml s	From 2.0



Example: System with OSSDs, evaluation device, actuator technology - without SDD



Example: System with OSSDs, evaluation device, actuator technology - with SDD

Legend

[1] Control system

[2] Evaluation device, e.g. PNOZ

[3] Safety devices

[4] Passive junction for connecting safety devices to the evaluation device and control system

[5] Fieldbus module for diagnostics and connection to bus system

[6] Wires to connect the safety devices to the evaluation device and fieldbus module

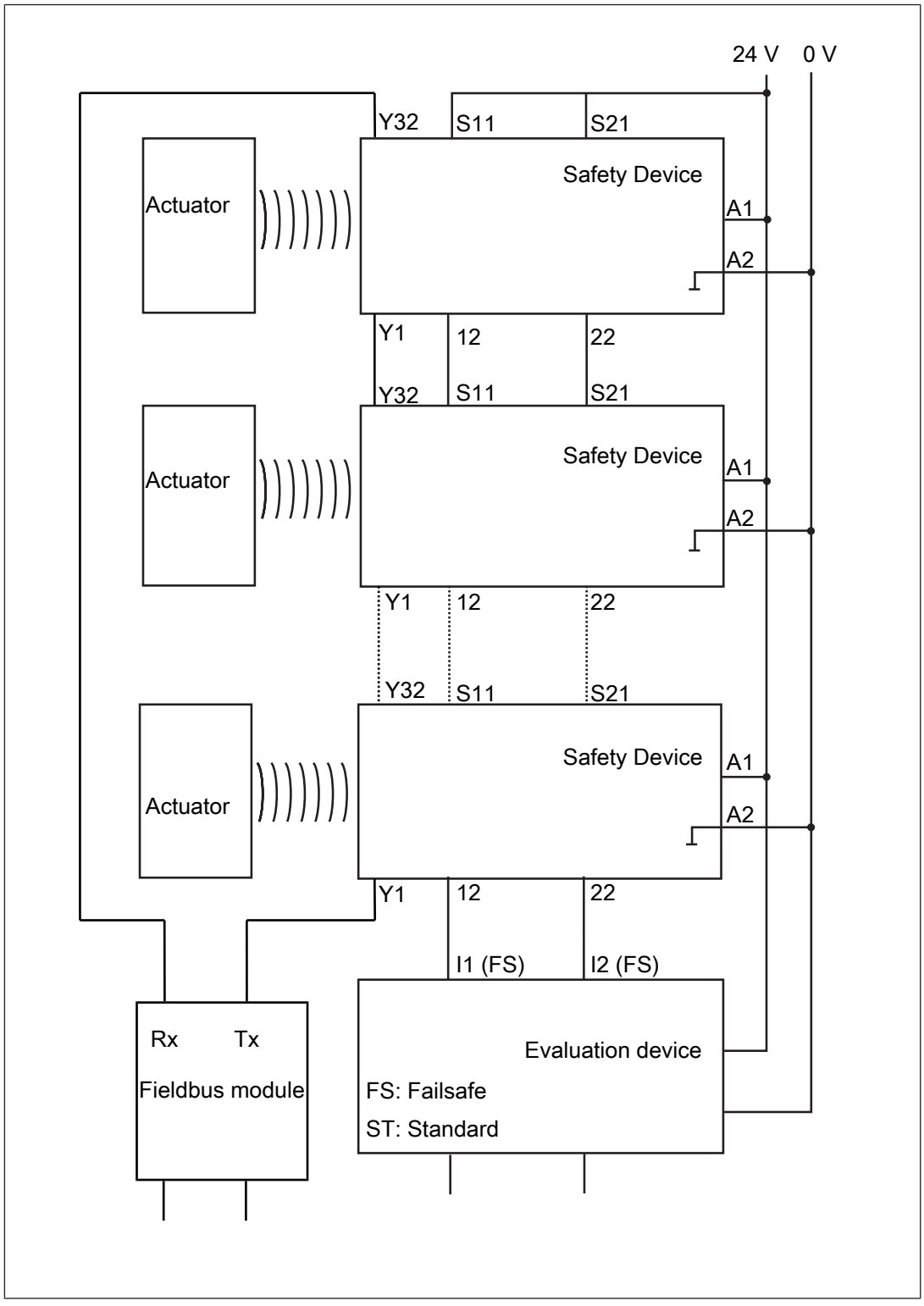
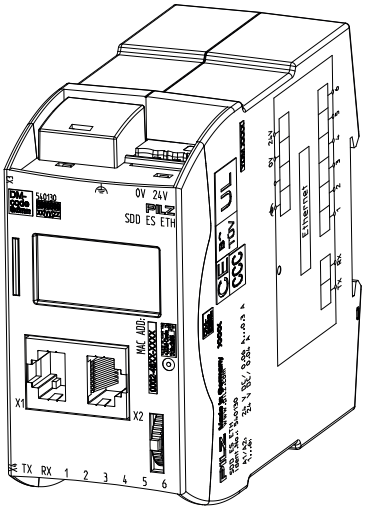
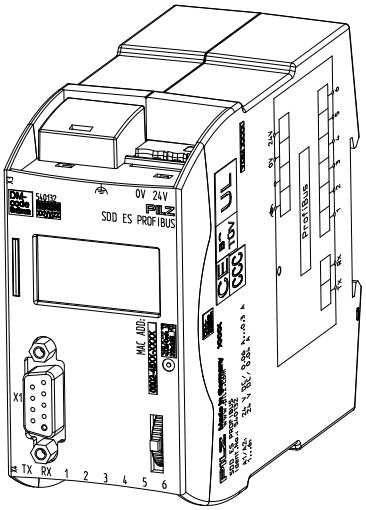


Fig.: Example for connecting three safety devices in series with SDD

2.2 Communication with fieldbus

The following fieldbus modules are possible:

Fieldbus module	Example representation
SDD ES ETH SDD ES EIP SDD ES PROFINET	
SDD ES PROFIBUS	

3 Safety

3.1 Intended use

- Safety device diagnostics are used to process diagnostic data.
  - Transfer of diagnostic data to a user program. This data may only be used for non-safety purposes, e.g. visualisation.
  - Display of diagnostic information on the display of the fieldbus module

The following is deemed improper use in particular:

- Any component, technical or electrical modification to a product
- Use of a product outside the areas described in the product documentation
- Any use that is not in accordance with the documented technical details.

3.2 Safety regulations

3.2.1 Additional documents that apply

Please read and take note of the following documents:

- Operating manual for the relevant Pilz safety device
- Operating manual of a passive junction, for example:
  - PSEN ix2 F4 code
  - PSEN ix2 F8 code
  - PDP67 F 4 code
  - PSEN Y junction
- Operating manual for the fieldbus module, for example SDD ES ETH or SDD ES PROFIBUS

You will need to be conversant with the information in these documents in order to fully understand this operating manual.

3.2.2 Use of qualified personnel

The products may only be assembled, installed, programmed, commissioned, operated, maintained and decommissioned by persons who are competent to do so.

A competent person is a qualified and knowledgeable person who, because of their training, experience and current professional activity, has the specialist knowledge required. To be able to inspect, assess and operate devices, systems and machines, the person has to be informed of the state of the art and the applicable national, European and international laws, directives and standards.

It is the company’s responsibility only to employ personnel who

- Are familiar with the basic regulations concerning health and safety / accident prevention,
- Have read and understood the information provided in the section entitled Safety
- Have a good knowledge of the generic and specialist standards applicable to the specific application.

3.2.3 Warranty and liability

All claims to warranty and liability will be rendered invalid if

- The product was used contrary to the purpose for which it is intended,
- Damage can be attributed to not having followed the guidelines in the manual,
- Operating personnel are not suitably qualified,
- Any type of modification has been made (e.g. exchanging components on the PCB boards, soldering work etc.).

3.2.4 Disposal

- When decommissioning, please comply with local regulations regarding the disposal of electronic devices (e.g. Electrical and Electronic Equipment Act).

4 System structure

4.1 Structure of a system with SDD - example

The example shows the structure of a safety system with Safety Devices, evaluation device, control system, actuators and a fieldbus module for SDD.

The Safety Devices can be connected decentrally via interface elements such as cable separators to the evaluation device and the fieldbus module, or centrally via terminals.

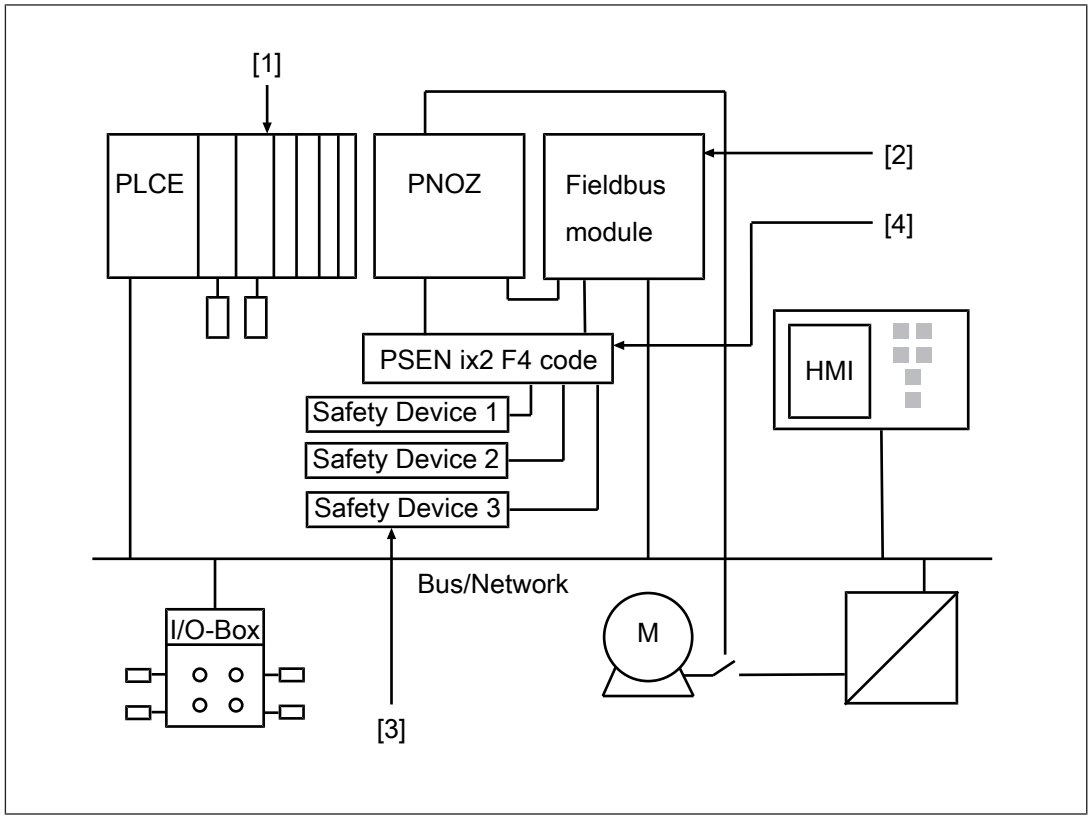


Fig.: Central connection of the fieldbus module

Legend

[1] Control system

[2] Evaluation device, e.g. PNOZ

[3] Safety devices (number in the example: 3)

[4] Connection of the safety devices to the evaluation device and control system via the interface PSEN ix2 F4 code

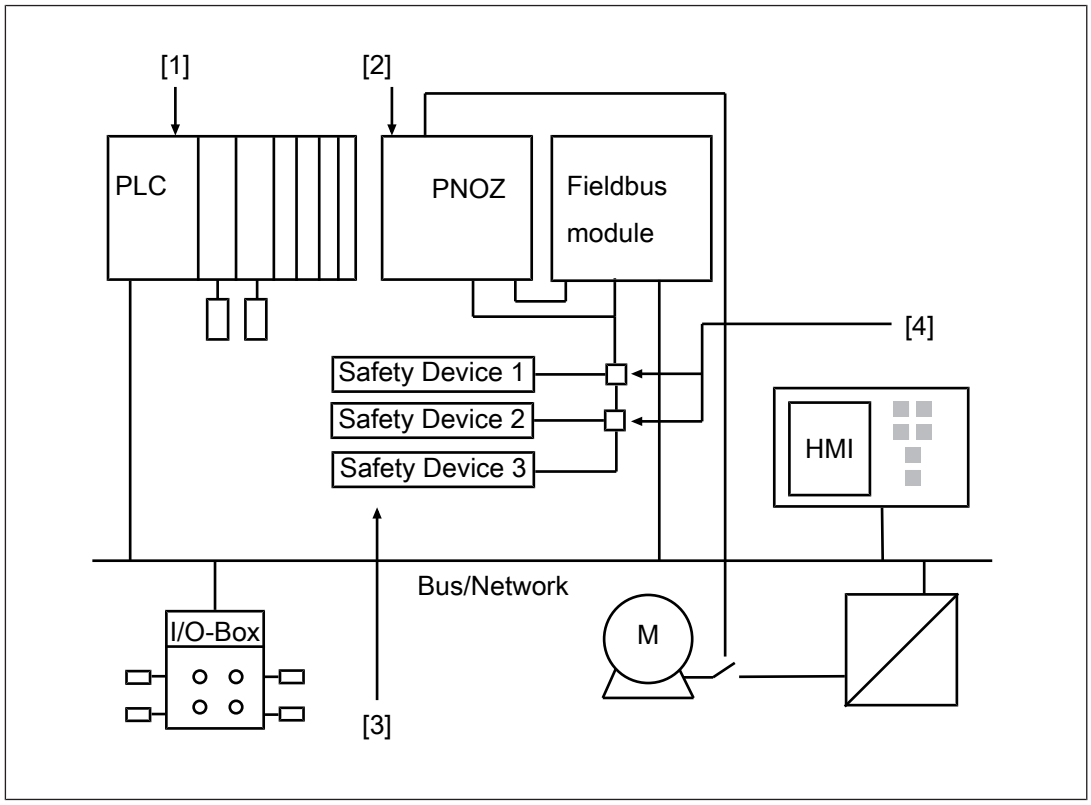


Fig.: Decentralised connection of the fieldbus module via PSEN Y Junction

Legend

[1] Control system

[2] Evaluation device, e.g. PNOZ

[3] Safety devices (number in the example: 3)

[4] Connection of the safety devices to the evaluation device and to the control system via PSEN Y Junction

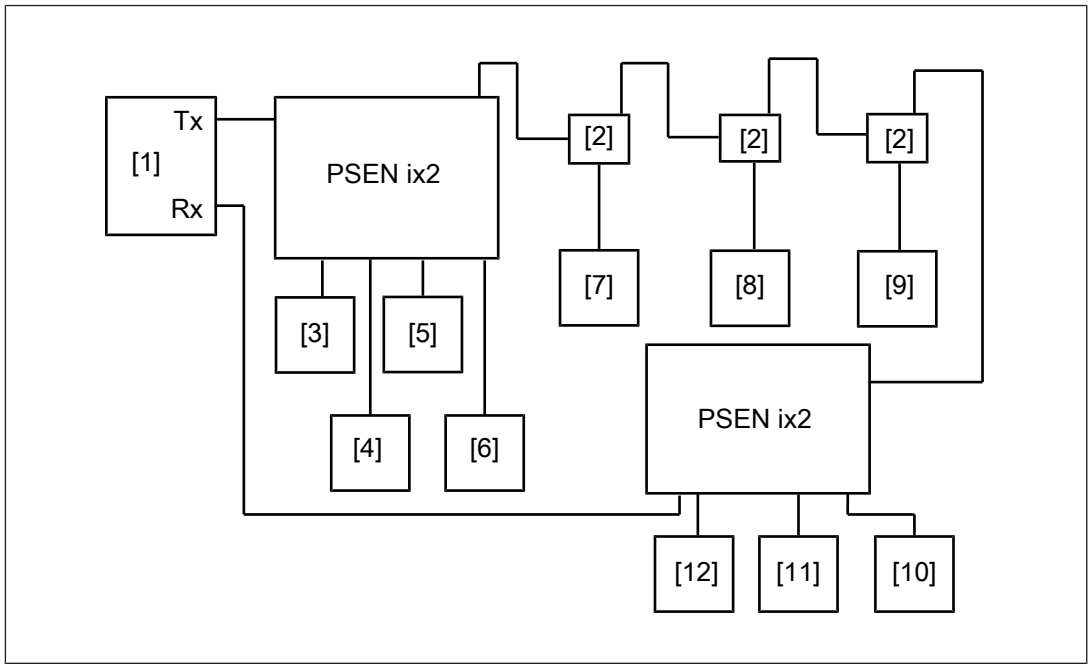


Fig.: Example for connecting 10 safety devices with two PSEN ix2 and three PSEN Y-junctions

Legend

[1]	Fieldbus module for diagnostics and connection to bus system
[2]	PSEN Y Junctions
[3-12]	Safety devices

4.2 Construction of safety circuits with the SDD

Examples for the construction of safety circuits with the SDD

In Wiring [19] further examples with details on the maximum cable lengths between the individual components are listed.

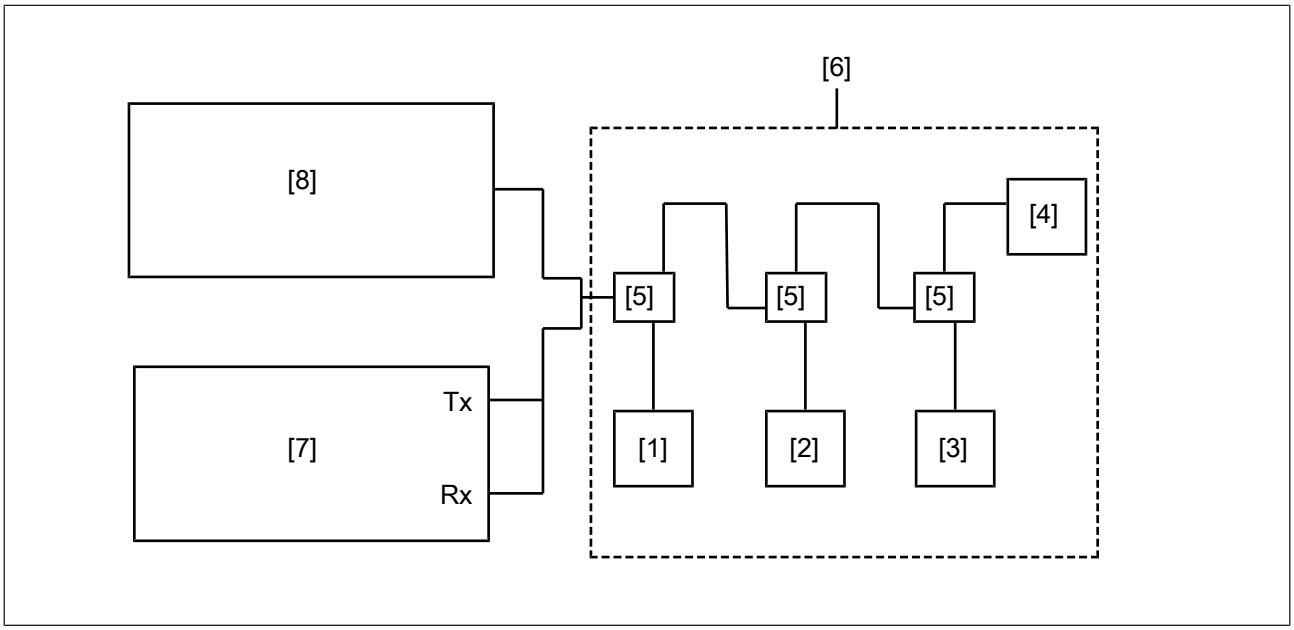


Fig.: One safety circuit, wiring in a chain

Legend

[1]-[4]	Safety Devices of the safety circuit
[5]	PSEN Y Junction
[6]	Safety circuit
[7]	Evaluation device (e.g. PNOZ) of the safety circuit, for connecting the safety signals
[8]	Fieldbus module (e.g. SDD ES ETH) for diagnostics of the Safety Devices of the safety circuit

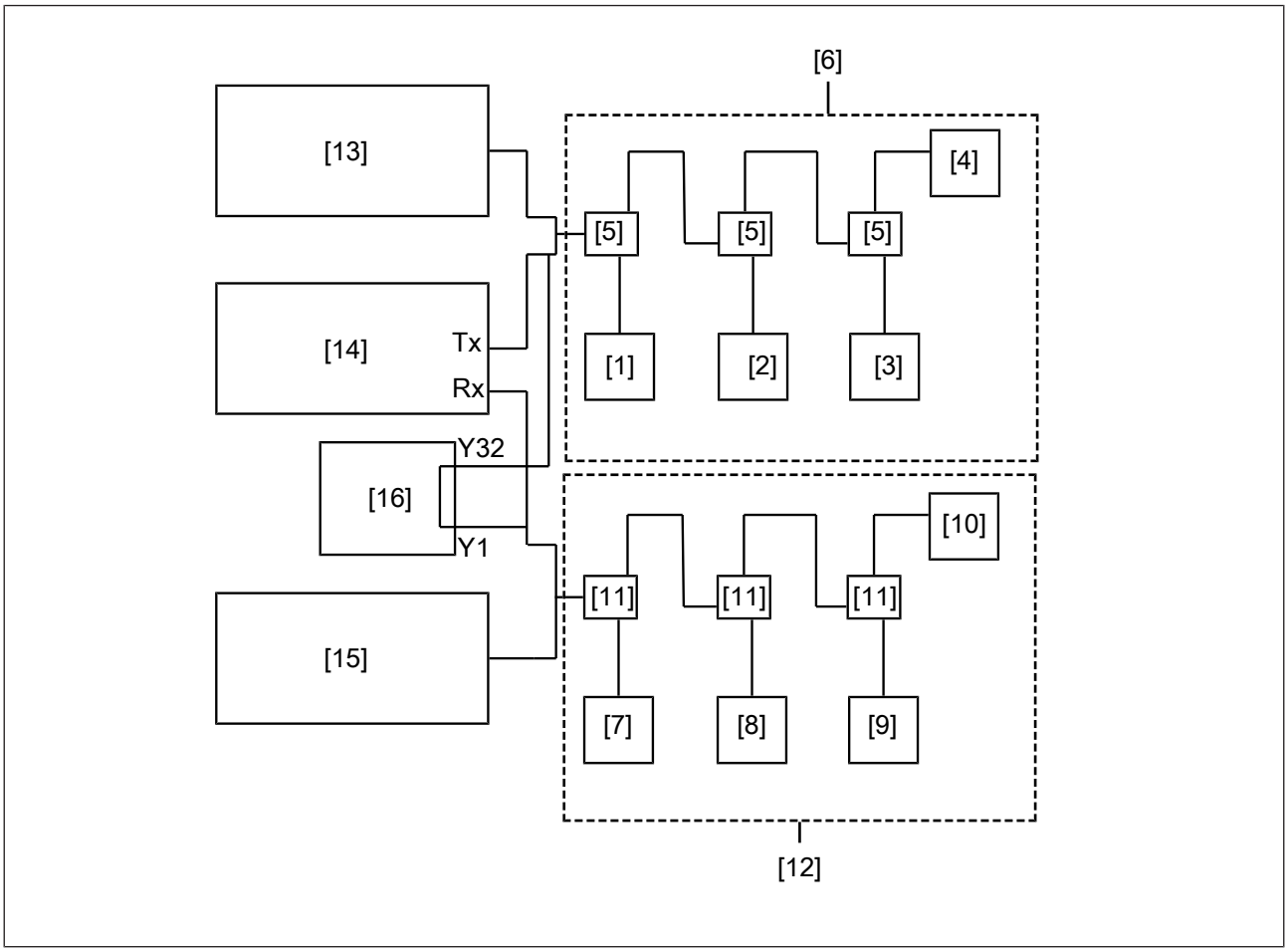


Fig.: Two safety circuits, wiring in two chains

Legend

[1]-[4]	Safety Devices of the safety circuit
[5]	PSEN Y Junctions
[6]	First safety circuit
[7]-[10]	Safety Devices of the safety circuit
[11]	PSEN Y Junctions
[12]	Second safety circuit
[13]	Evaluation device (e.g. PNOZ) of the first safety circuit, for connecting the safety signals
[14]	Fieldbus module (e.g. SDD ES ETH) for diagnostics of the Safety Devices from both safety circuits
[15]	Evaluation device (e.g. PNOZ) of the second safety circuit, for connecting the safety signals
[16]	Terminal for connecting the Y32 of the Safety Device [4] and Y1 of the Safety Device [7]

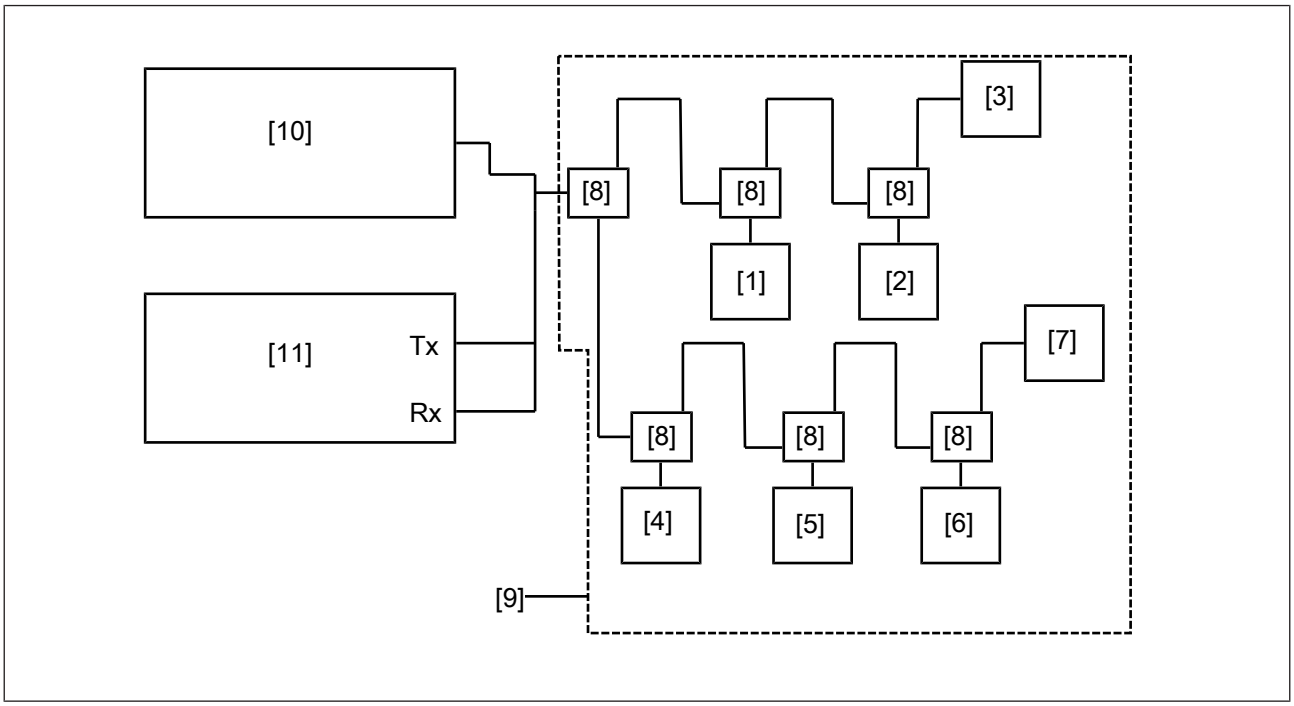


Fig.: A safety circuit connected to a fieldbus module, wiring split up in two chains

Legend

[1]-[7]	Safety Devices of the safety circuit
[8]	PSEN Y Junctions
[9]	Safety circuit
[10]	Evaluation device (e.g. PNOZ) of the safety circuit, for connecting the safety signals
[11]	Fieldbus module (e.g. SDD ES ETH) for diagnostics of the Safety Devices of the safety circuit

4.3 Examples for the connection of further devices

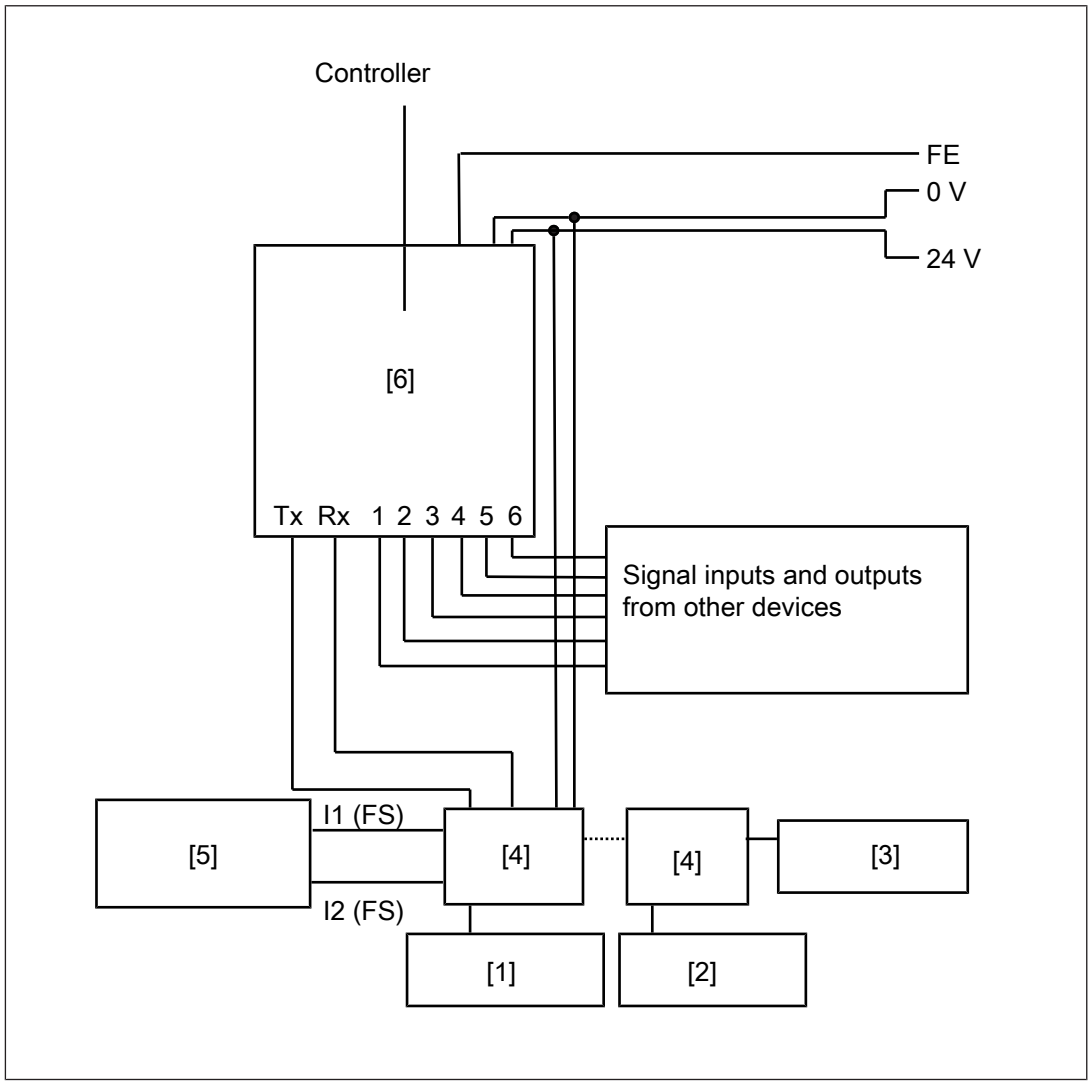


Fig.: Connection of further devices to the configurable inputs/outputs (GPIO)

Legend

[1]-[3] Safety devices

[4] PSEN Y junctions

[5] Evaluation device (e.g. PNOZ) of the safety circuit, for connecting the safety signals

[6] Fieldbus module (e.g. SDD ES ETH) for diagnostics of the safety devices of the safety circuit

4.4 Wiring

Wiring examples for applications with safety devices PSENcode

Please note:

- Details for wiring in the operating manuals of the safety devices
- Max. line capacitance: 65 pF/m
- Max. conductor resistance: 78 Ohm/km
- For the wiring of the Safety Devices with the SDD only the wiring aids of cable separators provided in Additional documents that apply [📖 10] may be used.
- Max. length of the connection cable between the fieldbus module and the first Safety Device (Safety Device 1 in Wiring in a chain): 50 m
- Max. length of the connection cable between two Safety Devices: 50 m
- Max. length of the connection cable from the last Safety Device (Safety Device 3 in Wiring in a chain) to the fieldbus module: 150 m

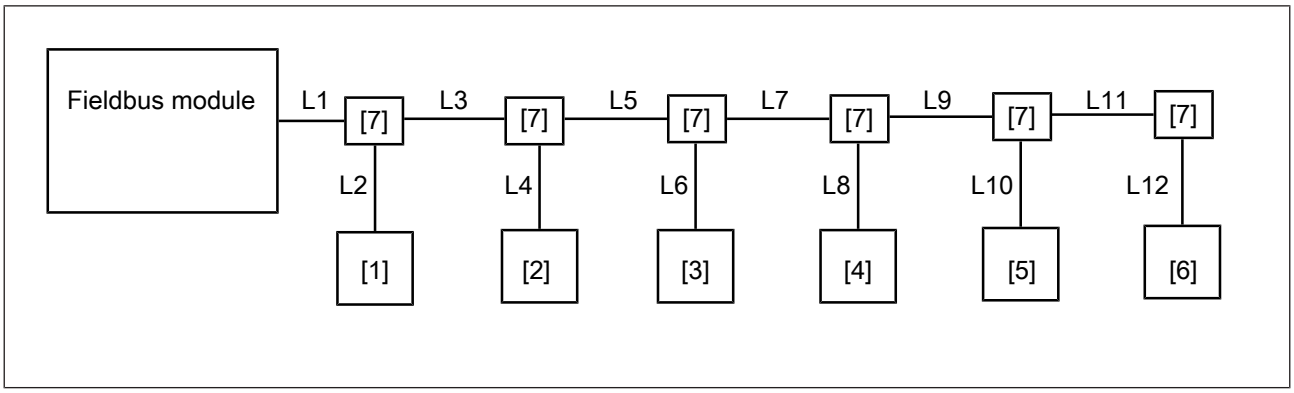


Fig.: Wiring in a chain

Legend

[1]-[6]	Safety devices
[7]	PSEN Y junctions
L1-L12	Connection cable

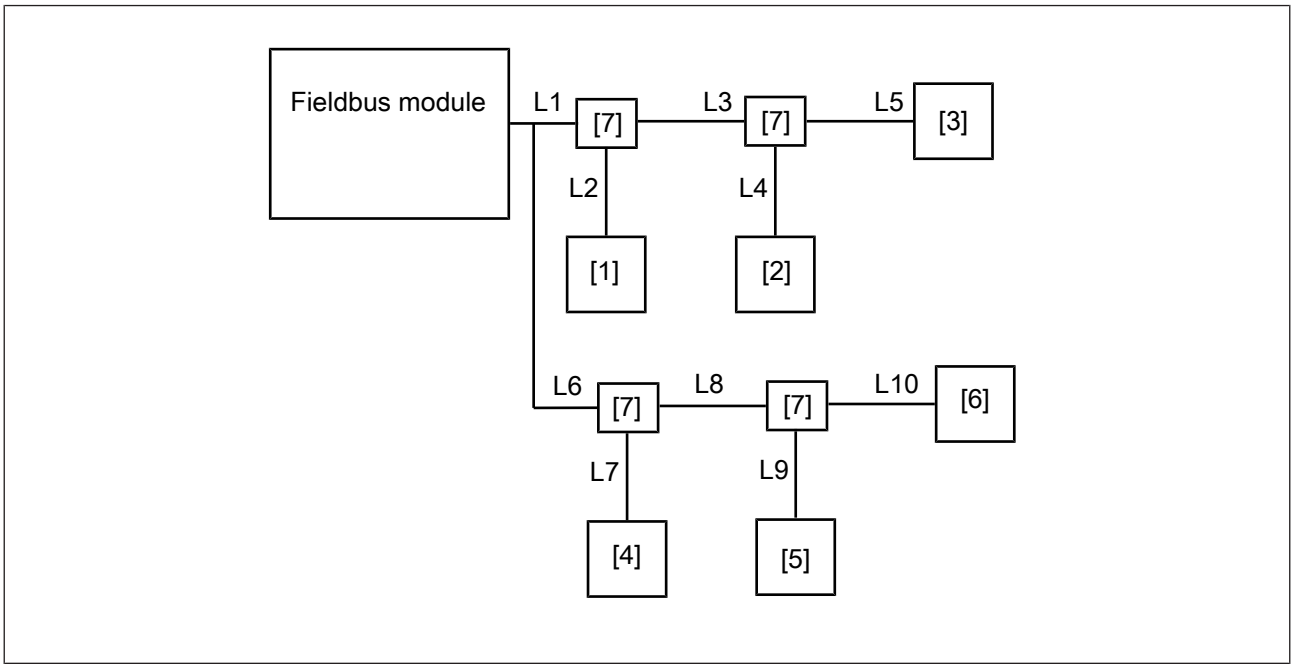


Fig.: Wiring in two chains

Legend

[1]-[6]	Safety devices
[7]	PSEN Y Junctions
L1-L10	Connection cable
L1+L2	Max. length < 50 m
L2+L3+L4	Max. length < 50 m
L4+L5	Max. length < 50 m
L5+L3+L1+L6+L7	Max. length < 50 m
L7+L8+L9	Max. length < 50 m
L9+L10	Max. length < 50 m
L10+L8+L6	Max. length < 150 m

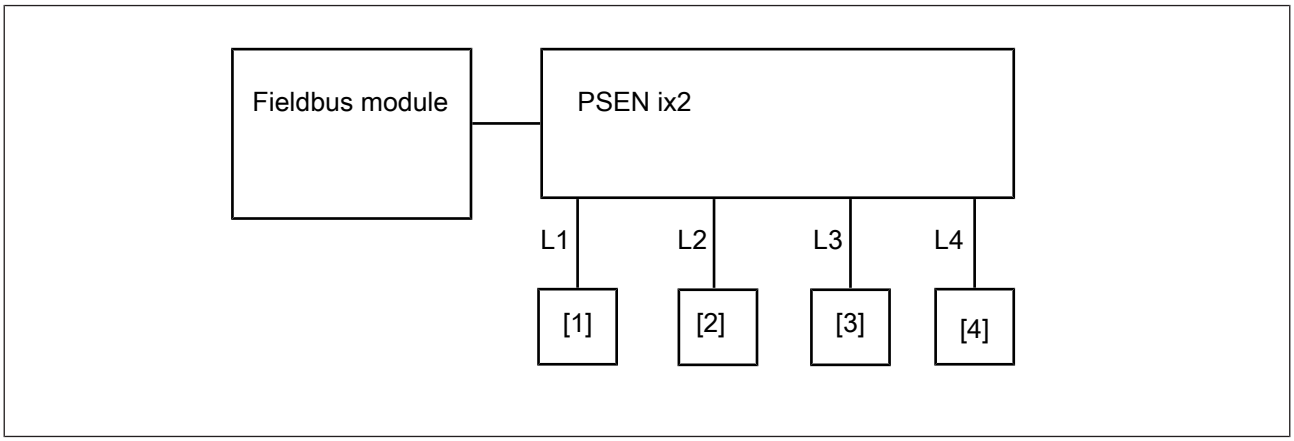


Fig.: Wiring centrally in the control cabinet

Legend

[1]-[4] Safety devices

L1-L4 Connection cable

L1+L2 Max. length < 50 m

L2+L3 Max. length < 50 m

L3+L4 Max. length < 50 m

Wiring examples for applications with safety devices PSENmlock and optionally with PSENcode

Please note:

- The features of the safety devices PSENmlock are decisive.

  Please comply with the information regarding wiring in the operating manuals of PSENmlock.

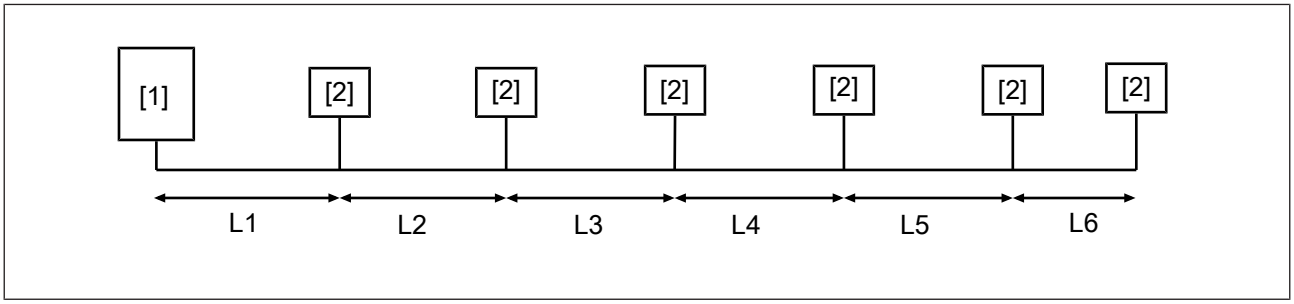


Fig.: Wiring of six safety devices PSENmlock

Legend

[1] Fieldbus module

[2] Safety devices

- Supply voltage at the safety control system ≥ 24 V

Number of safety switches	L1	L2	L3	L4	L5	L6	Overall length
1	120 m						120 m
2	60 m	60 m					120 m
3	50 m	50 m	20 m				120 m
4	50 m	30 m	20 m	20 m			120 m
5	50 m	20 m	20 m	20 m	10 m		120 m
6	20 m	20 m	20 m	20 m	20 m	20 m	120 m

5 List of data

5.1 Data types

The data stored in the registers are divided into the types.

- Process data
- Device data
- Configuration data

The transmitted data vary depending on the used device type (Coded Switch A, B)

Process data

- Actuator enable/safety gate,
- Guard locking/lock,
- Safety enable,
- States of the configurable inputs/outputs (GPIO), Wiring example [📖 18]
- Event registered,
- Operational readiness,
- Safety input 1+2,
- Safety output 1+2/OSSD1+2

Differences in the process data depending on the device type

With the device types Coded Switch A/Coded Switch B/Mlock System, there are differences in possible events and status information.

Product Name	Device type
PSEN cs1/2/3/4	Coded switch B
PSEN cs5/6	Coded switch A
PSEN ml b/s	Mlock system

Device data

All the data that are not listed under process data or configuration data.

Configuration data

Behaviour of the control of safety devices with guard locking.

5.2 Overview

5.2.1 Safety devices of all types

Content/description	Meaning/value range/measure	
Status of the fieldbus module	1	Devices are polled (LED start up lights up yellow)
	2	Regular operation (LED power lights up green, LED start up is off)
	3	Error in the SDD communication (fault LED flashes red)
Number of safety devices	Max. 16	
Event registered	A safety device registers an event. Details on the event are listed in further data fields.	
Actuator enable	Actuator is within the response range. The safety gate of the safety device was closed. Corresponds to LED Safety Gate.	
Guard locking	Status of the guard locking is signalled (activated or deactivated). Corresponds to LED Lock.	
Safety enable	OSSDs of the safety device have switched to ON state	
States of the configurable inputs/outputs (GPIO) of the fieldbus module	High signal at the GPIOs	
	010110	GPIO contacts 2, 4, 5 are high
Current time	Seconds since switching on the fieldbus module	
Events and status from safety device 1 Bits 0-31	Detailed description see Table of events and status information per safety device [📖 25]	
Number of safety device that detects a chain break.	2	Safety device 3 detects a break between safety device 2 and safety device 3.
Number of additional teach-in processes still permitted for the connected safety devices	0	with coded version of the safety device
	Max. 8	with fully coded version of the safety device
	Max. 1	with uniquely coded version of the safety device
Temperature of the connected safety devices in Celsius (only with PSENcode, Coded Switch A)	Temperature in safety device, rough standard value of the ambient temperature, no temperature sensor for the process.	
Short name of the actuator of the connected safety devices	256 short names are distinguished. A limited number of actuators at a machine can be distinguished and/or manipulations with a possible spare actuator can be detected.	
Software versions of connected safety devices	This data from the safety devices is used to align functionalities when expanding the Safety Device Diagnostics at a later stage.	
Product version of connected safety devices		
Protocol version of connected safety devices		

5.2.2 Table of events and status information per safety device

Bit	Meaning/action
0	1 = Quick change of state of the actuator
1	1 = Safety device in partial operation lock status
2	1 = Safety device in partially operated status
3	1 = Supply voltage is at the upper limit of the permitted range (see Technical details for safety device) Deviation coded switch B: Reserved
4	1 = Supply voltage is at the lower limit of the permitted range (see Technical details for safety device) Deviation coded switch B: Reserved
5	1 = Wrong actuator
6	0 = Teaching in process and subsequent reset completed 1 = New actuator has been taught in
7	1 = Supply voltage above the permitted range (see Technical details for safety device) Deviation coded switch B: Reserved
8	1 = Supply voltage below the permitted range (see Technical details for safety device) Deviation coded switch B: Reserved
9	Ready for operation
10	Actuator enable/safety gate
11	Safety input 1 (unusable with Mlock System)
12	Safety input 2 (unusable with Mlock System)
13	Safety output 1/OSSD 1
14	Safety output 2/OSSD 2
15	Pilz coding type coded 1 = Coded
16	Pilz coding type fully coded 1 = Fully coded
17	Pilz coding type uniquely coded 1 = Uniquely coded
18-31	Reserved

5.3 Writeable data (outputs)

Description	Measure/value range
States of the configurable inputs/ outputs (GPIO) of the fieldbus module (6 Bit field)	Here the GPIOs can be set to a high or low signal.
	010110 = Input/output 2, 4 and 5 are set to high signal
Delete event list (only with SDD ES ETH and SDD ES PROFINET)	Event list is deleted Old events are removed from the display (e.g. for a quick view while the plant is running). All the old events are retained, and they can be called up again via a reset of the voltage.
Counter for guard locking	The counter has to be increased by 1 each when the control system wants to change the guard locking of a safety device.
Intended state of the activation or deactivation of guard locking	The command is implemented when the counter of the guard locking has been increased by 1 **and** when there is the safety enable at S31 and S41 on the safety devices with guard locking.
	0 = A deactivation command is sent to all the devices with 0.
	1 = An activation command is sent to all the devices with 1.
Variable for the control of the guard locking activation Lock Autoinit	Lock Autoinit is required to guarantee the exclusive control of the Lock function via Safety Device Diagnostics, particularly when starting up the machine.
	▸ Activation means that PSENmlock does not switch when High signals come through the safety outputs, but the commands via the fieldbus module are missing. ▸ Deactivation means that activation will be lifted again. The entry can be overwritten by an entry on the display of the SDD ES device.

6 Fieldbus Modules

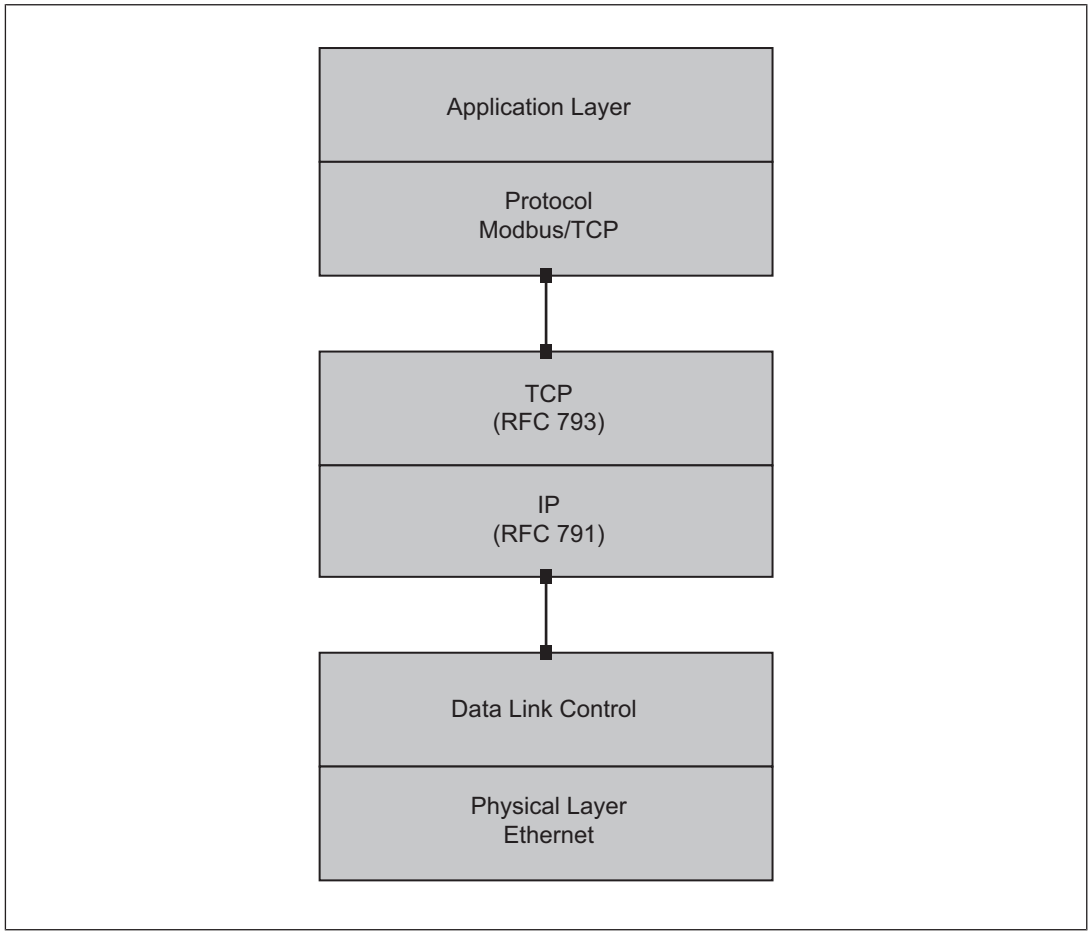
6.1 SDD ES ETH

For communication on the fieldbus, the input and output areas are reserved. The information is processed in byte mode or word mode.

6.1.1 Modbus/TCP

6.1.1.1 General

Modbus/TCP is an open fieldbus standard published by the User Group MODBUS-IDA (see www.Modbus-IDA.org).



Modbus/TCP is a protocol based on Industrial Ethernet (TCP/IP over Ethernet). It is one of the protocols with Client/Server communication. Data is transferred via a request/response mechanism using function codes (FC).

Modbus/TCP is connection-oriented, i.e. before usable data is transferred via Modbus/TCP, a connection must first be established between two Modbus/TCP interfaces. The initiator of the connection is called the Client. The communications partner with which the Client establishes the connection is called the Server. Whether a connection on a device assumes the role of Client or Server is defined when a connection is configured. As a result, the Server/Client role only applies for that specific connection.

Data can be accessed via various Modbus/TCP data areas.

The tables below show the relationship between Modbus/TCP data areas and the content of the data areas.

6.1.1.2 Default settings that cannot be modified

- Port number for data exchange via Modbus/TCP connection: 502
  - The port number is preset. It is not shown in the display of the fieldbus module - System description and it cannot be changed.
- Communication via http (web server): Port 80
- Auto negotiation and Auto crossover are activated
- max. 8 Modbus/TCP connections possible
- 2 users can simultaneously log into the area of the web application

6.1.1.3 Default settings that can be changed

- Timeout can be set from min. 2 s to max. 30 s (see SDD ES ETH output data [📖 34])

6.1.1.4 Recommended settings of the client

Pilz recommends the following connection settings of the Client:

- Connection cycle time: 100 ms
- Connection timeout: 5 s
- Do not activate Keep Alive

6.1.1.5 Function codes

The following function codes (FC) are available for communication of the safety devices via Modbus/TCP:

Function code	Function	
Input register		
FC 04	Read Input Register	The connection Client reads Word data from the connection Server, data length ≥ 1 Word, (receive data from 3x)
Output register		
FC 03	Read Holding Register	The connection Client reads Word data from the connection Server, data length ≥ 1 Word, (receive data from 4x)

Function code	Function	
FC 06	Write Single Register	The connection Client writes to one Word datum in the connection server, data length = 1 Word, content: Input data (send data to 4x)
FC 16	Write Multiple Registers	The connection Client writes to multiple Word data in the connection server, data length ≥ 1 Word, content: Input data (send data to 4x)

6.1.2 Input data

INFORMATION

The "Safety device diagnostics" system section starts the addressing for Modbus/TCP data areas at "1". On other devices, addressing may start at "0".
Please refer to the operating manual provided by the relevant manufacturer.

Register	Data type	Contents	Sample value in HEX/ Bit field
1	INT16U	Status of SDD ES ETH	0002 (hex) = 2 (decimal)
			Running
2	INT16U	Number of safety devices	0005 (hex) = 5 (decimal)
			5 safety devices connected
3	INT16U	Event registered Bit field with 16 Bit for the status of the 16 safety devices	000C (hex) = 0000000000001100 (binary)
			Safety device 3 and 4 signal an event
4	INT16U	Actuator enable/safety gate Bit field with 16 Bit for the status of the 16 safety devices	000B (hex) = 0000000000001011 (binary)
			Safety devices 1, 2 and 4 signal enable
5	INT16U	Guard locking/lock Bit field with 16 Bit for the status of the 16 safety devices	000C (hex) = 0000000000001100 (binary)
			Safety devices 3 and 4 signal guard locking active
6	INT16U	Safety enable (OSSD 1&2) Bit field with 16 Bit for the status of the 16 safety devices	000D (hex) = 0000000000001101 (binary)
			OSSD 1&2 of safety devices 1, 3 and 4 signal safety enable

Register	Data type	Contents	Sample value in HEX/ Bit field
7	INT16U	States of the inputs of SDD ES ETH Bit field with 6 Bit for the input values of GPIO	0007 (hex) = 00000000000111 (binary) Inputs 1, 2 and 3 are high
8	INT16U	Current time (seconds since switching on the SDD ES ETH)	012C (hex) = 300 (decimal) 5 minutes have passed since switching on the field-bus module
9	INT16U	Last signalled event from safety device 1, Bits 0-15 0 = Bit not set, 1 = Bit set (see Table of events and status information per safety device [📖 25])	0040 (hex) = 0000000001000000 (binary) Safety device 1 has taught-in a new actuator and it has to be restarted
10	INT16U	Last signalled event from safety device 1, Bits 16-31 0 = Bit not set, 1 = Bit set (see Table of events and status information per safety device [📖 25])	0004 (hex) = 0000000000000010 (binary) Safety device 1 is uniquely coded
11	INT16U	Reserved	
12	INT16U	Last signalled event from safety device 2, Bits 0-15	see Register 9
13	INT16U	Last signalled event from safety device 2, Bits 16-31	see Register 10
14	INT16U	Reserved	
…			
55	INT16U	Last signalled event from safety device 16, Bits 16-31	see Register 10
56	INT16U	Reserved	
57	INT8U	Number of safety device that detects a chain break	2 = Safety device 3 detects a break between safety device 2 and safety device 3

Register	Data type	Contents	Sample value in HEX/ Bit field/value range
65-80	INT16U	Number of additional teach-in processes still permitted for the connected safety devices (1 Word with 2 bytes per safety device)	Value range 0-8 0004 (hex) = 4 (decimal) 4 teach-in processes remain

Register	Data type	Contents	Sample value in HEX/ Bit field/value range	
81-96	INT16S	Temperature of the connected safety devices on Celsius (1 Word with 2 bytes per safety device)	0012 (hex) = 18 (decimal)	
			18 °C	
97-128	INT16U	Reserved		
129-144	INT16U	Actuator short name of the connected safety devices (1 Word with 2 bytes per safety device)	00F0 (hex) = 240 (decimal)	
			Actuator with the short name 240	
145-160	INT16U	Order number of safety device part 1	Not used	
161-176	INT16U	Order no. of the safety device part 2 (1 Word with 2 bytes per safety device)	Part 2 = 0008 (hex) Part 3 = 456B (hex	
			Part 2 + 3 = 0008456B = 542059 (decimal)	
177-192	INT16U	Order number of the safety device part 3 (1 Word with 2 bytes per safety device)	Part 2 = 0008 (hex) Part 3 = 456B (hex	
			Part 2 + 3 = 0008456B = 542059 (decimal)	
193-209	INT16U	Reserved		
209-224	INT16U	Serial number of the safety device part 2 (1 Word with 2 bytes per safety device)	Part 2 = 0000 (hex) Part 3 = 8C5E (hex	
			Part 2 + 3 = 00008C5E = 35934 (decimal)	
225-240	INT16U	Serial number of the safety device part 3 (1 Word with 2 bytes per safety device)	Part 2 = 0000 (hex) Part 3 = 8C5E (hex	
			Part 2 + 3 = 00008C5E = 35934 (decimal)	
241-273	INT16U	Reserved		
273-288	INT16U	Product version of the connected safety devices (1 Word with 2 bytes per safety device)	0002 (hex)	Version 2.00 (implemented in future)
289-320	INT16U	Reserved		

Register	Data type	Contents	Sample value in HEX/ Bit field/value range
1001	INT16U	Latest event First digit describes the safety device	1 - F = Safety device 1 - 16
		Second to fourth digit indicates the number	See Events of Safety Device Diagnostics [📖 52] (specific safety devices and all product groups)
1002	INT16U	Time of signalling the event (seconds in hex since the last start of the fieldbus module)	
1003	INT16U	Second latest event First digit describes the safety device	1 - F = Safety device 1 - 16
		Second to fourth digit indicates the number	See Events of Safety Device Diagnostics [📖 52] (specific safety devices and all product groups)
1004	INT16U	Time of signalling the event (seconds in hex since the last start of the fieldbus module)	
…			
1199	INT16U	Event 100 First digit describes the safety device	1 - F = Safety device 1 - 16
		Second to fourth digit indicates the number	See Events of Safety Device Diagnostics [📖 52] (specific safety devices and all product groups)
1200	INT16U	Time of signalling the event 100 (seconds in hex since the last start of the fieldbus module)	
1201-1300	INT16U	Reserved	

Register	Data type	Contents	Sample value in HEX/ Bit field/value range
1501	INT16U	Change of state of a safety device First digit describes the safety device	1 - F = Safety device 1 - 16
		Last digit describes the status	1 = Safety gate, actuator is within the response range 2 = Safety gate, the actuator is not within the response range 5 = LOCK closed 6 = LOCK open
1502	INT16U	Time of change of state (seconds in hex since the last start of the fieldbus module)	
1503	INT16U	Second latest change of state First digit describes the safety device	1 - F = Safety device 1 - 16
		Last digit describes the status	1 = Safety gate, actuator is within the response range 2 = Safety gate, the actuator is not within the response range 5 = LOCK closed 6 = LOCK open
1504	INT16U	Time of this change of state (seconds in hex since the last start of the fieldbus module)	
…			
1699	INT16U	Change of state 100 First digit describes the safety device	1 - F = Safety device 1 - 16
		Last digit describes the status	1 = Safety gate, actuator is within the response range 2 = Safety gate, the actuator is not within the response range 5 = LOCK closed 6 = LOCK open
1700	INT16U	Time of change of state 100 (seconds in hex since the last start of the fieldbus module)	

Register	Data type	Contents
2001-2003	INT16U	Reserved

6.1.3 Output data

INFORMATION

The "Safety device diagnostics" system section starts the addressing for Modbus/TCP data areas at "1". On other devices, addressing may start at "0".
Please refer to the operating manual provided by the relevant manufacturer.

Register	Data type	Content	Sample value in HEX/ Bit field
1	INT16U	States of the configurable inputs/ outputs of the SDD ES ETH (GPIO, 6 Bit field). Please note: States can only be polled by the PLC when PLC is selected for the safety device in the GPIO mapping of the field-bus module.	002A (hex) = 00000000101010 (binary) The GPIO 6, 4 and 2 are high
2	INT16U	Intended state of the guard locking	1111111111111011 (binary) The guard locking on safety device 3 is deactivated, it is activated on all the other safety devices
3	INT16U	Counter for implementing the intended state of the guard locking. Whenever the number increases, the intended state of the guard locking is implemented	0001 (hex) First implementation is performed.
4	INT16U	When starting, the fieldbus module activates the behaviour for the control of the guard locking (Lock Autoinit).	0001 (hex) = Guard locking activation only via Safety Device Diagnostics 0000 (hex) = Guard locking activation up to the first command via fieldbus using safety outputs of the safety control system

Register	Data type	Contents	Sample value in HEX/ Bit field/value range
1001	INT16U	Reserved	
1002	INT16U	Reserved	
1003	INT16U	Reserved	
1004	INT16U	Deletion of event list The output date has to be implemented as a trigger signal. A permanent prompt to delete can lead to a restart of the SDD ES ETH.	7789 (hex) Event list is deleted

Register	Data type	Contents
2001-2003	INT16U	Reserved

Register	Data type	Contents	Sample value in HEX/ Bit field/value range
4019	INT16U	Setting of the Timeout	Min. 2 s, max. 30 s 15000 (decimal) = 15 s

6.2 SDD ES PROFIBUS

6.2.1 Restriction when connecting PSENmlock

PSEN ml b/s can be connected to SDD ES PROFIBUS.

The control function of individual PSENmlock devices is not yet implemented.

6.2.2 GSD file

INFORMATION

The GSD file is available on the Internet at www.pilz.de.

The GSD file (device database file) is the device description file for PROFIBUS. The format of the GSD file is specified in the standards IEC 61158 and IEC 61784. All of a device's Slave-specific properties are stored in the GSD file.

6.2.3 Input data

Address	Data type	Contents	Sample value in HEX/Bit field
0	INT8U	Status of SDD ES PROFIBUS	02 (hex) = 2 (decimal)
			Running
1	INT8U	Number of safety devices	05 (hex) = 5 (decimal)
			5 safety devices connected
2-3	INT16U	Event registered Bit field with 16 Bit for the status of the 16 safety devices	000C (hex) = 0000000000001100 (binary)
			Safety device 3 and 4 signal events
4-5	INT16U	Actuator enable/safety gate Bit field with 16 Bit for the status of the 16 safety devices	000B (hex) = 0000000000001011 (binary)
			Safety devices 1, 2 and 4 signal enable
6-7	INT16U	Guard locking/lock Bit field with 16 Bit for the status of the 16 safety devices	000C (hex) = 0000000000001100 (binary)
			Safety devices 3 and 4 signal guard locking active
8-9	INT16U	Safety enable Bit field with 16 Bit for the status of the 16 safety devices	000D (hex) = 0000000000001101 (binary)
			Safety devices 1, 3 and 4 signal safety enable
10	INT8U	States of the inputs of SDD ES PROFIBUS Bit field with 6 Bit for the input values of GPIO	07 (hex) = 00000111 (binary)
			Inputs 1, 2 and 3 are high
11-12	INT16U	Current time (seconds since switching on the SDD ES PROFIBUS)	012C (hex) = 300 (decimal)
			5 minutes have passed since switching on the fieldbus module
13	INT8U	Last signalled event from safety device 1, Bits 0-7 0 = Bit not set, 1 = Bit set (see Table of events and status information per safety device [📖 25])	40 (hex) = 01000000 (binary)
			Safety device 1 has taught-in a new actuator and it has to be restarted
14	INT8U	Last signalled event from safety device 1, Bits 8-15 0 = Bit not set, 1 = Bit set (see Table of events and status information per safety device [📖 25])	02 (hex) = 00000010 (binary)
			Safety device 1 is ready for operation

Address	Data type	Contents	Sample value in HEX/Bit field
15	INT8U	Last signalled event from safety device 1, Bits 16-23 0 = Bit not set, 1 = Bit set (see Table of events and status information per safety device [📖 25])	04 (hex) = 00000100 (binary) Safety device 1 is uniquely coded
16-18	INT8U	Reserved	
19	INT8U	Last signalled event from safety device 2, Bits 0-7	see address 13
20	INT8U	Last signalled event from safety device 2, Bits 8-15	see address 14
21	INT8U	Last signalled event from safety device 2, Bits 16-23	see address 15
22-24	INT8U	Reserved	
…			
105	INT8U	Last signalled event from safety device 16, Bits 16-31	see address 15
106-108	INT8U	Reserved	
109	INT8U	Number of safety device that detects a chain break	2 = Safety device 3 detects a break between safety device 2 and safety device 3

Address	Data type	Contents	Sample value in HEX/ Bit field/value range	
110	INT8U	Number of safety device, about which additional information is displayed	06 (hex) = 6 (decimal) Of Safety Device 6, the short name of the actuator, software version, protocol version, product version, teach-in processes still permitted and temperature are displayed.	
111-130	INT16U	Reserved		
131-132	INT16U	Actuator short name of the connected Safety Devices (1 word with 2 bytes per Safety Device)	00F0 (hex) = 240 (decimal) Actuator with the short name 240	
133-134	INT16U	Software version of the connected safety devices (1 word with 2 bytes per safety device)	25 (hex)	Version 2.5
135-136	INT16U	Protocol version of the connected safety devices (1 word with 2 bytes per safety device)	10 (hex)	Version 1.00

Address	Data type	Contents	Sample value in HEX/ Bit field/value range	
137-138	INT16U	Product version of the connected safety devices (1 word with 2 bytes per safety device)	20 (hex)	Version 2.00 (implemented in future)
139-140	INT16U	Number of additional teach-in processes still permitted for the connected Safety Devices (1 word with 2 bytes per Safety Device)	0004 (hex) = 4 (decimal)	
			4 teach-in processes remain	
			Value range 0-8	
141-146	INT16U	Reserved		
147-148	INT16S	Temperature of the connected Safety Devices on Celsius (1 word with 2 bytes per Safety Device)	0012 (hex) = 18 (decimal)	
			18 °C	
149-152	INT16U	Reserved		

Address	Data type	Contents
153-164	INT16U	Reserved

Address	Data type	Contents
165-168	INT16U	Reserved

6.2.4 Output data

Address	Data type	Contents	Sample value in HEX/ Bit field
0	INT8U	States of the configurable inputs/ outputs of the SDD ES PROFIBUS (GPIOO 6 Bit field) Please note: States can only be polled by the PLC when PLC is selected for the safety device in the GPIO mapping of the fieldbus module.	2A (hex) = 00101010 (binary)
			The GPIO 6, 4 and 2 are high

Address	Data type	Contents	Sample value in HEX/ Bit field/value range
1	INT8U	Number of safety device, about which additional information is to be displayed	06 (hex) = 6 (decimal)
			The short name of the actuator, software version, protocol version, product version, teach-in processes still permitted and temperature are to be displayed from safety device 6
2-3	INT16U	Reserved	

Address	Data type	Contents	Sample value in HEX/ Bit field/value range
4-5	INT16U	Reserved	
6-7	INT16U	Reserved	

Byte	Description
8-11	Reserved

Byte	Description
12-15	Reserved

6.3 SDD ES EIP

For communication on the fieldbus, the input and output areas are reserved.

The individual classes are defined as follows.

Class	Application
Class 01H	General device data - acyclical access
Class A2H	Device data of SDD ES EIP and all safety devices - acyclical access
Class A3H	Device data of a safety device - acyclical access
Class A0H	Input data - cyclical access (implicit messaging)
Class A1H	Output data - cyclical access (implicit messaging)

The classes contain attributes. The size of the individual attributes results from the data type.

Data type	Size in Byte
Bool	1
Byte	1
INT	2
UINT	2
USINT	1
Word	2

6.3.1 Class 01H - General device data - acyclical access

Attri-butes	Access type	Data type	Contents	Value for the respective device
1	Get	UINT	Manufacturer	181
2	Get	UINT	Device type	12
3	Get	UINT	Product code established by Pilz	10
7	Get	Short String	Product name	SDD ES EIP

6.3.2 Class A2H - Device data of SDD ES EIP and all safety devices - acyclical access

At-trib-utes	Ac-cess type	Data type	Contents	Example
1	Get	US-INT	Status of SDD ES EIP	02 (hex) = 2 (decimal)
				Running
2	Get	US-INT	Number of safety devices	05 (hex) = 5 (decimal)
				5 safety devices connected

At-tributes	Access type	Data type	Contents	Example
3	Get	Word	Event registered Bit field with 16 Bit for the status of the 16 safety devices. The safety devices are numbered from 0 to 15.	000C (hex) = 0000000000001100 (binary) Safety device 2 and 3 signal an event
4	Get	Word	Actuator enable/safety gate Bit field with 16 Bit for the status of the 16 safety devices	000B (hex) = 0000000000001011 (binary) Safety device 0, 1 and 3 signal enable
5	Get/ Set	Word	Guard locking/lock Bit field with 16 Bit for the status of the 16 safety devices	0003 (hex) = 0000000000000011 (binary) Safety Device 0 and 1 signal guard locking active (get) or they are set active (set)
6	Get	Word	Safety enable (OSSD 1&2) Bit field with 16 Bit for the status of the 16 safety devices	000D (hex) = 0000000000001101 (binary) OSSD 1&2 of safety devices 0, 2 and 3 signal safety enable
7	Get	Byte	States of the inputs of SDD ES EIP Bit field with 6 Bit for the input values of GPIO	07 (hex) = 00000111 (binary) Inputs 1, 2 and 3 are high
8	Get	UINT	Current time (seconds since switching on the SDD ES EIP)	012C (hex) = 300 (decimal) 5 minutes have passed since switching on the fieldbus module
9	Get	USINT	Number of safety device that detects a chain break	2 = Safety device 3 detects a break between safety device 2 and safety device 3 Number of connected safety devices = chain intact
10	Get/ Set	Byte	States of the configurable inputs/outputs of the SDD ES EIP (GPIO, 6 Bit field).	0038 (hex) = 00000000101010 (binary) The GPIO 4, 5 and 6 are high
11	Get/ Set	Byte field	Reserved	
12	Get/ Set	UINT	Reserved	
13	Set	Bool	The SDD ES EIP status is set to 1. The device data is polled again.	01 (hex) = True (Bool) Polling the device data again

At-tributes	Access type	Data type	Contents	Example
14	Get/ Set	UINT	Activation of guard locking via SDD ES EIP	With each change of the counter the intended state is implemented.
15	Get/ Set	Bool	Lock Autoinit Variable for the control of the guard locking activation. The variable is polled only when SDD ES EIP starts, and it is permanently stored internally.	FALSE (bool) = Guard locking activation up to the first command via fieldbus using safety outputs of the safety control system TRUE (Bool) = Guard locking activation only via Safety Device Diagnostics
16	Set	Bool	Deletion of event list The event list is deleted by calling up the following command: Set_Attribute_Single_Service	True (Bool) = Event list is deleted
17-116	Get	USINT	Number of safety device that registers an event	00 (hex) = Safety device 1
		Word	Event	0902 (hex) = See Events of Safety Device Diagnostics [52] (of specific safety devices and all product groups)
		UINT	Time of event	012C (hex) = 5 minutes have passed since switching on the fieldbus module
117-216	Get	USINT	Number of safety device that changed its status	01 (hex) = Safety device 2
		Byte	Change of state	02 (hex) = Actuator changed
		UINT	Time of change of state	0040 (hex) = 64 seconds have passed since switching on the fieldbus module

6.3.3 Class A3H - Device data of a safety device - acyclical access

Per Safety Device there is one instance with the following attributes. The Common Service "Get_Attributes_All" is also implemented.

At-tributes	Access type	Data type	Contents	Example
1	Get	Bool	Actuator is within the response range. The safety gate of the safety device was closed	True (Bool) = The safety gate of the safety device was closed
2	Get/ Set	Bool	Guard locking/lock Bit field with 16 Bit for the status and activation of the 16 safety devices	Get True (Bool) = Guard locking is activated
				Set True (Bool) = Guard locking is to be activated
3	Get	Bool	OSSDs of the safety device have switched to ON state	True (Bool) = OSSD are in ON state
4	Get	Byte field	Serial number of the Safety Device in 3 sections (6 Byte)	Section 2 = 0000 (hex) Section 3 = 8C5E (hex)
				Section 2 + 3 = 00008C5E = 35934 (decimal)
5	Get	Byte field	Order number of the Safety Device in 3 sections (6 Byte)	Section 2 = 0008 (hex) Section 3 = 456B (hex)
				Section 2 + 3 = 0008456B = 542059 (decimal)
6	Get	Byte field	Short name of the actuator of the connected safety devices	0000000044 (hex) = 68
7	Get	Word	Software version of the safety device	0205 (hex) = Vers. 2.5
8	Get	Word	Protocol version of connected safety devices	0100 (hex) = Vers. 1.0
9	Get	Word	Product version	0200 (hex) = Vers. 2.0
10	Get	UINT	Number of additional teach-in processes still permitted for the connected safety devices	8 = 8 teach-in processes remain
11	Get	INT	Temperature of the connected safety devices in Celsius	0012 (hex) = 18 (decimal) = 18 °C
12	Get	Word	Reserved	
13	Get	Byte field	Device-specific events (6 Bytes)	See Table of events and status information per safety device [📖 25]

At-tributes	Access type	Data type	Contents	Example
14-61	Get	Bool	48 attributes correspond to the Bits in attribute 13 for the access to individual bits of attribute 13	See Table of events and status information per safety device [25]

6.3.4 Class A0H - input data - cyclical access (implicit messaging)

The class A0H includes the content of classes A2H and A3H and it is cyclically transferred by the fieldbus module in an individual 116 Byte chain to the controller.

The attributes specified in the table are combined in an individual large Byte chain during the cyclical transfer.

The offset of the attributes specifies the relative address of the relevant attributes.

The chain can be transferred then in an individual connection.

Offset of the attribute	Access type	Data type	Content	Example
0	Get	USINT	Status of SDD ES EIP	0002 (hex) = 2 (decimal)
				Running
1	Get	USINT	Number of safety devices	05 (hex) = 5 (decimal)
				5 safety devices connected
2	Get	WORD	Event registered Bit field with 16 Bit for the status of the 16 safety devices	000C (hex) = 0000000000001100 (binary)
				Safety device 3 and 4 signal events
4	Get	WORD	Actuator enable/safety gate Bit field with 16 Bit for the status of the 16 safety devices	000B (hex) = 0000000000001011 (binary)
				Safety devices 1, 2 and 4 signal enable
6	Get	WORD	Guard locking/lock Bit field with 16 Bit for the status of the 16 safety devices	0003 (hex) = 0000000000000011 (binary)
				Safety devices 1 and 2 signal guard locking active
8	Get	WORD	Safety enable Bit field with 16 Bit for the status of the 16 safety devices	000D (hex) = 0000000000001101 (binary)
				Safety devices 1, 3 and 4 signal safety enable
10	Get	BYTE	States of the inputs of SDD ES EIP Bit field with 6 Bit for the input values of GPIO	07 (hex) = 00000111 (binary)
				Inputs 1, 2 and 3 are high

Offset of the attribute	Access type	Data type	Content	Example
11	Get	UINT	Current time (seconds since switching on the SDD ES EIP)	012C (hex) = 300 (decimal) 5 minutes have passed since switching on the fieldbus module
13	Get	USINT	Number of safety device that detects a chain break	2 = Safety device 3 detects a break between safety device 2 and safety device 3. Number of connected safety devices = chain intact.
14	Get	BYTE	States of the configurable inputs/outputs of the SDD ES EIP (GPIO, 6 Bit field)	38 (hex) = 111000 (binary) The GPIO 4, 5 and 6 are high
15	Get	BYTE	Byte field 3 Bytes, reserved	
18	Get	UINT	Reserved	
20	Get	BYTE	Byte field 6x16 Bytes 6 Bytes each for the status of the 16 safety devices	See Table of events and status information per safety device [📖 25]

6.3.5 Class A1H - output data - cyclical access (implicit messaging)

The class A1H includes the content of classes A2H and it is cyclically transferred by the fieldbus module in an individual 10 Byte chain to the controller.

The attributes specified in the table are combined in an individual large Byte chain during the cyclical transfer.

The offset of the attributes specifies the relative address of the relevant attributes.

The chain can be transferred then in an individual connection.

Offset of the attribute	Access type	Data type	Content	Example
0	Set	BYTE	Here the configurable inputs/outputs (GPIO) can be set to a high or low signal.	38 (hex) = Input/output 4, 5 and 6 are set to high signal.
1	Set	WORD	Activates the guard locking of the safety device	0001 (hex) = Guard locking of the safety device 1 is to be activated
3	Set	UINT	Activation of guard locking via SDD ES EIP	With each change of the counter the intended state is implemented.
5	Set	BYTE	Byte field 3 Bytes, reserved	
8	Set	UINT	Reserved	

6.4 SDD ES PROFINET

6.4.1 GSDML file

INFORMATION

The GSDML file is available on the Internet at www.pilz.de.

6.4.2 Device data

6.4.2.1 Slot division

- Slot 1 - 16: Data of the connected safety devices.

  The sequence of assignment of a safety device to a slot results from the electrical wiring (see System structure [12]).

  – Slot 1 is used for safety device 1,

  – Slot 2 is used for safety device 2, etc.

- Slot 17: For data that concern the connected safety devices and for process data.

6.4.2.2 Submodule input data (slots 1- 16)

Offset	Data type	Contents	Sample value in HEX/Bit field
0	INT16U	Last signalled event from safety device 1, Bit 0 - 15 0 = Bit not set, 1 = Bit set (see Table of events and status information per safety device [25])	040 (hex) = 0000000001000000 (binary) Safety device 1 has taught-in a new actuator and it has to be restarted
2	INT16U	Last event, registered from safety device 1 Bit 16- 31 (see Table of events and status information per safety device [25])	0004 (hex) = 0000000000000010 (binary) Safety device 1 is uniquely coded
4	INT16U	Last event, registered from safety device 1 Bit 32- 47	Reserved
6	INT16U	Number of additional teach-in processes still permitted for the connected safety devices (1 Word with 2 bytes per safety device)	0004 (hex) = 4 (decimal) 4 teach-in processes remain

Offset	Data type	Contents	Sample value in HEX/Bit field
8	INT16S	Temperature of the connected safety devices in Celsius (1 Word with 2 bytes per safety device) (see explanation in Safety devices of all types [📖 24])	0012 (hex) = 18 (decimal)
			18 °C
10	INT16U	Reserved	
12	INT16U	Actuator short name of the connected safety devices (1 Word with 2 bytes per safety device)	00F0 (hex) = 240 (decimal)
			Actuator with the short name 240 The actual name is 1 Byte long. The High Byte is always 0, while the request of the long identifier is not activated.
14	INT16U	Product version of the connected safety devices (1 Word with 2 bytes per safety device)	0002 (hex)
			Version 2.00
16	INT32U	Ident.No	32 Bit number
20	INT8U	Ident.No Byte 4	ASCII character
21	INT8U	Ident.No Byte 5	ASCII character
22	INT32U	Serial number	32 Bit number
26	INT8U	Serial number Byte 4	ASCII character
27	INT8U	Serial number Byte 5	ASCII character
28	INT16U	Reserved	
30	INT16U	Reserved	

6.4.2.3 Submodule output data (slots 1- 16)

Offset	Data type	Contents	Sample value in HEX/Bit field
0	INT8U	Reserved	
1	INT8U	Reserved	
2	INT8U	Reserved	

6.4.3 Global data (slot 17)

6.4.3.1 Submodule input data

Offset	Data type	Contents	Sample value in HEX/Bit field
0	INT8U	Status of SDD ES PROFIBUS	0002 (hex) = 2 (decimal)
			Running
1	INT8U	Number of safety devices	05 (hex) = 5 (decimal)
			5 safety devices connected
2	INT16U	Event signalled (see Events of Safety Device Diagnostics [52])	000C (hex) = 0000000000001100 (binary)
		Bit field with 16 Bit for the status of the 16 safety devices	Safety device 3 and 4 signal events
		The safety devices are numbered from 0 to 15.	Worst case: The Bit is present for 300 ms after the sensor has signalled an event.
4	INT16U	Actuator enable/safety gate	000B (hex) = 0000000000001011 (binary)
		Bit field with 16 Bit for the status of the 16 safety devices	Safety devices 1, 2 and 4 signal enable
		(see Table of events and status information per safety device [25])	
6	INT16U	Guard locking/lock	000C (hex) = 0000000000001100 (binary)
		Bit field with 16 Bit for the status of the 16 safety devices	Safety devices 3 and 4 signal guard locking active
		(see Table of events and status information per safety device [25])	
8	INT16U	Safety enable	000D (hex) = 0000000000001101 (binary)
		Bit field with 16 Bit for the status of the 16 safety devices	Safety devices 1, 3 and 4 signal safety enable
		(see Table of events and status information per safety device [25])	
10	INT8U	States of the inputs of SDD ES PROFINET	07 (hex) = 00000111 (binary)
		Bit field with 6 Bit for the input values of GPIO	Inputs 1, 2 and 3 are high
11	INT16U	Current time (seconds since switching on the SDD ES PROFIBUS)	012C (hex) = 300 (decimal)
			5 minutes have passed since switching on the fieldbus module

Offset	Data type	Contents	Sample value in HEX/Bit field
13	INT8U	Number of safety device that detects a chain break	2 = Safety device 3 detects a break between safety device 2 and safety device 3
14-16	INT8U	Reserved	
17	INT8U	Reserved	

6.4.3.2 Submodule output data

Offset	Data type	Contents	Sample value in HEX/Bit field
0	INT16U	Guard locking/lock	000C (hex) = 0000000000001100 (binary)
			At the safety devices 3 and 4 guard locking is to be activated, all others deactivated
2	INT16U	Activation of guard locking via SDD ES PROFINET	With each change of the counter the intended state (see offset 0) is implemented.
4	INT8U	Lock Autoinit Variable for the control of the guard locking activation. The variable is polled only when SDD ES PROFINET starts, and it is permanently stored internally.	0x5a = Activation 0xc3 = Deactivation
5	INT8U	States of the outputs of SDD ES PROFINET Bit field with 6 Bit for the input values of GPIO	07 (hex) = 00000111 (binary)
			Outputs 1, 2 and 3 are high
6	INT16U	Delete event list	7789 (hex)
			Event list is deleted in transition from value 0x0000 to 0x7789. 1. Write value 0x0000 2. Write value 0x7789 Both values must consist of a least 2 cycles of the SDD ES PROFINET. Then the event list is deleted once.
8-10	INT16U	Reserved	
12-15	INT8U	Reserved	
16	INT16U	Reserved	

6.4.4 PROFINET Records

Record 1

- Max. number of events: 100
- Data length: 5 Byte x 100

Event	Contents
8 Bit	Number of the safety device that registers the event
16 bit	Number of event
16 bit	Time stamp of the signalling of the event

Record 2

- Reserved
- Max. number of data sets: 50
- Data length: 5 Byte x 50

Record 3

- Includes all the previously signalled changes of state. The oldest event is deleted when the max. number is exceeded.
- Max. number of events: 100
- Data length: 5 Byte x 100

Event	Contents
8 Bit	Number of the safety device that registers the event
16 bit	Number of event
16 bit	Time stamp of the signalling of the event

6.4.5 Diagnostics alarms

Sensor registers a general fault or a chain break

Alarm number = 256

When the red LED flashes, the diagnostic alarm is triggered. When the LED goes out again, the diagnostic alarm is withdrawn

Sensor registers error

Alarm number = 257

There is an individual diagnostic alarm for each sensor.

- If a sensor registers one of the relevant events (see list), the alarm is triggered
- When all the relevant events are reset, the alarm is withdrawn.

Event	Relevant	
1	Yes	The actuator has changed state quickly
2	Yes	Safety device in partially operated status
3	Yes	Safety device in partial operation lock status

Event	Relevant	
4	Yes	Supply voltage is at the upper limit of the permitted range (see Technical details for safety device)
5	Yes	Wrong actuator signalled on unique or fully coded versions of a safety device
6	Yes	New actuator has been taught in
7	Yes	Supply voltage above the permitted range (see Technical details for safety device)
8	Yes	Supply voltage below the permitted range (see Technical details for safety device).
18	Yes	Guard locking could not be activated or deactivated. Problem of voltage supply or heavy soling of the actuator. Check the wiring and the actuator.
19	Yes	Locking pin is not in locking position. Activate/deactivate the guard locking again.
8-47	No	various
900	Yes	Fieldbus module and communication with the safety devices have been restarted
901	Yes	Fieldbus module and communication with the safety devices have been restarted.
902	Yes	Chain break before the device that registers the event.
903	Yes	Internal error Change the safety device.
904	Yes	Internal error Check the wiring and switch the voltage off and then on again.
905	Yes	Fault on the safety device output Check the wiring and switch the voltage off and then on again.
906	Yes	Fault on the safety device OSSD1 output Check the wiring and switch the voltage off and then on again.
907	Yes	Fault on the safety device OSSD2 output Check the wiring and switch the voltage off and then on again.
908	Yes	Supply voltage above the permitted range (see Technical details for safety device)
909	Yes	Supply voltage below the permitted range (see Technical details for safety device)
910-915	Reserved	

7 Events of Safety Device Diagnostics

Number	Output text	Meaning/action
0x300	Safety Gate!	The actuator has changed state quickly. ▸ Check and correct the misalignment of switch and actuator and correct if necessary.
0x301	Input ∓	Safety device in partially operated status ▸ Check the wiring.
0x302	Input ∓, Restart	Safety device in partial operation lock status ▸ Check the wiring and switch the voltage off and then on again.
0x303	Power ↑, please ↓	Supply voltage is at the upper limit of the permitted range (see Technical details for safety device) ▸ Ensure the supply voltage corresponds to the Technical details for the connected safety device.
0x304	Power ↓, please ↑	Supply voltage is at the lower limit of the permitted range (see Technical details for safety device) ▸ Ensure the supply voltage corresponds to the Technical details for the connected safety device.
0x305	Actuator ∓	Wrong actuator signalled on unique or fully coded versions of a safety device ▸ Use the actuator stated in the operating manual for the safety device.
0x306	Actuator ♥	New actuator has been taught in ▸ Switch supply voltage off and then on again.
0x320	Lock ∓	Guard locking could not be activated or deactivated. ▸ Check the wiring of the control and the actuator.

Number	Output text	Meaning/action
0x321	Lock ∓	Guard locking is in a not defined position. ▸ Use the escape release to move the guard locking into the open state. Switch the voltage off and then on again
0x900 0x901	**Start UP**	Fieldbus module and communication with the safety devices have been restarted
0x902	Chain ∓	Open circuit ▸ Without stated safety device: Connection from and to the fieldbus module has been interrupted ▸ With stated safety device: Connection before the stated device has been interrupted Check the wiring and switch the voltage off and then on again.
0x903	Please ↔ Device	Internal error ▸ Change the safety device.
0x904	Please Reset	Internal error ▸ Check the wiring and switch the voltage off and then on again.
0x905	OSSD 1&2∓	Fault on the safety device output ▸ Check the wiring and switch the voltage off and then on again.
0x906	OSSD 1∓	Fault on the safety device OSSD1 output ▸ Check the wiring and switch the voltage off and then on again.
0x907	OSSD 2∓	Fault on the safety device OSSD2 output ▸ Check the wiring and switch the voltage off and then on again.
0x908	Power ↑, please ↓	Supply voltage above the permitted range (see Technical details for safety device) ▸ Reduce the supply voltage.

Number	Output text	Meaning/action
0x909	`Power ↓, please ↑`	Supply voltage below the permitted range (see Technical details for safety device) ▸ Increase the supply voltage.
0x90A	`Escape Release`	The escape release was operated. ▸ Reset the safety device to normal operation.

8 Configuration guide

Ask the Pilz customer service for sample programs for PASmulti for SDD ES ETH.

8.1 General implementation assistance

1. Define IP connection

   Establish the connection between the control system and the SDD ES fieldbus module.

   – ETH: Remote IP address range and function codes and further bus parameters,

   – PROFIBUS: Node ID and bus parameter

2. Define variables (see Variable list [56])

   The list in the appendix is a suggestion for a standard definition. Variables may be omitted, depending on which data are used.

   Example ETH with PASmulti:

   I_R4_Safety_Gate AT%I*: WORD;

   bSD1_Safety_Gate: BOOL;

   bSD2_Safety_Gate: BOOL;

   ….

3. Connect variables with the relevant appropriate fieldbus data

   Each variable is connected to a register or an address.

   Example ETH with PASmulti:

   I_R4_Safety_Gate is connected with IP connection 192.168.0.10 register 4

4. In the Program section [60] the variables can now be used to reduce the fieldbus data to individual meanings.

   Example ETH with PASmulti:

   bSD1_Safety_Gate := WORD_TO_BOOL(IN := I_R4_Safety_Gate AND WORD#2#0000_0000_0000_0001);

   bSD2_Safety_Gate := WORD_TO_BOOL(IN := I_R4_Safety_Gate AND WORD#2#0000_0000_0000_0010);

   ….

5. Use variables for further processing (e.g. In the visualisation or in a control program).

8.2 Program example

Features

- The program example is based on the details in Fieldbus modules [27].
- The program example is suitable for a machine with three Safety Devices PSENcode.
- The variable list and the program section can be expanded accordingly to more Safety Devices.
- Variables for process and device data of these three sensors have been declared.
- Variables for establishing the Device History and the Device Events are listed for the last three entries by way of example.
- Command for deleting the event list has been implemented. The data are no longer visible after a service case.
- Manipulations of the sensors are detected by monitoring the short name of the actuator.

8.2.1 Variable list

```
VAR
I_R1_Status_SDD_Communication AT%I*: WORD;
I_R2_Number_SD AT%I*: WORD;
I_R3_Event AT%I*: WORD;
I_R4_Safety_Gate AT%I*: WORD;
I_R5_Lock AT%I*: WORD;
I_R6_OSSD1_2 AT%I*: WORD;
I_R7_GPIO AT%I*: WORD;
I_R8_Time AT%I*: WORD;
I_R9_SD1_Info_Part1 AT%I*: WORD;
I_R10_SD1_Info_Part2 AT%I*: WORD;
I_R12_SD2_Info_Part1 AT%I*: WORD;
I_R13_SD2_Info_Part2 AT%I*: WORD;
I_R15_SD3_Info_Part1 AT%I*: WORD;
I_R16_SD3_Info_Part2 AT%I*: WORD;

I_R57_SD_Chain AT%I*: WORD;

I_R81_SD1_Temperature AT%I*: WORD;
I_R82_SD2_Temperature AT%I*: WORD;
I_R83_SD3_Temperature AT%I*: WORD;

I_R129_SD1_Actuator_Short_ID AT%I*: WORD;
I_R130_SD2_Actuator_Short_ID AT%I*: WORD;
I_R131_SD3_Actuator_Short_ID AT%I*: WORD;

I_R161_SD1_Ident_Number_Part2 AT%I*: WORD;
I_R162_SD2_Ident_Number_Part2 AT%I*: WORD;
I_R163_SD3_Ident_Number_Part2 AT%I*: WORD;

I_R177_SD1_Ident_Number_Part3 AT%I*: WORD;
```

```
I_R178_SD2_Ident_Number_Part3 AT%I*: WORD;
I_R179_SD3_Ident_Number_Part3 AT%I*: WORD;

I_R209_SD1_Serial_Number_Part2 AT%I*: WORD;
I_R210_SD2_Serial_Number_Part2 AT%I*: WORD;
I_R211_SD3_Serial_Number_Part2 AT%I*: WORD;

I_R225_SD1_Serial_Number_Part3 AT%I*: WORD;
I_R226_SD2_Serial_Number_Part3 AT%I*: WORD;
I_R227_SD3_Serial_Number_Part3 AT%I*: WORD;

I_R273_SD1_Product_Version AT%I*: WORD;
I_R274_SD2_Product_Version AT%I*: WORD;
I_R275_SD3_Product_Version AT%I*: WORD;
//The variable List can be further extended up to Register 320

//The variables represent the device event and device history list.
The list can be extended up by time values (even registers) and up
to 150 values
I_R1001_Event1 AT%I*: WORD;
I_R1003_Event2 AT%I*: WORD;
I_R1005_Event3 AT%I*: WORD;

I_R1501_Status1 AT%I*: WORD;
I_R1503_Status2 AT%I*: WORD;
I_R1505_Status3 AT%I*: WORD;

I_R1502_Time_Status1 AT%I*: WORD;
I_R1504_Time_Status2 AT%I*: WORD;
I_R1506_Time_Status3 AT%I*: WORD;

//Mlock System Locking
Q_R2_LockUnlock AT%Q* :WORD;
Q_R3_LockUnlock_Counter AT%Q* :Word;
// Mlock System Locking by SDD
Q_R4_Init_LockUnlock AT%Q* :WORD;

//Input variables are for the SDD overview
iSDD_Status: INT;
iSD_Number: INT;
iSD_Chain: INT;
bClear_Defeat: BOOL;
iTime: INT;

//Input Variables concerning the Safety Device - extendable up to
16 Safety Devices
bSD1: BOOL;
bSD1_Event: BOOL;
bSD1_Safety_Gate: BOOL;
bSD1_Lock: BOOL;
```

```
bSD1_OSSD1_2: BOOL;
bSD1_Ready: BOOL;
bSD1_Safety_Input1: BOOL;
bSD1_Safety_Input2: BOOL;
bSD1_Safety_OSSD1: BOOL;
bSD1_Safety_OSSD2: BOOL;
iSD1_Coding_Type: INT;
iSD1_Temperature: INT;
iSD1_Actuator_Short_ID: INT;
bSD1_Actuator_Short_ID_Check: BOOL;
iSD1_Ident_Number: DINT;
iSD1_Serial_Number: DINT;
iSD1_Product_Version: INT;

bSD2: BOOL;
bSD2_Event: BOOL;
bSD2_Safety_Gate: BOOL;
bSD2_Lock: BOOL;
bSD2_OSSD1_2: BOOL;
bSD2_Ready: BOOL;
bSD2_Safety_Input1: BOOL;
bSD2_Safety_Input2: BOOL;
bSD2_Safety_OSSD1: BOOL;
bSD2_Safety_OSSD2: BOOL;
iSD2_Coding_Type: INT;
iSD2_Temperature: INT;
iSD2_Actuator_Short_ID: INT;
bSD2_Actuator_Short_ID_Check: BOOL;
iSD2_Ident_Number: DINT;
iSD2_Serial_Number: DINT;
iSD2_Product_Version: INT;

bSD3: BOOL;
bSD3_Event: BOOL;
bSD3_Safety_Gate: BOOL;
bSD3_Lock: BOOL;
bSD3_OSSD1_2: BOOL;
bSD3_Ready: BOOL;
bSD3_Safety_Input1: BOOL;
bSD3_Safety_Input2: BOOL;
bSD3_Safety_OSSD1: BOOL;
bSD3_Safety_OSSD2: BOOL;
iSD3_Coding_Type: INT;
iSD3_Temperature: INT;
iSD3_Actuator_Short_ID: INT;
bSD3_Actuator_Short_ID_Check: BOOL;
iSD3_Ident_Number: DINT;
iSD3_Serial_Number: DINT;
iSD3_Product_Version: INT;
```

```
//Input variable for SDD GPIOs
bGPIO1_Input: BOOL;
bGPIO2_Input: BOOL;
bGPIO3_Input: BOOL;
bGPIO4_Input: BOOL;
bGPIO5_Input: BOOL;
bGPIO6_Input: BOOL;

//Variables for the Device Event and Device History
iSD_Event1_List: INT;
iSD_Event2_List: INT;
iSD_Event3_List: INT;

iEvent1_List: INT;
iEvent2_List: INT;
iEvent3_List: INT;

iSD_Status1_List: INT;
iSD_Status2_List: INT;
iSD_Status3_List: INT;

iStatus1_List: INT;
iStatus2_List: INT;
iStatus3_List: INT;

iTime_Status1_List: INT;
iTime_Status2_List: INT;
iTime_Status3_List: INT;

//Each Output Register has a variable
Q_R1_GPIO AT%Q* :WORD;
Q_R1004_Event_LIST_CLEAR AT%Q* :WORD;

//Output variable for SDD overview
Trigger:R_TRIG;
trigger_bEvent_List_clear: BOOL;
bEvent_List_clear: BOOL;

//Output variables for SDD GPIOs
bGPIO1_Output: BOOL;
bGPIO2_Output: BOOL;
bGPIO3_Output: BOOL;
bGPIO4_Output: BOOL;
bGPIO5_Output: BOOL;
bGPIO6_Output: BOOL;
```

// MLock System Device

bSD1_Set_LockUnlock: **BOOL**;

bSD2_Set_LockUnlock: **BOOL**;

```
bSD3_Set_LockUnlock: BOOL;
bSD4_Set_LockUnlock: BOOL;
bSD5_Set_LockUnlock: BOOL;
bSD6_Set_LockUnlock: BOOL;

END_VAR

VAR RETAIN PERSISTENT
//Variables which are availbe also after Power Reset. Needed for
Actuator Defeat Monitoring
iSD1_Actuator_Short_ID_Reference: INT;
iSD2_Actuator_Short_ID_Reference: INT;
iSD3_Actuator_Short_ID_Reference: INT;

bSD1_Actuator_NoDefeat_Detection: BOOL;
bSD2_Actuator_NoDefeat_Detection: BOOL;
bSD3_Actuator_NoDefeat_Detection: BOOL;

END_VAR
```

8.2.2 Program section

```
//Variables are mapped to register variable
//Variables for the SDD overview
iSDD_Status := WORD_TO_INT(IN := I_R1_Status_SDD_Communication);
iSD_Number := WORD_TO_INT(IN := I_R2_Number_SD);
iTime := WORD_TO_INT(IN := I_R8_Time);
iSD_Chain := WORD_TO_INT(IN := I_R57_SD_Chain);

//Safety Device specific variables
bSD1:= (iSD_Number >= INT#1);
bSD1_Event := WORD_TO_BOOL(IN := I_R3_Event AND
WORD#2#0000_0000_0000_0001);
bSD1_Safety_Gate := WORD_TO_BOOL(IN := I_R4_Safety_Gate AND
WORD#2#0000_0000_0000_0001);
bSD1_Lock := WORD_TO_BOOL(IN := I_R5_Lock AND
WORD#2#0000_0000_0000_0001);
bSD1_OSSD1_2 := WORD_TO_BOOL(IN := I_R6_OSSD1_2 AND
WORD#2#0000_0000_0000_0001);
bSD1_Ready := WORD_TO_BOOL(IN := I_R9_SD1_Info_Part1 AND
WORD#2#0000_0010_0000_0000);
bSD1_Safety_Input1 := WORD_TO_BOOL(IN := I_R9_SD1_Info_Part1 AND
WORD#2#0000_1000_0000_0000);
bSD1_Safety_Input2 := WORD_TO_BOOL(IN := I_R9_SD1_Info_Part1 AND
WORD#2#0001_0000_0000_0000);
bSD1_Safety_OSSD1 := WORD_TO_BOOL(IN := I_R9_SD1_Info_Part1 AND
WORD#2#0010_0000_0000_0000);
```

```
bSD1_Safety_OSSD2 := WORD_TO_BOOL(IN := I_R9_SD1_Info_Part1 AND
WORD#2#0100_0000_0000_0000);
IF WORD_TO_BOOL (IN := (I_R9_SD1_Info_Part1 AND
WORD#2#1000_0000_0000_0000)) THEN iSD1_Coding_Type :=INT#1;
ELSIF WORD_TO_BOOL (IN := (I_R10_SD1_Info_Part2 AND
WORD#2#0000_0000_0000_0001)) THEN iSD1_Coding_Type :=INT#2;
ELSIF WORD_TO_BOOL (IN := (I_R10_SD1_Info_Part2 AND
WORD#2#0000_0000_0000_0010)) THEN iSD1_Coding_Type :=INT#3;
END_IF;
iSD1_Temperature := WORD_TO_INT(IN := I_R81_SD1_Temperature);
iSD1_Ident_Number := WORD_TO_INT (IN := SHL(IN:= DWORD_TO_WORD
(IN := I_R161_SD1_Ident_Number_Part2), N:=16) OR
I_R177_SD1_Ident_Number_Part3);
iSD1_Serial_Number := WORD_TO_INT (IN := SHL(IN:= DWORD_TO_WORD
(IN := I_R209_SD1_Serial_Number_Part2), N:=16) OR
I_R225_SD1_Serial_Number_Part3);
iSD1_Product_Version := (WORD_TO_INT (IN := SHR(IN :=
I_R273_SD1_Product_Version, N:=8) AND WORD#16#00FF) + WORD_TO_INT
(IN:= I_R273_SD1_Product_Version)*INT#100) / INT#100;
//Monitoring Defeat - Checking, Tracking and Clearing
iSD1_Actuator_Short_ID := WORD_TO_INT(IN := I_R129_SD1_Actu-
ator_Short_ID);
bSD1_Actuator_Short_ID_Check := (iSD1_Actuator_Short_ID = iSD1_Ac-
tuator_Short_ID_Reference) OR (iSD1_Actuator_Short_ID = INT#0);
IF bSD1_Actuator_Short_ID_Check = FALSE THEN bSD1_Actuator_NoDe-
feat_Detection := FALSE;
END_IF;
IF bClear_Defeat THEN bSD1_Actuator_NoDefeat_Detection := TRUE;
END_IF;

bSD2:= (iSD_Number >= INT#2);
bSD2_Event := WORD_TO_BOOL(IN := I_R3_Event AND
WORD#2#0000_0000_0000_0010);
bSD2_Safety_Gate := WORD_TO_BOOL(IN := I_R4_Safety_Gate AND
WORD#2#0000_0000_0000_0010);
bSD2_Lock := WORD_TO_BOOL(IN := I_R5_Lock AND
WORD#2#0000_0000_0000_0010);
bSD2_OSSD1_2 := WORD_TO_BOOL(IN := I_R6_OSSD1_2 AND
WORD#2#0000_0000_0000_0010);
bSD2_Ready := WORD_TO_BOOL(IN := I_R12_SD2_Info_Part1 AND
WORD#2#0000_0010_0000_0000);
bSD2_Safety_Input1 := WORD_TO_BOOL(IN := I_R12_SD2_Info_Part1 AND
WORD#2#0000_1000_0000_0000);
bSD2_Safety_Input2 := WORD_TO_BOOL(IN := I_R12_SD2_Info_Part1 AND
WORD#2#0001_0000_0000_0000);
bSD2_Safety_OSSD1 := WORD_TO_BOOL(IN := I_R12_SD2_Info_Part1 AND
WORD#2#0010_0000_0000_0000);
bSD2_Safety_OSSD2 := WORD_TO_BOOL(IN := I_R12_SD2_Info_Part1 AND
WORD#2#0100_0000_0000_0000);
```

```
IF WORD_TO_BOOL (IN := (I_R12_SD2_Info_Part1 AND
WORD#2#1000_0000_0000_0000)) THEN iSD2_Coding_Type :=INT#1;
ELSIF WORD_TO_BOOL (IN := (I_R13_SD2_Info_Part2 AND
WORD#2#0000_0000_0000_0001)) THEN iSD2_Coding_Type :=INT#2;
ELSIF WORD_TO_BOOL (IN := (I_R13_SD2_Info_Part2 AND
WORD#2#0000_0000_0000_0010)) THEN iSD2_Coding_Type :=INT#3;
END_IF;
iSD2_Temperature := WORD_TO_INT(IN := I_R82_SD2_Temperature);
iSD2_Ident_Number := DWORD_TO_DINT (IN := SHL(IN:= WORD_TO_DWORD
(IN := I_R162_SD2_Ident_Number_Part2), N:=16) OR
I_R178_SD2_Ident_Number_Part3);
iSD2_Serial_Number := DWORD_TO_DINT (IN := SHL(IN:= WORD_TO_DWORD
(IN := I_R210_SD2_Serial_Number_Part2), N:=16) OR
I_R226_SD2_Serial_Number_Part3);
iSD2_Product_Version := (WORD_TO_INT (IN := SHR(IN :=
I_R274_SD2_Product_Version, N:=8) AND WORD#16#00FF) + WORD_TO_INT
(IN:= I_R274_SD2_Product_Version)*INT#100) / INT#100;
//Monitoring Defeat - Checking, Tracking and Clearing
iSD2_Actuator_Short_ID := WORD_TO_INT(IN := I_R130_SD2_Actu-
ator_Short_ID);
bSD2_Actuator_Short_ID_Check := (iSD2_Actuator_Short_ID = iSD2_Ac-
tuator_Short_ID_Reference) OR (iSD2_Actuator_Short_ID = INT#0);
IF bSD2_Actuator_Short_ID_Check = FALSE THEN bSD2_Actuator_NoDe-
feat_Detection := FALSE;
END_IF;
IF bClear_Defeat THEN bSD2_Actuator_NoDefeat_Detection := TRUE;
END_IF;

bSD3:= (iSD_Number >= INT#3);
bSD3_Event := WORD_TO_BOOL(IN := I_R3_Event AND
WORD#2#0000_0000_0000_0100);
bSD3_Safety_Gate := WORD_TO_BOOL(IN := I_R4_Safety_Gate AND
WORD#2#0000_0000_0000_0100);
bSD3_Lock := WORD_TO_BOOL(IN := I_R5_Lock AND
WORD#2#0000_0000_0000_0100);
bSD3_OSSD1_2 := WORD_TO_BOOL(IN := I_R6_OSSD1_2 AND
WORD#2#0000_0000_0000_0100);
bSD3_Ready := WORD_TO_BOOL(IN := I_R15_SD3_Info_Part1 AND
WORD#2#0000_0010_0000_0000);
bSD3_Safety_Input1 := WORD_TO_BOOL(IN := I_R15_SD3_Info_Part1 AND
WORD#2#0000_1000_0000_0000);
bSD3_Safety_Input2 := WORD_TO_BOOL(IN := I_R15_SD3_Info_Part1 AND
WORD#2#0001_0000_0000_0000);
bSD3_Safety_OSSD1 := WORD_TO_BOOL(IN := I_R15_SD3_Info_Part1 AND
WORD#2#0010_0000_0000_0000);
bSD3_Safety_OSSD2 := WORD_TO_BOOL(IN := I_R15_SD3_Info_Part1 AND
WORD#2#0100_0000_0000_0000);
IF WORD_TO_BOOL (IN := (I_R15_SD3_Info_Part1 AND
WORD#2#1000_0000_0000_0000)) THEN iSD3_Coding_Type :=INT#1;
```

```
ELSIF WORD_TO_BOOL (IN := (I_R16_SD3_Info_Part2 AND
WORD#2#0000_0000_0000_0001)) THEN iSD3_Coding_Type :=INT#2;
ELSIF WORD_TO_BOOL (IN := (I_R16_SD3_Info_Part2 AND
WORD#2#0000_0000_0000_0010)) THEN iSD3_Coding_Type :=INT#3;
END_IF;
iSD3_Temperature := WORD_TO_INT(IN := I_R83_SD3_Temperature);
iSD3_Ident_Number := DWORD_TO_DINT (IN := SHL(IN:= WORD_TO_DWORD
(IN := I_R163_SD3_Ident_Number_Part2), N:=16) OR
I_R179_SD3_Ident_Number_Part3);
iSD3_Serial_Number :=DWORD_TO_DINT (IN := SHL(IN:= WORD_TO_DWORD
(IN := I_R211_SD3_Serial_Number_Part2), N:=16) OR
I_R227_SD3_Serial_Number_Part3);
iSD3_Product_Version := (WORD_TO_INT (IN := SHR(IN :=
I_R275_SD3_Product_Version, N:=8) AND WORD#16#00FF) + WORD_TO_INT
(IN:= I_R275_SD3_Product_Version)*INT#100) / INT#100;
//Monitoring Defeat - Checking, Tracking and Clearing
iSD3_Actuator_Short_ID := WORD_TO_INT(IN := I_R131_SD3_Actu-
ator_Short_ID);
bSD3_Actuator_Short_ID_Check := (iSD3_Actuator_Short_ID = iSD3_Ac-
tuator_Short_ID_Reference) OR (iSD3_Actuator_Short_ID = INT#0);
IF bSD3_Actuator_Short_ID_Check = FALSE THEN bSD3_Actuator_NoDe-
feat_Detection := FALSE;
END_IF;
IF bClear_Defeat THEN bSD3_Actuator_NoDefeat_Detection := TRUE;
END_IF;

//Reading local input variables
bGPIO1_Input := WORD_TO_BOOL(IN := I_R7_GPIO AND
WORD#2#0000_0000_0000_0001);
bGPIO2_Input := WORD_TO_BOOL(IN := I_R7_GPIO AND
WORD#2#0000_0000_0000_0010);
bGPIO3_Input := WORD_TO_BOOL(IN := I_R7_GPIO AND
WORD#2#0000_0000_0000_0100);
bGPIO4_Input := WORD_TO_BOOL(IN := I_R7_GPIO AND
WORD#2#0000_0000_0000_1000);
bGPIO5_Input := WORD_TO_BOOL(IN := I_R7_GPIO AND
WORD#2#0000_0000_0001_0000);
bGPIO6_Input := WORD_TO_BOOL(IN := I_R7_GPIO AND
WORD#2#0000_0000_0010_0000);

//Writing local output variables
Q_R1_GPIO:= BOOL_TO_WORD(IN := bGPIO1_Output)OR
SHL(IN:= (BOOL_To_Word(IN := bGPIO2_Output)),N := 1)OR
SHL(IN:= (BOOL_To_Word(IN := bGPIO3_Output)),N := 2)OR
SHL(IN:= (BOOL_To_Word(IN := bGPIO4_Output)),N := 3)OR
SHL(IN:= (BOOL_To_Word(IN := bGPIO5_Output)),N := 4)OR
SHL(IN:= (BOOL_To_Word(IN := bGPIO6_Output)),N := 5);

// Writing single lock/unlock variables to output register 2 -> SDD
Q_R2_LockUnlock := BOOL_TO_WORD(IN := bSD1_Set_LockUnlock)OR
```

```
SHL(IN:= (BOOL_To_Word(IN := bSD2_Set_LockUnlock)),N := 1)OR
SHL(IN:= (BOOL_To_Word(IN := bSD3_Set_LockUnlock)),N := 2)OR
SHL(IN:= (BOOL_To_Word(IN := bSD4_Set_LockUnlock)),N := 3)OR
SHL(IN:= (BOOL_To_Word(IN := bSD5_Set_LockUnlock)),N := 4)OR
SHL(IN:= (BOOL_To_Word(IN := bSD6_Set_LockUnlock)),N := 5);

// Each time a single or all lock/unlock variables are set to high
or low counter counts up (+1)
IF Q_R2_LockUnlock <> bR2_LockUnlock_Temp THEN iCounter_Value :=
iCounter_Value +INT#1;
ELSE iCounter_Value := iCounter_Value;
END_IF;

// Mirroring status of Lock/Unlock register to temporary variable
for comparison in next cycle (If condition above)
bR2_LockUnlock_Temp:=Q_R2_LockUnlock;

//Event and History List
//Clearing Event List
Trigger(CLK := bEvent_List_clear,Q => trigger_bEvent_List_clear);
IF trigger_bEvent_List_clear THEN
Q_R1004_Event_List_clear :=WORD#16#7789;
ELSE Q_R1004_Event_List_clear :=WORD#16#0;
END_IF;

//Event List variables
iEvent1_List := WORD_TO_INT(IN := I_R1001_Event1 AND WORD#16#0FFF);
iEvent2_List := WORD_TO_INT(IN := I_R1003_Event2 AND WORD#16#0FFF);
iEvent3_List := WORD_TO_INT(IN := I_R1005_Event3 AND WORD#16#0FFF);

iStatus1_List := WORD_TO_INT(IN := I_R1501_Status1 AND
WORD#16#0FFF);
iStatus2_List := WORD_TO_INT(IN := I_R1503_Status2 AND
WORD#16#0FFF);
iStatus3_List := WORD_TO_INT(IN := I_R1505_Status3 AND
WORD#16#0FFF);

iTime_Status1_List := WORD_TO_INT(IN := I_R1502_Time_Status1);
iTime_Status2_List := WORD_TO_INT(IN := I_R1504_Time_Status2);
iTime_Status3_List := WORD_TO_INT(IN := I_R1506_Time_Status3);

//The number of the Safety Device is corrected because of Fault of
wiring to and from SDD ES modul is added
IF (iEvent1_List =INT#2305 or iEvent1_List =INT#2304) THEN
iSD_Event1_List := INT#0;
ELSIF (iEvent1_List =INT#2306) AND (INT#1 + WORD_TO_INT(IN := (SHR
(IN := I_R1001_Event1, N:=12))) > iSD_Number) THEN
iSD_Event1_List := INT#0;
ELSE iSD_Event1_List := INT#1 + WORD_TO_INT(IN := (SHR (IN :=
I_R1001_Event1, N:=12)));
```

```
END_IF;
IF (iEvent2_List =INT#2305 OR iEvent2_List =INT#2304) THEN
iSD_Event2_List := INT#0;
ELSIF (iEvent2_List =INT#2306) AND (INT#1 + WORD_TO_INT(IN := (SHR
(IN := I_R1003_Event2, N:=12))) > iSD_Number) THEN
iSD_Event2_List := INT#0;
ELSE iSD_Event2_List := INT#1 + WORD_TO_INT(IN := (SHR (IN :=
I_R1003_Event2, N:=12)));
END_IF;
IF (iEvent3_List =INT#2305 OR iEvent3_List =INT#2304) THEN
iSD_Event3_List := INT#0;
ELSIF (iEvent3_List =INT#2306) AND (INT#1 + WORD_TO_INT(IN := (SHR
(IN := I_R1005_Event3, N:=12)))) > iSD_Number THEN
iSD_Event3_List := INT#0;
ELSE iSD_Event3_List := INT#1 + WORD_TO_INT(IN := (SHR (IN :=
I_R1005_Event3, N:=12)));
END_IF;

iSD_Status1_List := INT#1 + WORD_TO_INT(IN := (SHR (IN :=
I_R1501_Status1, N:=12)));
iSD_Status2_List := INT#1 + WORD_TO_INT(IN := (SHR (IN :=
I_R1503_Status2, N:=12)));
iSD_Status3_List := INT#1 + WORD_TO_INT(IN := (SHR (IN :=
I_R1505_Status3, N:=12)));

END_PROGRAM
```

Support

Technical support is available from Pilz round the clock.

Americas
Brazil
+55 11 97569-2804
Canada
+1 888 315 7459
Mexico
+52 55 5572 1300
USA (toll-free)
+1 877-PILZUSA (745-9872)

Asia
China
+86 21 60880878-216
Japan
+81 45 471-2281
South Korea
+82 31 778 3300

Australia and Oceania
Australia
+61 3 95600621
New Zealand
+64 9 6345350

Europe
Austria
+43 1 7986263-0
Belgium, Luxembourg
+32 9 3217570
France
+33 3 88104003
Germany
+49 711 3409-444
Ireland
+353 21 4804983
Italy, Malta
+39 0362 1826711
Scandinavia
+45 74436332
Spain
+34 938497433
Switzerland
+41 62 88979-32
The Netherlands
+31 347 320477
Turkey
+90 216 5775552
United Kingdom
+44 1536 462203

You can reach our international hotline on:
+49 711 3409-222
support@pilz.com

Pilz develops environmentally-friendly products using ecological materials and energy-saving technologies. Offices and production facilities are ecologically designed, environmentally-aware and energy-saving. So Pilz offers sustainability, plus the security of using energy-efficient products and environmentally-friendly solutions.

BLUEECOMPETENCE Alliance Member
Partner of the Engineering Industry Sustainability Initiative

BRANDS OF THE CENTURY PILZ 2019 GERMAN STANDARDS

Quality Management ISO 9001:2015 REG. NO. 2462
Environmental Management ISO 14001 REG. NO. 2462
Energy Management ISO 50001 REG. NO. 2462

1003827-EN-08, 2021-03 Printed in Germany

We are represented internationally. Please refer to our homepage www.pilz.com for further details or contact our headquarters.

Headquarters: Pilz GmbH & Co. KG, Felix-Wankel-Straße 2, 73760 Ostfildern, Germany
Telephone: +49 711 3409-0, Telefax: +49 711 3409-133, E-Mail: info@pilz.com, Internet: www.pilz.com

PILZ
THE SPIRIT OF SAFETY