Original Programming Manual
SmartPLC SafeLine AC4S
with fieldbus interfaces

AC402S
AC412S
AC422S
AC432S

for ISO 13849 up to PL e
for IEC 61508 SIL 3 and 62061 up to SIL CL 3

Firmware release: 4.2.5
CODESYS Development System: 3.5.9.73

English

7391150/00 12/2017

Contents

1 Preliminary note 6

1.1 Legal and copyright information ... 6
1.2 Purpose of the document ... 6
1.3 Symbols and styles used ... 7
1.4 Overview: User documentation for AC4S ... 7
1.5 Overview: CODESYS documentation of 3S ... 8
1.6 Modification history ... 8

2 Safety instructions 9

2.1 General safety instructions ... 9
2.2 Required background knowledge ... 9
2.3 Warnings used ... 10

3 System requirements 11

3.1 Hardware ... 11
3.2 Software ... 11
3.3 Licensing ... 11

4 Installation 12

4.1 CODESYS programming software ... 12
4.1.1 Install CODESYS Development System ... 12
4.2 ifm AS-i Package ... 13
4.2.1 Components of the ifm package ... 13
4.2.2 Install the ifm package ... 14
4.2.3 Uninstall the ifm package ... 14

5 Getting started 15

5.1 Start CODESYS ... 15
5.2 Create CODESYS project ... 16
5.2.1 Create new project with SmartPLC SafeLine AC4S ... 17
5.2.2 Overview: Project structure with SmartPLC SafeLine AC4S ... 19
5.3 Use CODESYS online help ... 20
5.4 Create user account ... 20
5.5 Configure programming interface ... 21
5.5.1 Set communication path of standard PLC ... 21
5.5.2 Set communication path of the fail-safe PLC ... 22

6 System configuration 23

6.1 Configure standard PLC ... 23
6.2 Configure fail-safe PLC ... 24
6.3 Configure safe AS-i slaves ... 25
6.3.1 Remarks ... 25
6.3.2 Add safe AS-i slaves to the project ... 27
6.3.3 Remove safe AS-i slave from project ... 31
6.4 Configure safe devices at local I/O interface ... 32
6.4.1 Remarks ... 32
6.4.2 Add safe local device to project ... 34
6.4.3 Create test signal at local output ... 39

6.4.4 Remove safe local device from project 39
6.5 Configure standard devices at local I/O interface 39
6.6 Change IP settings 40
6.7 Add additional device to the project 41
6.8 Extend fieldbus functionality 42
6.8.1 Available fieldbus stacks 42
6.8.2 Use EtherCAT master 43
6.8.3 Add fieldbus stack 46

7 Programming 47

7.1 Objects of a standard PLC application 47
7.2 Create standard PLC application 48
7.2.1 Use remanent variables 48
7.2.2 Supported programming languages 49
7.2.3 Change system time of the device 49
7.3 Access standard input and output data 50
7.3.1 Options to access input and output data 50
7.3.2 Validity of the interface data 51
7.3.3 Process data of the AS-i slaves 52
7.3.4 Fieldbus data 54
7.3.5 Process data of the non-safe local devices 56
7.4 Use functions of the ifm package 57
7.4.1 Control interface of the ifm function blocks 57
7.4.2 Configure system 59
7.4.3 Configure AS-i master 59
7.4.4 Configure AS-i slaves 59
7.4.5 Manage AS-i network 60
7.4.6 Send commands to the system and the AS-i master 63
7.5 Use visualisations 64
7.5.1 Supported visualisation types 64
7.5.2 Add visualisation to a project 65
7.5.3 Create a visualisation 66
7.5.4 Configure visualisation 67
7.6 Cross communication 68
7.6.1 Use network variables 68
7.7 Configure task processing 70
7.7.1 Configure main task 70
7.7.2 Set parameters for visualisation task 70

8 Programming of the fail-safe PLC 71

8.1 Objects of a safe application 72
8.2 Create safe application 73
8.2.1 Supported program organization units (POUs) 74
8.2.2 Available safety libraries 74
8.2.3 Supported programming languages 75
8.2.4 Minimum number of AS-i slaves 75
8.2.5 Variable declaration with data transfer between standard and safety area 76
8.2.6 Available memory space 76
8.2.7 Pin safe project 76
8.3 Use safety functions of the ifm AS-i package 77
8.3.1 Safe AS-i slaves: Reset logical device 77
8.3.2 Safe AS-i slaves: Provide diagnostic information 78
8.3.3 Safe AS-i slaves: Transmit help signals HSI_1 and HSI_2 to safe AS-i output modules 78
8.3.4 Local I/O interface: Reset logical device 78
8.3.5 Local I/O interface: Provide diagnostic information 79
8.3.6 Local I/O interface: Access non-safe inputs/outputs 79
8.3.7 PLCopen Safety: Provide diagnostic information 79
8.4 Use safety functions of CODESYS 80
8.4.1 SafetyStandard library 80

8.4.2 SafetyPLCopen library ... 81
8.4.3 SafetyFSoEMaster library ... 82
8.5 Data exchange between standard PLC and fail-safe PLC ... 83
8.5.1 Remarks about the use of exchange variables ... 83
8.5.2 Use the factory-set replacement variables ... 84
8.5.3 Create additional objects for data exchange ... 86
8.5.4 Set interval of the data exchange ... 86
8.6 Safe cross communication ... 87
8.6.1 Set up fail-safe network variables ... 88
8.6.2 Use fail-safe network variables in the application ... 90
8.7 Safe fieldbus communication ... 91
8.7.1 Configure FSoE connection ... 91
8.7.2 Notes on the FSoEMaster ... 92
8.8 Set process safety time ... 93
8.8.1 Process safety time for the SmartPLC SafeLine AC4S ... 93
8.8.2 Set variable components of the response time ... 99
8.9 Configure safety task ... 100
8.9.1 Set cycle time of the fail-safe PLC ... 100

9 Operation 101

9.1 Transfer CODESYS project to device ... 101
9.1.1 Download standard application to SmartPLC SafeLine AC4S ... 102
9.1.2 Load safe application to SmartPLC SafeLine AC4S ... 103
9.1.3 Note: Projects with network variables ... 104
9.1.4 Teach code sequences of the safe AS-i slaves ... 104
9.1.5 Delete boot application via SD card ... 105
9.2 Operating states SmartPLC SafeLine AC4S ... 106
9.2.1 standard PLC ... 106
9.2.2 Fail-safe PLC ... 108
9.3 Reset ... 111
9.3.1 Standard PLC ... 111
9.3.2 Fail-safe PLC ... 113
9.4 Display web visualisation ... 114

10 Appendix 115

10.1 Library ACnnnn_Utils.library ... 116
10.1.1 Overview: AS-i functions (FB_ASi) ... 117
10.1.2 Overview: System functions (FB_System) ... 163
10.1.3 Enumeration types and complex variables ... 170
10.2 Library ACnnnn_SYS_CMD.library ... 178
10.2.1 ACnnnn_SysCmd ... 178
10.3 Library SF_IO.library ... 186
10.3.1 CtrlASi_InSlave ... 187
10.3.2 CtrlASi_OutSlave ... 189
10.3.3 CtrlASi_ResetAllSlaves ... 192
10.3.4 CtrlLocalInputs ... 194
10.3.5 Ctrl_SetDiagInfo ... 196
10.3.6 GetLocalInput ... 198
10.3.7 SetLocalOutput ... 199
10.4 Library SF_LogicalInterfaces.library ... 201
10.4.1 Remarks about logical devices ... 202
10.4.2 Logical devices for safe AS-i slaves ... 205
10.4.3 Logical devices for the local I/O interface ... 235

10.5 Error codes: Safe AS-i slaves 276
10.6 Error codes: Local I/O interface 278

11 Index 279

12 ifm weltweit • ifm worldwide • ifm à l'échelle internationale 283

1 Preliminary note

Contents

Legal and copyright information 6
Purpose of the document 6
Symbols and styles used 7
Overview: User documentation for AC4S 7
Overview: CODESYS documentation of 3S 8
Modification history 8

14801

1.1 Legal and copyright information

1631

- All product names, pictures, companies or other brands used on our pages are the property of the respective rights owners:
- AS-i is the property of the AS-International Association, (→ www.as-interface.net)
- CAN is the property of the CiA (CAN in Automation e.V.), Germany (→ www.can-cia.org)
- CODESYS™ is the property of the 3S – Smart Software Solutions GmbH, Germany (→ www.codesys.com)
- DeviceNet™ is the property of the ODVA™ (Open DeviceNet Vendor Association), USA (→ www.odva.org)
- EtherNet/IP® is the property of the →ODVA™
- EtherCAT® is a registered trade mark and patented technology, licensed by Beckhoff Automation GmbH, Germany
- IO-Link® (→ www.io-link.com) is the property of the →PROFIBUS Nutzerorganisation e.V., Germany
- ISOBUS is the property of the AEF – Agricultural Industry Electronics Foundation e.V., Deutschland (→ www.aef-online.org)
- Microsoft® is the property of the Microsoft Corporation, USA (→ www.microsoft.com)
- PROFIBUS® is the property of the PROFIBUS Nutzerorganisation e.V., Germany (→ www.profibus.com)
- PROFINET® is the property of the →PROFIBUS Nutzerorganisation e.V., Germany
- Windows® is the property of the →Microsoft Corporation, USA

1.2 Purpose of the document

21675

This document applies to the following devices of the "SmartSPS SafeLine AC4S" product family:

- SmartPLC SafeLine AC4S with Profinet device interface (AC402S)
- SmartPLC SafeLine AC4S with Profinet slave interface (AC412S)
- SmartPLC SafeLine AC4S with EtherNet/IP device interface (AC422S)
- SmartPLC SafeLine AC4S with EtherCAT slave interface (AC432S)

This document is to complement the device manuals of the a.-m. devices.

These instructions describe the configuration and programming of the device-internal standard PLC and the fail-safe PLC of AC4S by means of the CODESYS programming system.

1.3 Symbols and styles used

13839

► ... Instructions

> ... Reaction, result

→ ... Cross-reference or internet link

123 Decimal number
0x123 Hexadecimal number
0b010 Binary number

[...] Designation of pushbuttons, buttons or indications

1.4 Overview: User documentation for AC4S

18655

ifm electronic provides the following user documentation for the models of the device class "Fail-safe SmartPLC AC4S":

Document	Content / Description
Data sheet	Technical data of SmartPLC SafeLine AC4S as a table
Operating instructions *	▪ Notes on mounting and electrical installation of the SmartPLC SafeLine AC4S ▪ Set-up, description of the operating and display elements, maintenance information, scale drawing
Device manual	▪ Notes on operation of SmartPLC SafeLine AC4S via GUI and web interface ▪ Description of the cyclic and acyclic data records, fieldbus parameters and command interface ▪ Error description
Supplement device manual	▪ Description of the acyclic data sets and the command interface
Programming manual	▪ Creation of a project with the device using CODESYS ▪ Configuration of the device using CODESYS ▪ Programming of the standard PLC of the device ▪ Programming of the fail-safe PLC of the device ▪ Description of the device-specific CODESYS function libraries

Legend:
*... The operating instructions are supplied with the device.

All documents can be downloaded from ifm's website.

1.5 Overview: CODESYS documentation of 3S

20259

3S GmbH provides the following user documentation for programming standard PLC and the fail-safe PLC of AC4S:

Document	Content / Description
Online help	▪ Context-sensitive help ▪ Description of the CODESYS programming system and the safety extension
CODESYS installation and first steps	▪ Remarks about the installing of the CODESYS programming system ▪ First steps for handling the CODESYS programming system
Safety user manual	▪ Remarks about the safety-related project development with CODESYS ▪ Remarks about safety-related programming of the fail-safe PLC with CODESYS

After the installation of the CODESYS 3.5 programming system all documents are stored on the hard disk of the PC/laptop and can be accessed:

- Online help:
  `...\Program Files\3S CoDeSys\CoDeSys\Online-Help`
- CODESYS installation and first steps:
  `...\Program Files\3S CoDeSys\CoDeSys\Documentation`
- Safety user manual:
  `...\Program File\3S CoDeSys\CoDeSys\Documentation`

1.6 Modification history

21676

Version	Topic	Date
00	New creation of document	12/2017

2 Safety instructions

Contents

General safety instructions 9
Required background knowledge 9
Warnings used 10

213

2.1 General safety instructions

8516

Read this document before setting up the product and keep it during the entire service life.

Only use the product for its intended purpose.

If the operating instructions or the technical data are not adhered to, personal injury and/or damage to property may occur.

Improper or non-intended use may lead to malfunctions of the device, to unwanted effects in the application or to a loss of the warranty claims.

The manufacturer assumes no liability for any consequences caused by tampering with the device or incorrect use by the operator.

► Observe these operating instructions.

► Adhere to the warning notes on the product.

2.2 Required background knowledge

10317

This document is intended for specialists. Specialists are people who, based on their relevant training and experience, are capable of identifying risks and avoiding potential hazards that may be caused during operation or maintenance of the product.

To program SmartPLC SafeLine AC4S, the people should also be familiar with the software CODESYS 3.5 and the CODESYS Safety extension.

The people should also have knowledge of the following topics:

- Requirements on safety-relevant programming
- Standards DIN EN ISO 13849 and DIN EN 62061

2.3 Warnings used

13685

⚠ WARNING

Death or serious irreversible injuries may result.

⚠ CAUTION

Slight reversible injuries may result.

NOTICE

Property damage is to be expected or may result.

Important note
Non-compliance may result in malfunction or interference.

Information
Supplementary note.

3 System requirements

Contents

Hardware 11
Software.....11
Licensing.....11

16903

3.1 Hardware

16904

- Device of the SmartSPS SafeLine AC4S product family with firmware V4.2.5
- PC/laptop for CODESYS development system (→ system requirements CODESYS development system V3.x)
- Ethernet connection between CODESYS-PC/laptop and configuration interface (X3) of the device

3.2 Software

20253

To program the device-internal standard PLC and the fail-safe PLC of AC4S the following software components are necessary:

Component	Description	Version
CODESYS Development System	CODESYS programming software for PLC programming complying with the standard IEC 61131-3	3.5 SP9 Patch 7 Hotfix 3
Package ""CODESYS for ifm SmartPLC SafeLine""	▪ Safety extension for the CODESYS programming software ▪ Device and interface description of SmartPLC SafeLine AC4S ▪ Function libraries for programming of standard PLC ▪ Certified function libraries for programming the fail-safe PLC ▪ Certified libraries for the safety-relevant programming (SafetyPLCopen, safety standard)	1.5.2.10

The features and functions warranted in this manual can only be obtained by using the software components in the versions stated here.

On its website ifm electronic provides the required software components.
→ www.ifm.com > Service > Download > Industrial communication

3.3 Licensing

20254

By buying a device of the SmartSPS SafeLine AC4S the buyer has also purchased a licence valid for using the CODESYS 3.5 programming software.

4 Installation

Contents

CODESYS programming software 12
ifm AS-i Package 13

17146

4.1 CODESYS programming software

7282

The CODESYS Development System (short: CODESYS) is a platform for the creation of PLC applications according to the standard IEC 61131-3.

4.1.1 Install CODESYS Development System

18596

To install the software "CODESYS Development System":

► Install the programming system CODESYS 3.5 SP9 Patch 7 Hotfix 3 (→ CODESYS installation and first steps).

> CODESYS 3.5 SP9 Patch 7 Hotfix 3 is installed on the programming PC/laptop.

4.2 ifm AS-i Package

Contents

Components of the ifm package....13
Install the ifm package....14
Uninstall the ifm package....14

17679

► Familiarise yourself with the following CODESYS functions!
- Package Manager
  → Online help > CODESYS Development System > Manage packages and licences

4.2.1 Components of the ifm package

10982

ifm electronic provides the "CODESYS for ifm SmartPLC SafeLine"package (ifm package for short) for the programming of the SmartPLC SafeLine AC4S. The ifm package (file: `ifm_SmartPLC_SafeLine_V1_5_2_10.package`) contains the following certified components:

Component	Description
Plugins	CODESYS plugins
Files	Project templates, project examples, CODESYS safety user manual, CODESYS profiles
Online help files	Online helps (CODESYS safety, safety network variables)
Online help merge files	Online help system
Device description	Description of the device modules, interfaces and additional devices
Library	CODESYS libraries (SF_IO, SafetyPLCopen, SF_LogicalInterfaces, SafetyStandard, SafetySystemIO, SafetyNetVar, SafetyFSoEMaster, ACnnn_Utils. ACnnnn_SYS_CMD, IoStandard, SysSafetyIoBase_ifm, IoDrvEtherCAT, IoDrvEtherNetIP, IoDrvAL1020, IoDrvAL1030)

4.2.2 Install the ifm package

6966

To install the ifm AS-i package on the programming PC/laptop:

Requirements

> CODESYS 3.5 SP9 Patch 7 Hotfix 3 is installed on the programming PC/laptop.

1 Start CODESYS

► Start CODESYS.

> CODESYS user interface appears.

2 Install "CODESYS for ifm SmartPLC SafeLine"

► Select [Tools] > [Package Manager] to start the Package Manager.

> Window [Package Manager] appears.

► Click on [Install...] to start the installation dialogue.

► Select file `ifm_SmartPLC_SafeLine_V1_5_2_10.package` and carry out complete installation.

> Window [Package Manager] shows installed ifm package.

► Click on [Exit] to close the Package Manager.

► Note the remarks about the correct start of the programming system CODESYS Safety.
→ **Start CODESYS** (→ p. 15)

4.2.3 Uninstall the ifm package

12270

To uninstall the package "CODESYS for ifm SmartPLC SafeLine":

1 Start CODESYS

► Start CODESYS with administrator rights.

> CODESYS programming interface appears.

2 Uninstall the ifm package

► Select [Tools] > [Package Manager] to access the package manager.

> Window [Package Manager] shows the installed packages.

► Activate [Display version] checkbox.

> The window shows the version numbers of the installed packages.

► Select the package version to be uninstalled

► Click on [Uninstall...] to uninstall the selected package.

> The selected package version is uninstalled.

► Click on [Exit] to close the Package Manager.

5 Getting started

Contents

Start CODESYS 15
Create CODESYS project 16
Use CODESYS online help 20
Create user account 20
Configure programming interface 21

15858

5.1 Start CODESYS

19358

ifm electronic provides a special profile for CODESYS. The CODESYS profile "ifm electronic SmartPLC SafeLine V3.5.9.73" creates a preconfigured environment for the configuration and programming of an SmartPLC SafeLine AC4S automation system.

To start CODESYS with the profile "ifm electronic SmartPLC SafeLine V3.5.9.73":

Requirements

> Software components were correctly installed (→ **Install the ifm package** (→ p. 14))

1 Create desktop shortcut

► Delete the desktop shortcut of CODESYS created during installation.

► Create desktop shortcut of the following application:
[Start] > [All Programs] > [3S CODESYS] > [CODESYS] > [CODESYS without Profile]

2 Start CODESYS with the ifm profile

► Double click on desktop shortcut [CODESYS without Profile]

> Selection window appears.

► Select [ifm electronic SmartPLC SafeLine V3.5.9.73] from the [Version profile] list.

► Press [Continue] to apply the selection and load the profile.

> The CODESYS programming system starts using the selected profile.

5.2 Create CODESYS project

Contents

Create new project with SmartPLC SafeLine AC4S 17
Overview: Project structure with SmartPLC SafeLine AC4S 19

17129

► Familiarise yourself with the following CODESYS functions!
- Create CODESYS project
  → Online help > CODESYS Development System > Create and configure project
- Objects of the user interface
  → Online help > CODESYS Development System > Reference user interface

5.2.1 Create new project with SmartPLC SafeLine AC4S

12225

To avoid errors with a manual system configuration it is highly recommended to use the project template of ifm electronic for the creation of an SmartSPS SafeLine AC4S project in CODESYS.

Requirements:

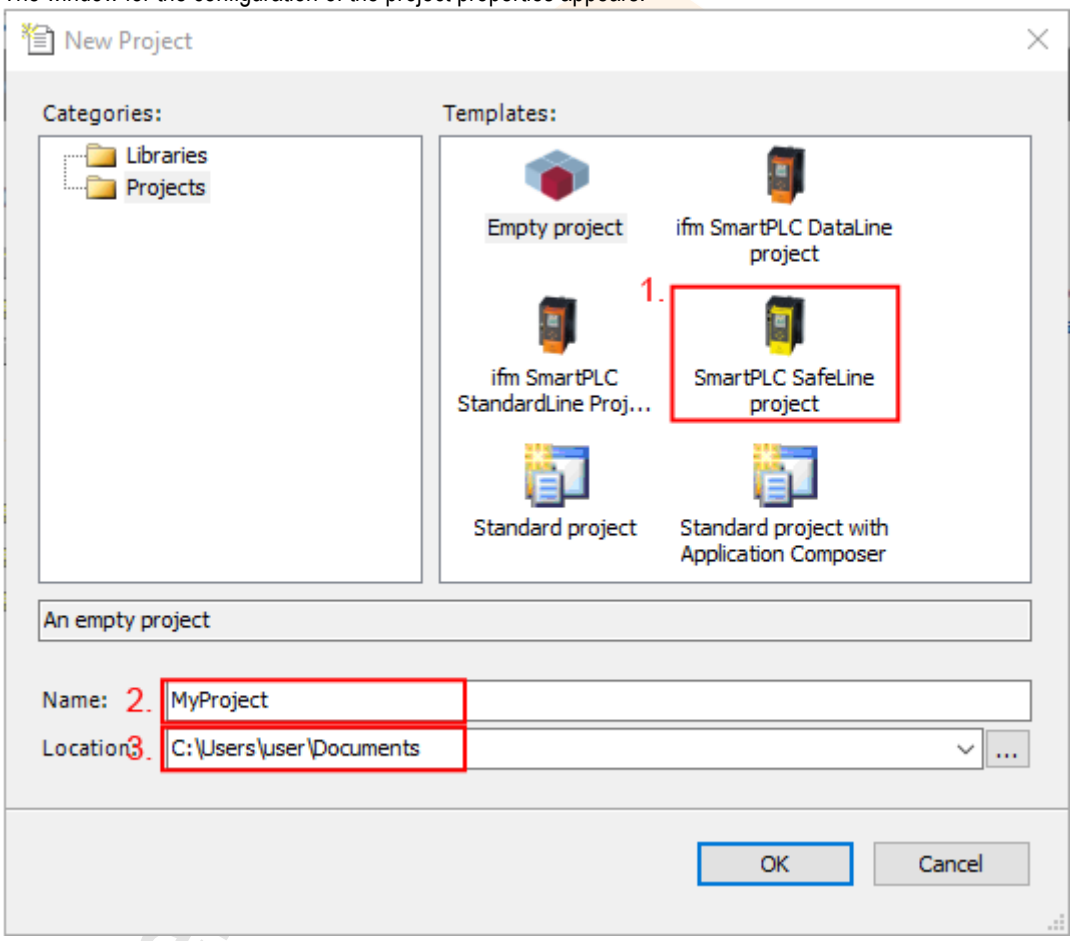
> All required software components are correctly installed (→ **Install the ifm package** (→ p. 14)).

> CODESYS was started (→ **Start CODESYS** (→ p. 15)).

1 Create new SmartPLC SafeLine AC4S project

► Select [File] > [New Project ...].

> The window for the configuration of the project properties appears.



► Set the following values:
1. [Templates]: Select [ifm_SmartPLC_SafeLine Project].
2. [Name]: Enter project name
3. [Location]: Select storage location for the project file.

► Click on [OK] to confirm the entered values.

> CODESYS creates a new project with an SmartPLC SafeLine AC4S.

> The window [Devices] shows the device tree of the project (→ **Overview: Project structure with SmartPLC SafeLine AC4S** (→ p. 19)).

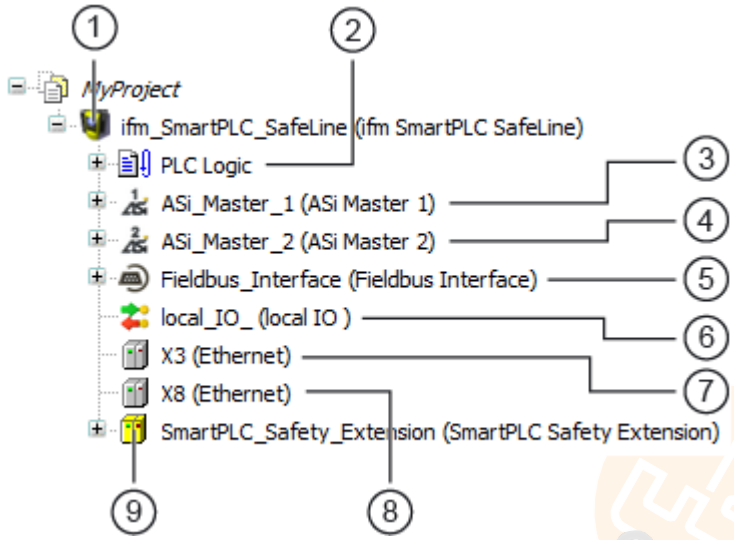
2 Save project

► Select [File] > [Save Project].

> CODESYS saves the project.

5.2.2 Overview: Project structure with SmartPLC SafeLine AC4S

12275



(1) [ifm_SmartPLC_SafeLine (ifm SmartPLC SafeLine)] provides access to the settings of the standard PLC of the SmartPLC SafeLine AC4S (→ **Configure standard PLC** (→ p. 23))

(2) [PLC Logic] contains the standard PLC applications of SmartPLC SafeLine AC4S.
→ **Objects of a standard PLC application** (→ p. 47)

(3) [ASi_Master_1] contains the following objects:

- Process data images of the standard AS-i slaves at AS-i master 1
  → **Access standard input and output data** (→ p. 50)
- Configuration nodes for safe AS-i slaves at AS-i master 1
  → **Configure safe AS-i slaves** (→ p. 25)

(4) [ASi_Master_2] contains the following objects:

- Process data images of the standard AS-i slaves at AS-i master 2
  → **Access standard input and output data** (→ p. 50)
- Configuration nodes for safe AS-i slaves at AS-i master 2
  → **Configure safe AS-i slaves** (→ p. 25)

(5) [Fieldbus_Interface] provides access to the input and output data of the fieldbus interface.
→ **Access standard input and output data** (→ p. 50)

(6) [local_IO] is the configuration node for safe devices at the local I/O interface
→ **Configure safe devices at local I/O interface** (→ p. 32)

(7) [X3] provides access to the configuration options of the configuration interface 2 (X3)
→ **Extend fieldbus functionality** (→ p. 42) and → **Change IP settings** (→ p. 40)

(8) [X8] provides access to the configuration options of the configuration interface 2 (X8)
→ **Extend fieldbus functionality** (→ p. 42) and → **Change IP settings** (→ p. 40)

(9) [SmartPLC_Safety_Extension] provides access to the following objects:

- Settings of the fail-safe PLC of AC4S (→ **Configure fail-safe PLC** (→ p. 24))
- Safe application (→ **Objects of a safe application** (→ p. 72))

5.3 Use CODESYS online help

6989

This manual only describes the integration, configuration and the programming of the SmartPLC SafeLine AC4S using the CODESYS development system.

For the description of user actions and user interface elements the CODESYS terminology will be used.

Standard functions and methods of CODESYS will not be described. At the beginning of each section there will be a reference to the corresponding chapters of the CODESYS online help.

To access the online help of the CODESYS development system:

► Start CODESYS.

> The CODESYS user interface appears.

► Press [F1].

> Online help of the CODESYS development system appears.

► Familiarise yourself with the CODESYS development system! In particular with the following topics:
- Names and functions of the user interface elements
- Basic menu functions
- Programming techniques and methods for data retention

5.4 Create user account

18968

► Familiarise yourself with the following CODESYS functions!
- Safety user administration and access control:
  → Online help > Add-Ons > CODESYS Safety > Safety user administration and access control
- Safety user configuration:
  → Online help > Add-Ons > CODESYS Safety > Safety user configuration

Safety-relevant objects and functions of a project must only be carried out by a user who is a member of the user groups "Safety" or "Safety.ExtendedLevel".

To create a user:

► Select [Project] > [Project settings]

► CODESYS user administration appears.

► Create user and add required user group.

► Create password.

5.5 Configure programming interface

20231

To transfer the created project to SmartPLC SafeLine AC4S a valid communication path has to be configured between the programming system CODESYS and SmartPLC SafeLine AC4S.

standard PLC and fail-safe PLC of AC4S use a separate hardware. Therefore it is necessary to set the communication paths for both controllers separately.

5.5.1 Set communication path of standard PLC

20233

The following interfaces can be configured as programming interface:

- Ethernet configuration interface 1 (X3)
- Ethernet configuration interface 2 (X8)

To configure the connection between CODESYS programming software and standard PLC of the SmartPLC SafeLine AC4S:

1 Preparations

► Connect CODESYS PC/laptop and configuration interface 1 (X3) or 2 (X8) of the device.

► Optional: Adjust IP settings of the Ethernet interfaces.

2 Select communication settings

► In the device tree: Double click on the [ifm_SmartPLC_SafeLine] symbol

► In the editor window: Click on [Communication Settings].

> The editor window shows the communication settings of the PLC.

3 Select gateway

► Select the requested gateway in the [Gateway] list.

> List shows selected gateway.

4 Set communication path

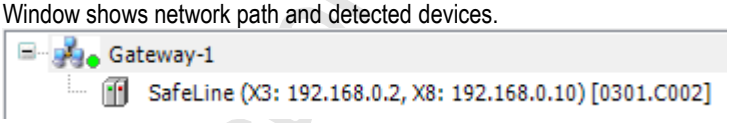
► Click on [Scan Network...]

> Window [Select Device] appears.

► Select the gateway node and click on [Scan network] to start the scan process.

> CODESYS scans network for devices.

> Window shows network path and detected devices.



► Select node of the SmartPLC SafeLine AC4S.

> Information field shows detailed information about selected node.

► Click on [OK] to set the communication path to standard PLC.

> CODESYS can transfer data to the standard PLC of the SmartPLC SafeLine AC4S.

5.5.2 Set communication path of the fail-safe PLC

20232

To configure the connection between CODESYS programming software and the fail-safe PLC of SmartPLC SafeLine AC4S:

Requirements:

> Communication path to standard PLC is correctly set (→ **Set communication path of standard PLC** (→ p. 21)).

1 Select communication settings

► In the project tree: Double click on the [SamrtPLC_Safety_Extension] symbol

> The editor window shows the tab [Communication Settings].

2 Set communication path

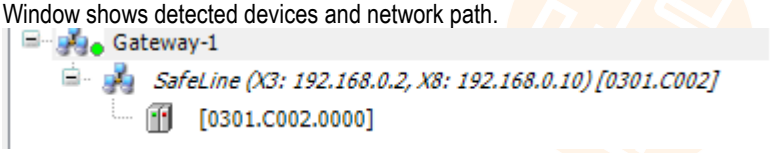
► Activate [Scan Network...]

> Window [Select Device] appears.

► Mark node [Gateway-1] and start scan process with [Scan network].

> Device scans network for devices.

> Window shows detected devices and network path.



► Select the sub-node of the SafeLine node in the network path.

> Information field shows detailed information about selected node.

► Click on [OK] to set the communication path to the fail-safe PLC.

3 Acknowledge connection to the fail-safe PLC

> Window [Connection to the Fail-safe PLC] appears.

► Enter the serial number of the device in the field [Instance identification].

► Click on [OK] to acknowledge the input.

> CODESYS verifies the connection to the fail-safe PLC.

> CODESYS can log in to the fail-safe PLC.

6 System configuration

Contents

Configure standard PLC 23
Configure fail-safe PLC 24
Configure safe AS-i slaves 25
Configure safe devices at local I/O interface 32
Configure standard devices at local I/O interface 39
Change IP settings 40
Add additional device to the project 41
Extend fieldbus functionality 42

18964

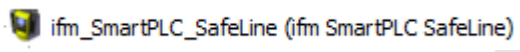
This chapter contains information about the configuration of the device-internal PLC, the inserted safe AS-i slaves, the safe sensors/actuators at the local I/O interface and the Ethernet configuration interface.

6.1 Configure standard PLC

18961

► Familiarise yourself with the following CODESYS functions!
- Generic device editor
  → Online help > CODESYS Development System > Reference user interface > Objects> object 'device' and generic device editor

The standard PLC is configured via the "Generic Device Editor" of the CODESYS programming system. The programmer can access the device editor of the standard PLC via the following node in the device tree:



To configure the device-internal standard PLC:

► In the device tree: Double-click on [ifm_SmartPLC_SafeLine]

> The editor window shows device editor of the device-internal standard PLC.

► Configure standard PLC.

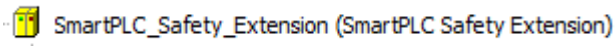
► Save the project to apply changes.

6.2 Configure fail-safe PLC

18960

► Familiarise yourself with the following CODESYS functions!
  - Editor of the safety controller
    → Online help > Add-Ons > CODESYS Safety > Editors > Editor of the safety controller

The fail-safe PLC of AC4S is configured via the "device editor of the safety controller" of the CODESYS programming system. The programmer can access the device editor of the fail-safe PLC via the following node in the device tree:



To configure the fail-safe PLC of AC4S:

► In the device tree: Double click on [SmartPLC_Safety_Extension (SmartPLC Safety Extension)]

> The editor window shows the device editor of the fail-safe PLC of AC4S.

► Configure fail-safe PLC as required.

The button [Firmware Update] in the tab [Sicherheitssteuerung] does not have any functionality in connection with the SmartPLC SafeLine AC4S.

► Do not activate the button.

An update of the firmware of the SmartPLC SafeLine AC4S must only be made by an authorised employee of ifm electronic gmbh.

► Contact your AS-i specialist for more information.

► After a firmware update the user must re-evaluate and compile the boot application and load it to the device for safety reasons.

6.3 Configure safe AS-i slaves

Contents

Remarks 25
Add safe AS-i slaves to the project 27
Remove safe AS-i slave from project 31

18974

The safe inputs and outputs are configured via the "Device editor for logical I/Os" of the programming system CODESYS Safety. In the following sections only the AC4S-specific functions are described.

► Familiarise yourself with the following CODESYS functions!

- Device editor for logical I/Os
  → Online help > Add-Ons > CODESYS Safety > Editors > Device editor for logical I/Os

6.3.1 Remarks

20227

Concept of the logical devices

20230

Safe AS-i slaves are inserted in a CODESYS project via logical devices. Depending on the type of the safe AS-i slave the logical devices fulfil different functions.

Safe AS-i input slaves

20228

Mapped to the classical behaviour of a programmable logic controller the logical device assumes the functions of signal detection and signal processing. The individual functions are:

- detect the code half-sequences generated by the safe AS-i input slave
- pre-process the code half sequences by means of the internal logic
- provide the result of the logical preprocessing as safe variable value

The programmer can configure the logical preprocessing via the parameter interface of the logical device.

Safe AS-i output slaves

20229

Safe AS-i output slaves are indirectly controlled via a virtual AS-i control slave. The AS-i control slave assumes following functions:

- generate a safe code sequence for unlocking a safe AS-i output slave

AC4S detects the AS-i control slave only after the download of the safety project to AC4S. Then the respective AS-i master generates a configuration error (unknown slave).

► Carry out projection adaptation to eliminate the error (→ Device manual, **Quick setup: Project AS-i networks**)

Available logical devices for safe AS-i slaves

18987

Following logical devices are available for safe AS-i slaves:

Designation	Type	Reference
SF_IN_ASi_forced	Input	→ **SF_IN_ASi_forced** (→ p. 206)
SF_IN_ASi_independent	Input	→ ***SF_IN_ASi_independent*** (→ p. 210)
SF_IN_ASi_conditionally_dependent	Input	→ **SF_IN_ASi_conditionally_dependent** (→ p. 214)
SF_IN_ASi_dependent	Input	→ **SF_IN_ASi_dependent** (→ p. 218)
SF_IN_ASi_dependent_filter_w_testreq	Input	→ **SF_IN_ASi_dependent_filter_w_testreq** (→ p. 222)
SF_IN_ASi_dependent_filter_ntestreq	Input	→ **SF_IN_ASi_dependent_filter_ntestreq** (→ p. 226)
SF_IN_ASi_dependent_filter_nshutdown	Input	→ **SF_IN_ASi_dependent_filter_nshutdown** (→ p. 230)
SF_OUTcontrol_ASi	Output	→ **SF_OUTcontrol_ASi** (→ p. 234)

Detailed information about the logical devices:
→ **Remarks about logical devices** (→ p. 202)

6.3.2 Add safe AS-i slaves to the project

8964

Safe AS-i slaves are added to a CODESYS project in 2 steps:

1. Add logical device of the safe AS-i slave to the project tree:
   → **Add logical device to the project tree** (→ p. 27)
2. Configure logical device:
   → **Configure logical device** (→ p. 29)

The user is responsible for the selection of the suitable logical devices. The user must ensure that the selected logical devices provide the functionality required for the safety application to be implemented.

► Observe the documentation of the logical devices (→ **Logical devices for safe AS-i slaves** (→ p. 205))!

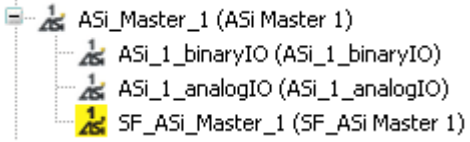
A logical device with a defined AS-i address must only be added to a safety project once.

Add logical device to the project tree

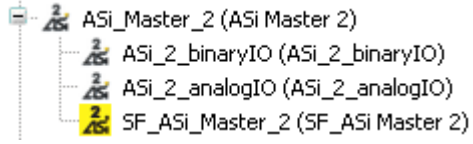
14806

Safe AS-i slaves are added to a standard branch of the device tree into the CODESYS project via the nodes [SF_ASi_Master_1] and [SF_ASi_Master_2]:

- Safe AS-i slaves connected to AS-i master 1:



- Safe AS-i slaves connected to AS-i master 2:



To add a safe AS-i input or output slave to the project:

Requirements:

> Safe AS-i slave is correctly installed and addressed.

> AS-i network is projected.

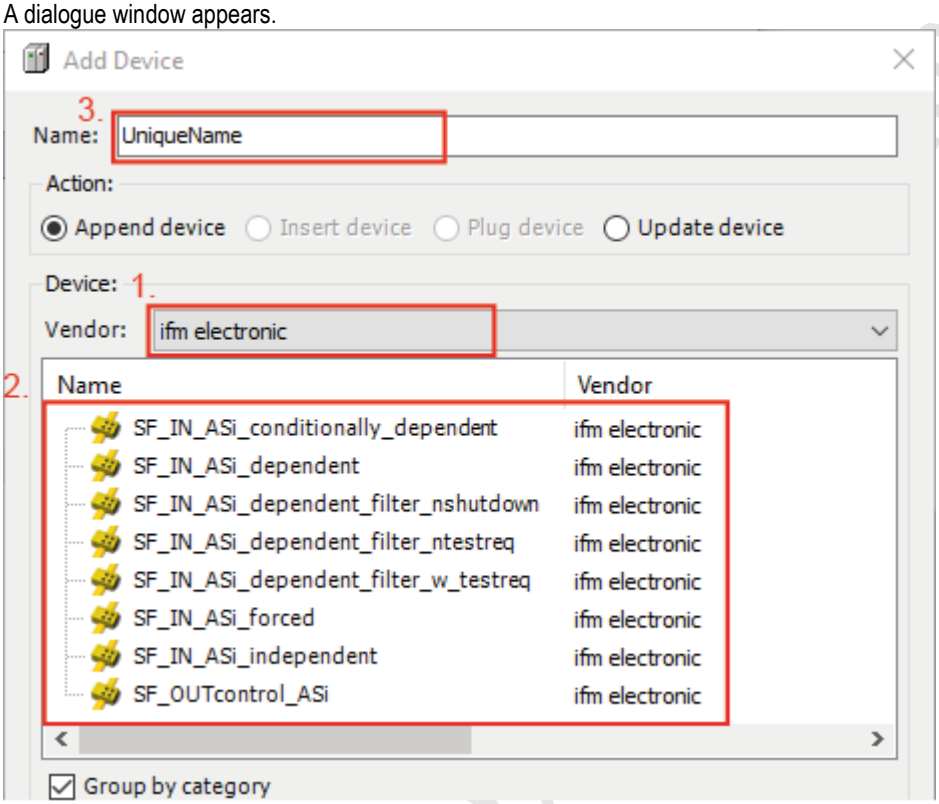
1 Select AS-i network

► In standard branch of the device tree: Mark safety nodes of the AS-i network in which the safe AS-i slave is installed:
– for AS-i 1: [ASi_Master_1] > [SF_ASi_Master_1]
– for AS-i 2: [ASi_Master_2] > [SF_ASi_Master_2]

2 Select logical device

► Select [Project] > [Add Device...].

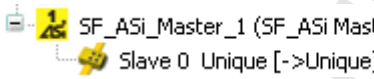
> A dialogue window appears.



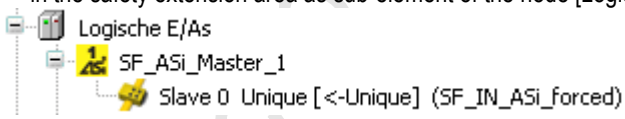
► Set the following values in the area [Device]:
1. [Vendor]: ifm electronic
2. Table: Select the required logical device (→ Remark)
3. [Name]: Enter unique name for the logical device

► Apply the set values with [Add Device].

> CODESYS adds the logical device to the following positions of the device tree:
– in the standard area as sub-element of the safety node of the selected AS-i network



– in the safety extension area as sub-element of the node [Logical I/Os]



► Close the dialoge window with [Close].

3 Assign AS-i address

► In standard branch of the device tree:
Double click on the added logical device

> Editor window shows the tab with configuration options of the logical device.

► Select tab [Safe ASi IO Configuration].

> Table shows parameters of the logical device.

► Enter the address of the safe AS-i slave in column [Value].

> Logical device is coupled to the safe AS-i slave.

> Symbol in the device tree shows assigned AS-i address.

4 Optional: Add more logical devices

► Repeat steps 1 to 3 to add additional logical devices to device tree.

Remarks about the configuration of the logical device: → **Configure logical device** (→ p. 29)
Remarks about removal of the logical device: → **Remove safe AS-i slave from project** (→ p. 31)

Configure logical device

18978

Configuration is made via the node [Logical I/Os] in the safety extension area of the device tree. All safe AS-i slaves inserted in the project are listed there as sub-nodes.

To configure the logical device of a safe AS-i slave:

► In the safety extension area of the device tree:
Double click on the symbol of the logical device at [Safety Logic] > [SafetyApp] > [Logical I/Os]

> Editor window shows following options:

Tab	Description
[Safe configuration]	Parameter interface of the logical device (→ **Set the parameters of the logical device** (→ p. 29))
[I/O mapping]	Variable image for access from safety application (→ **Map safe process signal to variable** (→ p. 30))
[Information]	Information about the safe AS-i slave (→ Online help (CODESYS Safety)

Set the parameters of the logical device

7096

The [Safe configuration] tab enables access to the parameter interface of the logical device. Number and type of available parameters depend on the selected logical device.

Improperly set parameters can lead to undesired behaviour of the system.

► Familiarise yourself with parameters of the logical device before the configuration (→ **Logical devices for safe AS-i slaves** (→ p. 205))!

All time indications have an inaccuracy of +/- 1 PLC cycle time. The user can freely set the cycle time of the device-internal fail-safe PLC within the defined limits.

► Note the set PLC cycle time for dimensioning the following parameters.
- Stabilising time (StabilisingTime)
- Synchronisation time (SynchronisingTime)
- Tolerance time (ToleranceTime)

To configure the logical preprocessing of the logical device:

> Select the tab [Safe configuration] in the device editor.

> Editor window shows table with available parameters of the logical device.

► Enter the required parameter values for all displayed parameters in column [Value].

> Entered values are applied.

Map safe process signal to variable

20246

The logical device generates a safe process signal. To be able to use this signal in the safety application it has to be mapped to a variable.

To map the created process signal of a logical device to a safe variable:

► Select the tab [I/O mapping] in the device editor.

> Editor window shows table with mapping settings.

► Enter the name of the variable in the variable field of bit 0 to which the safe process signal is to be mapped.

> CODESYS uses the entered designation to create a global variable of type SAFEBOOL.

> Safety application can access the safe process signal via the variable.

The created variable is stored in the global variable list (GVL) `IoConfigRemote_Globals`.

► Always choose an unambiguous name for the variable.

6.3.3 Remove safe AS-i slave from project

17125

To completely remove a safe AS-i slave from the project the 2 instances of the logical device must be deleted separately in the project tree.

1 Remove logical device from the standard area

► Mark logical device in standard branch of the device tree.

► Select [Edit] > [Delete].

> CODESYS removes the logical device from the standard area.

2 Remove logical device from the safety area

► Mark logical device in the safety extension area of the device tree.

► Select [Edit] > [Delete].

> CODESYS removes the logical device from the safety extension area.

6.4 Configure safe devices at local I/O interface

Contents

Remarks 32
Add safe local device to project 34
Create test signal at local output 39
Remove safe local device from project 39

18972

The safe inputs and outputs are configured via the "Device editor for logical I/Os" of CODESYS safety. In the following sections only the AC4S-specific functions are described.

► Familiarise yourself with the following CODESYS functions!

- Device editor for logical I/Os
  → Online help > Add-Ons > CODESYS Safety > Editors > Device editor for logical I/Os

6.4.1 Remarks

20227

Concept of the logical devices

18963

Safe devices at the local I/O interface are inserted in a CODESYS project via logical devices. Depending on the type of the safe device the logical devices fulfil different functions.

Safe devices at local inputs

18984

Mapped to the classical IPO behaviour of a programmable logic controller the logical device assumes the functions of signal detection and signal processing. The individual functions are:

- detect the signals generated by the safe devices
- pre-process the signals by means of the internal logic
- provide the result of the logical preprocessing as safe variable value

The programmer can configure the logical preprocessing via the parameter interface of the logical device.

Safe devices at local outputs

18966

Safe devices at the local outputs can be controlled via logical devices. The logical device fulfils the following functions:

- create a safe signal
- provide the safe signal at the selected output channels at the local I/O interface

Available logical devices for safe inputs/outputs

18983

Following logical devices are available for safe devices at the local I/O interface:

Designation	I/O type	Reference
SF_IN_local_forced	Input	→ **SF_IN_local_forced** (→ p. 236)
SF_IN_local_independent	Input	→ **SF_IN_local_independent** (→ p. 240)
SF_IN_local_conditionally_dependent	Input	→ **SF_IN_local_conditionally_dependent** (→ p. 245)
SF_IN_local_dependent	Input	→ **SF_IN_local_dependent** (→ p. 252)
SF_IN_local_dependent_filter_testreq	Input	→ **SF_IN_local_dependent_filter_testreq** (→ p. 257)
SF_IN_local_dependent_filter_ntestreq	Input	→ **SF_IN_local_dependent_filter_ntestreq** (→ p. 263)
SF_IN_local_dependent_filter_nshutdown	Input	→ **SF_IN_local_dependent_filter_nshutdown** (→ p. 267)
SF_OUT_local_single	Output	→ **SF_OUT_local_single** (→ p. 273)
SF_OUT_local_dual	Output	→ **SF_OUT_local_dual** (→ p. 274)
SF_OUT_local_testpulse	Output	→ **SF_OUT_local_testpulse** (→ p. 275)

Detailed information about the logical devices:
→ **Remarks about logical devices** (→ p. 202)

6.4.2 Add safe local device to project

18979

Safe devices at the local I/O interface are added to a CODESYS project in 2 steps:

1. Add logical device to the project tree:
   → **Add logical device to the project tree** (→ p. 34)
2. Configure logical device:
   → **Configure logical device** (→ p. 37)

The user is responsible for the selection of the suitable logical devices. The user must ensure that the selected logical devices provide the functionality required for the safety application to be implemented.

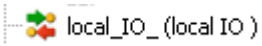
► Observe the documentation of the logical devices (→ **Logical devices for the local I/O interface** (→ p. 235))!

A logical device with defined input channels and/or output channels must only be added to a safety project once.

Add logical device to the project tree

18982

Safe devices at the local I/O interface are added to the project in standard branch of the device tree via the node [local_IO_]:



The user is responsible for selecting the suitable logical devices. The user must ensure that the selected logical devices provide the functionality required for the safety application to be implemented.

► Observe the documentation of the logical devices (→ **Logical devices for the local I/O interface** (→ p. 235))!

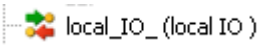
A logical device of a type with 2 defined input channels (Input_Channel_A, Input_Channel_B) must only be added to a safety project once.

Requirement:

> Devices are connected to the local I/O interface according to the required safety classification.

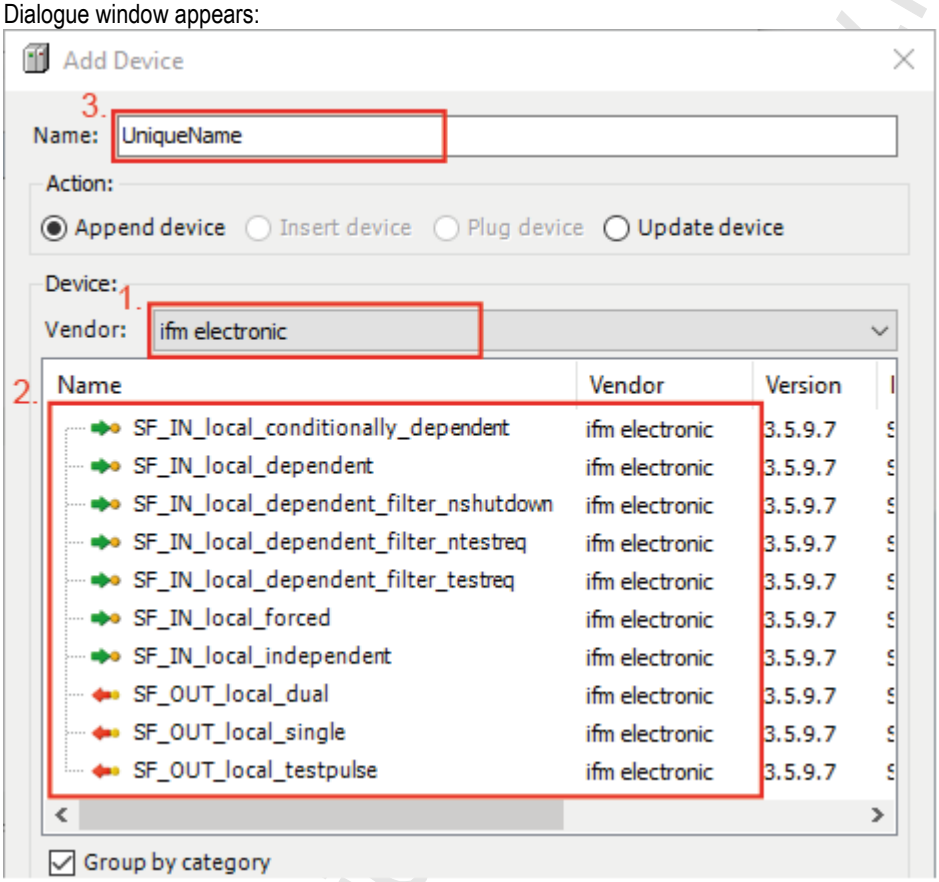
1 Add logical device

► In standard branch of the device tree: Select node [local_IO_]:



► Select [Project] > [Add Device...]

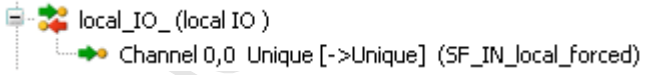
> Dialogue window appears:



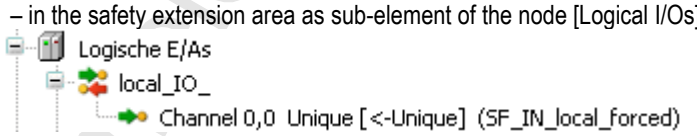
► Set the following parameters in group [Device]:
1. [Vendor]: Select ifm electronic.
2. Table: Mark the required logical device (→ Remark).
3. [Name]: Enter unambiguous instance name.

► Apply the set values with [Add Device].

> CODESYS adds instances of the logical device to the following positions of the device tree:
– in the standard area as sub-element of the node [local_IO_]



– in the safety extension area as sub-element of the node [Logical I/Os]



► Close the dialogue window with [Close].

2 Assign input and/or output channels

► In standard branch of the device tree:
Double click on the added logical device.

> Editor window shows configuration options of the logical device.

► Select tab [Safe Local IO Configuration].

> Table shows the following parameters in dependence on the selected logical device:

Interface type	Parameter	Description
Input (2 channels)	Input_Channel_A	Local input which is connected to sensor channel A
	Input_Channel_B	Local input which is connected to sensor channel B
Output (1 channel)	Output_Channel	Local output which is connected to the actuator or at which the test signal is to be provided.
Output (2 channels)	Output_Channel_A	Local output which is connected to actuator channel A.
	Output_Channel_B	Local output which is connected to actuator channel B.

► Enter the required value for each parameter in column [Value].

> Logical device is coupled with the set input and/or output channels.

> Symbol in the device tree shows assigned I/O channels.

3 Optional: Add more logical devices

► Repeat steps 1 to 2 to add additional logical devices to device tree.

Remarks about the configuration of the logical devices → **Configure safe devices at local I/O interface** (→ p. 32)
Remarks about removal of the logical device: → **Remove safe local device from project** (→ p. 39)

Configure logical device

18980

Configuration is made via the node [Logical I/Os] in the safety extension area of the device tree. All safe local devices added to the project are listed there as sub-node.

To configure the logical device of a safe sensor at a local input channel:

► In the safety extension area of the device tree:
Double click on the symbol of the logical device at [Safety Logic] > [SafetyApp] > [Logical I/Os]

> Editor window shows following options:

Tab	Description / Reference
[Safe configuration]	Parameter interface of the logical device (→ **Set the parameters of the logical device** (→ p. 37))
[I/O mapping]	Variable image for access from safety application (→ **Map safe process signal to variable** (→ p. 38))
[Information]	Information about the safe peripheral (→ CODESYS help)

Set the parameters of the logical device

18973

The [Safe configuration] tab enables access to the parameter interface of the logical device. Number and type of available parameters depend on the selected logical device.

Incorrectly set parameters can lead to undesired behaviour of the system.

► Familiarise yourself with parameters of the logical device before the configuration (→ **Logical devices for the local I/O interface** (→ p. 235))!

All time indications have an inaccuracy of +/- 1 PLC cycle time. The user can freely set the cycle time of the device-internal fail-safe PLC within the defined limits.

► Note the set PLC cycle time for dimensioning the following parameters.
- Stabilising time (StabilisingTime)
- Synchronisation time (SynchronisingTime)
- Tolerance time (ToleranceTime)

To configure the logical device:

► Select the [Safe configuration] tab in the device editor.

> The editor window shows a table with available parameters of the logical device.

► Enter the required values for all displayed parameters in [Value] column.

> Entered values are applied.

Map safe process signal to variable

18989

The logical device generates a safe process signal. To be able to use this signal in the safety application it has to be mapped to a variable.

To map the created process signal of a logical device to a safe variable:

► Select the tab [I/O mapping] in the device editor.

> Editor window shows table with mapping settings.

► Enter the name of the variable in the variable field of bit 0 to which the safe process signal is to be mapped.

> CODESYS uses the entered designation to create a global variable of type SAFEBOOL.

> Safety application can access the safe process signal via the variable.

The created variable is stored in the global variable list (GVL) `IoConfigRemote_Globals`.

► Always choose an unambiguous name for the variable.

6.4.3 Create test signal at local output

20240

Certain device types require an input signal as test pulse to detect a cross fault of the two input channels. By means of the logical device SF_OUT_local_testpulse the programmer can create a test signal at a local output.

To create a test signal at a local output:

► Add logical device "SF_OUT_loal_testpulse" to the project tree
(→ **Add logical device to the project tree** (→ p. 34))

The parameters of the test pulse are firmly set; there are no additional configuration options (→ **SF_OUT_local_testpulse** (→ p. 275)).

6.4.4 Remove safe local device from project

8963

To completely remove a safe device added to the local I/O interface the 2 instances of the logical device must be deleted separately from the project tree.

1 **Remove logical device from the standard area**
- ► Mark logical device in standard branch of the device tree.
- ► Select [Edit] > [Delete].
- > CODESYS removes the logical device from the standard area.

2 **Remove logical device from the safety area**
- ► Mark logical device in the safety extension area of the device tree.
- ► Select [Edit] > [Delete].
- > CODESYS removes the logical device from the safety extension area.

6.5 Configure standard devices at local I/O interface

6976

Standard devices are added without logical device to the CODESYS project. When the device has been properly connected to the local I/O interface, the programmer can access the process data directly from the safe application via function blocks.

Access to non-safe inputs and outputs of the local I/O interface is made only via the fail-safe PLC of SmartPLC SafeLine AC4S.

- To access the input/output data of the non-safe devices at the local I/O interface:
→ **Local I/O interface: Access non-safe inputs/outputs** (→ p. 79)
- To transfer process data between standard PLC and fail-safe PLC of SmartPLC SafeLine AC4S:
→ **Data exchange between standard PLC and fail-safe PLC** (→ p. 83)

6.6 Change IP settings

14524

► Familiarise yourself with the following CODESYS functions!

- Ethernet adapter configurator
  → Online help > Fieldbus support > Ethernet adapter configurator

The Ethernet configuration interfaces 1 (X3) and 2 (X8) must not be participants of the same EtherNet subnet. Non-compliance may result in connection problems between the CODESYS programming system and the device.

► Configure IP settings so that interfaces X3 and X8 are part of different Ethernet subnets.

To change the IP settings of the Ethernet configuration interfaces 1 (X3) and 2 (X8):

1 Select Ethernet interface

► In the device tree: Double click on the requested Ethernet interface (X3 or X8)

> The editor window shows the Ethernet device editor.

2 Change IP settings

► Change IP settings of the Ethernet interface.

> Save the project to apply changes.

6.7 Add additional device to the project

18967

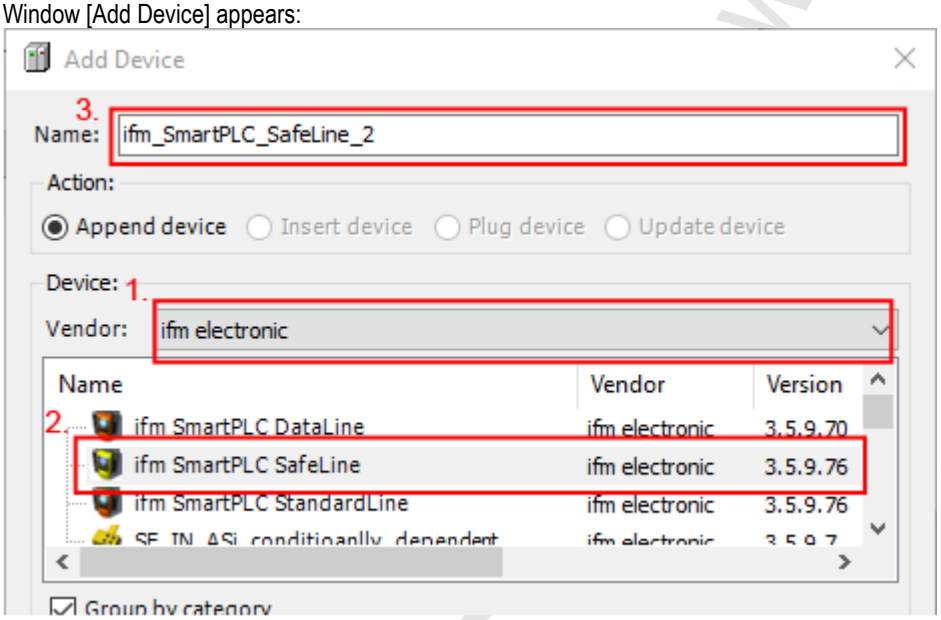
It is possible to manage several SmartPLC SafeLine AC4S within one CODESYS project.
To add another device to a CODESYS project:

Requirements:

> CODESYS project was correctly created (→ **Create new project with SmartPLC SafeLine AC4S** (→ p. 17)).

1 Add device

► Right-click on [MyProject]
> Context menu appears.
► Select [Add device...]
> Window [Add Device] appears:



► Set the following values:
1. [Vendor]: Select [ifm electronic].
2: [Device]: Select [ifm SmartPLC SafeLine] from the list.
3. [Name]: Enter unique name for the device.
► Click on [Add Device] to add the device to the project.
► Click on [Close] to close the window.
> CODESYS adds device to the device tree.

2 Configure the device

► Configure the device as requested (→ **System configuration** (→ p. 23)).

3 Save project

► Select [File] > [Save Project].
> CODESYS saves the project.

6.8 Extend fieldbus functionality

Contents

Available fieldbus stacks 42
Use EtherCAT master 43
Add fieldbus stack 46

15782

6.8.1 Available fieldbus stacks

13978

Besides the fieldbus functionality defined in the hardware the device can be operated as fieldbus device by a software extension. In this respect the user has to assign a fieldbus stack to the Ethernet configuration interfaces 1 (X3) or 2 (X8) in CODESYS and configure it. The device supports the following fieldbus stacks:

Description	Fieldbus	Further information
EtherCAT master	EtherCAT	→ **Use EtherCAT master** (→ p. 43)
Ethernet/IP scanner	EtherNet/IP	→ **Add fieldbus stack** (→ p. 46)
Modbus TCP master	Modbus TCP	
Modbus TCP slave device	Modbus TCP	

6.8.2 Use EtherCAT master

Contents

Add EtherCAT master 43
Configure EtherCAT master 44
Add and configure EtherCAT slave 45

7474

The device can also be operated as EtherCAT master. In this respect ifm electronic provides an EtherCAT master stack for the CODESYS programming system 3.5. The user has to assign one of the two Ethernet configuration interfaces (X3 or X8) to the EtherCAT master.

Add EtherCAT master

7475

Only one EtherCAT master can be used per device.

To add the EtherCAT master stack to the project:

Requirements:

- ► Open or create project with SmartPLC SafeLine AC4S.

1 Add EtherCAT master stack

- ► In the device tree: Right-click on [<CODESYS _Root>]
- ► In the context menu: select [Add device...]
- > Window [Add Device] appears.
- ► Set the following values:
  1. [Vendor]: Select [ifm electronic].
  2. In table: Select [EtherCAT Master].
- ► Click on [Add Device] to add the selected device to the project.
- ► Click on [Close] to close the window.
- > CODESYS adds EtherCAT master to the device tree.

CODESYS automatically adds an EtherCAT master call to the task with the shortest cycle time.

For notes on the configuration of task processing: → **Configure task processing** (→ p. 70)

Configure EtherCAT master

7477

► Familiarise yourself with the following CODESYS functions!
  - EtherCAT master
    → Online help > Fieldbus support > EtherCAT configurator > EtherCAT master

The user has to assign the Ethernet configuration interface 1 (X3) or 2 (X8) to the added EtherCAT master.

1 Requirements

> CODESYS laptop/PC is correctly connected to the device.
> Network path of standard PLC is set (→ **Configure programming interface** (→ p. 21)).

2 Assign Ethernet interface

► In the device tree: Double-click on [EtherCAT_Master (EtherCAT Master)]
► Click on the [General] tab.
> The editor window shows the configuration options of the EtherCAT master.
► Click on [Browse...]
> Window [Select Network Adapter] appears.
► Select requested Ethernet interface (eth0 = X3 or eth1 = X8).
► Click on [OK] to apply the selected Ethernet interface.
> Field [Source Address (MAC)] shows the MAC address of the selected Ethernet interface.
> Field [Network Name] shows the name of the Ethernet interface.
> EtherCAT master is coupled with selected Ethernet interface.

3 Configure EtherCAT master

► Set the parameters of the EtherCAT master as requested.
> EtherCAT master is configured.

Continue with: → **Add and configure EtherCAT slave** (→ p. 45)

Add and configure EtherCAT slave

7479

► Familiarise yourself with the following CODESYS functions!
  - EtherCAT slave
    → Online help > Fieldbus support > EtherCAT configurator > EtherCAT slave

To configure EtherCAT slaves for the operation with the EtherCAT master of the SmartPLC SafeLine AC4S:

Requirements:

> EtherCAT master is added to the project and configured (→ **Add EtherCAT master** (→ p. 43)).

> Device description file of the EtherCAT slave has been downloaded (→ manufacturer's website).

1 Optional: Add EtherCAT slave to the device repository

► Select [Tools] > [Device Repository...]

> Window [Device Repository] appears.

► Click on [Install...].

> Window [Install Device Description] appears.

► Select the device description of the EtherCAT slaves and click on [Open].

> CODESYS installs the device in the device repository.

2 Add EtherCAT slave

► In standard branch of the device tree: Right-click on [EtherCAT_Master (EtherCAT Master)]

► In the context menu: Select [Add device...]

> Window [Add Device] appears.

► Select the requested EtherCAT slave in the [Device] area.

► Click on [Add Device] to add the selected device to the project.

► Click on [Close] to close the window.

> CODESYS adds the EtherCAT slave to the project.

3 Configure EtherCAT slave

► In standard branch of the device tree: Double click on the added EtherCAT slave.

> The editor window shows the configuration options of the device.

► Configure EtherCAT slave as requested.

6.8.3 Add fieldbus stack

17702

► Familiarise yourself with the following CODESYS functions!
- Modbus configurator
  → Online help > Fieldbus support > Modbus configurator

To declare the Ethernet interface as a fieldbus interface:

1 Create/load CODESYS project

► Create or load CODESYS project with the SmartPLC SafeLine AC4S.

2 Add fieldbus stack

► In the device tree: Right-click on [X3 (Ethernet)] or [X8 (Ethernet)].

► In the context menu: Select [Add device...].

> Window [Add Device] appears.

► Set the following values:
1. [Vendor]: Select [<All vendors>].
2. In table: Select the requested fieldbus stack in the [Name] column.
3. [Name]: Enter name of the fieldbus stack.

► Click on [Add Device] to add the requested fieldbus stack to the project.

> In the device tree: CODESYS adds the selected fieldbus stack as sub-element of the Ethernet interface.

3 Configure the fieldbus stack

► Configure added fieldbus device.

► Save the project to apply changes.

7 Programming

Contents

Objects of a standard PLC application 47
Create standard PLC application 48
Access standard input and output data 50
Use functions of the ifm package 57
Use visualisations 64
Cross communication 68
Configure task processing 70

7074

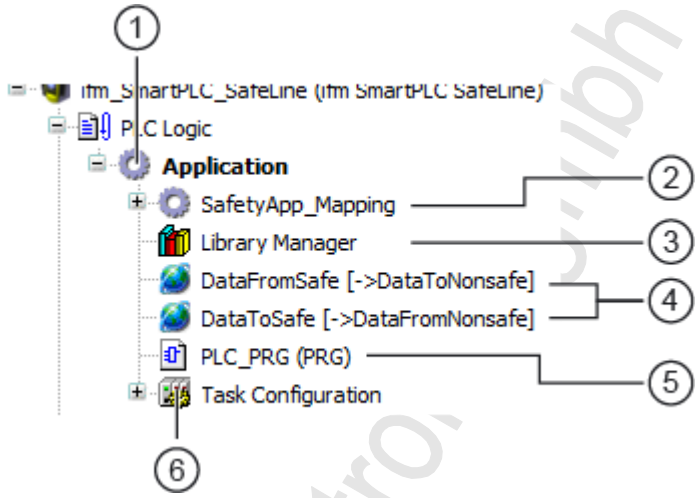
This chapter provides information about the programming of the standard PLC of the device.

► Familiarise yourself with the programming according to the standard IEC 61131-3!

7.1 Objects of a standard PLC application

18965

All objects of a standard PLC application are listed as subelements of the node [Application] in the standard branch of the device tree. In the basic configuration a standard PLC application contains the following objects:



(1) [Application] is the container of the standard PLC application

(2) [SafetyApp_Mapping] is a helper application for the data exchange between standard PLC and the fail-safe PLC (→ **Set interval of the data exchange** (→ p. 86))

(3) [Bibliotheksverwalter] provides access to the standard and device-specific function libraries

(4) [DataFromSafe] and [DataToSafe] are global variable lists for the data exchange between the standard PLC and the fail-safe PLC (→ **Data exchange between standard PLC and fail-safe PLC** (→ p. 83))

(5) [PLC_PRG (PRG)] provides access to the program editor of the standard application (→ **Create standard PLC application** (→ p. 48))

(6) [Taskkonfiguration] provides access to the settings of the task processing (→ **Configure main task** (→ p. 70))

7.2 Create standard PLC application

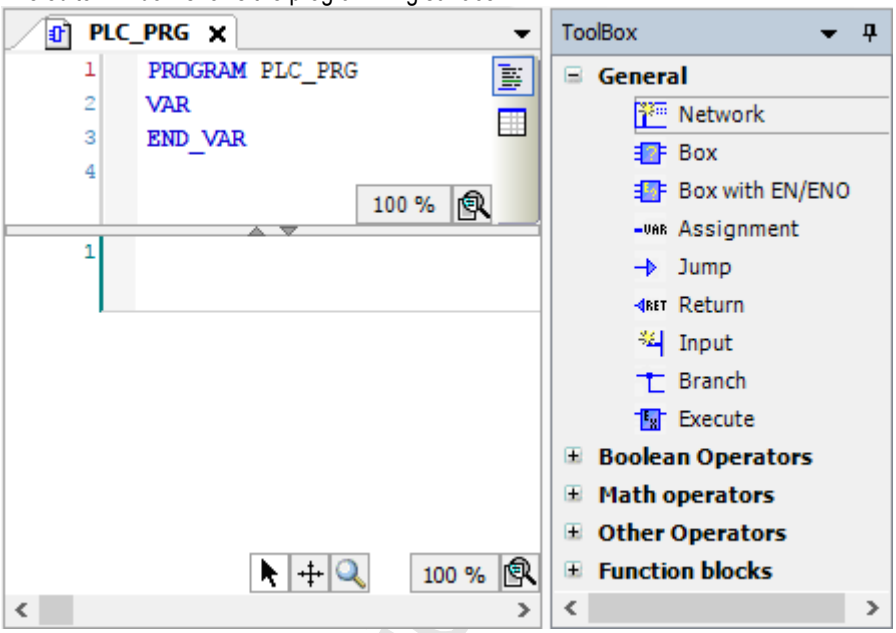
17691

► Familiarise yourself with the following CODESYS functions!
- Program application
  → Online help > CODESYS Development System > Program application
- Programming reference
  → Online help > CODESYS Development System > reference programming

To create a standard PLC application:

► In standard branch of the device tree: Double-click on [PLC_PRG (PRG)]

> The editor window shows the programming surface:



► Program the application in the editor.

7.2.1 Use remanent variables

18522

The standard PLC of the device supports the use of remanent variables. Variables declared as VAR RETAIN are stored in a memory area that is also maintained when the device is switched off.

The declaration of a variable as RETAIN also influences its behaviour when the standard PLC application is reset (→ **Supported reset variants** (→ p. 111)).

The memory area for RETAIN variables comprises 4072 bytes.

► Pay attention to the maximum size of the RETAIN memory area when declaring RETAIN variables!

7.2.2 Supported programming languages

18034

The following table shows which programming languages according to IEC 61131 are supported by the ifm function libraries:

Library	Function Block Diagram (FBD)	Sequential Function Chart (SFC)	Instruction List (IL)	Continuous Function Chart (CFC)	Ladder Diagram (LD)	Structured Text (ST)
ACnnnn_Utils.library	X	X	X	X	X	X
ACnnnn_SYS_CMD.library	X	X	X	X	X	X

Legend:
X ... is supported

7.2.3 Change system time of the device

18271

⚠ WARNING

Risk of undesired system behaviour!

The use of the CODESYS function `SysTimeRtcSet` for setting the time may lead to malfunction.

► To set the system time (date, time) of the device only use the following device-specific commands:
- Function block `Set_DateTime` (→ **Set_TimeDate** (→ p. 168))
- System command 0x1109 with function block ACnnnn_SysCmd (→ **ACnnnn_SysCmd** (→ p. 178))

7.3 Access standard input and output data

Contents

Options to access input and output data 50
Validity of the interface data 51
Process data of the AS-i slaves 52
Fieldbus data 54
Process data of the non-safe local devices 56

11185

► Familiarise yourself with the following CODESYS functions!
- Addresses according to IEC standard 61131-3:
  → Online help > CODESYS Development System > Programming Reference > Operanden > Addresses
- Access to IEC address via AT declaration:
  → Online help > CODESYS Development System > Programming Reference > Declaration > AT Declaration
- Definition of an ALIAS for an IEC address:
  → Online help > CODESYS Development System > Programming Reference > Data Types > References
- Coupling of a program variable to an address (mapping):
  → Online help > CODESYS Development System > Configuring I/O Links

7.3.1 Options to access input and output data

17621

In a CODESYS project, each input and output has a physical address according to the IEC standard (e.g. %IW5). CODESYS offers the following options to access this address from a standard PLC application and thereby to access the input and outputs data of the device:

- Direct access to IEC address
- Access to IEC address via AT declaration
- Definition of an ALIAS for an IEC address
- Link a program variable to an IEC address (mapping)

The IEC addresses (%I, %Q) of the inputs and outputs change if additional safe AS-i slaves are inserted in the CODESYS project.

► Always use symbolic programming for access to the inputs and outputs.

7.3.2 Validity of the interface data

18413

In order to facilitate the access to inputs and outputs of AS-i slaves, SmartPLC SafeLine AC4S projects offer clearly defined interfaces in the device tree (→ **Overview: Project structure with SmartPLC SafeLine AC4S** (→ p. 19)).

Depending on the active instance for accessing the outputs of the AS-i slaves (Manual, Gateway, PLC), the CODESYS data mapper only updates certain address areas of the interfaces. The following table shows which address areas of the i/o interfaces provide valid data values while in a certain operating mode:

Output control	I/O interfaces	Updated address areas / channels
Manual	[ASi_Master_1]	AS-i 1 Input (%IB, %IW)
	[ASi_Master_2]*	AS-i 2 Input (%IB, %IW)
	[Fieldbus_Interface]	▪ AS-i 1 Output (%IB, %IW) ▪ AS-i 2 Output (%IB, %IW)
Gateway	[ASi_Master_1]	ASi 1 Input (%IB, %IW)
	[ASi_Master_2]*	AS-i 2 Input (%IB, %IW)
	[Fieldbus_Interface]	▪ AS-i 1 Output (%IB, %IW) ▪ AS-i 2 Output (%IB, %IW)
PLC	[ASi_Master_1]	▪ AS-i 1 Input (%IB, %IW) ▪ AS-i 1 Output (%QB, %QW)
	[ASi_Master_2]*	▪ AS-i 2 Input (%IB, %IW) ▪ AS-i 2 Output (%QB, %QW)
	[Fieldbus_Interface]	▪ AS-i 1 Output (%IB, %IW) ▪ AS-i 2 Output (%IB, %IW)

* ... only available for devices with 2 AS-i masters

► When linking variables with inputs and outputs, only use interfaces in the project tree, that are updated by the CODESYS data mapper!

7.3.3 Process data of the AS-i slaves

17584

The project tree offers direct access to the cyclically updated process images of the inputs and outputs of the AS-i slaves.

(1) Digital input and output data of the slave at AS-i master 1: → **Digital input and output data** (→ p. 53)

(2) Analogue input and output data of the slaves at AS-i master 1: → **Analogue input and output data** (→ p. 53)

(3) Digital input and output data of the slave at AS-i master 2: → **Digital input and output data** (→ p. 53)

(4) Analogue input and output data of the slaves at AS-i master 2: → **Analogue input and output data** (→ p. 53)

Consider validity of the interface data (→ **Validity of the interface data** (→ p. 51))!

The function library `ACnnnn_Utils.library` contains the complex variable `ASi_NET`. The variable represents all inputs and outputs of a completely developed AS-i network. The programmer can use this data structure to store the process images of the inputs and outputs of an AS-i network. (→ **ASI_NET (STRUCT)** (→ p. 174))

Digital input and output data

17625

To access the digital process data of the slaves at AS-i master 1:

► In the device tree: Double click on [ASi_1_binaryIO]

> The editor window shows a structured list of the digital inputs and outputs of the AS-i slaves.

Variable	Mapping	Channel	Address	Type
		ASi Input bin	%IB1	
		Digital S(A) Slave	%IB1	ARRAY [1..31] OF BYTE
		Digital B Slave	%IB32	ARRAY [1..31] OF BYTE
		ASi Output bin	%QB1	
		Digital S(A) Slave	%QB1	ARRAY [1..31] OF BYTE
		Digital B Slave	%QB32	ARRAY [1..31] OF BYTE

► In column [Variable]: Mouse click on ⊞ to make individual variables visible.

To access the digital process data of the slaves at AS-i master 2 in a system with 2 AS-i masters:

► Double click on [ASi_2_binaryIO]

Analogue input and output data

17626

To access the analogue process data of the slaves at AS-i master 1:

► In the device tree: Double-click on [ASi_1_analogIO]

> Editor window shows a structured list of the analogue inputs and outputs of the AS-i slaves.

Variable	Mapping	Channel	Address	Type
		ASi Input	%IW32	ARRAY [1..31] OF SLAVEaANAaINaTYPE
		ASi Output	%QW32	ARRAY [1..31] OF SLAVEaANAaOUTaTYPE

► In column [Variable]: Mouse click on ⊞ to make individual variables visible.

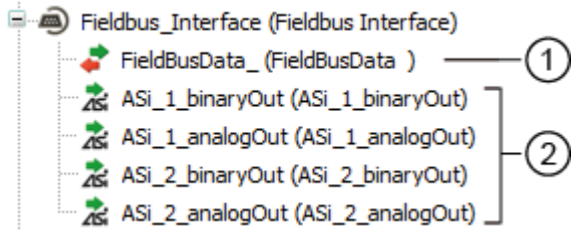
To access the analogue process data of the slaves atAS-i Master2 in a system with 2 AS-i masters:

► Double-click on [ASi_2_analogIO]

7.3.4 Fieldbus data

17585

The device tree offers direct access to the data that is transmitted between fieldbus and device.



(1) Data that the fieldbus transmits/receives via the fieldbus.
→ **Input and output data of the fieldbus interface** (→ p. 54)

(2) Output data of the AS-i slaves at AS-i master 1 and AS-i master 2 that is transmitted by the higher-level fieldbus PLC.
→ **Outputs data of the AS-i slaves** (→ p. 54)

Consider validity of the interface data (→ **Validity of the interface data** (→ p. 51))!

Input and output data of the fieldbus interface

17619

The input and output data of the fieldbus interface os transmitted in cycles of 120 words each. The programmer can access this data via IEC addresses.

To access the input and output data of the fieldbus interface:

► Device window: In the project tree, double-click on [FieldBusData_]

> Editor window shows a structured list of the inputs and outputs:

Variable	Mapping	Channel	Address	Type
⊞		Inputs from fieldbus	%IW373	ARRAY [0..119] OF WORD
⊞		Outputs to fieldbus	%QW373	ARRAY [0..119] OF WORD

► In column [Variable]: Mouse click on ⊞ to make individual variables visible.

Outputs data of the AS-i slaves

17620

The area contains all data, the higher-level Fieldbus controller cyclically sends to the outputs of the AS-i slaves via the fieldbus network. The data is structure like an AS-i network. The programmer can access this data via IEC addresses.

If the output access of the device is set to "PLC", the programmer can use the data bundled in this area to process the target values sent by the higher-level fieldbus controller to the CODESYS standard PLC.

Digital output data

17630

To access the digital output data of the slaves at AS-i master 1:

► Device window: In the project tree, double-click on [ASi_1_binaryIO]

> Editor window shows a structured list of the digital output data:

Variable	Mapping	Channel	Address	Type
		FB ASi Output bin	%IB986	
		Digital S(A) Slave	%IB986	ARRAY [1..31] OF BYTE
		Digital B Slave	%IB1017	ARRAY [1..31] OF BYTE

► In column [Variable]: Mouse click on ⊞ to make individual variables visible.

To access the digital output data of the slaves at AS-i master 2 in a system with 2 AS-i masters:

► Double-click on [ASi_2_binaryIO]

Analogue output data

17631

To access the analogue output data of the slaves at AS-i master 1:

► Device window: In the project tree, double-click on [ASi_1_analogOut]

> Editor window shows the structured list of the analogue output data:

Variable	Mapping	Channel	Address	Type
		FB ASi Output ana	%IW524	ARRAY [1..31] OF SLAVEaANAaINaTYPE

► In column [Variable]: Mouse click on ⊞ to make individual variables visible.

To access the analogue output data of the slaves at AS-i master 2 in a system with 2 AS-i masters:

► Double-click on [ASi_2_analogOut]

7.3.5 Process data of the non-safe local devices

8999

Access to non-safe inputs and outputs of the local I/O interface is made only via the fail-safe PLC of SmartPLC SafeLine AC4S.

- To access the input/output data of the non-safe devices at the local I/O interface:
  → **Local I/O interface: Access non-safe inputs/outputs** (→ p. 79)
- To transfer process data between standard PLC and fail-safe PLC of SmartPLC SafeLine AC4S:
  → **Data exchange between standard PLC and fail-safe PLC** (→ p. 83)

The process data of the non-safe inputs and outputs of the local I/O interface are not automatically updated on node [local_IO_].

7.4 Use functions of the ifm package

18489

The CODESYS package "CODESYS for ifm SmartPLC SafeLine" offers different functions for the programming of the device-internal CODESYS controller. In the following sections, these functions will be briefly described. To make orientation easier, the functions are grouped according to corresponding subjects and provided with a cross-reference to a detailed explanation in the document's appendix.

7.4.1 Control interface of the ifm function blocks

17660

All function blocks (FB) of the libraries ACnnnn_Utils.library and ACnnnn_SYS_CMD.library have inputs and outputs for control signals. The inputs activate the execution of the function block. The outputs provide information about the internal condition of the function block. Thanks to the signals, the programmer can create a control structure for a targeted processing of the FB and react to possible errors.

Number and designation of the FB control signals provide information about the type of FB execution:

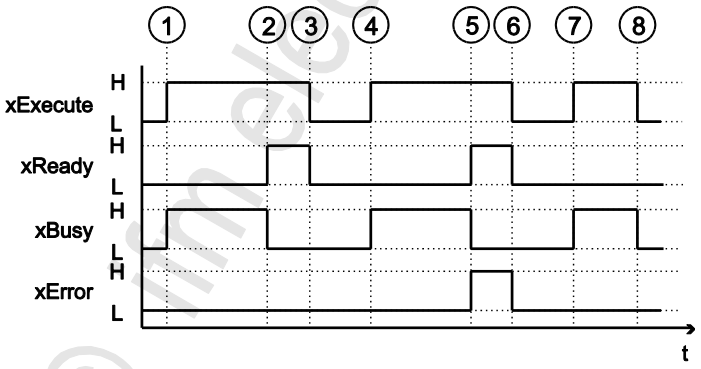
FB with one-time execution

17140

These function blocks perform their function exactly once after activation. To execute the function once again, the FB needs to be reactivated. FBs of this kind feature a control interface with the following inputs and outputs:

Designation	Type	Data type	Description	Possible values	
xExecute	Input	BOOL	Control FB execution	FALSE	Stop FB execution
				TRUE	Start FB execution
xReady	Output	BOOL	Indication of whether execution of the FB has been completed	FALSE	FB execution not yet completed
				TRUE	FB execution completed
xBusy	Output	BOOL	Indication of whether FB is active	FALSE	Function block is inactive
				TRUE	FB is active
xError	Output	BOOL	Indication of whether faults have occurred during execution of the FB	FALSE	FB executed correctly
				TRUE	Error occurred during execution of the FB
wDiagnostic	Output	WORD	Error code	FB specific	

The following figure shows the relation between the connections of the control signals:



①	xExecute = TRUE: xBusy = TRUE:	Rising edge (FALSE → TRUE) starts execution of the FB. FB execution has been started, but has not yet been completed.
②	xReady = TRUE: xBusy = FALSE: xError = FALSE:	FB execution completed; there are valid values on the data outputs. FB is no longer active. FB execution without faults.
③	xExecute = FALSE:	All signal outputs are set to FALSE and all internal states are reset.
④	→ ①	
⑤	xReady = TRUE: xBusy = FALSE: xError = TRUE:	FB execution is terminated. FB is no longer active. Errors occurred during FB execution; wDiagnostic provides error code.
⑥	→ ③	
⑦	→ ①	
⑧	xExecute = FALSE:	FB execution interrupted prior to completion; All signal outputs are set to FALSE and all internal states are reset.

FB with cyclic execution

17141

Function blocks which, when activated, cyclically perform their function until they are deactivated have the following control inputs and outputs:

Designation	Type	Data type	Description	Possible values	
xEnable	Input	BOOL	Control FB execution	FALSE	Stop FB execution
				TRUE	Start FB execution
xActive	Output	BOOL	Indication of whether execution of the FB has been completed	FALSE	FB execution not yet completed
				TRUE	FB execution completed
xError	Output	BOOL	Indication of whether faults have occurred during execution of the FB	FALSE	FB executed correctly
				TRUE	Error occurred during execution of the FB
wCycleCount	Output	WORD	Counters for the FB cycles	Integer value (hexadecimal representation)	
wDiagnostic	Output	WORD	Error code	FB specific	

7.4.2 Configure system

17450

To configure the system of the device, use the following function blocks:

Name	Description	Reference
QuickSetupASi_Master	Execute quick setup routine on an AS-i master	→ **QuickSetupASi_Master** (→ p. 166)
Set_TimeDate	Set system time (date, time) of the system	→ **Set_TimeDate** (→ p. 168)
Get_FieldbusInfo	Read fieldbus type, the status of the field bus connection and the parameters of the fieldbus interface	→ **Get_FieldbusInfo** (→ p. 164)

7.4.3 Configure AS-i master

17448

To configure the AS-i masters of the device, use the following function blocks:

Name	Description	Reference
Set_Mode	Set operating mode of the AS-i master (projecting mode or protected operation)	→ **Set_Mode** (→ p. 150)
Set_ASi_Config	Set diagnostic functions of the AS-i master (double address recognition, earth fault detection)	→ **Set_ASi_Config** (→ p. 146)
Set_AdressMode	Set automatic addressing of the AS-i master	→ **Set_AddressMode** (→ p. 144)

7.4.4 Configure AS-i slaves

17449

To configure the AS-i slaves, that are connected to the device, use the following function blocks:

Name	Description	Reference
Set_SlaveAddress	Change address of an AS-i slave	→ **Set_SlaveAddress** (→ p. 155)
Set_SlaveParameter	Change I/O configuration and ID codes (IO, ID, ID1, ID2) of an AS-i slave	→ **Set_SlaveParameter** (→ p. 159)
Set_SlaveExtendedID1	Extended ID1 of an AS-i slave	→ **Set_SlaveExtendedID1** (→ p. 157)

7.4.5 Manage AS-i network

17126

To manage the AS-i networks controlled by SmartPLC SafeLine AC4S, use the following function blocks:

Use complex variables

18528

There are different complex variables (STRUCT) at the programmer's disposal. They bundle logically associated data sets. Thereby, they facilitate the organisation of the data storage in the application and at the same time reduce the error rate when variables are declared.

The following complex variables are available:

Name	Description	Reference
ASI_NET	The complex variable contains the complete process image (inputs and outputs) of an AS-i network.	→ **ASI_NET (STRUCT)** (→ p. 174)
ASI_DATA	The complex variable contains the following components: ▪ Slave lists (LPS, LDS, LAS, LPF, LCE, LCEMS, LCEAS, LDAE) ▪ Parameter images (PI, PP) ▪ Configuration data of the AS-i slaves (CDI, PCD)	→ **ASI_DATA (STRUCT)** (→ p. 172) → **Get_ASi_Data** (→ p. 161)

Change network settings

17568

Name	Description	Reference
Set_ProjectAll	Execute projection adaptation on one AS-i master	→ **Set_ProjectAll** (→ p. 154)
Set_LPS	Change list of the projected slaves (LDS)	→ **Set_LPS** (→ p. 148)
Set_PCD	Change permanent projecting data (IO, ID, ID1, ID2) of all slaves on the AS-i master	→ **Set_PCD** (→ p. 152)

Read network settings

18532

To read the network settings cyclically and offer them in the application:

Name	Description	Reference
Get_ASi_Data	Read the following datasets for network management in batches and cycles: ▪ List of activated slaves (LAS) ▪ List of detected slaves (LDS) ▪ List of projected slaves (LPS) ▪ List of peripheral faults (LPF) ▪ List of configuration errors (LCE) ▪ List of configuration errors, missing slaves (LCEMS) ▪ List of configuration errors - additional slaves (LCEAS) ▪ List of double address errors (LDAE) ▪ Configuration data image (CDI) ▪ Permanent configuration data (PCD) ▪ Input parameters (PI) ▪ Output parameters (PP)	→ **Get_ASi_Data** (→ p. 161)

Alternatively, this data can be read separately with the following FB:

Read parameter images

17569

Name	Description	Reference
Get_InputParameter	Read parameters of the inputs of the slaves at the AS-i master (PI)	→ **Get_InputParameter** (→ p. 140)
Get_OutputParameter	Read parameters of the outputs of the slaves on the AS-i master (PP)	→ **Get_OutputParameter** (→ p. 142)

read slave lists

18530

Name	Description	Reference
Get_LPS	Read list of projected slaves (LPS)	→ **Get_LPS** (→ p. 124)
Get_LDS	Read list of detected slaves (LDS)	→ **Get_LDS** (→ p. 122)
Get_LAS	Read list of activated slaves (LAS)	→ **Get_LAS** (→ p. 120)
Get_LPF	Read list of peripheral faults (LPF)	→ **Get_LPF** (→ p. 134)
Get_LCE	Read list of configuration errors (LCE)	→ **Get_LCE** (→ p. 126)
Get_LCEMS	List of configuration errors - read missing slaves (LCEMS)	→ **Get_LCEMS** (→ p. 130)
Get_LCEAS	Read of the configuration errors - read additional slave (LCEAS)	→ **Get_LCEAS** (→ p. 128)
Get_LDAE	Read list of double address errors (LDAE)	→ **Get_LDAE** (→ p. 132)

Read configuration data of the slaves

18533

Name	Description	Reference
Get_CDI	Read configuration data image (IO, ID, ID1, ID2) of all slaves on the AS-i master	→ **Get_CDI** (→ p. 136)
Get_PCD	Read permanent configuration data of all slaves (IO, ID, ID1, ID2) on the AS-i master	→ **Get_PCD** (→ p. 138)

Read status of the voltage supply

18529

Name	Description	Reference
Get_ASi_PHY_Dat	Determine voltage supply status of the AS-i network	→ **Get_ASi_PHY_Dat** (→ p. 118)

7.4.6 Send commands to the system and the AS-i master

17659

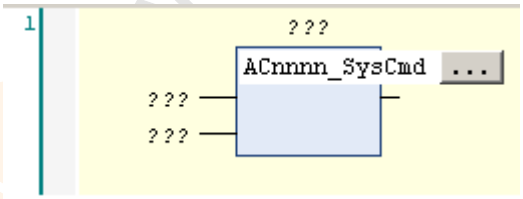
Similar to the acyclic transmission command channels and data sets of the device, the programmer can send commands to the system or an AS-i master with the FB ACnnnn_SysCmd (→ **ACnnnn_SysCmd** (→ p. 178)).

- System command overview: → **Table: System commands** (→ p. 179)
- Overview AS-i master commands: → **Table: AS-i master commands** (→ p. 180)

By default, the FB ACnnnn_SysCmd is hidden. To add the FB to a program module:

► Highlight the required network and add an empty function block with [FBD/LD/IL] > [Insert Empty Block].

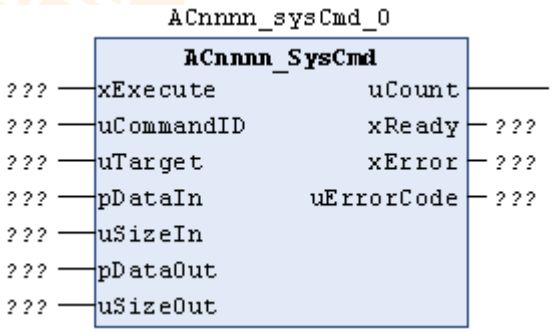
> Network shows empty FB.



► Double-click on the name field of the FB

► Enter designation `ACnnnn_SysCmd` and confirm with [ENTER].

> FB has inputs and outputs of the `ACnnnn_SysCmd`.



► Adjust inputs and outputs of the FB in accordance with the required command.

7.5 Use visualisations

Contents

Supported visualisation types....64
Add visualisation to a project....65
Create a visualisation....66
Configure visualisation....67

17059

► Familiarise yourself with the following CODESYS functions!
- Visualisations in CODESYS
  → Online help > CODESYS visualisation

7.5.1 Supported visualisation types

11022

The device supports the following CODESYS visualisation types:

- **Web visualisation (WebVisu)**
  A WebVisu allows graphic representation of selected process and control data in a web browser by means of an application-specific visualisation.

7.5.2 Add visualisation to a project

10930

To add a visualisation to a CODESYS project:

► open CODESYS project.
OR:
Create new CODESYS project. (→ **Create new project with SmartPLC SafeLine AC4S** (→ p. 17))

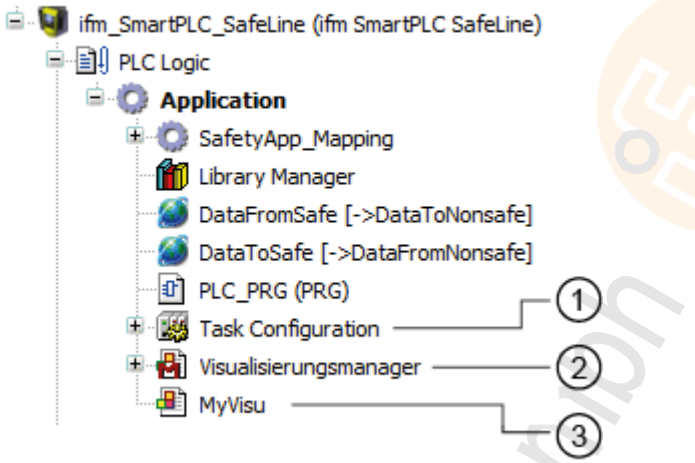
► In standard branch of the device tree: Select [Application].

► Select [Project] > [Add Object] > [Visualization...].

> Window [Add Visualization] appears.

► Enter a name for the visualisation in the field [Name] and click on [Add] to apply it.

> CODESYS adds the following elements to the standard branch of the device tree:



(1) [VISU_TASK]: Set the visualisation task properties: → **Set parameters for visualisation task** (→ p. 70)

(2) [Visualization Manager]: Set properties of the visualisations: → **Change properties of the web visualisation** (→ p. 67)

(3) [MyVisu]: Area for the creation of the visualisation objects: → **Create a visualisation** (→ p. 66)

7.5.3 Create a visualisation

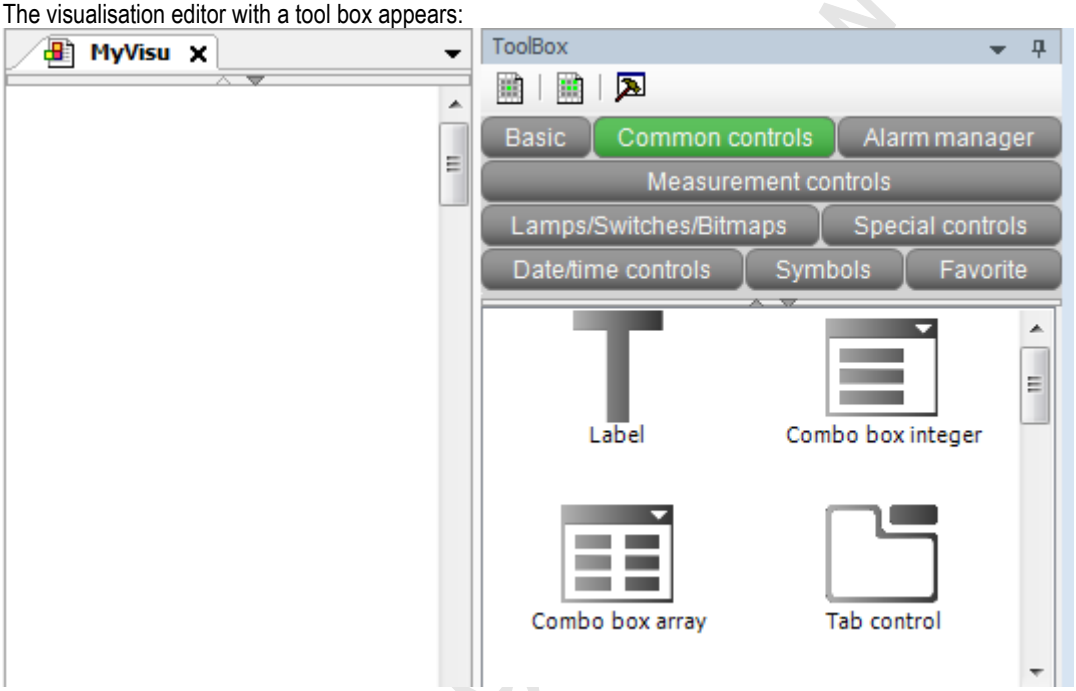
17061

Create a seperate visualisation object for each web visualisation.

To create a visualisation for a standard PLC application:

► In the standard branch of the device tree: double-click on [Visualization]

> The visualisation editor with a tool box appears:



► Create the visualisation using the tools.

► Save the project to apply changes.

7.5.4 Configure visualisation

11027

To configure the characteristics of the created visualisation select one of the following options:

- **Change properties of the web visualisation** (→ p. 67)

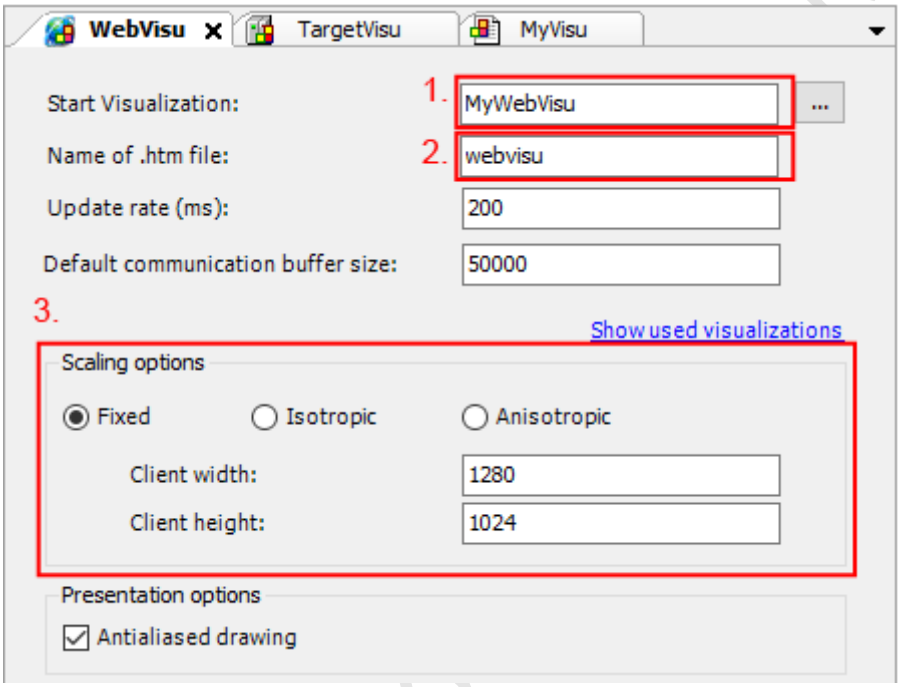
Change properties of the web visualisation

17065

To change the attributes of the web visualisation:

► In standard branch of the device tree: Double click on [Web-Visualisierung]

> The editor window shows attributes of the web visualisation:



► Set the following values:
1. Field [Start Visualization]: Select the created web visualisation.
2. Field [Name of .htm file]: Enter name for HTML file (→ Note).
3. Area [Scaling options]: Enter fixed width and height as shown.

In the field [Name of .htm file] enter the name by which the web visualisation is to be accessible in the web browser (→ **Display web visualisation** (→ p. 114)).

► Use only lower case when entering the name!

► Save the project to apply changes.

7.6 Cross communication

Contents

Use network variables 68

10293

► Familiarise yourself with the following CODESYS functions!
  ▪ → Online help > CODESYS Development System > Exchange data in the network > Network variables

The device supports the CODESYS function "network variables". The user can use it to transmit data between the standard PLCs of a project.

Network variables are transmitted according to the broadcast principle. A transmitter can send data to several receivers.

7.6.1 Use network variables

16189

► Observe the notes on the configuration of the Ethernet configuration interfaces 1 (X3) and 2 (X8). → **Change IP settings** (→ p. 40)

The network variables must only be transmitted in the subnet of the Ethernet interface via which the device cross communicates. The programmer must change the broadcast address of the network variable list accordingly.

Example:

- Ethernet interface: X3
- IP address of the Ethernet interface: 192.168.1.10
- Broadcast address to be set: 192.168.1.255

To create the cross communication between standard PLCs:

Requirements:

> SmartPLC SafeLine AC4S 1 and SmartPLC SafeLine AC4S 2 are participants of the same CODESYS project (→ **Add additional device to the project** (→ p. 41)).

> The SmartPLC SafeLine AC4S devices are coupled via the Ethernet network.

1 Set the network variable list of the transmitter

► In the standard branch of the device tree of the SmartPLC SafeLine AC4S 1: Select [PLC Logic] > [Application].

► Select [Project] > [Add Object] > [Network Variable List (Sender)...]

> Dialogue window appears.

► Set the following parameters as requested
1. [Name]: Unique name of the variable list
2. [Network type]: UDP
3. [Settings...]: Broadcast address (→ Note)
4. [Task]: Requested task

► Click on [Add] to apply the selected values.

> CODESYS creates the network variable list of the transmitter.

> The network variable list appears in the device tree as sub-node of the standard application.

2 Declare network variables

► In standard branch of the device tree: Double click on the network variable list of the transmitter

> The editor window shows the variable declaration.

► Declare network variables.

3 Create network variable list of the receiver

► In the standard branch of the device tree of the SmartPLC SafeLine AC4S 2: select [PLC Logic] > [Application].

► Select [Project] > [Add Object] > [Network Variable List (Receiver)...]

> Dialogue window appears.

► Set the following parameters as requested
1. [Name]: Unique name of the variable list
2. [Task]: Requested task
3. [Sender]: variable list of the transmitter

► Click on [Add] to apply the selected values.

> CODESYS creates the network variable list of the receiver.

> The network variable list appears in the device tree as sub-node of the standard application.

4 Optional: Create more receiver lists

► Repeat step 3 to connect additional devices to the variable list of the transmitter.

7.7 Configure task processing

4109

► Familiarise yourself with the following CODESYS functions!
  - Task configuration
    → Online help > CODESYS Development System > application programmable > task configuration

The processing of the tasks is controlled by parameters. The user can set the parameters for each task separately

CODESYS automatically creates the following tasks and visualisations during project creation:

Name	Description	Note
[MainTask]	Configuration of the main task (e.g. for main program [PLC_PRG (PRG)])	→ **Configure main task** (→ p. 70)
[VISU_TASK]	Configuration of the task for processing visualisation	→ **Set parameters for visualisation task** (→ p. 70)

7.7.1 Configure main task

18412

The basic settings of the task characteristics cover the requirements of many applications. In the event of non-optimum device performance the user must determine and set the optimum task characteristics himself.

To change parameters of a task:

► In standard branch of the device tree: Double click on [Taskkonfiguration] > [MainTask]

> The editor window shows the configuration of the main task.

► Set the parameters as requested.

> Selected value is applied.

7.7.2 Set parameters for visualisation task

17066

► Execute the visualisation task (VISU_TASK) with a priority that is as low as possible to avoid interruption of other tasks that are important for the core functions of the application.

► Execute the VISU_TASK in appropriate cyclic intervals to save the resources of the device-internal CODESYS standard PLC of the fieldbus network.

Each visualisation is executed separately from the program code in a separate task. To set the properties of the visualisation task:

► In the standard branch of the device tree: Double-click on [Task configuration] > [VISU_TASK]

> Editor window shows parameters of the visualisation task.

► Set the parameters as required.

► Save the project to apply changes.

8 Programming of the fail-safe PLC

Contents

Objects of a safe application 72
Create safe application 73
Use safety functions of the ifm AS-i package 77
Use safety functions of CODESYS 80
Data exchange between standard PLC and fail-safe PLC 83
Safe cross communication 87
Safe fieldbus communication 91
Set process safety time 93
Configure safety task 100

8689

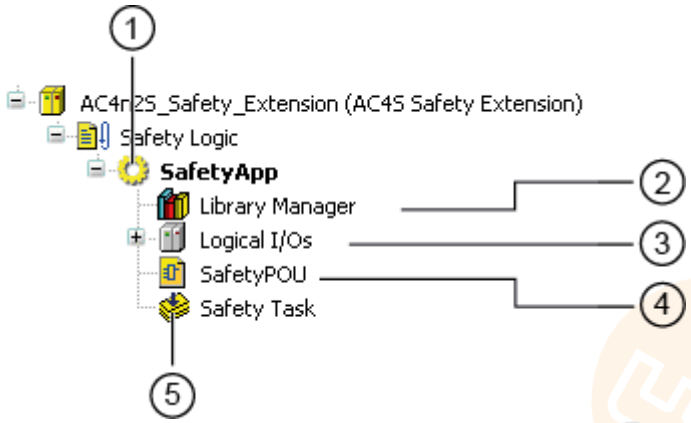
This chapter provides information about the programming of the fail-safe PLC of AC4S.

► Familiarise yourself with the following CODESYS functions!
- Safe applications
  → Online help > Add-Ons > CODESYS Safety > Device topology > Safety application
- Programming reference
  → Online help > Add-Ons > CODESYS Safety > Programming reference
- Concepts
  → Online help > Add-Ons > CODESYS Safety > Concepts

8.1 Objects of a safe application

7133

All objects of a safe application are listed as sub-elements of the node [SafetyApp] in the safety area of the device tree. In the basic configuration the safety-relevant application contains the following objects:



(1) [SafetyApp] is the container for objects of a safe application and provides access to the editor of the safety application object
→ **Pin safe project** (→ p. 76)

(2) [Library Manager] provides access to the certified safety libraries
→ **Use safety functions of CODESYS** (→ p. 80)
→ **Use safety functions of the ifm AS-i package** (→ p. 77)

(3) [Logical I/Os] contains following objects:

- Logical exchange devices for the data exchange between standard PLC and fail-safe PLC
  → **Data exchange between standard PLC and fail-safe PLC** (→ p. 83)
- Logical Devices of the inserted safe AS-i slaves
  → **Configure safe AS-i slaves** (→ p. 25)
- Logical devices of the safe devices at local I/O interface
  → **Configure safe devices at local I/O interface** (→ p. 32)

(4) [SafetyPOU] contains the program code of the safe application
→ **Create safe application** (→ p. 73)

(5) [Safety Task] provides access to the configuration options of the safety task
→ **Configure safety task** (→ p. 100)

8.2 Create safe application

Contents

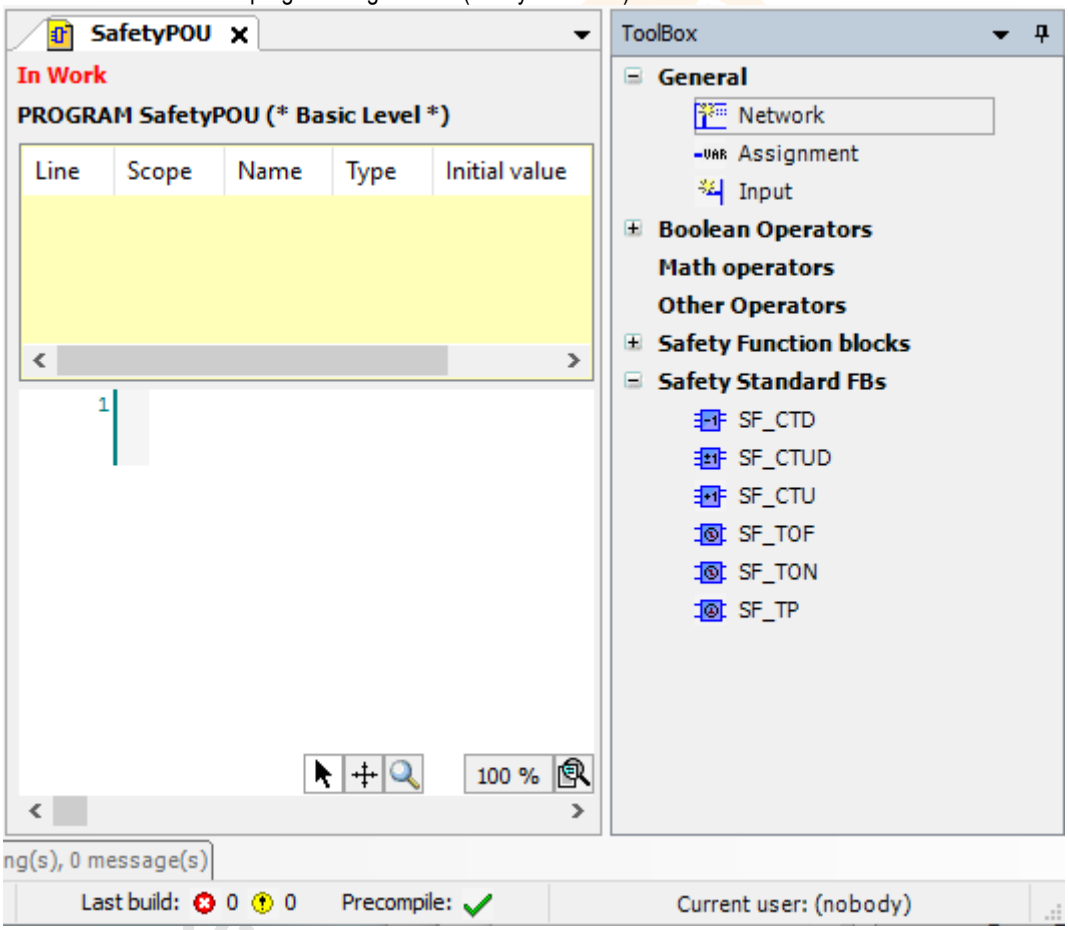
Supported program organization units (POUs) 74
Available safety libraries 74
Supported programming languages 75
Minimum number of AS-i slaves 75
Variable declaration with data transfer between standard and safety area 76
Available memory space 76
Pin safe project 76

6956

To create a safe application:

► In the safety extension area of the device tree:
Double click on [SafetyPOU]

> Editor window shows the programming surface (safety FB editor):



8.2.1 Supported program organization units (POUs)

19566

The programming code is entered in the POUs (program organization unit). CODESYS Safety provides 2 POU types for safe programming. They differ in the scope of usable functions.

Usable elements	Basic POU	Extended POU
General elements	Network, assignment, input	Network, assignment, input, jump, return
Boolean operators	AND, OR	AND, OR, XOR, NOT
Other operators	–	SEL, MUX
Mathematical operators	–	ADD, SUB, MUL, DIV, EQ, NE LT, LE, GT, GE
Safety standard functions (safetystandard.lib)	SF_CTD, SF_CTUD, SF_CTU, SF_TOF, SF_TON, SF_TP	SF_RS, SF_SR, SF_CTD, SF_CTUD, SF_CTU, SF_TOF, SF_TON, SF_TP, SF_F_TRIG, SF_R_TRIG
Safety PLCopen functions (safetyplcopen.lib)	All	All

For using extended POUs the developer must be a member of the user group "Safety.ExtendedLevel" (→ **Create user account** (→ p. 20)).

The use of extended POUs requires a more complex verification process due to the higher complexity of the program code.

► If possible, only use basic POUs to create a safe application.

The "SafetyPOU" created during the creation of the safety project is of type Basic POU (→ **Objects of a safe application** (→ p. 72)).

8.2.2 Available safety libraries

419

To create safe applications the programmer can use the following libraries with certified safety functions:

Library	Description	Reference
sf_io.library	Device-specific safety function blocks	→ **Use safety functions of the ifm AS-i package** (→ p. 77)
safetystandard.library	Safe standard function blocks (bistable function blocks, counters, timers, triggers)	→ **SafetyStandard library** (→ p. 80)
safetyplcopen.library	Safety function blocks of PLCopen Safety	→ **SafetyPLCopen library** (→ p. 81)
safetyfsoemaster.library	Function block for FSoE transmission	→ **SafetyFSoEMaster library** (→ p. 82)

8.2.3 Supported programming languages

19471

Following table shows which programming languages according to IEC 61131-3 are supported by certified safety libraries:

Library	Function block diagram (FBD)	Sequential function chart (SFC)	Instruction list (IL)	Continuous function chart (CFC)	Ladder diagram (LD)	Structured text (ST)
sf_io.library	X	–	–	–	–	–
safetystandard.library	X	–	–	–	–	–
safetyplcopen.library	X	–	–	–	–	–
safetyfsoemaster.library	X	–	–	–	–	–

Legend:
X ... is supported
– ... is not supported

8.2.4 Minimum number of AS-i slaves

18462

⚠ WARNING

Risk of data loss

Altogether 5 AS-i slaves min. have to be installed (standard or safety AS-i slaves) in each of the SmartPLC SafeLine AC4S-controlled AS-i network.

► If fewer than 5 AS-i slaves are part of the configuration of an AS-i network, provide the required number of AS-i slaves via virtual AS-i control slaves without function (→ **SF_OUTcontrol_ASi** (→ p. 234)).

8.2.5 Variable declaration with data transfer between standard and safety area

7075

To be able to use the variable value in an application from the other area the variable always has to be declared as exchange variable. Then the developer can use such a variable directly in the program code of the standard application or the safe application.

If the developer first of all creates a variable in the declaration part of the POU and then defines an exchange variable of the same name, CODESYS will generate an error message during compilation of the application.

Example:

Task:
A non-safe Boolean value from the standard application is to be used in the safe application.

How to proceed:

1. Declare exchange variable of type BOOL in the exchange device [DataFromNonsafe] and update variable list in [DataToSafe] (→ **Use the factory-set replacement variables** (→ p. 84)).
2. Assign a value to the variable in the standard application.
3. Use the exchange variable as source for FB or assignment in the safe application.

8.2.6 Available memory space

11571

The following memory location is available to the user for programming safe applications on the fail-safe PLC:

- IEC program code: 384 Kbytes
- IEC data: 128 Kbytes

8.2.7 Pin safe project

17169

► Familiarise yourself with the following CODESYS functions!
  - Pin object
    → Online help > Add-Ons > CODESYS Safety > Pinning

When a safe application has been completed, the project must be pinned. The states of all objects of the safe application are stored and saved against changes by means of a check sum.

8.3 Use safety functions of the ifm AS-i package

Contents

Safe AS-i slaves: Reset logical device 77
Safe AS-i slaves: Provide diagnostic information 78
Safe AS-i slaves: Transmit help signals HSI_1 and HSI_2 to safe AS-i output modules 78
Local I/O interface: Reset logical device 78
Local I/O interface: Provide diagnostic information 79
Local I/O interface: Access non-safe inputs/outputs 79
PLCopen Safety: Provide diagnostic information 79

9009

ifm electronic provides the `library SF_IO.lib`. It contains certified function blocks by means of which the programmer can create safe applications for the fail-safe PLC of AC4S. The library is automatically loaded by means of the AC4S template during creation of a safety project. The user can access the elements of the library via the library manager in the safety extension area of the device tree.

To add function blocks from the ifm library sf_io.lib to a network:

► Select network.

► Activate [Ctrl]+[B].

> The window [Input assistant] appears.

► Select the required FB at [SF_IO] and add it to the network with [OK].

8.3.1 Safe AS-i slaves: Reset logical device

601

To reset the logical device of a safe AS-i input slave from the locked error state and to change into the initialisation state the following function blocks are available:

Function block	Description	Reference
CtrlASi_InSlave	Reset safe AS-i input slave that is in the locked error state	→ **CtrlASi_InSlave** (→ p. 187)
CtrlASi_ResetAllSlaves	Reset all safe AS-i slaves of an AS-i network that are in the locked error state.	→ **CtrlASi_ResetAllSlaves** (→ p. 192)

The safe AS-i slave that is to be controlled by means of the FB CtrlASi_InSlave must be inserted in the device tree.

8.3.2 Safe AS-i slaves: Provide diagnostic information

429

To control the provision of the internal diagnostic information of a logical device the following function blocks are available:

Function block	Description	Reference
CtrlASi_InSlave	Provide diagnostic information of a safe AS-i input slave	→ **CtrlASi_InSlave** (→ p. 187)
CtrlASi_OutSlave	Provide diagnostic information of an AS-i control slave for safe outputs	→ **CtrlASi_OutSlave** (→ p. 189)

The safe AS-i slave that is to be controlled by means of the FB must be inserted in the device tree.

8.3.3 Safe AS-i slaves: Transmit help signals HSI_1 and HSI_2 to safe AS-i output modules

427

To transmit help signals HSI_1 und HSI_2 to the logical device of a safe AS-i control slave the following function block is available:

Function block	Description	Reference
CtrlASi_OutSlave	Transmit help signals HS1 and HS2 to AS-i control slave for safe outputs	→ **CtrlASi_OutSlave** (→ p. 189)

The safe AS-i control slave that is to be controlled by means of the FB must be inserted in the device tree.

A transmission of the two help signals HSI_1 and HSI_2 once started cannot be interrupted. Remarks about the status of the signal transmission are provided at the FB output `DiagCode`.

8.3.4 Local I/O interface: Reset logical device

434

To reset a logical device of a safe sensor at the local I/O interface from the locked error state the following function block is available:

Function block	Description	Reference
CtrlLocalInputs	Reset 2-channel peripheral at the local I/O interface that is in the locked error state.	→ **CtrlLocalInputs** (→ p. 194)

The safe device that is to be controlled by means of the FB must be inserted in the device tree.

8.3.5 Local I/O interface: Provide diagnostic information

428

To control the provision of the internal diagnostic information of the logical device of a safe sensor at the local I/O interface the following function block is available:

Function block	Description	Reference
CtrlLocalInputs	Provide diagnostic information of 2 safe inputs of the local I/O interface	→ **CtrlLocalInputs** (→ p. 194)

The safe device that is to be controlled by means of the FB must be added to the device tree.

8.3.6 Local I/O interface: Access non-safe inputs/outputs

426

To access the non-safe inputs and outputs of the local I/O interface the following function blocks are available:

Function block	Description	Reference
GetLocalInput	Read process data of a non-safe input of the local I/O interface	→ **GetLocalInput** (→ p. 198)
SetLocalOutput	Provide process data at a non-safe output of the local I/O interface	→ **SetLocalOutput** (→ p. 199)

The process data of the local inputs detected by means of the FB `GetLocalInput` is not safe (data type: BOOL).

► Do not implicitly use non-safe process data for a safe function.

To access non-safe process data of the local inputs and outputs with standard PLC of SmartPLC SafeLine AC4S:

► Use global exchange variables. (→ **Data exchange between standard PLC and fail-safe PLC** (→ p. 83))

8.3.7 PLCopen Safety: Provide diagnostic information

597

The function blocks of the PLCopen Safety library `SafetyPLCopen.lib` provide information about their internal state at the output `DiagCode`. To be able to use this diagnostic information in the safe application the following function block is available:

Function block	Description	Reference
Ctrl_SetDiagInfo	Provide diagnostic information of the function blocks of the safety PLCopen library SafetyPLCopen.lib in OSC	→ **Ctrl_SetDiagInfo** (→ p. 196)

The FB generates a separate message for each change of state in the online support centre (OSC). The ring memory of the OSC therefore reaches its capacity limits within a short time.

► Use FB Ctrl_SetDiagInfo only for debugging.

8.4 Use safety functions of CODESYS

15821

3S provide several libraries in the programming environment CODESYS Safety. They contain certified function blocks (FB) by means of which the programmer can create safe applications for the fail-safe PLC of AC4S. The libraries are automatically loaded during the creation of a safety project by means of AC4S (→ **Create new project with SmartPLC SafeLine AC4S** (→ p. 17)). The user can access the elements of the libraries via the library manager in the safety extension area of the device tree.

Following libraries are available:

- **SafetyStandard library** (→ p. 80)
- **SafetyPLCopen library** (→ p. 81)
- **SafetyFSoEMaster library** (→ p. 82)

8.4.1 SafetyStandard library

20249

The library `SafetyStandard.lib` provides safe versions of standard functions (counters, timers, triggers, bistable functions). The library contains the following functions:

Function block	Description
SF_RS	Safe bistable function block with dominant setting
SF_SR	Safe bistable function block with dominant resetting
SF_CTD	Safe counter (down)
SF_CTUD	Safe counter (up)
SF_TOF	Safe timer with switch-off delay
SF_TON	Safe timer with switch-on delay
SF_TP	Safe timer with clock generator
SF_F_TRIG	Safe trigger with falling-edge detection
SF_R_TRIG	Safe trigger with rising-edge detection

The developer is responsible for the use of the safety standard function blocks provided in CODESYS.

► Observe availability of the FBs (→ **Supported program organization units (POUs)** (→ p. 74))!

The complete documentation of the `SafetyStandard.lib` can be accessed via the online help of CODESYS Safety:
→ Online help > Add-Ons > CODESYS Safety > Libraries > SafetyStandard

8.4.2 SafetyPLCopen library

20248

The library `SafetyPLCopen.lib` contains the following certified function blocks:

Function block	Description
SF_Antivalent	Logical AND operator of 2 complementary signals; monitoring the discrepancy time
SF_EDM	Monitoring the basic status and the switching status of actuators that are controlled by safe output devices
SF_EmergencyStop	Monitoring an E-stop
SF_EnableSwitch	Evaluation of the signals of a manually operated, three-stage enabling switch
SF_Equivalent	Logical AND operator of 2 equivalent signals; monitoring the discrepancy time
SF_ESPE	Monitoring electro-sensitive protective equipment
SF_GuardLocking	Monitoring protective equipment with four-stage locking
SF_GuardMonitoring	Monitoring protective equipment with two-stage locking
SF_ModeSelector	Selection of an operating mode
SF_MutingPar	Parallel muting with 4 muting sensors
SF_MutingPar_2Sensor	Parallel muting with 2 muting sensors
SF_MutingSeq	Sequential muting with 4 muting sensors
SF_OutControl	Controlling a safe output with a non-safe signal and optional start disable
SF_SafetyRequest	Interface between user program and system environment
SF_TestabelSafetySensor	Periodic testing of a sensor / electro-sensitive protective equipment (ESPE) of type 2
SF_TwoHandControlTypeII	Function for "type 2 two-hand control"
SF_TwoHandControlTypeIII	Function for "type 3 two-hand control"

The developer is responsible for the use of the safety PLCopen function blocks provided in CODESYS.

The complete documentation of the `SafetyPLCopen.lib` can be accessed via the online help of CODESYS Safety:
→ Online help > Add-Ons > CODESYS Safety > Libraries > SafetyPLCopen

Detailed information about the specifications: → www.plcopen.org > PLCopen Safety

8.4.3 SafetyFSoEMaster library

11817

The library `SafetyFSoEMaster.lib` provides access to functions for the safe communication via EtherCAT (Fail-Safe over EtherCAT - FSoE). The library contains the following certified function blocks:

Function block	Description
FSoEMaster	Monitoring the safe transfer of data between EtherCAT master and EtherCAT slave (transmission / reception)

The developer is responsible for the use of the SafetyFSoEMaster function blocks provided in CODESYS.

The complete documentation of the library `SafetyFSoEMaster.lib` can be accessed via the online help of CODESYS Safety:
→ Online help > Add-Ons > CODESYS Safety > field buses > FSoE

8.5 Data exchange between standard PLC and fail-safe PLC

7085

standard PLC and fail-safe PLC of AC4S use a separate hardware. Data between the two areas is therefore exchanged via a special memory area to which both PLCs can access independently from each other.

The programmer can organise the data exchange between the areas via the CODESYS mechanism of the logical exchange variables.

► Familiarise yourself with the following CODESYS functions!
- GVL for logical exchange of data
  → Online help > Add-Ons > CODESYS Safety > Device topology > Logical I/Os > GVL for logical exchange
- Logical I/O for data exchange with the main controller
  → Online help > Add-Ons > CODESYS Safety > Device topology > Logical I/Os > Logical I/O for data exchange with the main controller

8.5.1 Remarks about the use of exchange variables

18482

The variable values between standard PLC and fail-safe PLC are transferred as non-safe data.

► Do not implicitly use non-safe variable values for safe functions.

If standard PLC is in STOP state, the declared exchange variables for the data transfer towards the safety PLC are written with the replacement value "0".

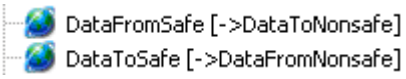
► Always transfer safe information created in the fail-safe area to the standard area unchanged.
The signal can then be further processed (e.g. negation) in the standard area

8.5.2 Use the factory-set replacement variables

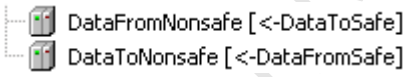
8961

When an SmartPLC SafeLine AC4S project is created by means of an SmartPLC SafeLine AC4S template, CODESYS automatically creates objects for the data exchange between standard PLC and the fail-safe PLC. For each transfer direction a coupled pair of global variable list (GVL) and logical exchange device each is generated.

GVL in the standard area:



Logical exchange devices in the safety area:



With these exchange objects of data type BYTE it is possible to transfer 8 bundled Boolean values in each direction.

To exclude access conflicts only one PLC has the write access to the logical exchange device for each exchange direction.

Following table shows the relations:

Global variable list (in standard branch of the device tree)	**Logical exchange device** (in the safety extension area of the device tree)	**Write access**
[DataToSafe]	[DataFromNonsafe]	standard PLC
[DataFromSafe]	[DataToNonsafe]	fail-safe PLC

Data transfer standard area >>> safety area

8962

To transfer data from the standard area of the device to the safety area of the device:

1 Declare exchange variables in the safety area

- ► In the safety extension area of the device tree:
  Double click on [DataFromNonsafe]
- > Editor window shows table for declaration of the exchange variables.
- ► Enter the designations for the exchange variables of data type BOOL in the column [Variable].

2 Map replacement variables with the standard area

- ► In standard branch of the device tree:
  Double click on [DataToSafe]
- > Editor window shows declaration part of the global variable list.
- ► Click on [Update].
- > CODESYS applies the declared exchange variable from [DataFromNonsafe] (step 1).
- > Declaration part shows the declared exchange variables.

The logical exchange devices must only be defined in the safety area. A change in the standard area is not permitted.

If variables are added or changed in the logical exchange device, it is mandatory for the developer to update the respective variable list (step 2).

Data transfer safety area >>> standard area

11187

To transfer data from the safety area of the device to the standard area of the device:

1 Declare exchange variables in the safety area

► In the safety extension area of the device tree:
Double click on [DataToNonsafe]

> Editor window shows table with currently declared variables.

► Enter the designations for the exchange variables of data type BOOL in the column [Variable].

Do not logically link non-safe with safe data to form a safe output signal

► In the standard area of the device tree:
Double click on [DataFromSafe]

> Editor window shows declaration part of the global variable list.

► Click on [Update].

> CODESYS applies the declared exchange variable from [DataToNonsafe] (step 1).

> Declaration part shows the declared exchange variables.

The logical exchange devices must only be defined in the safety area. A change in the standard area is not permitted.

If variables are added or changed in the logical exchange device, it is mandatory for the developer to update the respective variable list (step 2).

8.5.3 Create additional objects for data exchange

19570

If the capacities of the exchange variables created at the factory are not sufficient, the programmer can create additional exchange variables.

► Note the remarks (→ **Data exchange between standard PLC and fail-safe PLC** (→ p. 83))!

Only the following data types may be used for the data exchange between standard PLC and safety PLC: BYTE, WORD, LONG

Several safe values of data type BOOL can be transferred as a bundle in a variable of data type BYTE or WORD.

The transfer of arrays of exchange variables is not supported.

In each cycle of the safety PLC max. 256 bytes of data can be transferred in each direction (standard PLC >>> safety PLC, safety PLC >>> standard-SPS).

► When additional objects are created for the data exchange, observe the max. transferable data volume.

► Include the following objects in the calculation of the transfer data:

- safe cross communication (Safety NetVars): → **Set up fail-safe network variables** (→ p. 88)
- safe fieldbus communication (FSoE): → **Safe fieldbus communication** (→ p. 91)

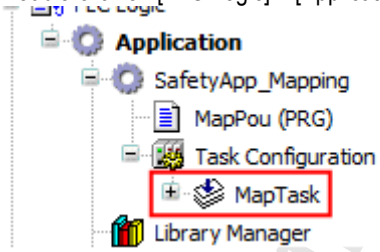
8.5.4 Set interval of the data exchange

8472

The user can set the intervals at which the data can be transferred between fail-safe PLC and standard PLC.

To set the interval of the variable mapping.

► In standard branch of the device tree:
Double-click on [PLC Logic] > [Application] > [SafetyApp_Mapping] > [Task configuration] > [MapTask]



> The editor window shows the configuration options for variable mapping.

► Select the value [Cyclic] in the [Type] list.

► Enter the requested interval in the [Interval] field.

► Save the project to apply changes.

8.6 Safe cross communication

Contents

Set up fail-safe network variables 88
Use fail-safe network variables in the application 90

11525

► Familiarise yourself with the following CODESYS functions!
- Fail-safe network variables
  → Online help > Add-Ons > CODESYS Safety > CODESYS Safety NetVars

The SmartSPS SafeLine AC4S device family supports the Safety NetVars CODESYS function (fail-safe network variables). Safety NetVars ensure transmission and reception of safe data between 2 fail-safe PLCs in a CODESYS project.

Safety NetVars work according to the transmitter-receiver principle. It is necessary to create a transmitter-receiver pair for each requested transmission direction. A transmitter can have several receivers.

8.6.1 Set up fail-safe network variables

21673

► Observe the notes on the configuration of the Ethernet configuration interfaces 1 (X3) and 2 (X8). → **Change IP settings** (→ p. 40)

Max. 10 connections are supported (transmitter >>> receiver). Max. 32 bytes may be transmitted per transmitter/receiver pair.

► Use the following formulas to determine the size of the transfer data per transmitter/receiver pair.

- Safety NetVars (transmitter):
  Transfer data = $sum_{SafetyNetVars(transmitter)}$*2+ 10 bytes
  $Sum_{SafetyNetVars(transmitter)}$: Add size of all declared variables (up to 16 SAFEBOOL = 2 bytes, each SAFEINT = 2 bytes, each SAFEWORD = 2 bytes)
- Safety NetVars (receiver):
  Transfer data = $sum_{SafetyNetVars(receiver)}$*2 + 10 bytes
  $Sum_{SafetyNetVars(receiver)}$: Add size of all declared variables (up to 16 SAFEBOOL = 2 bytes, each SAFEINT = 2 bytes, each SAFEWORD = 2 bytes)

In each cycle of the fail-safe PLC max. 256 bytes can be transferred in each direction between standard PLC and safety PLC.

► Observe the max. data volume that can be transferred for dimensioning the Safety NetVars.

► Include the following objects in the calculation of the transfer data:

- logical exchange objects: → **Data exchange between standard PLC and fail-safe PLC** (→ p. 83)
- safe fieldbus communication (FSoE): → **Safe fieldbus communication** (→ p. 91)

To transmit safe data from SmartPLC SafeLine AC4S 1 to SmartPLC SafeLine AC4S 2:

Requirements

> SmartPLC SafeLine AC4S 1 and SmartPLC SafeLine AC4S 2 are participants of the same CODESYS project (→ **Add additional device to the project** (→ p. 41)).

> The SmartPLC SafeLine AC4S devices 1 and 2 are coupled via the Ethernet network.

1 Create network variable list in the transmitter

► In the safety extension of the device tree of SmartPLC SafeLine AC4S 1: Select [Safety Logic] > [SafetyApp].

► Select [Project] > [Add Object] > [Safety network variable list (Sender)].

> Dialogue window appears.

► Enter a unique name for the list in the field [Name].

► Click on [Add].

> The editor window shows the configuration of the variable list of the sender.

2 Declare safe variables

► Right-click on the Editor window.

► Select [Insert Variable Declaration] in the context menu.

> Dialogue window appears

► Set the parameters of the safe variable as requested.

► Click on [OK] to add the safe variable to the network variable list.

> The editor window shows the declared variable.

► Optional: Repeat step 2 to add additional safe variables to the network variable list.

3 Create network variable list in the receiver

► In the safety extension of the device tree of SmartPLC SafeLine AC4S 2: Select [Safety Logic] > [SafetyApp].
► Select [Project] > [Add Object] > [Safety network variable list (Receiver)].
> Dialogue window appears.
► Enter a unique name for the list in the field [Name].
► Click on [Add].
> The editor window shows the configuration of the variable list of the receiver.

4 Couple network variable lists

► Double click on created network variable list (receiver)
► The editor window shows the configuration of the variable list.
► Select the requested network variable list (sender) in the [Associated Sender] list.
► Click on [Refresh].
► CODESYS couples the network variable lists of the transmitter and the receiver.
> The editor window shows the safe variables of the coupled network variable list.

5 Optional: Set up other fail-safe cross communication

► Repeat steps 1 to 3 for the opposite transmission direction.

8.6.2 Use fail-safe network variables in the application

10973

⚠ WARNING

Risk of personal injuries and damage to property

Stopping a fail-safe PLC in the Debug mode does neither lead to an interruption of the safe cross communication nor to a reset of the currently active signals, i.e. receiving fail-safe PLCs do not react. This behaviour is intended and is used for testing of the system and for trouble shooting.

► Make sure that there are never any people in the dangerous zone of the installation in the Debug mode.

To use the set Safety NetVars they have to be activated in the safe application of the SmartPLC SafeLine AC4S devices.

Requirements:

> Network variable lists were created in transmitter and receiver and correctly configured (→ **Set up fail-safe network variables** (→ p. 88)).

1 Activate network variable list of the transmitter

► Open safe application of the SmartPLC SafeLine AC4S (transmitter)

► Add NetVarSender function block

► Select the designation of the network variable list (transmitter) as instance name.

► Configure function block (→ CODESYS help)

2 Activate network variable list of the receiver

► Open safe application of the SmartPLC SafeLine AC4S (receiver).

► Add NetVarReceiver function block.

► Select the designation of the network variable list (receiver) as instance name.

► Configure function block (→ CODESYS help)

3 Optional: Activate more network variable lists

► Repeat steps 1 and 2 to activate additional network variable lists.

8.7 Safe fieldbus communication

Contents

Configure FSoE connection....91
Notes on the FSoEMaster....92

19865

► Familiarise yourself with the following CODESYS functions!
  ▪ Fail Safe over EtherCAT
    → Online help > Add-Ons > CODESYS Safety > field buses > FSoE
► Observe the notes on the configuration of the Ethernet configuration interfaces 1 (X3) and 2 (X8). → **Change IP settings** (→ p. 40)

The device supports the following protocols for safe communication via fieldbuses:

- Fail Safe over EtherCAT (FSoE)

8.7.1 Configure FSoE connection

21674

► Use an instance of the function block FSoEMaster for each fail-safe FSoE input/output module!
► Observe the notes on the function block FSoEMaster! (→ **Notes on the FSoEMaster** (→ p. 92))

The FSoE addresses and the connection IDs in an FSoE network have to be unambiguous. CODESYS does not see if an FSoE address or a connection ID is allotted several times.

To access the safe input/output data of the EtherCAT slave via an FSoE connection:

Requirements

> SmartPLC SafeLine AC4S is connected to the EtherCAT slave via Ethernet network.
> SmartPLC SafeLine AC4S is configured as EtherCAT master (→ **Add EtherCAT master** (→ p. 43)).

1 Add safe EtherCAT slave

► Add safe EtherCAT slave (→ **Add and configure EtherCAT slave** (→ p. 45))
> Safe module of the EtherCAT slave appears in the safe area of the device tree under [Logical I/Os].

2 Configure safe EtherCAT slave

► In the safe area of the device tree under [Logical I/Os]: Double click on safe module of the EtherCAT slave
► Select tab [Safe configuration].
> Device editor shows configuration of the safe module of the EtherCAT slave.
► Set the following parameters as required:

Parameter	Description	Possible values
[FSoE address]	FSoE address of the safe EtherCAT slave	1...255 (depending on the FSoE slave)
[Connection ID]	Number of the connection to the safe EtherCAT slave	1...65535
[Watchdog Time]	Watchdog time	

3 Map safe input/output signals of the EtherCAT slaves to variables

► In the safe area of the device tree under [Logical /I/Os]: Double click on safe module of the EtherCAT slave

► Select [I/O mapping] tab.

> Device editor shows variable mapping of the safe input/output signals.

► Map safe input/output signals of the EtherCAT slave to variables.

4 Access safe data of the EtherCAT slaves

► Open safe application of the SmartPLC SafeLine AC4S.

► Add function block FSoEMaster.

► Configure function block FSoEMaster as requested.

8.7.2 Notes on the FSoEMaster

18586

► Familiarise yourself with the following CODESYS functions!

- Function block FSoEMaster
  → Online help > Add-Ons > CODESYS Safety > field buses > FSoE > FSoEMaster

The documentation of the function block FSoEMaster in the CODESYS online help is not correct.

► Observe the following when you use the function block FSoEMaster:

- The parameter S_ActivateIN always has to be set to TRUE regardless of whether it is an EtherCAT input or output module.
- The diagnostic code 0x80nn indicates the current state of the FSoEMaster. The bits nn can have the following values:

Value	State	Description	
0x64	Reset	The connections are reset (outputs in the safe state)	
0x65	Session	The session ID is transferred (outputs are in the safe state)	
0x66	Connection	The connection ID is transferred (outputs are in the safe state)	
0x67	Parameter	The parameters are transferred (outputs are in the safe state)	
0x68	Data	Process data or replacement values are being transmitted. The output S_ReadyIN indicastes the data that are transferred:	
		FALSE	Transfer of a replacement value owing to an error recognised in the IO module.
		TRUE	Process data transmission

8.8 Set process safety time

Contents

Process safety time for the SmartPLC SafeLine AC4S 93
Set variable components of the response time 99

16681

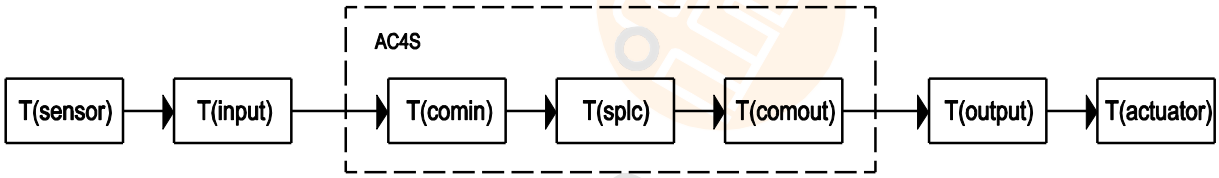
8.8.1 Process safety time for the SmartPLC SafeLine AC4S

17130

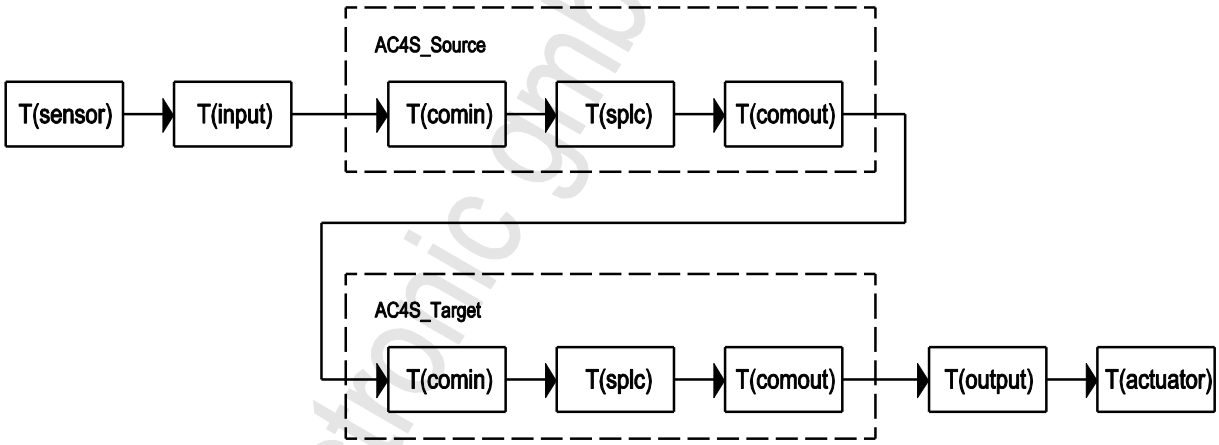
Basic information: → "Process safety time" in the original device manual

For calculating the process safety time of the entire system the following scenarios have to be distinguished:

- Without safe cross communication



- With safe cross communication



The following table shows the meaning of the individual components of the SmartPLC SafeLine AC4S:

Component	Description	Possible values	
T(sensor)	Response time of the safety sensor	→ Operating instructions / data sheet	
T(input)	Response time of the safety input device	→ Operating instructions / data sheet ▪ with local input: 0 ms	
T(comin)	Response time of the input communication channel (as from terminal)	AS-i	30ms
		Local IO	16ms
		FSoE	32ms + 4*T(MapTask) + 4*T(MainTask)
		Safety NetVars	32ms + 4*T(MapTask) + 4*T(MainTask)
T(splc)	Processing time of the fail-safe PLC	2*T(SafetyTask)	
T(comout)	Response time of the output communication channel (up to terminal)	AS-i	15 ms
		Local IO	8 ms
		FSoE	32ms + 4*T(MapTask) + 4*T(MainTask)
		Safety NetVars	32ms + 4*T(MapTask) + 4*T(MainTask)
T(output)	Response time of the output variable unit	→ Operating instructions / data sheet ▪ with local output: 0 ms	
T(actuator)	Reaction time of the actuator	→ Operating instructions / data sheet	

The process safety time is extended by 1 PLC cycle if the safe AS-i input slave or the safe local input channels are configured in CODESYS by means of the following logical devices:

- **SF_IN_ASi_dependent_filter_nshutdown** (→ p. 230)
- **SF_IN_local_dependent_filter_nshutdown** (→ p. 267)

Step 1: Calculate typical response times

10931

► Calculate typical response times of the safety functions of the device:

- Without safe cross communication

	AS-i output	Local output	FSoE slave output
AS-i input	30 ms + 2*T(SafetyTask) + 15 ms	30 ms + 2*T(SafetyTask) + 8 ms	30 ms + 2*T(SafetyTask) + 4*T(MainTask) + 4*T(MapTask) + 32 ms
Local input	16 ms + 2*T(SafetyTask) + 15 ms	16 ms + 2*T(SafetyTask) + 8 ms	16 ms + 2*T(SafetyTask) + 4*T(MainTask) + 4*T(MapTask) + 32 ms
FSoE Slave Eingang	32 ms + 4*T(MainTask) + 4*T(MapTask) + 2*T(SafetyTask) + 15 ms	32 ms + 4*T(MainTask) + 4*T(MapTask) + 2*T(SafetyTask) + 8 ms	32 ms + 4*T(MainTask) + 4*T(MapTask) + 2*T(SafetyTask) + 4*T(MainTask) + 4*T(MapTask) + 32 ms

- With safe cross communication (Safety NetVars)

	To device 2		
From device 1	**AS-i output**	**Local output**	**FSoE slave output**
AS-i input	30 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 15 ms	30 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 8 ms	30 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms
Lokaler Eingang	16 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 15 ms	16 ms 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 8 ms	16 ms 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms
FSoE Slave Eingang	4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 15 ms	4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 8 ms	4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 2*T(SafetyTask_Dev1) + 4*T(MapTask_Dev1) + 4*T(MainTask_Dev1) + 32 ms + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms + 2*T(SafetyTask_Dev2) + 4*T(MapTask_Dev2) + 4*T(MainTask_Dev2) + 32 ms

The formula in the table determine only the typical response time of the device. For the calculation of the typical process response time the delay times of the safety AS-i sensors/actuators and the respective input/output devices have to be added additionally (T(sensor) + T(input) and T(output) + T(aktuator)).

Step 2: Calculate the response times of the safety function

11231

► Make measurements for a real system.
► Ensure sufficient dimensioning of the watchdog times and task monitoring.

► Measure the maximum response time in all sections of the signal propagation path with safe fieldbus communication or safe cross communication.
  - safe fieldbus communication (FSoE): output tMaxRespTimeMS of the function block FSoEMaster
  - safe cross communication (Safety NetVars): output tMaxRespTime MS of the function block NetVarReceiver

► Calculate the response time of the safety function with the measured values:
  - without safe cross communication:

	AS-i output	**Local output**	**FSoE slave output**
AS-i input	-	-	30 ms + 2*T(SafetyTask) + tMaxRespTimeMS(FSoE_out)
Local input	-	-	16 ms + 2*T(SafetyTask) + tMaxRespTimeMS(FSoE_out)
FSoE slave input	tMaxRespTimeMS(FSoE_in) + 2*T(SafetyTask) + 15 ms	tMaxRespTimeMS(FSoE_in) + 2*T(SafetyTask) + 8 ms	tMaxRespTimeMS(FSoE_in) + 2*T(SafetyTask) + tMaxRespTimeMS(FSoE_out)

  - with safe cross communication:

	To device 2		
From device 1	**AS-i output**	**Local output**	**FSoE slave output**
AS-i input	30 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + 15 ms	30 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + 8 ms	30 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out)
Local input	16 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + 15 ms	16 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + 8 ms	16 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out)
FSoE Slave Eingang	tMaxRespTimeMS(FSoE_in) + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + 15 ms	tMaxRespTimeMS(FSoE_in) + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + 8 ms	tMaxRespTimeMS(FSoE_in) + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafeNetVars) + 2*T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out)

The formula in the table determine only the maximum response time of the devices. For the calculation of the maximum process response time the delay times of the safety AS-i sensors/actuators and the respective input/output devices have to be added additionally (T(sensor) + T(input) and T(output) + T(aktuator)).

Step 3: Define watchdog times

10918

Safe cross communication and safe fieldbus communication are monitored by means of watchdogs. Sufficiently long watchdog times have to be selected.

► Determine the watchdog times on the basis of the maximum response times of the safety function measured ins step 2. ifm recommends the following formulas:
  - safe cross communication: T(WD_SafetyNetVars) = 2*T(tMaxRespTimeMS)
  - safe fieldbus communication: T(WD_FSoE) = 2*T(tMaxRespTimeMS)

► Set watchdog times (**Set variable components of the response time** (→ p. 99)).

Step 4: Calculate the worst-case response time of the safety function

10922

The worst-case process safety time

► Calculate the worst case on the basis of the calculated and measured values.

- Without safe cross communication:

	AS-i output	**Local output**	**FSoE slave output**
AS-i input	-	-	T(WD_FSoE)
Local input	-	-	T(WD_FSoE)
FSoE slave input	T(WD_FSoE) + 2*T(SafetyTask) + 15 ms	T(WD_FSoE) + 2*T(SafetyTask) + 8 ms	T(WD_FSoE_in) + 2*T(SafetyTask) + 2*tMaxRespTimeMS(FSoE_out)

- With safe cross communication:

	To device 2		
From device 1	**AS-i output**	**Local output**	**FSoE slave output**
AS-i input	MAX(30 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2*T(SafetyTask_Dev2) + 15 ms; T(WD_SafetyNetVars) + 2*T(SafetyTask_Dev2) + 15 ms)	MAX(30 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2*T(SafetyTask_Dev2) + 8 ms; T(WD_SafetyNetVars) + 2*T(SafetyTask_Dev2) + 8 ms)	MAX(30 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2*T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out); T(WD_SafetyNetVars) + 2*T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out); T(WD_FSoE_out))
Local input	MAX(16 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2*T(SafetyTask_Dev2) + 15 ms; T(WD_SafetyNetVars) + 2*T(SafetyTask_Dev2) + 15 ms)	MAX(16 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2*T(SafetyTask_Dev2) + 8 ms; T(WD_SafetyNetVars) + 2*T(SafetyTask_Dev2) + 8 ms)	MAX(16 ms + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2*T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out); T(WD_SafetyNetVars) + 2*T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out); T(WD_FSoE_out))
FSoE slave input	MAX(T(WD_FSoE_in) + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2*T(SafetyTask_Dev2) + 15 ms; T(WD_SafetyNetVars) + 2*T(SafetyTask_Dev2) + 15 ms)	MAX(T(WD_FSoE_in) + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2*T(SafetyTask_Dev2) + 8 ms; T(WD_SafetyNetVars) + 2*T(SafetyTask_Dev2) + 8 ms)	MAX(T(WD_FSoE_in) + 2*T(SafetyTask_Dev1) + tMaxRespTimeMS(SafetyNetVars) + 2*T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out); T(WD_SafetyNetVars) + 2*T(SafetyTask_Dev2) + tMaxRespTimeMS(FSoE_out); T(WD_FSoE_out))

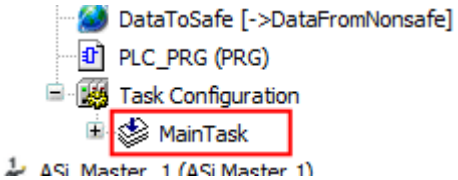
The response times of the output side have to be added to the calculated worst-case times (T(output) + T(actuator)).

8.8.2 Set variable components of the response time

8898

The variable components influencing the process safety time of the entire system can be set by the user at the following nodes in the CODESYS project:

- T(MainTask):

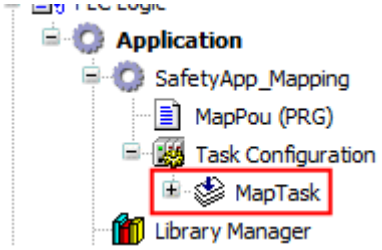


  - Location: standard branch of the device tree
  - Node: [MainTask]
  - Parameters: [Interval]

- *T(SafetyTask):

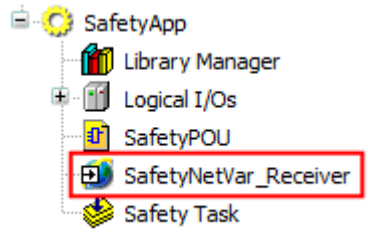
  - Location: safety extension of the device tree
  - Node: [Safety Task]
  - Parameters: [Cycle time]

- T(MapTask):



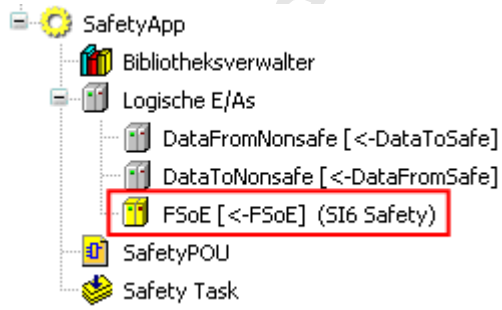
  - Location: standard branch of the device tree
  - Node: [MapTask]
  - Parameters: [Interval]

- T(WD_SafetyNetVars):



  - Location: safety extension of the AS-i device tree
  - Node: safety network variable (receiver)
  - Parameters: [Watchdog Time]

- T(WD_FSoE):



  - Location: safety extension of the AS-i device tree
  - Node: [Logical I/Os] > safe EtherCAT module
  - Parameters: [Watchdog Time]

8.9 Configure safety task

2631

The safety tasks of the safe application are configured via the "Safety Task Editor" of the programming system CODESY safety.

► Familiarise yourself with the following CODESYS functions!

- Safety Task Editor
  → Online help > Add-Ons > CODESYS Safety > Editors >Safety Task Editor

8.9.1 Set cycle time of the fail-safe PLC

18442

The programmer can freely set the cycle time of the fail-safe PLC. Whole values from 10...100 ms are valid.

To set the cycle time of the fail-safe PLC:

► In the safety extension area of the device tree:
Double-click on [Safety Logic] > [SafetyApp] > [Safety Task]

> Editor window shows the configuration options of the selected safety task.

► Enter the required cycle time in the field [Cycle time].

► Select the safety POUs from the list to which the set cycle time should apply.

> The set cycle time applies to all selected safety POUs.

9 Operation

Contents

Transfer CODESYS project to device 101
Operating states SmartPLC SafeLine AC4S 106
Reset 111
Display web visualisation 114

18492

9.1 Transfer CODESYS project to device

20257

To save the CODESYS project on SmartPLC SafeLine AC4S the following components must be transferred to the device separately:

- standard application "Application" (→ **Download standard application to SmartPLC SafeLine AC4S** (→ p. 102))
- safe application "SafetyApp" (→ **Load safe application to SmartPLC SafeLine AC4S** (→ p. 103))

► Note the remarks about the operating modes of standard PLC and the fail-safe PLC of SmartPLC SafeLine AC4S.
→ **Operating states SmartPLC SafeLine AC4S** (→ p. 106)

ifm system solutions and CODESYS applications created by the user must not be saved and executed on the SmartPLC SafeLine AC4S at the same time.

► Before loading a standard application to SmartPLC SafeLine AC4S, delete all ifm system solutions saved on the device (→ Device manual, Uninstall ifm apps)!

To be able to use an ifm system solution in a user project the functions must be inserted in the project via libraries to be ordered separately.

► Contact the AS-i specialist of ifm electronic!

9.1.1 Download standard application to SmartPLC SafeLine AC4S

15020

If standard network variables and safety network variables are used simultaneously: → **Note: Projects with network variables** (→ p. 104)

To transfer standard application of the CODESYS project to SmartPLC SafeLine AC4S:

Requirements:

> Communication path of standard PLC is correctly set (→ **Configure programming interface** (→ p. 21)).

> All ifm system solutions on the device were deleted (→ Device manual: **ifm-Apps deinstallieren**).

1 Download standard application to the device

► In standard branch of the device tree:
Mark standard application and activate with [Project] > [Set active application].

► Select [Online] > [Login].

> CODESYS changes to the online mode.

> CODESYS compiles standard application and transfers it to SmartPLC SafeLine AC4S.

2 Start standard application

► Start [Application].

> [Application] is in the RUN mode.

3 Optional: Create boot application

► Select [Online] > [Create boot application].

> CODESYS stores standard application non-volatilely on SmartPLC SafeLine AC4S.

If the parameter "Create implicit boot application on download" is activated in the features of the application, step 3 is not necessary any more.

To check if the parameter is active:

► Mark [Application].

► Select [View] > [Properties...].

> Tab [Boot application] shows the current parameter value.

9.1.2 Load safe application to SmartPLC SafeLine AC4S

20385

In the event of simultaneous use of network variables and safety network variables: → **Note: Projects with network variables** (→ p. 104)

On first log-in to the fail-safe controller the programmer is asked to enter an instance identification. The instance identification is the serial number of the device without the leading zeros.

Example:

- Serial number: `000000569158`
- Instance identification: `569158`

To determine the serial number of the device:

→ Label underneath the Fieldbus interface (X6/X7) or
→ GUI/web interface:

►  > 

To transfer the safe application of the CODESYS project to SmartPLC SafeLine AC4S:

Requirements:

> Communication path of the fail-safe PLC is correctly set (→ **Configure programming interface** (→ p. 21)).

1 Load safe application to SmartPLC SafeLine AC4S

► In the safety extension area of the device tree:
Mark [SafetyApp] and activate with [Projekt] > [Aktive Applikation setzen].

► Select [Online] > [Login].

> CODESYS changes to the online mode.

> CODESYS compiles the safe application and transfers it to the SmartPLC SafeLine AC4S.

► Observe the warnings and notes.

2 Start safe application

► Start [SafetyApp].

> [SafetyApp] is in the RUN state.

3 Optional: Teach code sequences

► Teach code sequences of the safe AS-i input slaves (→ **Process safety time for the SmartPLC SafeLine AC4S** (→ p. 93)).

4 Optional: Create safe boot application

► Pin safe application (→ CODESYS online help).

► Select [Online] > [Create boot application].

> CODESYS stores the safe application non-volatilely on SmartPLC SafeLine AC4S.

When the safe boot application has been created on the SmartPLC SafeLine AC4S, the fail-safe PLC remains in the "Debug operation" operating status.

► Note remarks about changing the operating states (→ **Change between the states** (→ p. 110))

9.1.3 Note: Projects with network variables

15210

If network variables and safety network variables are used simultaneously, the user has to observe the following sequence for the transmission of the applications to the device:

1 Create safety network variables

- ► Device 1: Create and configure safety network variable list (transmitter)
- ► Device 2: Create and configure safety network variable list (receiver)
- ► Optional: Create and configure more transmitter/receiver pairs

2 Transfer applications to devices

- ► On all devices: Log in to standard PLC and the fail-safe PLC.
- ► Transfer boot applications to both PLCs
- ► Check if safety NetVars function correctly.

3 Log out

- ► Log out from the standard PLC.

4 Create standard network variables

- ► Device 1: Create and configure network variable list (transmitter).
- ► Device 2: Create and configure network variable list (receiver).
- ► Optional: Create and configure more transmitter/receiver pairs.

5 Transfer application to devices

- ► On all devices: Log in to standard PLC.
- ► Transfer boot application to standard PLC.
- ► Check if network variables and safety network variables function correctly.

9.1.4 Teach code sequences of the safe AS-i slaves

18455

To teach the code sequence of a safe AS-i slave:

Requirements:

- > Safe AS-i slave is connected to SmartPLC SafeLine AC4S and addressed.
- > Parameter "EnableStartupTest" for all safe AS-i input slaves is activated.
- > All safe AS-i input slaves are enabled (e.g. E-stop: unlock).
- > Safe application is stored on the device and has been started (→ **Load safe application to SmartPLC SafeLine AC4S** (→ p. 103)).
- > Code table is blank.

1 Teach code sequences of a safe AS-i slave

- > SmartPLC SafeLine AC4S requires testing for safe AS-i input slave (OSC message).
- ► Confirm safe AS-i input slave and unlock again (switching sequence: lock ⇨ unlock)
- > Safe AS-i slave transmits code sequence to SmartPLC SafeLine AC4S.
- > SmartPLC SafeLine AC4S fills code table with code sequence of the safe AS-i input slave.
- > Code sequence of the safe AS-i input slave is taught.

2 Optional: Teach more code sequences

- ► Repeat step 1 for additional safe AS-i input slaves.

9.1.5 Delete boot application via SD card

15970

If after the start of a complex boot application the device is overloaded and does not respond any more to user inputs or login attempts, the boot application on the device needs to be forcibly deleted.

To delete the boot application on the device:

► Disable the write protection of the SD card.
► Create a file named `KillBootApp.txt` in the root directory of the SD card.
► Insert the SD card into the SD card slot of the device.
► Restart the device.
> Boot application on the device-internal standard PLC is deleted.
> The file `KillBootApp.txt` on the SD card is renamed in `KillBootApp.rdy`.

With this method the following data on the device-internal standard PLC is removed:

- all files of the boot application
- all CRC files
- directory with web and/or target visualisations
- data in the memory area F-RAM

9.2 Operating states SmartPLC SafeLine AC4S

20258

This chapter provides remarks about the operating modes and the states of the standard PLC, the fail-safe PLC of the device and the states of the applications.

9.2.1 standard PLC

7973

Operating mode of the standard PLC

13769

The standard PLC of the device can be operated in the following modes:

- **Offline mode**
  In the offline mode the user is logged out of the standard PLC or there is no connection between CODESYS and the standard PLC (e.g. connection loss).
- **Online mode**
  In the online mode the user is logged in to the standard PLC.

States of the standard PLC application

4136

The standard applications saved on the device are executed independently in separate tasks. A standard application can have the following states:

- **Unload**
  No application is saved on the standard PLC.
- **RUN**
  The standard application is executed (cyclically processed).
- **STOP**
  The standard application is not executed.

Display operating state of the standard application

10272

To display the current operating state of the standard PLC choose one of the following options:

- **CODESYS:**
  > In the device tree: Node of the standard application indicates the current state.
  OR:
  > In online mode the CODESYS status bar shows the current state of the standard application.
- **GUI / web interface of the device:**
  ►  >  > 
  ► Select the [Applications] tab.
  > The page displays the operating states of the standard PLC applications saved on the device.

Switch operating states

10264

To switch between the operating states of the standard application choose one of the following options:

Start standard PLC application

18027

To start a standard PLC application stored on the device:

- **CODESYS:**
  - ► In the standard branch of the device tree: highlight standard application as active application.
  - ► Use [Online] > [Login] to establish the connection with the CODESYS standard PLC.
  - ► Use [Debug] > [Start] to start the processing of the active application.
  - > Application goes to RUN state.
  - ► Optional: repeat process for additional applications.
- **GUI / web interface:**
  - ►  >  > 
  - ► Select the [Applications] tab.
  - ► Use [▲] / [▼] to select the required application.
  - > Page shows the operating status of the selected application.
  - ► Press [Start] function key to start the processing of the selected application.
  - > Application goes to RUN state.
  - ► Optional: repeat process for additional applications.

Stop standard PLC application

18029

To stop a standard PLC application stored on the device:

- **CODESYS:**
  - ► In the standard branch of the device tree: highlight standard application as active application.
  - ► Use [Online] > [Login] to establish the connection with the CODESYS standard PLC.
  - ► Use [Debug] > [Stop] to stop the processing of the active application.
  - > Application goes to STOP state.
  - ► Optional: repeat process for additional applications.
- **GUI / web interface:**
  - ►  >  > 
  - ► Select [Applications] tab.
  - ► Use [▲] / [▼] to select the required application.
  - > Page shows the operating status of the selected application.
  - ► Press [Stop] function key to stop the processing of the selected application.
  - > Application goes to STOP state.
  - ► Optional: repeat process for additional applications.

9.2.2 Fail-safe PLC

8665

Familiarise yourself with the following CODESYS functions!

- Introduction Safety Online
  → Online help > Add-Ons > CODESYS Safety > Online mode > Introduction: Safety online
- States of the safety controller
  → Online help > Add-Ons > CODESYS Safety > Online mode > States of the safety controller
- DEBUG mode
  → Online help > Add-Ons > CODESYS Safety > Online mode > Debug mode

The following features are of importance for the correct functioning of the fail-safe PLC of SmartPLC SafeLine AC4S:

- **Operation mode of the fail-safe PLC** (→ p. 108)
- **States of the fail-safe PLC** (→ p. 109)
- **States of the safe application** (→ p. 109)

Operation mode of the fail-safe PLC

3083

The fail-safe PLC of SmartPLC SafeLine AC4S can be operated in the following modes:

- **Offline mode**
  In the offline mode the user is logged out of the fail-safe PLC or there is no connection between CODESYS and the fail-safe PLC (e.g. connection loss).
- **Online mode**
  In the online mode the user is logged in to the fail-safe PLC.

States of the fail-safe PLC

7365

The fail-safe PLC of SmartPLC SafeLine AC4S can take the following states:

- **Unload**
  In the state "unload" no safe application is stored on SmartPLC SafeLine AC4S.
- **Unsafe operation (debug mode)**
  In the state "unsafe operation" a download application or a boot application is stored on SmartPLC SafeLine AC4S which is in the STOP state (→ **States of the safe application** (→ p. 109)).
  The debug mode has the following sub-states:
  - Unsafe (DL)
    In the sub-state "unsafe" a safe application is stored on SmartPLC SafeLine AC4S.
  - Unsafe (BA)
    In the sub-state "unsafe (BA)" a safe boot application is stored on SmartPLC SafeLine AC4S.
- **Safe operation**
  In the state "safe operation" a safe boot application is stored on SmartPLC SafeLine AC4S which is in the RUN state (→ **States of the safe application** (→ p. 109)).
  The state "safe operation" has the following sub-states:
  - SAFE + logged in
    In the sub-state "SAFE + logged in" the fail-safe PLC is in the online mode (→ **Operation mode of the fail-safe PLC** (→ p. 108)).
  - Logged out
    In the sub-state "logged out" the fail-safe PLC is in the offline mode (→ **Operation mode of the fail-safe PLC** (→ p. 108)).

In the online mode the current operating status of the safe PLC is displayed in the CODESYS status bar.

States of the safe application

18383

The safe application on the fail-safe PLC of SmartPLC SafeLine AC4S can take the following states:

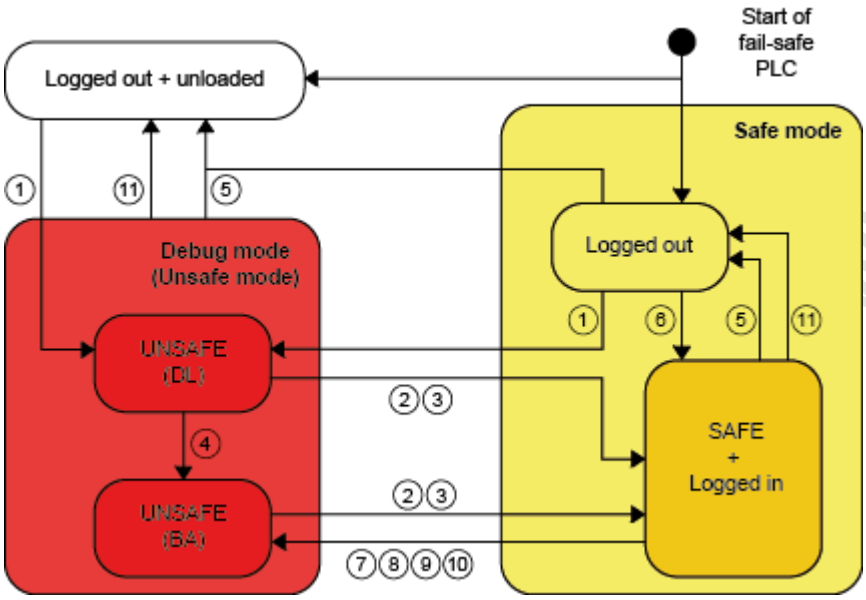
- **RUN**
  The safe application is executed (cyclic processing).
- **STOP**
  The safe application is not executed.
- **TERMINATED**
  The safe application was terminated due to a runtime error.

In the online mode the current state of the safe application is displayed in the CODESYS status bar.

Change between the states

7300

Following status diagram shows the relations and possible transitions between the individual operating modes and the states of the fail-safe PLC:



1. Log-in with download ([Online] > [Login])
2. Start boot application ([Debug] > [Start])
3. Start log-out again with boot application
4. Create boot application ([Online] > [Create boot application])
5. Log-out ([Online] > [Logout])
6. Log-in to boot application ([Online] > [Login])
7. Stop ([Debug] > [Stop])
8. Reset cold ([Online] > [Reset])
9. Write ([Debug] > [Write values])
10. Force ([Debug] > [Force values])
11. Connection loss between CODESYS and SmartPLC SafeLine AC4S (automatic transition)

9.3 Reset

Contents

Standard PLC 111
Fail-safe PLC 113

18025

9.3.1 Standard PLC

415

A set of standard PLC simultaneously starts a reset of the fail-safe PLC of SmartPLC SafeLine AC4S.

► Note remarks about the start-up behaviour of the controller (→ **Start-up behaviour of the controller**).

Supported reset variants

18613

The following table shows the reset variants supported by the device-internal CODESYS standard PLC and the resulting system behaviour:

Type of reset	System behaviour	Triggering actions
Reset (warm)	▪ standard application goes to STOP state. ▪ Standard variables (VAR) of the standard application are initialised. ▪ Remanent variables (VAR RETAIN) of the standard application keep their current values.	→ **Reset the standard application (warm)** (→ p. 112)
Reset (cold)	▪ standard application changes to the STOP state. ▪ All variables (VAR, VAR RETAIN) of the standard application are initialised.	→ **Reset the standard application (cold)** (→ p. 112)
Reset (default)	▪ standard application goes to STOP state. ▪ The standard application on the standard PLC is deleted. ▪ All variables (VAR, VAR RETAIN) of the standard application are initialised. ▪ standard PLC is reset to the default state.	→ **Reset the standard application (origin)** (→ p. 112)

A variable that has been declared without an initialisation value is initialised with the variable-specific standard value (e.g. `INT = 0`).

Reset the standard application (warm)

9069

To reset the standard application on the standard PLC, choose one of the following options:

- **CODESYS: command [Reset (warm)]**
  - ► In the standard branch of the device tree:
    Highlight the required standard application as active application.
  - ► Select [Online] > [Login] to establish a connection to the CODESYS standard PLC.
  - > CODESYS switches to online mode.
  - ► Select [Online] > [Reset warm] to reset the standard application.
- **GUI: command [Reset]**
  - ►  > 
  - ► Select [All Applications] tab.
  - ► Use [Reset] to reset all standard applications.

Reset the standard application (cold)

15687

To reset the standard application on the standard PLC, choose one of the following options:

- **Download the standard application to the device**
  - ► → **Download standard application to SmartPLC SafeLine AC4S** (→ p. 102)
- **CODESYS: command "Reset (cold)"**
  - ► In the standard branch of the device tree:
    Highlight the required standard application as active application.
  - ► Select [Online] > [Login] to establish a connection to the CODESYS standard PLC.
  - > CODESYS switches to online mode.
  - ► Select [Online] > [Reset cold] to reset the standard application.

Reset the standard application (origin)

18962

To reset the standard application on the standard PLC:

- **CODESYS: command "Reset (origin)"**
  - ► In the standard branch of the device tree:
    Highlight the required standard application as active application.
  - ► Select [Online] > [Login] to establish a connection to the CODESYS-standard PLC.
  - > CODESYS switches to online mode.
  - ► Select [Online] > [Reset origin] to reset the standard application.

9.3.2 Fail-safe PLC

17445

Supported types of reset

12304

The following table shows the types of reset supported by the device-internal fail-safe PLC and the resulting system behaviour:

Type of reset	System behaviour	Triggering actions
Reset (cold)	▪ Safe application changes to the STOP state. ▪ All variables (VAR, VAR RETAIN) of the safe application are re-initialised.	→ **Reset safe application (cold)** (→ p. 113)

Reset safe application (cold)

12303

To reset the safe application on the fail-safe PLC execute one of the following actions:

- **Download new safe application to device**
  ► → **Transfer CODESYS project to device** (→ p. 101)
- **CODESYS: Execute command "Reset (cold)"**
  ► Set [SafetyApp] as active application.
  ► Select [Online] > [Login].
  > Fail-safe PLC changes to the online mode.
  ► Select [Online] > [Reset cold] to reset the safe application on the device.

9.4 Display web visualisation

17063

To display the created web visualisation:

Requirements:

> PC/laptop is connected to the configuration interface (X3) of the device (→ device manual: **Configuration interface: connection concepts**)

CODESYS standard PLC appliation

- ► Download standard PLC application with web visualisation to the device and start it (→ **Download standard application to SmartPLC SafeLine AC4S** (→ p. 102)).
- ► On PC/laptop: Start web browser.
- ► Enter the following in the address line and press [ENTER] to confirm:
  `<IP address-of-the-device>:<8080>/myvisu.htm`

`myvisu` is the user-defined name of the visualisation (→ **Change properties of the web visualisation** (→ p. 67)).

> Web browser shows the web visualisation of the device.

ifm system solution

- ► Install the ifm system solution on the device and start it (→ device manual, **Install single/basic app** or **Install multi app**).
- ► Display informationen about the installed ifm app (→ device manual, **Show information about installed ifm apps**).
- ► Call hyperlink of the ifm app.

> Web browser shows the web visualisation of the ifm system solution.

10 Appendix

Contents

Library ACnnnn_Utils.library....116
Library ACnnnn_SYS_CMD.library....178
Library SF_IO.library....186
Library SF_LogicalInterfaces.library....201
Error codes: Safe AS-i slaves....276
Error codes: Local I/O interface....278

7156

10.1 Library ACnnnn_Utils.library

Contents

Overview: AS-i functions (FB_ASi) 117
Overview: System functions (FB_System) 163
Enumeration types and complex variables 170

17722

10.1.1 Overview: AS-i functions (FB_ASi)

Contents

Get_ASi_PHY_Dat 118
Get_LAS 120
Get_LDS 122
Get_LPS 124
Get_LCE 126
Get_LCEAS 128
Get_LCEMS 130
Get_LDAE 132
Get_LPF 134
Get_CDI 136
Get_PCD 138
Get_InputParameter 140
Get_OutputParameter 142
Set_AddressMode 144
Set_ASi_Config 146
Set_LPS 148
Set_Mode 150
Set_PCD 152
Set_ProjectAll 154
Set_SlaveAddress 155
Set_SlaveExtendedID1 157
Set_SlaveParameter 159
Get_ASi_Data 161

17459

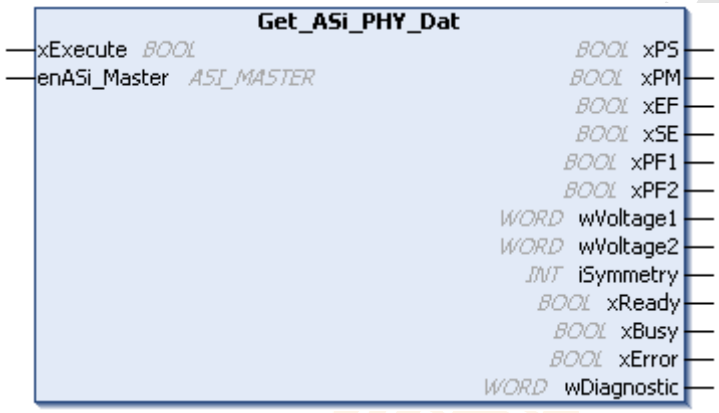
Get_ASi_PHY_Dat

16005

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:



Description

16040

The FB reads the physical data of the selected AS-i master and provides the values.

Input parameters

16041

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16042

Parameter	Data type	Description	Possible values	
xPS	BOOL	Voltage source (Power Source)	FALSE	Unit is supplied via Aux.
			TRUE	Unit is supplied via AS-i.
xPM	BOOL	Power24-Modul (PM)	FALSE	Power24 module missing.
			TRUE	Power24 module is inserted.
xEF	BOOL	Earth fault	FALSE	No earth fault
			TRUE	Supply voltage is asymmetric, earth fault suspected.
xSE	BOOL	Status of the earth fault detection	FALSE	Earth fault detection does not provide valid data (e.g. when AS-i voltage is lacking).
			TRUE	Earth fault detection provides valid data.
xPF1	BOOL	Voltage <22.5 V (power fail 22.5 V)	FALSE	No AS-i power fault (Classic APF)
			TRUE	AS-i power fail (Classic APF), i.e. AS-i voltage is below 22.5 V
xPF2	BOOL	Voltage <19.0V (power fail 19V)	FALSE	No AS-i power fail (24V-APF)
			TRUE	AS-i power fail (24V-APF), i.e. AS-i voltage is below 19.0 V
wVoltage1	WORD	Voltage AS-i+ to AS-i- in mV		
wVoltage2	WORD	Voltage FE to AS-i in mV		
iSymmetry	INT	Symmetry in % (-100% ... +100%)	0xFF9C ... 0x0000 ... 0x0064	-100% ... 0% ... +100%
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set

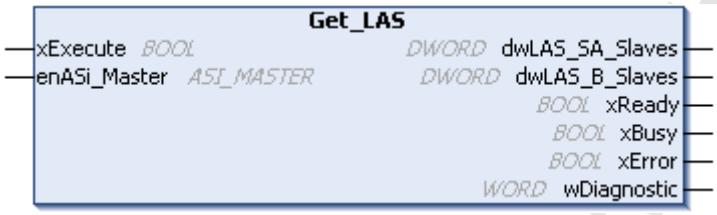
Get_LAS

16008

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:



Description

16068

The FB reads the list of activated slaves (LAS) of the selected AS-i master and provides the values.

Input parameters

16069

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16070

Parameter	Data type	Description	Possible values	
dwLAS_SA_Slaves	DWORD	List of the active S/A slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = address 0 ... Bit 31 (MSB) = address 31/31A	Per bit:	
			0	No single/A slave available
			1	Single/A slave available
dwLAS_B_Slaves	DWORD	List of the active B slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = not used ▪ Bit 1 = address 1B ... Bit 31 (MSB) = address 31B	Per bit:	
			0	No B slave available
			1	B slave available
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

Get_LDS

16013

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

16118

The FB reads the list of detected slaves (LDS) of the selected AS-i master and provides the values.

Input parameters

16119

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16120

Parameter	Data type	Description	Possible values	
dwLDS_SA_Slaves	DWORD	List of detected S/A slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = address 0 ... ▪ Bit 31 (MSB) = address 31/31A	Per bit:	
			0	no slave detected
			1	slave detected
dwLDS_B_Slaves	DWORD	List of detected B slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = not used ▪ Bit 1 = address 1B ... ▪ Bit 31 (MSB) = address 31B	Per bit:	
			0	No slave detected
			1	Slave detected
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

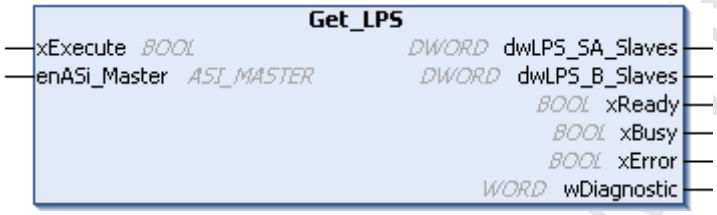
Get_LPS

16015

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:



Description

16130

The FB reads the list of projected slaves (LPS) at the selected AS-i master and provides the values.

Input parameters

16131

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16132

Parameter	Data type	Description	Possible values	
dwLPS_SA_Slaves	DWORD	List of the projected S/A slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = address 0 ... ▪ Bit 31 (MSB) = address 31/31A	Per bit:	
			0	Slave not projected
			1	Slave projected
dwLPS_B_Slaves	DWORD	List of the projected B slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = not used ▪ Bit 1 = address 1B ... ▪ Bit 31 (MSB) = address 31B	Per bit:	
			0	slave not projected
			1	slave projected
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

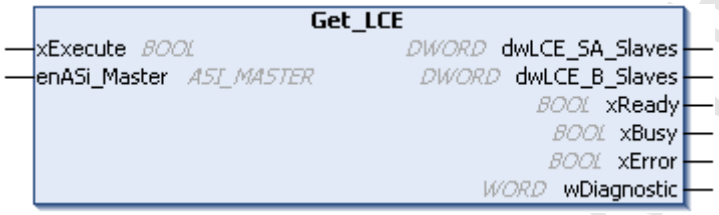
Get_LCE

16009

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:



Description

16075

The FB reads the list of configuration errors (LCE) of the selected AS-i master and provides the values.

Input parameters

16076

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16077

Parameter	Data type	Description	Possible values	
dwLCE_SA_Slaves	DWORD	List of configuration errors of the S/A slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = address 0 ... Bit 31 (MSB) = address 31/31A	Per bit:	
			0	No configuration error
			1	Configuration error
dwLCE_B_Slaves	DWORD	List of configuration errors of the B slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = not used ▪ Bit 1 = address 1B ... Bit 31 (MSB) = address 31B	Per bit:	
			0	No configuration error
			1	Configuration error
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

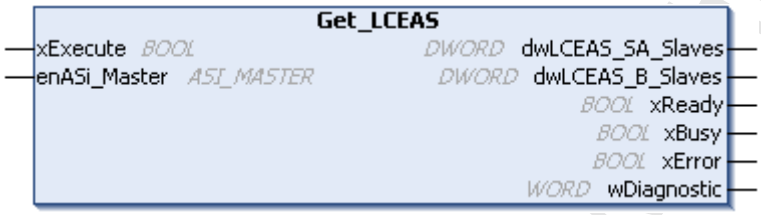
Get_LCEAS

16010

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:



Description

16098

The FB reads the list of existing but not projected slaves (List of Configuration Error – Additional Slave = LCEAS) of the selected AS-i master and provides the values.

Input parameters

16099

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16100

Parameter	Data type	Description	Possible values	
dwLCEAS_SA_Slaves	DWORD	List of configuration errors of the S/A slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = address 0 ... Bit 31 (MSB) = address 31/31A	Per bit:	
			0	No configuration error - additional slave
			1	Slave exists, but not projected
dwLCEAS_B_Slaves	DWORD	List of configuration errors of the B slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = not used ▪ Bit 1 = address 1B ... Bit 31 (MSB) = address 31B	Per bit:	
			0	No configuration error - additional slave
			1	Slave exists, but not projected
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

Get_LCEMS

16011

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

16106

The FB reads the list of projected but missing slaves (List of Configuration Error – Missing Slave = LCEMS) at the selected AS-i master and provides the values.

Input parameters

16107

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16108

Parameter	Data type	Description	Possible values	
dwLCEMS_SA_Slaves	DWORD	List of configured but missing S/A slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = address 0 ... Bit 31 (MSB) = address 31/31A	Per bit:	
			0	No configuration error - missing slave
			1	Slave is projected but not available
sdwLCEMS_B_Slaves	DWORD	List of the configured but non-existing B slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = not used ▪ Bit 1 = address 1B ... Bit 31 (MSB) = address 31B	Per bit:	
			0	No configuration error - missing slave
			1	Slave is projected but not available
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

Get_LDAE

16012

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

16112

The FB reads the double address errors (LDAE) of the selected AS-i master and provides the values in a list.

Input parameters

16113

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16114

Parameter	Data type	Description	Possible values	
dwLDAE_SA_Slaves	DWORD	List of the double address errors. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = address 0 ... Bit 31 (MSB) = address 31/31A	Per bit:	
			0	No double address error
			1	Double address error
dwLDAE_B_Slaves	DWORD	List of double address errors. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = not used ▪ Bit 1 = address 1B ... Bit 31 (MSB) = address 31B	Per bit:	
			0	No double address error
			1	Double address error
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

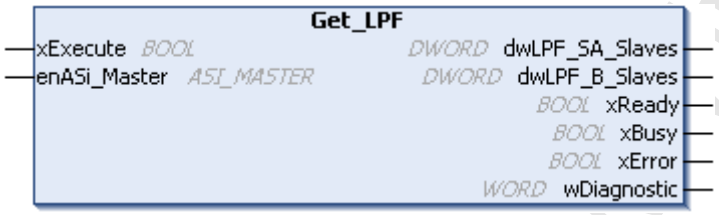
Get_LPF

16014

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:



Description

16124

The FB reads the list of peripheral faults (LPF) of the selected AS-i master and provides the values.

Input parameters

16125

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16126

Parameter	Data type	Description	Possible values	
dwLPF_SA_Slaves	DWORD	List of peripheral faults on S/A slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = address 0 ... Bit 31 (MSB) = address 31/31A	Per bit:	
			0	No peripheral fault
			1	Peripheral fault detected
dwLPF_B_Slaves	DWORD	List of peripheral faults on B slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = not used ▪ Bit 1 = address 1B ... Bit 31 (MSB) = address 31B	Per bit:	
			0	No peripheral fault
			1	Peripheral fault detected
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

Get_CDI

16006

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

16045

The FB reads the configuration data (Configuration Data Image = CDI) of the slaves at the selected AS-i master and provides the values in an array. The configuration data of a slave consists of the registers IO, ID, ID1 and ID2.

Input parameters

16046

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16047

Parameter	Data type	Description	Possible values	
awCDI	ARRAY [0...63] OF WORD	Configuration data of the slaves at the selected AS-i master	Per Word: Bits 0...3: I/O-Code Bits 4...7: ID-Code Bits 8...11: ID1-Code Bits 12...15: ID2-Code	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

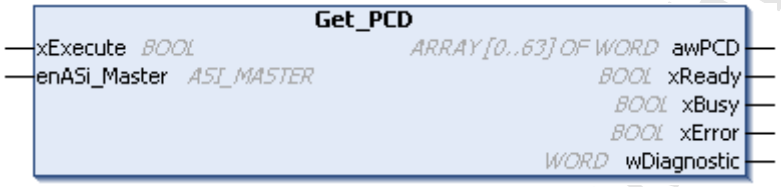
Get_PCD

16017

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:



Description

16141

The FB reads the projected configuration data (Projected Configuration Data Image = PCD) of the slaves on the selected AS-i master and provides the values in an array.

Input parameters

16142

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16143

Parameter	Data type	Description	Possible values	
awPCD	ARRAY [0...63] OF WORD	Permanent configuration files of the slaves on the selected AS-i master	per word: Bits 0...3: I/O-Code Bits 4...7: ID-Code Bits 8-11: ID1-Code Bits 12-15: ID2-Code Data in Word 0 is invalid!	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

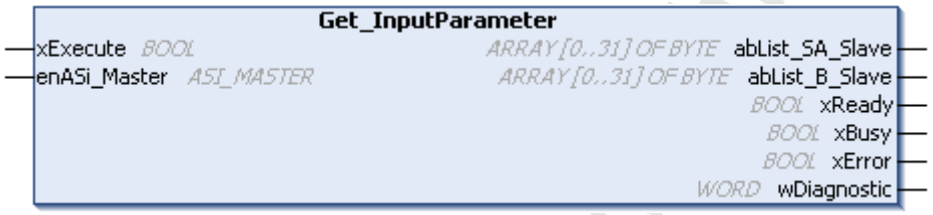
Get_InputParameter

16007

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:



Description

16056

The FB reads the input parameters of the slaves on the selected AS-i master and provides the values in 2 arrays for single A slaves and B slaves.

Input parameters

16057

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16058

Parameter	Data type	Description	Possible values	
abList_SA_Slave	ARRAY[0..31] OF BYTE	List of output parameters of S/A slaves in the selected AS-i master. Each byte contains the output parameters of an AS-i slave. – byte 0 (LSB) = res. – byte 1 = slave with address 1(A) ... – byte 31 = slave with address 31(A)	Per byte:	
			Bits 0...3:	P0-P3
abList_B_Slave	ARRAY[0..31] OF BYTE	List of output parameters of B slaves in the selected AS-i master. Each byte contains the output parameters of an AS-i slave. – byte 0 (LSB) = res. – byte 1 = slave with address 1B ... – byte 31 = slave with address 31B	Per byte:	
			Bits 0...3:	P0-P3
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

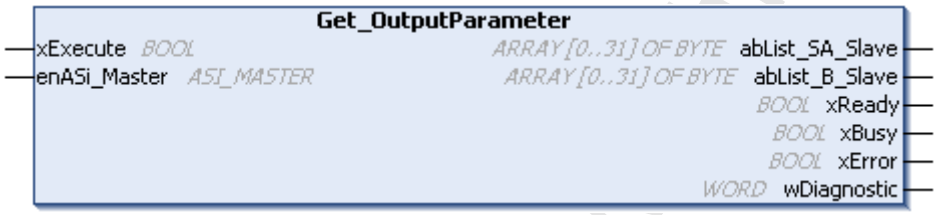
Get_OutputParameter

16016

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:



Description

16136

The FB reads the output parameters of the slaves on the selected AS-i master and provides the values for S/A slaves and B slaves in 2 separate arrays.

Input parameters

16137

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

16138

Parameter	Data type	Description	Possible values	
abList_SA_Slave	ARRAY[0..31] OF BYTE	List of output parameters of S/A slaves in the selected AS-i master. Each byte contains the output parameters of an AS-i slave. – byte 0 (LSB) = res. – byte 1 = slave with address 1(A) ... – byte 31 = slave with address 31(A)	Per byte:	
			Bits 0...3:	P0-P3
abList_B_Slave	ARRAY[0..31] OF BYTE	List of output parameters of B slaves in the selected AS-i master. Each byte contains the output parameters of an AS-i slave. – byte 0 (LSB) = res. – byte 1 = slave with address 1B ... – byte 31 = slave with address 31B	Per byte:	
			Bits 0...3:	P0-P3
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

Set_AddressMode

16018

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

16146

The FB activates/deactivates the parameter "Automatic addressing" for the selected AS-i master.

Input parameters

16147

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2
enAuto_Address_Mode	ASI_ADDRESS_MODE	Parameter activates/deactivates the automatic addressing mode.	Auto_address_disable	Automatic addressing inactive
			Auto_address_enable	Automatic addressing active

Output parameters

17017

Parameter	Data type	Description	Possible values	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set

Set_ASi_Config

16019

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

16149

The FB activates/deactivates the parameters "double address recognition" and "earth fault detection" for the selected AS-i master.

Input parameters

16150

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2
xDoubleAdrDetection	BOOL	Activate/deactivate "Double address recognition"	FALSE	Double address recognition inactive
			TRUE	Double address recognition active
xEarthFaultDetection	BOOL	Activate/deactivate "Earth-fault detection"	FALSE	Earth-fault detection inactive
			TRUE	Earth-fault detection active

Output parameters

17015

Parameter	Data type	Description	Possible values	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0001 Wrong parameter transferred, setting was not adopted.

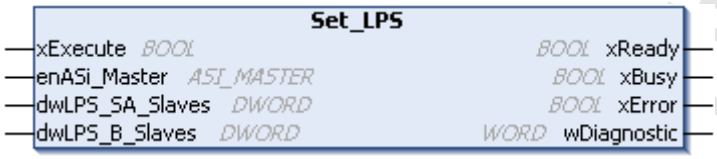
Set_LPS

16020

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:



Description

16153

The FB changes the list of projected slaves (LPS) in the selected AS-i master.

Input parameters

16154

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2
dwLPS_SA_Slaves	DWORD	List of the projected S/A slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = address 0 ... ▪ Bit 31 (MSB) = address 31/31A	Per bit:	
			0	Slave not projected
			1	Slave projected
dwLPS_B_Slaves	DWORD	List of the projected B slaves. Each bit represents an AS-i address: ▪ Bit 0 (LSB) = not used ▪ Bit 1 = address 1B ... ▪ Bit 31 (MSB) = address 31B	Per bit:	
			0	slave not projected
			1	slave projected

Output parameters

17016

Parameter	Data type	Description	Possible values	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0019 Master not in the projecting mode

Set_Mode

16021

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

16155

The FB changes the operating mode (protected operation, projection mode) of the selected AS-i master.

Input parameters

16156

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2
enMode_ASi_Master	ASI_MASTER_MODE	Operating mode of the AS-i master	Closed_mode	protected mode active
			Project_mode	projection mode active

Output parameters

17018

Parameter	Data type	Description	Possible values	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0003 Slave with address 0 found (slave not detected)

Set_PCD

16022

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

16151

The FB changes the configuration file (Permanent Configuration Data = PCD) of the slaves at the selected AS-i master.

Input parameters

16160

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2
awPCD	ARRAY [0...63] OF WORD	Permanent configuration files of the slaves on the selected AS-i master	per word: Bits 0...3: I/O-Code Bits 4...7: ID-Code Bits 8-11: ID1-Code Bits 12-15: ID2-Code Data in Word 0 is invalid!	

Output parameters

15574

Parameter	Data type	Description	Possible values	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0019 Master not in the projecting mode

Set_ProjectAll

16023

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

16161

The FB starts the projection adaptation on the selected AS-i master.

Input parameters

16125

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2

Output parameters

17020

Parameter	Data type	Description	Possible values	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0003 Slave with address 0 found (slave not detected)
- 0x0019 Master not in the projecting mode

Set_SlaveAddress

16024

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

16162

The FB changes the address of the selected AS-i slaves.

Input parameters

16165

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2
enASi_Slave	ASI_SLAVE	Address of the AS-i slave	Slave_n	AS-i slave to address n (n = 1 ... 31)
enASi_SlaveTyp	ASI_SLAVE_ TYP	Type of the AS-i slave	SA_Slave	Single or A slave
			B_Slave	B-Slave
enASi_Slave_new	ASI_SLAVE	New address of the AS-i slave	Slave_n	AS-i slave at address n (n = 1 ... 31)
enASi_SlaveTyp_new	ASI_SLAVE_ TYP	New type of the AS-i slave	SA_Slave	single slave or A slave
			B_Slave	B slave

Output parameters

17021

Parameter	Data type	Description	Possible values	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0001 Slave does not respond or change to offline mode during FB execution
- 0x0002 Slave with old address not found (slave not detected)
- 0x0003 Slave with address 0 found (slave not detected)
- 0x0004 Slave with new address found
- 0x0005 Error during deletion of the old address (Delete Error)
- 0x0006 ExtendedID1 could not be read after writing (Read Error)
- 0x0007 Error when writing ExtendedID1 (Set Error)
- 0x0008 New address temporary stored
- 0x0009 ExtendedID1 stored temporarily
- 0x0018 Master is not in normal operation.

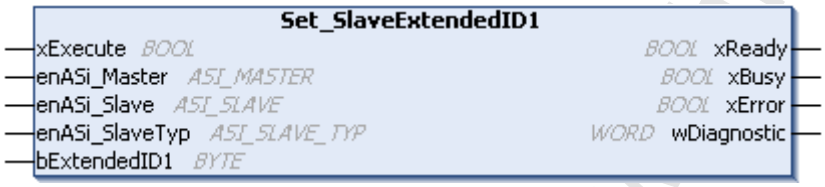
Set_SlaveExtendedID1

16025

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:



Description

16169

The FB changes the Extended ID1 of the selected AS-i slave.

Input parameters

16170

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2
enASi_Slave	ASI_SLAVE	Address of the AS-i slave	Slave_n	AS-i slave to address n (n = 1 ... 31)
enASi_SlaveTyp	ASI_SLAVE_ TYP	Type of the AS-i slave	SA_Slave	Single or A slave
			B_Slave	B-Slave
bExtendedID1	BYTE	Extended ID1 code of the selected AS-i slave	Extended ID1 code (hexadecimal representation)	

Output parameters

16148

Parameter	Data type	Description	Possible values	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0003 Slave with address 0 found (slave not detected)
- 0x0005 Error during deletion of the old address (Delete Error)
- 0x0006 ExtendedID1 could not be read after writing (Read Error)
- 0x0007 Error when writing ExtendedID1 (Set Error)
- 0x0009 ExtendedID1 stored temporarily
- 0x000E Invalid slave address (e.g. 0 or 0B specified)
- 0x0018 Master is not in normal operation.
- 0x0021 Invalid ExtendedID1 code

Set_SlaveParameter

16026

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

16172

The FB changes the parameters of the selected AS-i slave.

Input parameters

16173

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2
enASi_Slave	ASI_SLAVE	Address of the AS-i slave	Slave_n	AS-i slave to address n (n = 1 ... 31)
enASi_SlaveTyp	ASI_SLAVE_TYP	Type of the AS-i slave	SA_Slave	Single or A slave
			B_Slave	B-Slave
enASi_SlaveParam	BYTE	Parameters of the selected AS-i slave	Slave parameters (hexadecimal representation)	

Output parameters

17025

Parameter	Data type	Description	Possible values	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0001 Slave does not respond or change to offline mode during FB execution
- 0x000A Sslave not in LAS
- 0x000B Data content invalid (e.g. parameter value >7 for A/B slaves)
- 0x000E Invalid slave address (e.g. 0 or 0B specified)
- 0x0018 Master is not in normal operation.

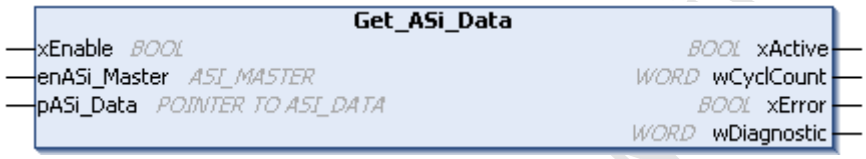
Get_ASi_Data

16000

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:



Description

16033

The FB reads the following data from the selected AS-i master and stores the values in a structure variable of data type ASI_DATA (→ **ASI_DATA (STRUCT)** (→ p. 172)):

- List of activated slaves – LAS
- List of detected slaves – LDS
- List of projected slaves – LPS
- List of configuration errors – LCE
- List of configuration errors - additional slave – LCEAS
- List of configuration errors - missing slave – LCEMS
- List of peripheral faults – LPF
- List of double address errors – LDAE
- Configuration data image – CDI
- Projected configuration data – PCD
- Input parameters of the AS-i slaves
- Outputs parameters of the AS-i slaves

Input parameters

16034

Parameter	Data type	Description	Possible values	
xEnable	BOOL	Control activiy of the FB	TRUE	FB is enabled
			FALSE	FB is disabled
enASi_Master	ASI_MASTER	Select AS-i master	Master_1	AS-i master 1
			Master_2	AS-i master 2
pASi_Data	POINTER TO ASI_DATA	Structure variable in which the read data is to be stored.	Variable must be declared!	

Output parameters

16035

Parameter	Data type	Description	Possible values	
xActive	BOOL	Signal confirms the FB execution	FALSE	Function block is inactive
			TRUE	FB is active (=is executed)
wCycleCount	WORD	Counter for FB cycles that were fully run through	Number in hexadecimal representation	
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0xnF01 Unknown error [1]
- 0xnF02 Unknown/invalid target [1]
- 0xnF03 Unknown command ID [1]
- 0xnF04 Unknown parameters [1]
- 0xnF05 Timeout during processing [1]

Legend:

[1] Get_ASi_Data executes the FB ACnnnn_SysCmd sequentially with different command IDs in order to determine the individual elements of the complex variable. In the returned error code, the nibble n indicates the command request where the error occurred. n can have the following values:

1 = error with "Get LAS, LDA, LPF, LCE"
2 = error with "Get LPS"
3 = error with "Get CDI"
4 = error with "Get PCD"
5 = error with "Get Input Parameter"
6 = error with "Get Output Parameter"
7 = error with "Get LCEMS, LCEAS, LDAE"

10.1.2 Overview: System functions (FB_System)

Contents

Get_FieldbusInfo 164
QuickSetupASi_Master 166
Set_TimeDate.....168

17460

Get_FieldbusInfo

17453

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

17454

The FB reads information about the fieldbus and provides the values in a list. The following information are read:

- Status of the fieldbus
- Fieldbus type
- Status of the Ethernet connection at port X6 and X7
- MAC addresses
- IP address of the fieldbus host

Input parameters

17455

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution

Output parameters

17456

Parameter	Data type	Description	Possible values	
aw_InfoList	ARRAY[0...18] OF WORD	Fieldbus information	Position of the data within the individual words → **DS18: Fieldbus information**	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0F01 Unknown error
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

QuickSetupASi_Master

15999

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

15902

The FB executes the quick setup routine on the selected AS-i masters.

Input parameters

15903

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
xMaster1	BOOL	Select AS-i master 1 for quick setup	FALSE	No execution of quick setup, AS-i configuration remains unchanged.
			TRUE	Execution of quick set up on AS-i master
xMaster2	BOOL	Select AS-i master 2 for quick setup	FALSE	No execution of quick setup, AS-i configuration remains unchanged.
			TRUE	Execution of quick set up on AS-i master

Output parameters

15920

Parameter	Data type	Description	Possible values	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0003 Slave with address 0 found (slave not detected)

Set_TimeDate

15998

Function block type: Function block (FB)

Library: ACnnnn_Utils.library

Symbol in CODESYS:

Description

15949

The FB sets the system time (time and date) of the device using the transmitted input values.

Input parameters

15988

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
bDay	BYTE	Day	0x01 ... 0x1F	1 ... 31
bMonth	BYTE	Month	0x01 ... 0x0C	January ... December
wYear	WORD	Year	0x07B3 ... 0x07F5	1971 ... 2037
bHour	BYTE	Hour	0x00 ... 0x17	0 ... 23
bMinute	BYTE	Minute	0x00 ... 0x3B	0 59
bSecond	BYTE	Second	0x00 ... 0x3B	0 59

Output parameters

15990

Parameter	Data type	Description	Possible values	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xBusy	BOOL	Signal indicates if the FB is executed.	FALSE	FB is deactivated or FB execution is terminated.
			TRUE	FB execution is started but not yet terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
wDiagnostic	WORD	Diagnostic information	→ List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 No specific error is set
- 0x0001 Transferred values for date/time are invalid and could not be set.
- 0x0002 NTP active, time could not be adopted.

10.1.3 Enumeration types and complex variables

Contents

Enumeration types (ENUM) 170
Complex variables (STRUCT) 172

15986

In addition to the standard data types, the CODESYS package from ifm electronic also features the following enumeration types (ENUM) and complex variables (STRUCT):

Enumeration types (ENUM)

16176

The library `ACnnnn_Utils` provides the following enumeration types (ENUM):

ASI_ADDRESS_MODE (ENUM)

16177

Designation	Description	Variable	Data type	Value
ASI_ADDRESS_MODE	AS-i autoaddressing mode	▪ Auto_address_enable	INT	0
		▪ Auto_address_disable	INT	1

ASI_MASTER (ENUM)

16178

Designation	Description	Variable	Data type	Value
ASI_MASTER	Identifier for AS-i masters	▪ Master_1	INT	1
		▪ Master_2	INT	2

ASI_MASTER_MODE (ENUM)

16179

Designation	Description	Variable	Data type	Value
ASI_MASTER_MODE	Operating mode of the AS-i master	▪ Closed_mode	INT	0
		▪ Project_mode	INT	1

ASI_SLAVE (ENUM)

16180

Designation	Description	Variable	Data type	Value
ASI_SLAVE	Identifier for AS-i slaves	▪ Slave_1	INT	1
		▪ Slave_2	INT	2
		▪ Slave_3	INT	3
		▪ Slave_4	INT	4
		▪ Slave_5	INT	5
		▪ Slave_6	INT	6
		▪ Slave_7	INT	7
		▪ Slave_8	INT	8
		▪ Slave_9	INT	9
		▪ Slave_10	INT	10
		▪ Slave_11	INT	11
		▪ Slave_12	INT	12
		▪ Slave_13	INT	13
		▪ Slave_14	INT	14
		▪ Slave_15	INT	15
		▪ Slave_16	INT	16
		▪ Slave_17	INT	17
		▪ Slave_18	INT	18
		▪ Slave_19	INT	19
		▪ Slave_20	INT	20
		▪ Slave_21	INT	21
		▪ Slave_22	INT	22
		▪ Slave_23	INT	23
		▪ Slave_24	INT	24
		▪ Slave_25	INT	25
		▪ Slave_26	INT	26
		▪ Slave_27	INT	27
		▪ Slave_28	INT	28
		▪ Slave_29	INT	29
		▪ Slave_30	INT	30
		▪ Slave_31	INT	31

ASI_SLAVE_TYP (ENUM)

16181

Designation	Description	Variable	Data type	Value
ASI_SLAVE_TYP	Type of the AS-i slave	▪ SA_Slave	INT	0
		▪ B_Slave	INT	1

Complex variables (STRUCT)

15991

The library `ACnnnn_Utils.library` provides complex variables of the data type STRUCT. They are used by the FBs, but they can also be used by the programmer in CODESYS projects for the device-internal standard PLC.

ASI_DATA (STRUCT)

15992

Name	Data type	Description	Possible values
▪ LDS_SA_Slave	DWORD	List of the active S/A slaves	Each bit represents an AS-i address: 0 = no slave active 1 = slave active → **DS9 – Slave lists LAS, LDS, LPF, LCE**, words 4...7
▪ LDS_B_Slave	DWORD	List of active B slaves	
▪ LAS_SA_Slave	DWORD	List of active S/A slaves	Each bit represents an AS-i address: 0 = no active slave 1 =active slave → **DS9 – Slave lists LAS, LDS, LPF, LCE**, words 0...3
▪ LAS_B_Slave	DWORD	List of active B slaves	
▪ LPF_SA_Slave	DWORD	List of peripheral faults (S/A slaves)	Each bit represents an AS-i address: 0 = no peripheral fault 1 = peripheral fault → **DS9 – Slave lists LAS, LDS, LPF, LCE**, words 8...11
▪ LPF_B_Slave	DWORD	List of peripheral faults (B slaves)	
▪ LCE_SA_Slave	DWORD	List with configuration errors (S/A slaves)	Each bit represents an AS-i address: 0 = no configuration error 1 = configuration error → **DS9 – Slave lists LAS, LDS, LPF, LCE**, words 12...15
▪ LCE_B_Slave	DWORD	List with configuration errors (B slaves)	
▪ LPS_SA_Slave	DWORD	List of projected S/A slaves	Each bit represents an AS-i address: 0 = no projected slave 1 = projected slave → **DS10 – Slave list LPS**
▪ LPS_B_Slave	DWORD	List of active B slaves	
▪ LCEMS_SA_Slave	DWORD	List of configuration errors: projected, but missing S/A slave	Each bit represents an AS-i address: 0 = no configuration error 1 = configuration error → **DS17 – AS-i master: Error lists LCEMS, LCEAS, LDAE**, words 0...3
▪ LCEMS_B_Slave	DWORD	List of configuration errors: projected, but missing B slave	
▪ LCEAS_SA_Slave	DWORD	List of configuration errors: additional S/A slave	Each bit represents an AS-i address: 0 = no error 1 = error → **DS17 – AS-i master: Error lists LCEMS, LCEAS, LDAE**, words 4...7
▪ LCEAS_B_Slave	DWORD	List of configuration errors: additional B slave	
▪ LDAE_SA_Slave	DWORD	List of double address errors (S/A slaves)	Each bit represents an AS-i address: 0 = no double address error 1 = double address error → **DS17 – AS-i master: Error lists LCEMS, LCEAS, LDAE**, words 8...11
▪ LDAE_B_Slave	DWORD	List of double address errors (B slaves)	
▪ CDI	ARRAY[0..63] OF WORD	Image of the current configuration data (CDI = Configuration Data Image)	One word is available per slave: Bits 0...3 = IO code Bits 4...7 = ID code Bits 8...11 = Extended ID1 code Bits 12...15 = Extended ID2 code → **DS11 – Actual configuration data (CDI)**

Name	Data type	Description	Possible values
▪ PCD	ARRAY[0..63] OF WORD	Image of the projected configuration data (PCD = Projected Configuration Data)	Each word contains data of one slave: Bits 0...3 = IO Bits 4...7 = ID Bits 8...11 = ID1 Bits 12...15 = ID2 → **DS12 – Projected configuration data (PCD)**
▪ InputParam_SA_Slave	ARRAY[0..31] OF BYTE	Input parameters of the S/A slaves	Each byte contains parameters of one S/A slave: Bits 0...3 = P0-P3 Bits 4...7 = reserved → **DS13 – Image of the input parameters of the slaves (PI)**, words 0...15
▪ InputParam_B_Slave	ARRAY[0..31] OF BYTE	Input parameters of the B slaves	Each byte contains parameters of one B slave: Bits 0...3 = P0-P3 Bits 4...7 = reserved → **DS13 – Image of the input parameters of the slaves (PI)**, words 16...31
▪ OutputParam_SA_Slave	ARRAY[0..31] OF BYTE	Output parameters of the S/A slaves	Each byte contains parameters of one S/A slave: Bits 0...3 = P0-P3 Bits 4...7 = reserved → **DS14 – Image of the output parameters of the slaves (PP)**, words 0...15
▪ OutputParam_B_Slave	ARRAY[0..31] OF BYTE	Output parameters of the B slaves	Each byte contains parameters of one B slave: Bits 0...3 = P0-P3 Bits 4...7 = reserved → **DS14 – Image of the output parameters of the slaves (PP)**, words 16...31

ASI_NET (STRUCT)

15993

The structure contains the complete process image (inputs and outputs) of an AS-i network.

Name	Data type	Description	Possible values
▪ binIO	ASI_BIN_IO	Binary input and output data	→ **ASI_BIN_IO (STRUCT)** (→ p. 174)
▪ anaIO	ASI_ANA_IO	Analogue input and output data	→ **ASI_ANA_IO (STRUCT)** (→ p. 175)

-

ASI_BIN_IO (STRUCT)

16002

The structure contains the process data of the digital input and output slaves:

Name	Data type	Description	Possible values
▪ bin_IN_Slaves	ASI_BIN_IN	Input data of the digital AS-i slaves	→ **ASI_BIN_IN (STRUCT)** (→ p. 175)
▪ bin_OUT_Slaves	ASI_BIN_OUT	Output data of the digital AS-i slaves	→-**ASI_BIN_OUT (STRUCT)** (→ p. 175)

ASI_BIN_IN (STRUCT)

16004

The structure contains the input data of the digital AS-i slaves:

Name	Data type	Description	Possible values
▪ SA_Slave	ARRAY[1...31] OF BYTE	Input data of the digital S/A slaves, 1 byte per S/A slave	corresponds to the 1st to 15th word of the acyclic data record 2 (DS2) → **DS2 – Digital inputs of the slaves and master flags**
▪ B_Slave	ARRAY[1...31] OF BYTE	Input data of the digital B slaves, 1 byte per B slave	corresponds to the 16th to 31st word of the acyclic data record 2 (DS2) → **DS2 – Digital inputs of the slaves and master flags**

ASI_BIN_OUT (STRUCT)

16003

The structure contains the output data of the digital AS-i slaves.

Name	Data type	Description	Possible values
▪ SA_Slave_bin_OUT	ARRAY[1...31] OF BYTE	Output data of the digital S/A slaves (1 byte per S/A slave)	corresponds to the 1st to 15th word of the acyclic data record 5 (DS5) → **DS5 – Digital outputs of the slaves**
▪ B_Slave_bin_OUT	ARRAY[1...31] OF BYTE	Output data of the digital B slaves (1 byte per B slave)	corresponds to the 16th to 31st word of the acyclic data record 5 (DS5) → **DS5 – Digital outputs of the slaves**

ASI_ANA_IO (STRUCT)

16001

The structure contains the process data of the analogue input and output slaves.

Name	Data type	Description	Possible values
▪ ana_IN_Slave	ARRAY[1..31] OF ASI_ANALOG_IN	Input data and status flags of the analogue AS-i slaves	→ **ASI_ANALOG_IN (STRUCT)** (→ p. 176)
▪ ana_OUT_Slave	ARRAY[1..31] OF ASI_ANALOG_OUT	Output data and status flags of the analogue AS-i slaves	→ **ASI_ANALOG_OUT (STRUCT)** (→ p. 177)

ASI_ANALOG_IN (STRUCT)

15994

The structure contains the process data of the analogue input slave as well as the transmitted status flags.

Designation	Data type	Description	Possible values
▪ chan_1	INT	Analogue value channel 1: S or A input slave	corresponds to the 1st to 4th word of an area comprising 5 words in the acyclic data records 3 and 4 (DS3+4) → **DS3 – Analogue inputs of slaves 1(A)...15(B)** → **DS4 – Analogue inputs of slaves 16(A)...31(B)**
▪ chan_2	INT	Analogue value channel 2: S or A input slave	
▪ chan_3	INT	Analogue value channel 3: S or B input slave	
▪ chan_4	INT	Analogue value channel 4: S or B input slave	
▪ flags	ASI_ANALOG_OUT_FLAGS	Status flags	→ **ASI_ANALOG_IN_FLAGS (STRUCT)** (→ p. 176)

ASI_ANALOG_IN_FLAGS (STRUCT)

15996

The structure contains the status flags of the analogue input slave.

Name	Data type	Description	Possible values
▪ V0	BOOL	Valid bit, channel 1	corresponds to the fifth word of an area comprising 5 words in the acyclic data records 3 and 4 (DS3+4) → **DS3 – Analogue inputs of slaves 1(A)...15(B)** → **DS4 – Analogue inputs of slaves 16(A)...31(B)**
▪ O0	BOOL	Overflow bit, channel 1	
▪ V1	BOOL	Valid bit, channel 2	
▪ O1	BOOL	Overflow bit, channel 2	
▪ V2	BOOL	Valid bit channel 3	
▪ O2	BOOL	Overflow bit, channel 3	
▪ V3	BOOL	Valid bit, channel 4	
▪ O3	BOOL	Overflow bit, channel 5	
▪ na1	BOOL	--	
▪ TOA	BOOL	Transfer output, S/A slave	
▪ na2	BOOL	--	
▪ TOB	BOOL	Transfer output, B slave	
▪ na3	BOOL	--	
▪ TIA	BOOL	Transfer input, S/A slave	
▪ na4	BOOL	--	
▪ TIB	BOOL	Transfer input, B slave	

ASI_ANALOG_OUT (STRUCT)

15995

The structure represents the process data of the analogue output slaves as well as the transmitted flags.

Designation	Data type	Description	Possible values
▪ chan_1	INT	Analogue value channel 1: S or A output slave	corresponds to the acyclic data records 6 and 7 (DS6, DS7) → **DS6 – Analogue outputs of slaves 1(A)...15(B)** → **DS7 – Analogue outputs of slaves 16(A)...31(B)**
▪ chan_2	INT	Analogue value channel 2: S or A output slave	
▪ chan_3	INT	Analogue value channel 3: S or B output slave	
▪ chan_4	INT	Analogue value channel 4: S or B output slave	
▪ flags	ASI_ANALOG_OUT_FLAGS	Status flags	→**ASI_ANALOG_OUT_FLAGS (STRUCT)** (→ p. 177)

ASI_ANALOG_OUT_FLAGS (STRUCT)

15997

The structure contains the status flags of the analogue output slave.

Name	Data type	Description	Possible values
▪ na1	BOOL	--	corresponds to the acyclic data record 8 (DS8) → **DS8 – Status flags of analogue output data of the slaves 1...31**
▪ na2	BOOL	--	
▪ na3	BOOL	--	
▪ na4	BOOL	--	
▪ na5	BOOL	--	
▪ na6	BOOL	--	
▪ na7	BOOL	--	
▪ na8	BOOL	--	
▪ OVA	BOOL	Output valid, S/A slave	
▪ na9	BOOL	--	
▪ OVB	BOOL	Output valid, B slave	
▪ n10	BOOL	--	
▪ TOA	BOOL	Transfer Output, S/A slave	
▪ na11	BOOL	--	
▪ TOB	BOOL	Transfer Output, B slave	
▪ na12	BOOL	--	

10.2 Library ACnnnn_SYS_CMD.library

Contents

ACnnnn_SysCmd 178

17723

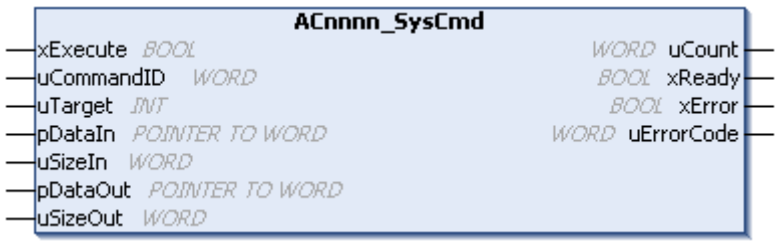
10.2.1 ACnnnn_SysCmd

15890

Function block type: Function block (FB)

Library: ACnnnn_SYS_CMD.library

Symbol in CODESYS:



Description

15893

Using the FB, individual commands can be sent to the system or to an AS-i master.
Each command refers to the data structures of one of the following elements:

- Command request channel: → **Example: Change language setting of the device** (→ p. 183)
- Command response channel: → **Example: Date / time / read NTP settings** (→ p. 184)
- Acyclic data record (DSx): → **Example: Read LCEMS, LCEAS and LDAE of AS-i master 2** (→ p. 185)

Input parameters

15895

Parameter	Data type	Description	Possible values	
xExecute	BOOL	Control execution of the FB	FALSE	Stop FB execution
			TRUE	Start one-time FB execution
uCommandID	WORD	ID of the command to be executed	→ **Table: System commands** (→ p. 179) → **Table: AS-i master commands** (→ p. 180)	
uTarget	INT	Device component to which the command is to be sent	0	system
			1	AS-i master 1
			2	AS-i master 2
pDataIn	POINTER TO WORD	Buffer for data that is assigned to the command as input parameters.	Commando parameter described in the command request channel (word 3 to 120)	
uSizeIn	WORD	Size of the buffer for the input parameters (number of bytes)		
pDataOut	POINTER TO WORD	Buffer for data that is returned as output parameters (results)	Return values described in the command reply channel (data as of word 5)	
uSizeOut	WORD	Size of the buffer for the output parameters (number of bytes)		

Table: System commands

17053

uCommandID	Description
0x0001	DS1 – System information
0x1101	Command 0x0101 – Quick set-up AS-i masters 1 + 2
0x1103	Command 0x0103 – Select user language
0x1104	Command 0x0104 – Change display settings
0x1105	Command 0x0105 – Set output control
0x1109	Command 0x0109 – Set date / time
0x110A	Command 0x010A – Configure the NTP server settings
0x110B	Command 0x010B – Read date / time / NTP settings
0x110C	Command 0x010C – Reboot system
0x110D	Command 0x010D – Read fieldbus info
0x1110	Command 0x0110 - Display target visualisation

Only execute system commands with the input parameter `uTarget = 0`!

Detailed information about the acyclic data sets and the command interface is given in the supplement to the device manual of the SmartSPS SafeLine AC4S (→ **Overview: User documentation for AC4S** (→ p. 7)).

Table: AS-i master commands

17054

uCommandID	Description / corresponding command
0x0002	DS2 – Digital slave inputs and master flags
0x0003	DS3 – Analogue inputs of the slaves 1(A)...15(B)
0x0004	DS4 – Analogue inputs of the slaves 16(A)...31(B)
0x0005	DS5 – Digital slave outputs
0x0006	DS6 – Analogue outputs of the slaves 1(A)...15(B)
0x0007	DS7 – Analogue outputs of the slaves 16(A)...31(B)
0x0008	DS8 – Status flags of the analogue outputs of slaves 1...31
0x0009	DS9 – Slave lists LAS, LDS, LPF, LCE
0x000A	DS10 – Slave list LPS
0x000B	DS11 – Current configuration data (CDI)
0x000C	DS12 – Projected configuration data (PCD)
0x000D	DS13 – Image of the input parameters of the slaves (PI)
0x000E	DS14 – Image of the outputs parameters of the slaves (PP)
0x000F	DS15 – Slave error counter, configuration error counter, AS-i cycle counter
0x0011	DS17 – AS-i master: Error lists LCEMS, LCEAS, LDAE
0x1001	Command 0x0001 – Change AS-i slave parameters
0x1003	Command 0x0003 – Project current AS-i network
0x1004	Command 0x0004 – Change LPS
0x1005	Command 0x0005 – Change the operating mode of the AS-i master
0x1006	Command 0x0006 – Change AS-i slave address
0x1007	Command 0x0007 - Set auto address mode of the AS-i master
0x1009	Command 0x0009 – Change extended ID1 in the AS-i slave
0x100A	Command 0x000A - Change PCD
0x100D	Command 0x000D – AS-i master supply voltage, symmetry, earth fault
0x1015	Command 0x0015 – Read ID string of an AS-i profile (S-7.4)
0x101A	Command 0x001A – Read AS-i master info
0x101C	Command 0x001C – Deactivate slave reset when changing to the protected mode
0x1021	Command 0x0021 - Read diagnostics string of an AS-i slave (S-7.4)
0x1022	Command 0x0022 - Read parameter string of an AS-i slave (S-7.4)
0x1023	Command 0x0022 - Write parameter string of an AS-i slave (S-7.4)
0x1024	Command 0x0024 – CTT2 standard read
0x1025	Command 0x0025 – CTT2 standard write
0x1026	Command 0x0026 – CTT2 vendor specific read
0x1027	Command 0x0027 – CTT2 vendor specific selective write
0x1040	Command 0x0040 – CTT2 device group read
0x1041	Command 0x0041 – CTT2 device group write
0x1042	Command 0x0042 – CTT2 vendor specific selective read from buffer
0x1043	Command 0x0043 – CTT2 vendor specific selective write from buffer

uCommandID	Description / corresponding command
0x1044	Command 0x0044 – CTT2 vendor specific selective read
0x1045	Command 0x0045 – CTT2 vendor specific selective write
0x1046	Command 0x0046 – CTT2 device group selective read
0x1047	Command 0x0047 – CTT2 device group selective write
0x1049	Command 0x0049 – CTT2 vendor specific exchange
0x104A	Command 0x004A – CTT2 device group exchange
0x104B	Command 0x004B – CTT2 device group selective read from buffer
0x104C	Command 0x004C – CTT2 device group selective write from buffer
0x1050	Command 0x0050 – Set AS-i master settings
0x1051	Command 0x0051 – Reset error counter

Only execute AS-i master commands with input parameter `uTarget = 1` or `2`!

Detailed information about the acyclic data sets and the command interface is given in the supplement to the device manual of the SmartSPS SafeLine AC4S (→ **Overview: User documentation for AC4S** (→ p. 7)).

Output parameters

15897

Parameter	Data type	Description	Possible values	
uCount	WORD	Number of valid bytes in pDataOut (uCount <= uSizeOut)	integer value in hexadecimal representation	
xReady	BOOL	Signal indicates if the execution of the FB is terminated.	FALSE	FB is inactive or being executed.
			TRUE	FB execution is terminated.
xError	BOOL	Signal indicates if errors occurred while the FB was executed.	FALSE	FB is disabled or presently executed or FB was executed without error.
			TRUE	An error occurred when the FB was executed.
uErrorCode	WORD	Error code of the executed command	Command error code of the corresponding command channel OR: → List below (Diagnostic codes)	

Diagnostic codes:

- 0xnF01 Unknown error [1]
- 0x0F02 Unknown/invalid target
- 0x0F03 Unknown command ID
- 0x0F04 Invalid parameters
- 0x0F05 Timeout during processing

Example: Change language setting of the device

17051

Task: Set the language of the graphic user surface of the device to "Spanish" with the FB ACnnnn_SysCmd.

Command type: Command request channel

Input parameters of the FB:

Parameter	Value	Explanation
uCommandID	0x1103	Corresponding commando channel: → **Command 0x0103 – Select user language** (→ device manual supplement)
uTarget	0	System command
pDataIn	arDataIn	▪ Variable of the data type Array of Words ▪ arDataIn contains the command parameters
uSizeIn	0x0001	arDataIn only consists of 1 line since word 1 and 2 as well as 4 to 120 of the command request channel are not considered.
pDataOut	--	irrelevant since command request channel
uSizeOut	--	irrelevant since command request channel

Content of arDataIn:

Word no.	Contents	Explanation
1	0x4553	Spanish

Example: Date / time / read NTP settings

17423

Task: Read the current system time and the NTP settings with the FB ACnnnn_SysCmd.
Command type: Command reply channel
Input parameters of the FB:

Parameter	Value	Declaration
uCommandID	0x110B	Corresponding commando channel: → **Command 0x010B – Read date / time / NTP settings** (→ device manual supplement)
uTarget	0	System command
pDataIn	--	irrelevant, since command reply channel
uSizeIn	--	irrelevant, since command reply channel
pDataOut	arDataOut	▪ variable of the data type Array of Words ▪ contains the return values of the reply channel
uSizeOut	0x0007	Array consists of 7 lines since the reply channel returns 7 words (words 5 ... 11).

Content of arDataOut:

Word no.	Content	
1	Month	Day
2	Year	Year
3	Minutes	Hours
4	reserved	Seconds
5	NTP offset	NTP status
6 ... 7	IP address NTP server	

Example: Read LCEMS, LCEAS and LDAE of AS-i master 2

17424

Task: Read the error lists LCEMS, LCEAS and LDAE of the AS-i master 2 with the FB ACnnnn_SysCmd.

Command type: acyclic data set

Input parameters of the FB:

Parameter	Value	Declaration
uCommandID	0x0011	Corresponding acyclic data set: → **DS17 – AS-i master: Error lists LCEMS, LCEAS, LDAE** (→ device manual supplement)
uTarget	1	Master command (1 = AS-i master 2))
pDataIn	--	irrelevant
uSizeIn	--	irrelevant
pDataOut	arDataOut	▪ Variable of the data type Array of Words ▪ contains the return values
uSizeOut	0x000C	Array consists of 12 lines since the data set DS17 has exactly 12 words

Content of arDataOut:

Word no.	Content
1 ... 4	LCEMS (list of configuration errors - missing slaves)
5 ... 8	LCEAS (list of configuration errors - additional slaves)
9 ... 12	LDAE (list of double address errors)

10.3 Library SF_IO.library

Contents

CtrlASi_InSlave 187
CtrlASi_OutSlave....189
CtrlASi_ResetAllSlaves 192
CtrlLocalInputs....194
Ctrl_SetDiagInfo 196
GetLocalInput 198
SetLocalOutput....199

18930

The library SF_IO.library from ifm electronic provides function blocks (FB) for the programming of safety-relevant PLC applications.

10.3.1 CtrlASi_InSlave

18240

Function block type: Function block (FB)

Library: sf_io.library

Symbol in CODESYS:

Description

18239

Function block (FB) to control the logical device of a safe AS-i input slave (ASi_SlaveAdr, ASi_Master).

The programmer can carry out the following actions via CtrlASi_InSlave:

- Unlock the logical device and reset it from the locked error state S_ERROR to the initialisation state S_INIT (RESET)

The FB provides the following state and diagnostic information as non-safe data:

- Logical values of the two safe code half-sequences of the AS-i input slave (Chan_A, Chan_B)
- State of the FB processing (ready)
- Error state of the function block (Error)
- Diagnostic information of the logical device (DiagCode)

The function block can be instanced as often as required within the limits of CODESYS.

Input parameters

18237

Parameter	Data type	Meaning	Possible values	
Enable	BOOL	FB activation	FALSE	FB is deactivated.
			TRUE	FB is activated.
ASi_SlaveAdr	INT	Address of the safe AS-i slave	1 ... 31	Slave address 1 ... slave address 31
ASi_Master	INT	AS-i master to which the safe AS-i slave is connected.	1	AS-i master 1
			2	AS-i master 2
Reset	BOOL	Control signal to reset the logical device from the locked error state (S_ERROR) to the initialisation state (S_INIT).	FALSE ⇨ TRUE	Control signal is transmitted
			other	No changes

The safe AS-i slave that is to be controlled by means of the FB must be inserted in the device tree.

Output parameters

18238

Parameter	Data type	Meaning	Possible values	
Chan_A	BOOL	Logical state of the safe signal channel A (= 1st code half-sequence)	FALSE	OFF
			TRUE	ON
Chan_B	BOOL	Logical state of the safe signal channel B (= 2nd code half-sequence)	FALSE	OFF
			TRUE	ON
Ready	BOOL	Status of the FB processing (validity of the output signals)	FALSE	FB not completely processed or deactivated; signals are invalid
			TRUE	FB completely processed; signals are valid
Error	BOOL	Error indication	FALSE	No error
			TRUE	Logical device is in the locked error state S_ERROR
DiagCode	WORD	Diagnostic code of the internal states of the controlled logical device	Scope of the diagnostic code depends on the controlled logical device: Description of the selected logical device (→ **Library SF_LogicalInterfaces.library** (→ p. 201))	

10.3.2 CtrlASi_OutSlave

18244

Function block type: Function block (FB)

Library: sf_io.library

Symbol in CODESYS:

Description

18243

Function block (FB) to control the logical device of a safe AS-i control slave (ASi_SlaveAdr, ASi_Master)

The programmer can carry out the following actions via CtrlASi_OutSlave:

- Transmit signal to unlock the safe AS-i slaves (auxiliary signal HSI_1)
- Transmit signal for automatic new start (auxiliary signal HSI_2)

The FB provides the following state and diagnostic information:

- State of the FB processing (ready)
- Error state of the function block (Error)
- Diagnostic information of the logical device (DiagCode)

The function block can be instanced as often as required within the limits of CODESYS.

Input parameters

18242

Parameter	Data type	Meaning	Possible values	
Enable	BOOL	FB activation	FALSE	FB is deactivated.
			TRUE	FB is activated.
ASi_SlaveAdr	INT	Address of the safe AS-i slave	1 ... 31	Slave address 1 ... slave address 31
ASi_Master	INT	AS-i master to which the safe AS-i slave is connected.	1	AS-i master 1
			2	AS-i master 2
HSI_1	BOOL	Auxiliary signal 1 (HSI_1) Reset signal to exit the locked error state (S_Error)	FALSE ⇨ TRUE	Auxiliary signal is transmitted once
			else	No action
HSI_2	BOOL	Auxiliary signal 2 (HSI_2) Control of the start-up behaviour of the controlled AS-i slave after communication error	FALSE ⇨ TRUE	Auxiliary signal is transmitted once
			else	No action

The safe AS-i slave that is to be controlled by means of the FB must be inserted in the device tree.

A triggered request of the two auxiliary signals must not be cancelled until the success or interruption of the transmission has been displayed.

► Do not set the inputs HSI_1 and/or HSI_2 to FALSE if a value with HS_ACK = OK or HS_ACK = nOK is provided at output DiagCode (→ **Output parameters** (→ p. 191)).

Detailed information about the auxiliary signals HSI_1 and HSI_2: → Operating instructions of the safe AS-i control slave

Output parameters

18241

Parameter	Data type	Meaning	Possible values	
Ready	BOOL	Status of the FB processing (validity of the output signals)	FALSE	FB not completely processed or deactivated; signals are invalid
			TRUE	FB completely processed; signals are valid
Error	BOOL	Error indication	FALSE	No error
			TRUE	Logical device is in the locked error state S_ERROR
DiagCode	WORD	Diagnostic code of the internal states of the controlled logical device	Scope of the diagnostic code depends on the controlled logical device: Description of the selected logical device (→ **Library SF_LogicalInterfaces.library** (→ p. 201))	

Diagnostic codes:

- 0x8000 Output = ON; HS_ACK = idle
- 0x8002 Output = OFF; HS_ACK = idle
- 0x8830 Output = ON; HS_ACK = OK
- 0x8832 Output = OFF; HS_ACK = OK
- 0x8840 Output = ON; HS_ACK = nOK
- 0x8842 Output = OFF; HS_ACK = nOK
- 0x8850 Output = ON; HS_ACK = RUN
- 0x8852 Output = OFF; HS_ACK = RUN

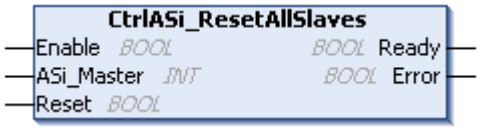
10.3.3 CtrlASi_ResetAllSlaves

18254

Function block type: Function block (FB)

Library: sf_io.library

Symbol in CODESYS:



Description

18253

Function block (FB) to control the logical device of several safe AS-i input slaves of an AS-i network (ASi_Master).

The programmer can carry out the following actions via CtrlASi_ResetAllSlaves:

- Unlock the logical device of all safe AS-i slaves in the indicated network and reset it from the locked error state S_ERROR to the initialisation state S_INIT (RESET)

The FB provides the following state and diagnostic information:

- State of the FB processing (ready)
- Error state of the function block (Error)

Input parameters

18252

Parameter	Data type	Meaning	Possible values	
Enable	BOOL	FB activation	FALSE	FB is deactivated.
			TRUE	FB is activated.
ASi_Master	INT	AS-i master to which the safe AS-i slave is connected.	1	AS-i master 1
			2	AS-i master 2
Reset	BOOL	Control signal to reset the logical device from the locked error state (S_ERROR) to the initialisation state (S_INIT).	FALSE ⇨ TRUE	Control signal is transmitted
			other	No changes

Output parameters

18255

Parameter	Data type	Meaning	Possible values	
Ready	BOOL	Status of the FB processing (validity of the output signals)	FALSE	FB not completely processed or deactivated; signals are invalid
			TRUE	FB completely processed; signals are valid
Error	BOOL	Error indication	FALSE	None of the logical devices is in the locked error state S_ERROR
			TRUE	At least one of the logical devices is in the locked error state S_ERROR

10.3.4 CtrlLocalInputs

18260

Function block type: Function block (FB)

Library: sf_io.library

Symbol in CODESYS:

Description

18268

Function block (FB) to control the logical device for two input channels of the local I/O interface (IN_Channel_A, IN_Channel_B)

The programmer can carry out the following actions via CtrlLocalInputs:

- Unlock the logical device and reset it from the locked error state S_ERROR to the initialisation state S_INIT (RESET)

The FB provides the following state and diagnostic information:

- Logical values of the two input channels (Chan_A, Chan_B)
- State of the FB processing (ready)
- Error state of the function block (Error)
- Diagnostic information of the logical device (DiagCode)

The FB can be instanced as often as required within the limits of CODESYS.

Input parameters

18267

Parameter	Data type	Meaning	Possible values	
Enable	BOOL	FB activation	FALSE	FB is deactivated.
			TRUE	FB is activated.
IN_Channel_A	INT	Local input to which sensor channel A is connected.	1 ... 8	Local input IN1 ... local input IN8
IN_Channel_B	INT	Local input to which sensor channel B is connected.	1 ... 8	Local input IN1 ... local input IN8
Reset	BOOL	Control signal to reset the logical device from the locked error state (S_ERROR) to the initialisation state (S_INIT).	FALSE ⇨ TRUE	Control signal is transmitted
			other	No changes

Output parameters

18269

Parameter	Data type	Meaning	Possible values	
Chan_A	BOOL	Logic state of signal channel A	FALSE =	OFF
			TRUE =	ON
Chan_B	BOOL	Logical state of the safe signal channel B	FALSE =	OFF
			TRUE =	ON
Ready	BOOL	Status of the FB processing (validity of the output signals)	FALSE	FB not completely processed or deactivated; signals are invalid
			TRUE	FB completely processed; signals are valid
Error	BOOL	Error indication	FALSE	No error
			TRUE	Logical device is in the locked error state S_ERROR
DiagCode	WORD	Diagnostic code of the internal states of the controlled logical device	→ **Error codes: Local I/O interface** (→ p. 278) OR: → List below (Diagnostic codes)	

Diagnostic codes:

- 0x0000 Initialisation

10.3.5 Ctrl_SetDiagInfo

18265

Function block type: Function block (FB)

Library: sf_io.library

Symbol in CODESYS:

Description

18264

Function block (FB) for processing the diagnostic information of a PLCopen function block.

The programmer can carry out the following actions via Ctrl_SetDiagInfo:

- Read diagnostic code (code) of a function block of the safety PLCopen library and transmit it to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.
- The created error message is structured as follows:
  Error PLCopen-FB (FB_Type), instance no. (FB_Number): DC=%s1, DC-1=%s2
  - FB_Type = ID of the function block
  - FB_Number = instance number of the function block
  - DC = current diagnostic code (as hexadecimal value)
  - DC-1 = last diagnostic code (as hexadecimal value)

The FB provides the following state and diagnostic information:

- State of the FB processing (ready)
- Display of an error during FB processing (Error)

The FB generates a separate message for each change of state in the online support centre (OSC). The ring memory of the OSC therefore reaches its capacity limits within a short time.

► Use FB Ctrl_SetDiagInfo only for debugging.

Input parameter

18263

Parameter	Data type	Meaning	Possible values	
Enable	BOOL	FB activation	FALSE	FB is deactivated.
			TRUE	FB is activated.
FB_Type	INT	ID of the function block (is assigned by the programmer)	0x3D6B	SF_Antivalent
			0x5096	SF_EDM
			0x50A9	SF_EmergencyStop
			0x5151	SF_EnableSwitch
			0x516E	SF_Equivalent
			0x56DD	SF_ESPE
			0x56E2	SF_GuardLocking
			0x571A	SF_GuardMonitoring
			0x5725	SF_ModeSelector
			0x5A03	SF_MutingPar
			0x5A3C	SF_MutingPar_2Sensor
			0x5BC4	SF_MutingSeq
			0x5BFB	SF_OutControl
			0x5C48	SF_SafetyRequest
			0x5C77	SF_TestableSafetySensor
			0x5D8F	SF_TwoHandControlTypeII
			0x5DB0	SF_TwoHandControlTypeIII
FB_Number	INT	Instance number of the function block (is assigned by the programmer)		
DiagCode	INT	Diagnostic code of the safety PLCopen function block	→ Operating instructions safety PLCopen library in CODESYS	

The diagnostic code of the safety PLCopen FBs are of data type WORD.

Output parameters

266

Parameter	Data type	Meaning	Possible values	
Ready	BOOL	Status of the transmission of the diagnostic code	FALSE	At present no transmission request
			TRUE	Transmission of the diagnostic code successfully completed
Error	BOOL	Error indication	FALSE	No error occurred during FB processing
			TRUE	Error occurred during FB processing; transmission of the diagnostic code failed

10.3.6 GetLocalInput

18335

Function block type: Function block (FB)

Library: sf_io.library

Symbol in CODESYS:

Description

18336

Function block (FB) for access to a non-safe input channel of the local I/O interface (IN_Channel)
The programmer can carry out the following actions via GetLocalInput:

- Detect the non-safe, logical state of the selected input channel and provide the value (IN).

The FB provides the following state and diagnostic information:

- State of the FB processing (ready)
- Error state of the function block (Error)

Input parameters

18337

Parameter	Data type	Meaning	Possible values	
Enable	BOOL	FB activation	FALSE	FB is deactivated.
			TRUE	FB is activated.
IN_Channel	INT	Local input channel to which the sensor is connected.	1 ... 8	Local input IN1 ... Local input IN8

Output parameters

18338

Parameter	Data type	Meaning	Possible values	
IN	BOOL	Logic state of the selected input channel	FALSE	OFF
			TRUE	ON
Ready	BOOL	Status of the FB processing (validity of the output signals)	FALSE	FB not completely processed or deactivated; signals are invalid
			TRUE	FB completely processed; signals are valid
Error	BOOL	Error indication	FALSE	No error
			TRUE	Error occurred during FB execution

10.3.7 SetLocalOutput

18341

Function block type: Function block (FB)

Library: sf_io.library

Symbol in CODESYS:

Description

18342

Function block (FB) for access to a non-safe output channel of the local I/O interface (OUT_Channel)
The programmer can carry out the following actions via SetLocalOutput:

- Provide a non-safe Boolean value (OUT) at the selected output channel

The FB provides the following state and diagnostic information:

- State of the FB processing (ready)
- Error state of the function block (Error)

Input parameters

18343

Parameter	Data type	Meaning	Possible values	
Enable	BOOL	FB activation	FALSE	FB is deactivated.
			TRUE	FB is activated.
OUT_Channel	INT	Local output to which the actuator is connected.	1 ... 4	Local output OUT1 ... local output OUT4
OUT	BOOL	Required logical state of the local output channel	FALSE	OFF
			TRUE	ON

The outputs of the local I/O interface can only implement a maximum switching frequency of 25 Hz.

► When digital signals are generated at the local outputs, observe the minimum switch-on or switch-off time of 20 ms.

Output parameters

18344

Parameter	Data type	Meaning	Possible values	
Ready	BOOL	Status of the FB processing (validity of the output signals)	FALSE	FB not completely processed or deactivated; signals are invalid
			TRUE	FB completely processed; signals are valid
Error	BOOL	Error indication	FALSE	No error
			TRUE	Error occurred during FB execution

10.4 Library SF_LogicalInterfaces.library

Contents

Remarks about logical devices....202
Logical devices for safe AS-i slaves....205
Logical devices for the local I/O interface....235

7269

The library SF_LogicalInterfaces.library provides logical devices.

10.4.1 Remarks about logical devices

Contents

State machines for logical preprocessing....202
States....202
State transitions....202
Diagnostic messages....204

18235

State machines for logical preprocessing

7036

The safe input signals are preprocessed by means of state machines. Each logical device has its own state machine which creates a safe process signal from the input signals and the set parameters.

A state machine consists of the following components:

- States (→ **States** (→ p. 202))
- State transitions (→ **State transitions** (→ p. 202))

States

14281

Number and type of state result from the scope of the logical evaluation function. The exact description of the states is given in the documentation of the diagnostic messages of the individual logical devices.

State transitions

18233

The state machine checks in each PLC cycle which conditions for a state transition are fulfilled.

State transitions can be triggered by the following conditions:

- Signal states of the input signals (→ **States of the input signals** (→ p. 203))
- System events (→ **System events** (→ p. 204))
- Timer events (→ **Timer events** (→ p. 204))

States of the input signals

18945

Depending on the type of logical device the following definitions apply:

- **AS-i slaves**
  Safe AS-i input slaves transmit their input data as unambiguous code sequences. A code sequence consists of 2 half-sequences with each half sequence representing a signal channel.
- **Local I/O interface**
  At the local I/O interface each input represents a signal channel.

The combination of the two signal channels results in the following states and the respective state events:

State	State / Description	Transition event
NoChan	The signal channels A and B are in the logical state FALSE	E_NoChan
ChanA_only	Signal channel A is in the logical state TRUE. Signal channel B is in the logical state FALSE.	E_ChanA_Only
ChanB_only	Signal channel A is in the logical state FALSE. Signal channel B is in the logical state TRUE.	E_ChanB_only
ChanAB	The signal channels A and B are in the logical state TRUE.	E_ChanAB
ChanInvalid*	The logical value of the signal channels is invalid (e.g. code sequence is incomplete, code sequence invalid)	E_ChanInvalid

* ... Only available for logical devices for AS-i input slaves

Sequence of the signal changes

10244

The combination of the input signals "ChanA_only" and "ChanB_only" and the state of the parameter "P_Chan_B_first" allows to define state transitions which consider the sequence of the signal change at the input channels:

State / Description	Transition event
ChanA = TRUE UND ChanB = FALSE UND Chan_B_first = TRUE	E_ChanA; P_ChanB_first = TRUE
ChanA = TRUE UND ChanB = FALSE UND Chan_B_first = FALSE	E_ChanA; P_ChanB_first = FALSE
ChanB = TRUE UND ChanA = FALSE UND Chan_B_first = TRUE	E_ChanB; P_ChanB_first = TRUE
ChanB = TRUE UND ChanA = FALSE UND Chan_B_first = FALSE	E_ChanB; P_ChanB_first = FALSE

Start-up test

12226

If with activated start-up test in the start state both input signals "ChanA" = TRUE and "ChanB" = TRUE, the state machine first goes through a state cycle until both input signals simultaneously take the value FALSE.

State / Description	Transition event
ChanA = TRUE UND ChanB = TRUE UND StartupTest = FALSE	E_ChanAB_without Startup Test
ChanA = TRUE UND ChanB = TRUE UND StartupTest = TRUE	E_ChanAB_with Startup Test

System events

18946

Following system events start processing of the state machine of the logical device:

- Start event after successful initialisation of the state machine (E_Start)

Following system events change the state machine of the logical device to the initialisation state (S_INIT):

- A reset signal generated in the safe application (E_ErrorReset)
- AS-i input slaves: Communication and configuration errors (→ **Error codes: Safe AS-i slaves** (→ p. 276))
- Local inputs/outputs: Configuration and system errors (→ **Error codes: Local I/O interface** (→ p. 278))

Timer events

18236

Timers are started when a state is exited. When the timer has elapsed, a respective event is created. This timer event is used by the event machine as condition for the transition to the next state. If one or several timer events occur simultaneously, the event with the highest priority is processed.

The logical devices can evaluate the following times when the safe process signal is generated:

Designation	Description	Timer event	Priority
InvalidTime	▪ Time span within which the signal channels may have an invalid state without the device changing into the locked error state. ▪ Firmly set	E_InvalTimeout	1
SynchTime	▪ Time span within which the signal channels have to take the same logical state. ▪ Programmable by programmer	E_SynchTimeout	2
StabilisingTime	▪ Time span within which the states of the signal channels have to be stable to trigger a state change in the evaluation logic. ▪ Programmable by programmer	E_StabTimeout	3
ToleranceTime	▪ Time span within which the state FALSE is tolerated at one of the two signal channels without switching-off or changing to the test state. ▪ Programmable by programmer	E_TolTimeout	4

Diagnostic messages

422

By means of the parameter "DisableLogging" the programmer can control the routing of state messages of the logical devices to the online support centre (OSC). If the parameter is deactivated, all state changes and error events are transmitted to the OSC. If the parameter is activated, only error events are transmitted to the OSC.

Message type	States	Description
Error message	S_Test_e	State for going through the start-up test
	S_Error_e	Locked error state
	S_Init_e	Initialisation state
Status message	S_OFF_e	Safe OFF
	S_ON_e	Safe OFF

10.4.2 Logical devices for safe AS-i slaves

Contents

SF_IN_ASi_forced 206
SF_IN_ASi_independent 210
SF_IN_ASi_conditionally_dependent 214
SF_IN_ASi_dependent 218
SF_IN_ASi_dependent_filter_w_testreq 222
SF_IN_ASi_dependent_filter_ntestreq 226
SF_IN_ASi_dependent_filter_nshutdown 230
SF_OUTcontrol_ASi 234

19796

SF_IN_ASi_forced

18194

Device type:	AS-i slave (input)
Library:	sf_logicalinterfaces.library
Symbol in CODESYS:	 SF_IN_ASi_forced

Description

18201

Logical device for the evaluation of a safe AS-i input slave with 2 positively-driven contacts/channels

SF_IN_ASi_forced detects switching states of the two signal channels of the AS-i slaves (ASi_SlaveAdr, ASi_Master). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- Simultaneity of the signal change of the two contacts/channels is monitored during switch-on and switch-off. If both signal channels do not take the same switching status during the fixed time, the logical device changes to the locked error state. (InvalidTime = const. = 100ms)

The programmer can activate the following functions as option:

- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.
- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.

To reset the logical device from the locked error state: → **CtrlASi_InSlave** (→ p. 187)

Parameter data

18195

Parameter	Data type	Meaning	Possible values	
ASi_SlaveAdr	INT	Address of the AS-i slave	1 ... 31	Slave address 1 ... slave address 31
ASi_Master	INT	AS-i master to which the AS-i slave is connected	1	AS-i master 1
			2	AS-i master 2
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON

* ... Default value

Process data

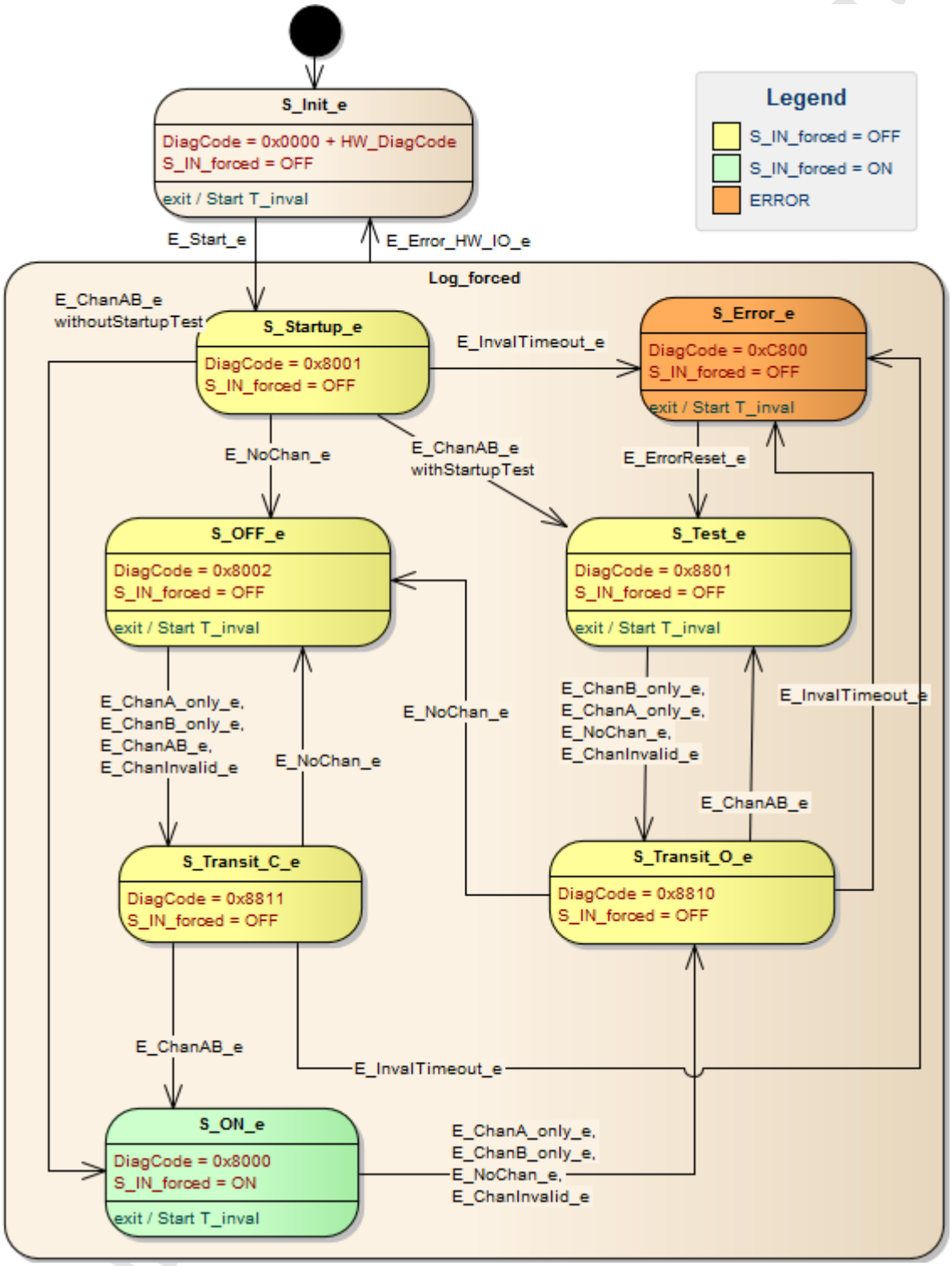
18200

Parameter	Data type	Meaning	Possible values	
S_IN_forced	SAFEBOOL	Safe process signal	FALSE	Safe OFF
			TRUE	Safe ON

State diagram (state machine)

18933

The state diagram shows the logical signal evaluation of the logical device:



Diagnostic information

19760

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187).

Following diagnostic codes exist:

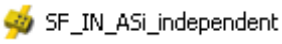
DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0xC800	S_Error	Locked error state	OFF

SF_IN_ASi_independent

18212

Device type: AS-i slave (input)

Library: sf_logicalinterfaces.library

Symbol in CODESYS: 

Description

18211

Logical device for a safe sensor with 2 contacts/channels acting independently of each other

SF_IN_ASi_independent detects switching states of the two signal channels/contacts of a safe AS-i slave (ASi_SlaveAdr, ASi_Master). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- The simultaneity of the signal change is not monitored.
- If the signal state of the two channels is invalid longer than the firmly set time, both signal channels have to take the state FALSE simultaneously before switch-on is possible again. (InvalidTime = const. = 2000ms)

The programmer can activate the following functions as option:

- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.
- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.

To reset the logical device from the locked error state and to read diagnostic information of the logical device: → **CtrlASi_InSlave** (→ p. 187)

Parameter data

18214

Parameter	Data type	Meaning	Possible values	
ASi_SlaveAdr	INT	Address of the AS-i slave	1 ... 31	Slave address 1 ... slave address 31
ASi_Master	INT	AS-i master to which the AS-i slave is connected	1	AS-i master 1
			2	AS-i master 2
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON

* ... Default value

Process data

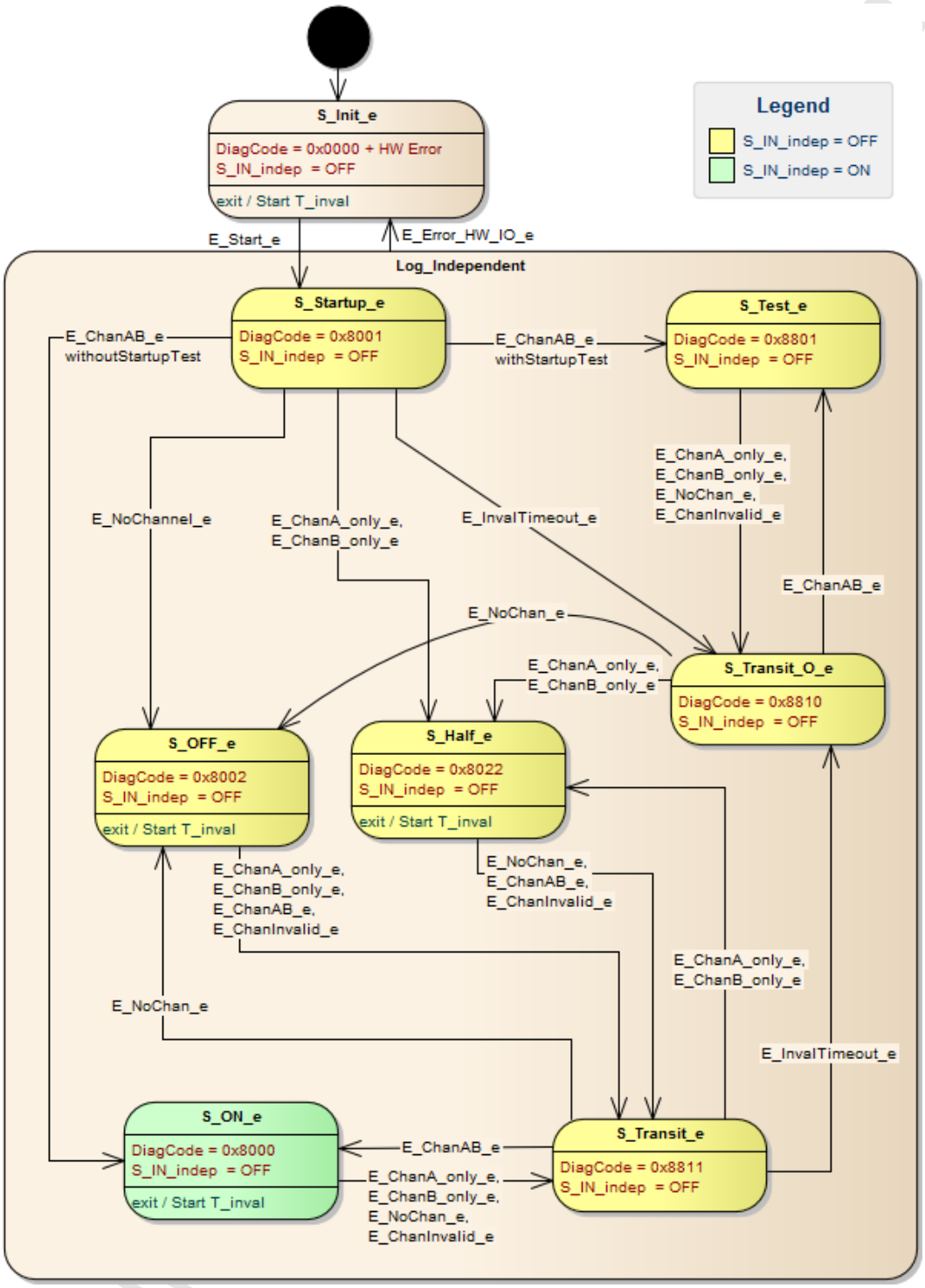
18213

Parameter	Data type	Meaning	Possible values	
S_IN_indep	SAFEBOOL	Safe process signal	FALSE	safety function OFF
			TRUE	safety function ON

State diagram (state machine)

18932

The state diagram shows the logical signal evaluation of the logical device:



Diagnostic information

19769

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187).

Following diagnostic codes exist:

DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0x8822	S_Half	1 channel = TRUE	OFF

SF_IN_ASi_conditionally_dependent

18207

Device type: AS-i slave (input)

Library: sf_logicalinterfaces.library

Symbol in CODESYS:  SF_IN_ASi_conditionally_dependent

Description

18208

Logical device for a safe sensor with 2 conditionally interdependently acting contacts/channels

SF_IN_ASi_conditionally_dependent detects the switching states of the two signal channels of the AS-i slave (ASi_SlaveAdr, ASi_Master). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- The sequence of the signal changes of the two contacts/channels is monitored during switch-on and switch-off. If the sequence of the signal change deviates from the set sequence, the logical device changes to the locked error state. (Chan_B_first)
- If the signal state of the two signal channels is invalid longer than the firmly set time, the logical device changes to the locked error state. (InvalidTime = const. = 140ms)

The programmer can activate the following functions as option:

- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.
- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.

To reset the logical device from the locked error state and to read diagnostic information of the logical device: → **CtrlASi_InSlave** (→ p. 187)

Parameter data

18209

Parameter	Data type	Meaning	Possible values	
ASi_SlaveAdr	INT	Address of the AS-i slave	1 ... 31	Slave address 1 ... slave address 31
ASi_Master	INT	AS-i master to which the AS-i slave is connected	1	AS-i master 1
			2	AS-i master 2
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON
Chan_B_first	BOOL	Required sequence of the signal state changes	FALSE	Channel A = TRUE before channel B = TRUE
			TRUE	Channel B = TRUE before channel B = TRUE

* ... Default value

Process data

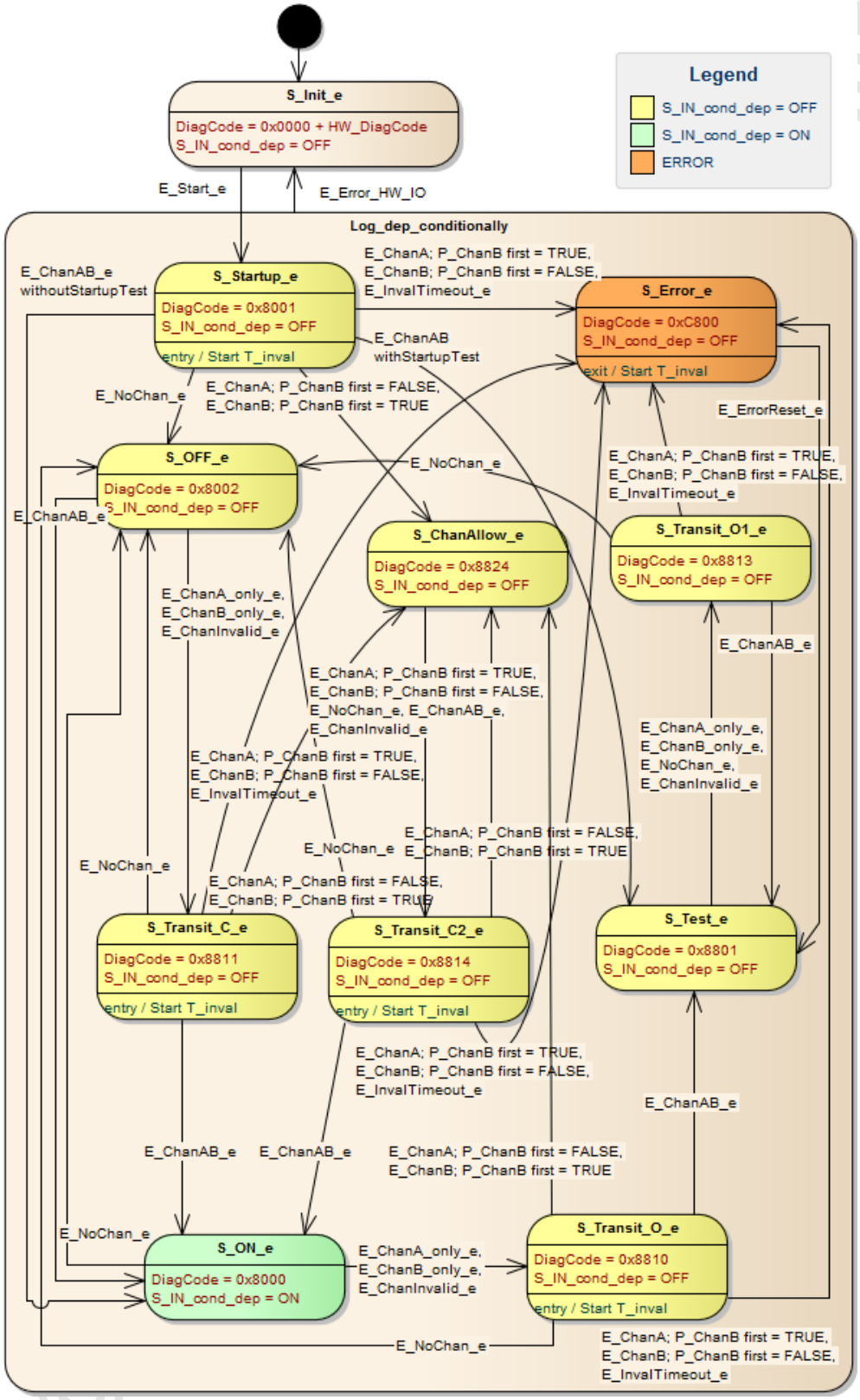
18210

Parameter	Data type	Meaning	Possible values	
S_IN_cond_dep	SAFEBOOL	Safe process signal	FALSE	Safe OFF
			TRUE	Safe ON

State diagram (state machine)

594

The state diagram shows the logical signal evaluation of the logical device:



Diagnostic information

19776

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187).

Following diagnostic codes exist:

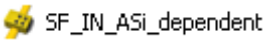
DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0x8813	S_Transit	Transition state	OFF
0x8814	S_Transit	Transition state	OFF
0x8824	S_ChanAllow	1 channel = TRUE, signal sequence OK	OFF
0xC800	S_Error	Locked error state	OFF

SF_IN_ASi_dependent

18202

Device type: AS-i slave (input)

Library: sf_logicalinterfaces.library

Symbol in CODESYS: 

Description

18203

Logical device for a safe sensor with 2 interdependently acting contacts/channels

SF_IN_ASi_dependent detects the switching states of the two signal channels/contacts of the AS-i slave (ASi_SlaveAdr, ASi_Master). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- Simultaneity of the signal change of the two contacts/channels is monitored during switch-on: If the signal states are not equal for a longer time than the set synchronisation time, both contacts/channels have to take the value FALSE simultaneously before switch-on is possible again. (SynchTime)
- If the signal state of the two signal channels is invalid longer than the firmly set time, the logical device changes to the locked error state. (InvalidTime = const. = 140ms)

The programmer can activate the following functions as option:

- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.
- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.

To reset the logical device from the locked error state and to read diagnostic information of the logical device: → **CtrlASi_InSlave** (→ p. 187)

Parameter data

18204

Parameter	Data type	Meaning	Possible values	
ASi_SlaveAdr	INT	Address of the AS-i slave	1 ... 31	Slave address 1 ... slave address 31
ASi_Master	INT	AS-i master to which the AS-i slave is connected	1	AS-i master 1
			2	AS-i master 2
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON
SynchTime	WORD	Synchronisation time	0 250 ... 500 ... 60000	Infinite 250 milliseconds ... 500 milliseconds * ... 60000 milliseconds

* ... Default value

Process data

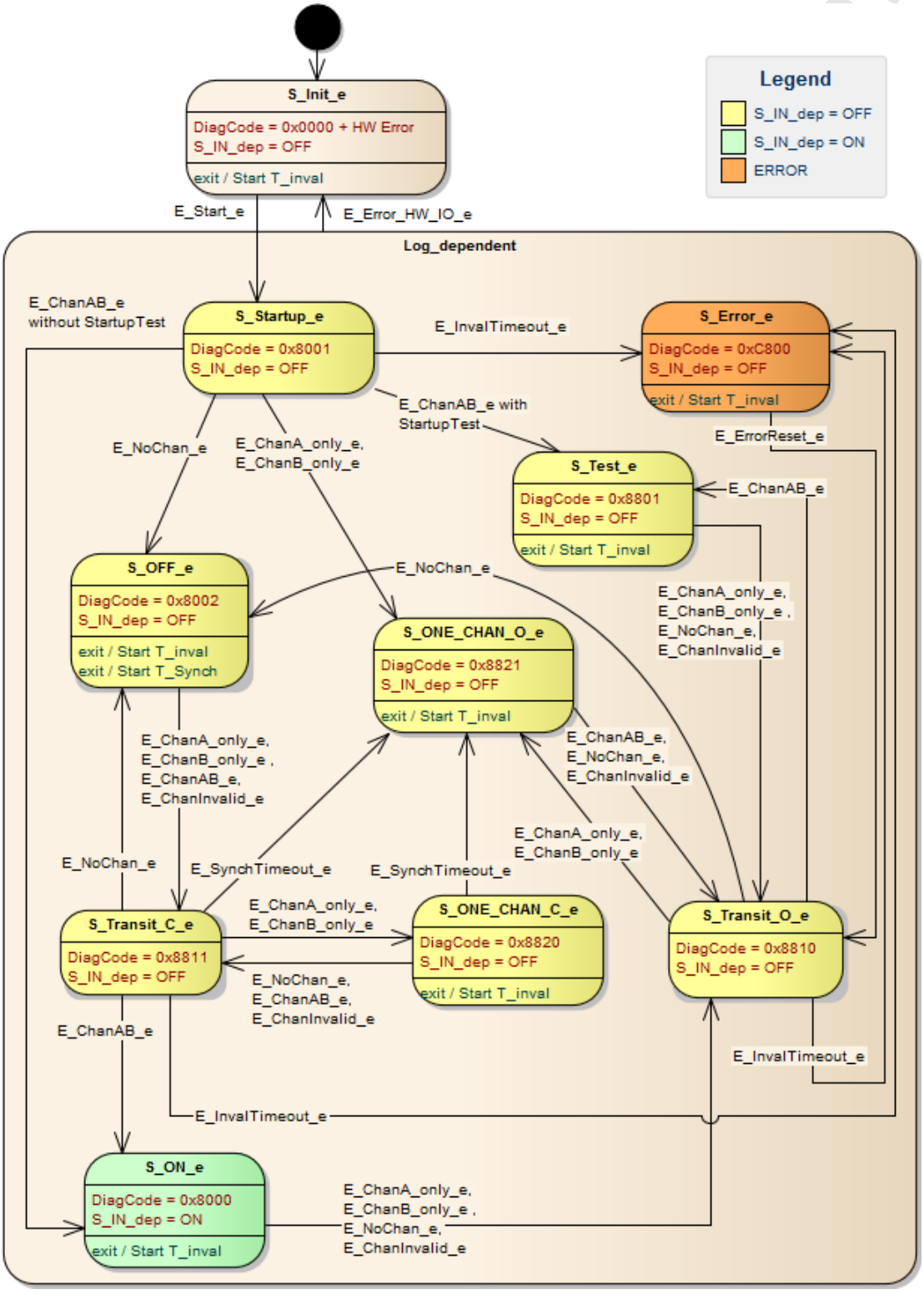
18205

Parameter	Data type	Meaning	Possible values	
S_IN_dependent	SAFEBOOL	Safe process signal	FALSE	Safe OFF
			TRUE	Safe ON

State diagram (state machine)

591

The state diagram shows the logical signal evaluation of the logical device:



Diagnostic information

19773

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187).

Following diagnostic codes exist:

DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0x8820	S_ONE_CHAN_C	1 channel = TRUE, test of the synchronisation time	OFF
0x8821	S_ONE_CHAN_O	1 channel = TRUE, synchronisation time elapsed	OFF
0xC800	S_Error	Locked error state	OFF

SF_IN_ASi_dependent_filter_w_testreq

18215

Device type: AS-i slave (input)

Library: sf_logicalinterfaces.library

Symbol in CODESYS:  SF_IN_ASi_dependent_filter_w_testreq

Description

18216

Logical device for a safe sensor with 2 interdependently acting contacts/channels

SF_IN_dependent_filter_w_testreq detects the switching states of the two contacts/channels of the AS-i slave (ASi_SlaveAdr, ASi_Master). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- The contacts/channels are debounced during switch-on: Signal changes that are applied shorter than the set stabilisation time do not lead to a change in state. (StabilisingTime).
- Simultaneity of the signal change of the two contacts/channels is monitored during switch-on: If the signal states are not equal for a longer time than the set synchronisation time, both contacts/channels have to take the value FALSE simultaneously before switch-on is possible again. (SynchTime)
- Before the two contacts/channels are switched on again, they have to be FALSE simultaneously.
- If after the start the two contacts/channels do not take a valid signal state within a firmly set time, the logical device changes to the locked error state. (InvalidTime = const. = 140 ms)
- If during switching both signals are longer invalid than the set stabilisation time, the logical device changes to the locked error state. (StabilisingTime)

The programmer can activate the following functions as option:

- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.
- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.

To reset the logical device from the locked error state and to read diagnostic information of the logical device: → **CtrlASi_InSlave** (→ p. 187)

Parameter data

18217

Parameter	Data type	Meaning	Possible values	
ASi_SlaveAdr	INT	Address of the AS-i slave	1 ... 31	Slave address 1 ... slave address 31
ASi_Master	INT	AS-i master to which the AS-i slave is connected	1	AS-i master 1
			2	AS-i master 2
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON
SynchTime	WORD	Synchronisation time	0 250 ... 500 ... 60000	Infinite 250 milliseconds ... 500 milliseconds * ... 60000 milliseconds
StabilisingTime	WORD	Max. bounce time	250 ... 10000	250 milliseconds * ... 10000 milliseconds

* ... Default value

Process data

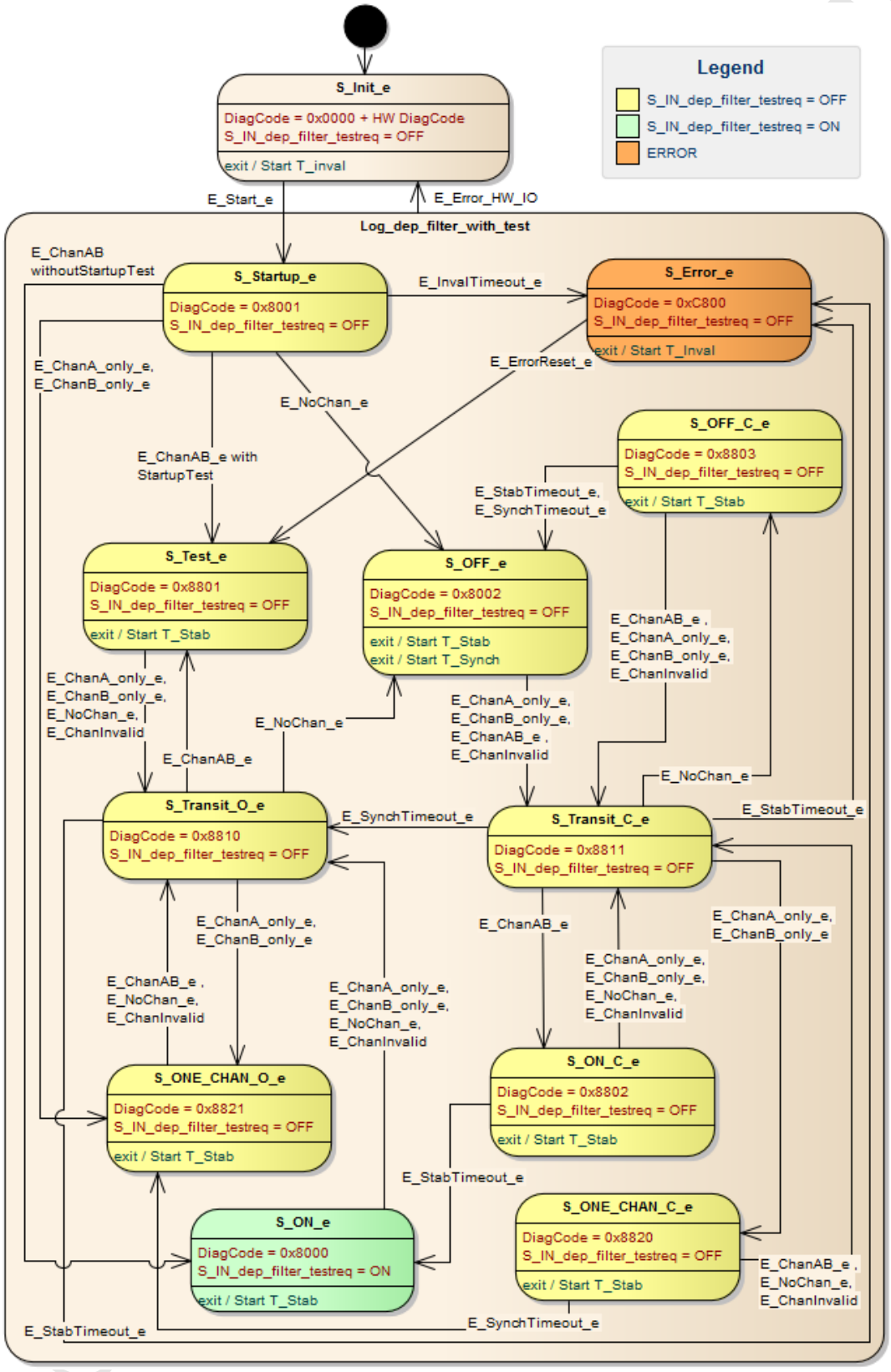
18218

Parameter	Data type	Meaning	Possible values	
S_IN_dep_filter_testreq	SAFEBOOL	Safe process signal	FALSE	Safety function OFF
			TRUE	Safety function ON

State diagram (state machine)

595

The state diagram shows the logical signal evaluation of the logical device:



Diagnostic information

19780

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187)

Following diagnostic codes exist:

DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8802	S_ON_C	Both channels = TRUE, check the set timer	OFF
0x8803	S_OFF_C	Both channels = FALSE, check the set timer	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0x8820	S_ONE_CHAN_C	1 channel = TRUE, test of the synchronisation time	OFF
0x8821	S_ONE_CHAN_O	1 channel = TRUE, synchronisation time elapsed	OFF
0xC800	S_Error	Locked error state	OFF

SF_IN_ASi_dependent_filter_ntestreq

18222

Device type: AS-i slave (input)

Library: sf_logicalinterfaces.library

Symbol in CODESYS:  SF_IN_ASi_dependent_filter_ntestreq

Description

18221

Logical device for a safe sensor with 2 interdependently acting contacts/channels

SF_IN_dependent_filter_ntestreq detects the switching states of the two signals/channels of the AS-i slave (ASi_SlaveAdr, ASi_Master). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- The contacts/channels are debounced during switch-on: Signal changes that are applied shorter than the set stabilisation time do not lead to a change in state. (StabilisingTime)
- Simultaneity of the signal change of the two contacts/channels is monitored during switch-on: If the signal states are not equal for a longer time than the set synchronisation time, both contacts/channels have to take the value FALSE simultaneously before switch-on is possible again. (SynchTime)
- If both contacts/channels take the value TRUE again during the switch-off process, switch-on occurs at once. When the tolerance time has elapsed, the two contacts/channels have to take the value FALSE again simultaneously before switch-on is possible again. (ToleranceTime)
- If after the start the two contacts/channels do not take a valid signal state within a firmly set time, the logical device changes to the locked error state. (InvalidTime = const. = 140 ms)
- If during switching both signals are longer invalid than the set stabilisation time, the logical device changes to the locked error state. (StabilisingTime)

The programmer can activate the following functions as option:

- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.
- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.

To reset the logical device from the locked error state and to read diagnostic information of the logical device: → **CtrlASi_InSlave** (→ p. 187)

Parameter data

18220

Parameter	Data type	Meaning	Possible values	
ASi_SlaveAdr	INT	Address of the AS-i slave	1 ... 31	Slave address 1 ... slave address 31
ASi_Master	INT	AS-i master to which the AS-i slave is connected	1	AS-i master 1
			2	AS-i master 2
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON
SynchTime	WORD	Synchronisation time	0 250 ... 500 ... 60000	Infinite 250 milliseconds ... 500 milliseconds * ... 60000 milliseconds
StabilisingTime	WORD	Max. bounce time	250 ... 10000	250 milliseconds * ... 10000 milliseconds
ToleranceTime	WORD	Tolerance time for switch-off	250 ... 1000	250 milliseconds * ... 1000 milliseconds

* ... Default value

Process data

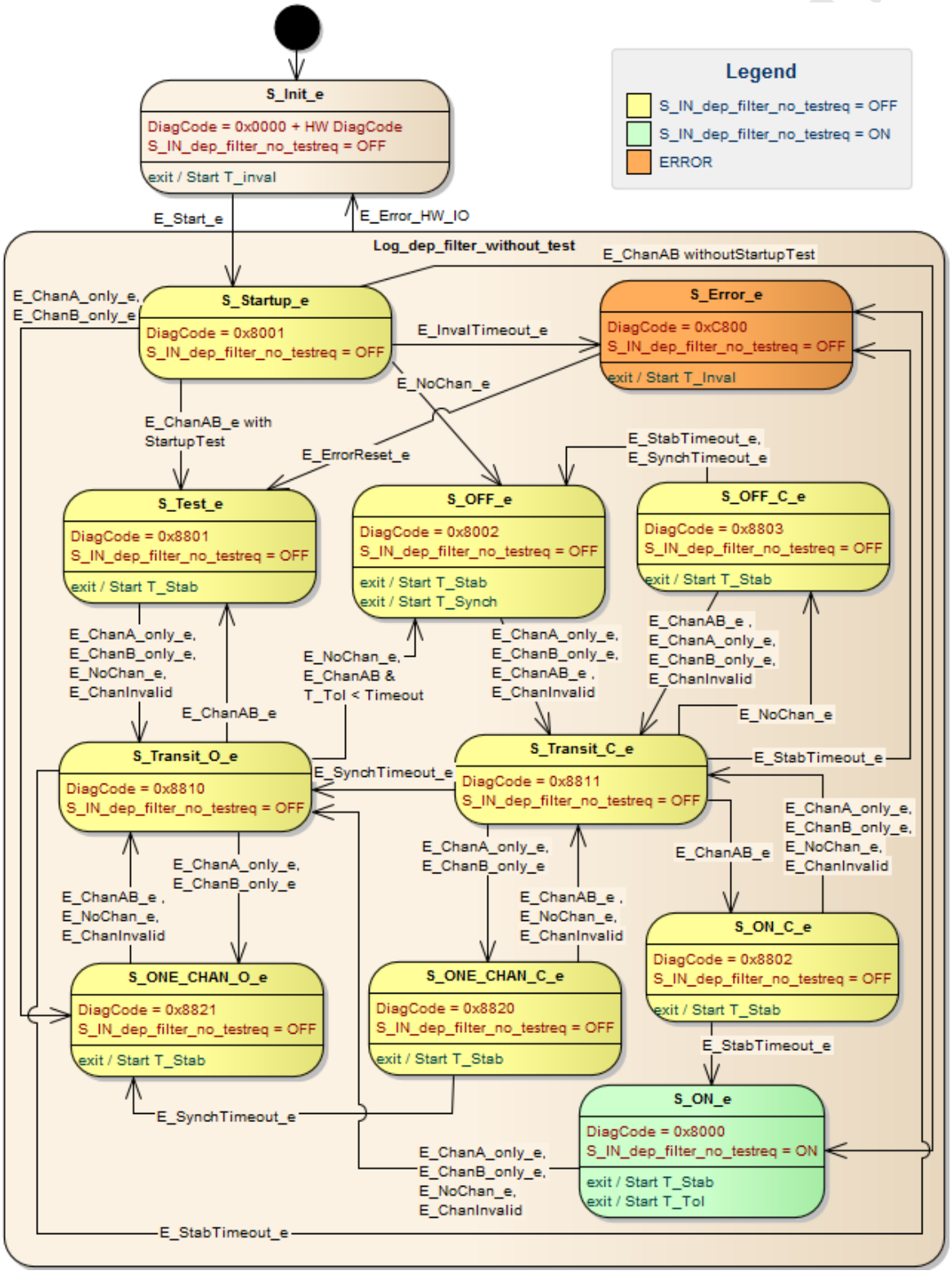
18223

Parameter	Data type	Meaning	Possible values	
S_IN_dep_filter_ntest	SAFEBOOL	Safe process signal	FALSE	safety function OFF
			TRUE	safety function ON

State diagram (state machine)

596

The state diagram shows the logical signal evaluation of the logical device:



Diagnostic information

19783

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187).

Following diagnostic codes exist:

DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8802	S_ON_C	Both channels = TRUE, check the set timer	OFF
0x8803	S_OFF_C	Both channels = FALSE, check the set timer	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0x8820	S_ONE_CHAN_C	1 channel = TRUE, test of the synchronisation time	OFF
0x8821	S_ONE_CHAN_O	1 channel = TRUE, synchronisation time elapsed	OFF
0xC800	S_Error	Locked error state	OFF

SF_IN_ASi_dependent_filter_nshutdown

18227

Device type: AS-i slave (input)

Library: sf_logicalinterfaces.library

Symbol in CODESYS:  SF_IN_ASi_dependent_filter_nshutdown

Description

18226

Logical device for a safe sensor with 2 interdependently acting contacts/channels

SF_IN_dependent_filter_nshutdown detects the switching states of the two contacts/channels of the AS-i slave (ASi_SlaveAdr, ASi_Master). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- The contacts/channels are debounced during switch-on: Signal changes that are applied shorter than the set stabilisation time do not lead to a change in state. (StabilisingTime)
- Simultaneity of the signal change of the two contacts/channels is monitored during switch-on: If the signal states are not equal for a longer time than the set synchronisation time, both contacts/channels have to take the value FALSE simultaneously before switch-on is possible again. (SynchTime)
- If both contacts/channels take the value TRUE again during the switch-off process, switch-on occurs at once. When the tolerance time has elapsed, the two contacts/channels have to take the value FALSE again simultaneously before switch-on is possible again. (ToleranceTime)
- If after the start the two contacts/channels do not take a valid signal state within a firmly set time, the logical device changes to the locked error state. (InvalidTime = const. = 140 ms)
- If during switching both signals are longer invalid than the set stabilisation time, the logical device changes to the locked error state. (StabilisingTime)

The programmer can activate the following functions as option:

- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.
- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.

To reset the logical device from the locked error state and to read diagnostic information of the logical device: → **CtrlASi_InSlave** (→ p. 187).

When this logical device is used, the process safety time (PST) of the system is extended by one cycle time of the fail-safe PLC.

► Note the remarks about process safety time (→ Device manual, **Process safety time**)!

► Observe the set cycle time of the fail-safe PLC (→ **Set cycle time of the fail-safe PLC** (→ p. 100))!

Parameter data

18225

Parameter	Data type	Meaning	Possible values	
ASi_SlaveAdr	INT	Address of the AS-i slave	1 ... 31	Slave address 1 ... slave address 31
ASi_Master	INT	AS-i master to which the AS-i slave is connected	1	AS-i master 1
			2	AS-i master 2
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON
SynchTime	WORD	Synchronisation time	0 250 ... 500 ... 60000	Infinite 250 milliseconds ... 500 milliseconds * ... 60000 milliseconds
StabilisingTime	WORD	Max. bounce time	250 ... 10000	250 milliseconds * ... 10000 milliseconds
ToleranceTime	WORD	Tolerance time for switch-off	250 ... 1000	250 milliseconds * ... 1000 milliseconds

* ... Default value

Process data

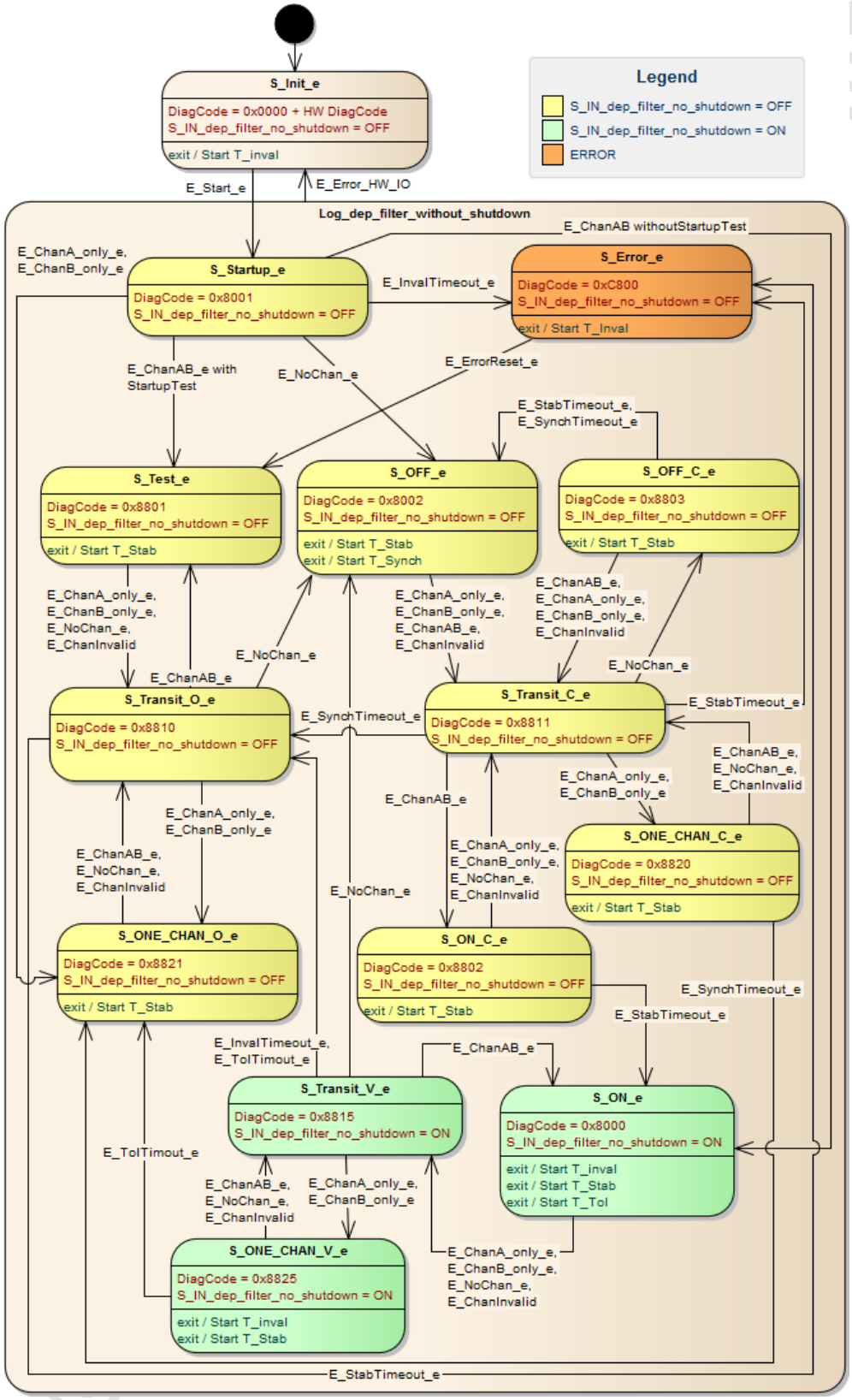
18228

Parameter	Data type	Meaning	Possible values	
S_IN_dep_nshutdown	SAFEBOOL	Safe process signal	FALSE	safety function OFF
			TRUE	safety function ON

State diagram (state machine)

18931

The state diagram shows the logical signal evaluation of the logical device:



Diagnostic information

19786

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187).

Following diagnostic codes exist:

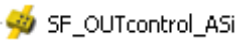
DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8802	S_ON_C	Both channels = TRUE, check the set timer	OFF
0x8803	S_OFF_C	Both channels = FALSE, check the set timer	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0x8815	S_Transit	Transition state	ON
0x8820	S_ONE_CHAN_C	1 channel = TRUE, test of the synchronisation time	OFF
0x8821	S_ONE_CHAN_O	1 channel = TRUE, synchronisation time elapsed	OFF
0x8825	S_ONE_CHAN_V	1 channel = TRUE; tolerance time not yet elapsed	ON
0xC800	S_Error	Locked error state	OFF

SF_OUTcontrol_ASi

18231

Device type: AS-i slave (output)

Library: sf_logicalinterfaces.library

Symbol in CODESYS: 

Description

18230

Logical device for a safe AS-i control slave (ASi_SlaveAdr, ASi_Master)

SF_OUTControl_ASi generates a safe output signal (S_OUT) from the configured process data value. The safe output signal controls the AS-i control slave that generates a safe code sequence with which one or several safe AS-i output slaves can be enabled.

The programmer can activate the following functions:

- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.

Parameter data

18229

Parameter	Data type	Meaning	Possible values	
ASi_SlaveAdr	INT	Address of the AS-i slave	1 ... 31	Slave address 1 ... slave address 31
ASi_Master	INT	AS-i master to which the AS-i slave is connected	1	AS-i master 1
			2	AS-i master 2
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)

* ... Default value

Process data

18232

Parameter	Data type	Meaning	Possible values	
S_OUT	SAFEBOOL	Release of the safe AS-i output slave	FALSE	safety function OFF
			TRUE	safety function ON

10.4.3 Logical devices for the local I/O interface

Contents

SF_IN_local_forced 236
SF_IN_local_independent 240
SF_IN_local_conditionally_dependent 245
SF_IN_local_dependent 252
SF_IN_local_dependent_filter_testreq 257
SF_IN_local_dependent_filter_ntestreq 263
SF_IN_local_dependent_filter_nshutdown 267
SF_OUT_local_single 273
SF_OUT_local_dual 274
SF_OUT_local_testpulse 275

19795

SF_IN_local_forced

18279

Interface type: Local I/O interface (input)

Library: sf_logicalinterfaces.library

Symbol in CODESYS: 

Description

18278

Logical device for safe sensors with 2 positively-driven contacts/channels

SF_IN_local_forced detects the two signals of a safe sensor at the local I/O interface (IN_Channel_A, IN_Channel_B). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- Simultaneity of the signal change of the two contacts/channels is monitored during switch-on and switch-off. If both signal channels do not take the same switching status during the fixed time, the logical device changes to the locked error state. (InvalidTime = const. = 100 ms)

The programmer can activate the following functions as option:

- EnableIN_Test:
  A test signal for cross-fault monitoring generated at a local output channel at the local I/O interface is evaluated.
- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.
- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.

If cross-fault detection is activated for 2-channel safe devices at the local I/O interface (EnableIN_Test = TRUE), the following applies:
Signal change at the inputs (IN_Channel_A, IN_Channel_B) must be applied longer than the cycle time of the test pulses (3 PLC cycles). Otherwise the logical device changes to the locked error state.

To reset the logical device from the locked error state and to read diagnostic information of the logical device: → **CtrlLocalInputs** (→ p. 194).

Parameter data

18277

Parameter	Data type	Meaning	Possible values	
IN_Channel_A	INT	Local input to which sensor channel A is connected.	1 ... 8	Local input IN1 ... local input IN8
IN_Channel_B	INT	Local input to which sensor channel B is connected.	1 ... 8	Local input IN1 ... local input IN8
EnableIN_Test	BOOL	Evaluation of a test signal which was created by a local output	FALSE	Test signal is ignored *
			TRUE	Test signal is evaluated
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON

* ... Default value

Process data

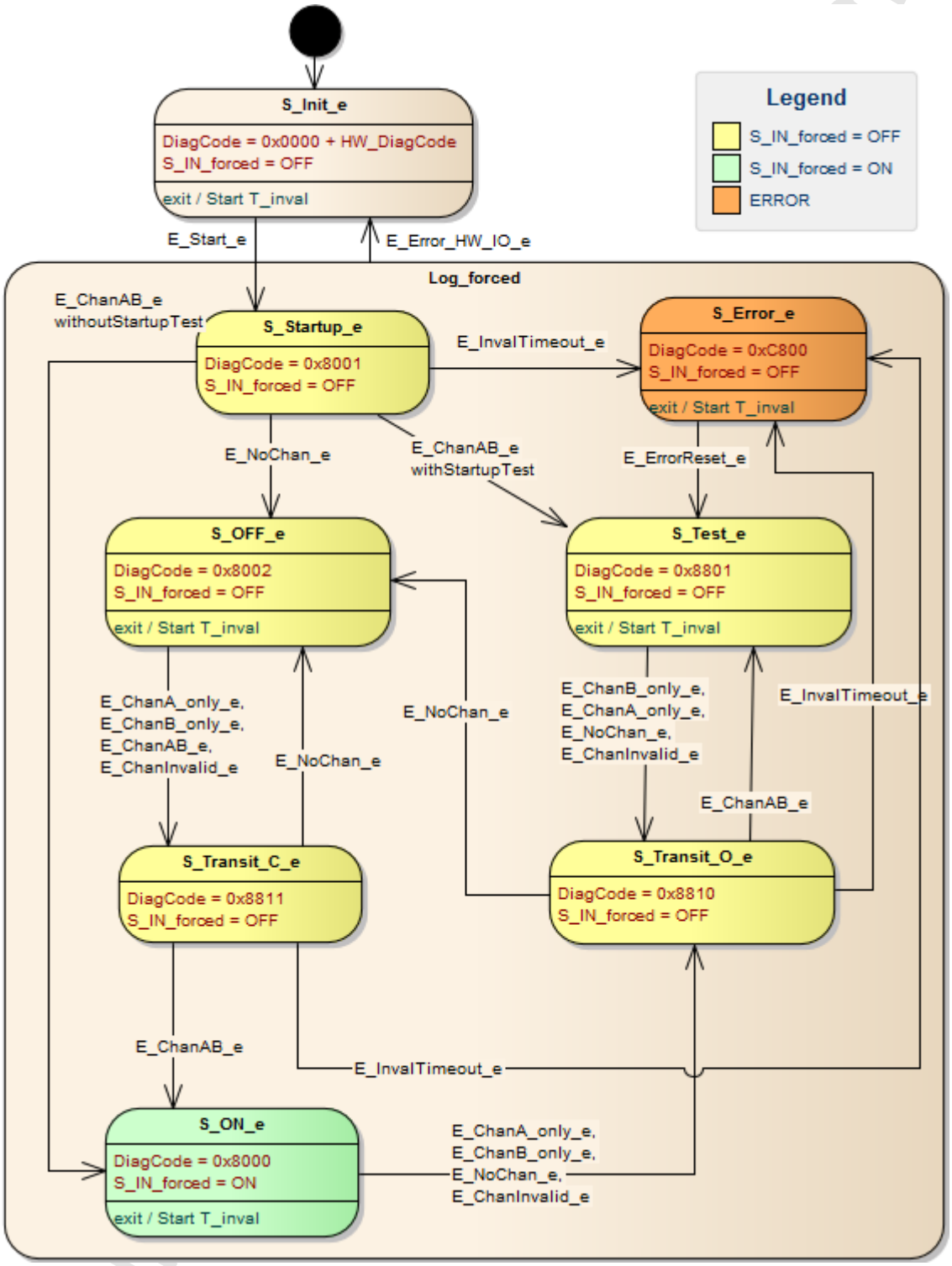
18276

Parameter	Data type	Meaning	Possible values	
S_IN_forced	SAFEBOOL	Safe process signal	FALSE	SAFE OFF
			TRUE	Safe ON

State diagram (state machine)

20685

The state diagram shows the logical signal evaluation of the logical device:



Diagnostic information

19760

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187).

Following diagnostic codes exist:

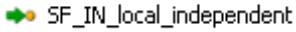
DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0xC800	S_Error	Locked error state	OFF

SF_IN_local_independent

18297

Interface type: Local I/O interface (input)

Library: sf_io.library

Symbol in CODESYS: 

Description

18298

Logical device for a safe sensor with 2 contacts/channels acting independently of each other

SF_IN_local_independent detects switching states of 2 input channels of the local I/O interface (IN_Channel_A, IN_Channel_B). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- The simultaneity of the signal change is not monitored.
- If the signal state of the two channels is invalid longer than the firmly set time, both signal channels have to take the state FALSE simultaneously before switch-on is possible again. (InvalidTime = const. = 140ms)

The programmer can activate the following functions as option:

- EnableIN_Test:
  A test signal for cross-fault monitoring generated at a local output channel at the local I/O interface is evaluated.
- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.
- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.

If cross-fault detection is activated for 2-channel safe devices at the local I/O interface (EnableIN_Test = TRUE), the following applies:
Signal change at the inputs (IN_Channel_A, IN_Channel_B) must be applied longer than the cycle time of the test pulses (3 PLC cycles). Otherwise the logical device changes to the locked error state.

To read the diagnostic information of the logical device: → **CtrlLocalInputs** (→ p. 194).

Parameter data

18299

Parameter	Data type	Meaning	Possible values	
IN_Channel_A	INT	Local input to which sensor channel A is connected.	1 ... 8	Local input IN1 ... local input IN8
IN_Channel_B	INT	Local input to which sensor channel B is connected.	1 ... 8	Local input IN1 ... local input IN8
EnableIN_Test	BOOL	Evaluation of a test signal which was created by a local output	FALSE	Test signal is ignored *
			TRUE	Test signal is evaluated
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON

* ... Default value

Process data

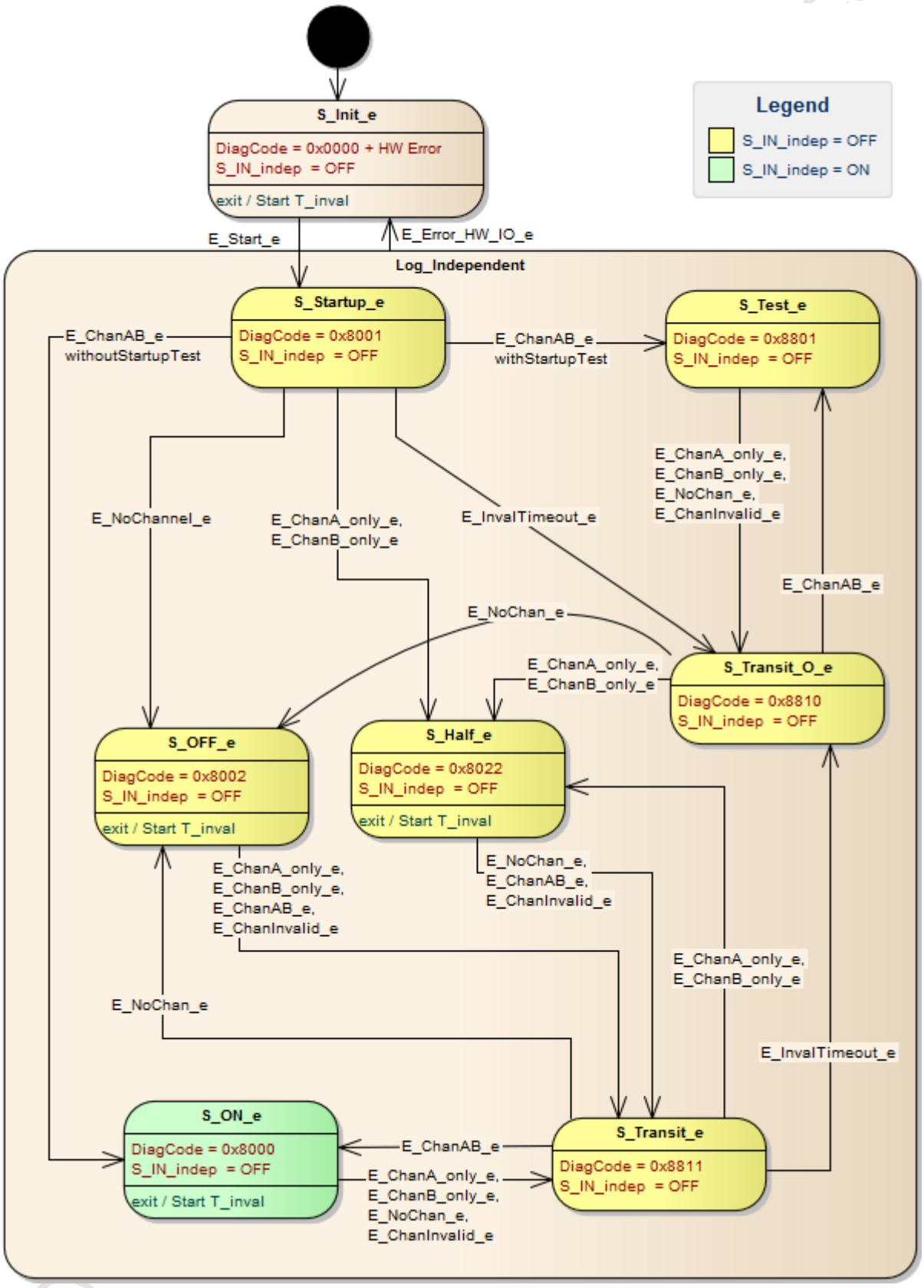
18300

Parameter	Data type	Meaning	Possible values	
S_IN_indep	SAFEBOOL	Status of the safety function	FALSE	Safe OFF or error
			TRUE	Safe ON and no error

State diagram (state machine)

20687

The state diagram shows the logical signal evaluation of the logical device:



Diagnostic information

19769

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187).

Following diagnostic codes exist:

DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0x8822	S_Half	1 channel = TRUE	OFF

SF_IN_local_conditionally_dependent

18287

Interface type: Local I/O interface (input)

Library: sf_logicalinterfaces.library

Symbol in CODESYS: 

Description

18290

Logical device for a safe sensor with 2 conditionally interdependently acting contacts/channels

SF_IN_local_conditionally_dependent detects the switching states of the two input channels of the local I/O interface (IN_Channel_A, IN_Channel_B). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- The sequence of the signal changes of the two contacts/channels is monitored during switch-on and switch-off. If the sequence of the signal change deviates from the set sequence, the logical device changes to the locked error state. (Chan_B_first)
- If both signal channels do not take the same switching status during the fixed time, the logical device changes to the locked error state. (InvalidTime = const. = 140ms)

The programmer can activate the following functions as option:

- EnableIN_Test:
  A test signal for cross-fault monitoring generated at a local output channel at the local I/O interface is evaluated.
- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.
- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.

If cross-fault detection is activated for 2-channel safe devices at the local I/O interface (EnableIN_Test = TRUE), the following applies:
Signal change at the inputs (IN_Channel_A, IN_Channel_B) must be applied longer than the cycle time of the test pulses (3 PLC cycles). Otherwise the logical device changes to the locked error state.

To reset the logical device from the locked error state and to read diagnostic information of the logical device: → **CtrlLocalInputs** (→ p. 194).

Parameter data

18288

Parameter	Data type	Meaning	Possible values	
IN_Channel_A	INT	Local input to which sensor channel A is connected.	1 ... 8	Local input IN1 ... local input IN8
IN_Channel_B	INT	Local input to which sensor channel B is connected.	1 ... 8	Local input IN1 ... local input IN8
EnableIN_Test	BOOL	Evaluation of a test signal which was created by a local output	FALSE	Test signal is ignored *
			TRUE	Test signal is evaluated
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON
Chan_B_first	BOOL	Required sequence of the signal state changes	FALSE	Channel A = TRUE before channel B = TRUE
			TRUE	Channel B = TRUE before channel B = TRUE

* ... Default value

Process data

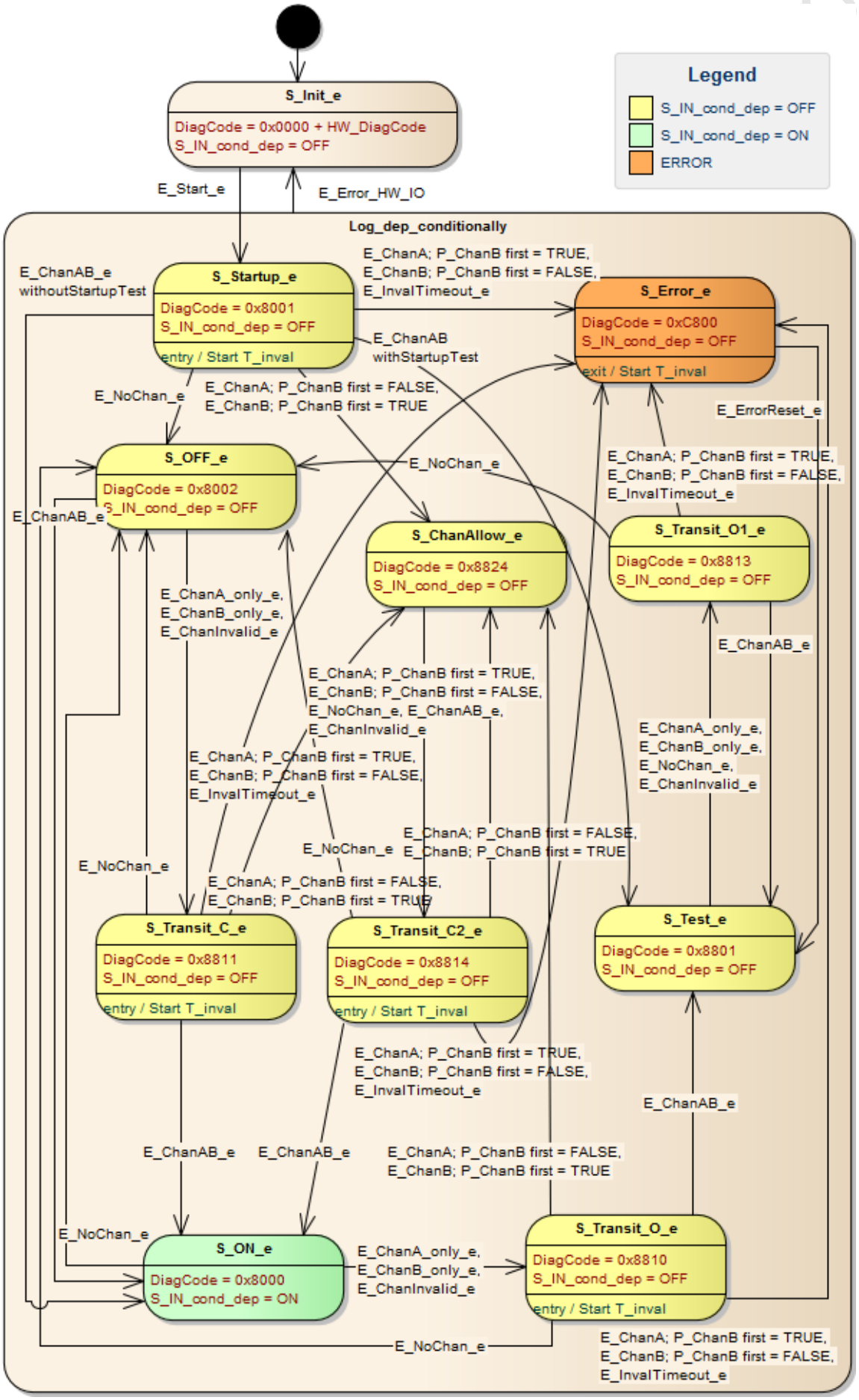
18289

Parameter	Data type	Meaning	Possible values	
S_IN_cond_dep	SAFEBOOL	Safe process signal	FALSE	Safe OFF or error state
			TRUE	Safe ON and no error state

State diagram (state machine)

20694

The state diagram shows the logical signal evaluation of the logical device:


Legend
S_IN_cond_dep = OFF
S_IN_cond_dep = ON
ERROR
S_Init_e
DiagCode = 0x0000 + HW_DiagCode
S_IN_cond_dep = OFF
E_Start_e
E_Error_HW_IO
Log_dep_conditionally
S_Startup_e
DiagCode = 0x8001
S_IN_cond_dep = OFF
entry / Start T_inval
E_ChanAB_e withoutStartupTest
E_ChanA; P_ChanB first = TRUE, E_ChanB; P_ChanB first = FALSE, E_InvalTimeout_e
E_ChanAB withStartupTest
E_NoChan_e
E_ChanA; P_ChanB first = FALSE, E_ChanB; P_ChanB first = TRUE
S_Error_e
DiagCode = 0xC800
S_IN_cond_dep = OFF
exit / Start T_inval
E_ErrorReset_e
S_OFF_e
DiagCode = 0x8002
S_IN_cond_dep = OFF
E_ChanAB_e
E_NoChan_e
E_ChanA; P_ChanB first = TRUE, E_ChanB; P_ChanB first = FALSE, E_InvalTimeout_e
S_ChanAllow_e
DiagCode = 0x8824
S_IN_cond_dep = OFF
S_Transit_O1_e
DiagCode = 0x8813
S_IN_cond_dep = OFF
E_ChanA_only_e, E_ChanB_only_e, E_ChanInvalid_e
E_ChanAB_e
E_ChanA; P_ChanB first = TRUE, E_ChanB; P_ChanB first = FALSE, E_NoChan_e, E_ChanAB_e, E_ChanInvalid_e
E_ChanA; P_ChanB first = TRUE, E_ChanB; P_ChanB first = FALSE, E_InvalTimeout_e
E_ChanA_only_e, E_ChanB_only_e, E_NoChan_e, E_ChanInvalid_e
E_ChanA; P_ChanB first = FALSE, E_ChanB; P_ChanB first = TRUE
E_NoChan_e
E_ChanA; P_ChanB first = FALSE, E_ChanB; P_ChanB first = TRUE
S_Transit_C_e
DiagCode = 0x8811
S_IN_cond_dep = OFF
entry / Start T_inval
S_Transit_C2_e
DiagCode = 0x8814
S_IN_cond_dep = OFF
entry / Start T_inval
S_Test_e
DiagCode = 0x8801
S_IN_cond_dep = OFF
E_ChanA; P_ChanB first = TRUE, E_ChanB; P_ChanB first = FALSE, E_InvalTimeout_e
E_ChanAB_e
E_ChanAB_e
E_ChanAB_e
E_ChanA; P_ChanB first = FALSE, E_ChanB; P_ChanB first = TRUE
E_NoChan_e
S_ON_e
DiagCode = 0x8000
S_IN_cond_dep = ON
E_ChanA_only_e, E_ChanB_only_e, E_ChanInvalid_e
S_Transit_O_e
DiagCode = 0x8810
S_IN_cond_dep = OFF
entry / Start T_inval
E_NoChan_e
E_ChanA; P_ChanB first = TRUE, E_ChanB; P_ChanB first = FALSE, E_InvalTimeout_e

Diagnostic information

19776

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187).

Following diagnostic codes exist:

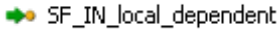
DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0x8813	S_Transit	Transition state	OFF
0x8814	S_Transit	Transition state	OFF
0x8824	S_ChanAllow	1 channel = TRUE, signal sequence OK	OFF
0xC800	S_Error	Locked error state	OFF

SF_IN_local_dependent

18286

Interface type: Local I/O interface (input)

Library: sf_logicalinterfaces.library

Symbol in CODESYS: 

Description

18285

Logical device for a safe sensor with 2 interdependently acting contacts/channels

SF_IN_local_dependent detects the switching states of the two input channels (IN_Channel_A, IN_Channel_B). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- Simultaneity of the signal change of the two contacts/channels is monitored during switch-on: If the signal states are not equal for a longer time than the set synchronisation time, both contacts/channels have to take the value FALSE simultaneously before switch-on is possible again. (SynchTime)
- If the signal state of the two signal channels is invalid longer than the firmly set time, the logical device changes to the locked error state. (InvalidTime = const. = 140ms)

The programmer can activate the following functions as option:

- EnableIN_Test:
  A test signal for cross-fault monitoring generated at a local output channel at the local I/O interface is evaluated.
- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.
- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.

If cross-fault detection is activated for 2-channel safe devices at the local I/O interface (EnableIN_Test = TRUE), the following applies:
Signal change at the inputs (IN_Channel_A, IN_Channel_B) must be applied longer than the cycle time of the test pulses (3 PLC cycles). Otherwise the logical device changes to the locked error state.

To reset the logical device from the locked error state and to read diagnostic information of the logical device: → **CtrlLocalInputs** (→ p. 194).

Parameter data

18284

Parameter	Data type	Meaning	Possible values	
IN_Channel_A	INT	Local input to which sensor channel A is connected.	1 ... 8	Local input IN1 ... local input IN8
IN_Channel_B	INT	Local input to which sensor channel B is connected.	1 ... 8	Local input IN1 ... local input IN8
EnableIN_Test	BOOL	Evaluation of a test signal which was created by a local output	FALSE	Test signal is ignored *
			TRUE	Test signal is evaluated
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON
SynchTime	WORD	Synchronisation time	0 250 ... 500 ... 60000	Infinite 250 milliseconds ... 500 milliseconds * ... 60000 milliseconds

* ... Default value

Process data

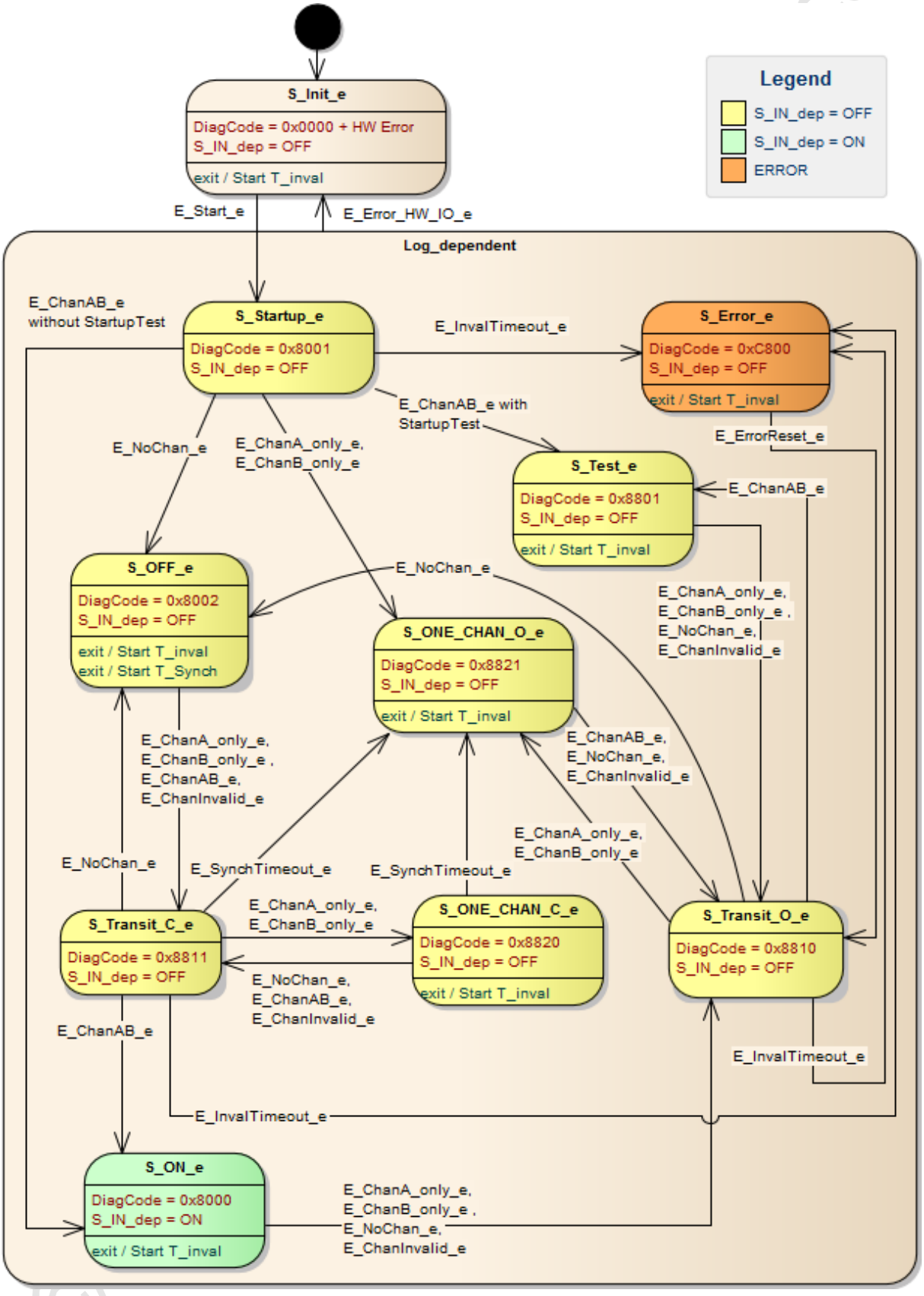
18283

Parameter	Data type	Meaning	Possible values	
S_IN_dep	SAFEBOOL	Safe process signal	FALSE	SAFE OFF
			TRUE	Safe ON

State diagram (state machine)

20689

The state diagram shows the logical signal evaluation of the logical device:



Diagnostic information

19773

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187).

Following diagnostic codes exist:

DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0x8820	S_ONE_CHAN_C	1 channel = TRUE, test of the synchronisation time	OFF
0x8821	S_ONE_CHAN_O	1 channel = TRUE, synchronisation time elapsed	OFF
0xC800	S_Error	Locked error state	OFF

SF_IN_local_dependent_filter_testreq

18301

Interface type: Local I/O interface (input)

Library: sf_logicalinterfaces.library

Symbol in CODESYS: 

Description

18302

Logical device for a safe sensor with 2 interdependently acting contacts/channels

SF_IN_local_dependent_filter_w_testreq detects the switching states of the two input channels (IN_Channel_A, IN_Channel_B). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- The contacts/channels are debounced during switch-on: Signal changes that are applied shorter than the set stabilisation time do not lead to a change in state. (StabilisingTime)
- Simultaneity of the signal change of the two contacts/channels is monitored during switch-on: If the signal states are not equal for a longer time than the set synchronisation time, both contacts/channels have to take the value FALSE simultaneously before switch-on is possible again. (SynchTime)
- Before the two contacts/channels are switched on again, they have to be FALSE simultaneously.
- If after the start the two contacts/channels do not take a valid signal state within a firmly set time, the logical device changes to the locked error state. (InvalidTime = const. = 140 ms)
- If during switching both signals are longer invalid than the set stabilisation time, the logical device changes to the locked error state. (StabilisingTime)

The programmer can activate the following functions as option:

- EnableIN_Test:
  A test signal for cross-fault monitoring generated at a local output channel at the local I/O interface is evaluated.
- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.
- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.

If cross-fault detection is activated for 2-channel safe devices at the local I/O interface (EnableIN_Test = TRUE), the following applies:
Signal change at the inputs (IN_Channel_A, IN_Channel_B) must be applied longer than the cycle time of the test pulses (3 PLC cycles). Otherwise the logical device changes to the locked error state.

To reset the logical device from the locked error state and to read diagnostic information of the logical device: → **CtrlLocalInputs** (→ p. 194).

Parameter data

18303

Parameter	Data type	Meaning	Possible values	
IN_Channel_A	INT	Local input to which sensor channel A is connected.	1 ... 8	Local input IN1 ... local input IN8
IN_Channel_B	INT	Local input to which sensor channel B is connected.	1 ... 8	Local input IN1 ... local input IN8
EnableIN_Test	BOOL	Evaluation of a test signal which was created by a local output	FALSE	Test signal is ignored *
			TRUE	Test signal is evaluated
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON
SynchTime	WORD	Synchronisation time	0 250 ... 500 ... 60000	Infinite 250 milliseconds ... 500 milliseconds * ... 60000 milliseconds
StabilisingTime	WORD	Max. bounce time	250 ... 10000	250 milliseconds * ... 10000 milliseconds

* ... Default value

Process data

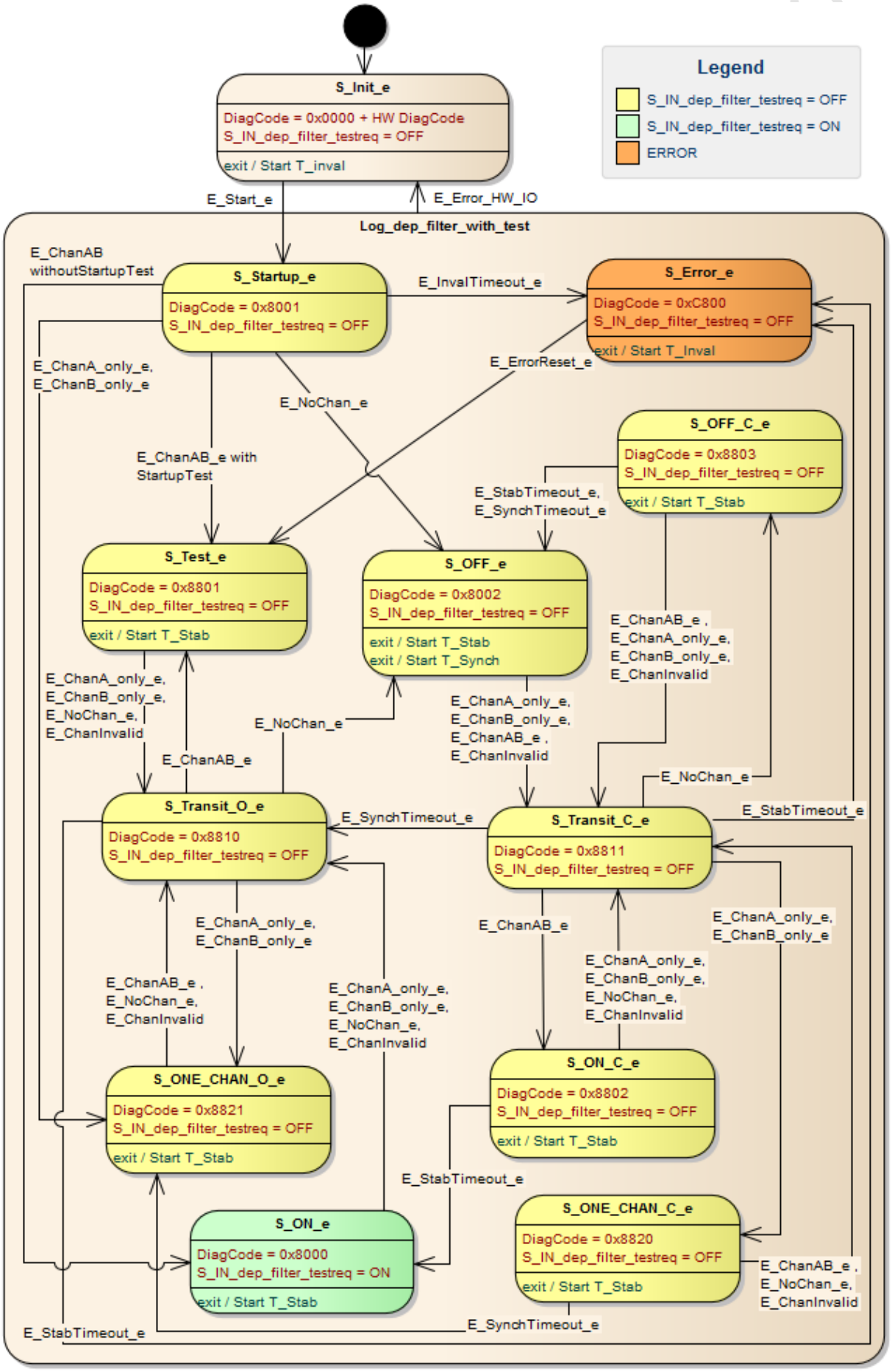
18304

Parameter	Data type	Meaning	Possible values	
S_IN_dep_filter_testreq	SAFEBOOL	Safe process signal	FALSE	SAFE OFF
			TRUE	Safe ON

State diagram (state machine)

20691

The state diagram shows the logical signal evaluation of the logical device:


Legend
S_IN_dep_filter_testreq = OFF
S_IN_dep_filter_testreq = ON
ERROR
S_Init_e
DiagCode = 0x0000 + HW DiagCode
S_IN_dep_filter_testreq = OFF
exit / Start T_inval
E_Start_e
E_Error_HW_IO
Log_dep_filter_with_test
E_ChanAB withoutStartupTest
S_Startup_e
DiagCode = 0x8001
S_IN_dep_filter_testreq = OFF
E_InvalTimeout_e
S_Error_e
DiagCode = 0xC800
S_IN_dep_filter_testreq = OFF
exit / Start T_Inval
E_ErrorReset_e
E_ChanA_only_e, E_ChanB_only_e
E_NoChan_e
E_ChanAB_e with StartupTest
S_OFF_C_e
DiagCode = 0x8803
S_IN_dep_filter_testreq = OFF
exit / Start T_Stab
E_StabTimeout_e, E_SynchTimeout_e
S_Test_e
DiagCode = 0x8801
S_IN_dep_filter_testreq = OFF
exit / Start T_Stab
S_OFF_e
DiagCode = 0x8002
S_IN_dep_filter_testreq = OFF
exit / Start T_Stab
exit / Start T_Synch
E_ChanAB_e , E_ChanA_only_e, E_ChanB_only_e, E_ChanInvalid
E_ChanA_only_e, E_ChanB_only_e, E_NoChan_e, E_ChanInvalid
E_NoChan_e
E_ChanAB_e
E_ChanA_only_e, E_ChanB_only_e, E_ChanAB_e , E_ChanInvalid
E_NoChan_e
S_Transit_O_e
DiagCode = 0x8810
S_IN_dep_filter_testreq = OFF
E_SynchTimeout_e
S_Transit_C_e
DiagCode = 0x8811
S_IN_dep_filter_testreq = OFF
E_StabTimeout_e
E_ChanA_only_e, E_ChanB_only_e
E_ChanAB_e
E_ChanA_only_e, E_ChanB_only_e
E_ChanAB_e , E_NoChan_e, E_ChanInvalid
E_ChanA_only_e, E_ChanB_only_e, E_NoChan_e, E_ChanInvalid
E_ChanA_only_e, E_ChanB_only_e, E_NoChan_e, E_ChanInvalid
S_ONE_CHAN_O_e
DiagCode = 0x8821
S_IN_dep_filter_testreq = OFF
exit / Start T_Stab
S_ON_C_e
DiagCode = 0x8802
S_IN_dep_filter_testreq = OFF
exit / Start T_Stab
E_StabTimeout_e
S_ON_e
DiagCode = 0x8000
S_IN_dep_filter_testreq = ON
exit / Start T_Stab
S_ONE_CHAN_C_e
DiagCode = 0x8820
S_IN_dep_filter_testreq = OFF
exit / Start T_Stab
E_ChanAB_e , E_NoChan_e, E_ChanInvalid
E_StabTimeout_e
E_SynchTimeout_e

Diagnostic information

19780

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187)

Following diagnostic codes exist:

DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8802	S_ON_C	Both channels = TRUE, check the set timer	OFF
0x8803	S_OFF_C	Both channels = FALSE, check the set timer	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0x8820	S_ONE_CHAN_C	1 channel = TRUE, test of the synchronisation time	OFF
0x8821	S_ONE_CHAN_O	1 channel = TRUE, synchronisation time elapsed	OFF
0xC800	S_Error	Locked error state	OFF

SF_IN_local_dependent_filter_ntestreq

18305

Interface type: Local I/O interface (input)

Library: sf_logicalinterfaces.library

Symbol in CODESYS: 

Description

18306

Logical device for a safe sensor with 2 interdependently acting contacts/channels

SF_IN_local_dependent_filter_w_testreq detects the switching states of the two input channels (IN_Channel_A, IN_Channel_B). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- The contacts/channels are debounced during switch-on: Signal changes that are applied shorter than the set stabilisation time do not lead to a change in state. (StabilisingTime)
- If the signal states are not equal for a longer time than the set synchronisation time, both contacts/channels have to take the value FALSE simultaneously before switch-on is possible again. (SynchTime)
- Simultaneity of the signal change of the two contacts/channels is monitored during switch-on: If both contacts/channels take the value TRUE again during the switch-off process, switch-on occurs at once. When the tolerance time has elapsed, the two contacts/channels have to take the value FALSE again simultaneously before switch-on is possible again. (ToleranceTime)
- If after the start the two contacts/channels do not take a valid signal state within a firmly set time, the logical device changes to the locked error state. (InvalidTime = const. = 140 ms)
- If during switching both signals are longer invalid than the set stabilisation time, the logical device changes to the locked error state. (StabilisingTime)

The programmer can activate the following functions as option:

- EnableIN_Test:
  A test signal for cross-fault monitoring generated at a local output channel at the local I/O interface is evaluated.
- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.
- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.

If cross-fault detection is activated for 2-channel safe devices at the local I/O interface (EnableIN_Test = TRUE), the following applies:
Signal change at the inputs (IN_Channel_A, IN_Channel_B) must be applied longer than the cycle time of the test pulses (3 PLC cycles). Otherwise the logical device changes to the locked error state.

To reset the logical device from the locked error state and to read diagnostic information of the logical device: → **CtrlLocalInputs** (→ p. 194).

Parameter data

18307

Parameter	Data type	Meaning	Possible values	
IN_Channel_A	INT	Local input to which sensor channel A is connected.	1 ... 8	Local input IN1 ... local input IN8
IN_Channel_B	INT	Local input to which sensor channel B is connected.	1 ... 8	Local input IN1 ... local input IN8
EnableIN_Test	BOOL	Evaluation of a test signal which was created by a local output	FALSE	Test signal is ignored *
			TRUE	Test signal is evaluated
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON
SynchTime	WORD	Synchronisation time	0 250 ... 500 ... 60000	Infinite 250 milliseconds ... 500 milliseconds * ... 60000 milliseconds
StabilisingTime	WORD	Max. bounce time	250 ... 10000	250 milliseconds * ... 10000 milliseconds
ToleranceTime	WORD	Tolerance time for switch-off	250 ... 1000	250 milliseconds * ... 1000 milliseconds

* ... Default value

Process data

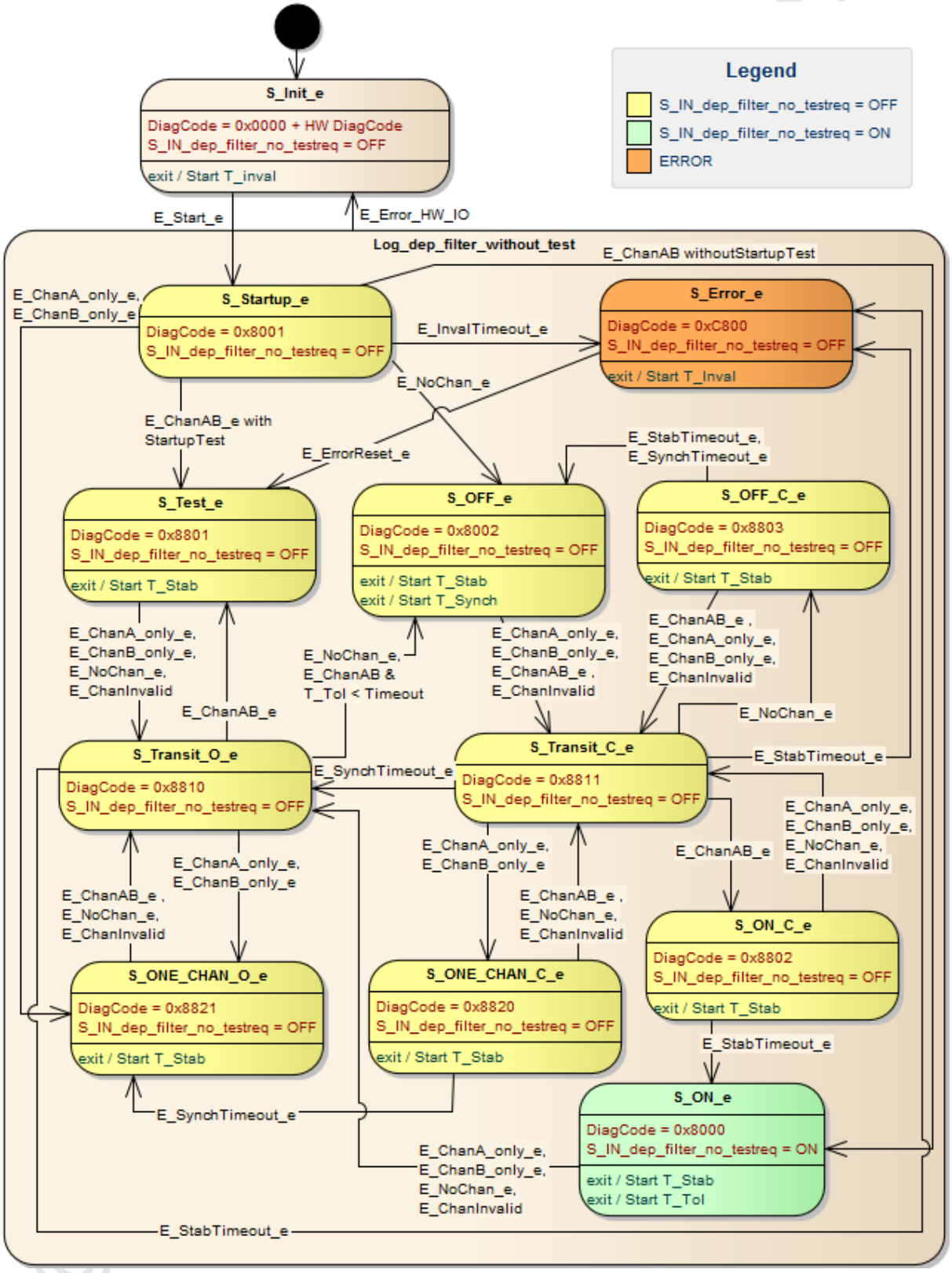
18308

Parameter	Data type	Meaning	Possible values	
S_IN_dep_filter_no_testreq	SAFEBOOL	Safe process signal	FALSE	SAFE OFF
			TRUE	Safe ON

State diagram (state machine)

20692

The state diagram shows the logical signal evaluation of the logical device:



Diagnostic information

19783

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187).

Following diagnostic codes exist:

DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8802	S_ON_C	Both channels = TRUE, check the set timer	OFF
0x8803	S_OFF_C	Both channels = FALSE, check the set timer	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0x8820	S_ONE_CHAN_C	1 channel = TRUE, test of the synchronisation time	OFF
0x8821	S_ONE_CHAN_O	1 channel = TRUE, synchronisation time elapsed	OFF
0xC800	S_Error	Locked error state	OFF

SF_IN_local_dependent_filter_nshutdown

18312

Device type: Local I/O interface (input)

Library: sf_logicalinterfaces.library

Symbol in CODESYS: 

Description

18311

Logical device for a safe sensor with 2 interdependently acting contacts/channels

SF_IN_local_dependent_filter_nshutdown detects the switching states of the two input channels of the local I/O interface (IN_Channel_A, IN_Channel_B). The logical device forms the logical AND connector from the two values and provides the result as safe control signal. The following additional conditions apply for the logical signal evaluation:

- The contacts/channels are debounced during switch-on: Signal changes that are applied shorter than the set stabilisation time do not lead to a change in state. (StabilisingTime)
- Simultaneity of the signal change of the two contacts/channels is monitored during switch-on: If the signal states are not equal for a longer time than the set synchronisation time, both contacts/channels have to take the value FALSE simultaneously before switch-on is possible again. (SynchTime)
- If both contacts/channels take the value TRUE again during the switch-off process, switch-on occurs at once. When the tolerance time has elapsed, the two contacts/channels have to take the value FALSE again simultaneously before switch-on is possible again. (ToleranceTime)
- If after the start the two contacts/channels do not take a valid signal state within a firmly set time, the logical device changes to the locked error state. (InvalidTime = const. = 140ms)

The programmer can activate the following functions as option:

- EnableIN_Test:
  A test signal for cross-fault monitoring generated at a local output channel at the local I/O interface is evaluated.
- StartupTest:
  Both signal channels must take the state FALSE simultaneously before the logical device can start with the signal evaluation.
- DisableLogging:
  Only error events are transmitted to the online support centre (OSC) of the SmartPLC SafeLine AC4S user interface.

If cross-fault detection is activated for 2-channel safe devices at the local I/O interface (EnableIN_Test = TRUE), the following applies:
Signal change at the inputs (IN_Channel_A, IN_Channel_B) must be applied longer than the cycle time of the test pulses (3 PLC cycles). Otherwise the logical device changes to the locked error state.

When this logical device is used, the process safety time (PST) of the system is extended by one cycle time of the fail-safe PLC.

► Note the remarks about process safety time (→ Device manual, **Process safety time**)!

► Observe the set cycle time of the fail-safe PLC (→ **Set cycle time of the fail-safe PLC** (→ p. 100))!

To reset the logical device from the locked error state and to read diagnostic information of the logical device: → **CtrlLocalInputs** (→ p. 194).

Parameter data

18310

Parameter	Data type	Meaning	Possible values	
IN_Channel_A	INT	Local input to which sensor channel A is connected.	1 ... 8	Local input IN1 ... local input IN8
IN_Channel_B	INT	Local input to which sensor channel B is connected.	1 ... 8	Local input IN1 ... local input IN8
EnableIN_Test	BOOL	Evaluation of a test signal which was created by a local output	FALSE	Test signal is ignored *
			TRUE	Test signal is evaluated
DisableLogging	BOOL	Transmission of the event messages to OSC of the device	FALSE	Following states of the state diagram are transmitted to the OSC: * ▪ S_ON ▪ S_OFF ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
			TRUE	Following states of the state diagram are transmitted to the OSC: ▪ S_Test ▪ S_Error ▪ S_Init (HW-Error)
EnableStartupTest	BOOL	Control execution of the start-up test	FALSE	No start-up test *
			TRUE	Start-up test is executed; slave waiting for start-up sequence OFF ⇨ ON
SynchTime	WORD	Synchronisation time	0 250 ... 500 ... 60000	Infinite 250 milliseconds ... 500 milliseconds * ... 60000 milliseconds
StabilisingTime	WORD	Max. bounce time	250 ... 10000	250 milliseconds * ... 10000 milliseconds
ToleranceTime	WORD	Tolerance time for switch-off	250 ... 1000	250 milliseconds * ... 1000 milliseconds

* ... Default value

Process data

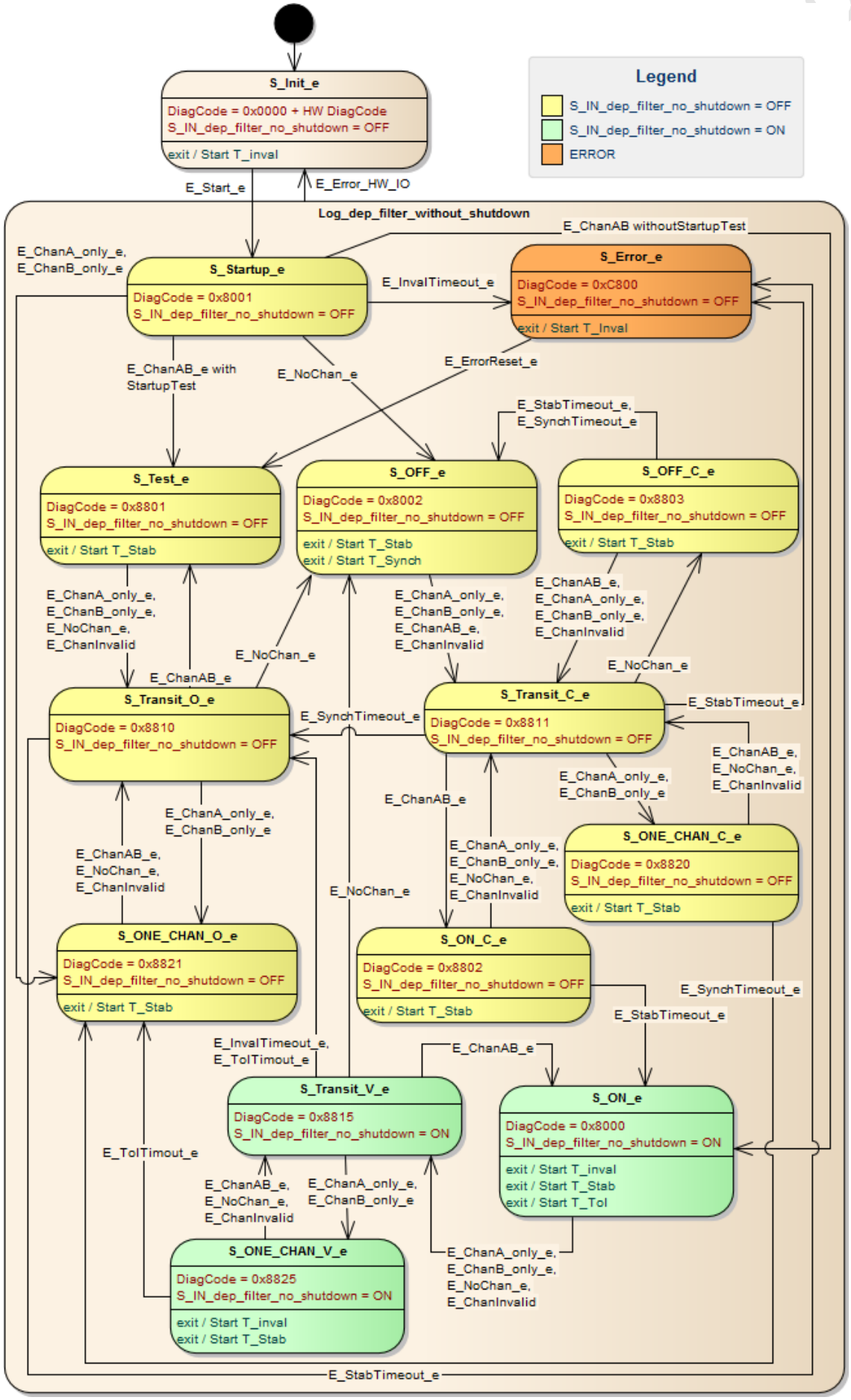
18309

Parameter	Data type	Meaning	Possible values	
S_IN_dep_ntshutdown	SAFEBOOL	Safe process signal	FALSE	Safe OFF or error
			TRUE	Safe ON and no error

State diagram (state machine)

20693

The state diagram shows the logical signal evaluation of the logical device:


Legend
S_IN_dep_filter_no_shutdown = OFF
S_IN_dep_filter_no_shutdown = ON
ERROR
S_Init_e
DiagCode = 0x0000 + HW DiagCode
S_IN_dep_filter_no_shutdown = OFF
exit / Start T_inval
E_Start_e
E_Error_HW_IO
Log_dep_filter_without_shutdown
E_ChanAB withoutStartupTest
E_ChanA_only_e,
E_ChanB_only_e
S_Startup_e
DiagCode = 0x8001
S_IN_dep_filter_no_shutdown = OFF
E_InvalTimeout_e
S_Error_e
DiagCode = 0xC800
S_IN_dep_filter_no_shutdown = OFF
exit / Start T_Inval
E_ChanAB_e with
StartupTest
E_NoChan_e
E_ErrorReset_e
E_StabTimeout_e,
E_SynchTimeout_e
S_Test_e
DiagCode = 0x8801
S_IN_dep_filter_no_shutdown = OFF
exit / Start T_Stab
S_OFF_e
DiagCode = 0x8002
S_IN_dep_filter_no_shutdown = OFF
exit / Start T_Stab
exit / Start T_Synch
S_OFF_C_e
DiagCode = 0x8803
S_IN_dep_filter_no_shutdown = OFF
exit / Start T_Stab
E_ChanA_only_e,
E_ChanB_only_e,
E_NoChan_e,
E_ChanInvalid
E_NoChan_e
E_ChanAB_e
E_ChanA_only_e,
E_ChanB_only_e,
E_ChanAB_e,
E_ChanInvalid
E_ChanAB_e,
E_ChanA_only_e,
E_ChanB_only_e,
E_ChanInvalid
E_NoChan_e
S_Transit_O_e
DiagCode = 0x8810
S_IN_dep_filter_no_shutdown = OFF
E_SynchTimeout_e
S_Transit_C_e
DiagCode = 0x8811
S_IN_dep_filter_no_shutdown = OFF
E_StabTimeout_e
E_ChanAB_e,
E_NoChan_e,
E_ChanInvalid
E_ChanA_only_e,
E_ChanB_only_e
E_ChanAB_e
E_ChanA_only_e,
E_ChanB_only_e
E_ChanAB_e,
E_NoChan_e,
E_ChanInvalid
S_ONE_CHAN_C_e
DiagCode = 0x8820
S_IN_dep_filter_no_shutdown = OFF
exit / Start T_Stab
E_ChanA_only_e,
E_ChanB_only_e,
E_NoChan_e,
E_ChanInvalid
E_NoChan_e
S_ONE_CHAN_O_e
DiagCode = 0x8821
S_IN_dep_filter_no_shutdown = OFF
exit / Start T_Stab
S_ON_C_e
DiagCode = 0x8802
S_IN_dep_filter_no_shutdown = OFF
exit / Start T_Stab
E_SynchTimeout_e
E_StabTimeout_e
E_InvalTimeout_e,
E_TolTimout_e
E_ChanAB_e
S_Transit_V_e
DiagCode = 0x8815
S_IN_dep_filter_no_shutdown = ON
S_ON_e
DiagCode = 0x8000
S_IN_dep_filter_no_shutdown = ON
exit / Start T_inval
exit / Start T_Stab
exit / Start T_Tol
E_TolTimout_e
E_ChanAB_e,
E_NoChan_e,
E_ChanInvalid
E_ChanA_only_e,
E_ChanB_only_e
S_ONE_CHAN_V_e
DiagCode = 0x8825
S_IN_dep_filter_no_shutdown = ON
exit / Start T_inval
exit / Start T_Stab
E_ChanA_only_e,
E_ChanB_only_e,
E_NoChan_e,
E_ChanInvalid
E_StabTimeout_e

Diagnostic information

19786

The logical device creates diagnostic codes. They reflect the active internal state of the evaluation logic (state machine). The diagnostic codes can be read with the following FB: → **CtrlASi_InSlave** (→ p. 187).

Following diagnostic codes exist:

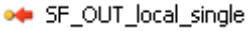
DiagCode	Name	Description	Output
0x0000	S_Init	Initialisation state	OFF
0x8001	S_Startup	Start state	OFF
0x8002	S_OFF	Safe OFF	OFF
0x8000	S_ON	Safe ON	ON
0x8801	S_Test	State for going through the start-up test	OFF
0x8802	S_ON_C	Both channels = TRUE, check the set timer	OFF
0x8803	S_OFF_C	Both channels = FALSE, check the set timer	OFF
0x8810	S_Transit	Transition state	OFF
0x8811	S_Transit	Transition state	OFF
0x8815	S_Transit	Transition state	ON
0x8820	S_ONE_CHAN_C	1 channel = TRUE, test of the synchronisation time	OFF
0x8821	S_ONE_CHAN_O	1 channel = TRUE, synchronisation time elapsed	OFF
0x8825	S_ONE_CHAN_V	1 channel = TRUE; tolerance time not yet elapsed	ON
0xC800	S_Error	Locked error state	OFF

SF_OUT_local_single

18313

Interface type: Local I/O interface (output)

Library: sf_logicalinterfaces.library

Symbol in CODESYS: 

Description

18314

Logical device for a 1-channel safe actuator at the local I/O interface

SF_OUT_local_single detects the Boolean value at the process data signal (OUT_). The logical device generates a safe Boolean signal and provides the value at the configured output channel of the local I/O interface (OUT_Channel).

Parameter data

18316

Parameter	Data type	Meaning	Possible values	
OUT_Channel	INT	Local output to which the actuator is connected.	1 ... 4	Local output OUT1 ... local output OUT4
DisableLogging	Reserved - settings have no influence on function of the logical device			

Process data

18315

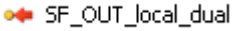
Parameter	Data type	Meaning	Possible values	
OUT_	SAFEBOOL	Target status of the output / the outputs	FALSE	SAFE OFF
			TRUE	Safe ON

SF_OUT_local_dual

18321

Device type: Local I/O interface (output)

Library: sf_logicalinterfaces.library

Symbol in CODESYS: 

Description

18323

Logical device to control a 2-channel safe actuator at the local I/O interface

SF_OUT_local_dual detects the Boolean value at the process data signal (OUT_). The logical device generates a safe Boolean signal and provides the value at the configured output channels of the local I/O interface (OUT_Channel_A, OUT_Channel_B).

Parameter data

18322

Parameter	Data type	Meaning	Possible values	
OUT_Channel_A	INT	Local output to which actuator channel A is connected.	1 ... 4	Local output OUT1 ... Local output OUT4
OUT_Channel_B	INT	Local output to which actuator channel B is connected.	1 = ... 4 =	Local output OUT1 ... Local output OUT4
DisableLogging	Reserved - settings have no influence on function of the logical device			

Process data

18324

Parameter	Data type	Meaning	Possible values	
OUT_	SAFEBOOL	Target status of the output / the outputs	FALSE	SAFE OFF
			TRUE	Safe ON

SF_OUT_local_testpulse

18317

Device type: Local I/O interface (output)

Library: sf_logicalinterfaces.library

Symbol in CODESYS: SF_OUT_local_testpulse

Description

18318

Logical device to generate a test pulse at an output channel of the local I/O interface

SF_OUT_local_testpulse generates a test pulse and provides it at the configured output channel of the local I/O interface (OUT_Channel). Pulse duration and frequency are firmly set:

Test pulse characteristics	
Pulse duration:	2 PLC cycles
Pulse frequency:	12 PLC cycles
Phase shift:	OUT1 ⇨ OUT2 ⇨ OUT3 ⇨ OUT4 ⇨ OUT1 = 3 PLC cycles

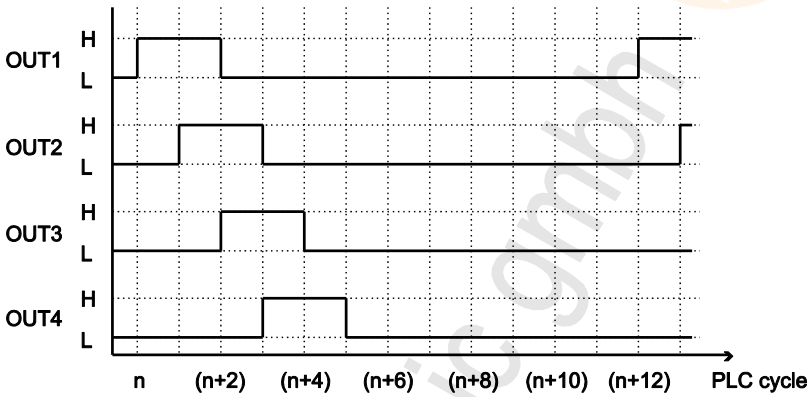


Figure: Signal characteristics and phase shift of the test pulse

Parameter data

18319

Parameter	Data type	Meaning	Possible values	
OUT_Channel	INT	Local output at which the test pulse is to be provided	1 ... 4	Local output OUT1 ... Local output OUT4

10.5 Error codes: Safe AS-i slaves

18256

Communication and configuration errors change the logical device to the initialisation state (S_INIT). Simultaneously a diagnostic code is provided to the control function block (→ CtrlASi_InSlave , → CtrlASi_OutSlave) which describes the error cause.

Following diagnostic codes are available:

DiagCode	Error name (FB) Error message (OSC)	Error description
0x4001	ASi_SYS_ERROR_CFG_MASTERID_INVALID Wrong Master-Id in ASi-Control-FUB	Wrong master in control FB or logical device
0x4002	ASi_SYS_ERROR_CFG_SLAVEADDR_INVALID Wrong Slave Address in ASi-Control-FUB	Wrong slave in the control FB or logical device (address 0 in the device tree CODESYS provokes an error during download)
0x4003	ASi_SYS_ERROR_CFG_MULTIPLE_SLAVE Try to config multiple Input Slaves with same Master-Id and Slave-Address	Incorrect configuration by two slaves with same address
0x4004	ASi_SYS_ERROR_CFG_MISMATCH_SLAVE Try to config a Input and Output Slave at the same Master-Id and Slave-Address	Incorrect configuration by input and output slaves with same address
0x4005	ASi_SYS_ERROR_CFG_CANTADD_SLAVE Can't add a new ASi-Input-Slave in case of a error return from AS-i-Control	New AS-i input slave could not be added to the project.
0x4009	ASi_SYS_ERROR_DIAGCFG_NOTCONFIGUED Try to monitor a not configured ASi-Slave in a AS-i-Control-FUB	Logical device not configured
0x400A	ASi_SYS_ERROR_DIAGCFG_HSI_REQUEST At CtrlASiOut-FUB : HSI1 and HSI2 Request at the same Time. Don't activate HSI1 and HSI2 together.	Help signals HSI1 and HSI2 were activated simultaneously.
0x4010	ASi_SYS_OUTPUT_CANTADD Can't add a new ASi-Output-Contol-Slave (ex. Max. count of ControlSlaves reached)	▪ Incorrect configuration by too many output control slaves in the project. ▪ This error can be masked by error 0x4009.
0x5800	ASi_HW_SUB_STATE_SLAVE_MISSING_TESTREQUEST Forced Testrequest from AS-i-SlaveCheck; Waiting for manual Test of related AS-i-Input-Slave	Test request detected.
0x5820	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_TIMEOUT Errorcode received from AS-i-Control	Error code received from AS-i control slave
0x5821	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_A_B_CHANNEL_OPEN Errorcode received from AS-i-Control	Error code received from AS-i control slave
0x5822	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_A_CHANNEL_OPEN Errorcode received from AS-i-Control	Error code received from AS-i control slave
0x5823	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_B_CHANNEL_OPEN Errorcode received from AS-i-Control	Error code received from AS-i control slave
0x5824	ASi_HW_CONFIG_SLAVE_ERROR_CODESEQUENCE_INVALID	Error code received from AS-i control

DiagCode	Error name (FB)	Error description
	Errorcode received from AS-i-Control	slave
0x5825	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_CODESEQUENCE_DOUBLY	▪ 2 slaves with the same code sequence in the AS-i network ▪ Errors 0x5825 and 0x5826 can occur simultaneously
	Errorcode received from AS-i-Control	
0x5826	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_CODESEQUENCE_RULES	▪ Error when the code sequence is taught ▪ Errors 0x5825 and 0x5826 can occur simultaneously
	Errorcode received from AS-i-Control	
0x5827	ASi_HW_CONFIG_SLAVE_TEACH_ERROR_NO_CODESEQUENCE	▪ Error when the code sequence is taught ▪ This error is masked on the control FB by error 0x5840.
	Errorcode received from AS-i-Control	
0x5830	ASi_HW_GLOBAL_COM_MISSING_BUS_CYCLE_EVENT	All bus errors can cover local errors.
	Errorcode received from AS-i-Control	
0x5831	ASi_HW_GLOBAL_COM_ADDRESS SEQUENCE ERROR	All bus errors can mask local errors.
	Errorcode received from AS-i-Control	
0x5832	ASi_HW_GLOBAL_COM_FAILURE_TO_LESS_SLAVES	Too few AS-i slaves at the bus
	Errorcode received from AS-i-Control	
0x5840	ASi_HW_ERROR_STATE_INIT	▪ No AS-i slave at the bus ▪ All bus errors can mask local errors. ▪ This error can be masked by error 0x5820.
	AS-i slave is missing	

10.6 Error codes: Local I/O interface

18259

Communication and configuration errors change the logical device to the initialisation state (S_INIT). Simultaneously a diagnostic code is provided to the control function block that describes the error cause (→ **CtrlLocalInputs** (→ p. 194)).

Following diagnostic codes are available:

DiagCode	Error name (FB) Error message (OSC)	Error description
0x4020	LIO_SYS_ERROR_CFG_CHANNEL_A_INVALID Wrong Id for Channel A in LIO-CtrlFUB	Wrong ID for channel A (channel A = 0 or >8)
0x4021	LIO_SYS_ERROR_CFG_CHANNEL_B_INVALID Wrong Id for Channel B in LIO-CtrlFUB	Wrong ID for channel B (channel A = 0 or >8)
0x4022	LIO_SYS_ERROR_CFG_MULTIPLE_CHANNEL Try to config multiple Local IOs with same Channel	Several logical devices with same channel assignment
0x4023	LIO_SYS_ERROR_CFG_DOUBLE_CHANNEL Try to configure both channels of a dual channel IO with the same channel number (Channel A equal Channel B)	Same ID used for both channels of a two-channel logical device
0x4024	LIO_SYS_ERROR_CFG_NOT_SUPPORTED Actually not supported output device (e.g. a DEV_SF_OUT_Local_dual)	Non-supported output device
0x4025	LIO_SYS_ERROR_CFG_EXHAUSTED Max. count of configurable Inputs reached	Maximum number of configurable inputs reached
0x4026	LIO_SYS_ERROR_INPUT_CANTADD Can't add a new Local Input in case of a error return from Local-IO-Control	Adding a local input failed
0x4031	LIO_SYS_ERROR_DIAGCFG_NOTCONFIGURED Try to monitor a not configured Local Input in a LIO-Control-FUB	▪ Non-configured local input ▪ This error can be masked by error 0x4031
0x4032	LIO_SYS_ERROR_DIAGCFG_CONFIGMISMATCH Try to config the same Input at single and dual	Wrong assignment of the input channels in a control FB OR An input configured as fail-safe is polled with FB GetLocalInput
0xE100	LIO_HW_ERROR_WRONG_TESTSIGNAL_e Detect unexpected Testsignal in a local Input channel (a reason could be a short cut)	Cross fault monitoring
0xE101	LIO_HW_ERROR_NO_TESTSIGNAL_e Can't detect a expected Testsignal (no connection)	Cross fault monitoring
0xE102	LIO_HW_INPUT_TEST_IN_PROCESS_e Wait for Test after both input signals switched ON.	Cross fault monitoring

11 Index

A

Access standard input and output data 50
ACnnnn_SysCmd 178
Add additional device to the project 41
Add and configure EtherCAT slave 45
Add EtherCAT master 43
Add fieldbus stack 46
Add logical device to the project tree 27, 34
Add safe AS-i slaves to the project 27
Add safe local device to project 34
Add visualisation to a project 65
Analogue input and output data 53
Analogue output data 55
Appendix 115
ASI_ADDRESS_MODE (ENUM) 170
ASI_ANA_IO (STRUCT) 175
ASI_ANALOG_IN (STRUCT) 176
ASI_ANALOG_IN_FLAGS (STRUCT) 176
ASI_ANALOG_OUT (STRUCT) 177
ASI_ANALOG_OUT_FLAGS (STRUCT) 177
ASI_BIN_IN (STRUCT) 175
ASI_BIN_IO (STRUCT) 174
ASI_BIN_OUT (STRUCT) 175
ASI_DATA (STRUCT) 172
ASI_MASTER (ENUM) 170
ASI_MASTER_MODE (ENUM) 170
ASI_NET (STRUCT) 174
ASI_SLAVE (ENUM) 171
ASI_SLAVE_TYP (ENUM) 171
Available fieldbus stacks 42
Available logical devices for safe AS-i slaves 26
Available logical devices for safe inputs/outputs 33
Available memory space 76
Available safety libraries 74

C

CAN-ID 109
Change between the states 110
Change IP settings 40
Change network settings 60
Change properties of the web visualisation 67
Change system time of the device 49
CODESYS programming software 12
Complex variables (STRUCT) 172
Components of the ifm package 13
Concept of the logical devices 25, 32
Configure AS-i master 59
Configure AS-i slaves 59
Configure EtherCAT master 44
Configure fail-safe PLC 24
Configure FSoE connection 91
Configure logical device 29, 37
Configure main task 70
Configure programming interface 21
Configure safe AS-i slaves 25
Configure safe devices at local I/O interface 32
Configure safety task 100
Configure standard devices at local I/O interface 39
Configure standard PLC 23
Configure system 59
Configure task processing 70
Configure visualisation 67
Control interface of the ifm function blocks 57
Create a visualisation 66
Create additional objects for data exchange 86
Create CODESYS project 16
Create new project with SmartPLC SafeLine AC4S 17
Create safe application 73
Create standard PLC application 48
Create test signal at local output 39
Create user account 20
Cross communication 68
Ctrl_SetDiagInfo 196
CtrlASi_InSlave 187
CtrlASi_OutSlave 189
CtrlASi_ResetAllSlaves 192
CtrlLocalInputs 194

D

Data exchange between standard PLC and fail-safe PLC 83
Data transfer safety area >>> standard area 85
Data transfer standard area >>> safety area 84
Delete boot application via SD card 105
Description 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 155, 157, 159, 161, 164, 166, 168, 178, 187, 189, 192, 194, 196, 198, 199, 206, 210, 214, 218, 222, 226, 230, 234, 236, 240, 245, 252, 257, 263, 267, 273, 274, 275
Diagnostic information 209, 213, 217, 221, 225, 229, 233, 239, 244, 251, 256, 262, 266, 272
Diagnostic messages 204
Digital input and output data 53
Digital output data 55
Display operating state of the standard application 106
Display web visualisation 114
Download standard application to SmartPLC SafeLine AC4S 102

E

Enumeration types (ENUM) 170
Enumeration types and complex variables 170
Error codes
  Local I/O interface 278
  Safe AS-i slaves 276
Example
  Change language setting of the device 183
  Date / time / read NTP settings 184
  Read LCEMS, LCEAS and LDAE of AS-i master 2 185
Extend fieldbus functionality 42

F

Fail-safe PLC 108, 113
FB with cyclic execution 58
FB with one-time execution 57
Fieldbus data 54

G

General safety instructions 9

Get_ASi_Data 161
Get_ASi_PHY_Dat 118
Get_CDI 136
Get_FieldbusInfo 164
Get_InputParameter 140
Get_LAS 120
Get_LCE 126
Get_LCEAS 128
Get_LCEMS 130
Get_LDAE 132
Get_LDS 122
Get_LPF 134
Get_LPS 124
Get_OutputParameter 142
Get_PCD 138
GetLocalInput 198
Getting started 15

H

Hardware 11

I

ID 109
ifm AS-i Package 13
ifm weltweit • ifm worldwide • ifm à l'échelle internationale 283
Input and output data of the fieldbus interface 54
Input parameter 197
Input parameters 118, 120, 122, 124, 126, 128, 130, 132, 134, 136, 138, 140, 142, 144, 146, 148, 150, 152, 154, 155, 157, 159, 161, 164, 166, 168, 179, 188, 190, 192, 194, 198, 199
Install CODESYS Development System 12
Install the ifm package 14
Installation 12

L

Legal and copyright information 6
Library ACnnnn_SYS_CMD.library 178
Library ACnnnn_Utils.library 116
Library SF_IO.library 186
Library SF_LogicalInterfaces.library 201
Licensing 11
Load safe application to SmartPLC SafeLine AC4S 103
Local I/O interface
- Access non-safe inputs/outputs 79
- Provide diagnostic information 79
- Reset logical device 78

Logical devices for safe AS-i slaves 205
Logical devices for the local I/O interface 235

M

Manage AS-i network 60
Map safe process signal to variable 30, 38
Minimum number of AS-i slaves 75
Modification history 8

N

Note
- Projects with network variables 104

Notes on the FSoEMaster 92

O

Objects of a safe application 72
Objects of a standard PLC application 47
Operating mode of the standard PLC 106
Operating states SmartPLC SafeLine AC4S 106
Operation 101
Operation mode of the fail-safe PLC 108
Options to access input and output data 50
Output parameters 119, 121, 123, 125, 127, 129, 131, 133, 135, 137, 139, 141, 143, 145, 147, 149, 151, 153, 154, 156, 158, 160, 162, 165, 167, 169, 182, 188, 191, 193, 195, 197, 198, 200
Outputs data of the AS-i slaves 54
Overview
- AS-i functions (FB_ASi) 117
- CODESYS documentation of 3S 8
- Project structure with SmartPLC SafeLine AC4S 19
- System functions (FB_System) 163
- User documentation for AC4S 7

P

Parameter data . 207, 211, 215, 219, 223, 227, 231, 234, 237, 241, 246, 253, 258, 264, 268, 273, 274, 275
PI controller 6
Pin safe project 76
PLCopen Safety
- Provide diagnostic information 79

Preliminary note 6
Process data 207, 211, 215, 219, 223, 227, 231, 234, 237, 241, 246, 253, 258, 264, 268, 273, 274
Process data of the AS-i slaves 52
Process data of the non-safe local devices 56
Process safety time for the SmartPLC SafeLine AC4S 93
Programming 47
Programming of the fail-safe PLC 71
Purpose of the document 6

Q

QuickSetupASi_Master 166

R

Read configuration data of the slaves 62
Read network settings 61
Read parameter images 61
read slave lists 61
Read status of the voltage supply 62
Remarks 25, 32
Remarks about logical devices 202
Remarks about the use of exchange variables 83
Remove safe AS-i slave from project 31
Remove safe local device from project 39
Required background knowledge 9
Reset 111
Reset safe application (cold) 113
Reset the standard application (cold) 112
Reset the standard application (origin) 112
Reset the standard application (warm) 112

S

Safe AS-i input slaves 25

Safe AS-i output slaves 26
Safe AS-i slaves
Provide diagnostic information 78
Reset logical device 77
Transmit help signals HSI_1 and HSI_2 to safe AS-i output modules 78
Safe cross communication 87
Safe devices at local inputs 32
Safe devices at local outputs 33
Safe fieldbus communication 91
Safety instructions 9
SafetyFSoEMaster library 82
SafetyPLCopen library 81
SafetyStandard library 80
Send commands to the system and the AS-i master 63
Sequence of the signal changes 203
Set communication path of standard PLC 21
Set communication path of the fail-safe PLC 22
Set cycle time of the fail-safe PLC 100
Set interval of the data exchange 86
Set parameters for visualisation task 70
Set process safety time 93
Set the parameters of the logical device 29, 37
Set up fail-safe network variables 88
Set variable components of the response time 99
Set_AddressMode 144
Set_ASi_Config 146
Set_LPS 148
Set_Mode 150
Set_PCD 152
Set_ProjectAll 154
Set_SlaveAddress 155
Set_SlaveExtendedID1 157
Set_SlaveParameter 159
Set_TimeDate 168
SetLocalOutput 199
SF_IN_ASi_conditionally_dependent 214
SF_IN_ASi_dependent 218
SF_IN_ASi_dependent_filter_nshutdown 230
SF_IN_ASi_dependent_filter_ntestreq 226
SF_IN_ASi_dependent_filter_w_testreq 222
SF_IN_ASi_forced 206
SF_IN_ASi_independent 210
SF_IN_local_conditionally_dependent 245
SF_IN_local_dependent 252
SF_IN_local_dependent_filter_nshutdown 267
SF_IN_local_dependent_filter_ntestreq 263
SF_IN_local_dependent_filter_testreq 257
SF_IN_local_forced 236
SF_IN_local_independent 240
SF_OUT_local_dual 274
SF_OUT_local_single 273
SF_OUT_local_testpulse 275
SF_OUTcontrol_ASi 234
Software 11
standard PLC 106
Standard PLC 111
Start CODESYS 15
Start standard PLC application 107
Start-up test 203
State 220
State diagram (state machine) ..208, 212, 216, 224, 228, 232, 238, 242, 247, 254, 259, 265, 269
State machines for logical preprocessing 202
State transitions 202
States 202
States of the fail-safe PLC 109
States of the input signals 203
States of the safe application 109
States of the standard PLC application 106
Step 1
Calculate typical response times 95
Step 2
Calculate the response times of the safety function 96
Step 3
Define watchdog times 97
Step 4
Calculate the worst-case response time of the safety function 98
Stop standard PLC application 107
Supported program organization units (POUs) 74
Supported programming languages 49, 75
Supported reset variants 111
Supported types of reset 113
Supported visualisation types 64
Switch operating states 107
Symbols and styles used 7
System configuration 23
System events 204
System requirements 11

T

Table
AS-i master commands 180
System commands 179
Teach code sequences of the safe AS-i slaves 104
Timer events 204
Transfer CODESYS project to device 101

U

Uninstall the ifm package 14
Use CODESYS online help 20
Use complex variables 60
Use EtherCAT master 43
Use fail-safe network variables in the application 90
Use functions of the ifm package 57
Use network variables 68
Use remanent variables 48
Use safety functions of CODESYS 80
Use safety functions of the ifm AS-i package 77
Use the factory-set replacement variables 84
Use visualisations 64

V

Validity of the interface data 51
Variable declaration with data transfer between standard and safety area 76

W

Warnings used 10

12 ifm weltweit • ifm worldwide • ifm à l'échelle internationale

Version: 2017-12-18 8310

ifm electronic gmbh • Friedrichstraße 1 • 45128 Essen
www.ifm.com • Email: info@ifm.com
Service hotline: 0800 / 16 16 16 (only Germany, Mo-Fr 07.00...18.00 h)

ifm Niederlassungen • Sales offices • Agences

D	Niederlassung Nord • 31135 Hildesheim • Tel. 05121 7667-0
	Niederlassung West • 45128 Essen • Tel. 0201 36475 -0
	Niederlassung Mitte-West • 58511 Lüdenscheid • Tel. 02351 4301-0
	Niederlassung Süd-West • 64646 Heppenheim • Tel. 06252 7905-0
	Niederlassung Baden-Württemberg • 73230 Kirchheim • Tel. 07021 8086-0
	Niederlassung Bayern • 82178 Puchheim • Tel. 089 80091-0
	Niederlassung Ost • 07639 Tautenhain • Tel. 036601 771-0
AE	ifm electronic FZC • Saif Zone, Sharjah • phone +971- 6-5573601
AR	ifm electronic s.r.l. • 1107 Buenos Aires • phone +54 11 5353-3436
AT	ifm electronic gmbh • 1120 Wien • phone +43 / 1 / 617 45 00
AU	ifm efector pty ltd. • Mulgrave Vic 3170 • phone +61 1300 365 088
BE	ifm electronic n.v./s.a. • 1731 Zellik • phone +32 2 481 0220
BG	ifm electronic eood • 1202 Sofia • phone +359 2 807 59 69
BR	ifm electronic Ltda. • 03337-000 Sao Paulo / SP • phone +55-11-2672-1730
CA	ifm efector Canada inc. • Mississauga, ON L5N 2X7 • phone +1 855-436-2262
CH	ifm electronic ag • 4624 Härkingen • phone +41 / 800 88 80 33
CL	ifm electronic SpA • Oficina 5041 Comuna de Conchalí • phone +56-2-32239282
CN	ifm electronic (Shanghai) Co. Ltd. • 201203 Shanghai • phone +86 21 3813 4800
CZ	ifm electronic, spol. s.r.o. • 140 00 Praha 4 • phone +420 267 990 211
DK	ifm electronic a/s • 2605 Brøndby • phone +45 70 20 11 08
ES	ifm electronic s.a. • 08820 El Prat de Llobregat • phone +34 93 479 30 80
FI	ifm electronic oy • 00440 Helsinki • phone +358 75 329 5000
FR	ifm electronic s.a. • 93192 Noisy-le-Grand Cedex • phone +33 0820 22 30 01
GB	ifm electronic Ltd. • Hampton, Middlesex TW12 2HD • phone +44 / 20 / 8213 0000
GR	ifm electronic monoprosopi E.P.E. • 15125 Amaroussio • phone +30 210 61 800 90
HU	ifm electronic kft. • 9028 Györ • phone +36-96 / 518-397
IN	ifm electronic India Private Limited • Kolhapur, 416234 • phone +91 / 231 / 267 27 70
IE	ifm electronic (Ireland) Ltd. • Dublin 22 • phone +353 / 1 / 461 32 00
IT	ifm electronic s.r.l. • 20864 Agrate Brianza (MB) • phone +39 39-6899982
JP	efector co., ltd. • Chiba-shi, Chiba 261-7118 • phone +81 043-299-2070
KR	ifm electronic Ltd. • 04420 Seoul • phone +82 2-790-5610
MX	ifm efector S. de R.L. de C.V. • San Pedro Garza Garcia, N.L. 66269 • phone +52-81-8040-3535
MY	ifm electronic Pte. Ltd • 47100 Puchong, Selangor • phone +603 8066 9853
NA	ifm elctronic (pty) Ltd • 25 Dr. W. Kulz Street Windhoek • phone +264 61 300984
NL	ifm electronic b.v. • 3843 GA Harderwijk • phone +31 341-438 438
NZ	ifm efector pty ltd • 930 Great South Road Penrose, Auckland • phone +64 / 95 79 69 91
PL	ifm electronic sp. z o.o. • 40-106 Katowice • phone +48 32 70 56 400
PT	ifm electronic s.a. • 4410-137 São Félix da Marinha • phone +351 223 71 71 08
RO	ifm electronic s.r.l • Sibiu 557260 • phone +40 269 224 550
RU	ifm electronic • 105318 Moscow • phone +7 495 921-44-14
SG	ifm electronic Pte Ltd • 609 916 Singapore • phone +65 6562 8661
SK	ifm electronic s.r.o. • 831 06 Bratislava • phone +421 244 872 329
SE	ifm electronic ab • 412 50 Göteborg • phone +46 31-750 23 00
TR	ifm electronic Ltd. Sti. • 34381 Sisli, Istanbul • phone +90 212 210 50 80
TW	ifm electronic • Kaohsiung City, 806, Taiwan R.O.C. • phone +886 7 3357778
UA	TOV ifm electronic • 02660 Kiew • phone +380 44 501-85-43
US	ifm efector inc. • Malvern, PA 19355 • phone +1 800-441-8246
VN	ifm electronic Vietnam Co., Ltd. • 700000 Ho Chi Minh City • phone +84-28-2253.6715
ZA	ifm electronic (Pty) Ltd. • 0157 Pretoria • phone +27 12 450 0412