SIEMENS

SIMATIC

Industrial Software SIMATIC Safety - Configuring and Programming

Programming and Operating Manual

Product Overview 1

Configuring 2

Safety Administration Editor 3

Access protection 4

Programming 5

F-I/O access 6

Implementation of user acknowledgment 7

Data exchange between standard user program and safety program 8

Safety-related communication 9

Compiling and commissioning a safety program 10

System acceptance 11

Operation and maintenance 12

STEP 7 Safety V21 instructions 13

Monitoring and response times A

Continued on next page

11/2025
A5E02714440-AR

Continued

Checklist B

Assignment of the F-BaseID during runtime (delayed transfer) (S7-1200 G2, S7-1500) C

Industrial Software
SIMATIC Safety - Configuring and Programming

Programming and Operating Manual

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER

indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING

indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION

indicates that minor personal injury can result if proper precautions are not taken.

NOTICE

indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified persons are those who, because of their training and experience, are familiar with the installation, assembly, commissioning, operation, decommissioning and disassembly of the product and can recognize risks and avoid possible hazards.

Proper use of Siemens products

Note the following:

⚠ WARNING

Siemens products may only be used for the application described in the catalog and the associated usage information. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens Aktiengesellschaft. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens Aktiengesellschaft
Digital Industries
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E02714440-AR
Ⓟ 10/2025 Subject to change

Copyright © Siemens 2011 - 2025.
All rights reserved

Important notes

Purpose of this documentation

The information in this documentation enables you to configure (Page 35) and program (Page 111) SIMATIC Safety fail-safe systems. In addition, you will obtain information on acceptance (Page 413) of a SIMATIC Safety F-system.

NOTE

The Programming and Operating Manual "SIMATIC Safety - Configuring and Programming" in its latest version (possibly including product information for the manual) is the relevant source of all information on functional safety regarding configuring and programming. This also applies in the event of discrepancies between this manual and other documentation on functional safety regarding configuring and programming of SIMATIC Safety.

You must heed all warnings in the Programming and Operating Manual "SIMATIC Safety - Configuring and Programming".

Basic knowledge requirements

General basic knowledge of automation engineering is needed to understand this documentation. Basic knowledge of the following is also necessary:

- Fail-safe automation systems
- Automation systems
  - S7-300
  - S7-400
  - S7-1200
  - S7-1500
  - S7-1500H
  - S7-1500 Virtual Controller
  - S7-1500 Software Controller
  - SIMATIC Drive Controller
- Distributed I/O systems on
  - PROFIBUS DP
  - PROFINET IO

- Totally Integrated Automation Portal, including:
  - Hardware configuration with the *hardware and network editor*
  - Programming in the LAD and FBD programming languages using the *program editor*.
  - Communication between CPUs
  - OPC UA for communication between CPUs with OPC UA Safety

Scope of this documentation

This documentation is valid for *STEP 7 Safety Advanced V21*.

STEP 7 Safety Advanced V21 is used for configuring and programming the SIMATIC Safety fail-safe system.

In this context, integration of the F-I/O listed below in SIMATIC Safety is also addressed:

- S7-1500/ET 200MP fail-safe modules
- ET 200SP fail-safe modules
- ET 200SP HA fail-safe modules
- ET 200S fail-safe modules
- ET 200eco fail-safe I/O modules
- ET 200eco PN fail-safe I/O modules
- ET 200AL fail-safe I/O modules
- ET 200pro fail-safe modules
- ET 200iSP fail-safe modules
- S7-300 fail-safe signal modules
- S7-1200 fail-safe modules
- Fail-safe GSD based DP slaves
- Fail-safe GSD based I/O devices

Approvals

The SIMATIC Safety F-system is certified for use in safety mode up to:

- Safety Integrity Level SIL3 in accordance with IEC 61508:2010
- Performance Level (PL) e and category 4 in accordance with ISO 13849-1:2015 or EN ISO 13849-1:2015

Incorporation in the information landscape

Depending on your application, you will need the following supplementary documentation when working with *STEP 7 Safety*.

This documentation includes references to the supplementary documentation where appropriate.

Documentation	Brief description of relevant content
For the SIMATIC Safety F-system	Depending on which F-CPU you are using, you will need the following documentation: • For the F-CPUs S7-1200/1500, a Product Information (https://support.industry.siemens.com/cs/ww/en/view/109478599) describes all deviations from the respective standard CPUs. • Each F-CPU S7-300/400 that can be used has its own product information. The product information describes the deviations from the respective standard CPUs. • The Device manuals (https://support.industry.siemens.com/cs/ww/en/ps/13717/man) describe the S7-1500 CPUs. • The "S7-300, CPU 31xC and CPU 31x: Installation" (https://support.industry.siemens.com/cs/ww/en/view/13008499) operating instructions describe the installation and wiring of S7-300 systems. • The "CPU 31xC and CPU 31x, Technical Data" (https://support.industry.siemens.com/cs/ww/en/view/12996906) device manual describes the CPUs 315-2 DP and PN/DP, the CPU 317-2 DP and PN/DP, and the CPU 319-3 PN/DP. • The "S7-400 Automation System, Installation (https://support.industry.siemens.com/cs/ww/en/view/1117849) installation manual describes the installation and wiring of S7-400 systems. • The "S7-400 Automation System, CPU Data" (https://support.industry.siemens.com/cs/ww/en/view/109818285) reference manual describes the CPUs 414-3 PN/DP, the CPU 416-2, and the CPU 416-3 PN/DP. • The "ET 200S Interface Module IM 151-7 CPU" (https://support.industry.siemens.com/cs/ww/en/view/12714722) manual describes the IM 151-7 CPU. • The "ET 200S, Interface Module IM 151-8 PN/DP CPU" (https://support.industry.siemens.com/cs/ww/en/view/47409312) manual describes the IM 151-8 PN/DP CPU. • The "ET 200S, Interface Module IM 154-8 CPU" (https://support.industry.siemens.com/cs/ww/en/view/44251850) manual describes the IM 154-8 CPU. • The "SIMATIC S7-1500 Software Controller CPU 1505SP (F), CPU 1507S (F) and CPU 1508S (F) SIMATIC Industrial OS Version (https://support.industry.siemens.com/cs/ww/en/view/109808199)" manual describes the SIMATIC S7-1500 Software Controllers CPU 1505SP (F), CPU 1507S (F) and CPU 1508S (F). • The function manuals (https://sieportal.siemens.com/su/blSqW) describe the S7-1500 Virtual Controller.
"S7-1200 Functional Safety manual (https://support.industry.siemens.com/cs/ww/en/view/104547552)" system manual	Describes the F-CPUs S7-1200 and the fail-safe modules S7-1200 (including installation, wiring, and technical specifications)

Documentation	Brief description of relevant content
"S7-1500/ET 200MP system manual (https://support.industry.siemens.com/cs/ww/en/view/59191792)" system manual and the product manuals (https://support.industry.siemens.com/cs/ww/en/ps/29727/man) for the corresponding S7-1500/ET 200MP fail-safe modules	Describes the hardware of the S7-1500 systems and the S7-1500/ET 200MP fail-safe modules (including installation, wiring, and technical specifications)
System manual "Redundant System S7-1500R/H (https://support.industry.siemens.com/cs/ww/en/view/109754833)"	Describes the hardware of S7-1500R/H systems (including design, wiring and technical specifications)
"SIMATIC Drive Controller (https://support.industry.siemens.com/cs/ww/en/view/109766665)" system manual and "SIMATIC Drive Controller (https://support.industry.siemens.com/cs/ww/en/view/109766666)" equipment manual	Describes the hardware of the SIMATIC Drive Controller (including design, wiring and technical specifications)
"SIMATIC S7-1500 S7-PLCSIM Advanced (https://support.industry.siemens.com/cs/ww/en/view/109955578)" function manual	Describes the operation of S7-PLCSIM Advanced
"ET 200SP distributed I/O system (https://support.industry.siemens.com/cs/ww/en/view/58649293)" system manual and the product manuals (https://support.industry.siemens.com/cs/ww/en/ps/14059/man) for the corresponding ET 200SP fail-safe modules	Describes the hardware of the ET 200SP fail-safe modules (including installation, wiring, and technical specifications)
System manual "Distributed I/O System ET 200SP HA (https://support.industry.siemens.com/cs/ww/en/view/109817158)" and the equipment manuals (https://support.industry.siemens.com/cs/ww/en/ps/28940/man) for the respective ET 200SP HA fail-safe modules	Describes the hardware of the ET 200SP HA fail-safe modules (including setup, wiring and technical specifications).
"ET 200eco Distributed I/O Station Fail-safe I/O Block (https://support.industry.siemens.com/cs/ww/en/view19033850)" manual	Describes the hardware of the ET 200eco fail-safe I/O module (including installation, wiring, and technical specifications)
Manual "ET 200eco PN F-DI 8 x 24 VDC, 4xM12 / F-DQ 3 x 24 VDC/2.0A PM, 3xM12 (https://support.industry.siemens.com/cs/ww/en/view/109765611)"	Describes the hardware of the ET 200eco PN fail-safe I/O module (including installation, wiring, and technical specifications)

Documentation	Brief description of relevant content
System manual "ET 200AL system manual (https://support.industry.siemens.com/cs/ww/en/view/89254965)" and equipment manual "Digital I/O Module F-DI 4+F-DQ 2x24VDC/2A, 4xM12" (https://support.industry.siemens.com/cs/ww/en/view/109798489)	Describes the hardware of the ET 200AL fail-safe I/O module (including installation, wiring, and technical specifications)
"Distributed I/O System ET 200S, Fail-Safe Modules (https://support.industry.siemens.com/cs/ww/en/view/27235629)" operating instructions	Describes the hardware of the ET 200S fail-safe modules (including installation, wiring, and technical specifications)
"S7-300 Automation System, ET 200M Distributed I/O System, Fail-safe Signal Modules (https://support.industry.siemens.com/cs/ww/en/view/19026151)" manual	Describes the hardware of the S7-300 fail-safe signal modules (including installation, wiring, and technical specifications)
"Distributed I/O system ET 200pro, fail-safe I/O modules (https://support.industry.siemens.com/cs/ww/en/view22098524)" operating instructions	Describes the hardware of the ET 200pro fail-safe modules (including installation, wiring, and technical specifications)
"ET 200iSP distributed I/O device - Fail-safe modules (https://support.industry.siemens.com/cs/ww/en/view/47357221)" operating instructions	Describes the hardware of the ET 200iSP fail-safe modules (including installation, wiring, and technical specifications)
"Communication (https://support.industry.siemens.com/cs/ww/en/view/59192925)" Function Manual	Contains the description of communication with OPC UA
Help on *STEP 7*	• Describes the operation of the standard tools in *STEP 7* • Contains information regarding configuration and parameter assignment of hardware • Contains a description of the programming languages FBD and LAD • Contains the description of communication with OPC UA

Fail-safe modules Manual Collection

The Manual Collection contains all the documentation for the SIMATIC fail-safe modules collected into one file.

You can find the Manual Collection on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109806400).

Guide

This documentation describes how to work with *STEP 7 Safety*. It includes instructions and reference sections (description of the instructions for the Safety program).

The following topics are addressed:

- Configuration of SIMATIC Safety
- Access protection for SIMATIC Safety
- Programming of the Safety program (safety-related user program)
- Safety-related communication
- Instructions for the Safety program
- Support for the system acceptance
- Operation and maintenance of SIMATIC Safety
- Monitoring and response times

Conventions

The following conventions apply:

- In this documentation, the terms "safety engineering" and "fail-safe engineering" are used synonymously. The same applies to the terms "fail-safe" and "F-".
- "F-systems" include redundant S7-1500HF systems. Special features and restrictions of H systems are described in the system manual "Redundant S7-1500R/H system (https://support.industry.siemens.com/cs/ww/en/view/109754833)" and also apply to the redundant S7-1500HF systems.
- *"STEP 7 Safety V21"* stands for *"STEP 7 Safety Advanced V21"*.
- "(S7-300)" indicates that the section **only** applies to S7-300 F-CPUs. S7-300 F-CPUs also includes the F-CPUs ET 200S and ET 200pro (IM F-CPUs).
- "(S7-400)" indicates that the section **only** applies to S7-400 F-CPUs.
- "(S7-1200 G1)" indicates that the section **only** applies to S7-1200 F-CPUs with article number 6ES721x-1xF**40**....
- "(S7-1200 G2)" indicates that the section **only** applies to S7-1200 F-CPUs with article number 6ES721x-1xF**50**....
- "(S7-1200)" indicates that the section **only** applies to S7-1200 G1 F-CPUs and S7-1200 G2 F-CPUs.
- "(S7-1500)" indicates that the section **only** applies to S7-1500 F-CPUs. S7-1500 F-CPUs also include S7-1500 HF-CPUs, ET 200SP F-CPUs, the CPU 151xpro F-2 PN, the S7-1500 F Software Controller, the SIMATIC Drive Controller and the S7-1500 F Virtual Controller. In the case of exceptions, these are indicated.

The scopes can be combined.

The term **Safety program** refers to the fail-safe portion of the user program and is used instead of "fail-safe user program", "F-program", etc. For purposes of contrast, the non-safety-related part of the user program is referred to as the "standard user program".

The **hardware configuration** includes the configuration of standard CPUs and standard I/Os as well as the configuration of F-CPUs and F-I/Os.

The **safety-related hardware configuration** includes the safety-related parameters of the F-CPUs and F-I/Os.

The **safety-related project data** includes the safety-related hardware configuration as well as the Safety program.

NOTE

Each warning is marked with a unique number at the end of the text. This enables you to easily reference other documents, for example, to obtain an overview of the safety requirements for the system.

Additional support

If you have further questions about the use of products presented in this manual, contact your local Siemens representative.

You can find information on whom to contact on the Web (https://www.automation.siemens.com/aspa_app).

A guide to the technical documentation for the various SIMATIC products and systems is available on the Web (http://www.siemens.com/simatic-tech-doku-portal).

You can find the online catalog and online ordering system on the Web (www.siemens.com/industrymall).

Training center

We offer courses to help you get started with the S7 automation system. To do this, contact your regional training center or the central training center in Nuremberg, Germany.

More information can be found on the Internet (http://www.sitrain.com).

Technical Support

To contact Technical Support for all Industry Automation products, use the Support Request Web form (http://www.siemens.com/automation/support-request).

More information can be found about our Technical Support on the Web (http://www.siemens.com/automation/service).

Important note for maintaining the operational safety of your system

NOTE

The operators of systems with safety-related characteristics must adhere to operational safety requirements. The supplier is also obliged to comply with special product monitoring measures. Siemens informs system operators in the form of personal notifications about product developments and properties which could be or become important issues in terms of operational safety.

You should subscribe to the corresponding notifications in order to obtain the latest information and to allow you to make any necessary modifications to your system.

Log in to the Industry Online Support. Follow the links below and click on "Email on update" on the right-hand side in each case:

- SIMATIC S7-300/S7-300F (https://support.industry.siemens.com/cs/products?pnid=13751&lc=en-WW)
- SIMATIC S7-400/S7-400H/S7-400F/FH (https://support.industry.siemens.com/cs/products?pnid=13828&lc=en-WW)
- SIMATIC S7-1500/SIMATIC S7-1500F/SIMATIC S7-1500HF (https://support.industry.siemens.com/cs/products?pnid=13716&lc=en-WW)
- SIMATIC S7-1200/SIMATIC S7-1200F (https://support.industry.siemens.com/cs/products?pnid=13683&lc=en-WW)
- Software Controller (https://support.industry.siemens.com/cs/products?pnid=13911&lc=en-WW)
- Distributed I/O (https://support.industry.siemens.com/cs/products?pnid=14029&lc=en-WW)
- STEP 7 (TIA Portal) (https://support.industry.siemens.com/cs/products?pnid=24471&lc=en-WW)
- Virtual Controller (https://support.industry.siemens.com/cs/ww/en/ps/29702)

Cybersecurity information

Siemens provides products and solutions with industrial cybersecurity functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial cybersecurity concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial cybersecurity measures that may be implemented, please visit
https://www.siemens.com/cybersecurity-industry.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Cybersecurity RSS Feed under
https://new.siemens.com/cert.

SiePortal

You can find current information on the following topics quickly and easily here:

- **Catalog and ordering system**

  Ordering system for automation and drive solutions on the basis of Totally Integrated Automation (TIA) and Totally Integrated Power (TIP).

- **Product support**

  All the information and extensive know-how on your product, technical specifications, FAQs, certificates, downloads, and manuals.

- **Application examples**

  Tools and examples to solve your automation tasks – as well as function blocks, performance information and videos.

- **Services**

  Information about Industry Services, Field Services, Technical Support, spare parts and training offers.

- **Forums**

  For answers and solutions concerning automation technology.

- **mySiePortal**

  Your personal work area in the SiePortal for notifications, support requests, and configurable documents.

This information is provided by SiePortal on the Internet (https://sieportal.siemens.com/).

Table of contents

1 Product Overview 23

1.1 Overview 23

1.2 Hardware and Software Components 24

1.3 Installing/uninstalling the STEP 7 Safety Advanced V21 license 30

1.4 Migrating projects from S7 Distributed Safety V5.4 SP5 to STEP 7 Safety Advanced 31

1.5 Migrating PLC programs to a n F-CPU S7-1500 31

1.6 Upgrading projects to the latest version of STEP 7 Safety 32

1.6.1 Upgrading projects from STEP 7 Safety as of V14 SP1 to the latest version 33

1.7 First steps 33

2 Configuring 35

2.1 Overview of Configuration 35

2.2 Peculiarities when configuring fail-safe GSD based DP slaves and fail-safe GSD based I/O devices 39

2.3 Configuring an F-CPU 41

2.3.1 Special features of the S7-1500 F Virtual Controller 47

2.3.1.1 Operating the S7-1500 F Virtual Controller without central Virtual Timeservice 49

2.3.1.2 Operating the S7-1500 F Virtual Controller with central Virtual Timeservice 52

2.4 Configuring F-I/O 53

2.5 Configuration control (option handling) for F-I/Os 57

2.5.1 Example 58

2.6 Configuring shared device 61

2.7 Configuring isochronous mode (S7-1200 G2, S7-1500) 62

2.8 Recommendation for PROFIsafe address assignment 63

2.9 Configurations supported by the SIMATIC Safety F-system 64

2.10 PROFIsafe addresses for F-I/O of PROFIsafe address type 1 67

2.11 PROFIsafe addresses for F-I/O of PROFIsafe address type 2 68

2.12 PROFIsafe addresses for F-I/O of PROFIsafe address type 3 70

2.13 Setting the F-destination address for F-I/O with DIP switches 73

2.14 Assigning a PROFIsafe address of the F-I/Os with SIMATIC Safety 73

2.14.1 Identifying F-modules 75

2.14.2 Assign PROFIsafe address 77

2.14.3 Assigning PROFIsafe address to an F-module in a shared device in multiple projects 77

2.14.4 Assigning PROFIsafe address to an F-module in a shared device in a common project 78

2.14.5 Changing the PROFIsafe address 79

3 Safety Administration Editor 80

3.1 Opening the Safety Administration Editor 83

3.2 "General" area of the Safety Administration Editor 83

3.3 "F-runtime group" area of the Safety Administration Editor 87
3.3.1 "F-runtime group" area 87
3.3.2 Pre-/postprocessing (S7-1200, S7-1500) 88

3.4 "F-blocks" area of the Safety Administration Editor 89

3.5 "F-compliant PLC data types" area of the Safety Administration Editor (S7-1200, S7-1500) 90

3.6 "Web server F-admins" area of the Safety Administration Editor (S7-1200 G1, S7-1500) 91

3.7 "Settings" area of the Safety Administration Editor 91

3.8 "Flexible F-Link" area of the Safety Administration Editor (S7-1200, S7-1500) 97

4 Access protection 99

4.1 Overview of access protection 99

4.2 Access protection for the safety-related project data 100
4.2.1 CPU-wide access protection for the safety-related project data 100
4.2.2 Project-wide access protection for the safety-related project data 102

4.3 Access protection for the F-CPU 103
4.3.1 Access protection through the use of protection levels 103
4.3.2 Access protection through user management 106

4.4 Access protection through organizational measures 108

5 Programming 111

5.1 Overview of Programming 111
5.1.1 Program structure of the safety program (S7-300, S7-400) 111
5.1.2 Program structure of the safety program (S7-1200, S7-1500) 113
5.1.3 Safety program in Software Unit (Safety Unit) (S7-1500) 116
5.1.4 Fail-Safe Blocks 117
5.1.5 Restrictions in the programming languages FBD/LAD 119
5.1.6 F-compliant PLC data types (UDT) (S7-1200, S7-1500) 126
5.1.6.1 Grouping PLC tags for inputs and outputs of F-I/O in structures (S7-1200, S7-1500) 127
5.1.6.2 Example of structured PLC tags for inputs and outputs of F-I/O (S7-1200, S7-1500) 128
5.1.7 Named value data types (S7-1500) 131
5.1.8 Editing PLC tags with external editors 131
5.1.9 Using Multiuser engineering 131
5.1.10 Deleting the safety program 131

5.2 Defining F-Runtime Groups 133
5.2.1 Rules for F-Runtime Groups of the Safety Program 133
5.2.2 Procedure for defining an F-runtime group (S7-300, S7-400) 134
5.2.3 Procedure for defining an F-runtime group (S7-1200, S7-1500) 138
5.2.4 F-runtime group communication (S7-300, S7-400) 143
5.2.5 F-runtime group communication (S7-1200, S7-1500) 146
5.2.6 Deleting an F-runtime group 149

5.2.7 Changing the F-runtime group (S7-300, S7-400) 149
5.2.8 Changing the F-runtime group (S7-1200, S7-1500) 149
5.3 Creating F-blocks 150
5.3.1 Creating F-blocks 150
5.3.2 Know-how protection 151
5.3.3 Reuse of F-blocks 152
5.4 Information interface to the safety program 154
5.4.1 F-shared DB (S7-300, S7-400) 154
5.4.2 F-runtime group information DB (S7-1200, S7-1500) 154
5.5 Programming startup protection 156
6 F-I/O access 157
6.1 Addressing F-I/O 157
6.2 Value status (S7-1200, S7-1500) 159
6.3 Process Data or Fail-Safe Values 162
6.4 F-I/O DB 164
6.4.1 Name and number of the F-I/O DB 164
6.4.2 Tags of the F-I/O DB 165
6.4.2.1 PASS_ON 166
6.4.2.2 ACK_NEC 166
6.4.2.3 ACK_REI 167
6.4.2.4 IPAR_EN 168
6.4.2.5 DISABLE 169
6.4.2.6 QBAD/PASS_OUT/DISABLED/QBAD_I_xx/QBAD_O_xx and value status 169
6.4.2.7 ACK_REQ 170
6.4.2.8 IPAR_OK 170
6.4.2.9 DIAG 171
6.4.2.10 BASEID_ACTIVE 172
6.4.3 Accessing tags of the F-I/O DB 172
6.5 Passivation and reintegration of F-I/O 172
6.5.1 After startup of F-system 173
6.5.2 After communication errors 175
6.5.3 After F-I/O or channel faults 177
6.5.4 Group passivation 181
6.6 F-I/O access via I-slave-slave communication (S7-300, S7-400) 183
6.6.1 Overview 183
6.6.2 Configuring Safety-Related I-Slave-Slave Communication 183
6.6.3 Safety-related I-slave-slave communication - F-I/O access via process image 189
6.6.4 Limits for data transfer of safety-related I-slave-I-slave communication 189
7 Implementation of user acknowledgment 190
7.1 Implementing User Acknowledgment in the Safety Program of the F-CPU of a DP Master or IO controller 190
7.2 Implementing user acknowledgment in the safety program of the F-CPU of a I-slave or I-device 194

8 Data exchange between standard user program and safety program 198

8.1 Data Transfer from the Safety Program to the Standard User Program 198

8.2 Data Transfer from Standard User Program to Safety Program 200

9 Safety-related communication 202

9.1 Communication via SENDDP/RCVDP or SENDS7/RCVS7(S7-300, S7-400) instructions: 202
9.1.1 Overview 202
9.1.2 Safety-related IO controller-IO controller communication 205
9.1.2.1 Configure safety-related IO controller-IO controller communication 205
9.1.2.2 Program safety-related IO controller-IO controller communication 209
9.1.2.3 Safety-related IO controller-IO controller communication - Limits for data transfer 212
9.1.3 Safety-related master-master communication 212
9.1.3.1 Configure safety-related master-master communication 212
9.1.3.2 Program safety-related master-master communication 217
9.1.3.3 Safety-related master-master communication:Limits for data transfer 220
9.1.4 Safety-related IO controller-I-device communication 220
9.1.4.1 Configuring safety-related communication between IO controller and I-device 220
9.1.4.2 Programming safety-related IO controller I-device communication 223
9.1.4.3 Safety-related IO-Controller-IO-Device communication - Limits for data transfer 225
9.1.5 Safety-related master-I-slave communication 226
9.1.5.1 Configuring safety-related master-I-slave communication 226
9.1.5.2 Program the safety-related master-I-slave or I-slave-I-slave communication 228
9.1.5.3 Limits for data transfer of safety-related master-I-slave or I-slave-I-slave communication 230
9.1.6 Safety-related I-slave-I-slave communication 231
9.1.6.1 Configure safety-related I-slave-I-slave communication 231
9.1.6.2 Programming safety-related I-slave-I-slave communication 235
9.1.6.3 Limits for data transfer of safety-related I-slave-I-slave communication 236
9.1.7 Safety-related IO controller-I-slave communication 236
9.1.8 Safety-related communication via S7 connections 237
9.1.8.1 Configuring safety-related communication via S7 connections 237
9.1.8.2 Programming safety-related communication via S7 connections 238
9.1.8.3 Safety-related communication via S7 connections - Limits of data transfer 242
9.1.9 Safety-related communication with other S7 F-systems 243
9.1.9.1 Introduction 243
9.1.9.2 Communication with S7 Distributed Safety via PN/PN coupler (IO controller-IO controller communication) 243
9.1.9.3 Communication with S7 Distributed Safety via DP/DP coupler (master-master communication) 244
9.1.9.4 Communication with S7 Distributed Safety via S7 connections 245
9.1.9.5 Communication with S7 F/FH Systems via S7 connections 247

9.2 Communication via SENDDP/RCVDP instructions (S7-1200, S7-1500): 248
9.2.1 Overview 248
9.2.2 Safety-related IO controller-IO controller communication 251
9.2.2.1 Configure safety-related IO controller-IO controller communication 251
9.2.2.2 Program safety-related IO controller-IO controller communication 254
9.2.2.3 Safety-related IO controller-IO controller communication - Limits for data transfer 259
9.2.3 Safety-related master-master communication 259
9.2.3.1 Configure safety-related master-master communication 259
9.2.3.2 Program safety-related master-master communication 262
9.2.3.3 Safety-related master-master communication:Limits for data transfer 266

9.2.4 Safety-related IO controller-I-device communication ... 267
9.2.4.1 Configuring safety-related communication between IO controller and I-device ... 267
9.2.4.2 Programming safety-related IO controller I-device communication ... 271
9.2.4.3 Safety-related IO-Controller-IO-Device communication - Limits for data transfer ... 274
9.2.5 Safety-related master-I-slave communication ... 275
9.2.5.1 Configuring safety-related master-I-slave communication ... 275
9.2.5.2 Programming safety-related master-I-slave communication ... 278
9.2.5.3 Limits for data transfer of safety-related master-I-slave communication ... 280
9.2.6 Safety-related IO controller-I-slave communication ... 281
9.2.7 Safety-related communication to S7 F-System S7 Distributed Safety ... 282
9.2.7.1 Communication with S7 Distributed Safety via PN/PN coupler (IO controller-IO controller communication) ... 282
9.2.7.2 Communication with S7 Distributed Safety via DP/DP coupler (master-master communication) ... 283
9.3 Communication via SENDDP/RCVDP instructions between S7-300/400 and S7-1200/1500 F-CPUs ... 283
9.4 Communication via SENDDP/RCVDP instructions: IO controller and I-device in different projects ... 284
9.4.1 Configuring safety-related communication between IO controller and I-device ... 284
9.4.2 Programming safety-related IO Controller I-device communication ... 285
9.5 Communication with OPC UA Safety (S7-1500) ... 286
9.5.1 Configuration and programming interfaces ... 287
9.5.1.1 Safety Parameter Interface (SPI) of the SafetyProvider ... 287
9.5.1.2 Safety Application Program Interface (SAPI) of the SafetyProvider ... 291
9.5.1.3 Safety Parameter Interface (SPI) of the SafetyConsumer ... 295
9.5.1.4 Safety Application Program Interface (SAPI) of the SafetyConsumer in the F-OPC UA Consumer DB ... 299
9.5.1.5 Other variables in the F-OPC UA Consumer DB ... 304
9.5.2 Configuring and programming a SafetyProvider ... 307
9.5.2.1 Preparing the Safety program (or standard user program) for creation of an OPC UA SafetyProvider information model ... 307
9.5.2.2 Creating an OPC UA SafetyProvider information model for an individual data structure ... 308
9.5.2.3 Creating a server interface for an OPC UA SafetyProvider information model ... 313
9.5.2.4 Programming the initialization or evaluation of SAPI parameters in your safety program (or standard user program) ... 317
9.5.3 Configuring and programming a SafetyConsumer ... 320
9.5.3.1 Creating a client interface for an OPC UA SafetyProvider information model ... 320
9.5.3.2 Configuring properties of a SafetyConsumer ... 324
9.5.3.3 Programming the initialization or evaluation of SAPI parameters in your Safety program (or standard user program) ... 326
9.5.3.4 Programming underlying OPC UA communication ("client instructions") using the F-OPC UA Consumer Com FB ... 328
9.5.4 Special use cases for communication with OPC UA Safety ... 330
9.5.4.1 Introduction ... 330
9.5.4.2 Safety Multicast ... 330
9.5.4.3 Series machines ... 331
9.5.4.4 Changing communication partners ... 332
9.5.5 Avoiding connection resource bottlenecks ... 334
9.5.6 Using diagnostic options when communicating with OPC UA Safety ... 335
9.5.6.1 ReadSafetyDiagnostics ... 335
9.5.6.2 SafetyConsumerType ... 335
9.5.6.2.1 Configuring and programming a SafetyConsumerType ... 335

9.5.6.2.2 Creating an OPC UA SafetyConsumerType information model ... 336
9.5.6.2.3 Creating a server interface for an OPC UA SafetyConsumerType information model ... 340
9.6 Communication with Flexible F-Link (S7-1200, S7-1500) ... 342
9.6.1 Flexible F-Link ... 342
9.6.2 F-communication DB (S7-1200, S7-1500) ... 346

10 Compiling and commissioning a safety program ... 349

10.1 Compiling the safety program ... 349
10.2 Safety program work memory requirements (S7-300, S7-400) ... 349
10.3 Downloading project data ... 350
10.3.1 Downloading project data to an S7-300/400 F-CPU ... 356
10.3.1.1 Downloading project data to an S7-300/400 F-CPU with memory card inserted (SIMATIC Micro memory card or flash card) ... 356
10.3.1.2 Downloading project data to an S7-400 F-CPU without flash card inserted ... 356
10.3.1.3 Downloading individual F-blocks to an S7-300/400 F-CPU ... 357
10.3.1.4 Downloading project data to a memory card and inserting at an F-CPU S7-300/400 ... 358
10.3.2 Downloading project data into an F-CPU S7-1200 ... 359
10.3.2.1 Downloading project data to an S7-1200 F-CPU without program card inserted ... 359
10.3.2.2 Downloading project data to an S7-1200 F-CPU with program card inserted ... 360
10.3.2.3 Loading project data of an F-CPU S7-1200 with a transfer card into the internal load memory ... 363
10.3.2.4 Downloading project data of an S7-1200 F-CPU from the internal load memory to an empty SIMATIC Memory Card ... 366
10.3.2.5 Downloading project data to a SIMATIC Memory Card and inserting it at an F-CPU S7-1200 ... 368
10.3.3 Downloading project data to an S7-1500 F-CPU ... 369
10.3.3.1 Downloading project data to an S7-1500 F-CPU ... 369
10.3.3.2 Downloading project data into a redundant S7-1500HF system ... 369
10.3.3.3 Downloading project data to an S7-1500 F Software Controller ... 370
10.3.3.4 Downloading project data to a SIMATIC Memory Card or removable storage device and inserting it into an F-CPU S7-1500 ... 372
10.3.4 Restoring a backup of the safety program to an F-CPU ... 374
10.3.5 Specifics on creating and installing an image file of an S7-1500 F Software Controller ... 375
10.3.6 Loading project data (including safety-related project data) from an F-CPU into a PG/PC (S7-1500/S7-1200 G2 V4.1) ... 377
10.3.7 Loading project data (including safety-related project data) from a memory card into a PG/PC (S7-1500/S7-1200 G2 V4.1) ... 378
10.3.8 Loading a PC system configuration file for S7-1500 F software controllers ... 379
10.3.8.1 Creating a configuration file ... 379
10.3.8.2 Importing the configuration file ... 380
10.3.8.2.1 Import via the PC Station Panel menu (only possible with firmware < V30.0) ... 380
10.3.8.2.2 Import by means of a script ... 381
10.4 Program identification ... 384
10.5 Comparing Safety Programs ... 386
10.6 Creating a safety summary ... 388
10.7 Testing the safety program ... 391
10.7.1 Overview of Testing the Safety Program ... 391
10.7.2 Disabled safety mode ... 392

10.7.2.1 Configuring the time limit for disabled safety mode (S7-1200, S7-1500) ... 392
10.7.2.2 Disabling safety mode ... 393
10.7.3 Testing the safety program ... 397
10.7.4 Testing the Safety program with S7-PLCSIM/S7-PLCSIM Advanced ... 399
10.7.5 Changing the safety program in RUN mode (S7-300, S7-400) ... 404
10.7.6 Safety program in RUN mode (S7-1200, S7-1500) ... 406
10.8 F-change history ... 412

11 System acceptance ... 413
11.1 Overview of System Acceptance ... 413
11.2 Correctness of the safety program including hardware configuration (including testing) ... 414
11.3 Completeness of the safety summary ... 415
11.4 Compliance of the system library elements used in the safety program with Annex 1 of the Report for the TÜV certificate ... 416
11.5 Compliance of the know-how protected F-blocks used in the Safety program with their safety summary. ... 417
11.6 Completeness and correctness of the hardware configuration ... 418
11.7 Completeness and correctness of safety-related communication ... 424
11.7.1 Completeness and correctness of safety-related communication ... 425
11.7.2 Procedure to check the correctness of the configuration of communication with OPC UA Safety ... 426
11.8 Identity of online and offline program ... 430
11.9 Other characteristics ... 431
11.10 Acceptance of Changes ... 433

12 Operation and maintenance ... 441
12.1 Notes on startup of the F-system ... 441
12.2 Notes on Safety Mode of the Safety Program ... 441
12.3 Replacing Software and Hardware Components ... 444
12.4 PFDavg and PFH values ... 446
12.5 Guide to Diagnostics ... 448

13 STEP 7 Safety V21 instructions ... 450
13.1 Connection of the Enable input EN and Enable output ENO ... 450
13.2 Temporal blurring for instructions with time processing ... 450
13.3 General ... 452
13.3.1 LAD ... 452
13.3.1.1 New network (STEP 7 Safety V21) ... 452
13.3.1.2 Empty box (STEP 7 Safety V21) ... 453
13.3.1.3 Open branch (STEP 7 Safety V21) ... 454
13.3.1.4 Close branch (STEP 7 Safety V21) ... 454
13.3.2 FBD ... 455
13.3.2.1 New network (STEP 7 Safety V21) ... 455
13.3.2.2 Empty box (STEP 7 Safety V21) ... 456

13.3.2.3 Open branch (STEP 7 Safety V21) ... 456
13.3.2.4 Insert binary input (STEP 7 Safety V21) ... 457
13.3.2.5 Invert RLO (STEP 7 Safety V21) ... 458

13.4 Bit logic operations ... 459
13.4.1 LAD ... 459
13.4.1.1 ---| |---: Normally open contact (STEP 7 Safety V21) ... 459
13.4.1.2 ---| / |---: Normally closed contact (STEP 7 Safety V21) ... 460
13.4.1.3 --|NOT|--: Invert RLO (STEP 7 Safety V21) ... 460
13.4.1.4 ---( )---: Assignment (STEP 7 Safety V21) ... 461
13.4.1.5 ---(R)---: Reset output (STEP 7 Safety V21) ... 462
13.4.1.6 ---(S)---: Set output (STEP 7 Safety V21) ... 463
13.4.1.7 SR: Set/reset flip-flop (STEP 7 Safety V21) ... 464
13.4.1.8 RS: Reset/set flip-flop (STEP 7 Safety V21) ... 466
13.4.1.9 --|P|--: Scan operand for positive signal edge (STEP 7 Safety V21) ... 467
13.4.1.10 --|N|--: Scan operand for negative signal edge (STEP 7 Safety V21) ... 468
13.4.1.11 P_TRIG: Scan RLO for positive signal edge (STEP 7 Safety V21) ... 470
13.4.1.12 N_TRIG: Scan RLO for negative signal edge (STEP 7 Safety V21) ... 471
13.4.2 FBD ... 472
13.4.2.1 AND logic operation (STEP 7 Safety V21) ... 472
13.4.2.2 OR logic operation (STEP 7 Safety V21) ... 473
13.4.2.3 X: EXCLUSIVE OR logic operation (STEP 7 Safety V21) ... 474
13.4.2.4 =: Assignment (STEP 7 Safety V21) ... 476
13.4.2.5 R: Reset output (STEP 7 Safety V21) ... 476
13.4.2.6 S: Set output (STEP 7 Safety V21) ... 478
13.4.2.7 SR: Set/reset flip-flop (STEP 7 Safety V21) ... 479
13.4.2.8 RS: Reset/set flip-flop (STEP 7 Safety V21) ... 480
13.4.2.9 P: Scan operand for positive signal edge (STEP 7 Safety V21) ... 481
13.4.2.10 N: Scan operand for negative signal edge (STEP 7 Safety V21) ... 483
13.4.2.11 P_TRIG: Scan RLO for positive signal edge (STEP 7 Safety V21) ... 484
13.4.2.12 N_TRIG: Scan RLO for negative signal edge (STEP 7 Safety V21) ... 485

13.5 Safety functions ... 486
13.5.1 ESTOP1: Emergency STOP/emergency OFF up to stop category 1 (STEP 7 Safety V21) ... 486
13.5.2 TWO_HAND: Two-hand monitoring (STEP 7 Safety Advanced V21) (S7-300, S7-400) ... 490
13.5.3 TWO_H_EN: Two-hand monitoring with enable (STEP 7 Safety V21) ... 492
13.5.4 MUTING: Muting (STEP 7 Safety Advanced V21) (S7-300, S7-400) ... 495
13.5.5 MUT_P: Parallel muting (STEP 7 Safety V21) ... 504
13.5.6 EV1oo2DI: 1oo2 evaluation with discrepancy analysis (STEP 7 Safety V21) ... 515
13.5.7 FDBACK: Feedback monitoring (STEP 7 Safety V21) ... 520
13.5.8 SFDOOR: Safety door monitoring (STEP 7 Safety V21) ... 525
13.5.9 ACK_GL: Global acknowledgment of all F-I/O in an F-runtime group (STEP 7 Safety V21) ... 530

13.6 Timer operations ... 532
13.6.1 TP: Generate pulse (STEP 7 Safety V21) ... 532
13.6.2 TON: Generate on-delay (STEP 7 Safety V21) ... 535
13.6.3 TOF: Generate off-delay (STEP 7 Safety V21) ... 537

13.7 Counter operations ... 540
13.7.1 CTU: Count up (STEP 7 Safety V21) ... 540
13.7.2 CTD: Count down (STEP 7 Safety V21) ... 542
13.7.3 CTUD: Count up and down (STEP 7 Safety V21) ... 544

13.8 Comparator operations ... 546

13.8.1 CMP ==: Equal (STEP 7 Safety V21) 546
13.8.2 CMP <>: Not equal (STEP 7 Safety V21) 547
13.8.3 CMP >=: Greater or equal (STEP 7 Safety V21) 548
13.8.4 CMP <=: Less or equal (STEP 7 Safety V21) 550
13.8.5 CMP >: Greater than (STEP 7 Safety V21) 551
13.8.6 CMP <: Less than (STEP 7 Safety V21) 552

13.9 Math functions 553
13.9.1 ADD: Add (STEP 7 Safety V21) 553
13.9.2 SUB: Subtract (STEP 7 Safety V21) 556
13.9.3 MUL: Multiply (STEP 7 Safety V21) 559
13.9.4 DIV: Divide (STEP 7 Safety V21) 562
13.9.5 NEG: Create twos complement (STEP 7 Safety V21) 565
13.9.6 ABS: Form absolute value (STEP 7 Safety V21) (S7-1200, S7-1500) 568

13.10 Move operations 570
13.10.1 MOVE: Move value (STEP 7 Safety V21) 570
13.10.2 RD_ARRAY_I: Read value from INT F-array (STEP 7 Safety V21) (S7-1200 G2, S7-1500) 571
13.10.3 RD_ARRAY_DI: Read value from DINT F-array (STEP 7 Safety V21) (S7-1200 G2, S7-1500) 573
13.10.4 WR_FDB: Write value indirectly to an F-DB (STEP 7 Safety Advanced V21) (S7-300, S7-400) 575
13.10.5 RD_FDB: Read value indirectly from an F-DB (STEP 7 Safety Advanced V21) (S7-300, S7-400) 577

13.11 Conversion operations 578
13.11.1 CONVERT: Convert value (STEP 7 Safety V21) 578
13.11.2 BO_W: Convert 16 data elements of data type BOOL to a data element of data type WORD (STEP 7 Safety V21) 580
13.11.3 W_BO: Convert a data element of data type WORD to 16 data elements of data type BOOL (STEP 7 Safety V21) 582
13.11.4 SCALE: Scale (STEP 7 Safety V21) 584
13.11.5 SCALE_D: Scale to data type DINT (STEP 7 Safety V21) (S7-1200, S7-1500) 586

13.12 Program control operations 588
13.12.1 JMP: Jump if RLO = 1 (STEP 7 Safety V21) 588
13.12.2 JMPN: Jump if RLO = 0 (STEP 7 Safety V21) 590
13.12.3 LABEL: Jump label (STEP 7 Safety V21) 592
13.12.4 RET: Return (STEP 7 Safety V21) 594
13.12.5 OPN: Open global data block (STEP 7 Safety Advanced V21) (S7-300, S7-400) 595

13.13 Word logic operations 596
13.13.1 AND: AND logic operation (STEP 7 Safety V21) 596
13.13.2 OR: OR logic operation (STEP 7 Safety V21) 597
13.13.3 XOR: EXCLUSIVE OR logic operation (STEP 7 Safety V21) 598

13.14 Shift and rotate 600
13.14.1 SHR: Shift right (STEP 7 Safety V21) 600
13.14.2 SHL: Shift left (STEP 7 Safety V21) 602

13.15 Operating 604
13.15.1 ACK_OP: Fail-safe acknowledgment (STEP 7 Safety V21) 604

13.16 Additional instructions 609
13.16.1 LAD 609
13.16.1.1 ---| |--- OV: Get status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) 609
13.16.1.2 ---| / |--- OV: Get negated status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) 610

13.16.2 FBD ... 611
13.16.2.1 OV: Get status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) ... 611
13.17 Communication ... 612
13.17.1 PROFIBUS/PROFINET ... 612
13.17.1.1 SENDDP and RCVDP: Send and receive data via PROFIBUS DP/PROFINET IO (STEP 7 Safety V21) ... 612
13.17.2 S7 communication ... 622
13.17.2.1 SENDS7 and RCVS7: Communication via S7 connections (STEP 7 Safety Advanced V21) (S7-300, S7-400) ... 622

A Monitoring and response times ... 629

A.1 Configuring monitoring times ... 630
A.1.1 Minimum monitoring time for the F-runtime group cycle time ... 632
A.1.2 Minimum monitoring time of the F-I/O access ... 632
A.1.3 Minimum monitoring time for communication via the SENDDP/RCVDP or SENDS7/RCVS7 instructions ... 633
A.1.4 Minimum monitoring time of communication with OPC UA Safety ... 634
A.1.5 Minimum monitoring time for communication with Flexible F-Link ... 634
A.1.6 Monitoring Time for Safety-Related Communication between F-Runtime Groups ... 634
A.2 Response Times of Safety Functions ... 635

B Checklist ... 638

C Assignment of the F-BaseID during runtime (delayed transfer) (S7-1200 G2, S7-1500) ... 643

C.1 Safety Extensions ... 645
C.1.1 BASEID_GEN ... 645
C.1.2 BASEID_DATA ... 646
C.1.3 BASEID_CODING ... 647
C.1.4 BASEID_HOST ... 649

Glossary ... 651

Index ... 665

Product Overview

1

1.1 Overview

SIMATIC Safety fail-safe system

The SIMATIC Safety fail-safe system is available to implement safety concepts in the area of machine and personnel protection (for example, for emergency STOP devices for machining and processing equipment) and in the process industry (for example, for implementation of protection functions for safety devices of instrumentation and controls and of burners).

⚠ WARNING

The SIMATIC Safety F-system is used to control processes that reach their safe state immediately through shutdown. This also applies in particular to redundant S7-1500HF systems.

SIMATIC Safety may only be used for controlling processes in which a shutdown does not pose a danger to humans or the environment.

When realizing safety applications including the creation of the safety-related project data you have to take into consideration the standards, directives and guidelines relevant for your application. In particular include standards in which the software creation process is described (for example IEC 61508-3 or ISO 13849-1). *(S062)*

Achievable safety requirements

SIMATIC Safety F-systems can satisfy the following safety requirements:

- Safety Integrity Level SIL3 in accordance with IEC 61508:2010
- Performance Level (PL) e and category 4 in accordance with ISO 13849-1:2015 or EN ISO 13849-1:2015

Functional safety in SIMATIC Safety

Functional safety is implemented primarily in the software. The SIMATIC Safety F-System brings the plant into a safe state in the event of a hazardous event and maintains this state. Safety mechanisms are contained mainly in the following components:

- In the safety-related user program (safety program) in the F-CPU
- In the fail-safe inputs and outputs (F-I/O)

The F-I/O ensure the safe processing of field information (sensors: e.g. emergency STOP pushbutton, light barriers; actuators, e.g. for motor control). They have all of the required hardware and software components for safe processing, in accordance with the required

Safety Integrity Level. The user only has to program the user safety function. Functional safety for the process can be provided by a user safety function or a fault reaction function. The F-system cyclically checks the integrity of the safety program and the safety-related data. This check is generally always performed when safety-related data leaves an F-runtime group via safety-related communication or F-I/O access. If this check detects a fault, the F-system executes the fault reaction function, which switches off the associated outputs and, if necessary, sets the F-CPU to STOP.

NOTE

If the F-CPU switches to STOP mode due to a fault reaction function, with a redundant S7-1500HF system, it leads to the STOP of the primary CPU **and** the backup CPU.

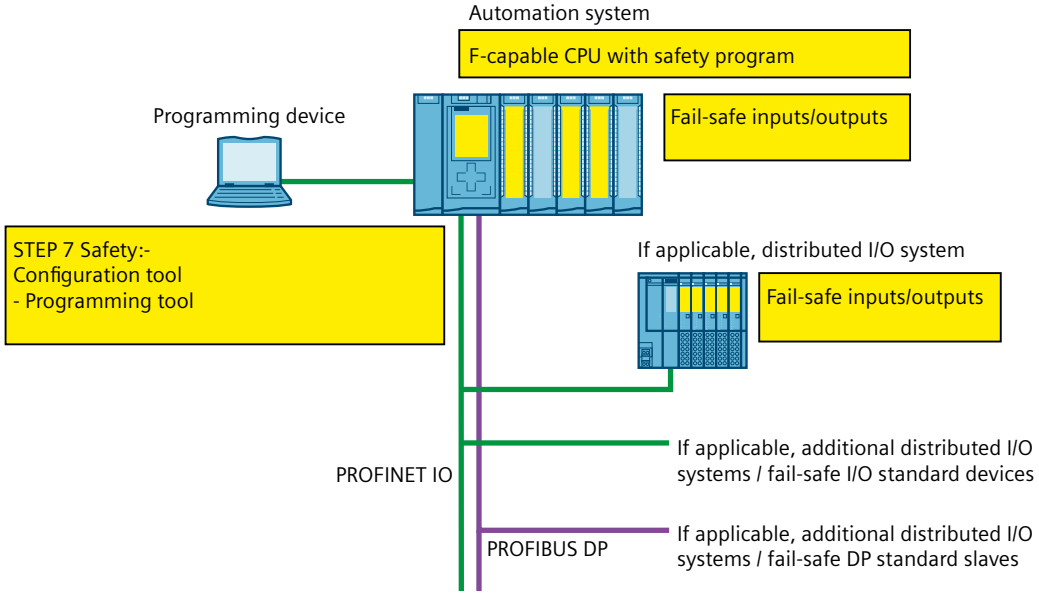
Example of user safety function and fault reaction function

In the event of overpressure, the F-system will open a valve (user safety function). In the event of a hazardous fault in the F-CPU, all outputs are deactivated (fault reaction function), whereby the valve is opened, and the other actuators also attain a safe state. For a non-faulty F-system, only the valve would be opened.

1.2 Hardware and Software Components

Hardware and software components of SIMATIC Safety

The following schematic diagram provides an overview of the hardware and software components required to configure and operate a SIMATIC Safety F-system.



Hardware components for PROFIBUS DP

You can use the following fail-safe components in SIMATIC Safety F-systems on PROFIBUS DP:

- F-CPUs with DP interface, such as CPU 1516F-3 PN/DP
- SIMATIC Drive Controller, for example, CPU 1504D TF
- Fail-safe inputs and outputs (F-I/O), such as:
  - S7-300 fail-safe signal modules in ET 200M
  - S7-1500/ET 200MP fail-safe modules
  - ET 200SP fail-safe modules
  - ET 200S fail-safe modules
  - ET 200pro fail-safe modules
  - ET 200iSP fail-safe modules
  - Fail-safe I/O modules ET 200eco (S7-300, S7-400)
  - Fail-safe GSD based DP slaves (light array, laser scanner, etc.)

You have the option to expand the configuration with standard I/O.

The following CPs/CMs can be used in a SIMATIC Safety F-system on PROFIBUS DP for connection to distributed F-I/O:

- CP 443-5 Extended
- CM 1243-5
- CM 1542-5
- CP 1542-5
- SP CM DP

Hardware components for PROFINET IO

You can use the following fail-safe components in SIMATIC Safety F-systems on PROFINET IO:

- F-CPUs with PN interface, e.g. CPU 1214FC DC/DC/DC
- SIMATIC Drive Controller, for example, CPU 1504D TF
- Fail-safe inputs and outputs (F-I/O), such as:
  - S7-300 fail-safe signal modules in ET 200M
  - S7-1500/ET 200MP fail-safe modules
  - ET 200SP fail-safe modules
  - ET 200SP HA fail-safe modules
  - ET 200S fail-safe modules
  - ET 200pro fail-safe modules
  - ET 200eco PN fail-safe I/O modules
  - ET 200AL fail-safe I/O modules
  - Fail-safe GSD based I/O devices (light array, laser scanner, etc.)

You have the option to expand the configuration with standard I/O.

The following CPs/CMs can be used in a SIMATIC Safety F-system on PROFINET IO for connection to distributed F-I/O:

- CP 443-1
- CP 443-1 Advanced-IT
- CM 1542-1

Hardware components for central configuration

You can use the following fail-safe components in SIMATIC Safety F-systems centrally on an F-CPU (not HF-CPU):

- S7-300/ET 200M fail-safe signal modules
- S7-1200 fail-safe modules
- S7-1500/ET 200MP fail-safe modules
- ET 200SP fail-safe modules
- ET 200S fail-safe modules
- ET 200pro fail-safe modules (can also be used with CPU 1516proF-2)

You have the option to expand the configuration with standard I/O.

STEP 7 Safety

This documentation describes *STEP 7 Safety Advanced V21*. *STEP 7 Safety* is the configuration and programming software for the SIMATIC Safety F-system. With *STEP 7 Safety*, you receive the following:

- The support for configuring the F-I/O in the *hardware and network editor* of TIA Portal
- The support for creating the Safety program using LAD and FBD and integrating error detection functions into the Safety program
- Instructions for programming your Safety program in LAD and FBD, which you are familiar with from the standard user programs
- Instructions for programming your Safety program in LAD and FBD with special safety functions

Moreover, *STEP 7 Safety* offers functions for comparing Safety programs and for assisting you with the system acceptance.

⚠ WARNING

The configuration of F-CPUs and F-I/Os as well as the programming of F-blocks must be carried out in TIA Portal as described in this documentation. To ensure safe operation with the SIMATIC Safety system, you must comply with the requirements of the relevant standards, such as IEC 61508. The section System acceptance (Page 413) describes procedures and tools of STEP 7 Safety to meet these requirements. *(S056)*

Optional packages

In addition to *STEP 7 Safety*, you can use option packages with F-I/O and F-CPUs and use instructions for programming your Safety program with special safety functions within the SIMATIC Safety F-system, for example SINUMERIK.

The installation, parameter assignment and programming as well as what is important to note during system acceptance, is described in the documentation for the specific optional packages.

Also read the notes in "Configurations supported by the SIMATIC Safety F-system (Page 64)".

TIA Portal Cloud Connector

⚠ WARNING

Use of the TIA Portal Cloud Connector is only intended for engineering work with TIA Portal. This means online access in productive operation (e.g. SCADA) is not permitted. This is particularly true for Safety programs. *(S068)*

Openness

Openness (https://support.industry.siemens.com/cs/ww/en/view/109826886) as part of *STEP 7 Safety* is supported with the functions listed below. The use of Openness in productive operation is not permitted.

As part of *STEP 7 Safety* the following is supported:

- Inserting / removing F-CPUs and F-I/Os
- Copying / deleting F-CPUs and F-I/Os from templates
- Compiling hardware
- Compiling software (incl. Safety program)
- Handling access protection for the safety-related project data (for TIA projects not protected by project protection)
- Reading/configuring F-parameters of the F-CPU (for S7-1500 F-CPUs)
- Reading/configuring F-parameters of F-I/Os
- Reading/configuring i-parameters of F-I/Os (for ET 200SP, ET 200AL, ET 200MP, ET 200eco PN, ET 200pro)
- Downloading safety-related project data to a SIMATIC Memory Card or storing it in a file folder
- Reading, declaring or deleting fail-safe tags in the PLC tag table
- Updating projects to the latest type versions of F-blocks
- Consistent station upload
- Export and import of F-blocks and F-compliant PLC data types (UDT)
- Comparison of hardware and software

- Version control interface (VCI) support
- Reading out PLC online fingerprint for the Safety program
- Storing safety-related project data as a PC System Configuration file
- Reading out the F-signatures and block signatures
- Adding and removing OPC UA Safety configurations (SafetyProvider/SafetyConsumer)

The following are not supported.

- Downloading to device

⚠ WARNING

When using Openness or other tools for operating the TIA Portal in combination with safety-related project data, integrity of the data must be ensured (for example, as part of the save or transfer functions of "Source Code Management" applications). If there is a connection of external tools, the requirements for offline support tools according to IEC 61508-3 must be observed. A violation of the integrity of the safety-related project data either offline or online cannot be detected by *STEP 7 Safety*. A final verification of the correctness of the safety-related project data has to be carried out as described in the section System acceptance (Page 413). *(S070)*

Openness programs can also be integrated into TIA Portal as an add-in. It is also possible to include so-called workflow add-ins, which are automatically called before a Safety program is compiled. See "Extending workflows" in the standard help.
If you have integrated such a workflow add-in, this is output in the safety summary.

Openness service

The Openness interface has been expanded to include the following services:

- `GlobalSettings` (see name area **Siemens.Engineering.Safety**), which provides the following actions:
  - `SafetyModificationsPossible(bool safetyModificationsPossible)`
  - `UsernameForFChangeHistory(string userName)`
  - `bool SafetyModificationsPossible()`
  - `string UsernameForFChangeHistory()`
- `SafetyBaseIdProvider` with the following action in the name area **Siemens.Engineering.Safety**:
  - `void GenerateBaseId()`

  `SafetySignatureProvider` which provides the following actions and properties:
  - `SafetySignatureComposition Signatures`
  - `UInt64 SafetySignature.Value`
  - `SafetySignatureType SafetySignature.Type`

- `SafetyAdministration` with the following actions and properties in the name area **Siemens.Engineering.Safety:**
  - `bool IsSafetyOfflineProgramPasswordSet`
  - `void SetSafetyOfflineProgramPassword(SecureString newPassword)`
  - `void RevokeSafetyOfflineProgramPassword(SecureString currentPassword)`
  - `bool IsLoggedOnToSafetyOfflineProgram`
  - `void LoginToSafetyOfflineProgram(SecureString currentPassword)`
  - `void LogoffFromSafetyOfflineProgram()`
- `Safety Administration`, which can be accessed from the `PLc-DeviceItem`.

  The `SafetyAdministration` service offers the following properties in the **Siemens.Engineering.Safety** namespace. These three properties include additional properties that are listed in the description.
  - `RuntimegroupComposition RuntimeGroups { get; }`
  - `SafetySettings Settings { get; }`
  - `SafetySignatureProvider ProgramSignatures { get; }`
- `SafetyPrintout` with the following action in the name area **Siemens.Engineering.Safety**:
  - `bool Print(SafetyPrintoutFilePrinter filePrinter, FileInfo fullOutputPath, string documentLayout, SafetyPrintoutOption documentationOption);`

More information can be found in the System Manual "SIMATIC TIA Portal Openness: Automating creation of projects (https://support.industry.siemens.com/cs/ww/en/view/109826886)" in the section "F-specific openness".

Virtual environments

⚠ WARNING

Use of virtual environments in the engineering system

Note that a hypervisor or a client software of a hypervisor is not permitted to perform any function that reproduces recorded message frame sequences with correct time response in a network with real F-CPUs and F-I/Os.

Make sure that this condition is met, for example, when using the following functions:

- Reset of captured statuses (snapshots) of the virtual machines (VMs)
- Suspending and resuming the VMs
- Replay of recorded sequences in the VMs
- Moving VMs between hosts in productive operation (e.g. Fault Tolerance (FT))
- Digital twins of VMs in the virtual environment

If in doubt, disable these functions in the settings (Hypervisor administration console). *(S067)*

Operating system installations with S7-150xS(P) F

⚠ WARNING

When using a PC with multiple Windows or Linux operating system installations (e.g. via boot manager), only one S7-150xS(P) F may be installed on this PC. *(S095)*

Safety program

You can create a Safety program using the *program editor*. You can program fail-safe FBs and FCs in the FBD or LAD programming languages using the instructions from *STEP 7 Safety* and create fail-safe DBs.

Safety checks are automatically performed and additional fail-safe blocks for error detection and fault reaction are inserted when the Safety program is compiled. This ensures that failures and errors are detected and appropriate reactions are triggered to maintain the F-system in the safe state or bring it to a safe state.

In addition to the Safety program, a standard user program can be run on the F-CPU. A standard program can coexist with a Safety program in an F-CPU because unintentional influencing of the safety-related data of the Safety program is uncovered by the standard user program.

Data can be exchanged between the safety program and the standard user program in the F-CPU by means of bit memory or data of a standard DB or by accessing the process image input and output.

Note possible restrictions/particularities when using Software Units or a Safety Unit (Page 116).

See also

Data Transfer from the Safety Program to the Standard User Program (Page 198)

1.3 Installing/uninstalling the STEP 7 Safety Advanced V21 license

After installation of the *STEP 7 Safety Advanced V21* license, the range of functions of *STEP 7 Safety Advanced V21* is available to you. This license is not required for S7-1200 F-CPUs.

Software requirements for *STEP 7 Safety Advanced V21*

At a minimum, the following software package must be installed on the programming device or PC:

- *SIMATIC STEP 7 Professional V21*

Installing the *STEP 7 Safety Advanced V21* license

1. Start the *Automation License Manager* on the programming device/PC on which the *"SIMATIC STEP 7 Professional V21"* software package is installed.
2. Install the *STEP 7 Safety Advanced V21* license as described in the *Automation License Manager help*.

Uninstalling the *STEP 7 Safety Advanced V21* license

To uninstall the *STEP 7 Safety Advanced V21* license, proceed as described in the *Automation License Manager help*.

1.4 Migrating projects from S7 Distributed Safety V5.4 SP5 to STEP 7 Safety Advanced

Migration of *S7 Distributed Safety* projects to *STEP 7 Safety Advanced* is no longer supported as of *STEP 7 Safety V21*. Migration is still possible with earlier versions of the *TIA Portal*. The project can then be upgraded from an earlier version to the current version in the usual way.

1.5 Migrating PLC programs to a n F-CPU S7-1500

To migrate an F-CPU S7-300/400 onto an F-CPU S7-1500, proceed as with the migration of a standard CPU S7-300/400 onto a CPU S7-1500.

Points to note after migration:

- Non-automatable actions
  - Creating an F-runtime group and assigning it to the main safety block.
  - The hardware configuration including the I/O of the initial F-CPU is not automatically transferred to an S7-1500 F-CPU. Implement the hardware configuration of the new CPU manually after migration.

    Please also read the sections
    "Specify F-destination address for F-I/O of PROFIsafe address type 1" and "Specify F-destination address for F-I/O of PROFIsafe address type 2" in chapter "Configuring an F-CPU (Page 41)". Otherwise, this can lead to a reassignment of the F-destination addresses in the configuration.
  - When using F-I/O with PROFIsafe Protocol Version = Expanded Protocol (XP) (for example S7-1500/ET 200MP F-modules), keep in mind that you need one byte more in the address area of S7-1200/1500 F-CPUs than in S7-300/400 F-CPUs.
  - Replacement of the OV instruction by the connection of the ENO output for mathematical functions (Page 553).
  - Replacement of the RD_FDB instruction by the instructions RD_ARRAY_I (Page 571) and RD_ARRAY_DI (Page 573).
  - Replacement of the F-runtime group communication through Communication via Flexible F-Link (Page 146).

- Instructions not supported:
  - MUTING
  - TWO_HAND
  - WR_FDB
  - OPN
  - SENDS7
  - RCVS7
- Data types not supported
  - DWORD
- Changes to safety program programming
  - F_GLOBDB.VKE0/1 replaced by FALSE/TRUE (Page 119).
  - Readable values from the F_GLOBDB replaced by the F-runtime group information DB. Additional information is available under F-shared DB (S7-300, S7-400) (Page 154) and F-runtime group information DB (S7-1200, S7-1500) (Page 154).
  - Replacement of the QBAD_I_xx or QBAD_O_xx tag by the value status. Additional information is available under Value status (S7-1200, S7-1500) (Page 159) and F-I/O DB (Page 164).
- New naming convention when naming the F-I/O DBs
- Modified behavior of QBAD and PASS_OUT (Page 169) tags for F-I/O with "RIOforFA safety" profile.

Compile the safety program and eliminate any compilation errors displayed.

NOTE

A new acceptance must be carried out following F-CPU migration.

See also

Programming (Page 111)

1.6 Upgrading projects to the latest version of STEP 7 Safety

Re-create the safety summary after an upgrade to a newer version of *STEP 7 Safety*. The versions listed in the new safety summary of *STEP 7 Safety* will then match the new version of *STEP 7 Safety*.

1.6.1 Upgrading projects from STEP 7 Safety as of V14 SP1 to the latest version

If you want to continue working with a project from *STEP 7 Safety* as of *V14 SP1*, you must upgrade the project to the latest version of *STEP 7 Safety*.

NOTE

The project must be consistent before upgrading.

Perform the upgrade following the usual procedure for *STEP 7*. After upgrading to the latest version, you must compile your safety program and download it in STOP.

If you want to compile a migrated safety program with STEP 7 Safety for the first time, we recommend that you first update to the latest available version of the instruction. You will then avoid number conflicts.

If you have upgraded a project <= V17 to the latest version and compiled it, the Collective F-signature remains the same, but the version comparison between the online and offline programs shows a change. The differences shown are due to the improved calculation of the signatures as of V18. Because the Collective F-Signature has remained the same, re-acceptance is not necessary. We recommend creating a new safety summary so that the new signatures are documented.

Keep in mind that existing change histories are not upgraded. All previous entries are deleted after the upgrade. If required, print out the change log before you upgrade.

If you have created your project with *STEP 7 Safety V18* and have stored the Safety program in a Safety unit in this project, you must upgrade the project and then re-create the safety summary, which replaces the Safety summary created with *STEP 7 Safety V18*. This is required for the SENDDP and RCVDP instructions in the "System library elements used in Safety program" section to be displayed correctly.

1.7 First steps

Getting Started in SIMATIC Safety

Three Getting Started documents are available to help you begin using SIMATIC Safety.

The Getting Started documentation is an instruction manual that provides a step-by-step description of how to create a project with SIMATIC Safety. It gives you the opportunity to quickly become familiar with the scope of features of SIMATIC Safety.

Contents

The Getting Started documentation describes the creation of a single, continuous project that is extended with each section. Based on the configuration, you program a fail-safe shutdown, make changes to the programming, and accept the system.

In addition to the step-by-step instructions, the Getting Started documentation also gives you background information for every new topic, which explains the functions used in more detail and how they interrelate.

Target audience

The Getting Started documentation is intended for beginners but is also suitable for users who are switching from *S7 Distributed Safety*.

Download

Three Getting Started documents are available as PDF files for free download in the Industry Online Support:

- STEP 7 Safety Advanced V11 with S7-300/400 F-CPUs (https://support.industry.siemens.com/cs/ww/en/view/49972838)
- STEP 7 Safety Basic V13 SP1 with S7-1200 F-CPUs (https://support.industry.siemens.com/cs/ww/en/view/104547552) (part of the manual "S7-1200 Functional Safety manual")
- STEP 7 Safety Advanced V13 with S7-1500 F-CPUs (https://support.industry.siemens.com/cs/ww/en/view/101177693)

Configuring

2

2.1 Overview of Configuration

Introduction

You basically configure a SIMATIC Safety F-system in the same way as a standard S7-300, S7-400, S7-1200, S7-1500 or ET 200MP, ET 200SP, ET 200SP HA, ET 200AL, ET 200S, ET 200iSP, ET 200eco, ET 200eco PN or ET 200pro automation system in *STEP 7*.

This section presents only the essential differences compared to standard configuration you encounter when configuring a SIMATIC Safety F-system.

This documentation distinguishes between three groups of F-I/O:

F-I/Os of PROFIsafe address type 1

F-I/Os which ensure the uniqueness of the PROFIsafe address solely with the F-destination address, for example, ET 200S F-modules. The PROFIsafe address is usually assigned by DIP switches.

F-I/Os of PROFIsafe address type 2

F-I/O for which the uniqueness of the PROFIsafe address can be ensured through the combination of F-source address and F-destination address, for example, S7-1500/ET 200MP F-modules. You normally assign the PROFIsafe address with *STEP 7 Safety*.

F-I/O of PROFIsafe address type 3

F-I/O for which the uniqueness of the PROFIsafe address can be ensured through an F-BaseID in addition to the combination of F-source address and F-destination address, for example, certain S7-1500/ET 200SP F-modules. The F-BaseID saves you the work of ensuring the uniqueness of the PROFIsafe address network-wide/worldwide yourself (for more information, see "PROFIsafe addresses for F-I/O of PROFIsafe address type 3 (Page 70)"). You normally assign the PROFIsafe address (including the F-BaseID) with *STEP 7 Safety*.

NOTE

The terminology "PROFIsafe address type 3" used in this documentation is equivalent to the designation "PROFIsafe address type 2 with BaseID". This applies in TIA Portal as well as to F-I/Os.

Which F-components can you configure with *STEP 7 Basic/STEP 7 Professional* and *STEP 7 Safety Advanced*?

The table below shows you which F-CPUs you can configure with *STEP 7 Safety Advanced*:

F-CPUs	*STEP 7 Safety Advanced*
S7-300	x
S7-400	x
S7-1200	x *
S7-1500(H)	x
SIMATIC Drive Controller	x
S7-1500 F Software Controller	x
S7-1500 F Virtual Controller	x

* You can configure S7-1200 F-CPUs without a license for *STEP 7 Safety*.

The table below shows you which F-I/Os you can configure with *STEP 7 Safety Advanced* as well as which PROFIsafe address type they support:

F-I/O	*STEP 7 Safety Advanced*	PROFIsafe address type
S7-300 F-SMs	x**	1
ET 200S F-modules	x	1
ET 200pro F-modules	x	1
ET 200iSP F-modules	x	1
ET 200eco DP F-I/O modules	With S7-300/400 F-CPUs (only PROFIsafe V1 mode)	1
ET 200eco PN F-I/O modules	x	2
S7-1200 F-modules (centrally on S7-1200 F-CPUs)	x***	2
ET 200SP F-modules	x	2 or 3
ET 200SP HA F-modules	x	2
S7-1500/ET 200MP F-modules	x	2
ET 200AL F-modules	x	2
fail-safe GSD based DP slaves	x	*
fail-safe GSD based I/O devices	x	*

* Consult the respective documentation to determine the PROFIsafe address type of a GSD based DP slave/GSD based I/O device. If in doubt, assume that the PROFIsafe address is type 1.

** F-SMs that only support PROFIsafe V1 mode can only be used on F-CPUs S7-300/400.

*** You can configure S7-1200 F-modules without a license for *STEP 7 Safety*.

Special F-parameters

For the F-functionality there are special F-parameters that you can review and set in the "Properties" of the fail-safe components (F-CPU and F-I/O). F-parameters are marked in yellow.

F-parameters are explained in "Configuring an F-CPU (Page 41)" and "Configuring F-I/O (Page 53)".

Changing safety-related parameters

NOTE

If you change a safety-related parameter (marked in yellow) for an F-I/O or an F-CPU, you must then compile and download the modified hardware configuration and the Compiling the safety program (Page 349) (context menu "Compile > Hardware and software (only changes)"). This also applies to changes to F-I/O that are not used in the safety program. F-I/O in standard operation is not affected by this.

Compiling the hardware configuration

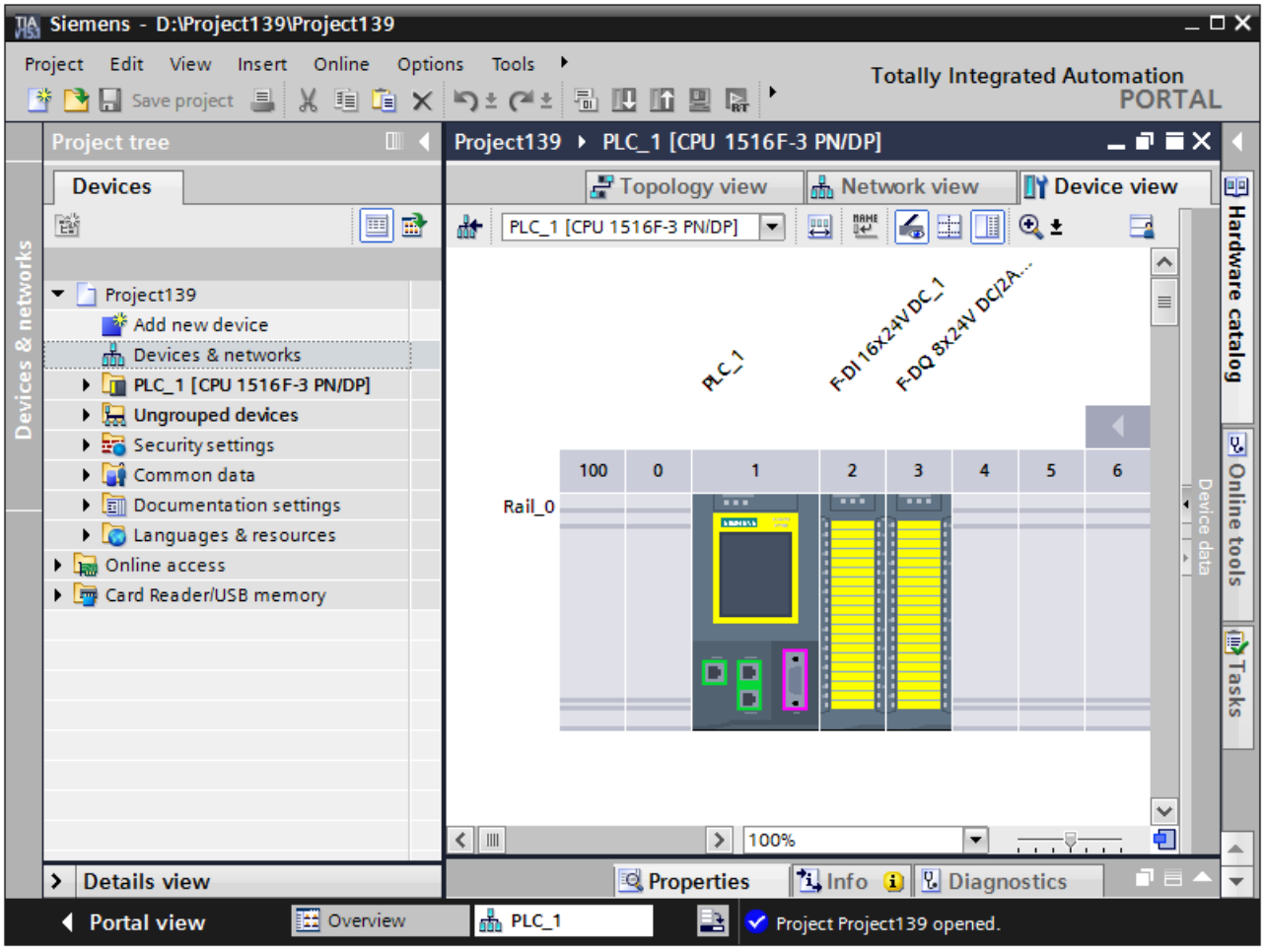
You must compile the hardware configuration of the SIMATIC Safety F-system (context menu "Compile > Hardware configuration"). A configured F-CPU with enabled F-capability is the only prerequisite for programming the safety program.

NOTE

Inconsistencies are possible when configuring the hardware and can also be saved. A full consistency check of the hardware configuration and possible connection data is performed only during compilation. Therefore, perform "Edit > Compile" regularly.

Example: Configured F-system in *STEP 7 Professional*

The following figure presents a configured F-system. You choose the fail-safe components in the "Hardware catalog" task card as you would do with standard components and place them in the work area of the network or device view. F-components are shown in yellow.



More information

For detailed information on F-I/O, refer to the *manuals for the relevant F-I/O*.

Which safety-related communication options can you configure?

You need to use the *hardware and network editor* to configure the following safety-related communication options (see Communication via SENDDP/RCVDP or SENDS7/RCVS7(S7-300, S7-400) instructions: (Page 202) or Communication via SENDDP/RCVDP instructions (S7-1200, S7-1500): (Page 248)):

- Communication with OPC UA Safety (Page 286)
- Communication with Flexible F-Link (Page 342)

- Safety-related master-master communication
- Safety-related master-master communication for *S7 Distributed Safety*
- Safety-related master-I-slave communication
- Safety-related I-slave-I-slave communication
- Safety-related I-slave-slave communication
- Safety-related IO controller-IO controller communication
- Safety-related IO controller-IO controller communication for *S7 Distributed Safety*
- Safety-related IO controller-I-device communication
- Safety-related IO controller-I-slave communication
- Safety-related communication via S7 connections
- Safety-related communication via S7 connections for *S7 Distributed Safety* or *S7 F Systems*

2.2 Peculiarities when configuring fail-safe GSD based DP slaves and fail-safe GSD based I/O devices

Requirement

In order to use fail-safe GSD based DP slaves for SIMATIC Safety, these GSD based slaves must be operated on PROFIBUS DP and support the PROFIsafe bus profile. When used in an S7-1200/1500 F-CPU, they must support the PROFIsafe bus profile in V2 mode.

Fail-safe GSD based DP slaves used in hybrid configurations on PROFIBUS DP and PROFINET IO downstream from a IE/PB link must support the PROFIsafe bus profile in V2 mode.

In order to use fail-safe GSD based I/O devices for SIMATIC Safety, the GSD based devices must be operated on PROFINET IO and support the PROFIsafe bus profile in V2 mode.

Configuration with GSD files

As is the case in a standard system, the basis for configuring fail-safe GSD based DP slaves/IO devices is the device specification in the GSD file (device master file).

A GSD file contains all of the properties of a GSD based DP slave or GSD based I/O device. For fail-safe GSD based DP slaves/GSD based I/O devices, certain parts are protected by a CRC.

The GSD files are supplied by the device manufacturers.

Protection of the data structure of the device in GSD files

The only GSD files supported are those that satisfy the requirements for protection defined as of *PROFIsafe Specification* V2.0 using a CRC stored in this file ("setpoint" for F_IO_StructureDescCRC).

The data structure described in the GSD file is checked when the F-I/O is added to the hardware configuration and when the hardware configuration is compiled. When an error is

detected, you should clarify whether the GSD file provided by the device manufacturer contains the setpoint for F_IO_StructureDescCRC.

Assignment and setting of the PROFIsafe address

⚠ WARNING

Check the documentation for your fail-safe GSD based DP slaves / fail-safe GSD based I/O devices to find out the valid PROFIsafe address type. If you do not find the necessary information, assume PROFIsafe address type 1. For configuring, proceed as described under PROFIsafe addresses for F-I/O of PROFIsafe address type 1 (Page 67), PROFIsafe addresses for F-I/O of PROFIsafe address type 2 (Page 68) or PROFIsafe addresses for F-I/O of PROFIsafe address type 3 (Page 70).

Set the F-source address for fail-safe GSD based DP slaves / fail-safe GSD based I/O devices according to the manufacturer's specifications. If the F-source address needs to correspond to the "Central F-source address" parameter of the F-CPU (PROFIsafe address type 2), you will find the latter in the "Properties" tab of the F-CPU. In this case, check the safety summary to verify that the value for the "Central F-source address" parameter corresponds to the value of the F-source address of the fail-safe GSD based DP slave/fail-safe GSD based I/O device. *(S053)*

Configuration procedure with GSD files

You import the GSD files to your project (see *Help on STEP 7 "GSD files"*).

1. Select the fail-safe GSD based DP slave / GSD based I/O device in the "Hardware catalog" task card and connect it to the relevant subnet in the network view.
2. Select the fail-safe GSD based DP slave/GSD based I/O device and insert the necessary F-modules, if this does not occur automatically.
3. Select the relevant F-module and open the "Properties" tab in the inspector window.

For fail-safe GSD based DP slaves/GSD based I/O devices (contrary to other F-I/O), the "Manual assignment of F-monitoring time" parameter is enabled. The result is that the value specified in the GSD file for the F-monitoring time is used as default value when the slaves/devices are plugged. You can change both values (time and type of assignment) later manually.

F-parameter "F_CRC_Seed" and "F_Passivation" for fail-safe GSD based I/O devices

The F-parameters "F_CRC_Seed" and "F_Passivation" influence the behavior of a fail-safe GSD based I/O device. The combination of the F-parameters cannot be set but is specified by selecting a corresponding F-module. Up to three F-module variants can be used, depending on the S7-300/400 or S7-1200/1500 F-CPU used.

F-module variant	F_CRC_Seed	F_Passivation	Behavior of the fail-safe GSD based I/O device	Can be used with F-CPU
1	Parameter does not exist	Parameter does not exist	The GSD based I/O device works with the Basic Protocol (BP) from PROFIsafe. The "RIOforFA-Safety" profile is not supported.	S7-300/400/1200/1-500*
2	CRC-Seed24/32	Device/module	The GSD based I/O device works with the Expanded Protocol (XP) from PROFIsafe. The "RIOforFA-Safety" profile is not supported.	S7-1200/1500
3	CRC-Seed24/32	Channel	The GSD based I/O device works with the Expanded Protocol (XP) from PROFIsafe. The "RIOforFA-Safety" profile is supported.	S7-1200/1500

* Only use the F-module variant 1 with S7-1200/1500 F-CPUs if neither F-module variant 2 nor 3 exists.

More information

You can find the description of the parameters in the Help on fail-safe GSD based DP slaves and GSD based I/O devices.

2.3 Configuring an F-CPU

Introduction

You configure the F-CPU basically in the same way as a standard automation system.

F-CPUs are always configurable in *STEP 7*, regardless of whether or not the *STEP 7 Safety* license is installed.

If you want to use the F-I/O in safety mode or in safety-related communication, the F-capability of the F-CPU must be enabled.

F-capability is activated by default when *STEP 7 Safety* license is installed.

Enabling F-capability

If you want to enable the F-capability, proceed as follows:

1. Select the F-CPU in the device or network view, and select the "Properties" tab in the Inspector window.
2. Select "Fail-safe" in the area navigation.
3. Enable the F-capability with the "Enable F-activation" button.

Disabling F-capability

If you want to disable the F-capability for an F-CPU because you intend to use the F-CPU as a standard CPU, note the following:

- You need F-access permission for the safety program, if assigned.
- The Safety Administration Editor (Page 80) is deleted from the project tree.
- The F-OBs are deleted. (S7-1200, S7-1500)
- All F-blocks are deleted.
- From now on you cannot use F-I/O in safety mode with this F-CPU.

If you want to disable the F-capability, proceed as follows:

1. Select the F-CPU in the device or network view, and select the "Properties" tab in the Inspector window.
2. Select "Fail-safe" in the area navigation.
3. Disable the F-capability with the "Disable F-activation" button.
4. Confirm the "Disable F-activation" dialog with "Yes".

Although you can enable the F-capability of the F-CPU without F-access permission, this permission is required to disable it.
If you do not have F-access permission and have enabled the F-capability accidentally, you can only disable it using "Undo".

Configuring the F-parameters of the F-CPU

In the "Properties" tab of the F-CPU, you can change or apply the default settings for the following parameters:

- F-destination address range
  - Low limit for F-destination addresses for F-I/Os of PROFIsafe address type 1
  - High limit for F-destination addresses for F-I/Os of PROFIsafe address type 1
- The F-source address for F-I/Os of PROFIsafe address types 2 and 3.
- The F-BaseID for F-I/O of PROFIsafe address type 3 at the F-CPU
  - Enable manual assignment and set F-BaseID
  - Button for automatic generation according to PROFIsafe specification
  - Activate delayed transfer (internal value 0x7C6)
- Default F-monitoring time for central or distributed F-I/O at the F-CPU

  (For HF-CPUs, only the default F-monitoring time for distributed F-I/O.)

NOTE

A change of the F-monitoring time for central or distributed F-I/O at the F-CPU results in modifications to the safety program when it is recompiled. A new acceptance may therefore be required.

Specify F-destination address for F-I/O of PROFIsafe address type 1

With the parameters "Low limit for F-destination addresses" and "High limit for F-destination addresses" you specify a range for this F-CPU in which the F-destination address of newly inserted F-I/Os of PROFIsafe address type 1 (Page 67) is assigned automatically. The F-destination address of an F-I/O is automatically moved to the destination address range of an F-CPU as soon as it is newly assigned to it.

The F-destination address is assigned in ascending order starting at the "Low limit for F-destination addresses". When no free F-destination address is available in the F-destination address range, the next available free F-destination address outside the F-destination address range is assigned and a warning is output during compilation.

The maximum possible F-destination address for ET 200S, ET 200eco, ET 200pro, ET 200iSP F-modules and S7-300 F-SMs is 1022.

The F-destination addresses for F-I/O of PROFIsafe address type 1 must be unique network-wide and CPU-wide.

By selecting different F-destination address ranges for different F-CPUs, you can define different ranges for the automatic assignment of the F-destination address. This is useful when you are operating multiple F-CPUs in one network. Subsequent manual address changes are possible. (see also Recommendation for PROFIsafe address assignment (Page 63))

Example:

You have configured the F-destination address range as follows:

- Low limit for F-destination addresses = 100
- High limit for F-destination addresses = 199

When inserting the first F-I/O of PROFIsafe address type 1, the F-destination address 100 is assigned. When inserting an additional F-I/O of PROFIsafe address type 1, the F-destination address 101 is assigned.

NOTE

The parameters "Low limit for F-destination addresses" and "High limit for F-destination addresses" have no effect on the following F-I/Os:

- SM 326; DI 8 x NAMUR (as of article number 6ES7326-1RF00-0AB0)
- SM 326; DO 10 x DC 24V/2A (article number 6ES7326-2BF01-0AB0)
- SM 336; AI 6 x 13 bit (article number 6ES7336-1HE00-0AB0)

Specifying the F-destination address range for F-I/O of PROFIsafe address types 2 and 3

The F-destination address of F-I/O of PROFIsafe address type 2 (Page 68) and F-I/O of PROFIsafe address type 3 (Page 70) is assigned automatically for each F-CPU in descending order starting from 65534. The low limit is the value configured with the parameter "High limit for F-destination addresses" (for F-I/O of PROFIsafe address type 1) + 1.
If the value configured with the parameter "High limit for F-destination addresses" is reached, a warning is output during compilation. (See also Recommendation for PROFIsafe address assignment (Page 63))

Specifying F-source address for F-I/O of PROFIsafe address types 2 and 3

You specify the F-source address for F-I/O of PROFIsafe address type 2 (Page 68) and for F-I/O of PROFIsafe address type 3 (Page 70) assigned to this F-CPU with the parameter "Central F-source address". The F-source address must be unique network-wide for F-I/O of PROFIsafe address type 2. The F-source address or F-BaseID must be unique network-wide for F-I/O of PROFIsafe address type 3 (see also Recommendation for PROFIsafe address assignment (Page 63))

NOTE

A change to the "Central F-source address" parameter has the consequence that the safety program is different after recompilation. A new acceptance may therefore be required because the F-source addresses of all F-I/Os of address types 2 and 3 are changed centrally as a result.

Specifying F-BaseID for F-I/O of PROFIsafe address type 3

You specify the F-BaseID for the F-CPU and for F-I/O of PROFIsafe address type 3 (Page 70) assigned to this F-CPU with the parameter "F-BaseID". This is a uniformly distributed 64-bit random number that is generated by an algorithm in accordance with the PROFIsafe specification.
A manual assignment of values generated otherwise in accordance with the PROFIsafe specification is possible provided that the requirement for uniqueness is complied with:

⚠ WARNING

Only use generation tools with which you can track adherence to the PROFIsafe generation rules, e.g. based on the documentation. The TIA Portal tools comply with these rules. *(S116)*

⚠ WARNING

The F-BaseID may only be assigned manually in exceptional cases, when you are sure that the PROFIsafe addresses of the F-I/Os of the F-CPU are unique (worldwide). F-I/Os with the same F-BaseID must differ in the combination of F-source and F-destination address. We explicitly advise against manual assignment and recommend generation with the tools offered in the TIA Portal.

With the manual assignment of value "0" as F-BaseID, you lose the advantages of PROFIsafe address type 3 and must adhere to the specifications for PROFIsafe address type 2 (see warning S052). *(S115)*

To enable assignment of the F-BaseID during startup of the F-CPU, "Delayed transfer" can be activated (see also Assignment of the F-BaseID during runtime (delayed transfer) (S7-1200 G2, S7-1500) (Page 643)).

NOTE

A change to the "F-BaseID " parameter has the consequence that the hardware configuration must be compiled and downloaded again because the F-BaseID values of all F-I/Os of address type 3 are changed centrally as a result. If you change the F-BaseID from "Delayed transfer" to another setting, you must also compile and download the safety program again.

Normally, a new acceptance of the safety-related project data (Page 433) is not required after a change of the F-BaseID because this does not change the behavior of the safety program and/or the F-I/Os. For this reason, it is also not included in the Collective F-Signature. However, an acceptance of the overall function on the destination hardware may still be necessary.

"Default F-monitoring time" parameter

Configure the "Default F-monitoring time" for monitoring the communication between the F-CPU and F-I/O.

You can adjust the F-monitoring time via the following parameters:

- "Default F-monitoring time for central F-I/O"
- "Default F-monitoring time for F-I/O of this interface"

The **default F-monitoring time for the central F-I/O** acts on the F-I/O that is arranged centrally, i.e. near the F-CPU. You set this parameter in the properties of the F-CPU (select F-CPU, then select "Properties > Fail-safe > F-parameters").

The **default F-monitoring time for the F-I/O of this interface** acts on the F-I/O that is assigned to this interface of the F-CPU (PROFIBUS or PROFINET). You change this parameter in the properties of the relevant interface (selection of the interface in the "Device view" tab, then "F-parameters").

The various settings available allow you to flexibly adapt the F-monitoring time to the conditions of your F-system, for example to take account of different bus cycles.

You can also change the F-monitoring time individually for each F-I/O in the F-I/O properties (see Configuring F-I/O (Page 53) or Peculiarities when configuring fail-safe GSD based DP slaves and fail-safe GSD based I/O devices (Page 39)).

NOTE

A change of the F-monitoring time for central or distributed F-I/O at the F-CPU results in modifications to the safety program when it is recompiled. A new acceptance may therefore be required.

NOTE

The default value for the "Default F-monitoring time for the F-I/O" for an HF-system is 2300 ms and is designed for an MRP ring. For line topology, you can set the "Default F-monitoring time for the F-I/O" smaller by a factor of approx. 2. You can determine the exact value as described in section "Monitoring and response times (Page 629)". You can set this value in the properties of the F-CPU (selection of the F-CPU, then "Properties > Fail-safe > F-parameters > PROFINET interface [X1]").

⚠ WARNING

It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time. *(S018)*

You can find more information in Monitoring and response times (Page 629).

Automatic generation of the safety program

The safety program of an F-CPU consists of one or two F-runtime groups that contain the F-blocks (see also Defining F-Runtime Groups (Page 133)). When the F-CPU (with activated F-capability) is inserted into the work area of the device view or network view, a safety program with an F-runtime group is generated automatically.

You can define in *STEP 7 Safety* that no F-runtime group is generated while inserting the F-CPU (with activated F-capability).

Proceed as follows:

1. Select the "Options > Settings" menu command.
2. Select the "STEP 7 Safety" area.
3. If it is not already disabled, disable automatic generation of an F-runtime group by deselecting the "Generate default fail-safe program" option.

This change has no influence on any existing safety programs; it only defines whether an F-runtime group is automatically generated for each one of the subsequently inserted F-CPUs.

Configuring the protection level of the F-CPU

⚠ WARNING

(S7-300, S7-400) In productive operation, it must be prevented that changes to the standard user program are unintentionally made in the safety program as well. For this purpose, you must configure the protection level "Write protection for fail-safe blocks" and configure a password for the F-CPU.

With the "Write protection" or "Write/read protection" protection level, the password would apply to the standard and the safety program. *(S001)*

⚠ WARNING

(S7-1200, S7-1500) In productive operation, the safety-related project data must be protected against unintentional changes.

If you use user management, the Runtime rights "Change parameter of the F-System (F-Admin)" and "Full access including fail-safe" and the engineering right "Manage users and roles" may only be granted to authorized users. You must also limit the assignment of user rights to trained personnel. Before the transition into productive operation, you must remove the "Change parameter of the F-System (F-Admin)" and "Full access including fail-safe" rights from the "Anonymous" user.

If you are not using user management or the "Access control via access levels" parameter is activated, then configure at least the "Full access (no protection)" protection level and assign a password under "Full access incl. fail-safe (no protection)". This protection level allows full access only to the standard user program and not the safety program.

If you select a higher protection level, for example to protect the standard user program, you must assign an additional password for "Full access (no protection)".

Assign different passwords for the individual access levels. *(S041)*

You configure the protection level following the same procedure as for standard CPUs.

For information on the password for the F-CPU, refer to Access protection (Page 99). Pay special attention to the warnings in Access protection for the F-CPU (Page 103).

2.3.1 Special features of the S7-1500 F Virtual Controller

When you configure an S7-1500 F Virtual Controller, you must provide an additional timer for forming the safe time within the S7-1500 F Virtual Controller. This timer must always involve a network card or a selected Ethernet interface of the Edge device.

In this context, observe the following warning:

⚠ WARNING

You must configure an Ethernet interface or network adapter for your S7-1500 F Virtual Controller or central Virtual Timeservice in the Edge Resource Manager of the respective Edge Device.

You need to add the respective network card via the installation interface in the configuration menu of the Industrial Edge applications S7-1500VF or the virtual time service.

- Ensure, for example using data sheets or manufacturer confirmation, that the network card you selected is a separate, non-integrated IEEE1588-capable network card within your device or an externally plugged IEEE1588-capable network card.
- Ensure that the network card has a built-in quartz for the functionality according to IEEE 1588, e.g. through data sheets or manufacturer confirmation.

You must not use network cards or Ethernet interfaces configured for the safe time of an S7-1500 F Virtual Controller for time synchronization via PTP. Neither read nor write access is permitted to the physical hardware clock (PHC) of the network card.

You must differentiate between two types of operation in the Edge environment of an S7-1500 F Virtual Controller.

- You can operate the S7-1500 F Virtual Controller on an IED. In this case, you must always configure the network card or Ethernet interface directly for the S7-1500 F Virtual Controller.
- In the second case, you can operate the S7-1500 F Virtual Controller on an IEVD within a host. Depending on the application, you can directly assign a network card or Ethernet interface to the S7-1500 F Virtual Controller in an IEVD, as in the IED case.

  If you operate multiple S7-1500 F Virtual Controllers, you can use the Virtual Timeservice offered as an Industrial Edge application. The Virtual Timeservice distributes the safe time centrally for all S7-1500 F Virtual Controllers within your host. For this, you must configure the network card or Ethernet interface for the Virtual Timeservice and set up a suitable network for distributing the time stamp. An S7-1500 F Virtual Controller supports up to two Virtual Timeservices in a network. You must label the network for the Virtual Timeservice for each S7-1500 F Virtual Controller as "virtual-timeservice".

  If you operate the S7-1500 F Virtual Controller on a virtual Edge device within a host, you must provide the network cards or Ethernet interfaces for forming the safe time of an S7-1500 F Virtual Controller in the hypervisor of your host directly, exclusively and not virtualized via PCI passthrough for the respective Virtual Timeservice application or your S7-1500 F Virtual Controller application.

Check that the network card driver for the hardware time stamping function based on PTP is compatible with the version of your hypervisor and the kernel version of your IE(V)D.

Note that a reconfiguration of the settings of an IE(V)D with an S7-1500 F Virtual Controller during operation can result in STOP. *(S101)*

2.3.1.1 Operating the S7-1500 F Virtual Controller without central Virtual Timeservice

Installing the S7-1500 F Virtual Controller application

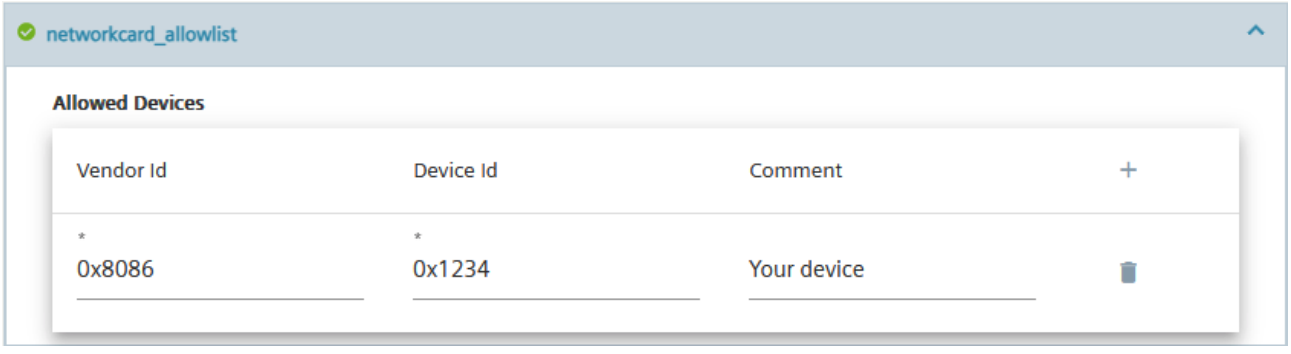
If you have not set up a central Virtual Timeservice and you want to configure a shared IEEE1588-capable interface (according to warning S101) for PROFINET communication and provision of the 2nd time base for the S7-1500 F Virtual Controller, add the network card or Ethernet interface to the "allow list" of the network cards.

NOTE

In the Installation menu, if you click "Back" and deselect the configuration with the plus sign, the previously configured networkcard_allowlist is installed/updated nonetheless. If you wish to correct your configuration, close the window by clicking the "X" and start the installation/update again.

Proceed as follows:

1. Select the "CPU 1517VF" app in the "Edge Management" under "Catalog".
2. Click "Install".
3. Add an IEEE1588-capable network card to the "vplc_x1" interface from the Resource Manager in accordance with warning S101. The "Application Manager" must be installed for this purpose.
   - To do this, check the box next to "networkcard_allowlist" and click on the plus sign in the "Allowed Devices" list.
   - Fill in "Vendor ID" and "Device ID" and, if desired, "Comment".



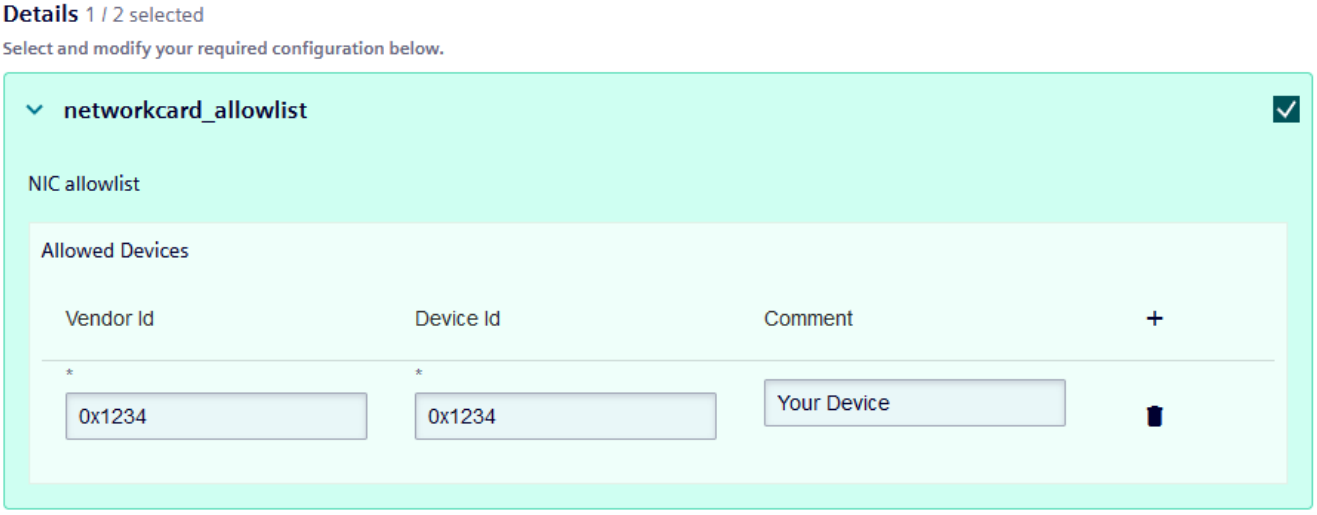
4. Click "Next".
5. Select the Edge device on which you want to install the app.
6. Click "Install Now".

Please note: The allow list can also be replaced later by updating the app.

If you are using the new UX of Edge via "new experience", follow these steps:

1. In "Edge Management", under Devices, select the Edge device on which you wish to install the "CPU 1517VF" app.
2. Click "Install application".
3. By means of the checkbox, select the "CPU 1517VF" app, whereupon the configuration menu for the app appears on the right.

4. Add an IEEE1588-capable network card to the "vplc_x1" interface from the Resource Manager in accordance with S101.
   - Open the "networkcard_allowlist" tab and click the plus sign in the "Allowed Devices" list.
   - Fill in "Vendor ID" and "Device ID" and, if desired, "Comment".
   - To install the configuration, put a check mark in the checkbox at the top right of the field so that the field becomes green.
5. Click "Install(1)".



Updating the S7-1500 F Virtual Controller application or replacing the allow list by updating the configuration

If you want to update the S7-1500 F Virtual Controller app without a set-up central Virtual Timeservice and **not** replace the allow list in the process, follow these steps:

1. Select the new version of the "CPU1517VF" app under "Catalog" in "Edge Management".
2. Click "Update". (For all Edge devices on which no version of the "CPU1517VF" app is installed, "Install" is displayed)
3. Use "Next" to jump to the "Configuration" menu. The green check mark next to "networkcard_allowlist" must be cleared.

If you want to update the S7-1500 F Virtual Controller app or update the configuration and replace the allow list in the process, follow these steps:

1. In "My Installed Apps", click on the respective app and select "Update Configuration" in the menu.

   For an update of the app, select the new version of the "CPU1517VF" app under "Catalog" in "Edge Management".
2. Click "Update".
3. Click on the plus sign.
4. Fill in "Vendor ID" and "Device ID" and, if desired, "Comment".

5. Set the green check mark to the left of "networkcard-allowlist".
6. Click "Next".
7. Select "Update Now".

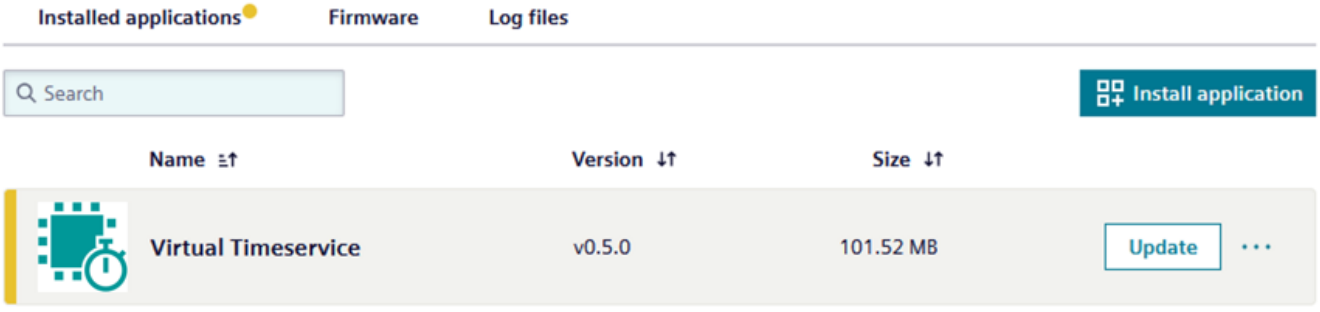
Note that when the configuration is updated, the allow list is only replaced after a STOP/START of the "CPU1517VF" app and not after a STOP/RUN of the S7-1500VF Virtual Controller.

If you are using the new UX of Edge via "new experience", follow these steps for an "Update Configuration":

1. Select the Edge device with your app in "Edge Management", under Devices.
2. Right-click on the three dots and select "Update Configuration".
3. Thereafter, follow the same procedure as for installing an app.
4. Click "Update(1)".

If you are using the new UX of Edge via "new experience", follow these steps for an "Update":

1. Select the Edge device with your app in "Edge Management", under Devices.
2. If there is an update available, you will see the "Update" button. Click "Update".
3. Thereafter, follow the same procedure as for installing an app.
4. Click "Update(1)".



NOTE

When you perform an Update Configuration, you need to stop and then restart the app for the configuration change to take effect.

2.3.1.2 Operating the S7-1500 F Virtual Controller with central Virtual Timeservice

Installation and setup of the Industrial Edge Virtual Timeservice application for forming a second time base for the S7-1500 F Virtual Controller

If you want to set up the Virtual Timeservice centrally for multiple S7-1500 F Virtual Controllers on a host with multiple S7-1500 F Virtual Controllers, observe the following:

- Install the Virtual Timeservice on a separate virtual Edge Device. You must not operate a Virtual Timeservice and an S7-1500 F Virtual Controller on the same virtual Edge Device.
- Connect all participating virtual Edge devices with an S7-1500 F Virtual Controller to the virtual Edge Device on which the central Virtual Timeservice has been installed and set up via a dedicated real-time capable network.
- This network should be used exclusively by the Virtual Timeservice for distributing the second time base.
- You can set up up to two central Virtual Timeservices for a dedicated network. To increase the availability of the system, you should set up two Virtual Timeservices.
- You must configure a separate IEEE1588-capable Ethernet interface or network card for each virtual time service in accordance with warning S101.
- Active PTP synchronization between the Virtual Timeservice and the S7-1500VF using, for example, boundary clock switches in the network can result in a connection abort and, thus, a STOP of the S7-1500VF. The active Virtual Timeservice ID indicated in the S7-1500VF can be changed by an active PTP-synchronizing component and consequently differ from the original Virtual Timeservice ID.

More information on setting up the Virtual Timeservice is provided in the "Virtual Timeservice App" Operating Manual.

Configuring a second network interface for the S7-1500 F Virtual Controller with a central Virtual Timeservice

For the S7-1500 F Virtual Controller to use the Virtual Timeservice as 2nd time base, follow these steps to configure network resources for the IEVD with the S7-1500 F Virtual Controller.

1. Open the "Settings" menu item in the IEVD web interface.
2. Go to the "Ressource Manager" tab.
3. To add a network, click on the plus sign to the right of "Isolatable Networks" under "Configuration".

   The new input fields appear at the end of the page (you may have to scroll down).
4. Under "Name", enter the name of the corresponding network interface.

   In "Annotations", right-click on the plus sign again to be able to assign a label for the interface.
5. For the S7-1500 F Virtual Controller with a Virtual Timeservice, create an additional entry for the provision of the 2nd time base next to the interface with label "vplc_x1".

   Assign the "virtual-timeservice" label to the interface to the real-time-capable network via which the time stamp is distributed and received by the S7-1500 F Virtual Controller
6. Click on "Submit" at the bottom to accept the entries.

Installing the S7-1500 F Virtual Controller application with central Virtual Timeservice

If you have set up a central Virtual Timeservice, install the app as described in the "Virtual Timeservice App" Operating Manual. Do not specify any additional network card under "Configurations" in the installation menu.

2.4 Configuring F-I/O

Introduction

You configure the S7-1500/ET 200MP, ET 200SP, ET 200SP HA, ET 200AL, ET 200S, ET 200eco (S7-300, S7-400), ET 200eco PN, ET 200pro, ET 200iSP F-modules, the S7-300 F-SMs and the S7-1200 F-modules as usual in *STEP 7*.

After you have inserted the F-I/O in the work area of the *device or network view*, you access the configuration dialogs by selecting the relevant F-I/O and the "Properties" tab.

NOTE

Changes to the parameter assignment result in modifications to the safety program when it is recompiled. A new acceptance may therefore be required.

The ET 200SP F-modules can be used in distributed system with:

- IM 155-6 PN ST as of firmware V1.1
- IM 155-6 PN HF
- IM 155-6 PN/2 HF as of firmware V4.2
- IM 155-6 PN/3 HF as of firmware V4.2
- IM 155-6 MF HF
- IM 155-6 PN HS
- IM 155-6 DP HF
- IM 155-6 PN R1

  You can find information on whether operation with R1 is supported for an ET 200SP F-module on the Internet. (https://support.industry.siemens.com/cs/ww/en/view/73021864)

The distributed use of S7-1500/ET 200MP F-modules is possible with:

- IM 155-5 PN BA as of firmware V4.3
- IM 155-5 PN ST as of firmware V3.0
- IM 155-5 PN HF as of firmware V2.0
- IM 155-5 DP ST as of firmware V3.0

The central use of S7-1500/ET 200MP F-modules is possible with S7-1500 F-CPUs as of firmware V1.7, distributed use as of firmware V1.5.

(S7-1200 G1) We recommend you limit the total number of F-I/Os used centrally or distributed in an S7-1200 G1 F-CPU to 12. Depending on the volume of project data, the maximum number of F-I/Os can be smaller.

⚠ WARNING

When you make changes in which the assignment of input/output addresses and wiring can change, then you must perform a wiring test (Page 397).

Examples for such changes are:

- Adding F-I/O
- Changing the start address of F-I/O
- Changing the slot position of F-I/O
- Changing
  - the rack
  - the slave/device address
  - the PROFIBUS DP/PROFINET IO subnet
  - the IP address
  - the device name

(S071)

Channel-granular passivation after channel faults

You can configure how the F-I/O will respond to channel faults, such as a short-circuit, overload, discrepancy error, or wire break, provided the F-I/O supports this parameter (e.g. for ET 200S or ET 200pro F-modules). You configure this response in the properties for the relevant F-I/O ("Behavior after channel fault" parameter). This parameter is used to specify whether the entire F-I/O or just the faulty channel(s) are passivated in the event of channel faults.

NOTE

(S7-300, S7-400) Note that channel-granular passivation increases the runtime of the F-runtime group(s) compared to passivation of the entire F-I/O (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).

"Channel failure acknowledge" parameter (S7-1200, S7-1500)

In the case of F-I/Os that support the "Channel failure acknowledge" channel parameter (for example S7-1500/ET 200MP F-modules, S7-1200 F-modules or ET 200AL F-modules), this function replaces the ACK_NEC tag of the F-I/O DB.

If an F-I/O fault is detected by the F-I/O, passivation of all channels of the relevant F-I/O occurs. If channel faults are detected, the relevant channels are passivated if "Passivate channel" is configured. If "Passivate the entire module" is configured, all channels of the relevant F-I/O are passivated. Once the F-I/O fault or channel fault has been eliminated, reintegration of the relevant F-I/O/the relevant channel occurs in line with the "Channel failure acknowledge" parameter.

- Automatically
- Manually

The parameter "Channel failure acknowledge" can be set individually for each channel with channel-granular passivation if "Adjustable" has been parameterized at the "Reintegration after channel fault" parameter.

⚠ WARNING

The parameter assignment "Channel failure acknowledge = Automatic" is only allowed if automatic reintegration is permitted for the relevant process from a safety standpoint.*(S045)*

NOTE

The default assignment for the "Channel failure acknowledge" parameter when the F-module is created is "Manually".

Organization block/Process image (S7-1200, S7-1500)

If you use F-I/O in standard mode, you can select the organization block/process image as you do for standard I/O.

If you use F-I/O in safety mode, no selection is possible. The process image is always updated automatically at the beginning or end of the F-runtime group (see section Program structure of the safety program (S7-1200, S7-1500) (Page 113)).

Contrary to F-I/O operated in non-isochronous mode, you need to selected a process image partition, such as PIP 1 for F-I/O that is operated in isochronous mode (see "Configuring isochronous mode (S7-1200 G2, S7-1500) (Page 62)").

Changing the name and number of the F-I/O DB

For more information, refer to the section "Fail-safe I/O data block (Page 164)".

Customizing the F-monitoring time for F-I/O

You can customize the F-monitoring time in the properties of the F-I/O under "F-parameters". This may be necessary to prevent a timeout being triggered when no error occurs and the F-I/O requires a longer F-monitoring time or assignment with a default F-monitoring time is not possible. For this purpose, activate the corresponding check box and assign an F-monitoring time.

NOTE

A change of the F-monitoring time for central or distributed F-I/O at the F-CPU results in modifications to the safety program when it is recompiled. A new acceptance may therefore be required.

⚠ WARNING

It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time. *(S018)*

You can find more information in Monitoring and response times (Page 629).

Group diagnostics for fail-safe S7-300 signal modules

By disabling a channel of the fail-safe signal module in the module properties, you also disable the group diagnostics for this channel.

Exception for S7-300/400 F-CPUs:

For the following S7-300 fail-safe signal modules, the monitoring of channel-specific diagnostic messages (for example, wire break, short-circuit) from the F-SM to the F-CPU is enabled and disabled with the "Group diagnostics" parameter.

- SM 326; DI 8 x NAMUR (as of article number 6ES7326-1RF00-0AB0)
- SM 326; DO 10 x DC 24V/2A (article number 6ES7326-2BF01-0AB0)
- SM 336; AI 6 x 13Bit (article number 6ES7336-1HE00-0AB0)

You should disable group diagnostics for **unused** input or output channels.

⚠ WARNING

(S7-300, S7-400) For the following S7-300 fail-safe signal modules (F-SMs) with activated safety mode, "Group diagnostics" must be enabled on all used channels:

- SM 326; DI 8 x NAMUR (article numbers 6ES7326-1RF00-0AB0 and 6ES7326-1RF01-0AB0)
- SM 326; DO 10 x DC 24V/2A (article number 6ES7326-2BF01-0AB0)
- SM 336; AI 6 x 13 Bit (article number 6ES7336-1HE00-0AB0)

(S003)

Diagnostic interrupts can be enabled optionally.

More information

For detailed description of the **parameters**, refer to the help on the properties of the respective F-I/O and in the respective *manual for the F-I/O*.

2.5 Configuration control (option handling) for F-I/Os

For configuration control (option handling) with F-I/Os proceed as with the standard I/O devices. Detailed information can be obtained by searching for "Configuration control (option handling)" in the help of *STEP 7*.

The following section describes what you have to observe additionally for F-I/Os.

Requirements

- The requirements that are specified under "Configuration control (option handling)" in the help of *STEP 7* are fulfilled.
- The requirements that are specified under "Configuration control (option handling)" in the help of *STEP 7* are fulfilled. Handle the F-I/O as standard I/O.
- V2.1 or higher is set as the safety system version.
- The F-I/Os for which you use the configuration control (option handling) are located
  - Distributed at an F-CPU S7-300/400/1200/1500
  - Centrally at an F-CPU S7-1500

  The PROFIsafe address of the F-I/Os are set or assigned.

NOTE

Only if the maximum configuration actually exists is assigning the PROFIsafe addresses (Page 73) possible.

Procedure

(S7-1200, S7-1500) Disable the F-I/Os not existing in the respective variant (option) by setting the DISABLE (Page 169) variable in the associated F-I/O DB (Page 164) to "1". This prevents the flashing of the error LED of the F-CPU and diagnostic entries of the safety program that reference these F-I/Os. With the DISABLED (Page 169) variable of the associated F-IO data block, you can evaluate whether an F-module is deactivated.

(S7-300, S7-400) To prevent the flashing of the error LEDs of the F-CPUs you do not have to observe anything further. You cannot suppress diagnostics entries.

⚠ WARNING

If configuration control is used, your actual configuration deviates from the configured maximum configuration. You identify F-I/Os that do not exist in the current option (station option) via control record as "not available".

If an F-I/O marked as "not available" is possibly nevertheless in the real system, it has to be ensured that substitute values (0) are provided for these F-I/Os in the safety program or output at the outputs. This is achieved by setting the DISABLE tag (S7-1200, S7-1500) or respectively PASS_ON tag (S7-300, S7-400) in the associated F-I/O DB to "1". *(S077)*

2.5.1 Example

Introduction

The following example shows how you to

- Select/detect a station option
- Deactivate F-I/Os that are not present in a station option (S7-1200, S7-1500)
- Provide your safety program for various station options

Safe selection/detection of the station option

You carry out a safe selection/detection of a station option with inputs of an F-I/O wired fixed to M/L+.

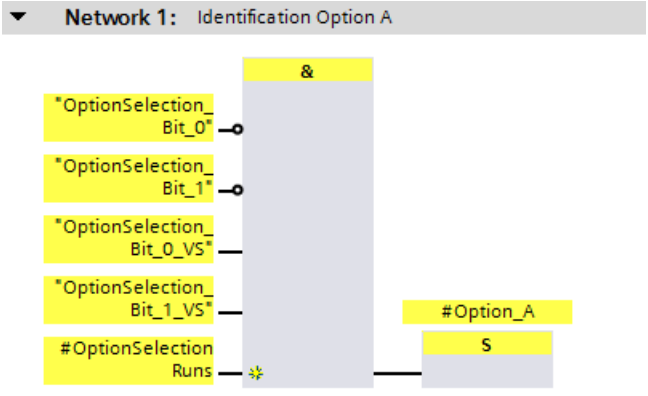
For example, you can select up to 4 station options with 2 inputs of an F-I/O.

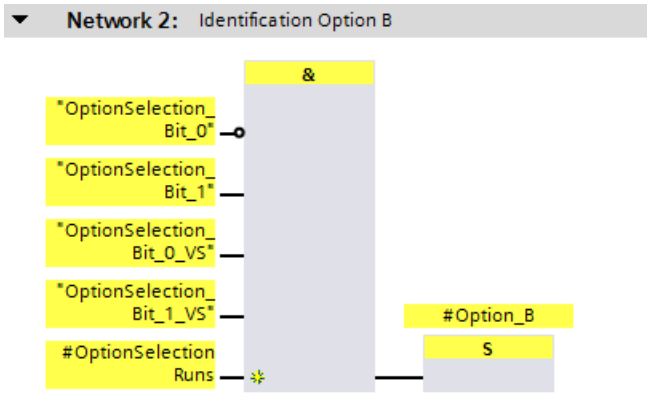
Option	OptionSelection_Bit_0	OptionSelection_Bit_1
Q	0	0
B	0	1
C	1	0
D	1	1

Note when detecting the station option that substitute values (0) are used for the inputs of the F-I/O in certain situations, e.g. during startup of the F-system or when F-I/O channel errors occur.

In these situations, the present station option cannot be detected. You should therefore also evaluate the value status of inputs and only apply the station option one-time after startup of the F-system.

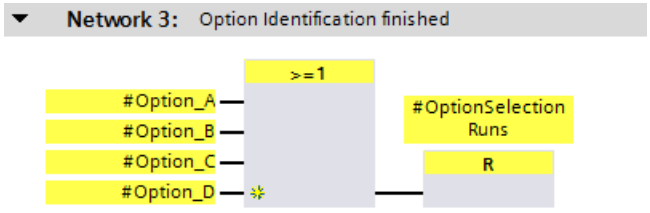
For one-time recognition of the station option, define a static local datum, for example, OptionSelectionRuns with default value "TRUE".





Correspondingly for options C and D.

As soon as a station option is detected, reset the static local datum for one-time detection of the station option:



NOTE

When you make the selection/detection of a station option only in the standard user program, only the "Station option" is available to you as a standard datum that is not secured.

Make sure that no dangerous states arise from this.

Read the section "Data exchange between standard user program and safety program (Page 198)".

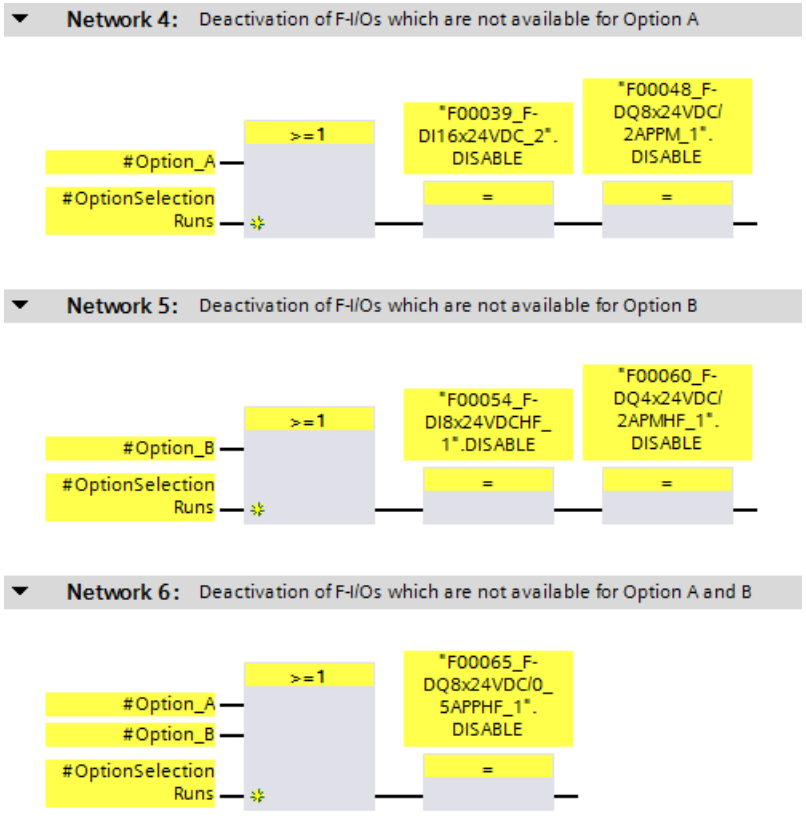
Disabling F-I/Os that are not present in a station option

If one or more F-I/Os are not present in a station option, you can prevent the blinking of the error LED of the F-CPU by disabling these F-I/Os.

In addition, diagnostic messages of the safety program that refer to these F-I/Os are suppressed.

NOTE

As long as the detection of the station option (during startup of the F-system) is not yet complete (OptionSelectionRuns = TRUE), you should disable all "optional" F-I/O devices.



Providing the safety program for various station options

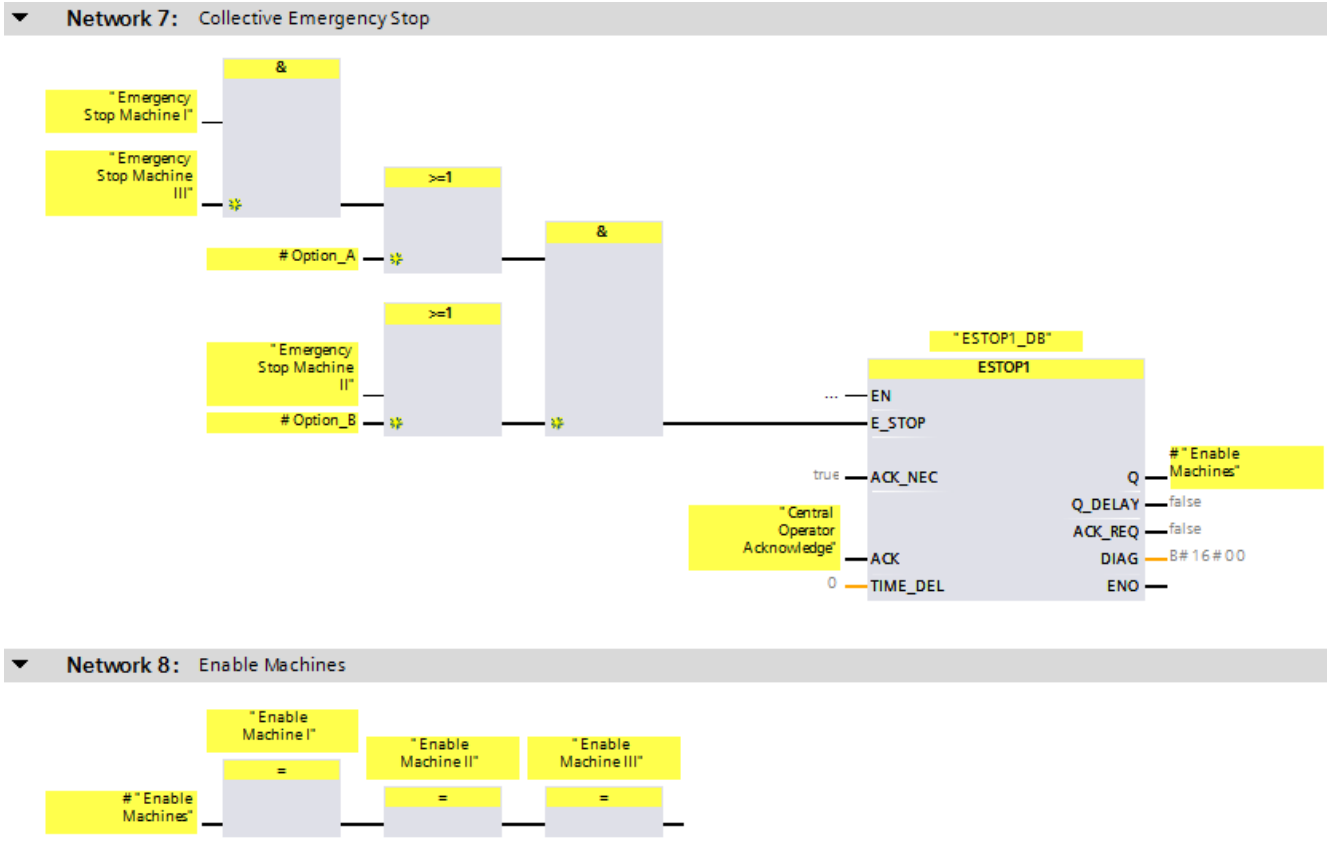
In the following example, the EMERGENCY STOP signals of different plant units or machines are combined into a collective EMERGENCY STOP signal.

Machines I and III and the corresponding F-I/O with the EMERGENCY STOP signal for machines I and III are not present with station option A.

Machine II and the corresponding F-I/O with the EMERGENCY STOP signal for machine II is not present with station option B.

The substitute values (0) are therefore used in the safety program for the EMERGENCY STOP signals from the respective unavailable machines.

In order to prevent the collective EMERGENCY STOP from being triggered because machines / EMERGENCY STOP signals are not present with certain station options, you can suppress the evaluation of the EMERGENCY STOP signal for unavailable machines by taking into account the present station option.



2.6 Configuring shared device

To configure shared devices follow the procedure as in the standard. The configuration is described in the *STEP 7* help under "Configuring shared devices" and in the function manual "PROFINET with STEP 7 (https://support.industry.siemens.com/cs/ww/en/view/49948856)" in the chapter "Shared Device".

F-destination addresses

Please also read the chapter "Recommendation for PROFIsafe address assignment (Page 63)" for assigning the F-destination address.

See also

Assigning PROFIsafe address to an F-module in a shared device in multiple projects (Page 77)

Assigning PROFIsafe address to an F-module in a shared device in a common project (Page 78)

2.7 Configuring isochronous mode (S7-1200 G2, S7-1500)

To configure isochronous mode for F-I/Os that support this mode, e.g. "Profisafe Telgr 902" submodule of the SINAMICS S120 CU310-2 PN V5.1 drive, proceed as in the standard. The configuration is described in the *STEP 7* help under "Configuring isochronous mode".

Note the following:

- Contrary to F-I/O operated in non-isochronous mode, you need to selected a process image partition, such as PIP 1 for F-I/O that is operated in isochronous mode.
  This process image partition must contain only F-I/O operated in isochronous mode and no standard I/O.
- The assigned isochronous mode interrupt OB must first be generated as F-OB by specifying a F-runtime group (see Procedure for defining an F-runtime group (S7-1200, S7-1500) (Page 138)). It is not possible to add an F-OB with event class "Synchronous Cycle" directly during the configuration of the isochronous mode.

Requirement

F-CPUs that support IRT.

Connection of F-I/O operated in isochronous mode to the isochronous mode interrupt OB

You access F-I/O operated in isochronous mode in the same way as you do standard I/O operated in isochronous mode, via the select process image partition.

Contrary to standard I/O that is operated in isochronous mode, the process image partition is updated by the F-system at the beginning or end of the F-OB (see Program structure of the safety program (S7-1200, S7-1500) (Page 113)).

No calling of the instructions SYNC_PI and SYNC_PO is required in the F-OB.

NOTE

With isochronously operated F-I/O, it is not ensured (fail-safe) that all input data of the F-I/Os assigned to the process image partition are consistently available at the beginning of the main safety block or all output data is transferred consistently to the F-I/Os, in other words, logically and temporally together. The consistency is only ensured within an F-I/O.

The consistency of all isochronous F-I/Os of the process image partition usually depends on the number of isochronous F-I/Os and the scope of the safety program in the isochronous mode interrupt OB.

If corresponding consistency requirements exist, you need to check the consistency of the input and output data yourself, e.g. by also transferring and evaluating time stamps in the input and output data of the isochronously operated F-I/O.

2.8 Recommendation for PROFIsafe address assignment

Before inserting the F-I/O, specify an address range for each F-CPU for the F-destination addresses of the F-I/O of PROFIsafe address type 1 (Page 67) that does not overlap with the address range of any other F-CPU network-wide or CPU-wide (system-wide). You define the range with the parameters "Low limit for F-destination addresses" and "High limit for F-destination addresses" (see also section Configuring an F-CPU (Page 41)).

In a redundant S7-1500HF system, consider both F-CPUs of the redundant S7-1500HF system as one F-CPU with regard to the PROFIsafe addresses. The "Low/high limit for F-destination addresses" or the "Central F-source address" and, if present, the F-BaseID is therefore set identically by the system for both F-CPUs.

The F-destination addresses of F-I/O of PROFIsafe address type 2 (Page 68) must not overlap with any address range of the F-I/O of PROFIsafe address type 1. The ranges of the F-destination addresses of the F-I/O of PROFIsafe address type 2 may overlap if the F-source addresses are different. This is the case for supported configurations (Page 64) if the "Central F-source address" parameter has been set differently for each F-CPU.

If the F-BaseID was generated individually for each F-CPU by the TIA Portal for an F-I/O of PROFIsafe address type 3, its F-source addresses and/or F-destination addresses can overlap with those of the F-I/Os of F-address type 1 or 2 of any other F-CPUs as needed.

Assign relatively low F-destination addresses for F-I/O of PROFIsafe address type 1 and relatively high F-destination addresses for F-I/O of PROFIsafe address type 2 and 3.

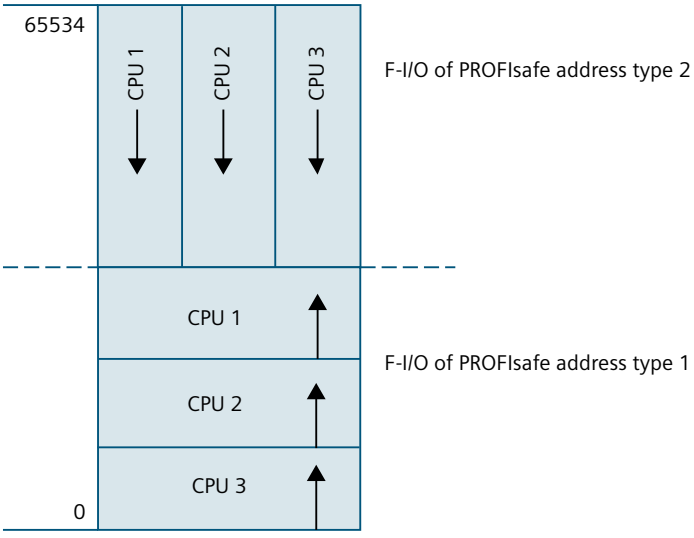


Figure 2-1 Address assignment for F-I/O of PROFIsafe address types 1 and 2

The safety summary (Page 415) lists the following information for each F-CPU:

- "Central F-source address" parameter (F-source address for F-I/O of PROFIsafe address type 2 and 3)
- Actually used range of the F-destination addresses of the assigned F-I/O of PROFIsafe address type 1
- Actually used range of the F-destination addresses of the assigned F-I/O of PROFIsafe address type 2
- Actually used range of the F-destination addresses of the assigned F-I/O of PROFIsafe address type 3

Any F-I/O configured using I-slave-slave communication is taken into account in the safety summary as part of the F-destination address range of the I-slave.

Any F-I/O configured in a shared device is indicated in the safety summary as part of the F-destination address range of the F-CPU to which this F-I/O is assigned.

2.9 Configurations supported by the SIMATIC Safety F-system

Supported configurations

F-I/Os (see Overview of Configuration (Page 35)) are supported in the following configurations:

Central configuration (also I-slave):

- The F-I/O is in the same rack as the associated F-CPU.
- The F-I/O is located in a subrack of the rack of the associated F-CPU.

Distributed configuration (at integrated DP-/PN interface of the CPU or at CP/CM):

- PROFIBUS DP (also after IE/PB link)
  - The F-I/O is located on a DP Slave.
  - The F-I/O is located on a DP Slave and is addressed via I-slave-slave communication. The assigned DP master (of the assigned IO controller of the IE/PB link) can be a standard CPU or an F-CPU.
- PROFINET IO
  - The F-I/O is located on an IO Device.
  - The F-I/O is located in a shared I-device.
  - The F-I/O is located in a shared device.

For a redundant S7-1500HF system the only supported configuration is:

Distributed configuration (at integrated PN interface of the HF-CPU)

- PROFIBUS DP after IE/PB link
  - The F-I/O is located on a DP slave
- PROFINET IO
  - The F-I/O is located in an IO device that is assigned to the IO systems of both F-CPUs.

    More information on IO devices for R1, S1 and S2 redundancy is available in the PROFINET Function Manual (https://support.industry.siemens.com/cs/ww/en/view/49948856) in section "PROFINET with the redundant S7-1500R/H system".

For F-I/O not listed in "Overview of Configuration (Page 35)", check the relevant documentation to see whether it is supported by the SIMATIC Safety F-system. If in doubt, treat these F-I/Os as part of a configuration that is not supported.

Checks performed by the SIMATIC Safety F-system

For supported configuration, the F-system checks:

- Whether the PROFIsafe operating mode parameter (F_Par_Version) is set to V2 mode in the PROFINET IO environment**.
- Whether the F-destination addresses have been assigned uniquely CPU-wide.
  You yourself must ensure the network-wide uniqueness of the PROFIsafe address when using F-modules with F-PROFIsafe address type 1 and/or PROFIsafe address type 2 (see also "Recommendation for PROFIsafe address assignment (Page 63)")
- Whether the F-source address for F-I/O of PROFIsafe address types 2 and 3 corresponds to the "Central F-source address" parameter of the F-CPU.
- Whether the F-BaseID for F-I/O of PROFIsafe address type 3 corresponds to the "F-BaseID" parameter of the F-CPU.

NOTE

For more information on the assignment of PROFIsafe addresses that are unique for the CPU and across the network, see this FAQ (https://support.industry.siemens.com/cs/ww/en/view/109740240).

Not supported configurations

For configurations that are not supported, the TIA Portal cannot automatically check the parameter assignment of the F-I/Os. In addition, it is not assured that these have been captured in the safety summary. You must therefore observe the following warning:

⚠ WARNING

If you create a configuration that is not one of the supported configurations, you must observe the following in addition to the warnings S051, S052, S053 and S115:

- Make sure that the F-I/O of this configuration appears in the safety summary and that an F-I/O DB has been created for it. Otherwise, you cannot use the F-I/O in this configuration. (Contact Customer Support.)
- For F-I/O in the PROFINET IO environment[1], you must check the safety summary for correctness of the PROFIsafe operating mode parameter (F_Par_Version). V2 mode must be set in the PROFINET IO environment. F-I/O that support only V1 mode must not be used in the PROFINET IO environment.
- Based on the safety summary, you must check whether the components of the PROFIsafe address centrally assigned for the F-CPU were transferred correctly into the respective F-I/Os:
  - PROFIsafe address type 3: F-BaseID from the "F-BaseID" parameter of the F-CPU.
  - PROFIsafe address type 2 and 3: F-source addresses from the "Central F-source address" parameter of the F-CPU.
- You must ensure the CPU-wide[2,3] and/or network-wide[4] unique assignment of the F-destination address for F-I/Os:
  - For F-I/O of PROFIsafe address type 1 or 2 (if you cannot set the F-source address to match the "Central F-source address" parameter of the F-CPU), you must ensure the network-wide[4] uniqueness of the PROFIsafe address through the unique assignment of the F-destination address alone.
  - In an unsupported configuration, you must check the uniqueness of the F-destination address individually for each F-I/O using the safety summary, regardless of the PROFIsafe address type.

    PROFIsafe address type 1: Network-wide[4] and CPU-wide[2,3]

    PROFIsafe address type 2 and 3: Only CPU-wide[2,3].

(S050)

1) The F-I/O is located in the "PROFINET IO environment" if at least part of safety-related communication with the F-CPU takes place via PROFINET IO. If the F-I/O is connected via I-slave-slave communication, also keep in mind the communication line to the DP master/IO controller.

2) "CPU-wide" means all F-I/Os assigned to an F-CPU: Central F-I/O of this F-CPU, F-I/O for which the F-CPU is DP master/IO controller and assigned F-I/O in a shared device. An F-I/O that is addressed using I-slave-slave communication is assigned to the F-CPU of the I-slave and not to the F-CPU of the DP master / IO controller.

3) In a redundant S7-1500HF system, both F-CPUs of the redundant S7-1500HF system are to be considered as one F-CPU with regard to the PROFIsafe addresses. The "Central F-source address" and, if present, the F-BaseID are therefore set identically for both F-CPUs by the system.

4) A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

2.10 PROFIsafe addresses for F-I/O of PROFIsafe address type 1

F-destination address

The uniqueness of the PROFIsafe address is ensured solely with the F-destination address. The F-source address is not displayed and has no effect on whether or not the PROFIsafe address is unique.

Therefore, the F-destination address must be unique network-wide and CPU-wide (see the following rules for address assignment).

To prevent incorrect parameter assignment, an F-destination address which is unique CPU-wide is automatically assigned during placement of the F-I/O in the work area of the device or network view as long as you only configure supported configurations (Page 64).

To ensure a network-wide unique F-destination address assignment when multiple DP master systems and PROFINET IO systems are operated on one network, you must set the "Low limit for F-destination addresses" and "High limit for F-destination addresses" in the properties of the F-CPU in SIMATIC Safety F-systems appropriately, before placing the F-I/O (see section "Recommendations for address assignment") so that the F-destination address ranges do not overlap.

When you change the F-destination address of an F-I/O, the CPU-wide uniqueness of the F-destination address is checked automatically for supported configurations. You yourself must ensure the network-wide uniqueness of the F-destination address.

For ET 200S, ET 200eco (PROFIBUS), ET 200pro, ET 200iSP F-modules and S7-300 F-SMs: You must set the F-destination address at the F-I/O with the DIP switch before you install the F-I/O. You can assign up to 1022 different F-destination addresses.

NOTE

(S7-300, S7-400) For the following fail-safe S7-300 signal modules, the F-destination address is the start address of the F-SM divided by 8:

- SM 326; DI 8 x NAMUR (as of article number 6ES7326-1RF00-0AB0)
- SM 326; DO 10 x DC 24V/2A (article number 6ES7326-2BF01-0AB0)
- SM 336; AI 6 x 13 Bit (article number 6ES7336-1HE00-0AB0)

You can show the columns "F-source address" and "F-destination address" in the device view of the device overview. The addresses displayed in these columns are for information purposes only. You must check the F-destination addresses in the safety summary during system acceptance.

Rules for address assignment

⚠ WARNING

F-I/Os of PROFIsafe address type 1 are uniquely addressed by their F-destination address (e.g. with the switch setting on the address switch).

The F-destination address (and therefore also the switch position on the address switch) of the F-I/O of PROFIsafe address type 1 must be unique network-wide* and CPU-wide**/*** (system-wide) **for the entire** F-I/O. For this, the F-I/O of PROFIsafe address type 2 and PROFIsafe address type 3 when F-BaseID is inactive (value = 0) must also be taken into account. *(S051)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

** "CPU-wide" means all F-I/O assigned to an F-CPU: Central F-I/O of this F-CPU, F-I/O for which the F-CPU is DP master/IO controller and assigned F-I/O in a shared device. An F-I/O that is addressed using I-slave-slave communication is assigned to the F-CPU of the I-slave and not to the F-CPU of the DP master / IO controller.

*** In a redundant S7-1500HF system, both F-CPUs of the redundant S7-1500HF system are to be considered as one F-CPU with regard to the PROFIsafe addresses. The "Central F-source address" and, if present, the F-BaseID are therefore set identically for both F-CPUs by the system.

Also note Recommendation for PROFIsafe address assignment (Page 63).

NOTE

For more information on the assignment of PROFIsafe addresses that are unique for the CPU and across the network, see this FAQ (https://support.industry.siemens.com/cs/ww/en/view/109740240).

See also

Completeness of the safety summary (Page 415)

2.11 PROFIsafe addresses for F-I/O of PROFIsafe address type 2

F-source address and F-destination address

The uniqueness of the PROFIsafe address is ensured by the combination of F-source address and F-destination address.

The PROFIsafe address must be unique network-wide and CPU-wide. This is the case if the following two conditions are met:

- The F-source address ("Central F-source address" parameter) of the F-CPU is unique network-wide. Keep this in mind for changes.
- The F-destination address of the F-module is unique CPU-wide.

You define the F-source address using the "Central F-source address" parameter in the F-CPU. Provided you only configure supported configurations (Page 64), this parameter is automatically applied as the F-source address and a CPU-wide unique F-destination address is assigned (usually in descending order starting with 65534).

For supported configurations, if you change the F-destination address, then the CPU-wide uniqueness of the F-destination address is checked automatically.

You must assign the F-source address and F-destination address to the F-I/O before you commission the F-I/O. You can find more information in Assigning a PROFIsafe address of the F-I/Os with SIMATIC Safety (Page 73).

You can show the columns "F-source address" and "F-destination address" in the device view of the device overview. The addresses displayed in these columns are for information purposes only. You must check the F-source and F-destination addresses in the safety summary during system acceptance.

Rules for address assignment

⚠ WARNING

F-I/O of PROFIsafe address type 2 is uniquely addressed using a combination of F-source address ("Central F-source address" parameter of the assigned F-CPU) and F-destination address.

The combination of F-source address and F-destination address for each F-I/O of PROFIsafe address type 2 must be unique network-wide* and CPU-wide**/*** (system-wide). In addition, the F-destination address must not be occupied by F-I/O of PROFIsafe address type 1.

If you use a supported configuration (Page 64), you only have to ensure that the "Central F-source address" parameter of all F-CPUs is unique network-wide*. *(S052)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

** "CPU-wide" means all F-I/O assigned to an F-CPU: Central F-I/O of this F-CPU, F-I/O for which the F-CPU is DP master/IO controller and assigned F-I/O in a shared device. An F-I/O that is addressed using I-slave-slave communication is assigned to the F-CPU of the I-slave and not to the F-CPU of the DP master / IO controller.

*** In a redundant S7-1500HF system, both F-CPUs of the redundant S7-1500HF system are to be considered as one F-CPU with regard to the PROFIsafe addresses. The "Central F-source address" and, if present, the F-BaseID are therefore set identically for both F-CPUs by the system.

Also note Recommendation for PROFIsafe address assignment (Page 63).

NOTE

For more information on the assignment of PROFIsafe addresses that are unique for the CPU and across the network, see this FAQ (https://support.industry.siemens.com/cs/ww/en/view/109740240).

See also

Completeness of the safety summary (Page 415)

2.12 PROFIsafe addresses for F-I/O of PROFIsafe address type 3

Introduction

As well as the F-source address and F-destination address, F-I/Os of the PROFIsafe address type 3 have an additional address named the F-BaseID.

NOTE

The terminology "PROFIsafe address type 3" used in this documentation is equivalent to the designation "PROFIsafe address type 2 with BaseID". This applies in TIA Portal as well as to F-I/Os.

This F-BaseID is a 64-bit number, which is valid centrally for all F-I/Os of PROFIsafe address type 3 of an F-CPU and must be randomly generated and evenly distributed by a generator. The TIA Portal has these generators so that you can easily generate an F-BaseID.

The advantage of the F-BaseID is that, for F-I/Os with PROFIsafe address type 3, you do not need to worry about uniqueness within the network and CPU. Network boundaries are irrelevant, so you do not need network isolators (e.g. SCALANCE SC622-2C) in systems in which only F-I/Os with F-address type 3 are used. Due to its 64-bit size and its generation as an evenly distributed random number, the F-BaseID is sufficiently likely to be unique worldwide.

You no longer need to manage or coordinate the PROFIsafe addresses of your plant for F-I/Os with PROFIsafe address type 3 to guarantee that they are unique network-wide. The TIA Portal ensures that the F-destination addresses are unique CPU-wide (for supported configurations (Page 64)). Therefore, you only need to generate a new F-BaseID.

Delayed transfer

Another advantage of the F-BaseID is the "delayed transfer" property. With this, you can have the F-BaseID generated in the user program on an F-CPU on the first startup. In this way, you can create series machine programs that you distribute to multiple F-CPUs by simply copying SD cards. *STEP 7 Safety* offers specific instructions, which are described in the appendix "Assignment of the F-BaseID during runtime (delayed transfer) (S7-1200 G2, S7-1500) (Page 643)", to generate the F-BaseID in the user program.

Requirements for using the F-BaseID

The F-BaseID is supported for S7-1200 G2 F-CPUs and S7-1500 F-CPUs as of firmware V4.1 or F-Software Controllers as of V40.1. Safety system version V2.8 or higher must also be set in the *Safety Administration Editor*.

To find out which F-modules support the F-BaseID, refer to the respective equipment manual.

Generation of the F-BaseID

To have an F-BaseID generated, press the "Generate F-BaseID" button in the "F-BaseID for PROFIsafe address type 3" section in the "Fail-safe" area of the properties of an F-CPU.

⚠ WARNING

The F-BaseID may only be assigned manually in exceptional cases, when you are sure that the PROFIsafe addresses of the F-I/Os of the F-CPU are unique (worldwide). F-I/Os with the same F-BaseID must differ in the combination of F-source and F-destination address. We explicitly advise against manual assignment and recommend generation with the tools offered in the TIA Portal.

With the manual assignment of value "0" as F-BaseID, you lose the advantages of PROFIsafe address type 3 and must adhere to the specifications for PROFIsafe address type 2 (see warning S052). *(S115)*

You can also generate the F-BaseID using tools outside of the TIA Portal and then assign it manually. However, observe the following warning in this case:

⚠ WARNING

Only use generation tools with which you can track adherence to the PROFIsafe generation rules, e.g. based on the documentation. The TIA Portal tools comply with these rules. *(S116)*

If you select the "Activate delayed transfer of F-BaseID" option, a specific value (0x7C6) is set for the F-BaseID which informs the F-CPU and all F-I/Os (of PROFIsafe address type 3) assigned to it that the F-BaseID will be determined and possibly re-generated on startup of the F-CPU. It is not possible to operate the F-I/Os with this F-BaseID (0x7C6). You can find more information on this in the appendix "Assignment of the F-BaseID during runtime (delayed transfer) (S7-1200 G2, S7-1500) (Page 643)".

Rules for address assignment

In a homogeneous network in which all F-I/O have PROFIsafe address type 3 and a generated F-BaseID, the following conditions must be met:

- The F-destination address of the F-module is unique CPU-wide.
- The F-BaseID has been generated individually for the F-CPU according to the requirements of the PROFIsafe specification.

The F-destination address is automatically assigned uniquely CPU-wide for supported configurations by the TIA Portal (normally in descending order from 65534) and there is an

automatic check that the F-destination address is unique CPU-wide in the event of changes. If you have the F-BaseID generated in the TIA Portal, the requirements of the PROFIsafe specification are adhered to.

⚠ WARNING

If you are uncertain whether the F-BaseID is unique, generate a new F-BaseID. *(S117)*

Note that F-I/Os of PROFIsafe address type 3 with inactive F-BaseID (value = 0) should be viewed the same as F-I/Os of PROFIsafe address type 2.

In a heterogeneous network with F-I/O of PROFIsafe address types 2 and 3, the F-source address ("Central F-source address" parameter) of the F-CPUs ensures that the PROFIsafe addresses are unique network-wide. Therefore, for the F-I/Os of PROFIsafe address type 3, you can use the F-BaseID with the value "0", but you must consider the network boundaries and the conditions for the F-destination address (see warning S052) for PROFIsafe address type 2. To be future-proof, generate an F-BaseID for the F-I/Os of PROFIsafe address type 3.

We recommend that you generate the F-BaseID directly before downloading the fail-safe project data to a productive system for the first time. In general, you are prompted at this time for the serial number of the productive F-CPU, and the PROFIsafe addresses of the F-I/Os of PROFIsafe address type 2 and 3 are not yet assigned. If you assign them after downloading, the newly generated F-BaseID is automatically also assigned.

Even with a generated F-BaseID, uniqueness is not automatically guaranteed. For example, when copying a TIA project or an SD card with loaded fail-safe project data, the F-BaseIDs contained therein are also copied. It is your responsibility to generate new F-BaseIDs in each copy after the copying in order to restore their uniqueness.

⚠ WARNING

For copy operations that also copy the value of the F-BaseID, e.g. copying the project or copying the SD card, we strongly recommend generating a new F-BaseID. Otherwise, you must ensure the uniqueness of the PROFIsafe addresses for F-I/Os in another way. *(S114)*

Projects in which you have activated the delayed transfer for the F-BaseID are an exception. In these projects, a valid F-BaseID is generated using the instructions made available by the TIA Portal on the first startup of the F-CPU. You can copy the SD card as much as you like before this first startup. Copying the TIA Portal project data is also possible later. Therefore, the delayed transfer is especially suitable for creating series machines.

2.13 Setting the F-destination address for F-I/O with DIP switches

Information on how to set the F-destination address for F-I/O with DIP switches is available in the documentation of the respective F-I/O.

2.14 Assigning a PROFIsafe address of the F-I/Os with SIMATIC Safety

Introduction

You assign the PROFIsafe address of PROFIsafe address type 2 (Page 68) or PROFIsafe address type 3 (Page 70), consisting of F-source address and F-destination address and, optionally for PROFIsafe address type 3, the F-BaseID for the following F-I/Os directly from *STEP 7 Safety*:

- ET 200SP fail-safe modules
- ET 200SP HA fail-safe modules
- S7-1500/ET 200MP fail-safe modules
- ET 200AL fail-safe I/O modules
- ET 200eco PN fail-safe I/O modules

Refer to the equipment manual of the respective F-I/O for the exact PROFIsafe address type.

The PROFIsafe addresses for S7-1200 F-modules are automatically assigned during download of the hardware configuration.

None of these F-I/O modules have a DIP switch for setting the unique F-destination address for each module.

A new assignment is required for the F-modules mentioned above in the following situations:

- Later placement of a fail-safe module during initial commissioning (not for ET 200eco PN)
- Intentional modification of the F-destination address
- Modification of the "Central F-source address" parameter for the associated F-CPU (changes the F-source address).
- Modification of the "F-BaseID" parameter of the associated F-CPU
- Replacement of the electronic coding element
- Commissioning of a mass-produced machine

In the following cases, a reassignment is not required for fail-safe ET 200SP modules, fail-safe ET 200SP HA modules and fail-safe S7-1500/ET 200MP modules:

- POWER OFF/POWER ON
- Replacement of an F-module (repair) without PG/PC
- Replacement of the BaseUnit (transfer of electronic coding element with assigned F-source address and F-destination address and, if present, F-BaseID to the new BaseUnit)
- Replacement of a BaseUnit without electronic coding element
- Changes in the design in case a new BaseUnit is inserted in front of a fail-safe module
- Repair/replacement of the interface module

Reassignment is not required for fail-safe ET 200eco PN and ET 200AL I/O modules in the following cases:

- POWER OFF/POWER ON
- Replacement of the compact device (transfer of electronic coding element with assigned F-source address and F-destination address and, if present, F-BaseID to the new compact device)

Basic procedure

NOTE

Assigning the PROFIsafe address for S7-1200 fail-safe modules

The procedure described below for identifying and assigning the PROFIsafe addresses is not required for S7-1200 fail-safe modules.

Note that an S7-1200 must not include an additional unconfigured F-module.

1. Set the F-destination address (Page 68) and F-source address (Page 68) and, optionally for F-I/Os of PROFIsafe address type 3, the F-BaseID (Page 70) in the hardware configuration in *STEP 7 Safety*.
2. Identify the F-modules ET 200SP, ET 200SP HA, S7-1500/ET 200MP, F-I/O modules ET 200AL or F-I/O modules ET 200eco PN to which you want to assign the PROFIsafe address.
3. Assign the PROFIsafe address to the F-I/Os.

Assigning the PROFIsafe address for F-I/Os via a redundant S7-1500HF system

In the "Assign PROFIsafe address" dialog under "Select PLC of HF-System", select the HF-CPU of the redundant S7-1500HF system via which you want to assign the PROFIsafe address.

The following table uses the system status to show you via which HF-CPU of the redundant S7-1500HF system you can assign the PROFIsafe address.

System status/F-CPU role	Primary	Backup
RUN-Redundant	x	x
RUN-Solo	x	-
STOP	x	-

x: Assignment possible, -: Assignment not possible

NOTE

Use of R1 devices

Assigning the PROFIsafe address for F-I/Os in an R1 device is only possible via the left interface module of the R1 device. Therefore, the left interface module is always displayed in the "Assign PROFIsafe address" dialog. The left interface module must be supplied with voltage for the assignment of the PROFIsafe address.

If the two HF-CPUs of the redundant S7-1500HF system are located in two separate subnets, the PG and the HF-CPU via which the assignment of the PROFIsafe address is to be performed must be located in the same subnet.

2.14.1 Identifying F-modules

Requirement

The following requirements must be met:

- The F-CPU and fail-safe modules are configured.
- When using an ET 200SP Open Controller, the hardware configuration of the ET 200SP Open Controller **and** of the fail-safe Software Controller must be downloaded.
- The F-CPU and fail-safe modules can be reached online.

⚠ WARNING

Make sure that the latest hardware configuration has been downloaded to the F-CPU before identification. By clicking the "Identification" button and then clicking on "Assign PROFIsafe address", you confirm the correctness of the PROFIsafe addresses for the F-I/Os.

Therefore, proceed carefully when confirming the F-I/Os by LED flashing or the serial number of the F-CPU with central F-I/Os or the serial number of the interface module with F-I/Os.

To do this, you must read and write down the serial number directly at the F-CPU* or the interface module**. Reading out the serial number via the online and diagnostics view of the TIA Portal is not permitted.

With F-I/O of PROFIsafe address type 3, you need to confirm the assignment by flashing or serial number even if you have only changed the F-BaseID.

A special rule applies for F-I/Os of the ET 200AL distributed I/O system. For these F-I/Os, only identification by LED flashing is permissible and only exactly one F-I/Os at a time may be selected for assignment of the PROFIsafe address.

* When an ET 200SP Open Controller is used, you must read and write down the serial number from the display of the S7-1500 Fail-safe Software Controller in menu "Overview > CPU".

** When an R1 device is used, you must read and write down the serial number of the left interface module. *(S046)*

Procedure

Proceed as follows to identify the F-modules:

1. Establish an online connection to the F-CPU to which the F-modules are assigned.
2. In the network view, select
   - the F-CPU with F-modules,
   - the interface module with F-modules,

   to which you want to assign PROFIsafe addresses.

   Alternatively, you can select only a single F-module.
3. Select "Assign PROFIsafe address" from the shortcut menu.

   NOTE

   If you integrate a fail-safe ET 200AL module into a configuration of the ET 200SP distributed I/O system (mixed configuration ET 200AL/ET 200SP with ET connection), you must open the dialog "Assign PROFIsafe address" in the device view of the ET 200SP separately for each fail-safe ET 200AL module.

4. Under "Assign PROFIsafe address by", select the method to be used for identifying the F-modules.
   - "Identification by LED flashing"

     This is the default setting. The DIAG and STATUS LEDs of the F-modules to be identified flash upon identification.
   - "Identification by serial number"

     If you cannot see the fail-safe modules directly, you can identify the fail-safe modules by the noted serial number of the F-CPU or interface module.

     NOTE

     The displayed serial number may be amended with a year number compared to the serial number printed on the F-CPU or the interface module. The serial numbers are nevertheless identical.

5. With identification in the "Assign" column, select all F-modules to which you want to assign the PROFIsafe address.

   If you select the F-CPU or the interface module in the "Assign" column, all F-modules of the station are selected.

   NOTE

   For F-I/Os of PROFIsafe address type 3 for which the F-BaseID is set to "Delayed transfer", there can be no identification or assignment (see section PROFIsafe addresses for F-I/O of PROFIsafe address type 3 (Page 70)). For identification and assignment of the F-source address and F-destination address only, the value "0" must be configured and downloaded for the F-Base ID of these F-I/Os.

6. Click the "Identification" button. Check whether the DIAG and STATUS LEDs for the F-modules whose PROFIsafe address you want to assign are flashing green. If you identify using the serial number, compare the displayed serial number to the noted serial number of the F-CPU with central fail-safe modules or the interface module with fail-safe modules.

   Observe the following behavior for certain F-modules and configuration variants:

   – If you have configured more F-modules than actually exist, a dialog is displayed. In this dialog, enter the number of F-modules actually existing and confirm the dialog.
   – If you have configured fewer F-modules than are actually available, an online-offline difference is shown and the assignment of the PROFIsafe address is not possible.

2.14.2 Assign PROFIsafe address

Requirement

The F-modules have been successfully identified.

Procedure

To assign a PROFIsafe address, proceed as follows:

1. In the "Confirm" column, select all the F-modules to which you want to assign the PROFIsafe address.
2. Use the "Assign PROFIsafe address" button to assign the PROFIsafe address to the fail-safe modules. You may have to enter the password of the F-CPU or need the user right "Full access including fail-safe".

   To assign the PROFIsafe address, you must confirm the "Confirm PROFIsafe address assignment" dialog within 60 seconds.

2.14.3 Assigning PROFIsafe address to an F-module in a shared device in multiple projects

Introduction

The assignment of the PROFIsafe address for a F-module in a shared device in different projects can only take place after the selection of the IO Controller (F-CPU) to which the F-module is assigned

Therefore, only the F-modules that are assigned to an IO controller (F-CPU) of this project can be selected for assigning the PROFIsafe address in the "Assign PROFIsafe address" dialog.

Requirement

The following requirements must be met:

- All requirements from the sections "Identifying F-modules (Page 75)" and "Assign PROFIsafe address (Page 77)".
- The hardware configuration of all IO controllers (F-CPUs) to which an F-module from the shared device has been assigned is loaded.
- All IO controllers (F-CPUs) to which an F-module from the shared device has been assigned and the shared device must be located in the same subnet.

Procedure

To identify the F-modules and assign the PROFIsafe address, proceed as described in sections "Identifying F-modules (Page 75)" and "Assign PROFIsafe address (Page 77)".

If the F-modules in a shared device are assigned to several IO controllers (F-CPUs) in different projects, make the assignment from the different projects one after the other.

See also

Configuring shared device (Page 61)

2.14.4 Assigning PROFIsafe address to an F-module in a shared device in a common project

Introduction

The assignment of the PROFIsafe address for a F-module in a shared device in a common project can only take place after the selection of the IO Controller (F-CPU) to which the F-module is assigned.
Therefore, only the F-modules that are assigned to the selected IO controller (F-CPU) can be selected for assigning the PROFIsafe address in the "Assign PROFIsafe address" dialog.

Requirement

The following requirements must be met:

- All requirements from the sections "Identifying F-modules (Page 75)" and "Assign PROFIsafe address (Page 77)".
- The hardware configuration of all IO controllers (F-CPUs) to which an F-module from the shared device has been assigned is loaded.
- All IO controllers (F-CPUs) to which an F-module from the shared device has been assigned and the shared device must be located in the same subnet.

Procedure

To identify the F-modules and assign the PROFIsafe address, proceed as described in sections "Identifying F-modules (Page 75)" and "Assign PROFIsafe address (Page 77)".

If the F-modules in a shared device in a common project are assigned to several IO controllers (F-CPUs), select an IO controller in the "Assign PROFIsafe address" dialog and first carry out the assignment for the F-modules that are assigned to the selected IO controller. Subsequently you select further IO controllers and proceed accordingly.

See also

Configuring shared device (Page 61)

2.14.5 Changing the PROFIsafe address

Changing the PROFIsafe address

NOTE

Keep in mind that after changing the PROFIsafe address of an F-I/O you must also carry out an acceptance (Page 418) including a check of your change (Page 433) in the safety summary (Page 388).

1. Change the PROFIsafe address (F-destination address, F-source address and, if present, F-BaseID) in the hardware configuration.
2. Compile the hardware configuration.
3. Download the hardware configuration to the F-CPU.
4. Select "Assign PROFIsafe address" from the shortcut menu.
5. Proceed as described under Identifying F-modules (Page 75) and Assign PROFIsafe address (Page 77).

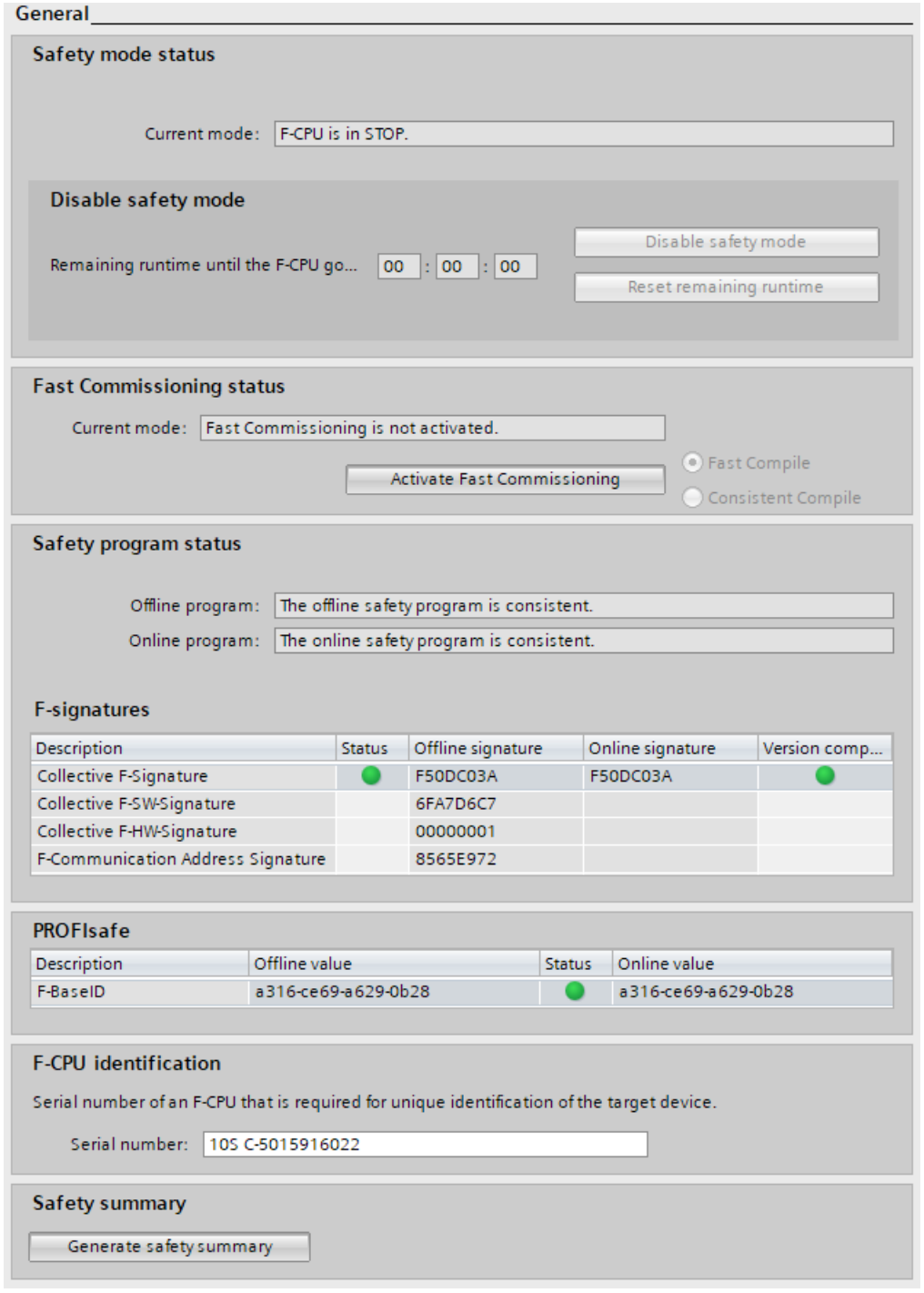
Safety Administration Editor

3

Overview

The *Safety Administration Editor* supports you as follows:

- Displaying of status of safety mode
- Disabling safety mode
- (S7-1200, S7-1500) Displaying and changing the Fast Commissioning Status
- Displaying of status of the safety program
- Displaying of Collective F-Signature
- (S7-1200, S7-1500) Show Collective F-SW-Signature
- (S7-1200, S7-1500) Show Collective F-HW-Signature
- (S7-1200, S7-1500) F-communication address signature
- (S7-1200 G2 as of firmware V4.1, S7-1500 as of firmware V4.1, S7-1500 F Software Controller as of V40.1) Display F-BaseID
- (S7-1200, S7-1500) Specify/change serial number for F-CPU identification
- Creating a safety summary
- Creating and organizing of F-runtime groups
- Displaying information on the F-blocks
- Displaying information about F-compliant PLC data types (UDT)
- Specifying/changing access protection
- (S7-1200 G1, S7-1500) Information for users with F-Admin permission
- Set/modify settings for the safety program, e.g. Enable F-change history
- (S7-1200, S7-1500) Create/display/delete F-communications via Flexible F-Link


General
Safety mode status
Current mode:
F-CPU is in STOP.
Disable safety mode
Remaining runtime until the F-CPU go...
00 : 00 : 00
Disable safety mode
Reset remaining runtime
Fast Commissioning status
Current mode:
Fast Commissioning is not activated.
Activate Fast Commissioning
Fast Compile
Consistent Compile
Safety program status
Offline program:
The offline safety program is consistent.
Online program:
The online safety program is consistent.
F-signatures
Description
Status
Offline signature
Online signature
Version comp...
Collective F-Signature
F50DC03A
F50DC03A
Collective F-SW-Signature
6FA7D6C7
Collective F-HW-Signature
00000001
F-Communication Address Signature
8565E972
PROFIsafe
Description
Offline value
Status
Online value
F-BaseID
a316-ce69-a629-0b28
a316-ce69-a629-0b28
F-CPU identification
Serial number of an F-CPU that is required for unique identification of the target device.
Serial number:
10S C-5015916022
Safety summary
Generate safety summary

The *Safety Administration Editor* is divided into the following areas:

- General

  Under "General" the following is displayed:

  – Safety mode status
  – Fast Commissioning Status
  – Safety program status
  – F-signatures
  – PROFIsafe
  – F-CPU identification
  – Safety summary

  More information on the "General" area can be obtained in ""General" area of the Safety Administration Editor (Page 83)".

- F-runtime group

  You define the blocks and properties of an F-runtime under "F-runtime group".

  You can find information on F-runtime groups at ""F-runtime group" area (Page 87)".

- F-blocks

  Under "F-blocks", you can find information on the F-blocks used in your safety program and their properties . More information on the "F-blocks" area can be obtained in ""F-blocks" area of the Safety Administration Editor (Page 89)".

- F-compliant PLC data types

  Under "F-compliant PLC data types", you obtain information on the created F-compliant PLC data types (UDT). There you also obtain information whether or not an F-compliant PLC data type (UDT) is used in the safety program. More information on "F-compliant PLC data types" can be found in ""F-compliant PLC data types" area of the Safety Administration Editor (S7-1200, S7-1500) (Page 90)".

- Access protection

  Under "Access protection", you can set up, change, or revoke the password for the safety program. Alternatively, you can set up access protection for safety-related project data via project protection. Access protection is mandatory for productive operation. More information on access protection can be found in "CPU-wide access protection for the safety-related project data (Page 100)".

- Web server F-Admins

  Under "Web server F-admins", you obtain information on users with the F-Admin attribute for the Web server of the F-CPU. More information on the "Web server F-Admins" area can be obtained in ""Web server F-admins" area of the Safety Administration Editor (S7-1200 G1, S7-1500) (Page 91)".

- Settings

  Under "Settings", you set the parameters for the safety program. Information on the settings for your safety program can be found in ""Settings" area of the Safety Administration Editor (Page 91)".

- Flexible F-Link

  In the "Flexible F-Link" area, you will find information about the configured F-communications via Flexible F-Link in tabular form and create F-communications. You can obtain information at ""Flexible F-Link" area of the Safety Administration Editor (S7-1200, S7-1500) (Page 97)".

See also

Program structure of the safety program (S7-1200, S7-1500) (Page 113)

Program structure of the safety program (S7-300, S7-400) (Page 111)

Defining F-Runtime Groups (Page 133)

3.1 Opening the Safety Administration Editor

Requirement

The *Safety Administration Editor* is visible as an element in the project tree, if you have configured a CPU as an F-CPU in the project, which means the "F-capability activated" option must be selected (in the properties of the F-CPU).

Procedure

To open the *Safety Administration Editor*, follow these steps:

1. Open the folder for your F-CPU in the project tree.
2. Double-click on "Safety administration" or right-click and select the corresponding shortcut menu for the *Safety Administration Editor*.

Result

The *Safety Administration Editor* for your F-CPU opens in the work area.

3.2 "General" area of the Safety Administration Editor

"Safety mode status"

The "Safety mode status" shows the current status of safety mode. The prerequisite is an existing online connection to the selected F-CPU.

The following statuses are possible:

- "Safety mode is activated"
- "The safety mode is not activated"
- "F-CPU is in STOP"
- "No active F-CPU available"
- "F-runtime group was not called"
- "The safety program is not called"
- "(No online connection)"

"Disable safety mode"

For existing online connection and active safety mode operation, you have the option of using the "Disable safety mode" button to disable safety mode for the selected F-CPU. Safety mode can be deactivated only for the entire safety program and not for individual F-runtime groups.

As of Safety system version V2.4, disabled Safety mode for S7-1200/1500 F-CPUs is time-limited.

To use Fast Commissioning mode, make sure that the option "Safety mode can be disabled" is selected in the "Settings" area (Page 91).

For more information, refer to the section "Disabling safety mode (Page 393)".

"Fast Commissioning Status"

To use the Fast Commissioning mode, make sure that the option "Safety mode can be disabled" is selected in the "Settings" area (Page 91) and that the safety program was compiled consistently. The Fast Commissioning mode can be executed with "Fast Compile" (default setting) or with "Consistent Compile". For more information, refer to the section "Safety program in RUN mode (S7-1200, S7-1500) (Page 406)".

"Safety program status"

"Safety program status" displays the current status of your online and offline program.

The following statuses are possible:

- Consistent

  With information if no password has been assigned. This information does not apply when project protection is used.
- Inconsistent
- Modified

- Fast Commissioning mode is active

  If Fast Commissioning mode was activated with "Fast Compile" and a change was made in the program.
- (No online connection)

  If no connection to the online program could be established.

"F-signatures"

When no online connection exists

Under "F-signatures" multiple signatures are displayed. Each signature is formed from different parts of the fail-safe project data.

- Collective F-Signature: This signature changes with each change of the fail-safe project data. It can contain the signatures described below.
- Collective F-SW-Signature (S7-1200, S7-1500): This signature changes with each change of the safety program.
- Collective F-HW-Signature (S7-1200, S7-1500): This signature changes with each change of the fail-safe HW configuration.
- F-Communication address signature (S7-1200, S7-1500): This signature changes with each change of the name or the F-communication UUID of communication connections with Flexible F-Link.

Note that the F-BaseID does not enter into the Collective F-HW-Signature or the Collective F-Signature.

The time of the last compilation process is displayed for the Collective F-Signature in the "Time stamp" column.

When an online connection exists

For an existing online connection, the following is displayed under the "Program signature":

- The status of safety program

Status	Meaning
	The online and offline Collective F-Signatures match.
	The online and offline Collective F-Signatures *do not* match.
—	The safety program status could not be determined.

- The online and offline Collective F-Signatures

- When the Collective F-Signatures match: Information on whether the F-block versions are consistent online and offline.

Status	Version comparison	Statement
	Not relevant	The online and offline Collective F-Signatures *do not* match.
		The online and offline Collective F-Signatures match, but the online versions of F-blocks differ from the offline versions.
		The online and offline Collective F-Signatures match, identical versions of F-blocks are being used online and offline.
Not relevant	—	The safety system versions could not be determined.

You can find more information on the consistency of the online safety program at under Identity of online and offline program (Page 430).

"PROFIsafe" (S7-1200 G2, S7-1500)

When no online connection exists

The F-BaseID is displayed under "PROFIsafe". It applies to all F-I/Os of PROFIsafe address type 3 (Page 35) assigned to the F-CPU. The F-BaseID allows a clear differentiation to be made between F-I/Os of PROFIsafe address type 3 having the same F-source address and F-destination address.

More information on the F-BaseID can be found in section "Configuring an F-CPU (Page 41)".

For S7-1200/1500 F-CPUs that do not support the F-BaseID, this area is grayed out.

When an online connection exists

When an online connection exists, the following is also displayed under "PROFIsafe":

Table 3-1 Status of the F-BaseID

Status	Meaning
	The online and offline F-BaseIDs match.
	The online and offline F-BaseIDs do *not* match.
	The F-BaseID could not be determined online or is not supported.

The online-offline comparison of the F-BaseID does not enter into the status of the "General" area.

"F-CPU identification" (S7-1200, S7-1500)

The serial number of the F-CPU to which the safety program is to be downloaded is entered under "F-CPU identification". This serial number is used to ensure connection to the correct F-CPU in the network. If the serial number is not entered at the time of download to the F-CPU, you must confirm it in the "Load preview" dialog to be able to continue the download. The confirmed serial number is then automatically entered in this field.

"Safety summary"

Under "Safety summary", you have the option to create the safety summary (Page 388) for the F-CPU.

See also

Program identification (Page 384)

3.3 "F-runtime group" area of the Safety Administration Editor

3.3.1 "F-runtime group" area

A safety program consists of one or two F-runtime groups.

General information on F-runtime groups can be found in "Program structure of the safety program (S7-300, S7-400) (Page 111)" and "Program structure of the safety program (S7-1200, S7-1500) (Page 113)".

You can find information on creating F-runtime groups at Defining F-Runtime Groups (Page 133)

(S7-1200, S7-1500) "Creating a global F-I/O status block"

You can create a standard block (FB) with the name "RTGx_GLOB_FIO_STATUS", which evaluates whether substitute values are output instead of process values for at least one F-I/O or at least one channel of an F-I/O of an F-runtime group x. The result of the evaluation is available at the "QSTATUS" output.

The following are not included:

- F-I/O that you disabled with the DISABLE tag in the F-I/O DB.
- Channel error of channels that you disabled when configuring an F-I/O.

The "RIOforFA_VALUE_STATUS" output is set to "1" if no substitute value is output for **any** F-I/O **with "RIOforFA-Safety"** profile of the F-runtime group x.

To generate this standard FB, you use the "Create global F-I/O status block" button. You can only create the standard FB when your safety program has been compiled. You can call the standard FB anywhere in your standard user program.

NOTE

When adding or deleting an F-I/O, you have to generate "RTGx_GLOB_FIO_STATUS" again.

See also

Process Data or Fail-Safe Values (Page 162)

3.3.2 Pre-/postprocessing (S7-1200, S7-1500)

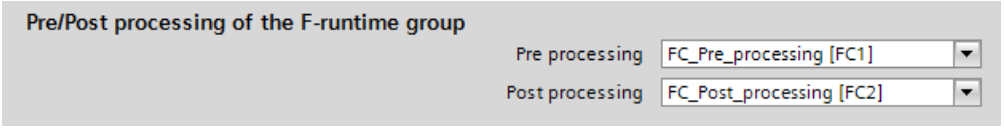
With pre-processing and post-processing you have the option of calling standard blocks (FCs) directly before or after an F-runtime group, for example for data transfer with fail-safe communication via Flexible F-Link (Page 342) or OPC UA Safety (Page 286). The calls are made in the same F-OB or synchronized with the F-runtime group.

Requirement

- Only standard-FCs usable.
- Only temporary local data and constants are permitted in the block interface of a standard-FC.

Procedure

1. Create the standard-FCs for the pre-processing and the post-processing.
2. Assign the standard-FCs in the Safety Administration Editor under "Pre-/post-processing of the F-runtime group".



NOTE

When you delete an assigned FC or overwrite it by copying, its selection as a pre-processing / post-processing block is automatically reset.

Effect on the safety program

- The runtime of the F-runtime group is extended by the runtime of the standard FCs for pre-/post-processing (influence on TRTG_CURR and TRTG_LONG in the F-runtime group information DB).
- Because the pre-processing / post-processing does not change the functionality of the safety program, the collective F-signature remains unchanged after compilation.

Load behavior

The calls of the selected standard FCs are placed during compiling or after the call of the main safety block in the F-OB.
This means that the STOP operating state is required during a subsequent download.

Changes in the contents at the selected standard FCs can take place in RUN.

Exceptions are changes of the block name and block numbers which also include the compilation of the safety program.

When a pre-processing/post-processing block is uploaded individually by the F-CPU, it does not automatically connect to the F-runtime group in the *Safety Administration Editor*.

If consistent loading of the F-CPU into the PG/PC is performed instead, the settings for pre-processing and post-processing are updated according to the online CPU.

3.4 "F-blocks" area of the Safety Administration Editor

Overview

The "F-Blocks" area helps you in the following tasks:

- Displaying the F-blocks used in your safety programs.
- Displaying the F-blocks used in the F-runtime groups.
- Displaying additional information about the F-blocks.

The F-blocks are hierarchically displayed as in the "Program blocks" folder.

A description of the F-blocks is available in "Creating F-blocks (Page 150)".

Displayed information

The following information is displayed in offline mode:

- Have displayed F-blocks been compiled and used?
- Function of F-blocks in safety program
- Block signature
- Group signature
- Change time stamp of the F-blocks

The following information is displayed in online mode:

- Status (whether block has the same time stamp online and offline)
- Function of F-block in the safety program
- Block signature offline
- Group signature offline
- Block signature online
- Group signature online

NOTE

During the offline-online comparison, the comparison statuses may occasionally differ between the *comparison editor* and status display in the *Safety Administration Editor*. The decisive status is the result of the comparison in the *comparison editor*, since this is the only comparison that takes into account the contents of the F-blocks.

NOTE

Group signatures are formed from the signatures of the F-blocks contained in the group and in subgroups. Only F-blocks that are used in the safety program are taken into account. Only the group signatures up to the 6th hierarchy level are displayed.

Filter function



Using the filter function, you can select whether you want to view all F-blocks of a certain F-runtime group or the entire safety program.

- Select "All F-blocks " from the drop-down list to view all F-blocks.
- Select an F-runtime group from the drop-down list to see all F-blocks of this F-runtime group.

3.5 "F-compliant PLC data types" area of the Safety Administration Editor (S7-1200, S7-1500)

Overview

Under "F-compliant PLC Data Types" you obtain information on the F-compliant PLC data types (UDT) you have defined.

You can delete F-compliant PLC data types (UDT) from the shortcut menu.

More information on the F-compliant PLC data types (UDT) can be found in "F-compliant PLC data types (UDT) (S7-1200, S7-1500) (Page 126)".

Displayed information

The following information is displayed for F-compliant PLC data types (UDT) in offline mode:

- Is the F-compliant PLC data type used in the Safety program?
- Time stamp of the last change.

The following information is displayed for F-compliant PLC data types (UDT) in online mode:

- Status (whether the F-compliant PLC data types (UDT) have the same time stamp online and offline)

The F-compliant PLC data types (UDT) are displayed hierarchically as in the folder "PLC Data Types".

Double-click the F-compliant PLC data type (UDT) to open it for editing.

More information can be found on the symbols in the "Status" column at "Comparing Safety Programs (Page 386)".

NOTE

During offline-online comparison, the comparison statuses between the *comparison editor* and status display in the *Safety Administration Editor* can be different under certain circumstances. The comparison result in the *comparison editor* is decisive, since this is the only comparison that takes into account the contents of the F-compliant PLC data types (UDT).

3.6 "Web server F-admins" area of the Safety Administration Editor (S7-1200 G1, S7-1500)

The "Web server F-Admins" area is displayed for S7-1500 F-CPUs only for firmware < V3.1 and for S7-1200 G1 F-CPUs only for firmware <V4.7.

You need "F-admin" rights in order to restore a backup (Page 374) via the Web server or Web API interface of your F-CPU. You assign the "F-admin" right in the hardware configuration of the F-CPU under the user management of the Web server.

In this section, you obtain information on which users have the "F-admin" right online or offline for F-CPUs that support this right. You can see from this whether a change to the "F-admin" right is active on the F-CPU. In order to make a change to the "F-admin" right effective, you have to load the configuration to the F-CPU.

If you are using user management, this area is omitted.

See also

Completeness and correctness of the hardware configuration (Page 418)

3.7 "Settings" area of the Safety Administration Editor

"Number ranges of the generated F-system blocks"

The number ranges assigned here are used by the F-System for new, automatically generated F-blocks.

At this point, you can select whether the number ranges are managed by the F-system or if a fixed range specified by you is used.

- "F-system managed"

  The number ranges are managed automatically by the F-system, depending on the F-CPU used. The F-system selects an available number range. The start and end ranges of the number ranges are displayed.

- "Fixed range"

  You can select the start and end ranges of the number ranges from the available range. The available range depends on the F-CPU used.

  An invalid number range selection is indicated by an error message.

  The only check performed during configuration is whether the configured low limit is less than or equal to the high limit and within the available range of the F-CPU. The check as to whether the configured range is sufficiently large is first made during compiling. You need

to ensure a sufficiently large range. Where the available range is insufficient, a compiling error occurs. Not all blocks are generated and the safety program is not executable.

Changes will become valid only during the next compilation. The automatically created F-blocks may be moved into the new area during compilation. The F-I/O DBs are an exception. They always retain their original number that you may change in the properties of the F-I/O.

"Safety system version"

This parameter is used to specify the safety system version (including version of the F-system blocks and automatically generated F-blocks, see Overview of Programming (Page 111)).

A number of versions are available:

Version	S7-300/4-00	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.6	—	x	—	x	These versions have identical functions. Depending on the set version, the result may be different runtimes of the F-runtime group(s) (see Excel file for response time calculation on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109783831)).
2.0	x	x[1]	—	x[2]	
2.1	—	x[1]	—	x[2]	Additionally supports the variables "DISABLE" and "DISABLED" in the F-I/O DB
2.2	—	x[1]	—	x[2]	Supports the safety-related CPU-CPU communication and F-runtime group communication with Flexible F-Link.
2.3	—	x[1]	—	x[2]	This version has identical functions to version 2.2.
2.4	—	x[1, 3]	—	x[2]	Additionally supports: • F-CPUs • Fast Commissioning mode with "Fast Compile" • Runtime for the deactivated safety mode
2.5	—	x	—	x[4]	Additionally supports: • Fast Commissioning mode with "Consistent Compile" (S7-1500) • F-I/Os with more than 13 bytes net data (e.g. F-AI 8xI 0(4)..20mA)
2.6	—	x	—	x	The functions of this version are identical to those of version 2.5.
2.7	—	x	x	x	Additionally supports: • S7-1500 F Virtual Controller • OPC UA Safety
2.8	—	x	x[5]	x[5]	Additionally supports: • F-BaseID[5] • Improved response time of F-I/O for inputs[6]

[1] Supported as of firmware V4.2

[2] Supported as of firmware V2.0

[3] Fast Commissioning mode with "Fast Compile" is supported as of firmware V4.5.

[4] Fast Commissioning mode with "Consistent Compile" is supported as of firmware V3.0. With the S7-1500 F-Software Controller, it is available only for approved IPCs as of firmware V30.0.

[5] The F-BaseID is supported for S7-1200 G2 and S7-1500 as of firmware V4.1 or F-Software Controller as of V40.1.

[6] If the F-OB of the associated F-runtime group has the event class "Cyclic interrupt".

Usually, you do not need to make any settings for this parameters.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

"Local data used in safety program" (S7-300, S7-400)

You use this parameter to specify the amount of temporary local data (in bytes) that is available for the call hierarchy below the main safety block.

The setting applies to each F-runtime group of a safety program. More information on F-runtime groups can be found in "Program structure of the safety program (S7-1200, S7-1500) (Page 113)" and "Program structure of the safety program (S7-300, S7-400) (Page 111)".

The **minimum possible amount** is determined by the local data requirement of the F-blocks generated automatically when the safety program is compiled.

For this reason, you must provide at least 440 bytes. However, the local data requirement for the automatically added F-blocks may be higher depending on the local data requirement of the F-blocks you created with FBD or LAD.

Therefore, provide as much local data as possible. If there is not enough local data available for the automatically added F-blocks (440 bytes or more), the safety program will be compiled nevertheless.

Data in automatically added F-DBs are then used instead of local data. However, this increases the runtime of the F-runtime group(s). You will receive a notice when the automatically added F-blocks require more local data than configured.

⚠ WARNING

The calculated maximum runtime of the F-runtime group using the Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) is no longer correct in this case because the calculation assumes sufficient availability of F-local data.

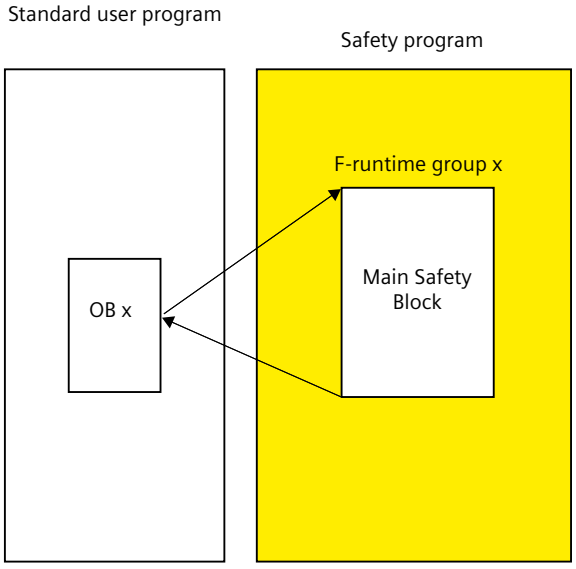
In this case, use the value you configured for the maximum cycle time of the F-runtime group (F-monitoring time) as the maximum runtime of the F-runtime group when calculating the maximum response times in the event of an error and for any runtimes of the standard system using the above-mentioned Excel file. *(S004)*

The maximum possible amount depends on:

- Local data requirement of the main safety block and the higher-level standard user program. For this reason, you should call the main safety blocks directly in OBs (cyclic interrupt OBs, whenever possible), and additional local data should not be declared in these cyclic interrupt OBs.
- Maximum volume of local data of the utilized F-CPU (see Technical Specifications in the product information for the utilized F-CPU). For S7-400 F-CPUs, you can configure the local data for each priority class. Therefore, assign the largest possible local data volume for the priority classes in which the safety program (the main safety blocks) is called (e.g., OB 35).

Maximum possible amount of local data as a function of local data requirement of main safety block and higher-level standard user program (S7-300, S7-400):

Case 1: Main safety block called directly from OBs

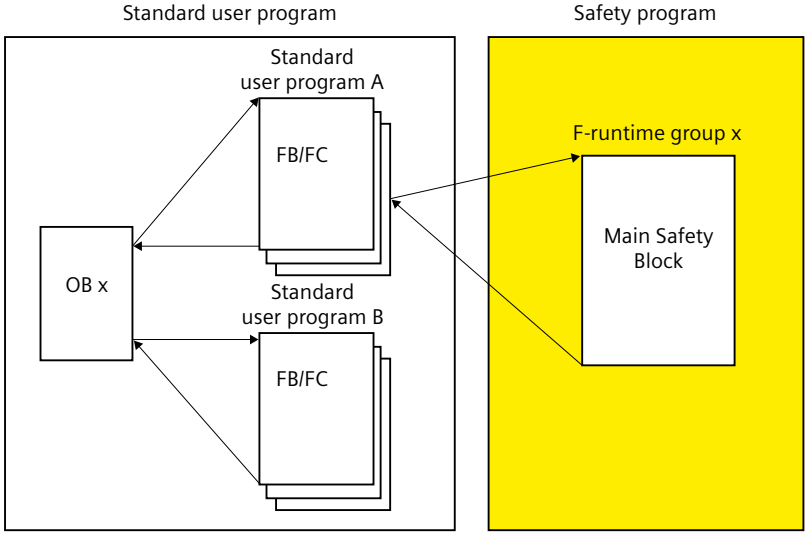


Set the "Local data used in safety program" parameter to the maximum amount of local data of the utilized F-CPU minus the local data requirement of the main safety block (if the main safety block has 2 F-runtime groups, use the largest local data requirement) and minus the local data requirement of the calling OBx (if there are 2 F-runtime groups, use the OB with the largest local data requirement).

Note: If you have not declared any temporary local data in the main safety blocks and calling OBx, the local data requirement of the main safety blocks is 6 bytes and the local data requirement of the calling OBx is 26 bytes. You can derive the local data requirement of the main safety blocks and calling OBx from the program structure.

Select the utilized F-CPU in the project tree and then "Tools > Call structure". The table gives the local data requirement in the path or for the individual blocks (see also the help on *STEP 7*).

Case 2: Main safety block not called directly from OBs



Set the "Local data settings" parameter to the value calculated for Case 1, minus the local data requirement of standard user program A (if standard user program A has 2 F-runtime groups, use the largest local data requirement).

Note: You can derive the local data requirement of the standard user program A from the program structure.

Select the utilized F-CPU in the project tree and then "Tools > Call structure". The table gives the local data requirement in the path or for the individual blocks (see also the help on *STEP 7*).

"Advanced settings"

"Safety mode can be disabled"

With this option you can prevent the safety mode for a safety program from being disabled.

Under "Runtime for the deactivated safety mode", you set the time after which the F-CPU goes to STOP after safety mode is deactivated.

When you change the setting for this option, you need to recompile the safety program and download it to the F-CPU for the change to become effective. This changes the Collective F-Signature and the Collective F-SW-Signatures of your safety program.

To exclude an unintentional disabling of the safety mode, make sure that the option is disabled in productive operation.

"Enable F-change history"

Enable the logging of changes to the safety program by using the "Enable F-change history" option. For more information, refer to the section "F-change history (Page 412)".

"Enable consistent upload from the F-CPU" (S7-1500)

This option allows you to load the loaded project data (including safety-related project data) consistently from the F-CPU to the PG/PC.

The option can only be activated if the F-CPU and the firmware of the F-CPU supports the loading of the project data (including safety-related project data).

F-CPUs S7-1500 as of firmware V2.1 are supported. S7-1500 F Software Controllers are not supported.

At every change to this option you have to load the project data to the F-CPU.

Note that the activation of this option extends the loading of the safety-related project data into the F-CPU.

"Activate variable F-communication- IDs" (S7-1200, S7-1500)

If you activate this option, you can supply the DP_DP_ID input of the SENDDP or RCVDP instructions with the variable values from a global F-DB.

⚠ WARNING

The value for the respective F-communication ID (input DP_DP_ID; data type: INT) can be freely selected**; however, it must be unique network-wide* and CPU-wide**** at all times for all safety-related communication connections. You must check the uniqueness in the safety summary during acceptance of the safety program.

You must supply constant values*** to the DP_DP_ID and LADDR inputs when calling the instruction. Direct write access to DP_DP_ID and LADDR in the associated instance DB is not permitted in the safety program! *(S016)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, no connection is established at the DP_DP_ID input for a F-communication ID "0".

*** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, the DP_DP_ID input can also be supplied with variable values from a global F-DB. In this case also, you have to check during acceptance of the safety program that uniqueness is ensured at all times. You need to check the algorithm for forming the variable value accordingly. If you cannot ensure a unique F-communication ID during startup of the safety program, because it is only specified after startup of the safety program, you must make sure that the value at the DP_DP_ID input is "0" during this phase.

**** With a redundant S7-1500HF system, both F-CPUs of the redundant S7-1500HF system are to be regarded as one single F-CPU with regard to the DP_DP_ID.

"System-generated objects" (S7-1200, S7-1500)

"Creates F-I/O DBs without prefix"

When you select this option, the names of the F-I/O DBs (Page 164) are created without prefix.

"Clean up"

The "Clean up" button is provided for service and support purposes and cleans up the result of the fail-safe compilation.

3.8 "Flexible F-Link" area of the Safety Administration Editor (S7-1200, S7-1500)

In the "Flexible F-Link" area, you create new F-communications, obtain information on existing F-communications and delete F-communications.

Requirement

- S7-1500 F-CPUs as of firmware V2.0
- S7-1200 G1 F-CPUs as of firmware V4.2
- S7-1200 G2 F-CPUs as of firmware V1.0
- Safety system version as of V2.2

Information on created F-communications

In the "Flexible F-Link" area, you receive information on configured F-communications in tabular form:

- CPU-wide unique name of F-communication
- F-compliant PLC data type (UDT) for send/receive data
- Direction of F-communication: Transmitting/receiving
- F-monitoring time of F-communication
- F-Communication UUID
- Tag for send data
- Tag for receive data

Creating F-communication

1. In an empty row of the table click "<Add new>"
2. Assign a name to the communication connection.
3. Select an F-compliant PLC data type (UUID) for the communication connection.

   If you have not yet created an F-compliant PLC data type (UDT) for the communication connection or wish to create a new one, create a new F-compliant PLC data type (UDT) (Page 126) with any structure. Note that the size can be up to 1500 bytes.
4. Select the direction of the communication connection ("Send" or "Receive").
5. Select the F-monitoring time of the communication connection (Page 630).

The UUID of the F-communication is displayed via Flexible F-Link in the "F communication UUID" column. The F-communication UUID ensures sufficient uniqueness of the safety-related communication ID even across network limits.

The "Tag for send data" column shows you the newly created tag for send data of the F-communication DB (Page 346).

The "Tag for receive data" column shows you the newly created tag for receive data of the F-communication DB (Page 346).

You can find the newly created F-communication DB for this F-communication under "Program blocks\System blocks\STEP 7 Safety\F-communication DBs".

Deleting F-communication

1. Select the entire row and confirm "Delete" in the shortcut menu. You can also delete multiple F-communications at the same time.

Copying F-communication

1. Select the entire row and confirm "Copy" in the shortcut menu. You can also copy multiple F-communications at the same time.
2. With the "Paste" menu command, you can paste the copied F-communications into the table as often as needed. The UUID for the respective F-communication is retained during copying. If necessary, re-generate the UUID.

Generating a new F-communication-UUID

1. Select the entire row and confirm "Generate UUID" in the shortcut menu. You can also generate multiple UUIDs at the same time.

See also

Flexible F-Link (Page 342)

F-runtime group communication (S7-1200, S7-1500) (Page 146)

Access protection

4

Access protection is necessary for productive operation

Access protection to the SIMATIC Safety F-system is mandatory for productive operation.

⚠ WARNING

Access to the SIMATIC Safety F-system without access protection is intended for test purposes, commissioning, etc., when the system is not in productive operation. Without access protection, you need to ensure the safety of the plant through other organizational measures (Page 651).

Before you transition into productive operation, you must have set up and activated access protection. *(S005)*

NOTE

We recommend making the settings for access protection in the "PLC security settings" dialog when creating the project.

4.1 Overview of access protection

Introduction

You can control access to the SIMATIC Safety F-system with the access protection for the safety-related project data and the access protection for the F-CPU.

Access protection for the safety-related project data

Access protection for the safety-related project data (Page 100) is available in two forms:

- Assignment of a password for the safety-related project data (F-CPU-granular), assigned in the *Safety Administration Editor*
- Setup of project protection for all safety-related project data in the TIA project

Access protection for the F-CPU

The access protection (Page 103) is set at the F-CPU level. To do this, you assign a password for the F-CPU in the F-CPU configuration or assign F rights as part of user management, if supported by the respective F-CPU (only 1200/1500 F-CPUs as of certain firmware versions (Page 106)). For S7-300/400 F-CPUs, the password is also used to identify the F-CPU and must

therefore be unique network-wide. For S7-1200/1500 F-CPUs, the password no longer needs to be unique because the serial number is used for identification.

4.2 Access protection for the safety-related project data

The access protection for the safety-related project data is in effect for all operations in which the safety-related project data can be inconsistent offline, for example:

- When changing the Safety program
- When changing and deleting F-runtime groups
- When changing safety-related parameters of F-I/O

NOTE

Note that, for usability reasons, the F-BaseID falls under the offline access protection for the safety-related project data, but it is not covered by the online access protection of the F-CPU for safety-related project data. Here, the F-BaseID is treated like a standard parameter.

There is an exception for switching between an F-BaseID with the value "Delayed transfer" and an F-BaseID with a specific value (either 0 or an active F-BaseID).

- When compiling

NOTE

Note that the access protection for the safety-related project data is only effective if you are making the changes in the TIA Portal. Although the TIA Portal detects changes to the project data made directly in the related files in the file system, these changes cannot be prevented or undone.

Safety program recompilation is required after changes to standard DBs to which the safety program has read or write access.

4.2.1 CPU-wide access protection for the safety-related project data

Requirement

The project does not use user management to protect the project data

Setting up CPU-wide access protection for safety-related project data

To set up CPU-wide access protection for safety-related project data, assign a password for the safety program. Proceed as follows:

1. Open the folder for your F-CPU in the project tree.
2. Select "Safety Administration" and select "Go to access protection" in the shortcut menu.

   Alternatively, double-click on "Safety Administration". The *Safety Administration Editor* of the F-CPU will open. Select "Access protection" in the navigation area.

3. Under "Offline safety program protection", click "Setup" and enter the password (max. 30 characters) for the safety program in the following dialog in the "New password" and "Confirm password" fields.
4. Confirm the assigned password with "OK".

You have set up CPU-wide access protection for safety-related project data and have gained access permission for the safety-related project data.

NOTE

Use different passwords for the F-CPU and the safety program to optimize access protection.

Changing the password for safety-related project data

You can change the password for the safety-related project data as long as you have the necessary access permissions. It also takes place in the "Access protection" area (via "Change" button) and is carried out as usual under Windows through entry of the old and double entry of the new password.

Deleting CPU-wide access protection for safety-related project data

To delete CPU-wide access protection for safety-related project data, delete the password for the safety program. Proceed as follows:

1. Open the folder for your F-CPU in the project tree.
2. Select "Safety Administration" and select "Go to access protection" in the shortcut menu.

   Alternatively, double-click on "Safety Administration". The *Safety Administration Editor* of the F-CPU will open. Select "Access protection" in the navigation area.
3. Click the "Change" button.
4. Under "Old password", enter the password for the safety program.
5. Click "Revoke" and then on "OK".

Gaining access permission through login to the safety program

Log in to the safety program as follows:

1. Open the folder for your F-CPU in the project tree.
2. Select "Safety Administration" and select "Go to access protection" in the shortcut menu.

   Alternatively, double-click on "Safety Administration". The *Safety Administration Editor* of the F-CPU will open. Select "Access protection" in the navigation area.
3. Enter the password for the safety program in the "Password" input field.
4. Select the "Login" button.

Validity of access permission for safety-related project data

If access permission for safety-related project data was obtained by entering the password, this remains until the project is closed. When the TIA Portal is closed, any project that is still open is automatically closed and any access permission granted is canceled.

Cancelling access permission through logoff

The access permission for safety-related project data can be cancelled as follows:

- By clicking the "Log off" button in the "Access protection" area in the "*Safety Administration Editor*".
- In the shortcut menu for the *Safety Administration Editor* (access by right-clicking).
- By using the lock symbol in the line of the *Safety Administration Editor*.

The user will then be prompted to enter the password for the safety program again the next time an action requiring a password is performed.

Displaying the availability of access authorization

The status of the access authorization is displayed in the project tree as follows:

- The access authorization is available when the lock symbol in the line of the *Safety Administration Editor* is shown unlocked.
- The access permission is not available, if the lock symbol shows a closed lock.
- If no lock symbol is shown, no password was assigned.

4.2.2 Project-wide access protection for the safety-related project data

As an alternative to CPU-wide access protection for the safety-related project data with a password in the *Safety Administration Editor*, you can use the project protection for the TIA project.

To do this, assign the "Edit safety-related project data" function right to users who are allowed to edit safety-related project data. To do this, proceed as usual for function rights. You can find more information in the help for *STEP 7* under "Managing users and roles".

NOTE

When project-wide project access protection is activated, any existing passwords for safety-related project data for all F-CPUs of the project are deleted.

Users who need the "Edit safety-related project data" function right may require additional function rights, for example:

- "Modify PLC program" to change the safety program
- "Downloading to PLC" to load the safety-related project data into the F-CPU
- "Modify hardware configuration" to make changes to the safety-related hardware configuration

4.3 Access protection for the F-CPU

The access protection for the F-CPU is effective for all operations in which the safety-related project data can be changed online, for example:

- When downloading the safety program to the F-CPU
- When downloading the hardware configuration to the F-CPU, also when changes are made to F-I/O not used in the Safety program

NOTE

Note that, for usability reasons, the F-BaseID falls under the access protection for the safety-related project data, but it is not covered by the access protection of the F-CPU for safety-related project data. Here, the F-BaseID is treated like a standard parameter.

There is an exception for switching between an F-BaseID with the value "Delayed transfer" and an F-BaseID with a specific value (either 0 or an active F-BaseID).

- (S7-1200, S7-1500) When downloading a hardware configuration to the F-CPU containing safety-related changes
- When assigning the PROFIsafe address
- When loading or deleting F-blocks used in the safety program
- When disabling safety mode or when resetting the remaining runtime (the password must always be entered, even if access permission for the F-CPU is still valid.)
- When restoring a backup of the F-CPU

  Exception with S7-1200/1500 F-CPUs: If neither the Safety program nor the password for the F-CPU or users with the rights "Change F-System parameters (F-Admin)" or "Full access including fail-safe" are changed as a result, the access protection for the F-CPU is not queried.

4.3.1 Access protection through the use of protection levels

This section describes the access protection for F-CPUs through the use of protection levels.

You can use access levels for the following F-CPUs:

- S7-300/400
- S7-1200 G1 up to firmware 4.6 (as of firmware V4.7 only if the option "Use access control via access levels" is activated)
- S7-1200 G2 if the parameter "Use access control via access levels" is selected.
- S7-1500 up to firmware V3.0 (as of firmware V3.1 only if the parameter "Use access control via access levels" is activated)
- S7-1500 F Software Controller up to firmware V3x.0 (as of firmware V3x.1 only if the parameter "Use access control via access levels" is activated)

Setting up access protection for the F-CPU

To set up access protection for the F-CPU, assign a password in the F-CPU configuration. Proceed as described in the *STEP 7* help under "Configuring access levels".

Clicking on the link "Go to the "Protection" area of the F-CPU" in the "Access protection" area of the *Safety Administration Editor* takes you directly to the password for the F-CPU.

⚠ WARNING

(S7-300, S7-400) In productive operation, it must be prevented that changes to the standard user program are unintentionally made in the Safety program as well. For this purpose, you must configure the protection level "Write protection for fail-safe blocks" and configure a password for the F-CPU.

With the "Write protection" or "Write/read protection" protection level, the password would apply to the standard and the Safety program. *(S001)*

⚠ WARNING

(S7-1200, S7-1500) In productive operation, the safety-related project data must be protected against unintentional changes.

If you use user management, the Runtime rights "Change parameter of the F-System (F-Admin)" and "Full access including fail-safe" and the engineering right "Manage users and roles" may only be granted to authorized users. You must also limit the assignment of user rights to trained personnel. Before the transition into productive operation, you must remove the "Change parameter of the F-System (F-Admin)" and "Full access including fail-safe" rights from the "Anonymous" user.

If you are not using user management or the "Access control via access levels" parameter is activated, then configure at least the "Full access (no protection)" protection level and assign a password under "Full access incl. fail-safe (no protection)". This protection level allows full access only to the standard user program and not the Safety program.

If you select a higher protection level, for example to protect the standard user program, you must assign an additional password for "Full access (no protection)".

Assign different passwords for the individual access levels. *(S041)*

You enable access protection by downloading (Page 350) the hardware configuration to the F-CPU.

⚠ WARNING

(S7-300, S7-400) If **several F-CPUs** can be reached via a network (such as Industrial Ethernet) by **the same programming device or PC**, you must take the following additional measures to ensure that the safety-related project data is downloaded to the correct F-CPU:

Use passwords specific to each F-CPU, such as a uniform password for the F-CPUs with attached Ethernet address for each.

Note the following:

- A point-to-point connection must be used to activate the access protection of an F-CPU when the hardware configuration is loaded for the first time (similar to assigning an MPI address to an F-CPU for the first time).
- Before downloading the safety-related project data to an F-CPU, existing access permission for any other F-CPU must first be cancelled.
- The last download of the safety-related project data prior to switching to productive operation must be made with enabled access protection. *(S021)*

⚠ WARNING

When using tools for the automation or operation (of TIA Portal or web server) with which the access protection for the F-CPU can be removed, the safety-related project data may not be protected against unintentional changes anymore. When using corresponding tools, the requirements regarding access protection of the F-CPU must still be met. *(S078)*

Changing the password for the F-CPU

For the new password to become valid after a change to an access level password for the F-CPU, you must download the changed configuration into the F-CPU. If necessary, you must enter the "old" password for the F-CPU for this load operation. The F-CPU must be in STOP mode.

Deleting access protection for the F-CPU

To delete configured access protection for the F-CPU, change the access level and delete the passwords for the F-CPU that are no longer required. To do this, proceed as in the standard.

Obtaining access permission for the F-CPU

You obtain access permission for the F-CPU by entering the password for the F-CPU, according to the configured protection level, prior to performing an action requiring a password.

Validity of access permission for the F-CPU

Access permission for the F-CPU remains valid until the TIA Portal is closed or access permission is revoked.

Canceling access permission for the F-CPU

You cancel the access permission with the menu command "Online > Delete access rights" auf.

WARNING

The safety-related project data must be protected against unintentional changes.

For this you must cancel the access permission and the online connection for the F-CPU whenever you leave the PG/PC.

Alternatively, you can also take other organizational measures (Page 651). *(S006)*

4.3.2 Access protection through user management

This section describes the access protection for F-CPUs through the use of user management and the runtime right for the F-CPU.

You can use local user management for the following F-CPUs if the "Use access control via access levels" parameter is not activated:

- S7-1200 G1 as of firmware V4.7
- S7-1200 G2 as of firmware V1.0
- S7-1500 as of firmware V3.1
- S7-1500 F Software Controller as of firmware V3x.1

You can use central user management for the following F-CPUs:

- S7-1200 G2 as of firmware V4.1
- S7-1500 as of firmware V4.0
- S7-1500 F Software Controller as of firmware V4x.0

Setting up access protection for the F-CPU

To set up access protection for the F-CPU, use the user management with the function right for the F-CPU. Proceed as described in the *STEP 7* help under "Local user management" or "Central user management" and "Settings for users and roles".

If the "Use access control via access levels" option is selected in addition to "Enable access control", you must assign at least a password for "Full access including fail-safe" for the F-CPU in the F-CPU configuration. Proceed as described in the *STEP 7* help under "Configuring access levels".

You can go directly to the settings for access protection of the F-CPU if you click the link "Go to the "Protection" area of the F-CPU" in the "Access protection" area in the *Safety Administration Editor*.

⚠ WARNING

(S7-1200, S7-1500) In productive operation, the safety-related project data must be protected against unintentional changes.

If you use user management, the Runtime rights "Change parameter of the F-System (F-Admin)" and "Full access including fail-safe" and the engineering right "Manage users and roles" may only be granted to authorized users. You must also limit the assignment of user rights to trained personnel. Before the transition into productive operation, you must remove the "Change parameter of the F-System (F-Admin)" and "Full access including fail-safe" rights from the "Anonymous" user.

If you are not using user management or the "Access control via access levels" parameter is activated, then configure at least the "Full access (no protection)" protection level and assign a password under "Full access incl. fail-safe (no protection)". This protection level allows full access only to the standard user program and not the safety program.

If you select a higher protection level, for example to protect the standard user program, you must assign an additional password for "Full access (no protection)".

Assign different passwords for the individual access levels. *(S041)*

You enable access protection by downloading (Page 350) the hardware configuration to the F-CPU.

⚠ WARNING

When using tools for the automation or operation (of TIA Portal or web server) with which the access protection for the F-CPU can be removed, the safety-related project data may not be protected against unintentional changes anymore. When using corresponding tools, the requirements regarding access protection of the F-CPU must still be met. *(S078)*

Changing access protection for the F-CPU

For the changed users or the changed password to take effect after a change to the users with F-right or a password change for the F-CPU, you must download the changed configuration to the F-CPU and select the "Download all user management data (reset to project settings)" option in the "Load preview" dialog when downloading. For this download, you must legitimate yourself as a user with F-right or enter the existing password for the F-CPU, if necessary. For changes in the "Access protection via protection levels" area, the F-CPU must be in STOP.

Changes that you make as part of the central user management on the associated UMC server of the F-CPU (e.g. a new user password) take effect immediately without a download.

Deleting access protection for the F-CPU

To delete the access protection for the F-CPU, grant the user Anonymous the F-right ("Full access including fail-safe").

Obtaining access permission for the F-CPU

You obtain access permission for the F-CPU by legitimating yourself as a user with F-right or when "Use access control via access levels" mode is enabled by entering the password for the F-CPU prior to performing an action requiring a password.

Validity of access permission for the F-CPU

Access permission for the F-CPU remains valid until the TIA Portal is closed, access permission is revoked or the maximum legitimation time is exceeded. A maximum legitimation time of up to 600 minutes can be set for each user.

Canceling access permission for the F-CPU

You revoke the access permission with the menu command "Online > Delete access rights".

⚠ WARNING

The safety-related project data must be protected against unintentional changes.

For this you must cancel the access permission and the online connection for the F-CPU whenever you leave the PG/PC.

Alternatively, you can also take other organizational measures (Page 651). *(S006)*

4.4 Access protection through organizational measures

To prevent a safety program from being swapped without authorization by exchanging removable media (e.g. flash card, SIMATIC Micro Memory Card or hard disk with S7-1500 F Software Controller), you must observe the following warning:

⚠ WARNING

You must limit access to the F-CPU through organizational measures (Page 651) to persons who are authorized to plug in removable devices. *(S079)*

To prevent unintentional access to an S7-1500 F Virtual Controller and the virtual time service, you need to observe the following warning:

⚠ WARNING

You must take organizational measures (Page 651) to limit access to an IE(V)D with an S7-1500 F Virtual Controller or a virtual time service to persons who are authorized and appropriately trained to commission and operate an IE(V)D.

The following change must not be made by a system administrator under any circumstances, either at the level of a host hypervisor, in Industrial Edge Management or within the IE(V)D operating system:

- Change of data storage of the portion of the Linux file system of the IE(V)D managed by Docker.

If you operate the S7-1500 F Virtual Controller, note the following:

- You may not use images to duplicate the S7-1500 F Virtual Controller because the uniqueness of the S7-1500 F Virtual Controller when loading cannot be guaranteed based on the unique serial number.
- When creating images (e.g. snapshots) of virtual machines or Edge Devices with an S7-1500 F Virtual Controller, you must label them clearly. When importing an image, use program identification to ensure that the correct Safety program is executed.
- If you use automated mechanisms for mirroring images of virtual machines (e.g. cyclical mechanisms provided by the hypervisor for data backup), you need to perform the following steps each time you import new safety-related project data. This ensures that the correct Safety program is executed when the mirrored images of your virtual machines are started:
  1. Load the safety-related project data into the active target CPU
  2. Ensure that hypervisor mechanisms are used to delete all virtual machine images of the active target CPU.
  3. Make a program identification in the active target CPU.
- If you use a mechanism to provide distributed shared data storage (e.g. vSAN) without multiple protection of the data (FTT Failures To Tolerate = 0) in the virtualized environment, you have to ensure for each change of the configuration of this data storage to other retentive memory areas that the previously used memory area does not contain any safety-related project data by deleting this. If you use multiple protection of the data (FTT > 0) in this distributed data storage, you have to perform program identification before each restart of the active target CPU.

(S100)

To ensure that an S7-1500 F Software Controller is not uninstalled or installed unintentionally, you must observe the following warning:

⚠ WARNING

You must limit access to an S7-1500 F Software Controller by organizational measures (Page 651) to persons who are authorized to uninstall and install or repair an S7-1500 F Software Controller (e.g. via the Windows administrator privileges (ADMIN)). *(S075)*

With an S7-1500 F-Software Controller with a PC station version up to incl. V2.1, the function "Delete Configuration" on the PC Station Panel is only offered when no access protection (no protection) is set up on the F-CPU.

With a PC station as of V2.2, it is checked whether the current Windows user is a member of the "Failsafe Operators" Windows user group. When the logged-on Windows user is a member of the group, he/she can execute the "Delete Configuration" function even when the F-password is set. If the logged-on Windows user is not a member of the group, the PC station behaves as up to V2.1.

We therefore recommend that you do not set up F-access protection until after commissioning.

To prevent unintentional restoration of the safety-related project data, formatting of the F-CPU and deletion of program folders via the display of an S7-1500 F-CPU, you must observe the following warning:

⚠ WARNING

The password for the display of the F-CPU may only be disclosed to persons who are authorized to restore safety-related project data, to format the F-CPU and to delete program folders. If a password is not set up for the display, you must protect the display through organizational measures (Page 651) against unauthorized operation. *(S063)*

To prevent unintentional restoration of the safety-related project data of the safety program with the web server in an S7-1200/1500 F-CPU, you must observe the following warning:

⚠ WARNING

The "Change parameter of the F-System (F-Admin)" authorization for the web server (including web server API) without password protection (user "Everybody", or user "Anonymous" when using the user management) is intended only for testing, commissioning, etc. This means only when the system is not in productive operation. In this case, you must ensure the safety of the system through other organizational measures (Page 651).

Before the transition to productive operation, you must remove the rights "Change parameter of the F-System (F-Admin)" and "Full access including fail-safe" from the "Everybody" or "Anonymous" user.

The password of the web server user with "Change parameter of the F-System (F-Admin)" right may only be disclosed to authorized persons. After downloading the hardware configuration, check whether only permitted users of the web server have the "Change parameter of the F-System (F-Admin)" right on the F-CPU. To do so, use the online mode of the *Safety Administration Editor* for an F-CPU without user management or the "Security settings\Users and roles" tab for an F-CPU with user management.

Saving the login file and the password of the web server in the browser is only permitted when usage by unauthorized persons is prevented through other organizational measures (Page 651).

If you use the ticket mechanism, handling of the tickets must be just as restrictive as handling of the user password of the web server with the "Change parameter of the F-System (F-Admin)" right. *(S064)*

Programming

5

5.1 Overview of Programming

Introduction

A safety program consists of F-blocks that you create using the FBD or LAD programming language and F-blocks that are automatically added. Fault detection measures and fault reaction measures are automatically added to the safety program you create, and additional safety-related tests are performed. Moreover, you have the option to incorporate special ready-made safety functions in the form of instructions into your safety program.

An overview of the following is given below:

- The structure of the safety program
- The fail-safe blocks
- Differences in the programming of the safety program with FBD/LAD compared to programming of standard user programs

5.1.1 Program structure of the safety program (S7-300, S7-400)

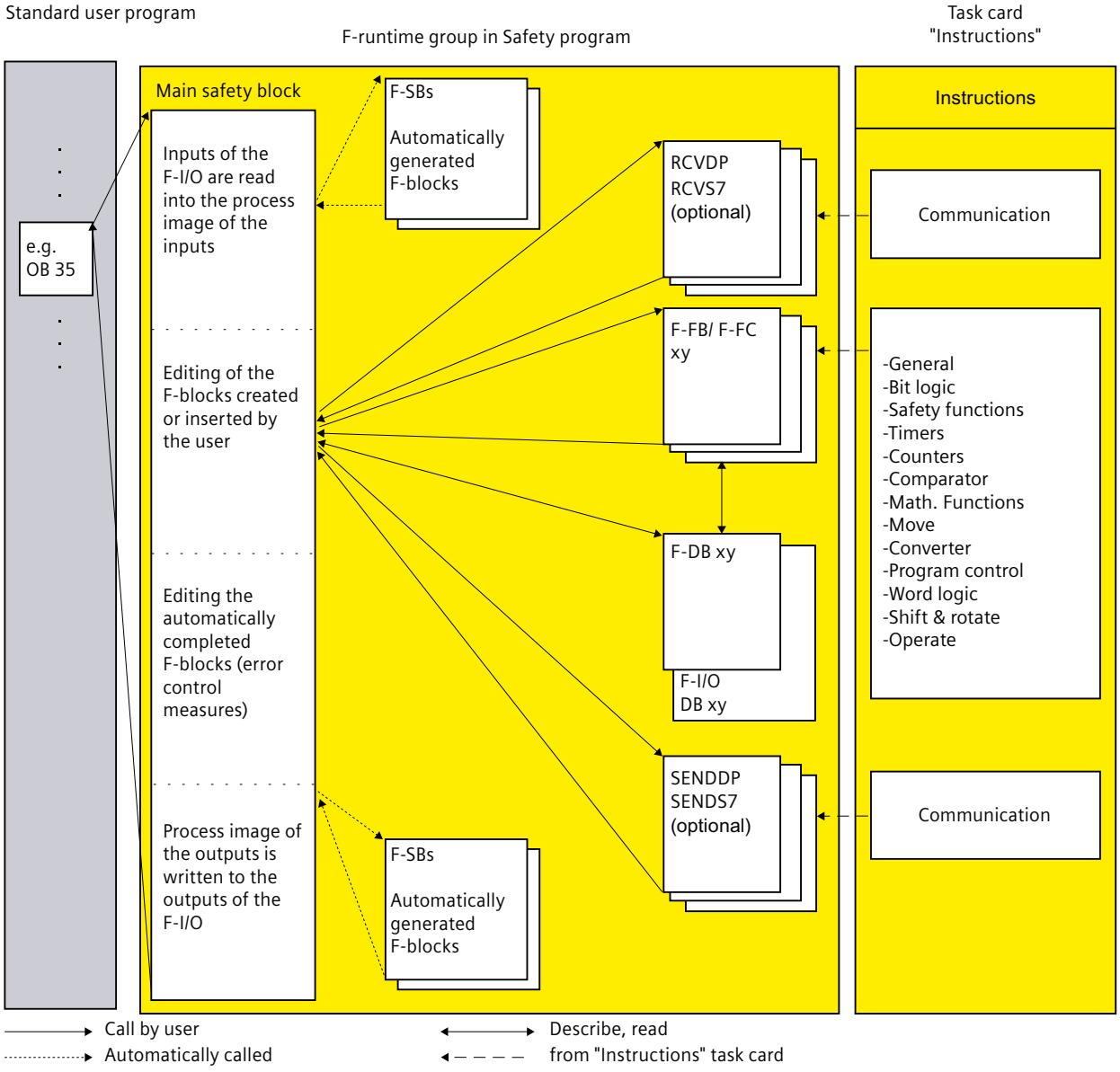
Representation of program structure

For structuring purposes, a safety program consists of one or two F-runtime groups.

Each F-runtime group contains:

- F-blocks that you create using FBD or LAD or that are inserted from the project library or global libraries
- F-blocks that are added automatically (F-system blocks, automatically generated F-blocks, and F-I/O DBs)

Below is a schematic diagram of a safety program or an F-runtime group for an S7-300/400 F-CPU.

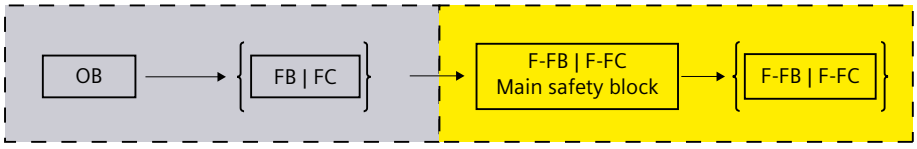


Main safety block

The main safety block is the first F-block of the safety program that you program yourself. During compiling, it is supplemented by additional invisible calls of F-system blocks.

You must assign the main safety block to an F-runtime group (Page 133).

The main safety block in an S7-300/400 F-CPU is called from any block in the standard user program. We recommend a call from an OB 3x.



F-runtime groups

To improve handling, a safety program consists of one or two "F-runtime groups". An F-runtime group is a logical construct of several related F-blocks that is formed internally by the F-system.

An F-runtime group consists of the following:

- A main safety block (an F-FB/F-FC that you assign to the calling OB (FB/FC) as needed)
- Any additional F-FBs/F-FCs that you call from the main safety block
- One or more F-DBs, as needed
- F-I/O DBs
- F-blocks from the project library or global libraries
- F-system blocks F-SBs
- Automatically generated F-blocks

Structuring the safety program in two F-runtime groups

You can divide your safety program into two F-runtime groups. By having parts of the safety program (one F-runtime group) run in a faster priority class, you achieve faster safety circuits with shorter response times.

You can only transfer data between the F-runtime groups using F-runtime group communication (Page 143).

5.1.2 Program structure of the safety program (S7-1200, S7-1500)

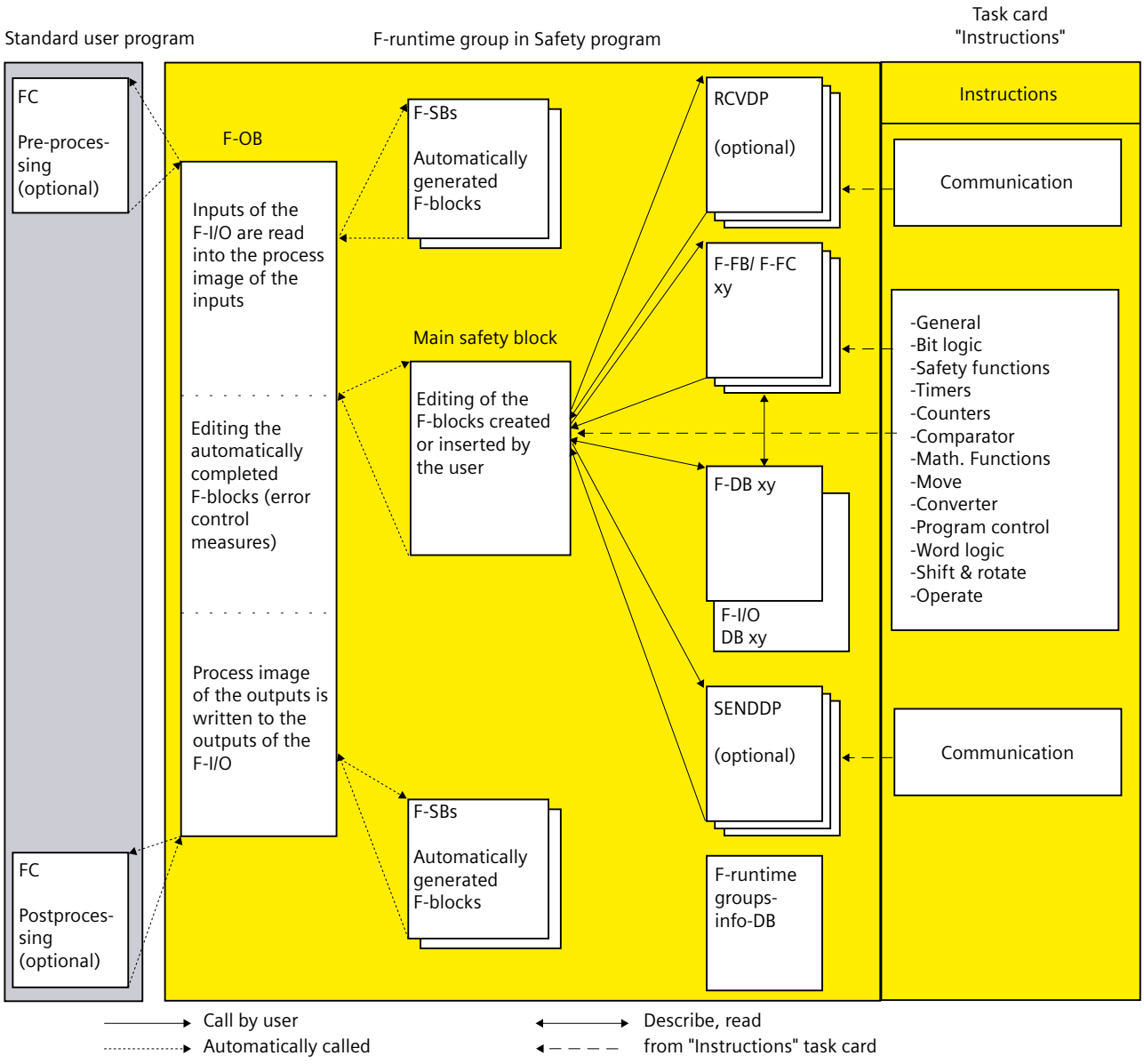
Representation of program structure

For structuring purposes, a safety program consists of one or two F-runtime groups.

Each F-runtime group contains:

- F-blocks that you create using FBD or LAD or that are inserted from the project library or global libraries
- F-blocks that are added automatically (F-system blocks F-SBs, automatically generated F-blocks, F-runtime DB, and F-I/O DBs)

Below is a schematic diagram of a safety program or an F-runtime group for an S7-1200/1500 F-CPU.

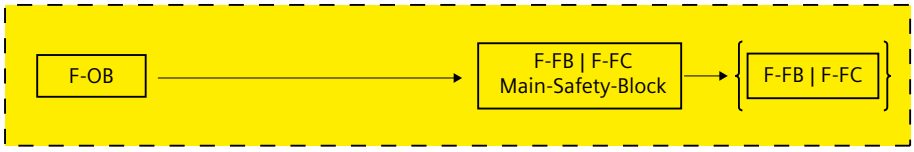


Main safety block

The main safety block is the first F-block of the safety program that you program yourself.

You must assign the main safety block to an F-runtime group (Page 133).

The main safety block in an S7-1200/1500 F-CPU is called by the F-OB assigned to the F-runtime group.



F-runtime groups

To improve handling, a safety program consists of one or two "F-runtime groups". An F-runtime group is a logical construct of several related F-blocks that is formed internally by the F-system.

An F-runtime group consists of the following:

- An F-OB which calls the main safety block
- A main safety block (an F-FB/F-FC that you assign to the F-OB)
- Any additional F-FBs or F-FCs that you call from the main safety block
- One or more F-DBs, as needed
- F-I/O DBs
- F-runtime group information DB
- F-blocks from the project library or global libraries
- F-system blocks F-SBs
- Automatically generated F-blocks
- A preprocessing and/or postprocessing block, as needed (see Pre-/postprocessing (S7-1200, S7-1500) (Page 88))

Pre-/postprocessing of an F-runtime group

You have the option of calling blocks of the standard user program (FCs) directly before or after an F-runtime group, for example, for data transfer during fail-safe communication via Flexible F-Link. (see Pre-/postprocessing (S7-1200, S7-1500) (Page 88))

Structuring the safety program in two F-runtime groups

You can divide your safety program into two F-runtime groups. By having parts of the safety program (one F-runtime group) run in a faster priority class, you achieve faster safety circuits with shorter response times.

You can only transfer data between the F-runtime groups using F-runtime group communication (Page 146).

See also

F-runtime group information DB (S7-1200, S7-1500) (Page 154)

5.1.3 Safety program in Software Unit (Safety Unit) (S7-1500)

Introduction

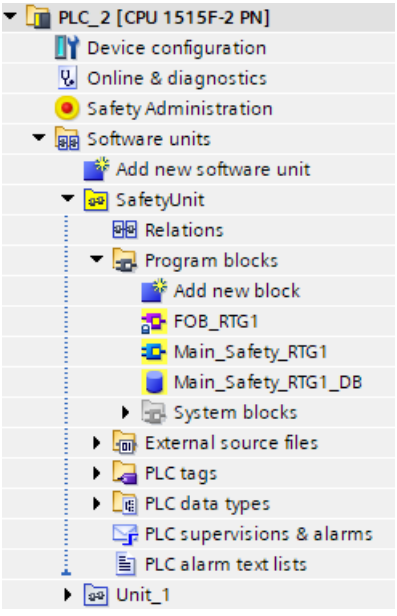
You can use Safety Units if you want to use the advantages of the Software Units for the safety program as well, e.g. independent compiling and loading.
The blocks of the safety program are then located exclusively in the Safety Unit and can no longer be created in the "Program blocks" folder directly below the F-CPU.
You must make the decision for the Safety Unit before creating the F-CPU. After creating the F-CPU, the decision for this F-CPU can no longer be changed.

Requirement

- F-CPU S7-1500 as of FW V2.6
- The check box "Manages safety program in 'Safety Units' environment" is selected under "Options/Settings/STEP 7 Safety".
- The F-CPU is newly created.

Safety program in the Safety Unit

After a new F-CPU has been created, the safety program is located in the Safety Unit.



Store the transfer DBs for the data transfer from/to the standard user program (Page 198) in the Safety unit.

Information on Software Units is available in the help for *STEP 7* under "Using Software Units".

Particularities

- You cannot store the Safety Unit in a library.
- System blocks in the Safety Unit cannot be published.

5.1.4 Fail-Safe Blocks

F-blocks of an F-runtime group

The following table shows the F-blocks that you use in an F-runtime group:

F-block	Function	S7-300/400 F-CPUs	S7-1200/1-500 F-CPUs
Main safety block	The first step in programming of the Safety program is the main safety block. The main safety block in S7-300/400 F-CPUs is an F-FC or F-FB (with instance DB), which is called by a standard block (recommendation: OB 35) from the standard user program. The main safety block in S7-1200/1500 F-CPUs is an F-FC or F-FB (with instance DB), which is called by the F-OB.	X	X
F-FB/F-FC	Both in the main safety block as well as additional F-FBs and F-FCs, you can perform the following: • Program the Safety program with the instructions available for F-blocks • Call other created F-FBs/F-FCs for structuring the Safety program • Insert F-blocks from the project library or global libraries	X	X
F-DB	Optional fail-safe data blocks that can be read- and write-accessed within the entire Safety program.	X	X
F-I/O DB	An F-I/O DB is automatically generated for each F-I/O when it is configured. You access the tags of the F-I/O DB in connection with F-I/O access.	X	X
F-shared DB	The F-shared DB is a fail-safe data block that contains all of the shared data of the Safety program and additional information needed by the F-system.	X	—
F-runtime group information DB	An F-runtime group information DB is created when you create an F-runtime group. The F-runtime group information DB provides information on the F-runtime group and on the Safety program as a whole.	—	X
DB for F-runtime group communication	Created for safety-related communication between F-runtime groups in the *Safety Administration Editor*. With this, you can read fail-safe tags that were made available in an F-DB from an F-runtime group in a different F-runtime group.	X	—

F-block	Function	S7-300/400 F-CPUs	S7-1200/1-500 F-CPUs
F_SystemInfo_DB	More information (e.g. signatures) on the Safety program can be found in the F_SystemInfo_DB. You cannot access this information in the Safety program.	—	X
F-communication DB	For each communication connection with Flexible F-Link, an F-communication DB is generated automatically when the communication connection is created. You access the tags of the F-communication DB in connection with Flexible F-Link communication connections.	—	X
F-OPC UA Consumer DB	For communication connections via OPC UA Safety, an F-OPC UA Consumer DB is automatically generated for each SafetyConsumer when the OPC UA SafetyProvider information model is imported. You access the variables of the F-OPC UA Consumer DB in connection with OPC UA Safety communication connections.	—	X

NOTE

You are not permitted to insert F-system blocks from the "System blocks" folder in a main safety block/F-FB/F-FC.

Instructions for the Safety program

In the "Instructions" task card, you can find instructions for the F-CPU used and which you can to use program the Safety program.

You can find instructions that you know from the standard user program, such as bit logic operations, mathematical functions, functions for program control, and word logic operations.

Moreover, there are instructions with safety functions, e.g., for two-hand monitoring, discrepancy analysis, muting, emergency STOP/emergency OFF, safety door monitoring, feedback monitoring and instructions for safety-related communication between F-CPUs.

More information

For a detailed description of the instructions for the Safety program, refer to Overview of instructions (Page 450).

Using instruction versions

As with the instructions for the standard user program, there may also be different versions of the instructions for the Safety program.

More information on instruction versions can be found in the help on STEP 7 in "Basics for instruction versions".

More information on the differences of the individual versions of the instructions for the Safety program can be found in the relevant chapter of the instructions.

NOTE

Note the following:

- If you change the version of an instruction used in the Safety program in the task card "Instructions" to a version which does not have identical functions, the functioning of your Safety program may change after recompiling the Safety program. In addition to the signature of the F-block that uses the instruction, the Collective F-Signature and the Collective F-SW-Signature of your Safety program also change. You may have to perform an acceptance test (Page 433).
- (S7-300, S7-400) If you use a know-how protected F-block in your Safety program which uses an instruction which is not the same version as that set in the task card "Instructions", when the program is compiled without entering the password for the know-how protected F-block, it is adjusted to the version set in the task card "Instructions". If the instruction versions do not have identical functions, the functioning of the know-how protected F-block may change and always its signature.

5.1.5 Restrictions in the programming languages FBD/LAD

LAD and FBD programming languages

The user program in the F-CPU typically consists of a standard user program and a safety program.

The standard user program is created using standard programming languages such as SCL, STL, LAD, or FBD.

For the safety program, LAD or FBD may be used with certain restrictions in the instructions and the applicable data types and operand areas. Also note the restrictions for the individual instructions.

Supported instructions

The instructions available depend on the F-CPU used. You can find the supported instructions in the description of the instructions (starting from STEP 7 Safety V21 instructions (Page 450)).

NOTE

Enable input EN cannot be connected.

(S7 -300, S7-400) Output ENO cannot be connected.

(S7-1200, S7-1500) Output ENO can only be connected with the following instructions:

- ADD: Add (Page 553)
- SUB: Subtract (Page 556)
- MUL: Multiply (Page 559)
- DIV: Divide (Page 562)
- NEG: Create twos complement (Page 565)
- ABS: Form absolute value (Page 568)
- CONVERT: Convert value (Page 578)

Supported data types and parameter types

Only the following data types are supported:

- BOOL
- INT
- WORD
- DINT
- DWORD (S7-300, S7-400)
- TIME
- ARRAY, ARRAY[*] when using the instructions RD_ARRAY_I: Read value from INT F-array (STEP 7 Safety V21) (S7-1200 G2, S7-1500) (Page 571) and RD_ARRAY_DI: Read value from DINT F-array (STEP 7 Safety V21) (S7-1200 G2, S7-1500) (Page 573).

  Restrictions:

  – ARRAY only in F-DBs
  – ARRAY limits: 0 up to max. 10000
  – ARRAY[*] only as InOut parameter (InOut) in F-FCs and F-FBs
  – Only ARRAY of INT or ARRAY of DINT permissible
- F-compliant PLC data type (UDT) (S7-1200, S7-1500) (Page 126)
- Named value data types (S7-1500) (Page 131)

NOTE

If the result of an instruction is located outside the permitted range for this data type, the F-CPU may switch to STOP. The cause of the diagnostics event is entered in the diagnostic buffer of the F-CPU.
You must therefore ensure that the permitted range for the data type is observed when creating the program, select a matching data type or use the ENO output for S7-1200/1500 F-CPUs or the OV instruction (Page 609) for S7-300/400 F-CPUs.

Note the description of the individual instructions.

Non-permitted data and parameter types

The following types are **not** permitted:

- All types not listed in the section "Supported data types and parameters types" (e.g. BYTE, REAL)
- Compound data types (e.g. STRING, ARRAY (S7-300, S7-400, S7-1200 G1), STRUCT, PLC data type (UDT) (S7-300, S7-400), Named value data types (S7-300, S7-400, S7-1200))
- Parameter types (e.g. BLOCK_FB, BLOCK_DB, ANY)

Supported operand areas

The system memory of an F-CPU is divided into the same operand areas as the system memory of a standard CPU. You can access the operand areas listed in the table below from within the safety program.

Table 5-1 Supported operand areas

Operand area	Description
Process image of the inputs	
• Of F-I/O	Only read-only access to input channels of F-I/O is possible. Transfer to InOut parameters of an F-FB or F-FC is therefore not valid either. The process image of the inputs of F-I/O is updated prior to the start of the main safety block.
• Of standard I/O	Input channels of standard I/O can only be accessed read-only. Transfer to InOut parameters of an F-FB or F-FC is therefore not valid either. In addition, a process-specific validity check is required. See the *STEP 7 help* for the update times of the process image of the inputs of standard I/O.
Process image of the outputs	
• Of F-I/O	Only write-only access to output channels of F-I/O is possible. Transfer to InOut parameters of an F-FB or F-FC is therefore not valid either. In the safety program, the values for the outputs of the F-I/O are calculated and stored in the process image of the outputs. The process image of the outputs for F-I/O is updated after the end of the main safety block.

Operand area	Description
• Of standard I/O	Output channels of standard I/O are write-only channels. Transfer to InOut parameters of an F-FB or F-FC is therefore not valid either. In the safety program, the values for the outputs of the standard I/O are also calculated and stored in the process image of the outputs, if needed. See the *STEP 7 help* for the update times of the process image of the outputs of standard I/O.
Bit memory	This area is used for data exchange with the standard user program. In addition, read access requires a process-specific validity check. A particular element of the bit memory can be either read- or write-accessed in the safety program. Transfer to InOut parameters of an F-FB or F-FC is therefore not valid either. Note that it is only permitted to use bit memory for connecting the standard user program and the safety program; it must not be used as a buffer for F-data.
Data blocks	
• F-DB	Data blocks store information for the program. They can either be defined as global data blocks such that all F-FBs, F-FCs, or main safety blocks can access them or assigned to a particular F-FB or main safety block (instance DB). A tag of a shared DB can only be accessed from one F-runtime group, and an instance DB only from the F-runtime group in which the corresponding F-FB/instruction is called.
• DB	This area is used for data exchange with the standard user program. In addition, read access requires a process-specific validity check. For a tag of a DB, either read access or write access is possible in the safety program. Transfer to InOut parameters of an F-FB or F-FC is therefore not valid either. Note that the tags of a DB can only be used for transferring data between the standard user program and the safety program; DBs must not be used as a buffer for F-data.
Temporary local data	This memory area holds the temporary tags of a block (or F-block) while the (F-) block is being executed. The local data stack also provides memory for transferring block parameters and for saving intermediate results.

Data type conversion

As with the standard user program, there are two possibilities for data type conversion in the safety program:

- Implicit conversion

  The implicit conversion is executed as in the standard user program with the following restrictions: The bit length of the source data type has to match the bit length of the destination data type.

- Explicit conversion

  You use an explicit conversion instruction (Page 578) before the actual instruction is executed.

Slice access

Slice access is not possible in the safety program.

Non-permitted operand areas

Access via units other than those listed in the table above is **not** permitted. The same applies to access to operand areas not listed, in particular:

- Data blocks that were automatically added

  Exception: Certain tags in the F-I/O DB (Page 164) and in the F-shared DB (S7-300, S7-400) (Page 154) or F-runtime group information DB (S7-1200, S7-1500) (Page 154)
- I/O area: Inputs
- I/O area: Outputs

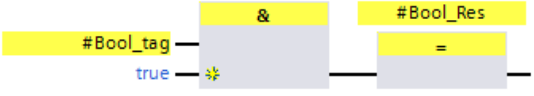
Boolean constants "0" or "FALSE" and "1" or "TRUE" (S7-300, S7-400)

The Boolean constants "0" or "FALSE" and "1" or "TRUE" are available for S7-300/400 F-CPUs as "Tags" "RLO0" and "RLO1" in the F-shared DB. You access them through a fully qualified DB access ("F_GLOBDB".RLO0 or "F_GLOBDB".RLO1).

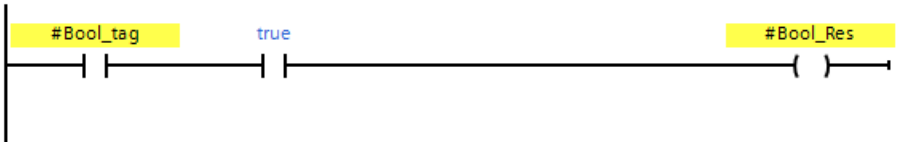
Boolean constants "0" or "FALSE" and "1" or "TRUE" (S7-1200, S7-1500)

The Boolean constants "0" or "FALSE" and "1" or "TRUE" are available for S7-1200/1500 F-CPUs to assign parameters during block calls. You can enter this directly in FBD or LAD at the respective block inputs.

Example FBD:



Example LAD:



As an alternative, as before you have the option to also set "1" or "TRUE" in a tag using the "Assignment (Page 461)" instruction.

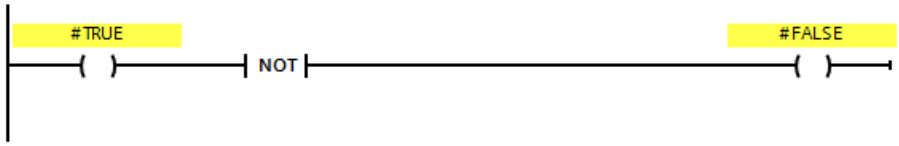
To do so, do not interconnect the box input of the "Assignment" instruction in FBD. In LAD, you interconnect the input directly with the supply rail.

You obtain a tag with "0" or "FALSE" by subsequent inversion with the instruction "Invert RLO (Page 458)".

Example FBD:



Example LAD:



Operand area of temporary local data: Particularities

NOTE

Note when using the operand area of temporary local data that the first access of a local data element in a main safety block/F-FB/F-FC must always be a write access. This initializes the local data element.

Make sure that a temporary local data element is initialized prior to the first JMP, JMPN, or RET instruction.

The "local data bit" should be initialized with the Assign ("=") (FBD) or ("--( )") (LAD) instruction. Assign the local data bit a signal state of "0" or "1" as a Boolean constant.

Local data bits cannot be initialized with the Flip Flop (SR, RS), Set Output (S) or Reset Output (R) instructions.

The F-CPU can go to STOP if this is not observed. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

"Fully qualified DB access"

Access to tags of a data block in an F-FB/F-FC is "fully qualified DB access". This also applies to initial access to tags of a data block after a jump label.

For S7-300/400 F-CPUs, only initial access needs to be "fully qualified DB access". Alternatively, you can use the instruction "OPN".

Example of "fully qualified DB access":

Assign a name for the F-DB, e.g. "F_Data_1". Use the names assigned in the declaration of the F-DB instead of the absolute addresses.

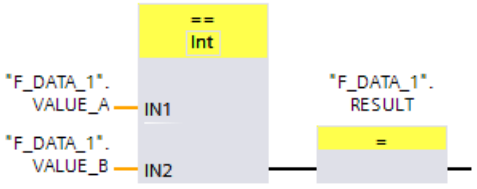


Figure 5-1 Example with fully-qualified access

Example of "non-fully qualified DB access" (S7-300, S7-400):

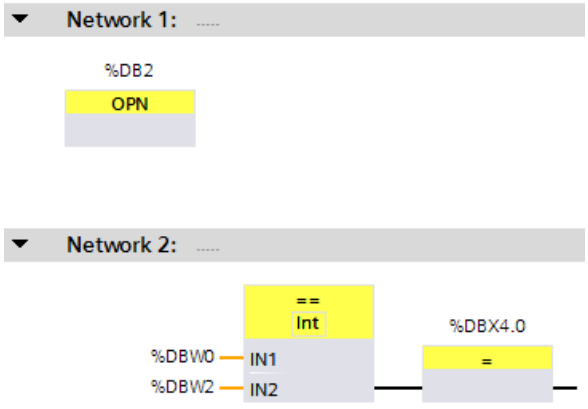


Figure 5-2 Example without fully-qualified access

Access to instance DBs

You can also access instance DBs of F-FBs with fully qualified access, e.g., for transfer of block parameters. It is not possible to access static local data in single/multi-instances of other F-FBs.

(S7-300, S7-400) Note that accessing instance DBs of F-FBs that are not called in the safety program can cause the F-CPU to go to STOP mode.

Note the following warning for S7-1200/1500 F-CPUs:

⚠ WARNING

When you program global access to instance DBs of F-FBs, you must ensure that the associated F-FB is called in the safety program. *(S096)*

5.1.6 F-compliant PLC data types (UDT) (S7-1200, S7-1500)

Introduction

You declare and use F-compliant PLC data types (UDT) as you would standard PLC data types (UDT). You can use F-compliant PLC data types (UDT) in the safety program as well as in the standard user program.

Differences to standard PLC data types (UDT) are described in this chapter.

Information on the use and declaration of standard PLC data types (UDT) is available in the *STEP 7 help* under "Declaring PLC data types".

Declaring F-compliant PLC data types (UDT)

You declare F-compliant PLC data types (UDT) as you would PLC data types (UDT).

In F-compliant PLC data types (UDT), you can use all data types (Page 119) that you can also use in safety programs. Exception: ARRAY.

Proceed as follows for declaration:

1. Click on "Add new PLC data type" in the "PLC Data Types" folder in the project tree.
2. To create an F-compliant PLC data type (UDT), enable the option "Create F-compliant PLC data type" in the "Add new PLC data type" dialog.
3. Proceed as described in the *STEP 7 help* under "Programming structure of PLC data types".

You specify default values for F-compliant PLC data types (UDT) during the declaration.

Using F-compliant PLC data types (UDT)

You use F-compliant PLC data types as you would standard PLC data types (UDT).

Nesting depth for F-compliant PLC data types

For F-compliant PLC data types, the maximum nesting depth is limited compared to the standard PLC data types (maximum nesting depth = 8). There is a dependency on the call sequence of the block in which a tag of the nested F-compliant PLC data type is declared. Each call level of F-FCs or multi-instance F-FBs reduces the maximum nesting depth of the F-compliant PLC data type used. For multi-instance F-FBs, the caller counts as an additional level.

When a tag of a nested F-compliant PLC data type is declared in a global F-DB, its maximum nesting depth is 7.

Example 1

The main safety block (level 1) calls an F-FB as multi-instance (level 2) which, in turn, calls an F-FC (level 3) in which a tag of the type of a nested F-compliant PLC data type is declared. This means a maximum nesting depth of 5 is available for the fail-safe compliant PLC data type used by the tag.

Example 2

The main safety block calls an F-FB as a single instance (level 1) which, in turn, calls an F-FC (level 2) in which a tag of the type of a nested F-compliant PLC data type is declared. This means a maximum nesting depth of 6 is available for the fail-safe compliant PLC data type used by the tag.

Changes to F-compliant PLC data types (UDT)

You need the password for the safety program to change F-compliant PLC data types (UDT). Regardless if you are using the F-compliant PLC data type (UDT) in an F-block, in a standard block or not at all.

See also

"F-compliant PLC data types" area of the Safety Administration Editor (S7-1200, S7-1500) (Page 90)

5.1.6.1 Grouping PLC tags for inputs and outputs of F-I/O in structures (S7-1200, S7-1500)

You group PLC tags for inputs and outputs of F-I/O in structures (structured PLC tag) as you would for inputs and outputs of standard I/O.

Use F-compliant PLC data types (UDT).

Rules

When creating structured PLC tags for inputs and outputs of F-I/O, you must also observe the following rules in addition to the rules in the standard:

- You must not group inputs/outputs of standard I/O and F-I/O at the same time in a structured PLC tag.
- You may only group inputs/outputs of actually existing channels (channel value and value status) in a structured PLC tag.

  See also Addressing F-I/O (Page 157)
- You may only group inputs/outputs of channels (channel value and value status) that are enabled in the hardware configuration in a structured PLC tag.

  See also Addressing F-I/O (Page 157)
- You may only group inputs of channels (channel value and value status) that provide the result of the "1oo2 sensor evaluation" with set "1oo2 sensor evaluation".

  See also Addressing F-I/O (Page 157)
- You should combine all inputs and outputs of an F-I/O in a structured PLC tag. When divided into several structured PLC tags, these can only begin with multiples of 16 bits. This also applies to nested F-compliant PLC data types (UDT). See rules in the standard.

  The F-CPU can go to STOP mode if this is disregarded. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

- A structured PLC tag that groups outputs of an F-I/O must not overlap with other PLC tags.

  The F-CPU can go to STOP mode if this is disregarded. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.
- In a structured PLC tag for inputs and outputs of F-I/O, the data types WORD, DWORD and TIME are not permitted in the associated PLC data type (UDT).

NOTE

To observe these rules, you must declare the F-compliant PLC data type that is used for the structured PLC tag accordingly.

You can find the addresses allocated to a structured PLC tag in the "IO tags" tab of an F-I/O configuration.

5.1.6.2 Example of structured PLC tags for inputs and outputs of F-I/O (S7-1200, S7-1500)

Introduction

This example uses the F-module 4 F-DI/3 F-DO DC24V/2A with 1oo2 evaluation to demonstrate how you use structured PLC tags for access to F-I/O.

Channel structure of the 4 F-DI/3 F-DO DC24V/2A F-module

The table below sets out the channel structure and address assignment of the F-module 4 F-DI/3 F-DO DC24V/2A with 1oo2 evaluation. You may only access existing and enabled channels (addresses I15.0 to I15.3 and I16.0 to I16.3). These channels provide the result of 1oo2 evaluation generated internally in the F-module.

Table 5-2 Channel structure and addresses of the channel values of inputs with 1oo2 evaluation

Channel	Address
DI channel 0 channel value	I15.0
DI channel 1 channel value	I15.1
DI channel 2 channel value	I15.2
DI channel 3 channel value	I15.3
—	I15.4
—	I15.5
—	I15.6
—	I15.7

Table 5-3 Channel structure and addresses of the value status of the inputs with 1oo2 evaluation

Channel	Address
DI channel 0 value status	I16.0
DI channel 1 value status	I16.1
DI channel 2 value status	I16.2
DI channel 3 value status	I16.3
—	I16.4
—	I16.5
—	I16.6
—	I16.7

Table 5-4 Channel structure and addresses of the value status of outputs

Channel	Address
DO channel 0 value status	I17.0
DO channel 1 value status	I17.1
DO channel 2 value status	I17.2
DO channel 3 value status	I17.3

Table 5-5 Channel structure and addresses of the channel values of outputs

Channel	Address
DO channel 0 channel value	Q15.0
DO channel 1 channel value	Q15.1
DO channel 2 channel value	Q15.2
DO channel 3 channel value	Q15.3

Creating F-compliant PLC data types (UDT)

Create two F-compliant PLC data types (UDT), for example, for access to all channels.

The figure below shows an F-compliant PLC data type (UDT) for access to the channel values and value status of the inputs with 1oo2 evaluation:

4 F-DI/3 F-DO DC24V/2A_DI

	Name	Data type	Comment
1	CH_DI_0	Bool	Channel 0 (DI)
2	CH_DI_1	Bool	Channel 1 (DI)
3	CH_DI_2	Bool	Channel 2 (DI)
4	CH_DI_3	Bool	Channel 3 (DI)

The figure below shows the F-compliant PLC data type (UDT) for access to the channel values and value status of the outputs:

4 F-DI/3 F-DO DC24V/2A_DO

	Name	Data type	Comment
1	CH_DO_0	Bool	Channel 0 (DO)
2	CH_DO_1	Bool	Channel 1 (DO)
3	CH_DO_2	Bool	Channel 2 (DO)

Using F-compliant PLC data types (UDT)

As demonstrated in the figure below, you can use the two F-compliant PLC data types (UDT) that you have created in an F-FC (e.g. "Motor"):

Motor

	Name	Data type	Comment
1	Input		
2	Motor SS DI	"4 F-DI/3 F-DO DC24V/2A_DI"	Motor interface channel values DI
3	CH_DI_0	Bool	Kanal 0 (DI)
4	CH_DI_1	Bool	Kanal 1 (DI)
5	CH_DI_2	Bool	Kanal 2 (DI)
6	CH_DI_3	Bool	Kanal 3 (DI)
7	Motor SS DI VS	"4 F-DI/3 F-DO DC24V/2A_DI"	Motor interface value status DI
8	CH_DI_0	Bool	Kanal 0 (DI)
9	CH_DI_1	Bool	Kanal 1 (DI)
10	CH_DI_2	Bool	Kanal 2 (DI)
11	CH_DI_3	Bool	Kanal 3 (DI)
12	Motor SS DO VS	"4 F-DI/3 F-DO DC24V/2A_DO"	Motor interface value status DI
13	CH_DO_0	Bool	Kanal 0 (DO)
14	CH_DO_1	Bool	Kanal 1 (DO)
15	CH_DO_2	Bool	Kanal 2 (DO)
16	Output		
17	Motor SS DO	"4 F-DI/3 F-DO DC24V/2A_DO"	Motor interface channel values DO
18	CH_DO_0	Bool	Kanal 0 (DO)
19	CH_DO_1	Bool	Kanal 1 (DO)
20	CH_DO_2	Bool	Kanal 2 (DO)

Creating structured PLC tag for the F-module 4 F-DI/3 F-DO DC24V/2A

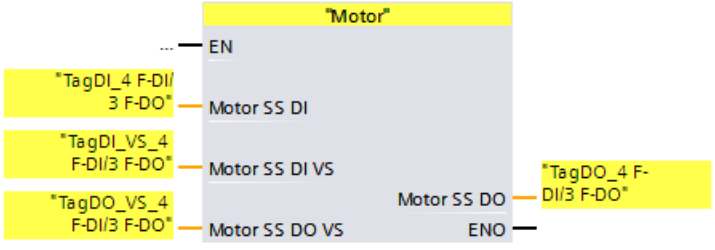
Create structured PLC tags for the F-module 4 F-DI/3 F-DO DC24V/2A:

Tag table_1

	Name	Data type	Address	Comment
1	TagDI_4 F-DI/3 F-DO	"4 F-DI/3 F-DO DC24V/2A_DI"	%I15.0	Motor 1 DI
2	TagDI_VS_4 F-DI/3 F-DO	"4 F-DI/3 F-DO DC24V/2A_DI"	%I16.0	Motor 1 DI VS
3	TagDO_VS_4 F-DI/3 F-DO	"4 F-DI/3 F-DO DC24V/2A_DO"	%I17.0	Motor 1 DO VS
4	TagDO_4 F-DI/3 F-DO	"4 F-DI/3 F-DO DC24V/2A_DO"	%Q15.0	Motor 1 DO

Accessing the F-FC

Transfer the structured PLC tags you have created when you call the F-FC (e.g. "Motor"):



See also

Addressing F-I/O (Page 157)
Value status (S7-1200, S7-1500) (Page 159)

5.1.7 Named value data types (S7-1500)

You use named value data types as in the standard.

For named value data types in a safety program, you can only use the data types supported by the safety program (Page 119) (except TIME) as base data types.

NOTE

If you change Named value data types, this also changes the collective F signature.

You can find more information on named value data types in the *STEP 7 online help* under "Named value data types" and "Declaring named value data types".

5.1.8 Editing PLC tags with external editors

To edit PLC tags with external editors follow the procedure as in the standard. Additional information can be found in the *STEP 7* help in "Editing PLC tags with external editors".

Note the following:

NOTE

After importing a tag table which contains tags used in the safety program, the collective F-signature of the safety program is reset.

To form the collective F-signature again you have to recompile the project data. For this, with access protection set up for the safety program, you need a valid access authorization for the safety program.

If you would like to edit PLC tags with external editors, we therefore recommend that you store PLC tags to be used in the safety program in a separate tag table.

5.1.9 Using Multiuser engineering

If you want to use Multiuser engineering, proceed as described in the *STEP 7* help under "Using Multiuser engineering".

5.1.10 Deleting the safety program

Deleting individual F-blocks

To delete an F-block, follow the same procedure as in *STEP 7*.

Deleting an F-runtime group

See Deleting an F-runtime group (Page 149)

(S7-300, S7-400) Remove all calls that you have used to call the safety program (Main_Safety).

Deleting the entire safety program for S7-300/400 F-CPUs *with* inserted memory card (SIMATIC Micro Memory Card or flash card)

To delete an entire safety program, proceed as follows:

1. Delete all F-blocks (shown with yellow symbol) in the project tree.
2. Remove all calls that you have used to call the safety program (Main_Safety).
3. Select the F-CPU in the *hardware and network editor* and clear the "F-capability activated" option in the properties of the F-CPU.
4. Compile the project data of the F-CPU

   The offline project no longer contains a safety program.
5. To delete a safety program on the Memory Card (SIMATIC Micro Memory Card or Flash Card), insert the Memory Card (SIMATIC Micro Memory Card or Flash Card) in the programming device or PC or in a SIMATIC USB prommer.
6. Select the menu command "Project > Card Reader/USB memory > Show Card Reader/USB memory" in the menu bar.
7. Open the "SIMATIC Card Reader" folder and delete the Memory Card.
8. Insert the Memory Card into the F-CPU.
9. Perform a memory reset of the F-CPU (flashing of the STOP LED requests a memory reset).

You can then download the offline standard user program to the F-CPU.

Deleting the entire safety program for S7-400 F-CPUs *without* inserted flash card

To delete an entire safety program, proceed as follows:

1. Delete all F-blocks (shown with yellow symbol) in the project tree.
2. Remove all calls that you have used to call the safety program (Main_Safety).
3. Select the F-CPU in the *hardware and network editor* and clear the "F-capability activated" option in the properties of the F-CPU.
4. Compile the project data of the F-CPU

   The offline project no longer contains a safety program.
5. Perform a memory reset on the F-CPU (in the "Online tools" task card of the F-CPU).

You can then download the offline standard user program to the F-CPU.

Delete the entire safety program for SIMATIC S7-1200/1500 F-CPUs

To delete an entire safety program, proceed as follows:

1. Delete all F-blocks (shown with yellow symbol) in the project tree.
2. Select the F-CPU in the *hardware and network editor* and clear the "F-capability activated" option in the properties of the F-CPU.
3. Compile the project data of the F-CPU

   The offline project no longer contains a safety program.

You can then download the offline standard user program to the F-CPU.

5.2 Defining F-Runtime Groups

5.2.1 Rules for F-Runtime Groups of the Safety Program

Rules

Note the following:

- The channels (channel values and value status) of an F-I/O can only be accessed from a single F-runtime group.
- Tags of the F-I/O DB of an F-I/O can only be accessed from one F-runtime group and only from that F-runtime group from which the channels or value status of this F-I/O are also accessed (if access is made).
- F-FBs can be used in more than one F-runtime group but they must be called with different instance DBs.
- Instance DBs of F-FB can only be accessed from the F-runtime group in which the associated F-FB is called.
- A tag of a global F-DB (except the F-global DB) can only be accessed from one F-runtime group (however, a global F-DB can be used in more than one F-runtime group).
- (S7-300, S7-400) A DB for F-runtime group communication can be read and write accessed by the F-runtime group to which it was assigned as "DB for runtime group communication", but only read-accessed by the "receiver" F-runtime group.
- (S7-300, S7-400) An F-communication DB can only be accessed from one F-runtime group.
- (S7-1200, S7-1500) You must not call the main safety block yourself. It is automatically called by the assigned F-OB.

NOTE

F-OBs are protected by F-system know-how. The OB start information of F-OBs therefore cannot be evaluated.

- (S7-1200, S7-1500). The F-OB should be created with the highest priority of all OBs.

  NOTE

  The cycle time/runtime of an F-OB can be prolonged by, among other things, communication load, the processing of higher-priority interrupts, as well as by test and commissioning functions. In a redundant S7-1500HF system, it is also dependent on the current system state.

- (S7-300, S7-400) The main safety block may only be called once from a standard block. Multiple calls can cause the F-CPU to go to STOP mode.
- (S7-300, S7-400) For optimal use of temporary local data, you must call the F-runtime group (the main safety block) directly in an OB (cyclic interrupt OB, if possible); you should not declare any additional temporary local data in this cyclic interrupt OB.
- (S7-300, S7-400) Within a cyclic interrupt OB, the F-runtime group should be executed **before** the standard user program; i.e. it should be called at the very beginning of the OB, so that the F-runtime group is always called at fixed time intervals, regardless of how long it takes to process the standard user program.

  For this reason, the cyclic interrupt OB should also not be interrupted by higher priority interrupts.
- The process image of the inputs and outputs of standard I/O, bit memory, and tags of DBs in the standard user program may be accessed either as read-only or read/write from more than one F-runtime group. (see also Data exchange between standard user program and safety program (Page 198))
- F-FCs can generally be called in more than one F-runtime group.

NOTE

You can improve performance by writing sections of the program that are not required for the safety function in the standard user program.

When determining which elements to include in the standard user program and which to include in the Safety program, you should keep in mind that the standard user program can be modified and downloaded to the F-CPU more easily. In general, changes in the standard user program do not require an acceptance.

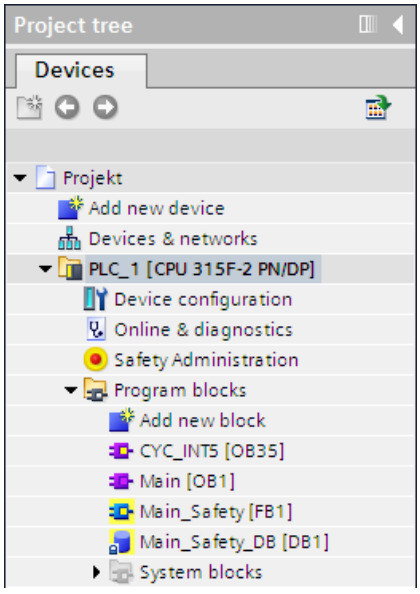
5.2.2 Procedure for defining an F-runtime group (S7-300, S7-400)

Requirements

- You have inserted an S7-300/400 F-CPU in your project.
- In the "Properties" tab of the F-CPU, the "F-capability activated" check box is selected (default setting).

F-runtime group created by default

STEP 7 Safety inserts F-blocks for an F-runtime group in the project tree by default after an F-CPU has been added. When you open the "Program blocks" folder, you see the (F-)blocks of the F-runtime group (CYC_INT5 [OB 35], Main_Safety [FB 1], and Main_Safety_DB [DB1]) in the project tree.



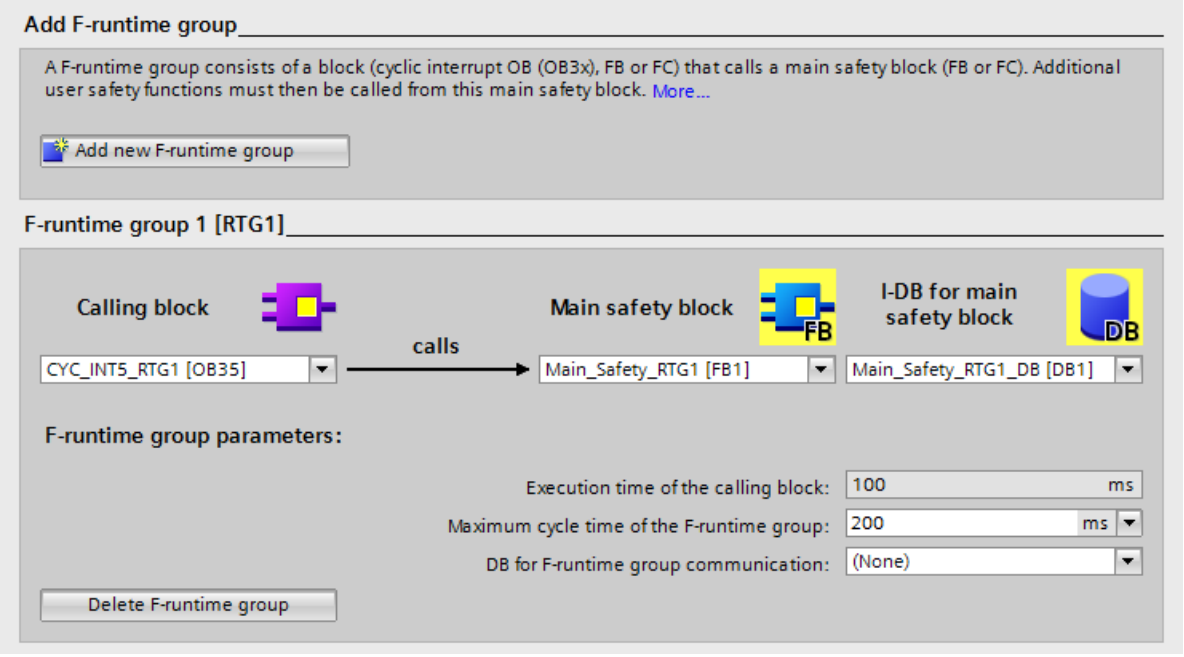
The following section describes how you modify the settings / parameters of the F-runtime group created by default or add and additional F-runtime group.

Procedure for defining an F-runtime group

Proceed as follows to define an F-runtime group:

1. Open the *Safety Administration Editor* by double-clicking in the project tree.
2. Select "F-runtime group" in the area navigation.

   Result: The work area for defining an F-runtime group with the (default) settings for F-runtime group 1 opens.



3. Specify the block in which the main safety block is to be called.

   Cyclic interrupt OB 35 is suggested here by default. The advantage of cyclic interrupt OBs is that they interrupt the cyclic program processing in OB 1 of the standard user program at fixed intervals. This means that the safety program is called and run at fixed time intervals in a cyclic interrupt OB.

   In this input field, you can select only those blocks that were created in the LAD, FBD, or STL programming language. If you select a block here, the call is inserted automatically into the selected block and, if necessary, removed from a previously selected block.

   If you want to call the main safety block in a block that was created in another programming language, you must program this call itself. The input field is then not editable (grayed out), and you can change the call only in the calling block and not the *Safety Administration Editor*.

4. Assign the desired main safety block to the F-runtime group. If the main safety block is an FB, you must also assign an instance DB.

   Main_Safety [FB1] and Main_Safety_DB [DB1] are suggested by default.

5. The F-CPU monitors the F-cycle time of the F-runtime group. For "Maximum cycle time of F-runtime group", enter the maximum permitted time between two calls of the F-runtime group.

⚠ WARNING

The call interval of the F-runtime group is monitored for a maximum value. This means that monitoring is performed to determine whether the call is executed often enough, but not whether it is executed too often or, for example, isochronously. Fail-safe timers must therefore be implemented using the TP, TON, or TOF instructions (Page 532) from the "Instructions" task card and not using counters (OB calls). *(S007)*

⚠ WARNING

The response time of your safety function depends, among other things, on the cycle time of the F-OB, the runtime of the F-runtime group and, when distributed F-I/O is used, the parameter assignment of PROFINET/PROFIBUS.

Therefore, the configuration/parameter assignment of the standard system influences the response time of your safety function.

Examples:

– Increasing the priority of a standard OB compared to an F-OB can extend the cycle time of the F-OB or the runtime of the F-runtime group due to the higher-priority processing of the standard OB. Note that during the creation of technology objects, OBs with very high priority may be created automatically.
– The change of the send clock cycle of PROFINET changes the cycle time of an F-OB with event class "Synchronous cycle".

Note that the configuration / parameter assignment of the standard system is not subject to access protection for the safety-related project data and does not lead to a modification of the Collective F-Signature.

If you do not take organizational measures to prevent changes in the configuration / parameter assignment of the standard system with influence on the response time, you must always use the monitoring times for the calculation of the maximum response time of a safety function (see Configuring monitoring times (Page 630)).

The monitoring times are protected against change with the access protection of the safety-related project data and are captured by the Collective F-Signature as well as the Collective F-SW-Signature.

When calculating with the Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) you need to consider the value that is specified for "Any standard system runtimes" as value for the maximum response time. *(S085)*

6. If one F-runtime group is to provide tags for evaluation to another F-runtime group of the safety program, assign a DB for F-runtime group communication. Select an F-DB for "DB for F-runtime group communication". (See also F-runtime group communication (S7-300, S7-400) (Page 143))
7. If you want to create a **second F-runtime group**, click the "Add new F-runtime group" button.

8. Assign an F-FB or F-FC as the main safety block to a calling block. This F-FB or F-FC is automatically generated in the project tree, if not already present.
9. If the main safety block is an F-FB, assign an instance DB to the main safety block. The instance DB is generated automatically in the project tree.
10. Follow steps 3 to 5 above to complete generation of the second F-runtime group.

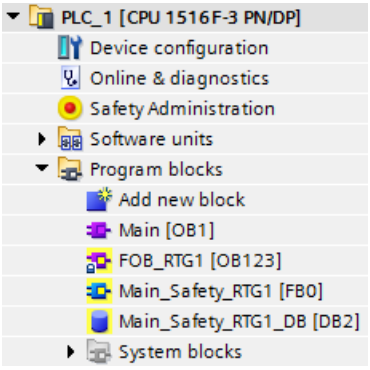
5.2.3 Procedure for defining an F-runtime group (S7-1200, S7-1500)

Requirements

- You have inserted an S7-1200/1500 F-CPU in your project.
- In the "Properties" tab of the F-CPU, the "F-capability activated" check box is selected (default setting).

F-runtime group created by default

STEP 7 Safety inserts F-blocks for an F-runtime group in the project tree by default after an F-CPU has been added. When you open the "Program blocks", you see the (F-)blocks of the F-runtime group (FOB_RTG1 [OB123], Main_Safety_RTG1 [FB1] and Main_Safety_RTG1_DB [DB1]) in the project tree.



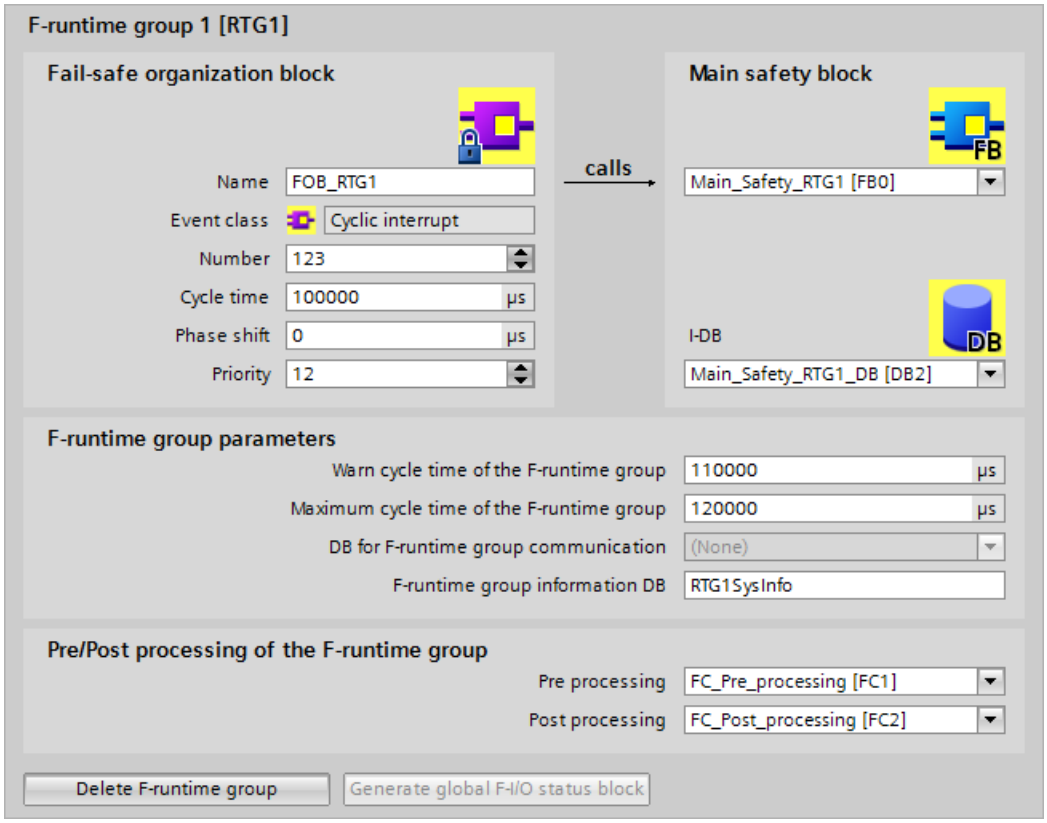
The following section describes how you modify the settings / parameters of the F-runtime group created by default or add and additional F-runtime group.

Procedure for defining an F-runtime group

Proceed as follows to define an F-runtime group:

1. Open the *Safety Administration Editor* by double-clicking in the project tree.
2. Select "F-runtime group" in the area navigation.

   Result: The work area for defining an F-runtime group with the (default) settings for F-runtime group 1 opens.



3. Assign a name for the F-OB under "F-OB".
4. Specify the event class of the F-OB when you create a new F-runtime group.

   For an F-OB you can select between the event classes "Program cycle", "Cyclic interrupt" or "Synchronous cycle".

   In the case of the F-runtime group created by default, the F-OB has the event class "Cyclic interrupt". To change the event class of the F-OB of an already created F-runtime group, you need to delete and F-runtime group and create a new one.

NOTE

We recommend creating the F-OB with the event class "Cyclic interrupt". The safety program will then be called and executed at fixed time intervals.

F-OBs with the event class "Synchronous cycle" are only recommended in conjunction with F-IOs that support isochronous mode, for example submodule "Profisafe Telgr 902" of drive SINAMICS S120 CU310-2 PN V5.1.

F-OBs with the event class "Program cycle" are not recommended, as these have the lowest priority "1" (see below).

NOTE

Note the maximum permissible number of OBs (incl. F-OBs) with event class "Synchronous cycle" (see technical specifications in the product manuals of the S7-1500 CPUs).

5. If required, you can manually change the number of the F-OB proposed by the system. To do so, note the number ranges valid for the relevant event class.
6. Assign cycle time, phase shift and priority parameters for F-OBs with event class "Cyclic interrupt".

   Assign the priority parameter for F-OBs with event class "Synchronous cycle".

   – Select a cycle time which is less than the "Maximum cycle time of the F-runtime group" and less than the "Cycle time warning limit of F-runtime group".
   – Select a phase shift which is less than the cycle time.
   – If possible, select a priority that is higher than the priority of all other OBs.

NOTE

Through a high priority of the F-OB you ensure that the run time of the safety program and the reaction time of your safety functions (Page 630) are influenced as little as possibly by the standard user program.

NOTE

For F-OBs with event class "Synchronous cycle" you need to also assign parameters for application cycle (ms) and possibly delay time (ms) after defining the F-runtime group and the connection of the isochronous F-I/O to the isochronous mode interrupt OB. You can find these parameters in the "Properties" dialog box of the isochronous mode interrupt OB in the "Isochronous mode" group. Proceed as described in the STEP 7 help under "Configuring isochronous mode interrupt OBs".

7. Assign the calling main safety block to the F-OB. If the main safety block is an FB, you must also assign an instance DB.

   Main_Safety_RTG1 [FB1] and Main_Safety_RTG1_DB [DB1] are suggested by default.

8. The F-CPU monitors the F-cycle time of the F-runtime group. Two parameters are available:
   - If the "Warn cycle time of the F-runtime group" is exceeded, a maintenance demanded is written to the diagnostics buffer of the F-CPU and the MAINT LED of the F-CPU is activated. This parameter can, for example, be used to determine whether the cycle time exceeds a required value without the F-CPU switching to STOP mode.

     NOTE

     The maintenance demanded is marked as an outgoing event by a STOP/RUN transition of the F-CPU. In an HF-system you must set both HF-CPUs or the redundant S7-1500HF system to STOP before you restart the HF-CPUs.

     As an alternative, you can mark the maintenance demanded as outgoing with the standard instruction "ACK_FCT_WARN". To do so, call the instruction "ACK_FCT_WARN" with a rising edge at the input parameter "ACK_WARN" in your standard user program.

     A renewed violation of the warning limit is not signaled until after a STOP/RUN transition of the F-CPU.

   - If the "Maximum cycle time of F-runtime group" is exceeded, the F-CPU will go to STOP. For "Maximum cycle time of F-runtime group", select the maximum time that can elapse between two calls of this F-runtime group (max. 20000000 µs).

     ⚠ WARNING

     The call interval of the F-runtime group is monitored for a maximum value. This means that monitoring is performed to determine whether the call is executed often enough, but not whether it is executed too often or, for example, isochronously. Fail-safe timers must therefore be implemented using the TP, TON, or TOF instructions (Page 532) from the "Instructions" task card and not using counters (OB calls). *(S007)*

⚠ WARNING

The response time of your safety function depends, among other things, on the cycle time of the F-OB, the runtime of the F-runtime group and, when distributed F-I/O is used, the parameter assignment of PROFINET/PROFIBUS.

Therefore, the configuration/parameter assignment of the standard system influences the response time of your safety function.

Examples:

– Increasing the priority of a standard OB compared to an F-OB can extend the cycle time of the F-OB or the runtime of the F-runtime group due to the higher-priority processing of the standard OB. Note that during the creation of technology objects, OBs with very high priority may be created automatically.

– The change of the send clock cycle of PROFINET changes the cycle time of an F-OB with event class "Synchronous cycle".

Note that the configuration / parameter assignment of the standard system is not subject to access protection for the safety-related project data and does not lead to a modification of the Collective F-Signature.

If you do not take organizational measures to prevent changes in the configuration / parameter assignment of the standard system with influence on the response time, you must always use the monitoring times for the calculation of the maximum response time of a safety function (see Configuring monitoring times (Page 630)).

The monitoring times are protected against change with the access protection of the safety-related project data and are captured by the Collective F-Signature as well as the Collective F-SW-Signature.

When calculating with the Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) you need to consider the value that is specified for "Any standard system runtimes" as value for the maximum response time. *(S085)*

The "Cycle time warning limit of F-runtime group" must be configured as less than or equal to the "Maximum cycle time of F-runtime group".

9. If necessary, you can change the name suggested by the system for the F-runtime group information DB (Page 154) under "F-runtime group information DB".
10. If required, you can select blocks of the standard program (FCs) with regard to preprocessing or postprocessing of an F-runtime group (see Pre-/postprocessing (S7-1200, S7-1500) (Page 88))
11. If you want to create a **second F-runtime group**, click the "Add new F-runtime group" button. Follow steps 3 to 10 above.

5.2.4 F-runtime group communication (S7-300, S7-400)

Safety-related communication between F-runtime groups

Safety-related communication can take place between the two F-runtime groups of a safety program. This means fail-safe tags can be provided by one F-runtime group in an F-DB and read in another F-runtime group.

NOTE

A DB for F-runtime group communication can be read and write accessed by the F-runtime group to which it was assigned as "DB for runtime group communication", while it can only be read-accessed by the "receiver" F-runtime group.

Tip: You can improve performance by structuring your safety program in such a way that as few tags as possible are exchanged between the F-runtime groups.

Procedure for defining a DB for F-runtime group communication

You define the DB for F-runtime group communication in the work area "F-runtime groups". Proceed as follows:

1. Click "F-runtime groups" in "*Safety Administration Editor*".
2. Select an existing F-DB in the "DB for F-runtime group communication" field or assign a new one.
3. Assign a name to the F-DB.

Up-to-dateness of tags read from another F-runtime group

NOTE

Tags read are up-to-date as at the time of the last completed processing cycle of the F-runtime group providing the tags prior to start-up of the F-runtime group reading the tags.

If the tags supplied undergo multiple changes during runtime of the F-runtime group supplying the tags, the F-runtime group reading the tags only receives the last change (see figure below).

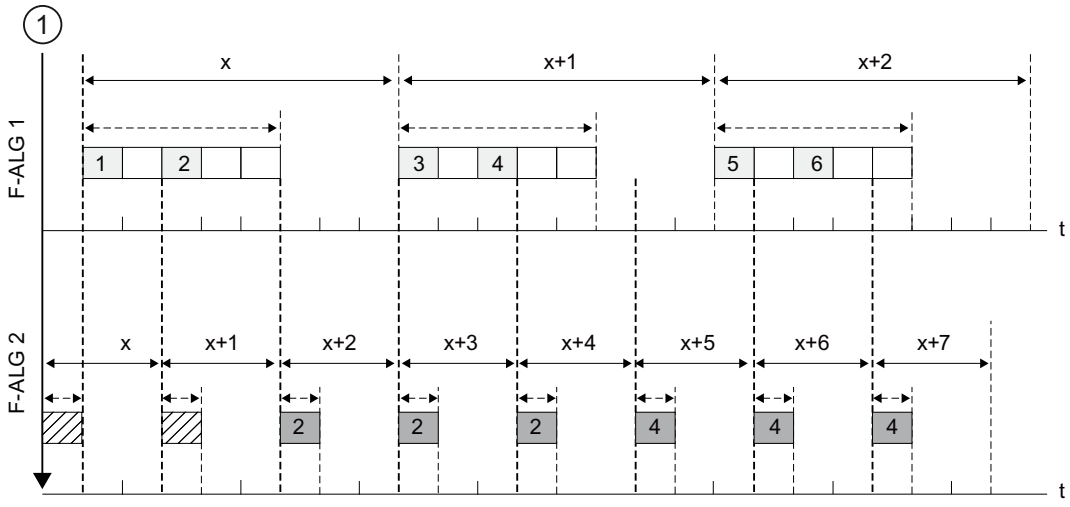
Assignment of fail-safe values

After F-system start-up, fail-safe values are supplied to the F-runtime group which has read access to tags in the DB for F-runtime group communication of another F-runtime group (for example, F-runtime group 2). These are the values you specified as initial values in the DB for F-runtime group communication of F-runtime group 1.

F-runtime group 2 reads the fail-safe values the first time it is called. The second time the F-runtime group 2 is called, it reads the latest tags if F-runtime group 1 has been processed completely between the two calls of F-runtime group 2. If F-runtime group 1 has not been processed completely, F-runtime group 2 continues to read the fail-safe values until F-runtime group 1 is completely processed.

The behavior is illustrated in the two figures below.

Reading tags from F-runtime group 1, which has a longer OB cycle and lower priority than F-runtime group 2



① Startup of F-system

Cycle time of the (F-)OB in which the F-runtime group is called.

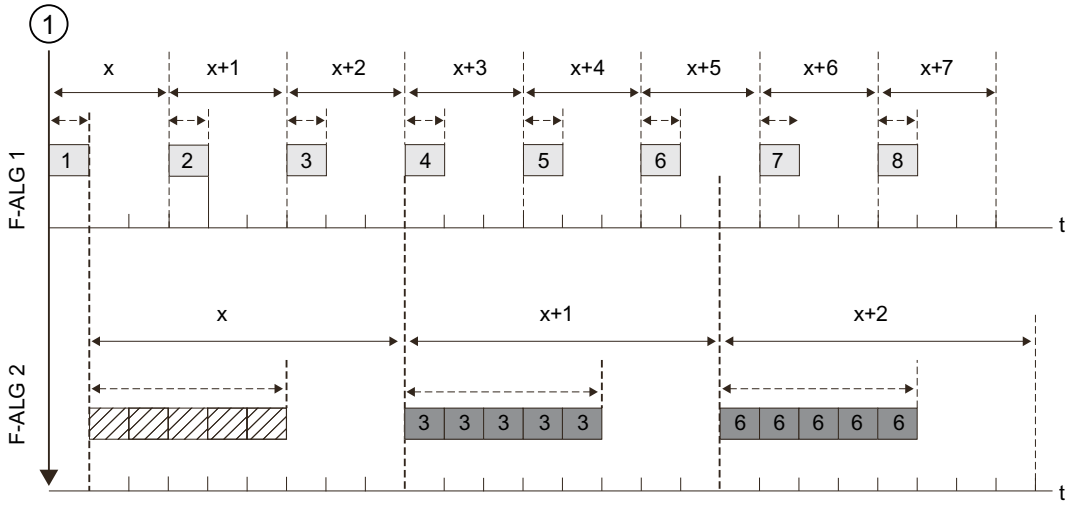
Runtime of the F-runtime group

1 ... Tag of F-runtime group 1, written to DB for F-runtime group communication of F-runtime group 1

2 ...Tag of F-runtime group 2, read in DB for F-runtime group communication of F-runtime group 1

Initial value in the DB for F-runtime group communication

Reading tags from F-runtime group 1, which has a shorter OB cycle and higher priority than F-runtime group 2



① Startup of F-system

Cycle time of the (F-)OB in which the F-runtime group is called.

Runtime of the F-runtime group

1 ... Tag of F-runtime group 1, written to DB for F-runtime group communication of F-runtime group 1

2 ... Tag of F-runtime group 2, read in DB for F-runtime group communication of F-runtime group 1

Initial value in the DB for F-runtime group communication

F-runtime group supplying tags is not processed

NOTE

If the F-runtime group whose DB for F-runtime group communication is to supply the tags is not processed (the main safety block of the F-runtime group is not called), the F-CPU goes to STOP mode. One of the following diagnostics events is then entered in the diagnostics buffer of the F-CPU:

- Error in safety program: cycle time exceeded
- Number of the relevant main safety block (of F-runtime group that is not processed)
- Current cycle time in ms: "0"

5.2.5 F-runtime group communication (S7-1200, S7-1500)

Introduction

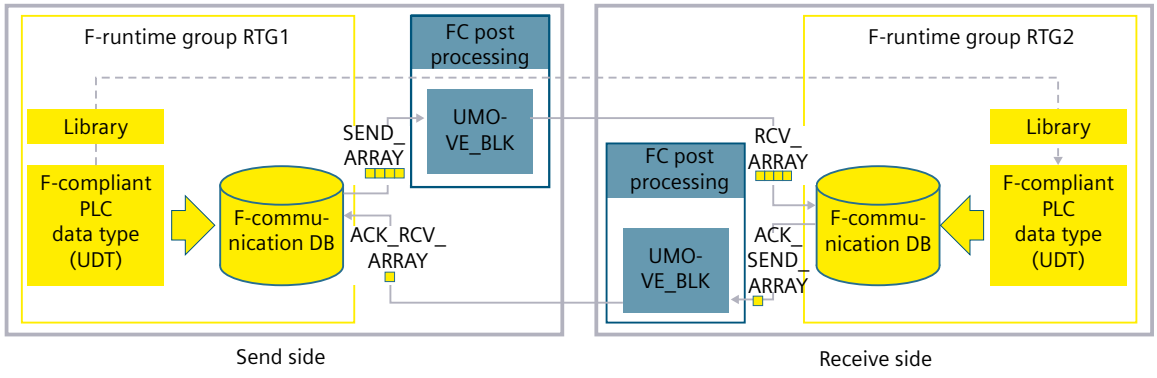
With the help of Flexible F-Link you realize a F-runtime group communication.

With Flexible F-Link, a coded F-array is made available for the send data of the F-runtime group. The coded F-array is transferred to the other F-runtime group with standard instructions such as UNMOVE_BLK.

Requirement

- S7-1500 F-CPUs as of firmware V2.0
- S7-1200 G1 F-CPUs as of firmware V4.2
- S7-1200 G2 F-CPUs as of firmware V1.0
- Safety system version as of V2.2

F-runtime group communication



If you want to send a data fail-safe from one F-runtime group to another F-runtime group, follow these steps:

1. Create a non-nested F-compliant PLC data type (UDT) for the F-runtime group communication. The size can be up to 1500 bytes, including a maximum of 800 elements of data type BOOL.
2. Create two safety-related communications for F-runtime group communication in the Safety Administration Editor in the "Flexible F-Link" area. One safety-related communication each for the send and receive sides.

3. Configure the same F-monitoring time and F-communication UUID for each F-runtime group communication relationship.

   Information on calculating the F-monitoring times can be found in Monitoring and response times (Page 629).

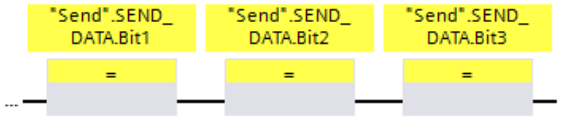
   For example:

Flexible F-Link settings

	Name	PLC Data Type	Direction	F-monitoring time: (ms)	F-communication UUID
1	Send	Data	Send	500	5d61f443-ede6-4e4e-9db6-e79af634f9ad
2	Receive	Data	Receive	500	5d61f443-ede6-4e4e-9db6-e79af634f9ad

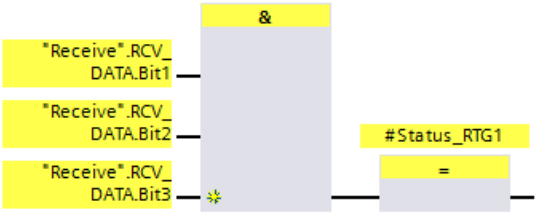
4. On the send side (e.g. RTG1) supply the data of the F-communication DB (Page 346) for the send direction with the data to be sent.

   For example:

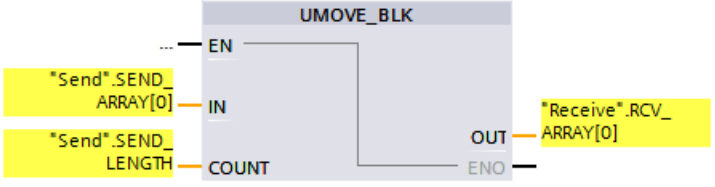


5. On the receive side (e.g. RTG2), read the received data from the F-communication DB (Page 346) for the receive direction.

   For example:



6. Call the instruction "UMOVE_BLK" in the F-runtime group for the send data (e.g. RTG1) in the FC for post processing (Page 88).
7. Interconnect the "UMOVE_BLK" instruction for the data to be sent as follows:

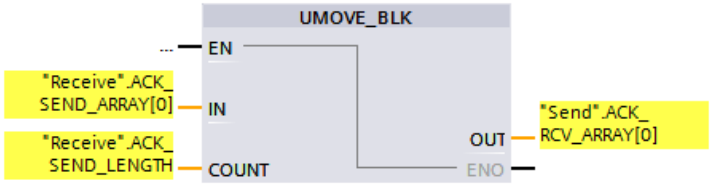


   "Send" is the F-communication DB (Page 346) of the F-runtime group that sends the data.

   "Receive" is the F-communication DB (Page 346) of the F-runtime group that receives the data.

8. Call the instruction "UMOVE_BLK" in the F-runtime group for acknowledgment (e.g. RTG2) in the FC for post processing (Page 97).

9. Interconnect the "UMOVE_BLK" instruction for the acknowledgment connection as follows:



"Receive" is the F-communication DB (Page 346) of the F-runtime group that sends the acknowledgment telegram.

"Send" is the F-communication DB (Page 346) of the F-runtime group that receives the acknowledgment telegram.

10. Compile the user program.

11. Download the user program to the F-CPU.

⚠ WARNING

For each communication connection via Flexible F-Link, you must check that the offsets of the elements of the F-compliant PLC data type (UDT) used for the communication connection match on the transmit and receive sides when you accept the safety program.

You can find the offsets in the respective safety summary. To do this identify the communication connection by means of the UUID. *(S088)*

⚠ WARNING

When a new Flexible F-Link communication is created in the Safety Administration Editor, a unique F-communication UUID for the communication is provided by the system. When communication is copied in the Safety Administration Editor within the parameterization table or copied to another F-CPU, the F-communication UUIDs are not regenerated and therefore not unique anymore. If the copy is used to configure a new communication relationship, you yourself must ensure the uniqueness. To do this, select the relevant UUIDs and generate them again via the "Generate UUID" context menu. You must ensure the uniqueness in the safety summary during acceptance, at the latest. *(S087)*

⚠ WARNING

It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time. *(S018)*

Up-to-dateness of tags read from another F-runtime group

The same statements as those in the section "F-runtime group communication (S7-300, S7-400) (Page 143)" apply (except for the storage locations written or read or initial values).

5.2.6 Deleting an F-runtime group

Deleting an F-runtime group

To delete an F-runtime group, proceed as follows:

1. In the area navigation of the *Safety Administration Editor*, click on the F-runtime group to be deleted.
2. Select the "Delete F-runtime group" button in the work area.
3. Confirm the dialog with "Yes".
4. Perform a Compilation of the safety program (Page 349) (menu command "Edit > Compile") for your changes to take effect.

The assignment of the F-blocks to an F-runtime group (to the calling block of the main safety block) is deleted. However, the F-blocks continue to exist.

5.2.7 Changing the F-runtime group (S7-300, S7-400)

Changing an F-runtime group

You can make the following changes for each F-runtime group in your safety program in the corresponding "F-runtime group" work area:

- Specify another block as the calling block of the main safety block.
- Specify another F-FB or F-FC as main safety block.
- Enter a different or new I-DB for the main safety block.
- Change the value for the maximum cycle time of the F-runtime group
- Specify another DB as a data block for F-runtime group communication.

5.2.8 Changing the F-runtime group (S7-1200, S7-1500)

Changing an F-runtime group

You can make the following changes for each F-runtime group in your safety program in the corresponding "F-runtime group" work area:

- Change the name, number, cycle time, phase shift and priority of the F-OB.
- Specify another F-FB or F-FC as main safety block.
- Enter a different or new I-DB for the main safety block.
- Change the value for the maximum cycle time and the cycle time warning limit of the F-runtime group.

- Assign another name for the F-runtime group information DB.
- Specify an FC for the preprocessing and postprocessing.

5.3 Creating F-blocks

5.3.1 Creating F-blocks

Introduction

In order to create F-FBs, F-FCs, and F-DBs for the safety program, you should follow the same basic procedure as in the standard. In the following, only the deviations from the procedure for standard blocks are presented.

Creating F-FBs, F-FCs, and F-DBs

You create F-blocks in the same way as for standard blocks. Proceed as follows:

1. In the project tree of the F-CPU or the Safety Unit, double-click on "Add new block" under "Program blocks".
2. In the dialog that appears, specify the type, name, and language and select the "Create F-block" check box. (If you do not select the check box, a standard block is created.)
3. After the dialog is confirmed, the F-block is opened in the *Program editor*.

Note the following

Note the following important instructions:

NOTE

- You may not declare block parameters in the block interface of the main safety block because they cannot be supplied.
- (S7-1200, S7-1500) You can edit start values in instance DBs.
- The function "Apply actual values" is not supported.
- You may not access static local data in single instances or multi-instances of other F-FBs.
- You must always initialize outputs of F-FCs.

  The F-CPU can go to STOP mode if the information above is not observed. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.
- (S7-300, S7-400) If you wish to assign an address from the data area (data block) to a formal parameter of an F-FC as an actual parameter, you have to use fully qualified DB access.
- Its inputs may only be accessed by a block in reading mode and its outputs only in writing mode.

  Use an in/out if you wish to have both read and write access.
- For greater clarity, assign meaningful names to the F-blocks you have created.

Copying/pasting F-blocks

You can copy F-FBs, F-FCs, and F-DBs in exactly the same was as blocks of the standard user program.

Exceptions:

- (S7-1200, S7-1500) You may not copy an F-OB.
- You cannot copy blocks from the folder "Program blocks > System blocks".
- (S7-1500) F-blocks cannot be copied from the Safety Unit to a standard unit or between the Safety Unit and the "Program blocks" folder of the F-CPU.

See also

Changing the F-runtime group (S7-1200, S7-1500) (Page 149)

5.3.2 Know-how protection

For know-how protection of F-blocks, proceed as described in the *STEP 7* help under "Protecting blocks".

Requirements

Note the following with regard to know-how protection of F-blocks for S7-1200/1500 F-CPUs:

- An F-block to which you wish to assign know-how protection must be called in the Safety program.
- Before you can set up the know-how protection for an F-block, the Safety program must be consistent. For this purpose, compile (Page 349) the Safety program.

NOTE

No source code is output in the safety summary for know-how-protected F-blocks. Therefore, create the safety summary (for example, for carrying out a code review or for F-block acceptance) before you set up the know-how protection.

NOTE

If you want to edit the program code and/or the block interface of a know-how protected F-block, we recommend that you do not open the F-block by entering a password, but rather remove the know-how protection completely and set it up again after compiling.

NOTE

(S7-1200, S7-1500) When a know-how protected F-block or F-blocks called by it are renamed, the signature of the know-how protected F-block is not changed until the password is entered when opening or removing the know-how protection.

NOTE

When know-how protected F-blocks are used, warnings and error messages whose cause can lie in the know-how protected F-blocks can be displayed during compilation of the Safety program. The warnings and error messages contain corresponding information. Example: In a know-how protected F-block, you perform read access to a tag of the standard user program to which write access is taking place in a different (know-how protected) F-block.

For S7-1200/1500 F-CPUs, you can obtain additional information in the section "Know-how protected F-blocks in the Safety program" of the safety summary.

See also

Reuse of F-blocks (Page 152)

5.3.3 Reuse of F-blocks

Introduction

You can reuse F-blocks that you have already tested and, if applicable, approved, in other Safety programs – without having to test and approve them again.

You can protect the content of the F-block by setting up the know-how protection. You may only set this protection after acceptance of the F-block to ensure that the safety summary contains this F-block in its entirety.

Like standard blocks, you can store F-blocks as master copies or types in global libraries or in the project library.

More information can be found in the *STEP 7 help* under "Using libraries".

Creating safety summary for the F-block to be reused

Create a safety summary with the following information for F-blocks that you want to reuse.

S7-300/400 F-CPUs

- Signature and initial value signature of the know-how protected F-block
- Versions of all the used versioned LAD/FBD instructions
- Signatures and initial value signatures of all called F-FBs and F-FCs
- Signatures of all F-DBs accessed by the fail-safe block to be reused.

S7-1200/1500 F-CPUs

- Signature of the know-how protected F-block
- Safety system version when setting up the know-how protection
- Versions of all the used versioned LAD/FBD instructions
- Signatures of all called F-FBs and F-FCs
- Signatures of all F-DBs accessed by the fail-safe block to be reused.

The safety summary should also contain a description of the functionality of the F-block, in particular if it is know-how protected.

The required information is obtained by creating the safety summary of the Safety program in which you originally created, tested and approved the F-block to be reused.

This safety summary can also serve directly as the safety summary for the F-block to be reused.

Checks when using the F-block to be reused

When reusing the F-block, ensure the following:

- The signature and initial value signature (S7-300, S7-400) of the F-block are unchanged.
- (SIMATIC S7-1200, S7-1500) The documented safety system version is set.
- The documented (or functionally identical) versions of the versioned LAD/FBD instructions are set. You can find information about the instruction versions in the description of the instructions.
- (S7-1200, S7-1500) The called and accessed F-blocks with the documented signatures are used.
- (S7-300, S7-400) The called F-blocks with the documented signatures and initial value signatures are used.

If the version conflicts cannot be eliminated due to other dependencies, contact the author of the know-how protected block in order to obtain a compatible approved version.

See also

Compliance of the know-how protected F-blocks used in the Safety program with their safety summary. (Page 417)

5.4 Information interface to the safety program

5.4.1 F-shared DB (S7-300, S7-400)

Function

The F-shared DB is a fail-safe data block that contains all of the shared data of the safety program and additional information needed by the F-system. The F-shared DB is automatically inserted when the hardware configuration is compiled.

Using its name F_GLOBDB, you can evaluate certain data elements of the safety program in the standard user program.

Reading an F-shared DB in the standard user program

You can read out the following information in the F-shared DB in the standard user program or on an operator control and monitoring system:

- Operating mode: safety mode or disabled safety mode ("MODE" tag)
- Error information "Error occurred when executing safety program" ("ERROR" tag)
- Collective F-Signature ("F_PROG_SIG" tag)
- Compilation date of the safety program ("F_PROG_DAT" tag, DATE_AND_TIME data type)

You use fully qualified access to access these tags (e.g. ""F_GLOBDB".MODE").

5.4.2 F-runtime group information DB (S7-1200, S7-1500)

Introduction

The F-runtime group information DB provides key information on the corresponding F-runtime group and on the safety program as a whole.

The F-runtime group information DB is generated automatically when an F-runtime group is created. A symbol, for example, "RTG1SysInfo", is assigned for the F-runtime group information DB. You can change the name in the *Safety Administration Editor*.

You must not access the information in the "F_SYSINFO" parameter from the safety program.

Information in the F-runtime group information DB

The F-runtime group information DB provides the following information:

Name	Data type	For processing in the safety program	For processing in the standard user program	Description
MODE	BOOL	x	x	1 = Disabled safety mode
MODE_REMAINING_TIME	TIME	x	x	Remaining runtime in deactivated safety mode until the F-CPU switches to STOP [1]
MODE_VPLC_TIME_SERVICE	BOOL	—	x	1 = Redundant operation of the Virtual Time Service application [2]
F_SYSINFO				
MODE	BOOL	—	x	1 = Disabled safety mode
TCYC_CURR	DINT	—	x	Current cycle time of the F-runtime group, in ms
TCYC_LONG	DINT	—	x	Longest cycle time of the F-runtime group, in ms
TRTG_CURR	DINT	—	x	Current runtime of the F-runtime group, in ms
TRTG_LONG	DINT	—	x	Longest runtime of the F-runtime group, in ms
T1RTG_CURR	DINT	—	x	Not supported by *STEP 7 Safety V21*.
T1RTG_LONG	DINT	—	x	Not supported by *STEP 7 Safety V21*.
F_PROG_SIG	DWORD	—	x	Collective F-Signature of the safety program
F_PROG_DAT	DTL	—	x	Compilation date of the safety program
F_RTG_SIG	DWORD	—	x	F-runtime groups signature
F_RTG_DAT	DTL	—	x	Compilation date of the F-runtime group
VERS_S7SAF	DWORD	—	x	Version identifier for *STEP 7 Safety*

1 When the F-CPU switches to STOP after the runtime for the deactivated safety mode has expired, the remaining time still present in the last cycle is displayed.

2 Only relevant for S7-1500 F Virtual Controller.

You access the content of the F-runtime group information DB with fully qualified addressing. Either collectively with the F_SYSINFO PLC data type (UDT), for example, "RTG1SysInfo.F_SYSINFO", provided by the F-system or individual information, for example, "RTG1SysInfo.F_SYSINFO.MODE".

See also

Program identification (Page 384)

5.5 Programming startup protection

Introduction

⚠ WARNING

Unintentional restart

In contrast to the standard user program, when the safety program starts up, all tags of the F-DBs are generally initialized with their start values. This means that saved error information is lost. The F-system performs automatic reintegration of the F-I/O.

If an unintentional startup of your process is not allowed, you must program restart/startup protection in the safety program. For this, the output of process values must be blocked until user acknowledgment takes place (see "Implementation of user acknowledgment (Page 190)"). This user acknowledgment must not take place until safe output of process values is possible.

You must expect an unintentional restart in the following cases:

- After a STOP generated by PG/PC operation, mode switch, communication function or "STP" instruction
- After an operating error
- After a STOP caused by a fault reaction function
- The following applies to the S7-1500 F Virtual Controller:

  After a STOP or a restart of the S7-1500 F Virtual Controller via Industrial Edge Management.

(S008)

Example of restart/startup prevention

In order to implement a restart/startup prevention, it must be possible to detect a startup. To detect a startup, you declare a tag of data type BOOL with initial value "TRUE" in an F-DB.

Block the output of process data when this tag has the value "1," for example, by passivating F-I/O using the PASS_ON tag in the F-I/O DB.

If the output of the process values is safely possible and errors have been corrected, reset this tag by a user acknowledgment.

See also

Implementing User Acknowledgment in the Safety Program of the F-CPU of a DP Master or IO controller (Page 190)

Implementing user acknowledgment in the safety program of the F-CPU of a I-slave or I-device (Page 194)

F-I/O DB (Page 164)

F-I/O access

6

The following sections contain a general description of F-I/O access.

The following sections are only valid to a limited extent for technology modules (e.g. F-TM Count 1x1Vpp sin/cos HF). You can find more detailed information in the respective equipment manuals.

For F-I/O with integrated safety functions (such as ET 200ecoPN M12-L F), outputs can be written from both the safety program of the F-CPU and from the integrated safety functions. You can find information on which values ultimately take effect here in the respective manuals.

6.1 Addressing F-I/O

Overview

Below you will find a description of how to address the F-I/O in the safety program and which rules must be observed in the process.

Addressing via the process image

As with standard I/O, you access F-I/O (e.g., S7-1500/ET 200MP fail-safe modules) via the **process image** (PII and PIQ).

Direct reading (with I/O identification ":P") of inputs or writing of outputs is not possible in the safety program.

Updating the process image

The process image of the inputs of F-I/O is updated at the start of the F-runtime group. The process image of the outputs of F-I/O is updated at the end of the F-runtime group (see Program structure of the safety program (S7-300, S7-400) (Page 111) or Program structure of the safety program (S7-1200, S7-1500) (Page 113)). For more information on updating the process image, refer to the note in Data Transfer from the Safety Program to the Standard User Program (Page 198).

The communication required between the F-CPU (process image) and the F-I/O to update the process image uses a special safety protocol in accordance with PROFIsafe.

Rules

- You may only address a channel (channel value and value status) of an F-I/O in **one** F-runtime group. The first programmed addressing defines the assignment for the F-runtime group.
- You may only address a channel (channel value and value status) of an F-I/O with a unit that matches the data type of the channel.
  Example: To access input channels of data type BOOL, you must use the "input (bit)" (I x.y) unit. Access to the 16 consecutive input channels of the data type BOOL via the unit "input word" (IW x) is not possible.
- Analog channels always need to be entered in the tag table with data type INT or DINT. The data types WORD, DWORD and TIME are not permissible.
- Address only inputs and outputs that reference an actually existing channel (channel value and value status) (e.g. for an F-DO 10xDC24V with start address 10 only the outputs Q10.0 to Q11.1 for the channel values and the inputs I10.0 to I11.1 for the value status). Keep in mind that due to the special safety protocol, the F-I/O occupies a larger area of the process image than is required for the existing and enabled channels on the F-I/O (channel values and value status). To find the area of the process image where the channels (channel values and value status) are stored (channel structure), refer to the relevant manuals for the F-I/O.
- Channels can be disabled for certain F-I/O (such as ET 200SP fail-safe modules or S7-1500/ET 200MP fail-safe modules). Address only channels (channel value and value status) that are enabled in the hardware configuration. When you address channels that are disabled in the hardware configuration, a warning may be output when compiling the safety program.
- For certain F-I/O (such as ET 200SP fail-safe modules or S7-1500/ET 200MP fail-safe modules), a "1oo2 (2v2) sensor evaluation" can be specified. Two channels are grouped into one channel pair in this case, and the result of the "1oo2 sensor evaluation" is usually provided under the address of the channel with the lower channel number (see relevant manuals of the F-I/O). Address only this channel (channel value and value status) of the channel pair. When you address a different channel, a warning may be output when compiling the safety program.

⚠ WARNING

If you use an additional component between the F-CPU (S7-300, S7-400) and the F-I/O that copies the safety message frame in accordance with PROFIsafe between the F-CPU (S7-300, S7-400) and F-I/O per user program, you must test all safety functions affected by the copy function whenever you change the user-programmed copy function. *(S049)*

See also

Safety-related I-slave-slave communication - F-I/O access via process image (Page 189)
Value status (S7-1200, S7-1500) (Page 159)

6.2 Value status (S7-1200, S7-1500)

Properties

The value status is additional binary information for the channel value of an F-I/O. The value status is entered in the process image input (PII).

The value status is supported by fail-safe modules S7-1500/ET 200MP, ET 200SP, ET 200SP HA, ET 200AL, ET 200eco PN, ET 200S, ET 200iSP, ET 200pro, S7-1200 or F-SMs S7-300, fail-safe GSD based IO devices as well as fail-safe GSD based DP slaves that support the "RIOforFA-Safety" profile. Information about the value status can be found in the documentation of the respective F-I/O.

We recommend you amend the name of the channel value with "_VS" for the value status, for example, "TagIn_1_VS".

The value status provides information on the validity of the corresponding channel value:

- 1: A valid process value is output for the channel.
- 0: A fail-safe value is output for the channel.

The channel value and value status of an F-I/O can only be accessed from the same F-runtime group.

Location of value status bits in the PII for F-I/O with digital inputs

The value status bits come straight after the channel values in the PII.

Table 6-1 Example: Address assignment in PII for F-I/O with 16 digital input channels

Byte in the F-CPU	Assigned bits in F-CPU per F-I/O:							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
x + 0	DI_7	DI_6	DI_5	DI_4	DI_3	DI_2	DI_1	DI_0
x + 1	DI_{15}	DI_{14}	DI_{13}	DI_{12}	DI_{11}	DI_{10}	DI_9	DI_8
x + 2	Value status DI_7	Value status DI_6	Value status DI_5	Value status DI_4	Value status DI_3	Value status DI_2	Value status DI_1	Value status DI_0
x + 3	Value status DI_{15}	Value status DI_{14}	Value status DI_{13}	Value status DI_{12}	Value status DI_{11}	Value status DI_{10}	Value status DI_9	Value status DI_8

x = Module start address

The location of the channel values in the PII can be found in the device manual for the F-I/O.

Location of value status bits in the PII for F-I/O with digital outputs

The value status bits in the PII are mapped with the same structure as the channel values in the PIQ.

Table 6-2 Example: Address assignment in PIQ for F-I/O with 4 digital output channels

Byte in the F-CPU	Assigned bits in F-CPU per F-I/O:							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
x + 0	—	—	—	—	DQ_3	DQ_2	DQ_1	DQ_0

x = Module start address

Table 6-3 Example: Address assignment in PII for F-I/O with 4 digital output channels

Byte in the F-CPU	Assigned bits in F-CPU per F-I/O:							
	7	6	5	4	3	2	1	0
x + 0	—	—	—	—	Value status DQ_3	Value status DQ_2	Value status DQ_1	Value status DQ_0

x = Module start address

The location of the channel values in the PIQ can be found in the device manual for the F-I/O.

Location of value status bits in the PII for F-I/O with digital outputs and digital inputs

The value status bits come directly after the channel values in the PII in the following order:

- Value status bits for the digital inputs
- Value status bits for the digital outputs

Table 6-4 Example: Address assignment in PIQ for F-I/O with 2 digital input channels and 1 digital output channel

Byte in the F-CPU	Assigned bit in the F-CPU per F-I/O:							
	7	6	5	4	3	2	1	0
x + 0	—	—	—	—	—	—	—	DQ_0

x = Module start address

Table 6-5 Example: Address assignment in PII for F-I/O with 2 digital input channels and 1 digital output channel

Byte in the F-CPU	Assigned bits in F-CPU per F-I/O:							
	7	6	5	4	3	2	1	0
x + 0	—	—	—	—	—	—	DI_1	DI_0
x + 1	—	—	—	—	—	—	Value status DI_1	Value status DI_0
x + 2	—	—	—	—	—	—	—	Value status DQ_0

x = Module start address

The location of the channel values in the PII and PIQ can be found in the device manual for the F-I/O.

Location of value status bits in the PII for F-I/O with analog inputs

The value status bits come straight after the channel values in the PII.

Table 6-6 Example: Address assignment in PII for F-I/O with 6 analog input channels (data type INT)

Byte in the F-CPU	Assigned bytes/bits in the F-CPU per F-I/O:							
	7	6	5	4	3	2	1	0
x + 0	Channel value AI_0							
...	...							
x + 10	Channel value AI_5							
x + 12	—	—	Value status AI_5	Value status AI_4	Value status AI_3	Value status AI_2	Value status AI_1	Value status AI_0

x = Module start address

The location of the channel values in the PII can be found in the device manual for the F-I/O.

Location of value status bits in the PII for F-I/O with analog outputs

The value status bits are mapped in the PII.

Table 6-7 Example: Address assignment in PIQ for F-I/O with 6 analog output channels (data type INT)

Byte in the F-CPU	Assigned bytes in the F-CPU per F-I/O:							
	7	6	5	4	3	2	1	0
x + 0	Channel value AO_0							
...	...							
x + 10	Channel value AO_5							

x = Module start address

Table 6-8 Example: Address assignment in PII for F-I/O with 6 analog output channels (data type INT)

Byte in the F-CPU	Assigned bits in F-CPU per F-I/O:							
	7	6	5	4	3	2	1	0
x + 0	—	—	Value status AO_5	Value status AO_4	Value status AO_3	Value status AO_2	Value status AO_1	Value status AO_0

x = Module start address

The location of the channel values in the PIQ can be found in the device manual for the F-I/O.

6.3 Process Data or Fail-Safe Values

When are fail-safe values used?

The safety function requires that fail-safe values (0) be used instead of process data for passivation of the entire F-I/O or individual channels of an F-I/O in the following cases. This applies both to digital channels (data type BOOL) as well as for analog channels (data type INT or DINT):

- When the F-system starts up
- When errors occur during safety-related communication (communication errors) between the F-CPU and F-I/O using the safety protocol in accordance with PROFIsafe
- When F-I/O faults and channel faults occur (such as wire break, short circuit, and discrepancy errors)
- As long as you enable passivation of the F-I/O with PASS_ON = 1 in the F-I/O DB (see below)
- As long as you disable the F-I/O with DISABLE = 1 in the F-I/O DB (see below)

Fail-safe value output for F-I/O/channels of an F-I/O

When **passivation** occurs for an **F-I/O with inputs**, the F-system provides the safety program with fail-safe values (0) in the PII instead of the process data pending at the fail-safe inputs of the F-I/O.

The overflow or underflow of a channel of the **SM 336; AI 6 x 13Bit** or **SM 336; F-AI 6 x 0/4 ... 20 mA HART** is recognized by the F-system as an F-I/O/channel fault. The fail-safe value 0 is provided in place of $7FFF_H$ (for overflow) or 8000_H (for underflow) in the PII for the safety program.

If you want to process fail-safe values other than "0" in the safety program for an F-I/O with inputs **for analog channels of data type INT or DINT**, you can assign individual fail-safe values for QBAD = 1 and value status = 0 or QBAD_I_xx/QBAD_O_xx = 1 (instructions JMP/JMPN, LABEL and MOVE). For details about the characteristics go to QBAD/PASS_OUT/DISABLED/QBAD_I_xx/QBAD_O_xx and value status (Page 169).

WARNING

For an F-I/O with digital input channels (data type BOOL), the value provided in the PII must always be processed in the safety program, regardless of the value status or QBAD/QBAD_I_xx. *(S009)*

When **passivation** occurs in an **F-I/O with outputs**, the F-system outputs fail-safe values (0) at the fail-safe outputs instead of the output values provided by the safety program in the PIQ.

<table>
<tr><th>State of associated PAA/PIQ by ...</th><th>F-I/O with "RIOforFA-Safety" profile with S7-1200/1500 F-CPUs</th><th>F-I/O without "RIOforFA-Safety" profile with S7-1200/1500 F-CPUs</th><th>F-I/O with S7-300/400 F-CPUs</th></tr>
<tr><td>Startup of F-system[1]</td><td colspan="3" rowspan="2">The F-system overwrites the PII/PIQ with fail-safe values (0).</td></tr>
<tr><td>Communication errors</td></tr>
<tr><td>F-I/O faults</td><td rowspan="3">The F-system overwrites the PII with fail-safe values (0). In the PII the values formed in the safety program are retained.</td><td colspan="2" rowspan="2">The F-system overwrites the PII/PIQ with fail-safe values (0).</td></tr>
<tr><td>Channel faults in configuration of passivation for complete F-I/O</td></tr>
<tr><td>Channel faults during configuration of channel-granular passivation</td><td colspan="2">For the affected channels: The F-system overwrites the PII/PIQ with fail-safe values (0).</td></tr>
<tr><td>As long as passivation of the F-I/O is activated in the F-I/O DB with PASS_ON = 1</td><td colspan="3">The F-system overwrites the PII/PIQ with fail-safe values (0).</td></tr>
<tr><td>As long as the F-I/O is deactivated in the F-I/O DB with DISABLE = 1</td><td colspan="2">The F-system overwrites the PII/PIQ with fail-safe values (0).</td><td>-</td></tr>
</table>

[1] During the delayed transfer of the F-Base ID, the F-system remains in startup.

Reintegration of F-I/O/channels of an F-I/O

The switchover from fail-safe values (0) to process data (**reintegration of an F-I/O**) takes place **automatically** or following **user acknowledgment** in the F-I/O DB. The reintegration method depends on the following:

- The reason for passivation of the F-I/O or channels of the F-I/O
- At F-I/Os without the "Channel failure acknowledge" channel parameter on the value of the ACK_NEC variable of the associated F-IO data blocks (Page 164).
- At F-I/Os with the "Channel failure acknowledge" channel parameter (for example F-modules S7-1500 / ET 200 MP / F-modules SIMATIC S7-1200) on the value of the channel parameter.

For fail-safe GSD based DP slaves / GSD based I/O devices with "RIOforFA-Safety" profile, consult the respective documentation.

NOTE

Note that for channel faults in the F-I/O, channel-granular passivation takes place if configured accordingly in the *hardware and network editor*. For the concerned channels, fail-safe values (0) are output.

Reintegration after channel faults reintegrates all channels whose faults were eliminated; faulty channels remain passivated.

See also

Configuring F-I/O (Page 53)

6.4 F-I/O DB

Introduction

An F-I/O DB is automatically generated for each F-I/O (in safety mode) when the F-I/O is configured in the *hardware and network editor*. The F-I/O DB contains tags that you can evaluate or can/must write to in the safety program. It is not permitted to change the initial values of the tags directly in the F-I/O DB. When an F-I/O is deleted, the associated F-I/O DB is also deleted.

Access to an F-I/O DB

You access tags of the F-I/O DB for the following reasons:

- For reintegration of F-I/O after communication errors, F-I/O faults, or channel faults
- If you want to passivate the F-I/O depending on particular states of your safety program (for example, group passivation)
- If you want to deactivate the F-I/O (for example, configuration control)
- For changing parameters for fail-safe GSD based DP slaves/GSD based I/O devices
- If you want to evaluate whether fail-safe values or process data are output

6.4.1 Name and number of the F-I/O DB

The name of the F-I/O DB is formed by:

- the fixed prefix "F"
- the start address of the F-I/O, and the names entered in the properties of the F-I/O in the *hardware and network editor* or in the device view (max. the first 24 characters)

Example: F00004_F-DI24xDC24V_1

The number is assigned within the number range defined in the "Settings" area (Page 91) of the *Safety Administration Editor*.

Option "Creates F-I/O DBs without prefix" (S7-1200, S7-1500)

When you select the option "Creates F-I/O DBs without prefix" in the "Settings" (Page 91) area in the *Safety Administration Editor*, the name is formed only from:

- the name entered in the properties of the F-I/O in the *hardware and network editor* or in the device view (max. 117 characters)

Example: F-DI24xDC24V_1

Changing the name and number of the F-I/O DB

You change the name by changing the name entered in the properties of the F-I/O in the *hardware and network editor* or in the device view.

You change the number in the "Properties"/"F-parameters" tab of the associated F-I/O.

6.4.2 Tags of the F-I/O DB

The following table presents the variables of an F-I/O DB:

	Tag	Data type	Description	Initial value
Tags that you can or must write	PASS_ON	BOOL	1=enable passivation	0
	ACK_NEC	BOOL	1=acknowledgment for reintegration required in the event of F-I/O or channel faults	1
	ACK_REI	BOOL	1=acknowledgment for reintegration	0
	IPAR_EN	BOOL	Tag for parameter reassignment of fail-safe GSD based DP slaves/GSD based I/O devices or, in the case of SM 336; F-AI 6 x 0/4 ... 20 mA HART, for enabling HART communication	0
	DISABLE[1]	BOOL	1=Deactivate F-I/O	0
Tags that you can evaluate	PASS_OUT	BOOL	Passivation output	1
	QBAD	BOOL	1=Fail-safe values are output	1
	ACK_REQ	BOOL	1=Acknowledgment request for reintegration	0
	IPAR_OK	BOOL	Tag for parameter reassignment of fail-safe GSD based DP slaves/GSD based I/O devices or, in the case of SM 336; F-AI 6 x 0/4 ... 20 mA HART, for enabling HART communication	0
	DIAG	BYTE	Non-fail-safe service information	0
	DISABLED[1]	BOOL	1=F-I/O is deactivated	0
	QBAD_I_xx	BOOL	1=Fail-safe values are output to input channel xx (S7-300, S7-400)	1
	QBAD_O_xx	BOOL	1=Fail-safe values are output to output channel xx (S7-300, S7-400)	1
	BASEID_ACTIVE[2]	LWORD	Active F-BaseID Non-fail-safe information	0

1 As of Safety system version V2.1 for S7-1200/1500

2 As of Safety system version V2.8 for S7-1200 G2/1500

Differences in evaluation in S7-1200/1500 F-CPUs and S7-300/400

The following table describes the differences in the evaluation of tags of the F-I/O DB and the value status depending on the F-I/O and F-CPU used.

Tag in the F-I/O DB or value status	F-I/O with "RIOforFA-Safety" profile with S7-1200/1500 F-CPU	F-I/O without "RIOforFA-Safety" profile with S7-1200/1500 F-CPU	F-I/O with S7-300/400 F-CPU
ACK_NEC	—[2]	x	x
QBAD[3]	x	x	x
PASS_OUT[3]	x	x	x
QBAD_I_xx[1]	—	—	x
QBAD_O_xx[1]	—	—	x
Wertstatus[1]	x	x	—

[1] QBAD_I_xx and QBAD_O_xx display the validity of the channel value channel-granular and thus correspond to the inverted value status with S7-1200/1500. Value status or QBAD_I_xx and QBAD_O_xx are not available with fail-safe DP GSD based slaves or fail-safe GSD based I/O devices without the "RIOforFA-Safety" profile.

[2] In the case of F-I/Os that support the "Channel failure acknowledge" channel parameter (for example S7-1500/ET 200MP F-modules or S7-1200 F-modules), this replaces the ACK_NEC variable of the F-IO data block.

[3] For details about the characteristics, see "QBAD/PASS_OUT/DISABLED/QBAD_I_xx/QBAD_O_xx and value status"

6.4.2.1 PASS_ON

The PASS_ON tag allows you to enable passivation of an F-I/O, for example, depending on particular states in your safety program.

Using the PASS_ON tag in the F-I/O DB, you can passivate F-I/O; channel-granular passivation is not possible.

As long as PASS_ON = 1, **passivation** of the associated F-I/O occurs.

6.4.2.2 ACK_NEC

If an F-I/O fault is detected by the F-I/O, **passivation** of the relevant F-I/O occurs. If channel faults are detected and channel granular passivation is configured, the relevant channels are passivated. If passivation of the entire F-I/O is configured, all channels of the relevant F-I/O are passivated. Once the F-I/O fault or channel fault has been eliminated, **reintegration** of the relevant F-I/O occurs, depending on ACK_NEC:

- With ACK_NEC = 0, you can assign an **automatic reintegration**.
- With ACK_NEC = 1, you can assign a **reintegration** by **user acknowledgment**.

WARNING

The parameter assignment of the ACK_NEC tag of the F-I/O DB with the value "0" is only allowed if automatic reintegration is permitted for the relevant process from a safety standpoint. *(S010)*

NOTE

The initial value for ACK_NEC is 1 after creation of the F-I/O DB. If you do not require automatic reintegration, you do not have to write ACK_NEC.

See also

After F-I/O or channel faults (Page 177)

6.4.2.3 ACK_REI

When the F-system detects a communication error or an F-I/O fault for an F-I/O, the relevant F-I/O is passivated. If channel faults are detected and channel granular passivation is configured, the relevant channels are passivated. If passivation of the entire F-I/O is configured, all channels of the relevant F-I/O are passivated. **Reintegration** of the F-I/O/channels of the F-I/O after the fault has been eliminated requires a **user acknowledgment** with a positive edge at variable ACK_REI of the F-I/O DB:

- After every communication error
- After F-I/O or channel faults only with parameter assignment "Channel failure acknowledgement = manual" or ACK_NEC = 1

Reintegration after channel faults reintegrates all channels whose faults were eliminated.

Acknowledgment is not possible until tag ACK_REQ = 1.

In your safety program, you must provide a user acknowledgment by means of the ACK_REI tag for each F-I/O.

WARNING

A user acknowledgment (Page 190) must always take place with a signal that was generated by manual operation. Implementation with an automatically generated signal is not permitted. *(S011)*

NOTE

Alternatively, you can use the "ACK_GL" instruction to carry out reintegration of the F-I/O following communication errors or F-I/O/channel faults (ACK_GL: Global acknowledgment of all F-I/O in an F-runtime group (STEP 7 Safety V21) (Page 530)).

6.4.2.4 IPAR_EN

The IPAR_EN tag corresponds to the iPar_EN_C tag in the PROFIsafe bus profile as of PROFIsafe Specification V1.20 and higher.

Fail-safe GSD based DP slaves/GSD based I/O devices

To find out when this tag must be set or reset when parameters of fail-safe GSD based DP slaves/GSD based I/O devices are reassigned, consult the PROFIsafe Specification V1.20 or higher or the documentation for the fail-safe GSD based DP slave/GSD based I/O device.

Note that IPAR_EN = 1 does not trigger passivation of the relevant F-I/O.

If passivation is to occur when IPAR_EN = 1, you must also set tag PASS_ON = 1.

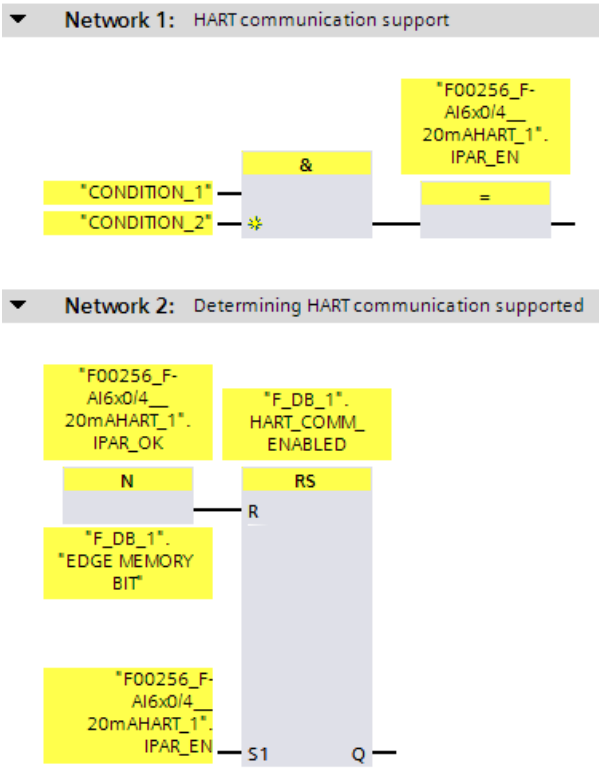
HART communication with SM 336; F-AI 6 x 0/4 ... 20 mA HART

If you set the IPAR_EN tag to "1" while parameter "HART_Tor" = "switchable" is assigned, the HART communication for the SM 336; F-AI 6 x 0/4 ... 20 mA HART is enabled. Setting this tag to "0" disables the HART communication. The F-SM acknowledges the enabled or disabled HART communication with tag IPAR_OK = 1 or 0.

Enable HART communication only when your system is in a status, in which any reassignment of parameters for an associated HART device can be done without any risk.

If you want to evaluate the "HART communication enabled" status in your safety program, e.g., for the purpose of programming interlocks, you must build up the information as shown in the following example. This is necessary to ensure that the information is properly available even if communication errors occur while the HART communication is enabled with IPAR_EN = 1. Only change the status of the IPAR_EN tag during this evaluation if there is no passivation due to a communication error or F-I/O or channel fault.

Example of enabling HART communication



Additional information on HART communication with SM 336; F-AI 6 x 0/4 ... 20 mA HART can be found in the Automation System S7-300, ET 200M Distributed I/O System manual, Fail-Safe Signal Modules (https://support.industry.siemens.com/cs/ww/en/view/19026151) manual and in the help on the F-module.

6.4.2.5 DISABLE

The DISABLE variable allows you to deactivate an F-I/O.

As long as DISABLE = 1, the associated F-I/Os are **passivated**.

Diagnostics entries of the safety program may no longer be entered in the diagnostics buffer of the F-CPU for this F-I/O (for example, due to communication error).

Existing diagnostics entries are marked as outgoing.

6.4.2.6 QBAD/PASS_OUT/DISABLED/QBAD_I_xx/QBAD_O_xx and value status

The following table set outs differences in the reaction of the channel values and QBAD, PASS_OUT, DISABLED, QBAD_I_xx/QBAD_O_xx variables and of the value status depending on the F-I/O and F-CPU used.

Fail-safe value output after...	**F-I/O with "RIOforFA-Safety" profile with S7-1200/1500 F-CPU**	**F-I/O without profile "RIOforFA-Safety" with S7-1200/1500 F-CPUs**	**F-I/O with S7-300/400 F-CPU**
Startup of F-system[2]	QBAD and PASS_OUT = 1 DISABLED **unchanged** For **all** channels: Channel value = fail-safe value (0) Value status = 0[1]		QBAD and PASS_OUT = 1 For **all** channels: Channel value = fail-safe value (0) QBAD_I_xx and QBAD_O_xx = 1[1]
Communication errors			
F-I/O faults			
Channel faults in configuration of passivation for complete F-I/O			

1 Value status or QBAD_I_xx and QBAD_O_xx are not available for fail-safe GSD based DP slaves and fail-safe GSD based I/O devices without the "RIOforFA-Safety" profile

2 During delayed transfer of the F-Base ID, the F-system remains in startup.

Fail-safe value output after...	F-I/O with "RIOforFA-Safety" profile with S7-1200/1500 F-CPU	F-I/O without profile "RIOforFA-Safety" with S7-1200/1500 F-CPUs	F-I/O with S7-300/400 F-CPU
Channel faults during configuration of channel-granular passivation	QBAD, PASS_OUT and DISABLED **unchanged** For the **affected** channels: Channel value = fail-safe value (0) Value status = 0	QBAD and PASS_OUT = 1 DISABLED **unchanged** For the **affected** channels: Channel value = fail-safe value (0) Value status = 0[1]	QBAD and PASS_OUT = 1 For the **affected** channels: Channel value = fail-safe value (0) QBAD_I_xx and QBAD_O_xx = 1[1]
As long as passivation of the F-I/O is activated in the F-I/O DB with PASS_ON = 1	QBAD = 1, PASS_OUT and DISABLED **unchanged** For **all** channels: Channel value = fail-safe value (0) Value status = 0[1]		QBAD = 1, PASS_OUT **unchanged** For **all** channels: Channel value = fail-safe value (0) QBAD_I_xx and QBAD_O_xx = 1[1]
As long as the F-I/O is deactivated in the F-I/O DB with DISABLE = 1	QBAD, PASS_OUT and DISABLED = 1 For **all** channels: Channel value = fail-safe value (0) Value status = 0[1]		-

1 Value status or QBAD_I_xx and QBAD_O_xx are not available for fail-safe GSD based DP slaves and fail-safe GSD based I/O devices without the "RIOforFA-Safety" profile

2 During delayed transfer of the F-Base ID, the F-system remains in startup.

6.4.2.7 ACK_REQ

When the F-system detects a communication error or an F-I/O fault or channel fault for an F-I/O, the relevant F-I/O or individual channels of the F-I/O are passivated. ACK_REQ = 1 signals that **user acknowledgment** is required for reintegration of the relevant F-I/O or channels of the F-I/O.

The F-system sets ACK_REQ = 1 as soon as the fault has been eliminated and user acknowledgment is possible. For channel-granular passivation, the F-system sets ACK_REQ = 1 as soon as the channel fault is corrected. User acknowledgment is possible for this fault. Once acknowledgment has occurred, the F-system resets ACK_REQ to 0.

NOTE

For F-I/O with outputs, acknowledgment after F-I/O or channel faults may only be possible some minutes after the fault has been eliminated, until the necessary test signal is applied (see *F-I/O manuals*).

6.4.2.8 IPAR_OK

The IPAR_OK tag corresponds to the iPar_OK_S tag in the PROFIsafe bus profile, PROFIsafe Specification V1.20 and higher.

Fail-safe GSD based DP slaves/GSD based I/O devices

To find out how to evaluate this tag when parameters of fail-safe GSD based DP slaves or GSD based I/O devices are reassigned, consult the PROFIsafe Specification V1.20 or higher or the documentation for the fail-safe GSD based DP slave/GSD based I/O device.

For HART communication with SM 336; F-AI 6 x 0/4 ... 20 mA HART see Chapter IPAR_EN (Page 168).

6.4.2.9 DIAG

The DIAG tag provides non-fail-safe information (1 byte) about errors or faults that have occurred for service purposes. You can read out this information by means of operator control and monitoring systems or, if applicable, you can evaluate it in your standard user program. DIAG bits are saved until you perform an acknowledgment with the ACK_REI tag or until automatic reintegration takes place.

Structure of DIAG

Bit no.	Assignment	Possible error causes	Remedies
Bit 0	Timeout detected by F-I/O	The PROFIBUS/PROFINET connection between F-CPU and F-I/O is faulty. The value of the F-monitoring time for the F-I/O is set too low. The F-I/O are receiving invalid parameter assignment data or	• Check the PROFIBUS/PROFINET connection and ensure that there are no external sources of interference. • Check the parameter assignment of the F-I/O. If necessary, set a higher value for the monitoring time. Recompile the hardware configuration, and download it to the F-CPU. Recompile the safety program. • Check the diagnostics buffer of the F-I/O. • Turn the power of the F-I/O off and back on.
		Internal F-I/O fault or	Replace F-I/O
		Internal F-CPU fault	Replace F-CPU
Bit 1	F-I/O fault or channel fault detected by F-I/O[1]	See *F-I/O manuals*	See *F-I/O manuals*
Bit 2	CRC error or sequence number error detected by F-I/O	See description for bit 0	See description for bit 0
Bit 3	Reserved	—	—
Bit 4	Timeout detected by F-system	See description for bit 0	See description for bit 0
Bit 5	Sequence number error detected by F-system[2]	See description for bit 0	See description for bit 0
Bit 6	CRC-error detected by F-system	See description for bit 0	See description for bit 0
Bit 7	Addressing error[3]	—	Contact Service & Support

[1] Not for F-I/O that support the "RIOforFA-Safety" profile.

[2] For S7-300/400 F-CPUs only

[3] For S7-1200/1500 F-CPUs only

6.4.2.10 BASEID_ACTIVE

The BASEID_ACTIVE tag contains the active F-BaseID as non-fail-safe information. The tag is available as of Safety System Version V2.8 and is only relevant if both the F-CPU and the F-I/O support PROFIsafe address type 3.

6.4.3 Accessing tags of the F-I/O DB

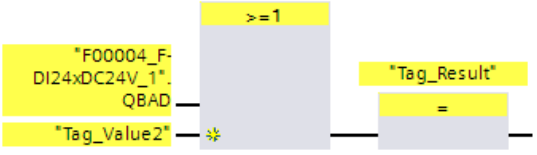
Rule for accessing tags of the F-I/O DB

Tags of the F-I/O DB of an F-I/O can only be accessed from the F-runtime group from which the channels of this F-I/O are accessed (if access is made).

"Fully qualified DB access"

You can access the tags of the F-I/O DB via a "fully qualified DB access" (that is, by specifying the name of the F-I/O DB and by specifying the name of the tag).

Example of evaluating the QBAD tag



See also

F-I/O DB (Page 164)

6.5 Passivation and reintegration of F-I/O

Overview

In the following you can find information on passivation and reintegration of F-I/O.

Signal sequence charts

The signal sequences presented below represent typical signal sequences for the indicated behavior.

Actual signal sequences and, in particular, the relative position of the status change of individual signals can deviate from the given signal sequences within the scope of known "fuzzy" cyclic program execution factors, depending on the following:

- The F-I/O used
- The F-CPU used
- The cycle time of the (F-)OB in which the associated F-runtime group is called
- The target rotation time of the PROFIBUS DP or the update time of the PROFINET IO

NOTE

The signal sequences shown refer to the status of signals in the user's safety program.

6.5.1 After startup of F-system

Behavior after a startup

Fail-safe value output after startup of the F-system	F-I/O with "RIOforFA-Safety" profile with S7-1200/1500 F-CPU	F-I/O without profile "RIOforFA-Safety" with S7-1200/1500 F-CPU	Every F-I/O with S7-300/400 F-CPU
Passivation of the entire F-I/O occurs during startup.	QBAD and PASS_OUT = 1 For **all** channels: Channel value = fail-safe value (0) Value status = 0*		QBAD and PASS_OUT = 1 For **all** channels: Channel value = fail-safe value (0) QBAD_I_xx and QBAD_O_xx = 1*

* Value status or QBAD_I_xx and QBAD_O_xx are not available for fail-safe GSD based DP slaves and fail-safe GSD based I/O devices without the "RIOforFA-Safety" profile.

Reintegration of F-I/O

Reintegration of the F-I/O, i.e. the provision of process values in the PII or the output of process values provided in the PIQ at the fail-safe outputs, takes place **automatically**, regardless of the setting at the ACK_NEC tag or the configuration "Channel failure acknowledge", no sooner than the second cycle of the F-runtime group after F-system startup.

You can find more information on pending F-communication, F-I/O or channel errors during startup of the F-system in the sections After communication errors (Page 175) and After F-I/O or channel faults (Page 177).

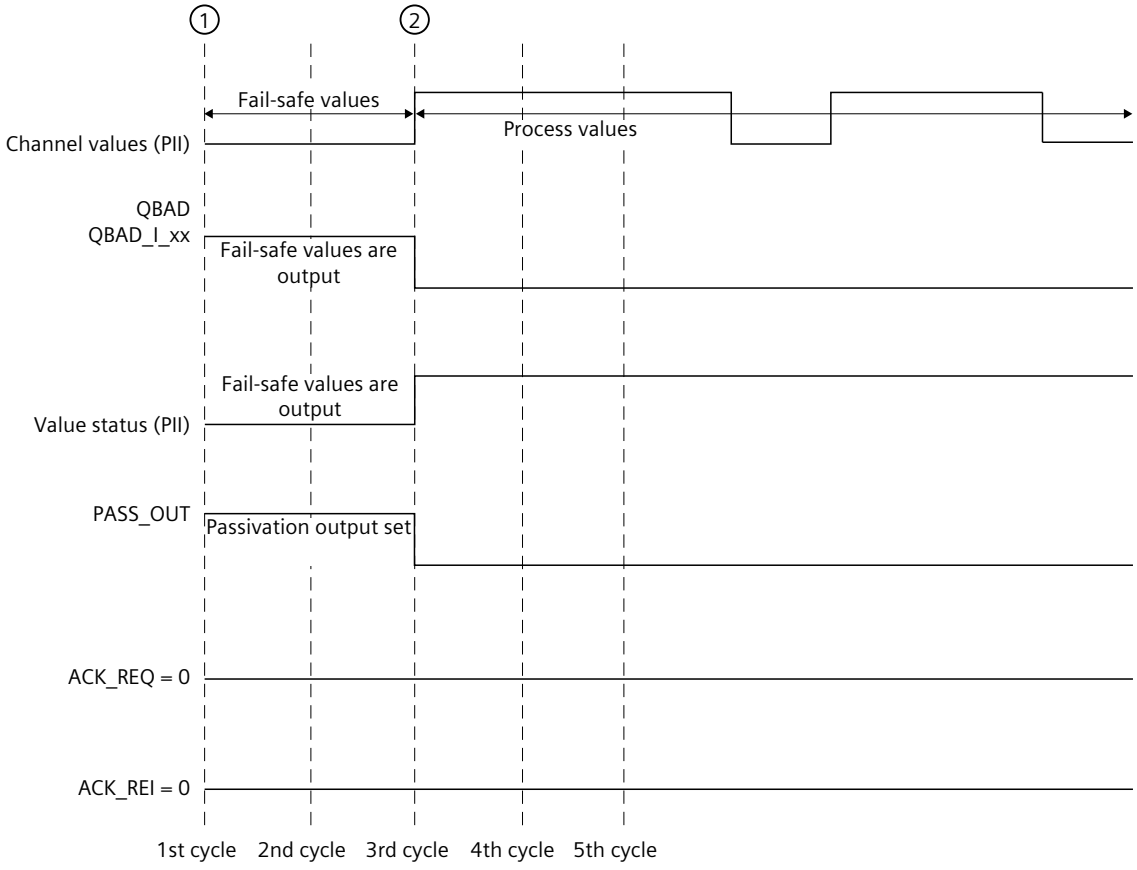
For fail-safe GSD based DP slaves/GSD based I/O devices with "RIOforFA-Safety" profile, consult the respective documentation for the fail-safe GSD based DP slave/GSD based I/O device.

Depending on the F-I/O you are using and the cycle time of the F-runtime group and PROFIBUS DP/PROFINET IO, several cycles of the F-runtime group can elapse before reintegration occurs.

If communication between the F-CPU and F-I/O takes longer to establish than the F-monitoring time set in the properties for the F-I/O, automatic reintegration does not take place.

Signal sequence for passivation and reintegration of F-I/O after F-system startup

Example for an F-I/O with inputs:



① Startup of F-system/passivation

② Automatic reintegration (e.g. third cycle)

⚠ WARNING

Unintentional restart

In contrast to the standard user program, when the safety program starts up, all tags of the F-DBs are generally initialized with their start values. This means that saved error information is lost. The F-system performs automatic reintegration of the F-I/O.

If an unintentional startup of your process is not allowed, you must program restart/startup protection in the safety program. For this, the output of process values must be blocked until user acknowledgment takes place (see "Implementation of user acknowledgment (Page 190)"). This user acknowledgment must not take place until safe output of process values is possible (see "Programming startup protection (Page 156)").

You must expect an unintentional restart in the following cases:

- After a STOP generated by PG/PC operation, mode switch, communication function or "STP" instruction
- After an operating error
- After a STOP caused by a fault reaction function
- The following applies to the S7-1500 F Virtual Controller:

  After a STOP or a restart of the S7-1500 F Virtual Controller via Industrial Edge Management.

(S008)

6.5.2 After communication errors

Behavior after communication errors

Fail-safe value output after communication errors	F-I/O with "RIOforFA-Safety" profile with S7-1200/1500 F-CPU	F-I/O without profile "RIOforFA-Safety" with S7-1200/1500 F-CPU	Every F-I/O with S7-300/400 F-CPU
If a communication error between the F-CPU and the F-I/O is detected, all channels of the entire F-I/O are passivated.	QBAD and PASS_OUT = 1 For **all** channels: Channel value = fail-safe value (0) Value status = 0*		QBAD and PASS_OUT = 1 For **all** channels: Channel value = fail-safe value (0) QBAD_I_xx and QBAD_O_xx = 1*

* Value status or QBAD_I_xx and QBAD_O_xx are not available for fail-safe GSD based DP slaves and fail-safe GSD based I/O devices without the "RIOforFA-Safety" profile.

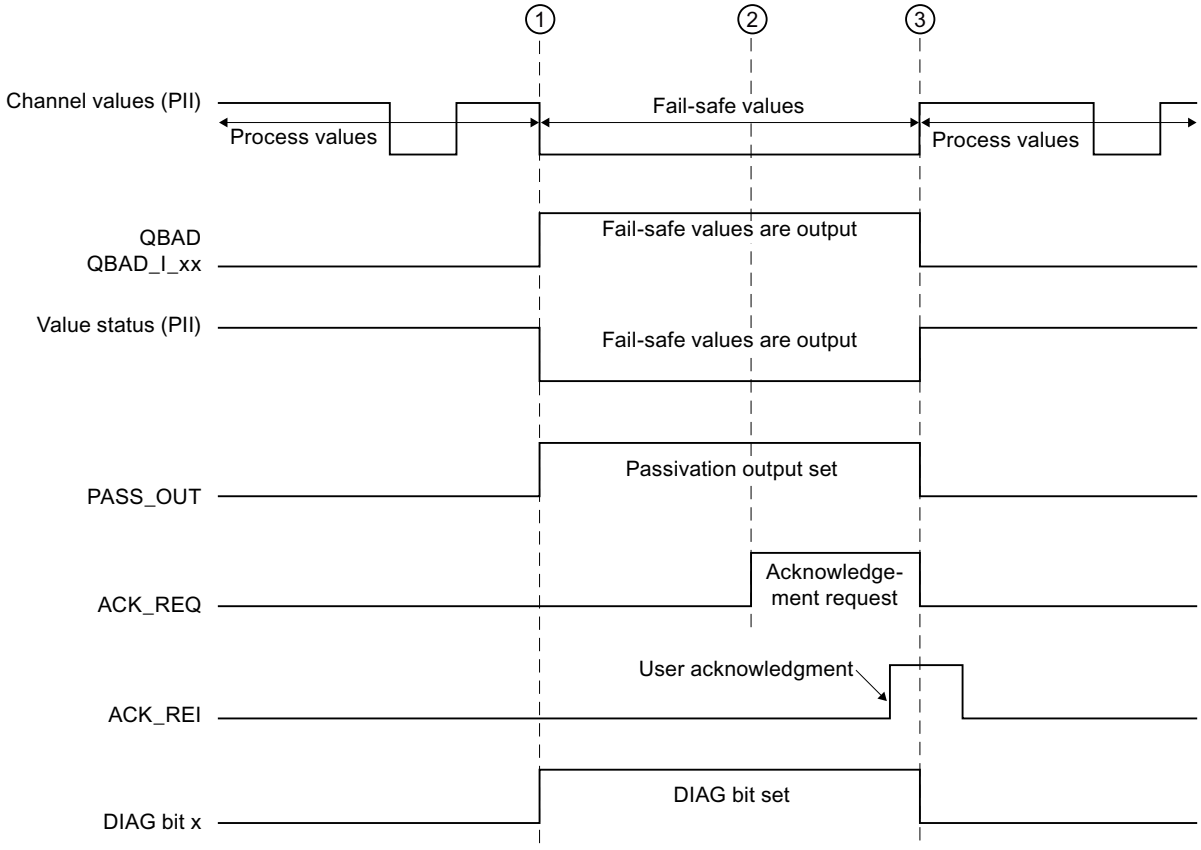
Reintegration of F-I/O

Reintegration of the relevant F-I/O, that is, provision of process values in the PII or output of process data provided in the PIQ at the fail-safe outputs, takes place only when the following occurs:

- All communication errors have been eliminated and the F-system has set tag ACK_REQ = 1
- A **user acknowledgment** with a positive edge has occurred:
  - At the ACK_REI tag of the F-I/O DB (Page 167) or
  - At the ACK_REI_GLOB input of the "ACK_GL" instruction (ACK_GL: Global acknowledgment of all F-I/O in an F-runtime group (STEP 7 Safety V21) (Page 530))

Signal sequence for passivation and reintegration of F-I/O after communication errors

Example for an F-I/O with inputs:



① Communication error/passivation
② All communication errors have been eliminated
③ Reintegration

See also

Implementing User Acknowledgment in the Safety Program of the F-CPU of a DP Master or IO controller (Page 190)

Implementing user acknowledgment in the safety program of the F-CPU of a I-slave or I-device (Page 194)

6.5.3 After F-I/O or channel faults

Behavior after F-I/O faults

Fail-safe value output after F-I/O faults	**F-I/O with "RIOforFA-Safety" profile with S7-1200/1500 F-CPU**	**F-I/O without profile "RIOforFA-Safety" with S7-1200/1500 F-CPU**	**Every F-I/O with S7-300/400 F-CPU**
If an F-I/O fault is detected by the F-system, passivation of all channels of the entire F-I/O occurs.	QBAD and PASS_OUT = 1 For **all** channels: Channel value = fail-safe value (0) Value status = 0*		QBAD and PASS_OUT = 1 For **all** channels: Channel value = fail-safe value (0) QBAD_I_xx and QBAD_O_xx = 1*

* Value status or QBAD_I_xx and QBAD_O_xx are not available for fail-safe GSD based DP slaves and fail-safe GSD based I/O devices without the "RIOforFA-Safety" profile.

Behavior after channel fault

Fail-safe value output after channel faults	**F-I/O with "RIOforFA-Safety" profile with S7-1200/1500 F-CPU**	**F-I/O without profile "RIOforFA-Safety" with S7-1200/1500 F-CPU**	**Every F-I/O with S7-300/400 F-CPU**
When passivation for **complete** F-I/O is configured: If a channel fault is detected by the F-system, passivation of all channels of the entire F-I/O occurs.	QBAD and PASS_OUT = 1 For **all** channels: Channel value = fail-safe value (0) Value status = 0*		QBAD and PASS_OUT = 1 For **all** channels: Channel value = fail-safe value (0) QBAD_I_xx and QBAD_O_xx = 1*
With configuration of **channel-granular** passivation: If a channel fault is detected by the F-system, passivation of all the affected channels of the entire F-I/O occurs.	QBAD and PASS_OUT **unchanged** For the **affected** channels: Channel value = fail-safe value (0) Value status = 0	QBAD and PASS_OUT = 1 For the **affected** channels: Channel value = fail-safe value (0) Value status = 0*	QBAD and PASS_OUT = 1 For the **affected** channels: Channel value = fail-safe value (0) QBAD_I_xx and QBAD_O_xx = 1*

* Value status or QBAD_I_xx and QBAD_O_xx are not available for fail-safe GSD based DP slaves and fail-safe GSD based I/O devices without the "RIOforFA-Safety" profile.

Reintegration of F-I/O

Reintegration of the relevant F-I/O or the relevant channels of the F-I/O, that is, provision of process data in the PII or output of process data provided in the PIQ at the failsafe outputs, takes place only when the following occurs:

- All F-I/O or channel faults have been eliminated.

If you have configured channel-granular passivation for the F-I/O, the relevant channels are reintegrated once the fault is corrected; any faulty channels remain passivated.

Reintegration is performed depending on your setting for the ACK_NEC tag or the "Channel failure acknowledge" parameter (configuration of the S7-1500/ET 200MP F-module and S7-1200 F-module)

- With ACK_NEC = 0 or the configuration "Channel failure acknowledge = automatic", **automatic reintegration** is performed as soon as the F-system detects that the fault has been corrected. For F-I/O with inputs, reintegration takes place right away. For F-I/O with outputs or F-I/O with inputs and outputs, depending on the F-I/O you are using, reintegration can take several minutes, first after the necessary test signals have been applied, which are used by the F-I/O to determine that the fault has been eliminated.
- With ACK_NEC = 1 or the configuration "Channel failure acknowledge = manual", reintegration is performed only as a result of user acknowledgment with a positive edge at the ACK_REI tag of the F-I/O DB or at the ACK_REI_GLOB input of the "ACK_GL" instruction. Acknowledgment can be made as soon as the F-system detects that the fault has been eliminated and tag ACK_REQ = 1 has been set.

  For fail-safe GSD based I/O devices with "RIOforFA-Safety" profile, consult the respective documentation for the fail-safe GSD based I/O device .

⚠ WARNING

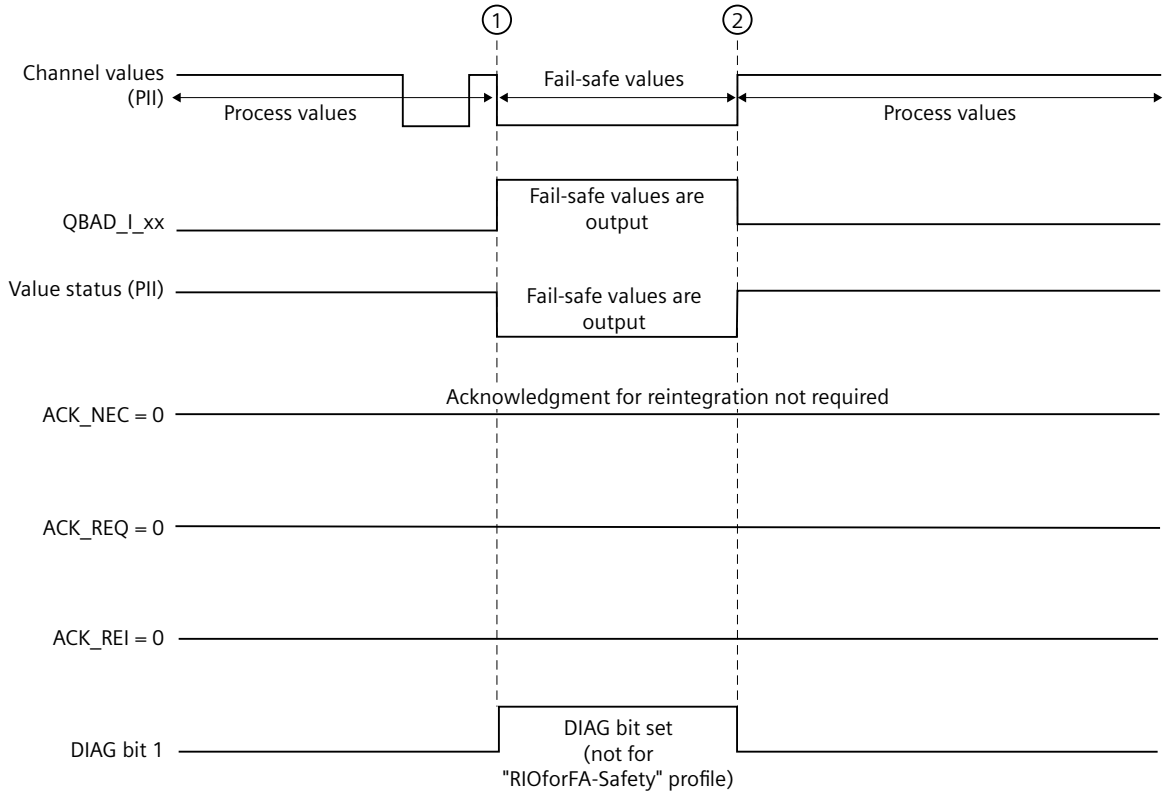
Following a supply voltage failure of the F-I/O lasting less than the F-monitoring time assigned for the F-I/O, automatic reintegration can occur regardless of the setting for the ACK_NEC tag or the "Channel failure acknowledge" configuration.

If automatic reintegration is not permitted for the relevant process in this case, you must program startup protection by evaluating tags QBAD or QBAD_I_xx and QBAD_O_xx or value status or PASS_OUT.

In the event of a supply voltage failure of the F-I/O lasting longer than the specified F-monitoring time for the F-I/O, the F-system detects a communication error. *(S012)*

Signal sequence for passivation and reintegration of F-I/O after F-I/O faults or channel faults when ACK_NEC = 0 or configuration "Channel failure acknowledge = automatic" (for passivation of entire F-I/O after channel faults)

Example for an F-I/O with inputs:



① F-I/O or channel faults
Passivation

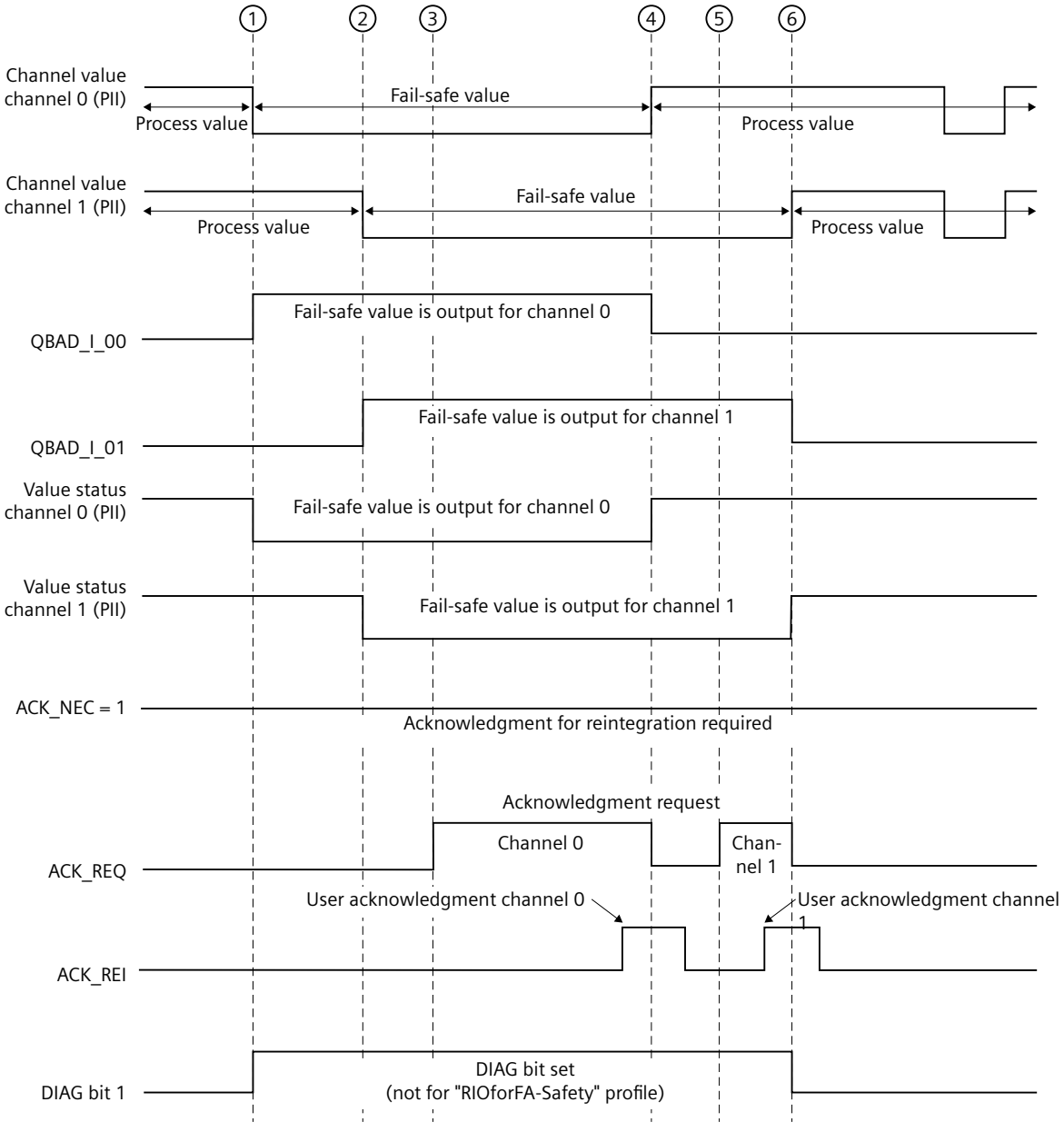
② F-I/O or channel faults corrected
Automatic reintegration

Signal sequence for passivation and reintegration of F-I/O after F-I/O faults or channel faults when ACK_NEC = 1 or configuration "Channel failure acknowledge = manual" (for passivation of entire F-I/O after channel faults)

For the signal sequence for passivation and reintegration of F-I/O after F-I/O or channel faults when ACK_NEC = 1 or configuration "Channel failure acknowledge = manual" (initial value), see After communication errors (Page 175).

Signal sequence for passivation and reintegration of F-I/O after channel faults when ACK_NEC = 1 or configuration "Channel failure acknowledge = manual" (for channel-granular passivation)

Example for an F-I/O with inputs:



① Channel fault for channel 0/passivation of channel 0
② Channel fault for channel 1/passivation of channel 1
③ Channel fault eliminated for channel 0
④ Reintegration of channel 0
⑤ Channel fault eliminated for channel 1
⑥ Reintegration of channel 1

6.5.4 Group passivation

Programming a group passivation

If you want to enable passivation of additional F-I/O when an F-I/O or a channel of an F-I/O is passivated by the F-system, you can use the PASS_OUT/PASS_ON tags to perform **group passivation** of the associated F-I/O.

Group passivation by means of PASS_OUT/PASS_ON can, for example, be used to force simultaneous reintegration of all F-I/O after startup of the F-system.

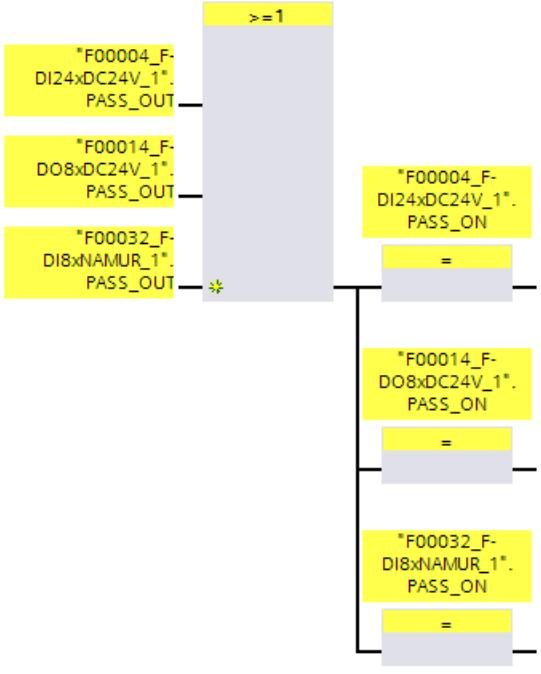
For group passivation, you must OR all PASS_OUT tags of the F-I/O in the group and assign the result to all PASS_ON tags of the F-I/O in the group.

During use of fail-safe values (0) due to group passivation by means of PASS_ON = 1, the QBAD tag of the F-I/O of this group = 1.

NOTE

Note the different behavior of PASS_OUT for F-I/O with/without "RIOforFA-Safety" profile (see table in section QBAD/PASS_OUT/DISABLED/QBAD_I_xx/QBAD_O_xx and value status (Page 169)).

Example of group passivation

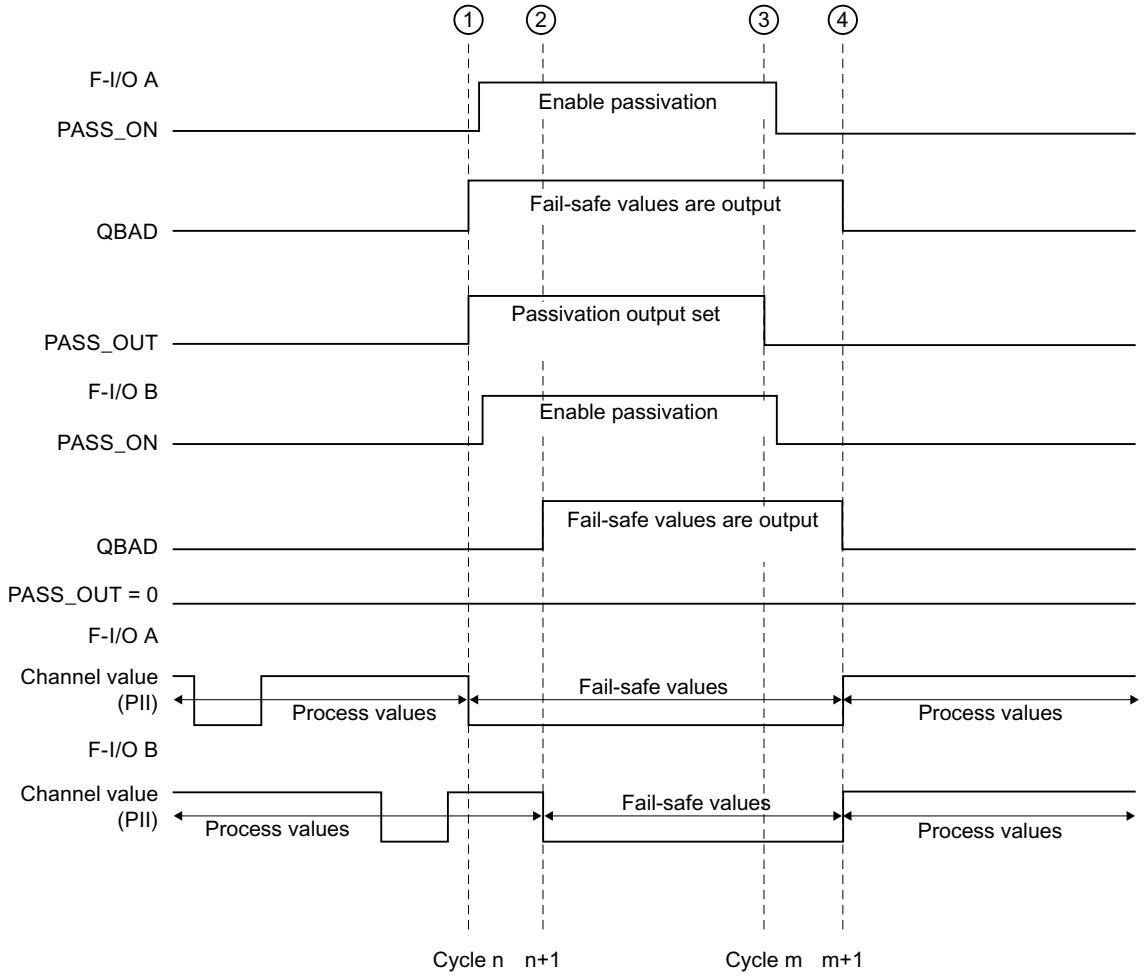


Reintegration of F-I/O

Reintegration of F-I/O passivated by group passivation occurs **automatically**, if a reintegration (**automatic** or **through user acknowledgment**) occurs for the F-I/O that triggered the group passivation (PASS_OUT = 0).

Signal sequence for group passivation following communication error

Example for two F-I/O with inputs:



① Communication error in F-I/O A
Passivation of F-I/O A

② Passivation of F-I/O B

③ Communication error in F-I/O A corrected and acknowledged

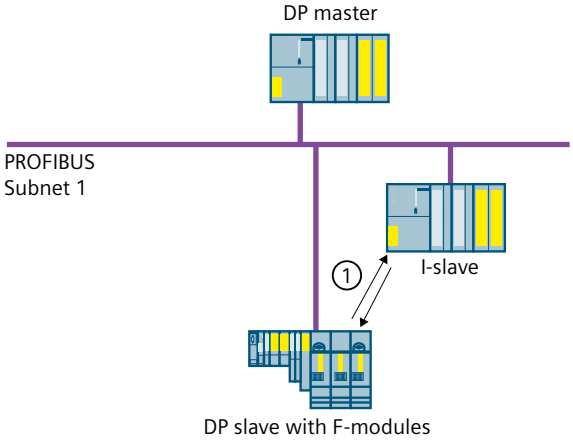
④ Reintegration F-I/O A and B

6.6 F-I/O access via I-slave-slave communication (S7-300, S7-400)

6.6.1 Overview

Principle of F-I/O access via I-slave-slave communication

The following figure shows the principle of F-I/O access via I-slave-slave communication.



① F-I/O access via I-slave-slave communication

F-I/O access via I-slave-slave communication

F-I/O access via I-slave-slave communication is possible with F-I/O in a DP slave that supports safety-related I-slave-slave communication, for example with all ET 200SP F-modules with IM 155-6 DP HF, firmware > V3.1, with all ET 200S F-modules with IM 151-1 HF, with all fail-safe S7-300 signal modules with IM 153-2, as of article number 6ES7153-2BA01-0XB0, firmware > V4.0.0.

Safety-related communication between the safety program of the F-CPU of an I-slave and F-I/O of a DP slave takes place using direct data exchange, as in the standard program. The process image is used to access the channels of the F-I/O in the safety program of the F-CPU of the I-slave.

6.6.2 Configuring Safety-Related I-Slave-Slave Communication

Introduction

Safety-related communication between the safety program of the F-CPU of an I-slave and F-I/O in a DP slave takes place using direct data exchange (**F**-DX-Mod), same as in standard programs.

You do not need any additional hardware for I-slave-slave communication.

I-slave-slave communication is also possible:

- when the assigned DP master is a standard CPU, if the standard CPU supports direct data exchange
- when instead of a DP master, an IO controller is networked with the I-slaves via an IE/PB link

An F-I/O DB is automatically generated for each F-I/O when it is configured in the *hardware and network editor*; this is required for the F-I/O access via safety-related I-slave-slave communication. The F-I/O DB is initially created in the safety program of the DP master, provided it is an F-CPU with F-activation. Only with the setup of the F-DX-Mod connection is the F-I/O DB created in the safety program of the I-slave and deleted in the safety program of the DP master.

The process image input is used to access the channels of the F-I/O in the safety program of the F-CPU of the I-slave (see description in Safety-related I-slave-slave communication - F-I/O access via process image (Page 189)).

Restrictions

NOTE

Safety-related I-slave-slave communication with F-I/O is possible in a DP slave that supports safety-related I-slave-slave communication, for example with all ET 200SP F-modules with IM 155-6 DP HF, firmware > V3.1, with all ET 200S F-modules with IM 151-1 HF, with all fail-safe S7-300 signal modules with IM 153-2, as of article number 6ES7153-2BA01-0XB0, firmware > V4.0.0.

NOTE

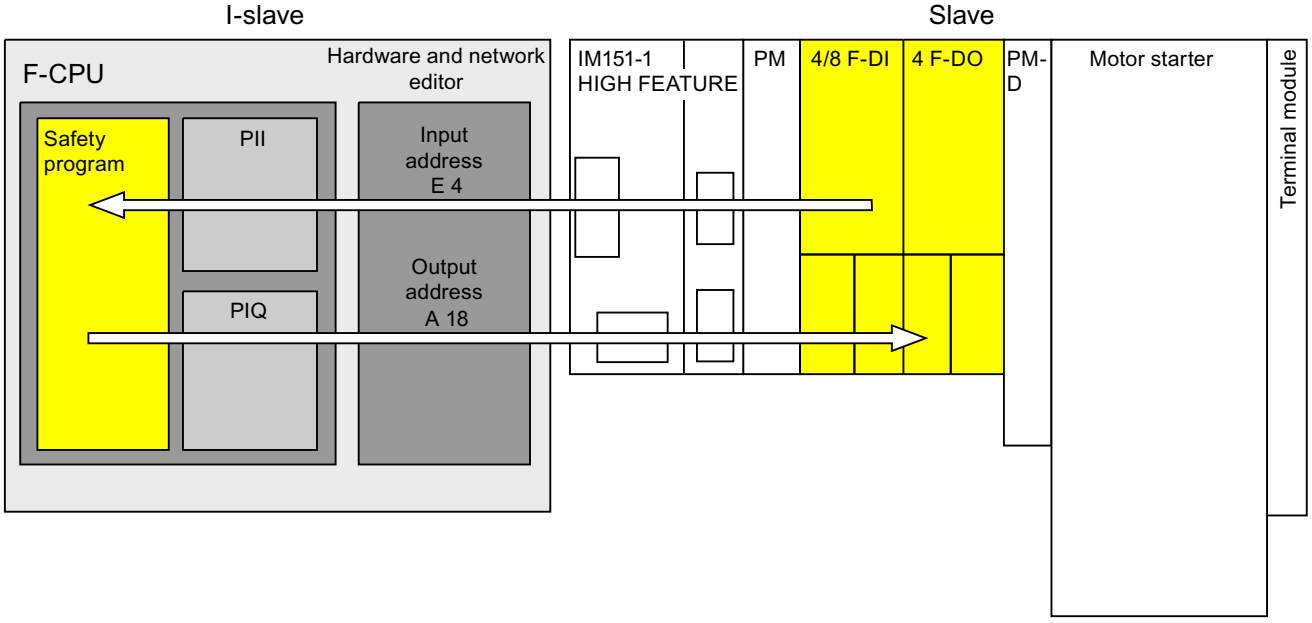
With safety-related I-slave-slave communication, make sure that the CPU of the DP master is powered up before the F-CPU of the I-slave.

Otherwise, depending on the F-monitoring time specified for the F-I/O, the F-system can detect an error in safety-related communication (communication error) between the F-CPU and the F-I/O assigned to the I-slave. This means that the F-I/O are not reintegrated automatically after F-system startup. They are instead only reintegrated after a user acknowledgment with a positive edge in the ACK_REI tag of the F-I/O DB (see also After communication errors (Page 175) and After startup of F-system (Page 173)).

Configuring transfer areas

For every safety-related communication connection between an I-slave and slave, you must configure transfer areas in the *hardware and network editor*.

The transfer area is assigned a label when it is created to identify it as the communication relationship. For example, "F-DX-Mod_PLC_2-PLC_1_1" for the first F-DX-Mod connection between F-CPU 1 and F-CPU 2.



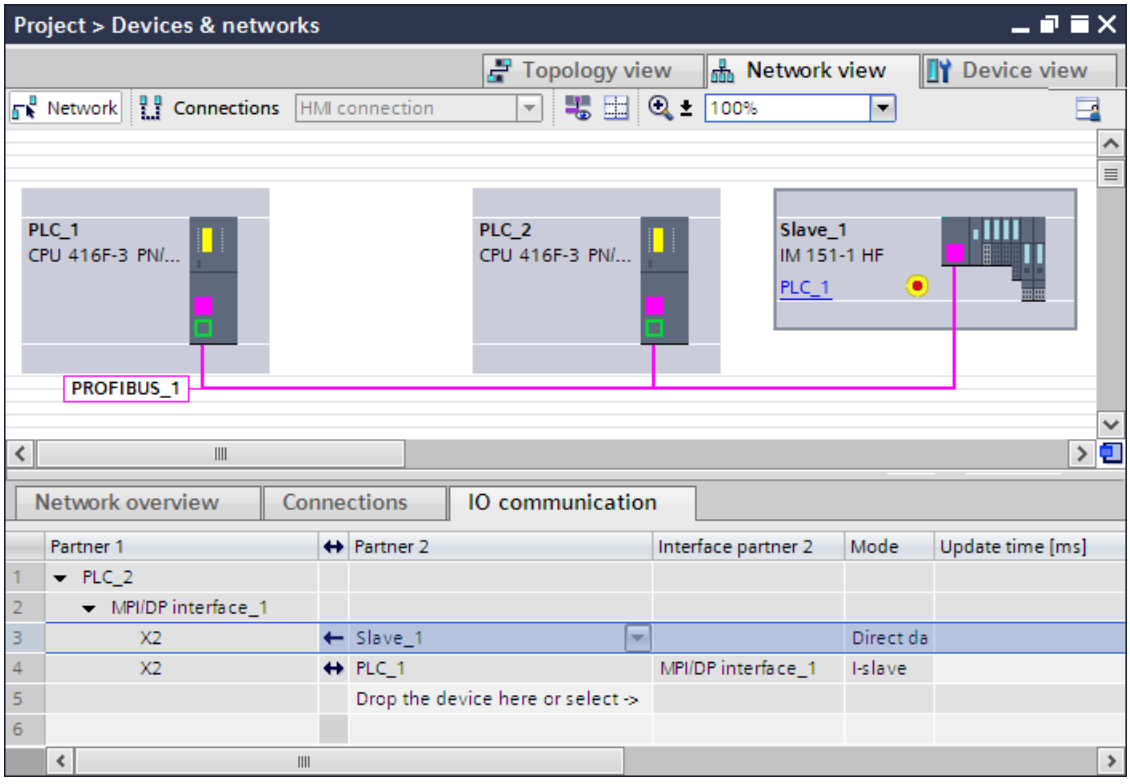
Configuration procedure using the example of an ET 200S with F-modules in the slave

The procedure for configuring safety-related I-slave-slave communication is identical to that in the standard system.

Proceed as follows:

1. Insert two F-CPUS from the "Hardware catalog" task card into the project.
2. Insert a suitable DP slave, e.g. IM 151-1 HF, article no. 6ES7151-1BA0... from the "Hardware catalog" task card into the network view of the *hardware and network editor*.
3. Insert a power module, a 4/8 F-DI module and a 4 F-DQ module in the device view of the ET 200S.
4. Activate "DP slave" mode (I-slave) for F-CPU 2 in the properties of its DP interface and assign this to F-CPU 1.
5. Assign the DP interface of the IM 151-1 HF to the DP master (F-CPU 1).
6. Select the DP interface of F-CPU 2 (I-slave) in the network view.
7. Select the "I/O communication" tab.

8. Use a drag-and-drop operation in the network view to move the ET 200S to the "Partner 2" column in the "I/O-communication" tab.

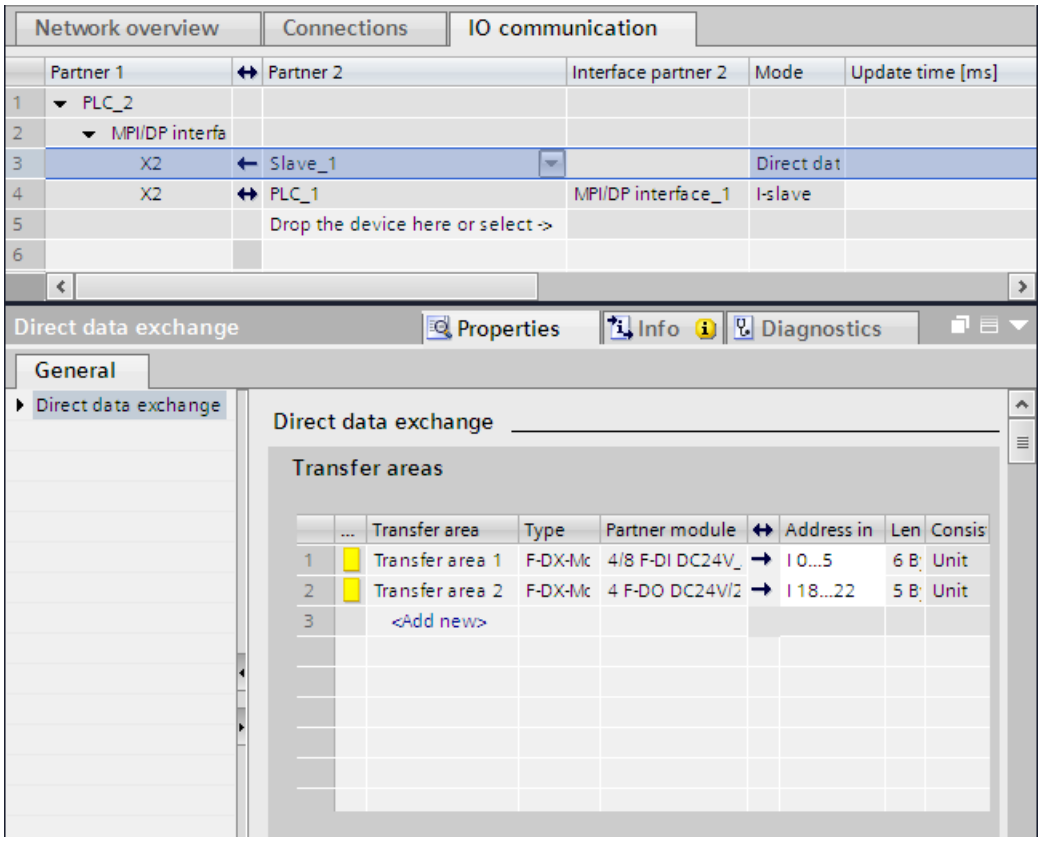


9. Click in the newly created line (←).
10. In "Transfer areas", create an F-DX-Mod connection (type "F-DX-Mod"). The F-DX-Mod connection is marked in yellow in the table. The addresses for the "partner module" 4/8 F-DI in the I-Slave (PLC_2) are displayed. You can change the addresses directly in the table, if required.

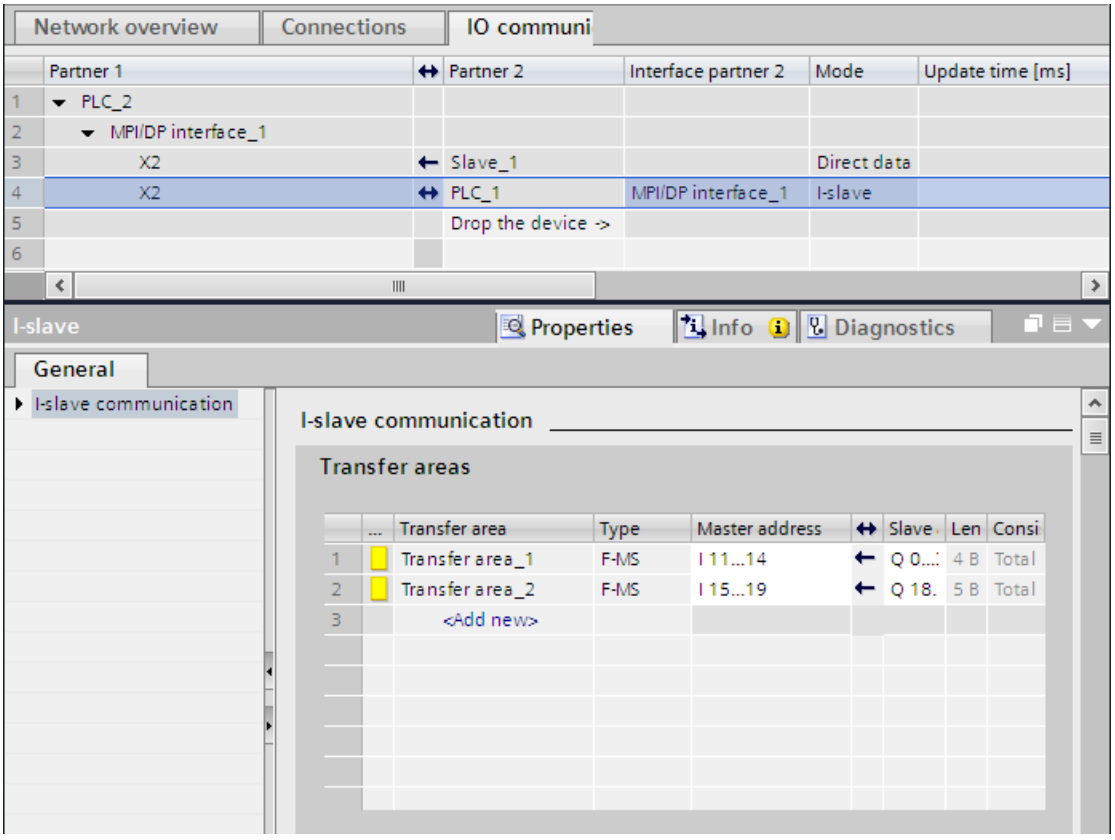
    This completes the configuration for the 4/8 F-DI module.
11. In "Transfer areas", create another F-DX-Mod connection.

12. Change the partner module to the 4 F-DO module, either directly in the "Transfer areas" table or in the details of transfer area 2, if the 4 F-DO module was not already selected.

    This completes the configuration for the 4 F-DO module.



In the "I-slave communication table" of the I-slave, a transfer area (type F-MS) for the master CPU is automatically created for each F-DX-Mod connection:



Change in configuration of I-slave-slave communication

WARNING

If you have added or deleted I-slave-slave communication for an F-I/O, you must compile and download the hardware configuration of the DP master as well as the hardware configuration of the I-slave.

The Collective F-Signature in the F-CPU of the I-slave and the Collective F-Signature in the F-CPU of the DP master (if a safety program exists there, too) are set to "0". You must then recompile the safety program(s). *(S019)*

6.6.3 Safety-related I-slave-slave communication - F-I/O access via process image

For F-I/O access via I-slave-slave communication, you use the process image (PII or PIQ) to access the F-I/O in the Safety program of the F-CPU of the I-slave. This is the same as F-I/O access to F-I/O that are directly assigned to an I-slave or DP master. In the I-slave you access the F-I/O with the addresses that were assigned for the F-DX-Mod connection in "Transfer areas" ("Direct data exchange" table).

In this case, ignore the displayed operand area. Access F-I/O with inputs using the PII and F-I/O with outputs using PIQ.

Information on I/O access can be found in F-I/O access (Page 157).

6.6.4 Limits for data transfer of safety-related I-slave-I-slave communication

Limits for data transfer

Note the maximum limit of 244 bytes of input data or 244 bytes of output data for transfer between an I-slave and a DP master.

An example of the amount of output data and input data that are assigned for safety-related communication is shown in the table below for an ET 200S 4/8 F-DI and an ET 200S 4 F-DO:

Safety-related communication	Communication connection	Assigned input and output data*	
		Between I-slave and DP master	
		Output Data in the I-slave	Input data in the I-slave
I-slave-slave	I-slave-slave communication with 4/8 F-DI	4 bytes	6 bytes
	I-slave-slave communication with 4 F-DO	5 bytes	5 bytes

* Example for 4/8 F-DI and 4 F-DO of ET 200S

Consider all additional configured safety-related and standard communication connections (F-MS, F-DX, F-DX-Mod., MS and DX connections) for the maximum limit of 244 bytes of input data or 244 bytes of output data for transfer between an I-slave and a DP master. If the maximum limit of 244 bytes of input data or 244 bytes of output data is exceeded, you will receive a corresponding error message.

Implementation of user acknowledgment 7

7.1 Implementing User Acknowledgment in the Safety Program of the F-CPU of a DP Master or IO controller

Options for user acknowledgment

Depending on the result of the risk analysis, you have the following options for implementing a user acknowledgment:

- With an acknowledgment key that you connect to an F-I/O with inputs.
- With an operator control and monitoring system.
- With an acknowledgment key that you connect to a standard I/O with inputs.
- By other mechanisms for reading in the user acknowledgment.

User acknowledgment by means of acknowledgment key

NOTE

When you implement user acknowledgment by means of acknowledgment key, and a communication error, an F-I/O fault, or a channel fault occurs in the F-I/O to which the acknowledgment key is connected, then it will not be possible to acknowledge the reintegration of this F-I/O.

This "blocking" can only be removed by a STOP-to-RUN transition of the F-CPU. In a redundant S7-1500HF system, you must set both HF CPUs or the redundant S7-1500HF system to STOP before you restart the HF CPUs.

Consequently, it is recommended that you also provide for an acknowledgment by means of an operator control and monitoring system, in order to acknowledge reintegration of an F-I/O to which an acknowledgment key is connected.

A user acknowledgment may be issued using an acknowledgment key connected to a standard I/O with inputs or other mechanisms for reading the user acknowledgment if the risk analysis allows this.

User acknowledgment by means of an operator control and monitoring system

For implementation of a user acknowledgment by means of an operator control and monitoring system, the ACK_OP: Fail-safe acknowledgment (STEP 7 Safety V21) (Page 604) instruction is required.

Procedure for programming user acknowledgment by means of an operator control and monitoring system (S7-300, S7-400)

1. Select the "ACK_OP" instruction in the "Instructions" task card and place it in your safety program. The acknowledgment signal for evaluating user acknowledgments is provided at output OUT of ACK_OP.
2. On your operator control and monitoring system, set up a field for manual entry of an "acknowledgment value" of "6" (1st step in acknowledgment) and an "acknowledgment value" of "9" (2nd step in acknowledgment).

   or

   Assign function key 1 to transfer an "acknowledgment value" of "6" (1st step in acknowledgment) once, and function key 2 to transfer an "acknowledgment value" of "9" (2nd step in acknowledgment) once. You must assign the InOut IN (in the data area of the ACK_OP instruction) to the field or the function keys.
3. Optional: In your operator control and monitoring system, evaluate output Q in the instance DB of ACK_OP to show the time frame within which the 2nd step in acknowledgment must occur or to indicate that the 1st step in acknowledgment has already occurred.

If you want to perform a user acknowledgment exclusively from a programming device or PC using the watch table (monitor/modify tag) without having to disable safety mode, then you must transfer an operand (memory word or DBW of a DB of the standard user program) at InOut IN when calling ACK_OP. You can then transfer "acknowledgment values" "6" and "9" on the programming device or PC by modifying the memory word or DBW of a DB once. The memory word or DBW of a DB must not be written by the program.

NOTE

If you connect the InOut IN to a memory word or DBW of a DB, you must use a separate memory word or DBW of a DB of the standard user program for each instance of the ACK_OP instruction at the InOut IN.

⚠ WARNING

The two acknowledgment steps must **not** be triggered by one single operation, for example, by programming them including the time conditions and using one function key to trigger them.

The two separate acknowledgment steps also prevent erroneous triggering of an acknowledgment, for example, by your non-fail-safe operator control and monitoring system. *(S013)*

⚠ WARNING

If you have operator control and monitoring systems and F-CPUs that are interconnected and use the ACK_OP instruction for fail-safe acknowledgment, you need to ensure that the intended F-CPU will be addressed **before** you perform the two acknowledgment steps.

- Store a network-wide* unique name for the F-CPU in a DB of your standard user program in each F-CPU.
- In your operator control and monitoring system, set up a text box from which you can read out the F-CPU name from the DB online before executing the two acknowledgment steps.
- Optional:

  In your operator control and monitoring system, set up a text box to permanently store the F-CPU name. Then, you can determine whether the intended F-CPU is being addressed by simply comparing the F-CPU name read out online with the permanently stored name. *(S014)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet.

NOTE

The configuration of your operator control and monitoring system does not have any effect on the Collective F-Signature.

Procedure for programming user acknowledgment by means of an operator control and monitoring system (S7-1200, S7-1500)

1. Select the "ACK_OP" instruction in the "Instructions" task card and place it in your safety program. The acknowledgment signal for evaluating user acknowledgments is provided at output OUT of ACK_OP.
2. Assign the ACK_ID input an identifier between 9 and 30000 for the acknowledgment.
3. Assign the InOut IN a memory word or DBW of a DB of the standard user program.

   NOTE

   You must supply the InOut IN with a separate memory word or DBW of a DB of the standard user program for each instance of the ACK_OP instruction.

4. On your operator control and monitoring system, set up a field for manual entry of an "acknowledgment value" of "6" (1st step in acknowledgment) and the "Identifier" configured at the ACK_ID input (2nd step in acknowledgment).

   or

   Allocate a function key 1 for a one-time transfer of the "acknowledgment value" of "6" (1st step in acknowledgment) and a function key 2 for a one-time transfer of the "Identifier" set at the ACK_ID input (2nd step in acknowledgment).
   You must assign the memory word or the DBW of the DB of the standard user program assigned to the InOut IN to the field or the function keys.

5. Optional: In your operator control and monitoring system, evaluate output Q in the instance DB of ACK_OP to show the time frame within which the 2nd step in acknowledgment must occur or to indicate that the 1st step in acknowledgment has already occurred.

⚠ WARNING

The two acknowledgment steps must not be triggered by one single operation, for example, by programming them including the time conditions and using one function key to trigger them.

The two separate acknowledgment steps also prevent erroneous triggering of an acknowledgment, for example, by your non-fail-safe operator control and monitoring system. *(S013)*

⚠ WARNING

If you have operator control and monitoring systems and F-CPUs that are interconnected and use the ACK_OP instruction for fail-safe acknowledgment, you need to ensure that the intended F-CPU will be addressed **before** you perform the two acknowledgment steps.

Alternative 1:

- The value for each identifier of the acknowledgment (ACK_ID input; data type: INT) can be freely selected in the range from 9 to 30000, but must be unique network-wide* for all instances of the ACK_OP instruction.
  You must supply the ACK_ID input with constant values when calling the instruction. Direct read or write access to the ACK_ID tag in the associated instance DB is not permitted!

Alternative 2:

- Store a network-wide* unique name for the F-CPU in a DB of your standard user program in each F-CPU.
- In your operator control and monitoring system, set up a text box from which you can read out the F-CPU name from the DB online before executing the two acknowledgment steps.
- Optional:

  In your operator control and monitoring system, set up a text box to permanently store the F-CPU name. Then, you can determine whether the intended F-CPU is being addressed by simply comparing the F-CPU name read out online with the permanently stored designation. *(S047)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet.

NOTE

The supply of the IN iInOut of the ACK_OP instruction as well as the configuration of your operator control and monitoring system do not have any effect on the Collective F-Signature, the Collective F-SW-Signature or the signature of the block that calls the ACK_OP instruction.

Changes to the supply of the IN InOut or to the configuration of your operator control and monitoring system therefore do not result in a changed Collective F-Signature/Collective F-SW-Signature/signature of the calling block.

Example of procedure for programming a user acknowledgment for reintegrating an F-I/O

1. Optional: set the ACK_NEC tag in the respective F-I/O DB (Page 166) to "0" if automatic reintegration (without user acknowledgment) is to take place after an F-I/O fault or a channel fault.

   ⚠ WARNING

   The parameter assignment of the ACK_NEC tag of the F-I/O DB with the value "0" is only allowed if automatic reintegration is permitted for the relevant process from a safety standpoint. *(S010)*

2. Optional: Evaluate the QBAD or QBAD_I_xx/QBAD_O_xx (S7-300, S7-400) tags or value status (S7-1200, S7-1500) or DIAG in the respective F-I/O DB to trigger an indicator light in the event of an error, and/or generate error messages on the operator control and monitoring system in your standard user program by evaluating the above tags or the value status. These messages can be evaluated before performing the acknowledgment operation. Alternatively, you can evaluate the diagnostic buffer of the F-CPU.
3. Optional: Evaluate the ACK_REQ tag in the respective F-I/O DB, for example, in the standard user program or on the operator control and monitoring system, to query or to indicate whether user acknowledgment is required.
4. Assign the input of the acknowledgment key or the OUT output of the ACK_OP instruction to the ACK_REI tag in the respective F-I/O DB or the ACK_REI_GLOB input of the ACK_GL instruction (see above).

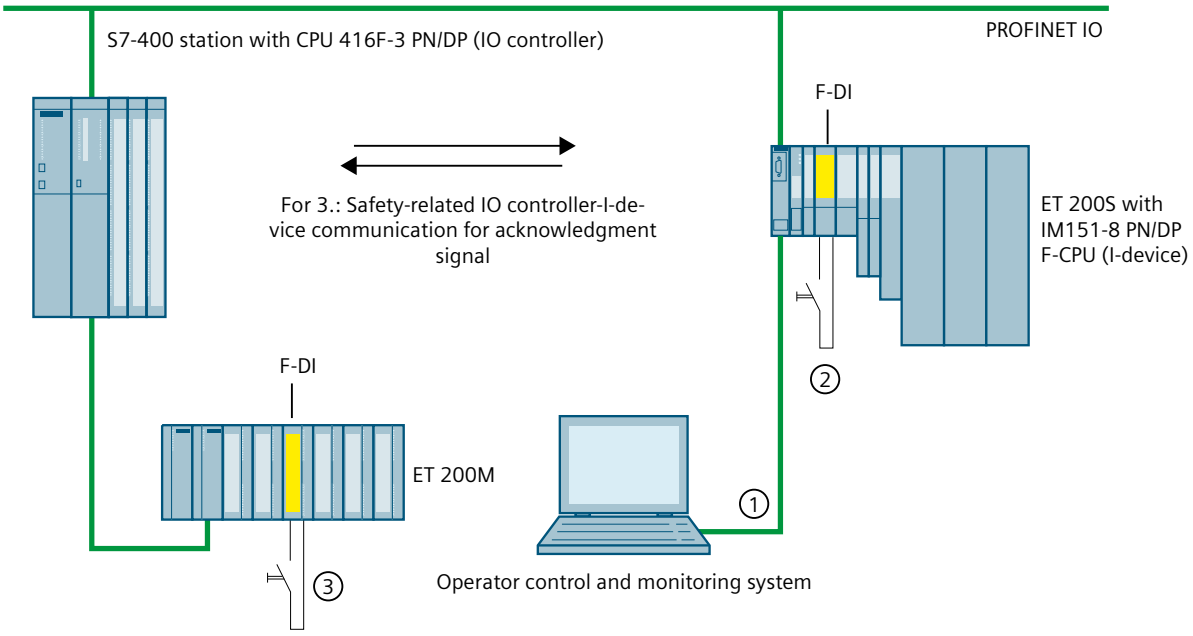
7.2 Implementing user acknowledgment in the safety program of the F-CPU of a I-slave or I-device

Options for user acknowledgment

You can implement a user acknowledgment by means of:

1. An HMI system that you can use to access the F-CPU of the I-slave/I-Device
2. An acknowledgment key that you connect to an F-I/O with inputs that is assigned to the F-CPU of the I-slave/I-Device
3. An acknowledgment key that you connect to an F-I/O with inputs that is assigned to the F-CPU of the DP master/IO controller

These three options are illustrated in the figure below.



1. User acknowledgment by means of an HMI system with which you can access the F-CPU of the I-slave/I-device

The ACK_OP: Fail-safe acknowledgment (STEP 7 Safety V21) (Page 604) instruction is required to implement user acknowledgment with an HMI system that you can use to access the F-CPU of the I-slave/I-device.

Programming procedure

Follow the procedure described in "Implementing User Acknowledgment in the Safety Program of the F-CPU of a DP Master or IO controller (Page 190)" under "Programming procedure ...".

From your HMI system, you can then directly access the instance DB of ACK_OP in the I-slave/I-Device.

2. User acknowledgment by means of an acknowledgment key at an F-I/O with inputs that are assigned to the F-CPU of the I-slave/I-device

NOTE

In the event of a communication error, F-I/O fault, or channel fault in the F-I/O to which the acknowledgment key is connected, an acknowledgment for reintegration of this F-I/O is no longer possible.

This "blocking" can only be removed by a STOP-to-RUN transition of the F-CPU of the I-slave/I-Device.

Consequently, it is recommended that you also provide for an acknowledgment by means of an HMI system that you can use to access the F-CPU of the I-slave/I-device, in order to acknowledge reintegration of an F-I/O to which an acknowledgment key is connected (see 1).

3. User acknowledgment by means of an acknowledgment key at an F-I/O with inputs that are assigned to the F-CPU of the DP master/IO controller

If you want to use the acknowledgment key that is assigned to the F-CPU at the DP master/IO controller to perform user acknowledgment in the safety program of the F-CPU of an I-slave/I-device as well, you must transmit the acknowledgment signal from the safety program in the F-CPU of the DP master/IO controller to the safety program in the F-CPU of the I-slave/I-device using safety-related master-I-slave/IO controller-I-device communication.

Programming procedure

1. Place the SENDDP (Page 612) instruction in the safety program in the F-CPU of the DP master/IO controller.
2. Place the RCVDP (Page 612) instruction in the safety program in the F-CPU of the I-slave/I-Device.
3. Supply an input SD_BO_xx of SENDDP with the input of the acknowledgment key.
4. The acknowledgment signal for evaluating user acknowledgments is now available at the corresponding RD_BO_xx output of RCVDP.

   The acknowledgment signal can now be read in the program sections in which further processing is to take place with fully qualified access directly in the associated instance DB (for example, "RCVDP_DB".RD_BO_02).

5. Supply the corresponding input SUBBO_xx of RCVDP with FALSE (fail-safe value 0) to ensure that user acknowledgment is not accidentally triggered before communication is established for the first time after startup of the sending and receiving F-systems, or in the event of a safety-related communication error.

NOTE

If a communication error, an F-I/O error, or a channel fault occurs at the F-I/O to which the acknowledgment key is connected, then an acknowledgment for reintegration of this F-I/O will no longer be possible.

This "blocking" can only be removed by a STOP-to-RUN transition of the F-CPU of the DP master/IO controller.

Consequently, it is recommended that you also provide for an acknowledgment by means of an HMI system that you can use to access the F-CPU of the DP master/IO controller, in order to acknowledge reintegration of the F-I/O to which an acknowledgment key is connected.

If a safety-related master-I-slave/IO controller-I-Device communication error occurs, the acknowledgment signal cannot be transmitted, and an acknowledgment for reintegration of safety-related communication is no longer possible.

This "blocking" can only be removed by a STOP-to-RUN transition of the F-CPU of the I-slave/I-Device.

Consequently, it is recommended that you also provide for an acknowledgment by means of an HMI system that you can use to access the F-CPU of the I-slave/I-Device, in order to acknowledge reintegration of the safety-related communication for transmission of the acknowledgment signal (see 1).

Data exchange between standard user program and safety program

8

You have the option of transferring data between the safety program and the standard user program. Tags can be transferred using DBs, F-DBs and bit memory:

	From the standard user program		From the safety program	
	Read access	**Write access**	**Read access**	**Write access**
Tag from DB	Permitted	Permitted	A tag from the DB can be read-accessed *or* write-accessed	
Tag from F-DB	Permitted	**Not permitted**	Permitted	Permitted
Bit memory	Permitted	Permitted	Bit memory can be read-accessed *or* write-accessed	

You can also access the process image of the standard I/O and F-I/O:

		From the standard user program		From the safety program	
		Read access	**Write access**	**Read access**	**Write access**
Process image of standard I/O	PII	Permitted	Permitted	Permitted	**Not permitted**
	PIQ	Permitted	Permitted	**Not permitted**	Permitted
Process image of F-I/O	PII	Permitted	**Not permitted**	Permitted	**Not permitted**
	PIQ	Permitted	**Not permitted**	**Not permitted**	Permitted

Note possible restrictions/particularities when using Software Units or a Safety Unit (Page 116).

Decoupling of the safety program from the standard user program

For data exchange between standard user program and safety program, we recommend that you define special data blocks (transfer data blocks) in which the data to be exchanged is stored. This action allows you to decouple the blocks of the standard user program and the safety program. The changes in the standard user program do not affect the safety program (and vice versa) provided these data blocks are not modified.

8.1 Data Transfer from the Safety Program to the Standard User Program

Reading data of the safety program in the standard user program

The standard user program can read all data of the safety program, for example using symbolic (fully qualified) accesses to the following:

- The instance DBs of the F-FBs ("Name of instance DB".Signal_x)
- F-DBs (for example "Name of F_DB".Signal_1)

- The process image input and output of F-I/O (for example "Emergency_Stop_Button_1" (I 5.0))

NOTE

For S7-300/400 F-CPUs

The process image input of F-I/O is updated not only at the start of the F-runtime group, but also by the standard operating system.

To find the standard operating system update times, refer to the *Help on STEP 7* under "Process image input and output". For F-CPUs that support process image partitions, also bear in mind the update times when process image partitions are used. For this reason, when the process image input of F-I/O is accessed in the standard user program, it is possible to obtain different values than in the safety program. The differing values can occur due to:

– Different update times
– Use of fail-safe values in the safety program

To obtain the same values in the standard user program as in the safety program, you must not access the process image input in the standard program until after execution of an F-runtime group. In this case, you can also evaluate the QBAD or QBAD_I_xx tag in the associated F-I/O DB in the standard user program, in order to find out whether the process image input is receiving fail-safe values (0) or process data. When using process image partitions, also make sure that the process image is not updated by the standard operating system or by the UPDAT_PI instruction between execution of an F-runtime group and evaluation of the process image input in the standard user program.

NOTE

For S7-1200/1500 F-CPUs

The process image input of F-I/O is updated prior to processing the main safety block.

During the data transfer from the safety program to the standard user program, the integrity of the safety program and the safety-related data is not checked (see also section "Product Overview (Page 23)").

Writing data from the safety program to the standard user program

You can also write safety program data directly in the safety program to the standard user program (see also the table of supported operand areas in: Restrictions in the programming languages FBD/LAD (Page 119)).

This is also true for non-fail-safe service information (e.g. outputs DIAG of the fail-safe instructions).

Note the table in section "Data exchange between standard user program and safety program (Page 198)".

When writing to a tag of the ARRAY data type, the index can either be a constant or a tag.

8.2 Data Transfer from Standard User Program to Safety Program

As a basic principle, only fail-safe data or fail-safe signals from F-I/O and other safety programs (in other F-CPUs) can be processed in the safety program, as standard tags are unsafe.

If you have to process tags from the standard user program in the safety program, however, you can evaluate either bit memory, tags from a standard DB, or the process image input (PII) of standard I/O in the safety program (see table of supported operand areas in: Restrictions in the programming languages FBD/LAD (Page 119)).

Note the table in section "Data exchange between standard user program and safety program (Page 198)".

When reading a tag of the ARRAY data type, the index must be a constant. Tags are not permitted as index.

Note that structural changes to standard data blocks which are used in the safety program lead to inconsistencies of the safety program and can therefore only be executed if access permission for the safety-related project data is available. In this case the Collective F-Signature is the same as the original again after compilation. To prevent this effect, use "transfer data blocks" between the standard user program and the safety program.

⚠ WARNING

Because the values in these tags are not generated safely, you must program additional process-specific plausibility checks in the safety program to ensure that no dangerous states can arise. If bit memory, a tag of a standard DB, or an input of standard I/O is used in both F-runtime groups, you must perform the plausibility check separately in each F-runtime group. *(S015)*

To facilitate checks, all PLC tags from the standard user program that are evaluated in the safety program are printed out when creating the safety summary. (Page 388)

PLC tags from the standard user program that are evaluated in know-how-protected F-blocks are not included in the safety summary. The plausibility check must already be implemented by the creator of the know-how protected F-blocks.

Examples: Programming plausibility checks

- Use Comparison (Page 546) instructions to check whether tags from the standard user program exceed or fall below permitted high and low limits. You can then influence your safety function with the result of the comparison.
- Use the ---(S)---: Set output (STEP 7 Safety V21) (Page 463), ---(R)---: Reset output (STEP 7 Safety V21) (Page 462) or SR: Set/reset flip-flop (STEP 7 Safety V21) (Page 464) instructions, for example, with tags from the standard user program to allow a motor to be switched off, but not switched on.
- For switch-on sequences, use the AND logic operation instruction, for example, to logically combine tags from the standard user program with switch-on conditions that you derive from fail-safe tags.

If you want to process tags from the standard user program in the safety program, bear in mind that there is not a sufficiently simple method of checking plausibility for all tags.

Reading tags from the standard user program that can change during the runtime of an F-runtime group

If you want to read tags from the standard user program (bit memory, tags of a standard DB, or PII of standard I/O) in the safety program, and these tags can be changed - either by the standard user program or an operator control and monitoring system - during runtime of the F-runtime group in which they are read (for example because your standard user program is being processed by a higher-priority cyclic interrupt), you must use bit memory or tags of a standard DB for this purpose. We recommend using standard FCs for preprocessing (Page 88) for S7-1200/1500 F-CPUs.

(S7-300, S7-400) You must write the bit memory or tags of a standard DB with the tags from the standard user program immediately before calling the F-runtime group.

You are then permitted to access only this bit memory or these tags of a standard DB in the safety program.

Also note that **clock memory** that you defined when configuring your F-CPU in the "Properties" tab can change during runtime of the F-runtime group, since clock memory runs asynchronously to the F-CPU cycle.

NOTE

The F-CPU can go to STOP if this is not observed. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

Safety-related communication

9

9.1 Communication via SENDDP/RCVDP or SENDS7/RCVS7(S7-300, S7-400) instructions:

9.1.1 Overview

Introduction

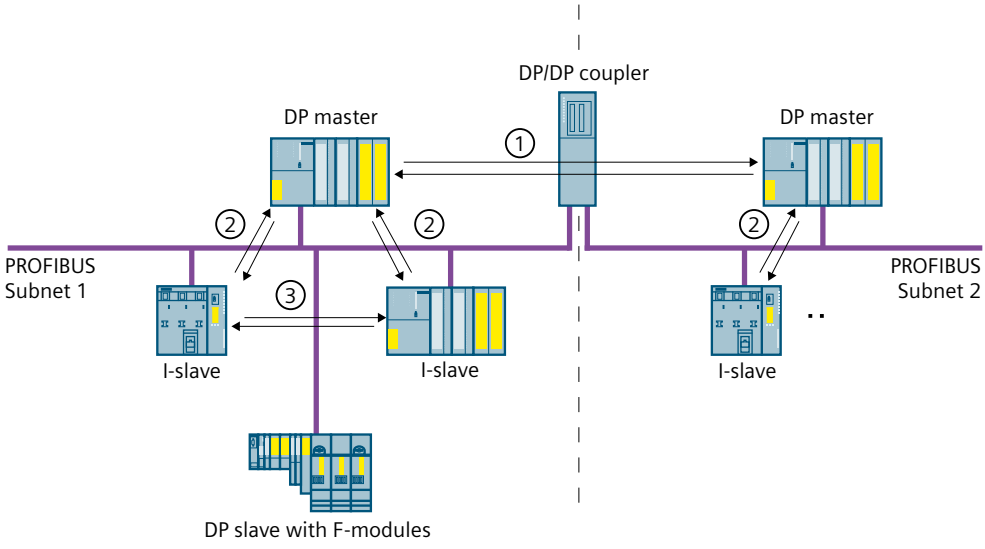
This section gives an overview of the options for safety-related communication options using the SENDDP/RCVDP or SENDS7/RCVS7 instructions in SIMATIC Safety F-systems.

Options for safety-related CPU-CPU communication

Safety-related CPU-CPU communication	On subnet	Additional hardware required	Required instructions
IO controller-IO controller communication	PROFINET IO	PN/PN coupler	SENDDP/RCVDP
Master-master communication	PROFIBUS DP	DP/DP coupler	SENDDP/RCVDP
IO controller-I-device communication	PROFINET IO	—	SENDDP/RCVDP
Master-I-slave communication	PROFIBUS DP	—	SENDDP/RCVDP
I-slave-I-slave communication	PROFIBUS DP	—	SENDDP/RCVDP
IO controller-I-slave communication	PROFINET IO and PROFIBUS DP	IE/PB link	SENDDP/RCVDP
Safety-related communication via S7 connections	Industrial Ethernet	—	SENDS7/RCVS7
IO controller-IO controller communication for S7 Distributed Safety	PROFINET IO	PN/PN coupler	SENDDP/RCVDP
Master-master communication for S7 Distributed Safety	PROFIBUS DP	DP/DP coupler	SENDDP/RCVDP
Safety-related communication to S7 Distributed Safety or S7 F Systems via S7 connections	Industrial Ethernet	—	SENDS7/RCVS7

Overview of safety-related communication via PROFIBUS DP

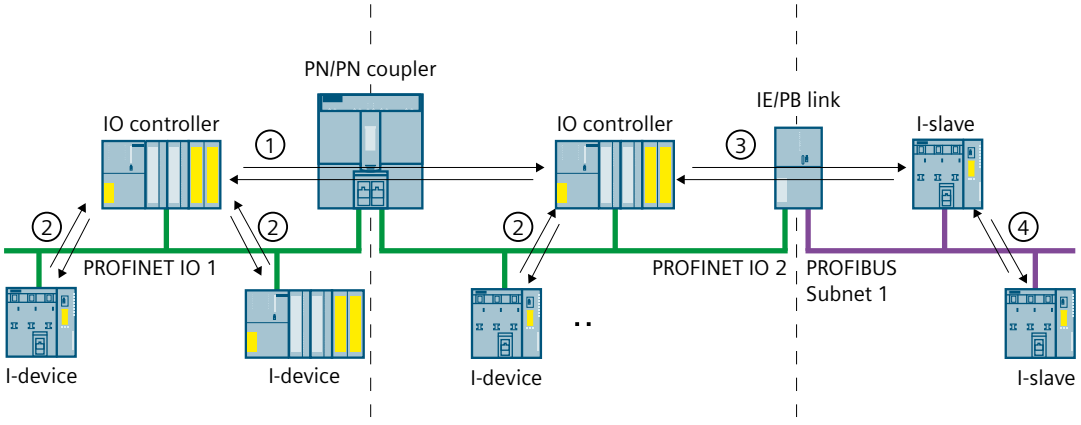
The figure below presents an overview of the three options for safety-related communication via PROFIBUS DP in SIMATIC Safety F-systems with S7-300/400 F-CPUs.



① Safety-related master-master communication
② Safety-related master-I-slave communication
③ Safety-related I-slave-I-slave communication

Overview of safety-related communication via PROFINET IO

The figure below presents an overview of the four options for safety-related communication via PROFINET IO in SIMATIC Safety F-systems with S7-300/400 F-CPUs. If an IE/PB-link is used, safety-related communication is possible between assigned I-slaves.



① Safety-related IO controller-IO controller communication
② Safety-related IO controller-I-device communication
③ Safety-related IO controller-I-slave communication
④ Safety-related I-slave-I-slave communication integrating an IO controller

Safety-related CPU-CPU communication via PROFIBUS DP or PROFINET IO

In safety-related CPU-CPU communication, a fixed amount of fail-safe data of the data type INT or BOOL is transmitted fail-safe between the safety programs in F-CPUs of DP masters/I-slaves or IO controllers/I-devices.

The data are transferred using the SENDDP instruction for sending and the RCVDP instruction for receiving. The data are stored in configured transfer areas of the devices. Each transfer area consists of one input and one output address area.

Safety-related I-slave-slave communication via PROFIBUS DP

Safety-related I-slave-slave communication with F-I/O is possible in a DP slave that supports safety-related I-slave-slave communication, for example with all ET 200SP F-modules with IM 155-6 DP HF, firmware > V3.1, with all ET 200S F-modules with IM 151-1 HF, with all fail-safe S7-300 signal modules with IM 153-2, as of article number 6ES7153-2BA01-0XB0, firmware > V4.0.0.

Safety-related communication between the safety program of the F-CPU of an I-slave and F-I/O of a DP slave takes place using direct data exchange, as in the standard program. The process image is used to access the channels of the F-I/O in the safety program of the F-CPU of the I-slave.

Safety-related CPU-CPU communication via Industrial Ethernet

Safety-related CPU-CPU communication via Industrial Ethernet is possible using S7 connections, both from and to the following:

- S7-300 F-CPUs via the integrated PROFINET interface
- S7-400 F-CPUs via the integrated PROFINET interface or a CP 443-1 or CP 443-1 Advanced-IT

In safety-related communication via S7 connections, a specified amount of fail-safe data of data type BOOL, INT, WORD, DINT, DWORD, or TIME is transferred fail-safe between the safety programs of the F-CPUs linked by the S7 connection.

The data transfer makes use of the SENDS7 instruction for sending and the RCVS7 instruction for receiving. Data are exchanged using one F-DB ("F-communication DB") each at the sender and receiver ends.

Safety-related CPU-CPU communication with *S7 Distributed Safety* or *S7 F-systems*

Safety-related communication is possible from F-CPUs in *SIMATIC Safety* to F-CPUs in *S7 Distributed Safety* or *S7 F-systems*.

9.1.2 Safety-related IO controller-IO controller communication

9.1.2.1 Configure safety-related IO controller-IO controller communication

Introduction

Safety-related communication between safety programs of the F-CPUs of IO controllers takes place over a PN/PN coupler that you set up between the F-CPUs.

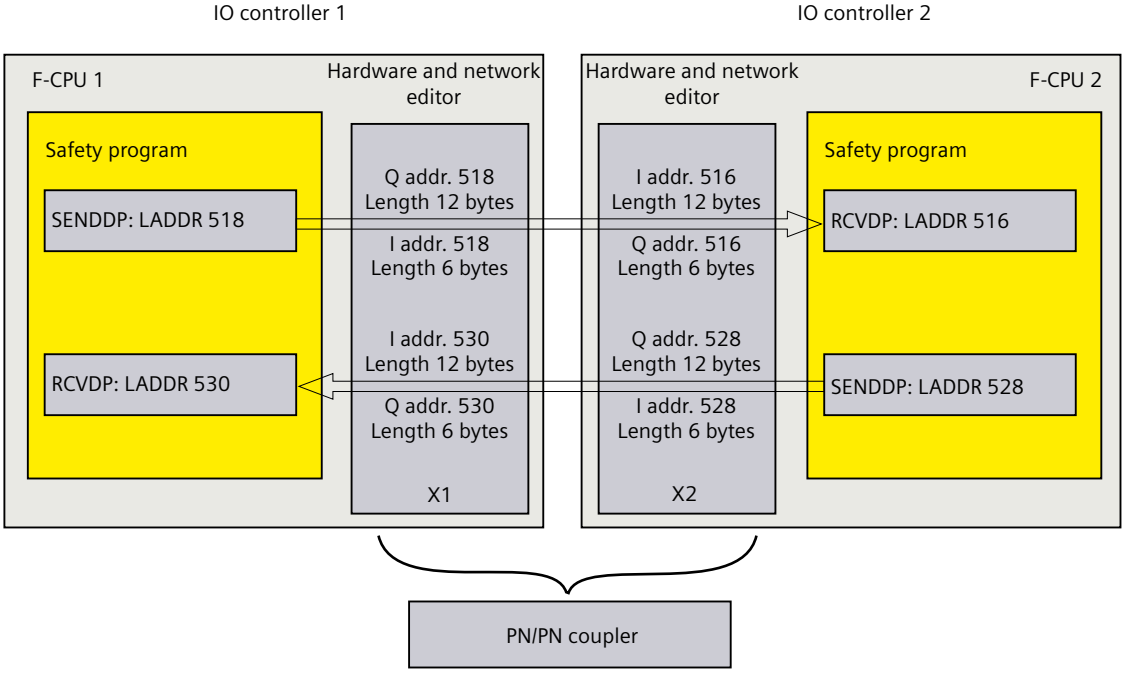
For 416F-2 DP CPUs without an integrated PROFINET interface, use a CP 443-1 or CP 443-1 Advanced-IT.

NOTE

Deactivate the "Data validity display DIA" parameter in the properties for the PN/PN coupler in the *hardware and network editor*. This is the default setting. Otherwise, safety-related IO controller-IO controller communication is not possible.

Configuring transfer areas

You must configure one transfer area for output data and one transfer area for input data in the *hardware and network editor* for each safety-related communication connection between two F-CPUs in the PN/PN coupler. The figure below shows how both of the F-CPUs are able to send **and** receive data (bidirectional communication). One transfer area for output data and one transfer area for input data must be configured in the PN/PN coupler for each of the two communication connections.



Rules for defining transfer areas

The transfer area for output data and the transfer area for input data for the **data to be sent** must begin with the same start address. A total of 12 bytes (consistent) is required for the transfer area for output data; 6 bytes (consistent) are required for the transfer area for input data.

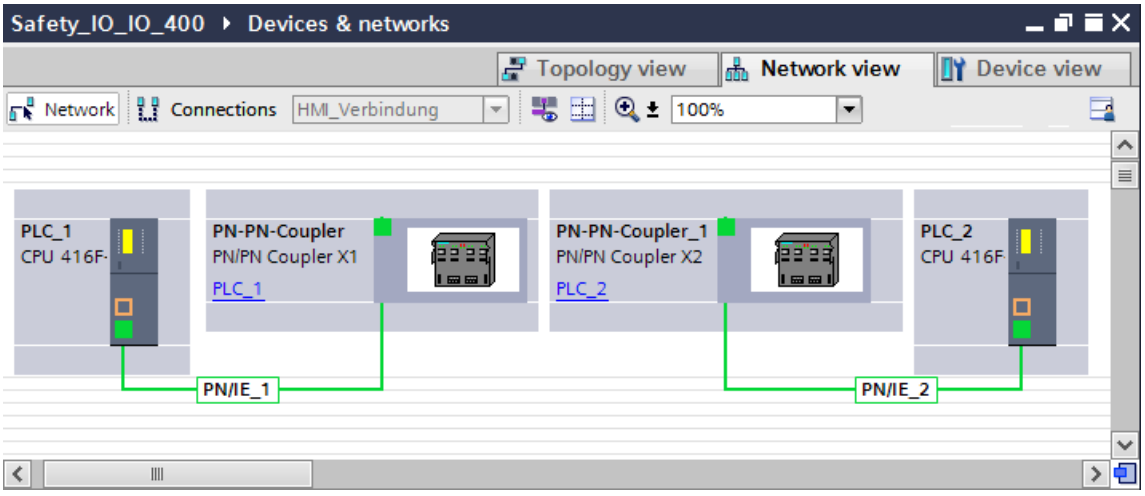
The transfer area for input data and the transfer area for output data for the **data to be received** must begin with the same start address. A total of 12 bytes (consistent) is required for the transfer area for input data; 6 bytes (consistent) are required for the transfer area for output data.

Procedure for configuration

The procedure for configuring safety-related IO controller-IO controller communication is identical to that in the standard system.

Proceed as follows:

1. Insert two F-CPUS from the "Hardware catalog" task card into the project.
2. Switch to the network view of the *hardware and network editor*.
3. Select a PN/PN Coupler X1 and a PN/PN Coupler X2 from "Other field devices\PROFINET IO\Gateway\Siemens AG\PN/PN Coupler" in the "Hardware catalog" task card and insert them into the network view of the hardware and network editor.
4. Connect the PN interface of the F-CPU 1 with the PN interface of the PN/PN Coupler X1 and the PN interface of the F-CPU 2 with the PN interface of PN/PN Coupler X2.



5. Switch to the device view of PN/PN Coupler X1 for bidirectional communication connections i.e. where each F-CPU is both to send and to receive data. Select the following modules from "IN/OUT" in the "Hardware catalog" task card (with filter activated), and insert them in the "Device overview" tab:
   - One "IN/OUT 6 bytes / 12 bytes" module and
   - One "IN/OUT 12 bytes / 6 bytes" module

6. In the properties of the modules, assign the addresses outside the process image as follows:

   For the "IN/OUT 6 bytes / 12 bytes" module for sending data for example:

   – Input addresses: Start address 518
   – Output addresses: Start address 518

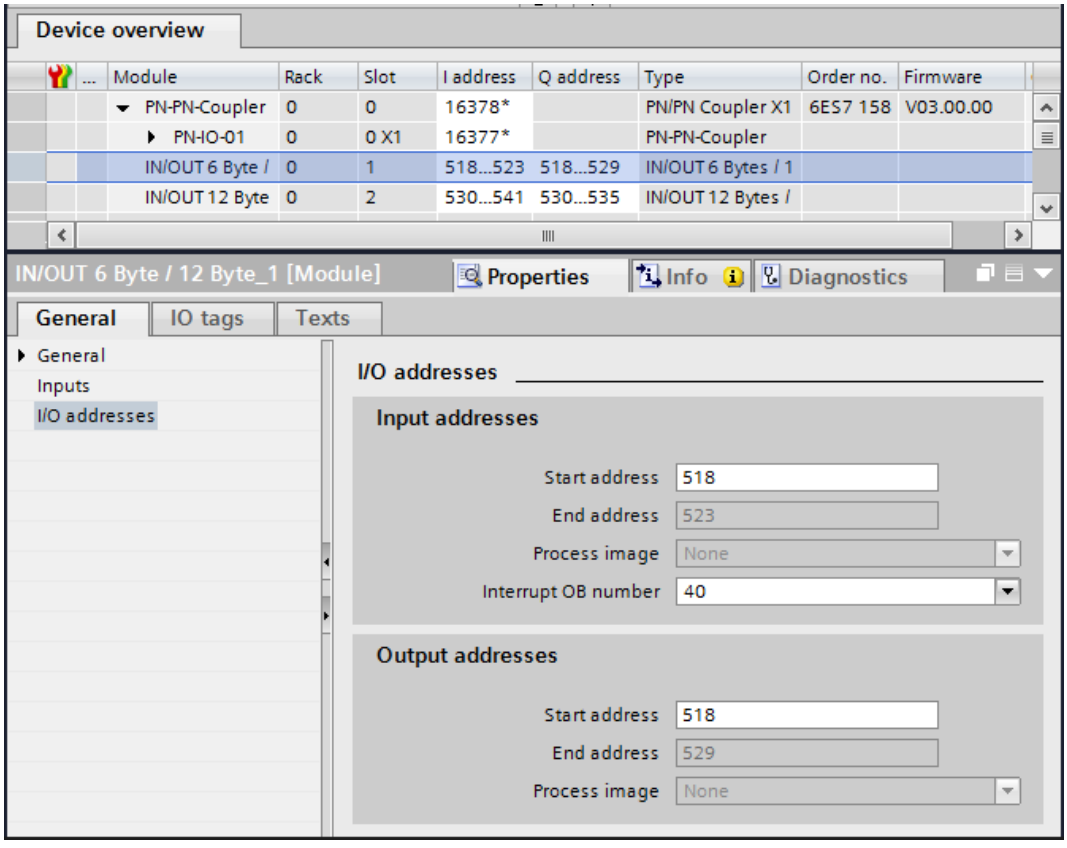
   For the "IN/OUT 12 bytes / 6 bytes" module for receiving data for example:

   – Input addresses: Start address 530
   – Output addresses: Start address 530

NOTE

Make sure that you assign identical start addresses for the address areas of the output and input data.

Tip: Make a note of the start addresses of the transfer areas. You need these to program the SENDDP and RCVDP blocks (LADDR input).



7. Select the following modules from "IN/OUT" in the device view of PN/PN coupler X2 and insert them in the "Device overview" tab:

   – One "IN/OUT 12 bytes / 6 bytes" module and
   – One "IN/OUT 6 bytes / 12 bytes" module

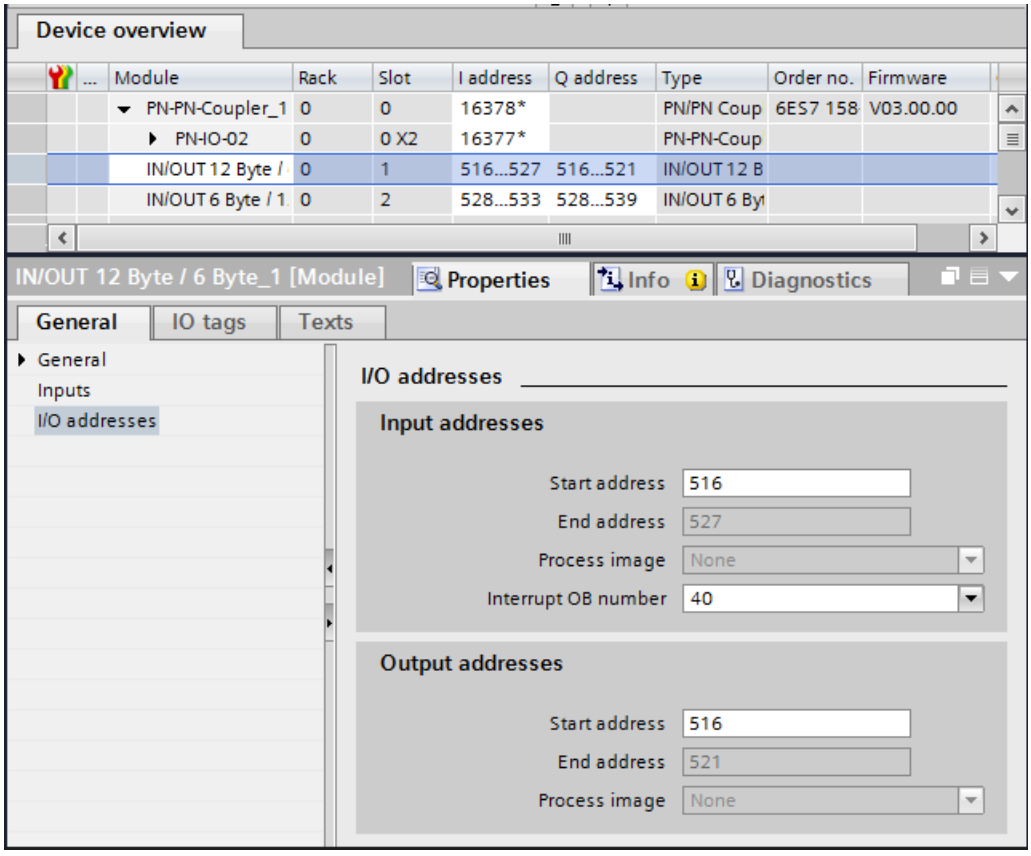
8. In the properties of the modules, assign the addresses outside the process image as follows:

   For the "IN/OUT 12 bytes / 6 bytes" module for receiving data for example:

   – Input addresses: Start address 516
   – Output addresses: Start address 516

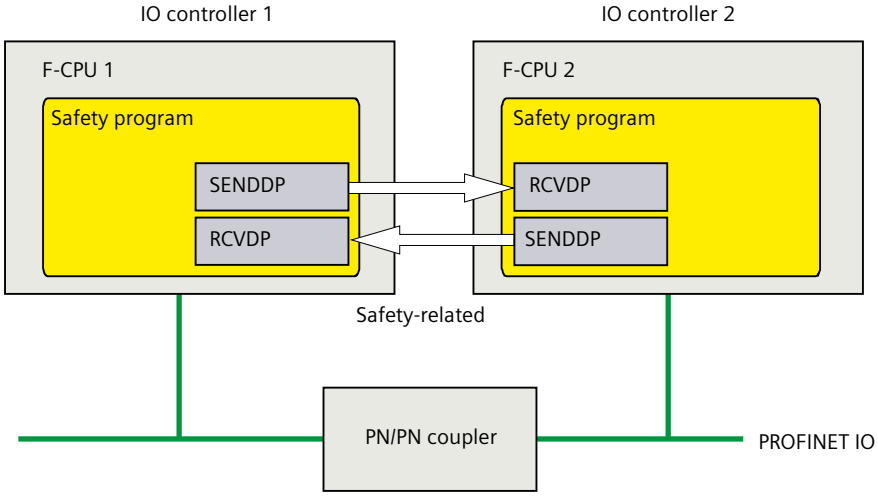
   For the "IN/OUT 6 bytes / 12 bytes" module for sending data for example:

   – Input addresses: Start address 528
   – Output addresses: Start address 528



9.1.2.2 Program safety-related IO controller-IO controller communication

Communication via SENDDP and RCVDP instructions



Safety-related communication between the F-CPUs of the IO controllers uses the SENDDP and RCVDP instructions for sending and receiving, respectively. These can be used to perform a fail-safe transfer of a *fixed* amount of fail-safe data of the data type INT or BOOL.

You can find these instructions in the "Instructions" task card under "Communication". The RCVDP instruction **must** be called at the start of the main safety block. The SENDDP instruction **must** be called at the end of the main safety block.

Note that the send signals are not sent until after the SENDDP instruction call at the end of execution of the relevant F-runtime group.

You can find a detailed description of the SENDDP and RCVDP instructions in SENDDP and RCVDP: Send and receive data via PROFIBUS DP/PROFINET IO (STEP 7 Safety V21) (Page 612).

Requirement for programming

The transfer areas for input and output data for the PN/PN coupler must be configured.

Programming procedure

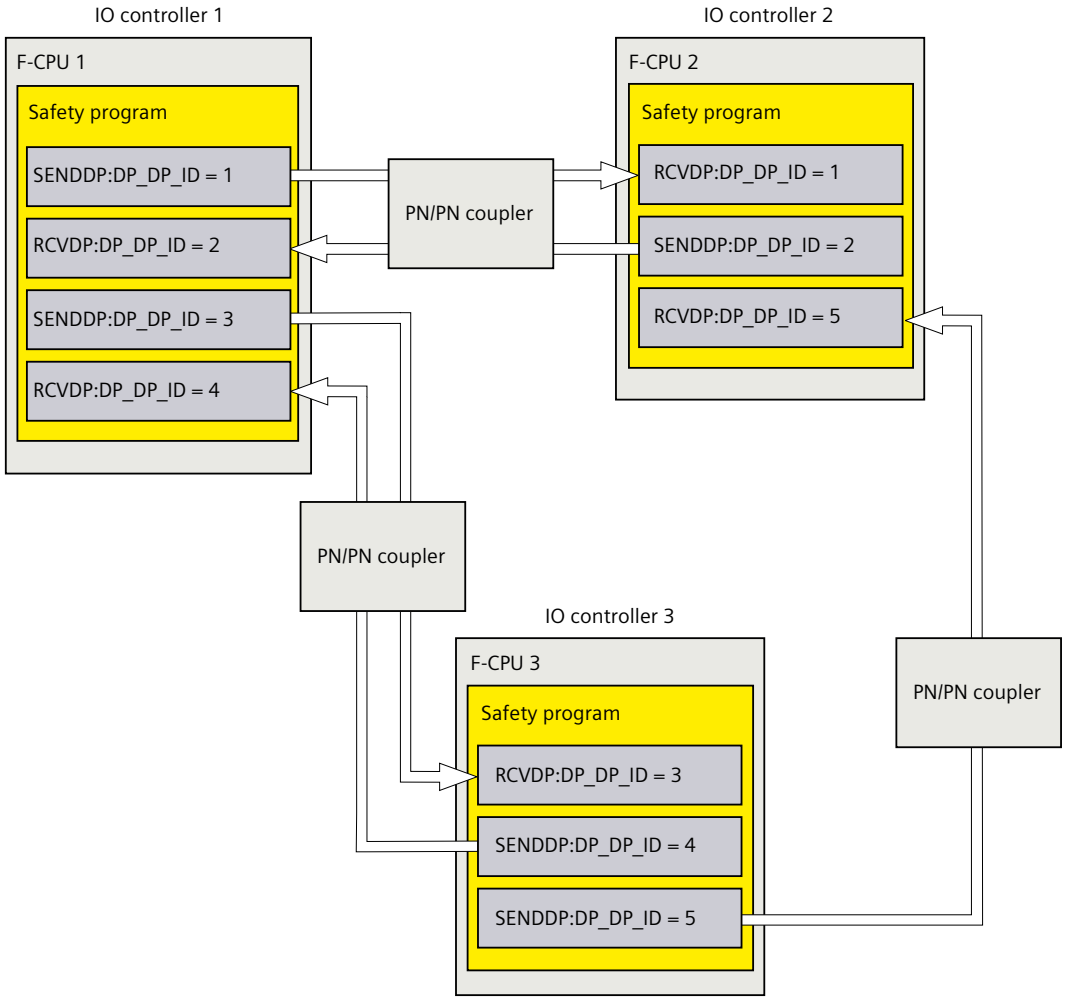
You program safety-related IO controller-IO controller communication as follows:

1. In the safety program from which data is to be sent, call the SENDDP instruction (Page 612) for sending at the end of the main safety block.
2. In the safety program in which data is to be received, call the RCVDP instruction (Page 612) for receiving at the start of the main safety block.
3. Assign the start addresses of the output and input data transfer areas of the PN/PN coupler configured in the *hardware and network editor* to the respective LADDR inputs.

   You must carry out this assignment for every communication connection for each of the F-CPUs involved.

4. Assign the value for the respective F-communication ID to the DP_DP_ID inputs. This establishes the communication relationship between the SENDDP instruction in one F-CPU and the RCVDP instruction in the other F-CPU: The associated instructions receive the same value for DP_DP_ID.

   The figure below contains an example of how to specify the F-communication IDs at the inputs of the SENDDP and RCVDP instructions for 5 safety-related IO controller-IO controller communication relationships.



WARNING

The value for the respective F-communication ID (input DP_DP_ID; data type: INT) can be freely selected; however, it must be unique network-wide* and CPU-wide at all times for all safety-related communication connections. You must check the uniqueness in the safety summary during acceptance of the safety program.

You must supply constant values to the DP_DP_ID and LADDR inputs when calling the instruction. Direct write access to DP_DP_ID and LADDR in the associated instance DB is not permitted in the safety program! *(S016)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

5. Supply the SD_BO_xx and SD_I_xx inputs of SENDDP with the send signals. To cut down on intermediate signals when transferring block parameters, you can write the value directly to the instance DB of SENDDP using fully qualified access (for example, "Name SENDDP_1".SD_BO_02) before calling SENDDP.
6. Supply the RD_BO_xx and RD_I_xx outputs of RCVDP with the signals that you want to process further in other program sections or use fully qualified access to read the received signals directly in the associated instance DB in the program sections to be processed further (e.g., "Name RCVDP_1".RD_BO_02).
7. Supply the SUBBO_xx and SUBI_xx inputs of RCVDP with the fail-safe values that are to be output by RCVDP in place of the process data until communication is established for the first time after startup of the sending and receiving F-systems or in the event of an error in safety-related communication.
   - Specification of constant fail-safe values:

     For data of data type INT, you can enter constant fail-safe values directly as constants in the SUBI_xx input (initial value = "0"). If you want to specify a constant fail-safe value "TRUE" for data of the data type BOOL, provide the tag "F_GOBDB".VKE1 for the SUBBO_xx input (initial value = "FALSE").
   - Specification of variable substitute values:

     If you want to specify variable substitute values, define a tag that you calculate through your safety program in an F-DB and specify this tag (fully qualified) in the SUBI_xx or SUBBO_xx input.

     ⚠ WARNING

     Note: The program logic for calculating variable substitute values can only be inserted after the RCVDP calls, because there must be no program logic before the RCVDP calls. This is why the initial values of the substitute values are active in all RCVDP instructions in the first cycle after a startup of the F-system. You must therefore assign appropriate initial values for these tags. *(S017)*
8. Configure the TIMEOUT inputs of the RCVDP and SENDDP instructions with the required monitoring time.

   ⚠ WARNING

   It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time. *(S018)*

   Information on calculating the monitoring times can be found in Monitoring and response times (Page 629).

9. Optional: Evaluate the ACK_REQ output of the RCVDP instruction, for example, in the standard user program or on the operator control and monitoring system in order to query or to indicate whether user acknowledgment is required.
10. Supply the ACK_REI input of the RCVDP instruction with the acknowledgment signal for reintegration.
11. Optional: Evaluate the SUBS_ON output of the RCVDP or SENDDP instruction in order to query whether the RCVDP instruction is outputting the fail-safe values assigned in the SUBBO_xx and SUBI_xx inputs.
12. Optional: Evaluate the ERROR output of the RCVDP or SENDDP instruction, for example, in the standard user program or on the operator control and monitoring system in order to query or to indicate whether a communication error has occurred.
13. Optional: Evaluate the SENDMODE output of the RCVDP instruction in order to query whether the F-CPU with the associated SENDDP instruction is in disabled safety mode (Page 393).

9.1.2.3 Safety-related IO controller-IO controller communication - Limits for data transfer

NOTE

If the data quantities to be transmitted exceed the capacity of the SENDDP / RCVDP correlated instructions, a second (or third) SENDDP / RCVDP call can be used. This requires configuration of an additional communication connection via the PN/PN coupler. Whether or not this is possible with one single PN/PN coupler depends on the capacity restrictions of the PN/PN coupler.

9.1.3 Safety-related master-master communication

9.1.3.1 Configure safety-related master-master communication

Introduction

Safety-related communication between safety programs of the F-CPUs of DP masters takes place via a DP/DP coupler.

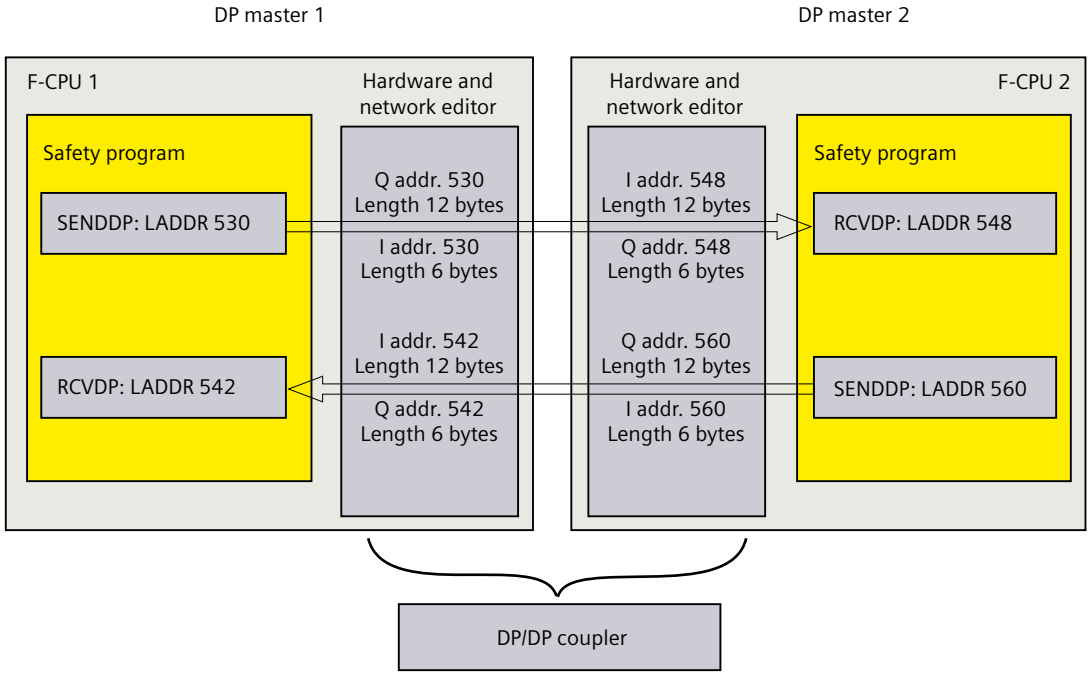
NOTE

Switch the data validity indicator "DIA" on the DIP switch of the DP/DP coupler to "OFF". Otherwise, safety-related CPU-CPU communication is not possible.

Configuring transfer areas

You must configure one transfer area for output data and one transfer area for input data in the *hardware and network editor* for each safety-related communication connection between two F-CPUs in the DP/DP coupler. The figure below shows how both of the F-CPUs are able to send and receive data (bidirectional communication). One transfer area for output

data and one transfer area for input data must be configured in the DP/DP coupler for each of the two communication connections.



Rules for defining transfer areas

The transfer area for input data and the transfer area for output data for the **data to be sent** must begin with the same start address. A total of 6 bytes (consistent) is required for the transfer area for input data; 12 bytes (consistent) are required for the transfer area for output data.

The transfer area for input data and the transfer area for output data for the **data to be received** must begin with the same start address. A total of 12 bytes (consistent) is required for the transfer area for input data; 6 bytes (consistent) are required for the transfer area for output data.

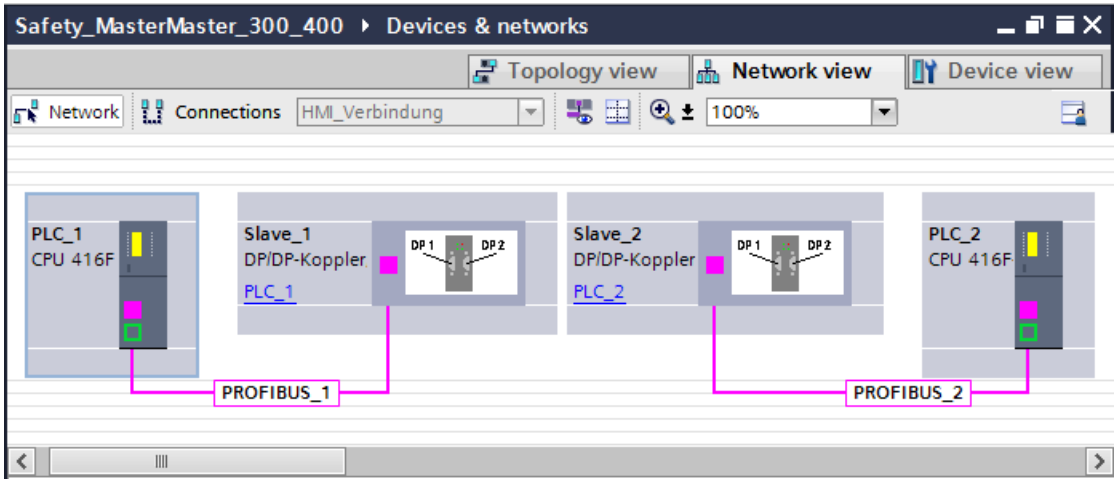
Procedure for configuration

The procedure for configuring safety-related master-master communication is identical to that in the standard system.

Proceed as follows:

1. Insert two F-CPUS from the "Hardware catalog" task card into the project.
2. Switch to the network view of the *hardware and network editor*.
3. Select a DP/DP coupler from "Other field devices\PROFIBUS DP\Gateways\Siemens AG\DP/DP Coupler" in the "Hardware catalog" task card and insert it into the network view of the hardware and network editor.
4. Insert a second DP/DP coupler.

5. Connect a DP interface of F-CPU 1 to the DP interface of a DP/DP coupler and a DP interface of F-CPU 2 to the DP interface of the other DP/DP coupler.



6. A free PROFIBUS address is assigned automatically in the properties of the DP/DP-coupler in the device view. You must set this address on the DP/DP coupler of PLC 1, either by using the DIP switch on the device or in the configuration of the DP/DP coupler (see DP/DP Coupler (https://support.industry.siemens.com/cs/ww/en/view/1179382) manual).
7. Switch to the device view of the DP/DP coupler for PLC1 for bidirectional communication connections i.e. where each F-CPU is both to send and to receive data. Select the following modules from the "Hardware catalog" task card (with filter activated), and insert them in the "Device overview" tab:
   - One "6 bytes I/12 bytes Q consistent" module, and
   - One "12 bytes I/6 bytes Q consistent" module
8. In the properties of the modules, assign the addresses outside the process image as follows:

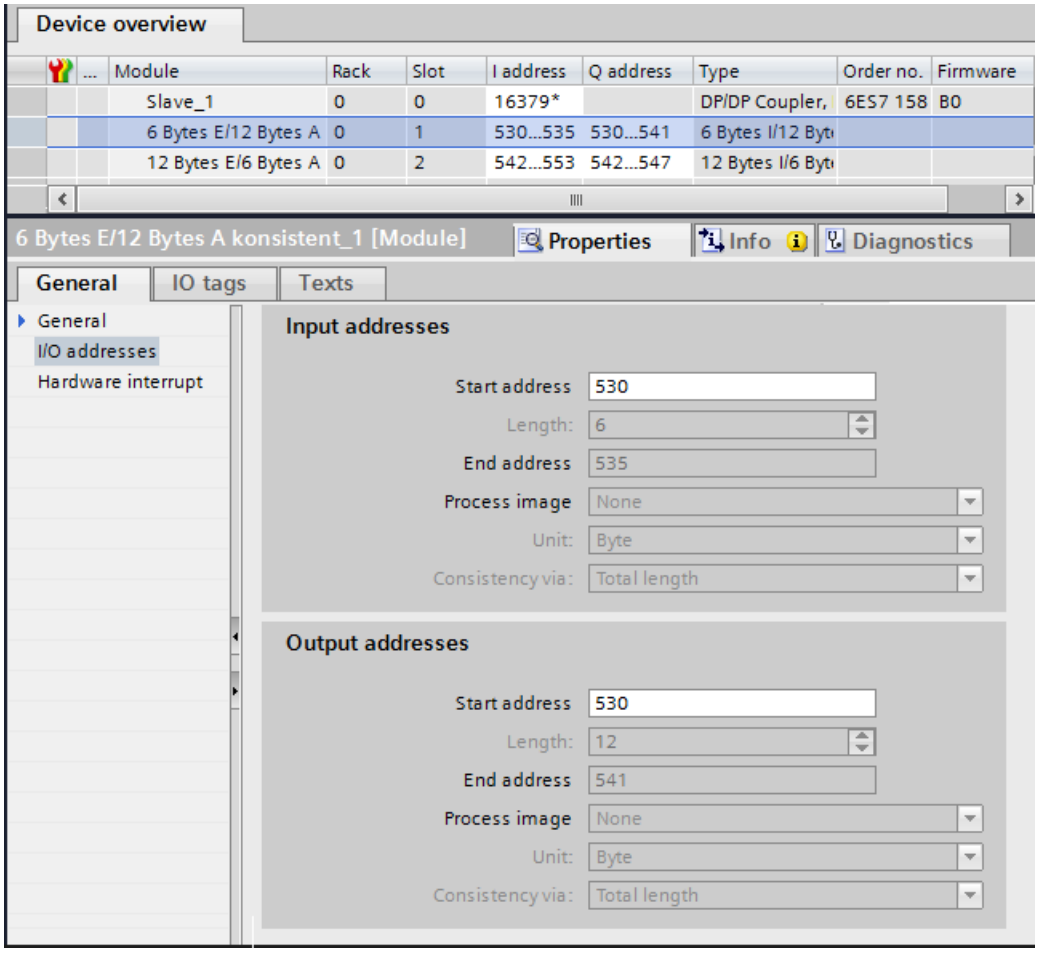
   For "6 bytes I/12 bytes Q consistent" module for sending data for example:
   - Input addresses: Start address 530
   - Output addresses: Start address 530

   For "12 bytes I/6 bytes Q consistent" module for receiving data for example:
   - Input addresses: Start address 542
   - Output addresses: Start address 542

   NOTE

   Make sure that you assign identical start addresses for the address areas of the output and input data.

   Tip: Make a note of the start addresses of the transfer areas. You need these to program the SENDDP and RCVDP blocks (LADDR input).



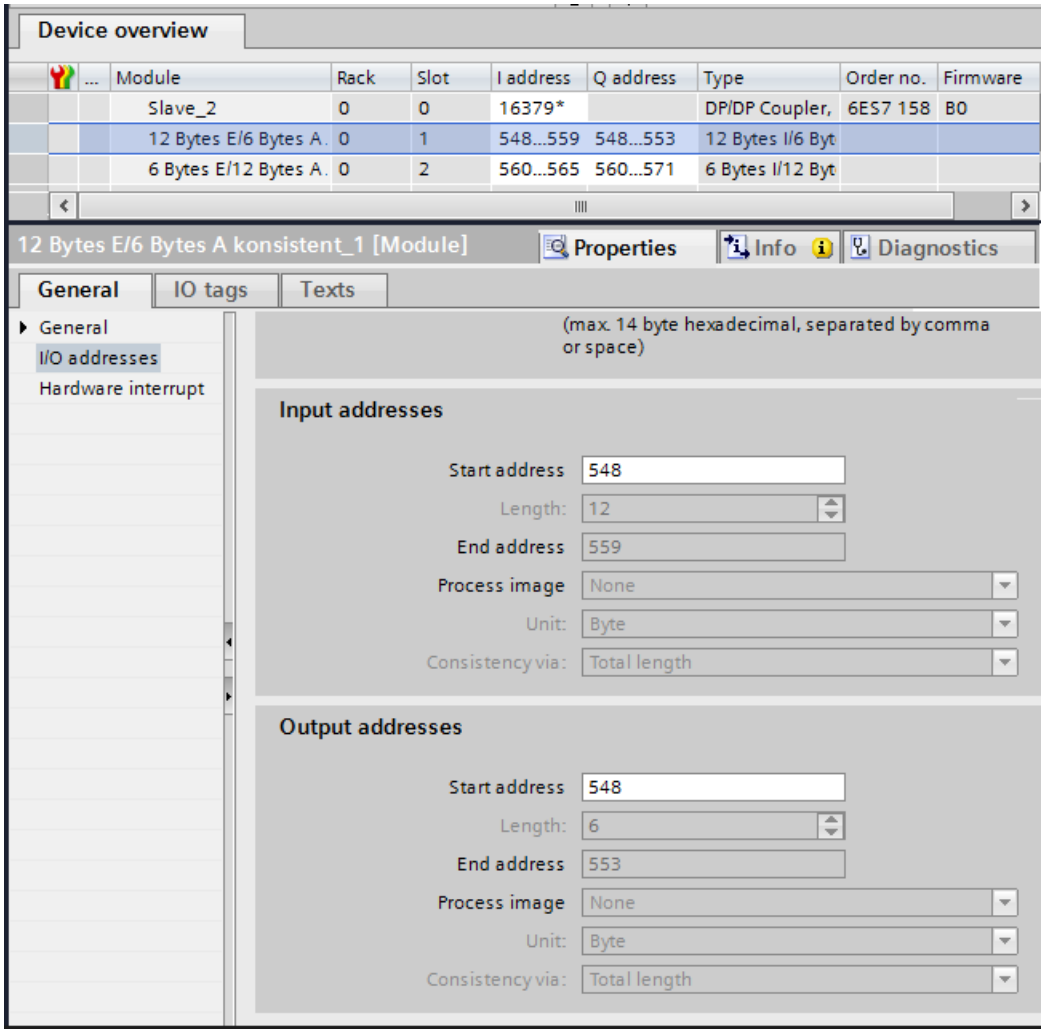
9. Select the following modules from the "Hardware catalog" task card (with filter activated) in the device view of DP/DP coupler PLC2, and insert them in the "Device overview" tab:
   - One "12 bytes I/6 bytes Q consistent" module, and
   - One "6 bytes I/12 bytes Q consistent" module
10. In the properties of the modules, assign the addresses outside the process image as follows:

    For "12 bytes I/6 bytes Q consistent" module for receiving data for example:

    - Input addresses: Start address 548
    - Output addresses: Start address 548

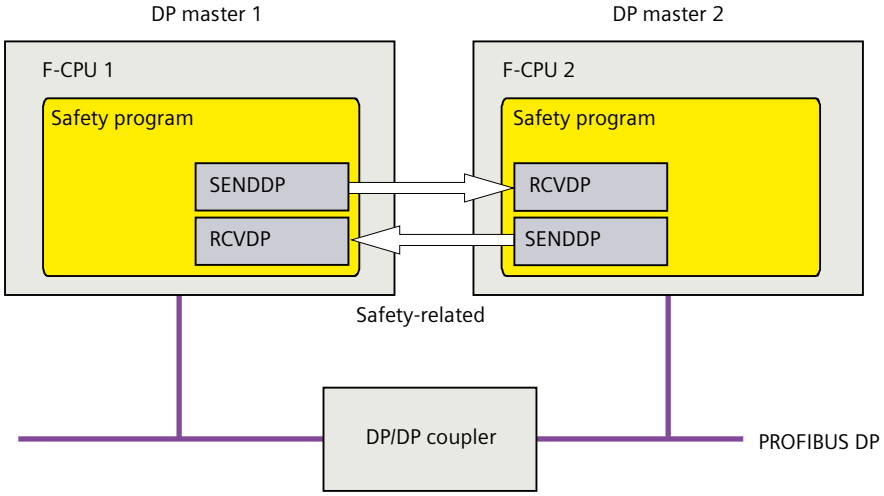
    For "6 bytes I/12 bytes Q consistent" module for sending data for example:

    - Input addresses: Start address 560
    - Output addresses: Start address 560


Device overview
... Module Rack Slot I address Q address Type Order no. Firmware
Slave_2 0 0 16379* DP/DP Coupler, 6ES7 158 B0
12 Bytes E/6 Bytes A. 0 1 548...559 548...553 12 Bytes I/6 Byt
6 Bytes E/12 Bytes A. 0 2 560...565 560...571 6 Bytes I/12 Byt
12 Bytes E/6 Bytes A konsistent_1 [Module]
Properties
Info
Diagnostics
General
IO tags
Texts
General
I/O addresses
Hardware interrupt
(max. 14 byte hexadecimal, separated by comma or space)
Input addresses
Start address 548
Length: 12
End address 559
Process image None
Unit: Byte
Consistency via: Total length
Output addresses
Start address 548
Length: 6
End address 553
Process image None
Unit: Byte
Consistency via: Total length

9.1.3.2 Program safety-related master-master communication

Communication via SENDDP and RCVDP instructions



Safety-related communication between the F-CPUs of the DP master uses the SENDDP and RCVDP instructions for sending and receiving, respectively. These can be used to perform a fail-safe transfer of a *fixed* amount of fail-safe data of the data type INT or BOOL.

You can find these instructions in the "Instructions" task card under "Communication". The RCVDP instruction **must** be called at the start of the main safety block. The SENDDP instruction **must** be called at the end of the main safety block.

Note that the send signals are not sent until after the SENDDP instruction call at the end of execution of the relevant F-runtime group.

You can find a detailed description of the SENDDP and RCVDP instructions in SENDDP and RCVDP: Send and receive data via PROFIBUS DP/PROFINET IO (STEP 7 Safety V21) (Page 612).

Requirement for programming

The transfer areas for input and output data for the DP/DP coupler must be configured.

Programming procedure

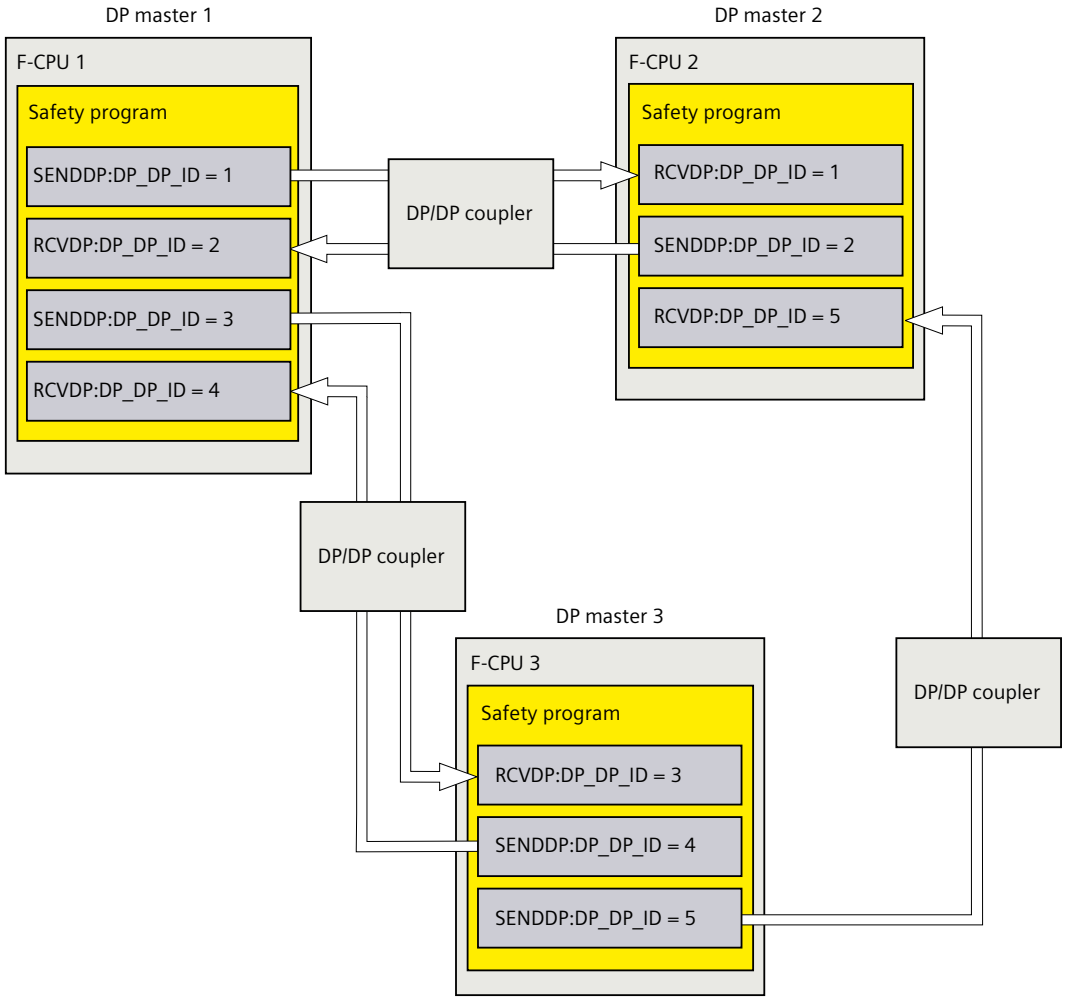
You program safety-related master-master communication as follows:

1. In the safety program from which data is to be sent, call the SENDDP instruction (Page 612) for sending at the end of the main safety block.
2. In the safety program in which data is to be received, call the RCVDP instruction (Page 612) for receiving at the start of the main safety block.
3. Assign the start addresses of the transfer areas for output and input data of the DP/DP coupler configured in the *hardware and network editor* to the respective LADDR inputs.

   You must carry out this assignment for every communication connection for each of the F-CPUs involved.

4. Assign the value for the respective F-communication ID to the DP_DP_ID inputs. This establishes the communication relationship between the SENDDP instruction in one F-CPU and the RCVDP instruction in the other F-CPU: The associated instructions receive the same value for DP_DP_ID.

   The figure below contains an example of how to specify the F-communication IDs at the inputs of the SENDDP and RCVDP instructions for 5 safety-related master-master communications relationships.



WARNING

The value for the respective F-communication ID (input DP_DP_ID; data type: INT) can be freely selected; however, it must be unique network-wide* and CPU-wide at all times for all safety-related communication connections. You must check the uniqueness in the safety summary during acceptance of the safety program.

You must supply constant values to the DP_DP_ID and LADDR inputs when calling the instruction. Direct write access to DP_DP_ID and LADDR in the associated instance DB is not permitted in the safety program! *(S016)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

5. Supply the SD_BO_xx and SD_I_xx inputs of SENDDP with the send signals. To cut down on intermediate signals when transferring block parameters, you can write the value directly to the instance DB of SENDDP using fully qualified access (for example, "Name SENDDP_1".SD_BO_02) before calling SENDDP.
6. Supply the RD_BO_xx and RD_I_xx outputs of RCVDP with the signals that you want to process further in other program sections or use fully qualified access to read the received signals directly in the associated instance DB in the program sections to be processed further (e.g. "Name RCVDP_1".RD_BO_02).
7. Supply the SUBBO_xx and SUBI_xx inputs of RCVDP with the fail-safe values that are to be output by RCVDP in place of the process data until communication is established for the first time after startup of the sending and receiving F-systems or in the event of an error in safety-related communication.
   - Specification of constant fail-safe values:

     For data of data type INT, you can enter constant fail-safe values directly as constants in the SUBI_xx input (initial value = "0"). If you want to specify a constant fail-safe value for data of the data type BOOL, provide the tag "F_GLOBDB".VKE1 for the SUBBO_xx input (initial value = "FALSE").
   - Specification of variable substitute values:

     If you want to specify variable substitute values, define a tag that you calculate through your safety program in an F-DB and specify this tag (fully qualified) in the SUBI_xx or SUBBO_xx input.

     ⚠ WARNING

     Note: The program logic for calculating variable substitute values can only be inserted after the RCVDP calls, because there must be no program logic before the RCVDP calls. This is why the initial values of the substitute values are active in all RCVDP instructions in the first cycle after a startup of the F-system. You must therefore assign appropriate initial values for these tags. *(S017)*
8. Configure the TIMEOUT inputs of the RCVDP and SENDDP instructions with the required monitoring time.

   ⚠ WARNING

   It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time. *(S018)*

   Information on calculating the monitoring times can be found in Monitoring and response times (Page 629).

9. Optional: Evaluate the ACK_REQ output of the RCVDP instruction, for example, in the standard user program or on the operator control and monitoring system in order to query or to indicate whether user acknowledgment is required.
10. Supply the ACK_REI input of the RCVDP instruction with the acknowledgment signal for reintegration.
11. Optional: Evaluate the SUBS_ON output of the RCVDP or SENDDP instruction in order to query whether the RCVDP instruction is outputting the fail-safe values assigned in the SUBBO_xx and SUBI_xx inputs.
12. Optional: Evaluate the ERROR output of the RCVDP or SENDDP instruction, for example, in the standard user program or on the operator control and monitoring system in order to query or to indicate whether a communication error has occurred.
13. Optional: Evaluate the SENDMODE output of the RCVDP instruction in order to query whether the F-CPU with the associated SENDDP instruction is in disabled safety mode (Page 393).

9.1.3.3 Safety-related master-master communication:Limits for data transfer

NOTE

If the data quantities to be transmitted exceed the capacity of the SENDDP / RCVDP correlated instructions, a second (or third) SENDDP / RCVDP call can be used. This requires configuration of an additional connection via the DP/DP coupler. Whether or not this is possible with one single DP/DP coupler depends on the capacity restrictions of the DP/DP coupler.

9.1.4 Safety-related IO controller-I-device communication

9.1.4.1 Configuring safety-related communication between IO controller and I-device

Introduction

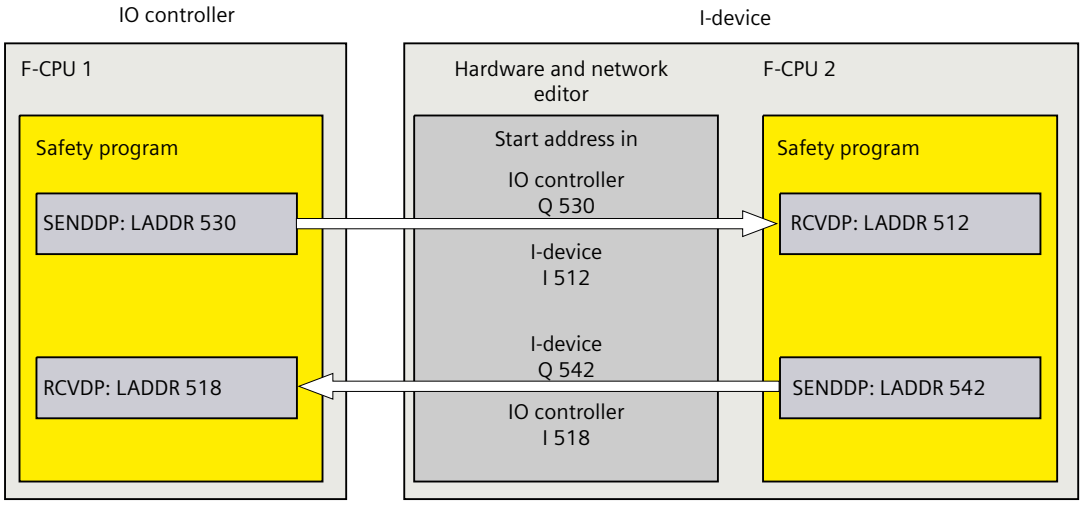
Safety-related communication between the safety program of the F-CPU of an IO controller and the safety program(s) of the F-CPU(s) of one or more I-devices takes place via IO controller-I-device transfer areas (**F**-CD) in PROFINET IO, as in standard systems.

You do not need any additional hardware for IO controller-I-device communication.

The F-CPU to be used as an I-device must support the "IO-device" operating mode.

Configuring transfer areas

For every safety-related communication connection between two F-CPUs, you must configure transfer areas in the *hardware and network editor*. The figure below shows how both of the F-CPUs are able to send and receive data (bidirectional communication).



The transfer area is assigned a label when it is created to identify it as the communication relationship. For example, "F-CD_PLC_2 PLC_1_1" for the first F-CD connection between IO controller F-CPU 1 and I-device F-CPU 2.

You assign the start addresses of the transfer areas to the LADDR input of the SENDDP and RCVDP instructions in the safety programs.

Procedure for configuration

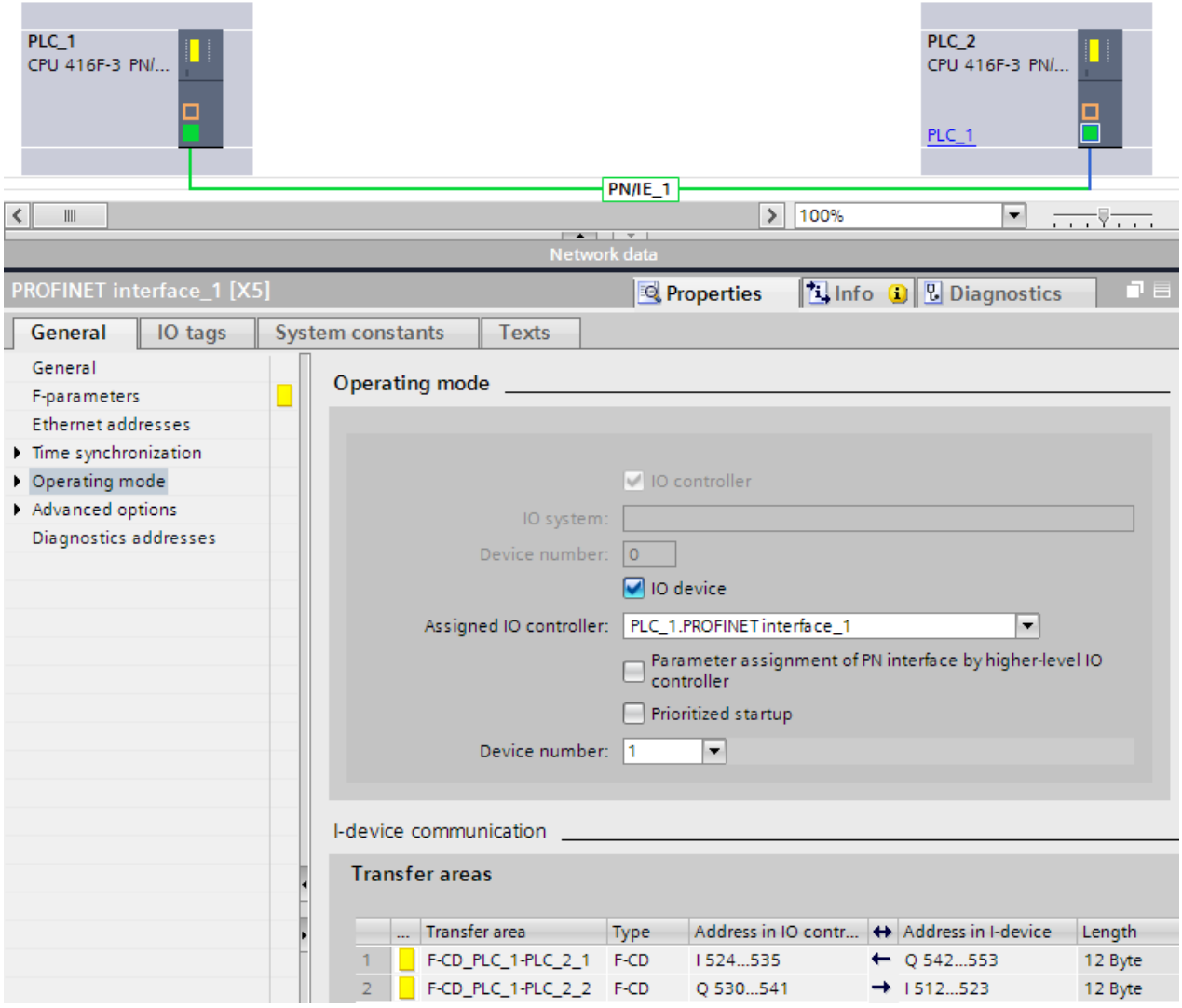
The procedure for configuring safety-related IO controller-I-device communication is identical to that in the standard system.

Proceed as follows:

1. Insert two F-CPUS from the "Hardware catalog" task card into the project.
2. Enable the "IO Device" mode for F-CPU 2 in the properties of its PN interface and assign this PN interface to a PN interface of F-CPU 1.
3. Select the PROFINET interface of F-CPU 2. Under "Transfer areas", you create an F-CD connection (type "F-CD") for sending to the IO controller (←). The F-CD connection is shown in yellow in the table and the address areas in the I-device and IO controller assigned outside of the process image are displayed.

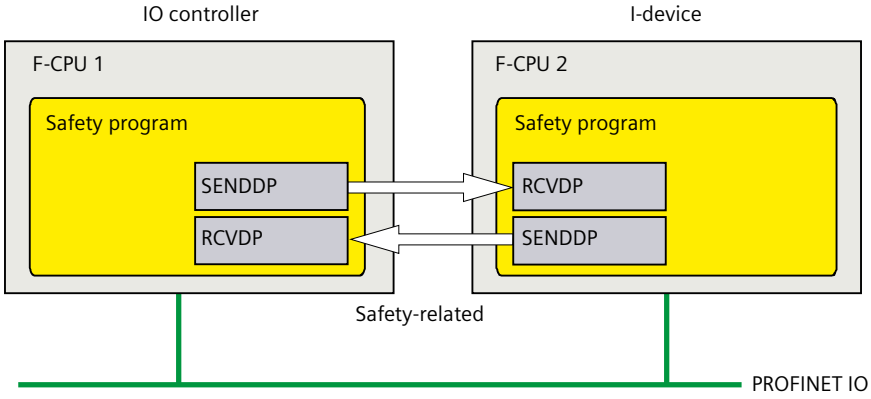
   In addition, an acknowledgment connection is created automatically for each F-CD connection. (see "Transfer area details").
4. Create an additional F-CD connection for receiving from the IO controller.

5. In the transfer area you just created, click the arrow in order to change the transfer direction to receiving from IO controller (→).



9.1.4.2 Programming safety-related IO controller I-device communication

Communication via SENDDP and RCVDP instructions



Safety-related communication between the F-CPUs of the IO controller and an I-device makes use of the SENDDP and RCVDP instructions for sending and receiving, respectively. These can be used to perform a fail-safe transfer of a *fixed* amount of fail-safe data of the data type INT or BOOL.

You can find these instructions in the "Instructions" task card under "Communication". The RCVDP instruction **must** be called at the start of the main safety block. The SENDDP instruction **must** be called at the end of the main safety block.

Note that the send signals are not sent until after the SENDDP instruction call at the end of execution of the relevant F-runtime group.

You can find a detailed description of the SENDDP and RCVDP instructions in SENDDP and RCVDP: Send and receive data via PROFIBUS DP/PROFINET IO (STEP 7 Safety V21) (Page 612).

Requirement for programming

The transfer areas must be configured.

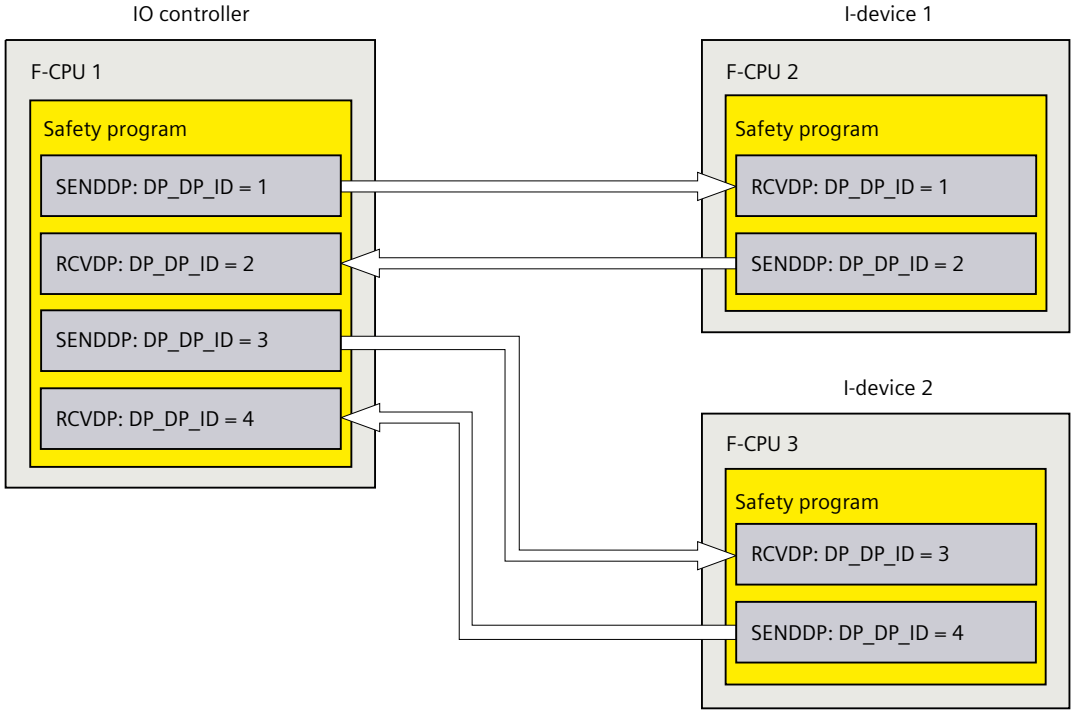
Programming procedure

The procedure for programming safety-related IO controller-I-device communication is the same as that for programming safety-related IO controller-IO controller communication (see Program safety-related IO controller-IO controller communication (Page 209)).

The assignment of the start addresses of the transfer areas to the LADDR input of the SENDDP/RCVDP instructions can be obtained from the following table.

Instruction	Start address LADDR	
	From row	From column
SENDDP in the IO controller	→	Address in the IO controller
RCVDP in the IO controller	←	Address in the IO controller
SENDDP in the I-device	←	Address in the IO device
RCVDP in the I-device	→	Address in the IO device

The figure below contains an example of how to specify the F-communication IDs for the inputs of the SENDDP and RCVDP instructions for 4 safety-related IO controller-I-device communication relationships.



⚠ WARNING

The value for the respective F-communication ID (input DP_DP_ID; data type: INT) can be freely selected; however, it must be unique network-wide* and CPU-wide at all times for all safety-related communication connections. You must check the uniqueness in the safety summary during acceptance of the safety program.

You must supply constant values to the DP_DP_ID and LADDR inputs when calling the instruction. Direct write access to DP_DP_ID and LADDR in the associated instance DB is not permitted in the safety program! *(S016)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In

PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

⚠ WARNING

It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time.*(S018)*

Information on calculating the monitoring times can be found in Monitoring and response times (Page 629).

9.1.4.3 Safety-related IO-Controller-IO-Device communication - Limits for data transfer

Limits for data transfer

If the amount of data to be transferred is greater than the capacity of related SENDDP/RCVDP instructions, you can use additional SENDDP/RCVDP instructions. Configure additional transfer areas for this purpose. Remember the maximum limit of 1440 bytes of input data or 1440 bytes of output data for transfer between an I-device and a IO controller.

The following table shows the amount of output and input data assigned in safety-related communication connections:

Safety-related communication	Communication connection	Assigned input and output data			
		In the IO controller		In the I-device	
		Output data	Input data	Output data	Input data
IO controller-I-Device	**Sending:** I-Device 1 to IO controller	6 bytes	12 bytes	12 bytes	6 bytes
	Receiving: I-Device 1 from IO controller	12 bytes	6 bytes	6 bytes	12 bytes

Consider all additional configured safety-related and standard communication connections (transfer areas of type F-CD and CD) for the maximum limit of 1440 bytes of input data or 1440 bytes of output data for transfer between an I-device and an IO controller. In addition, data are assigned for internal purposes such that the maximum limit may be reached sooner.

When the limit is exceeded, a corresponding error message is displayed.

9.1.5 Safety-related master-I-slave communication

9.1.5.1 Configuring safety-related master-I-slave communication

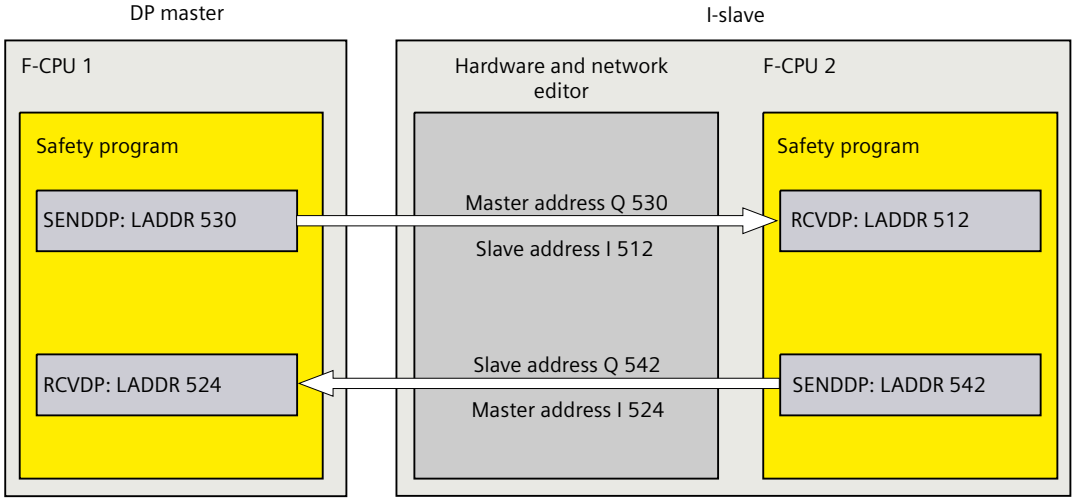
Introduction

Safety-related communication between the safety program of the F-CPU of a DP master and the safety program(s) of the F-CPU(s) of one or more I-slaves takes place via master-I-slave transfer areas (**F-**MS), as in standard systems.

You do not need a DP/DP coupler for master-I-slave communication.

Configuring transfer areas

For every safety-related communication connection between two F-CPUs, you must configure transfer areas in the *hardware and network editor*. The figure below shows how both of the F-CPUs are able to send and receive data (bidirectional communication).



The transfer area is assigned a label when it is created to identify it as the communication relationship. For example, "F-MS_PLC_2-PLC_1_1" for the first F-MS connection between DP master F-CPU 1 and I-slave F-CPU 2.

You assign the start addresses of the transfer areas to the LADDR input of the SENDDP and RCVDP instructions in the safety programs.

Procedure for configuration

The procedure for configuring safety-related master-I-slave communication is identical to that in the standard system.

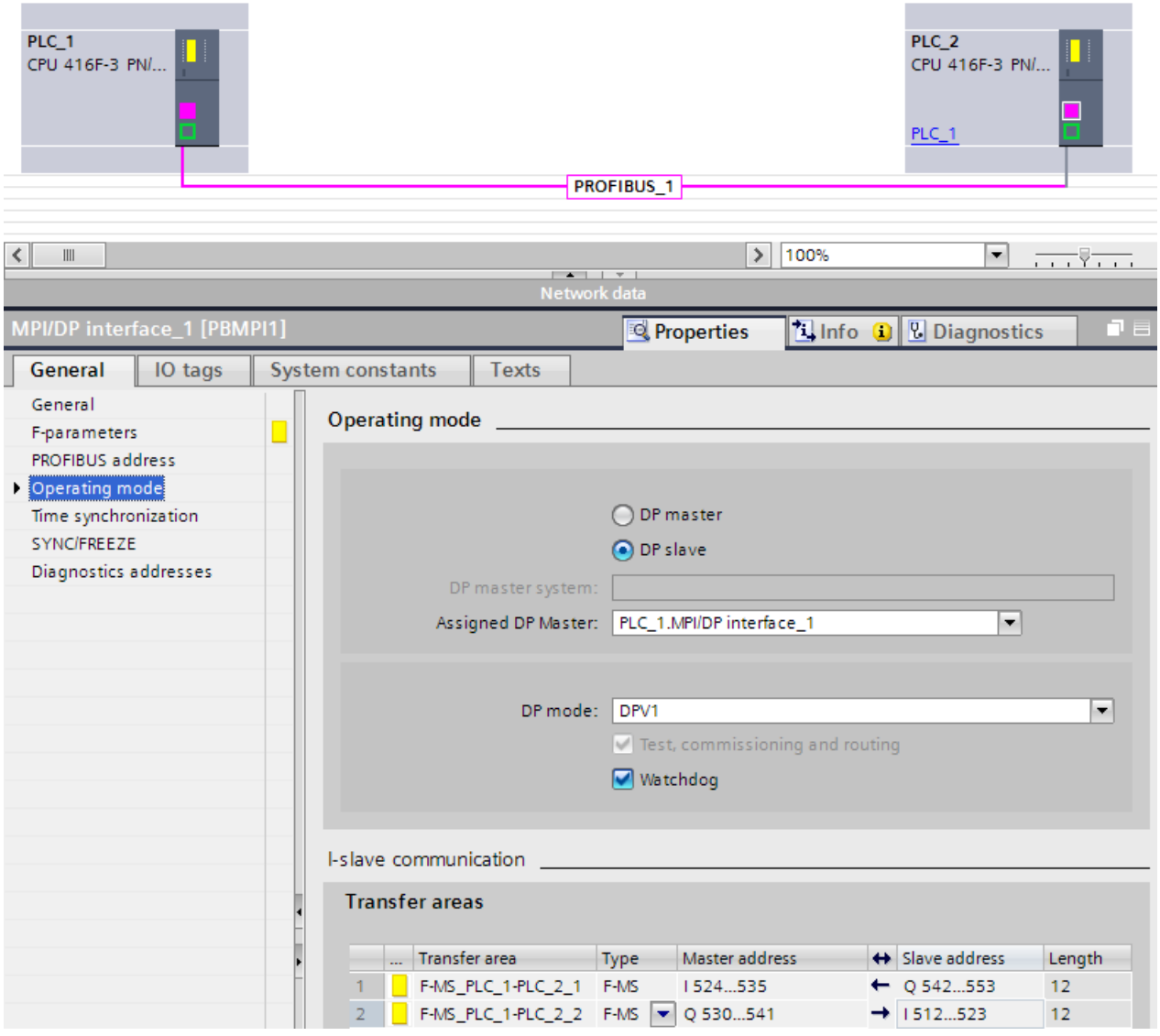
Proceed as follows:

1. Insert two F-CPUS from the "Hardware catalog" task card into the project.
2. Activate the "DP slave" mode (I-slave) for F-CPU 2 in the properties of its DP interface and assign this DP interface to a DP interface of F-CPU 1.

3. Select the PROFIBUS interface of F-CPU 2. Under "Transfer areas", you create an F-MS connection (type "F-MS") for sending to the DP master (←). The F-MS connection is shown in yellow in the table and the transfer areas in the I-slave and DP master assigned outside of the process image are displayed.

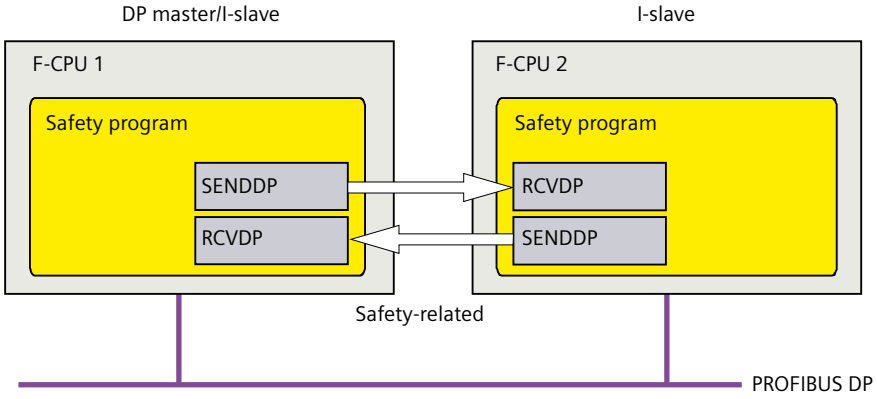
   In addition, an acknowledgment connection is created automatically for each F-MS connection. (see "Transfer area details").

4. Create an additional F-MS connection for receiving from the DP master.
5. In the transfer area you just created, click the arrow in order to change the transfer direction to receiving from DP master (→).



9.1.5.2 Program the safety-related master-I-slave or I-slave-I-slave communication

Communication via SENDDP and RCVDP instructions



Safety-related communication between the F-CPUs of the DP master and an I-slave or between the F-CPUs of multiple I-slaves makes use of the SENDDP and RCVDP instructions for sending and receiving, respectively. These can be used to perform a fail-safe transfer of a *fixed* amount of fail-safe data of the data type INT or BOOL.

You can find these instructions in the "Instructions" task card under "Communication". The RCVDP instruction **must** be called at the start of the main safety block. The SENDDP instruction **must** be called at the end of the main safety block.

Note that the send signals are not sent until after the SENDDP instruction call at the end of execution of the relevant F-runtime group.

You can find a detailed description of the SENDDP and RCVDP instructions in SENDDP and RCVDP: Send and receive data via PROFIBUS DP/PROFINET IO (STEP 7 Safety V21) (Page 612).

Requirement for configuration

The transfer areas must be configured.

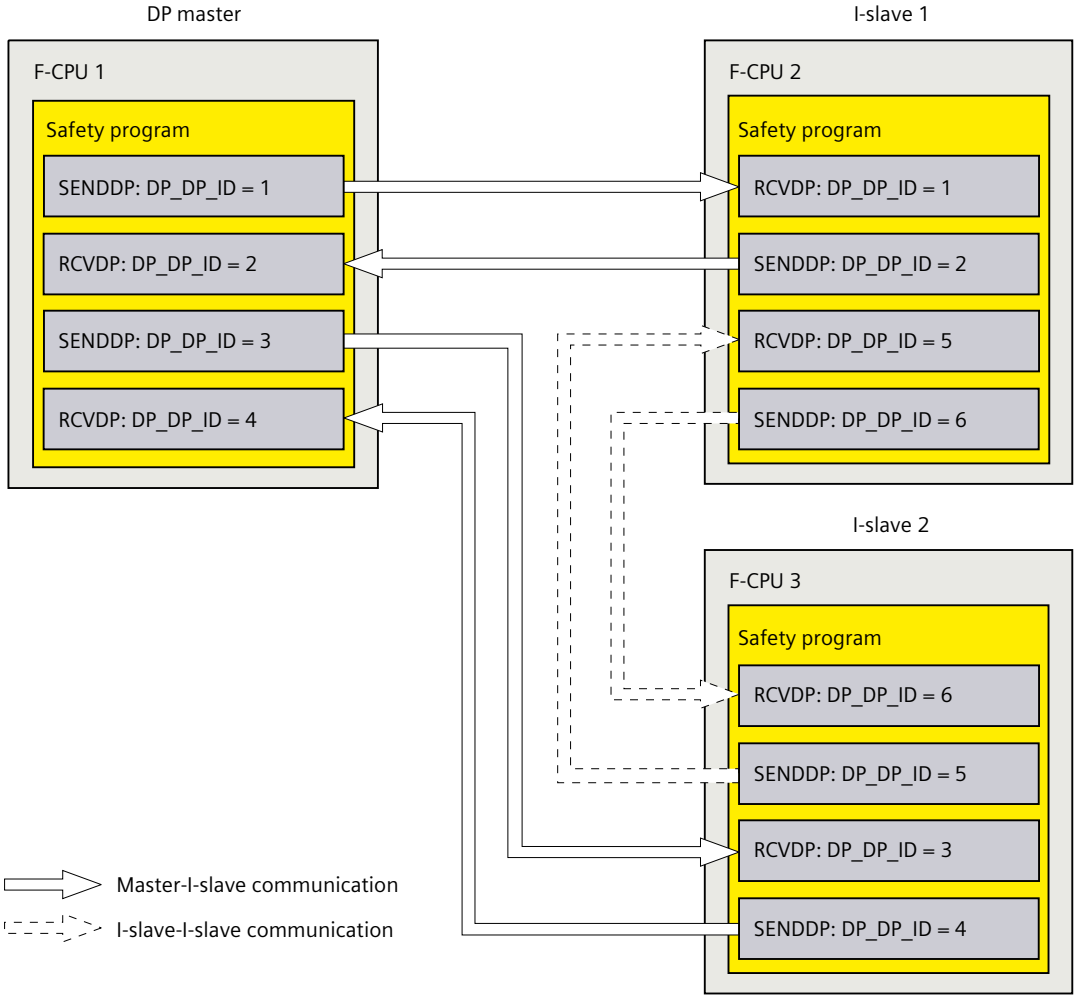
Programming procedure

The procedure for programming safety-related master-I-slave communication or I-slave-I-slave communication is the same as that for programming safety-related master-master communication (see Program safety-related master-master communication (Page 217)).

The assignment of the start addresses of the transfer areas to the LADDR input of the SENDDP/RCVDP instructions can be obtained from the following table.

Instruction	Start address LADDR	
	From row	From column
SENDDP in the DP master	→	Master address
RCVDP in the DP master	←	Master address
SENDDP in the I-slave	←	Slave address
RCVDP in the I-slave	→	Slave address

The figure below contains an example of how to specify the F-communication IDs at the inputs of SENDDP and RCVDP instructions for four safety-related master-I-slave and two I-slave-I-slave communications relationships.



⚠ WARNING

The value for the respective F-communication ID (input DP_DP_ID; data type: INT) can be freely selected; however, it must be unique network-wide* and CPU-wide at all times for all safety-related communication connections. You must check the uniqueness in the safety summary during acceptance of the safety program.

You must supply constant values to the DP_DP_ID and LADDR inputs when calling the instruction. Direct write access to DP_DP_ID and LADDR in the associated instance DB is not permitted in the safety program! *(S016)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

⚠ WARNING

It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time. *(S018)*

Information on calculating the monitoring times can be found in Monitoring and response times (Page 629).

9.1.5.3 Limits for data transfer of safety-related master-I-slave or I-slave-I-slave communication

Limits for data transfer

If the amount of data to be transferred is greater than the capacity of related SENDDP/RCVDP instructions, you can use additional SENDDP/RCVDP instructions. Configure additional transfer areas for this purpose. Note the maximum limit of 244 bytes of input data or 244 bytes of output data for transfer between an I-slave and a DP master.

The following table shows the amount of output and input data assigned in safety-related communication connections:

Safety-related communication	Communication connection	Assigned input and output data					
		DP master		I-slave 1		I-slave 2	
		Output data	Input data	Output data	Input data	Output data	Input data
Master-I-slave	**Sending:** I-slave 1 to DP master	6 bytes	12 bytes	12 bytes	6 bytes	—	—
	Receiving: I-slave 1 from DP master	12 bytes	6 bytes	6 bytes	12 bytes	—	—
I-slave-I-slave	**Sending:** I-slave 1 to I-slave 2	—	18 bytes	12 bytes	6 bytes	6 bytes	12 bytes
	Receiving: I-slave 1 from I-slave 2	—	18 bytes	6 bytes	12 bytes	12 bytes	6 bytes

Consider all additional configured safety-related and standard communication connections (transfer areas of type F-MS-, F-DX-, F-DX-Mod., MS-, DX- and DX-Mod) for the maximum limit of 244 bytes of input data or 244 bytes of output data for transfer between an I-device and a DP master F-MS, F-DX, F-DX-Mod., MS, DX). If the maximum limit of 244 bytes of input data or 244 bytes of output data is exceeded, you will receive a corresponding error message.

9.1.6 Safety-related I-slave-I-slave communication

9.1.6.1 Configure safety-related I-slave-I-slave communication

Introduction

Safety-related communication between the safety program of the F-CPUs of I-slaves takes place using direct data exchange (**F-**DX) – same as in standard programs.

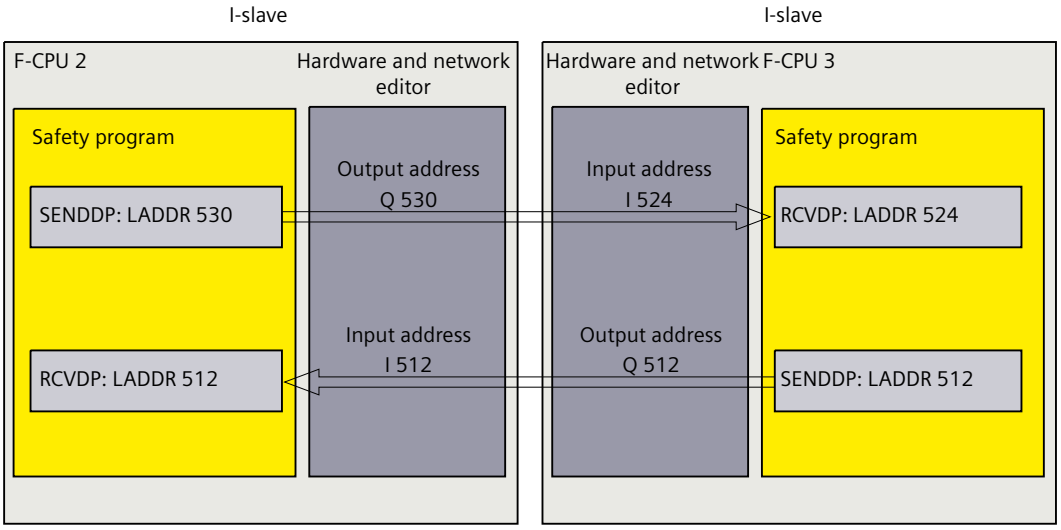
You do not need any additional hardware for I-slave-I-slave communication.

I-slave-I-slave communication is also possible:

- If the assigned DP master is a standard CPU that supports direct data exchange
- when instead of a DP master, an IO controller is networked with the I-slaves via an IE/PB link

Configuring transfer areas

For every safety-related communication connection between two I-slaves, you must configure transfer areas in the *hardware and network editor*. In the figure below, both of the I-slaves are to be able to send and receive data (bidirectional communication).



The transfer area is assigned a label when it is created to identify it as the communication relationship. For example, "F-DX_PLC_2-PLC_1_1" for the first F-DX connection between F-CPU 1 and F-CPU 2.

You assign the start addresses of the transfer areas to the LADDR input of the SENDDP and RCVDP instructions in the safety programs.

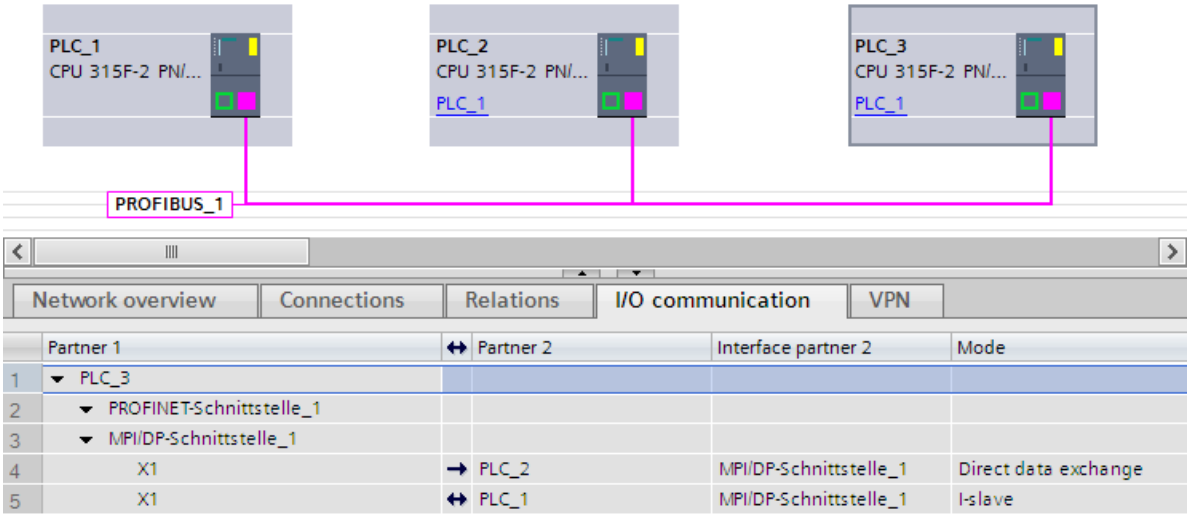
Procedure for configuration

The procedure for configuring safety-related I-slave-I-slave communication is identical to that in the standard system. Proceed as follows:

1. Insert three F-CPUS from the "Hardware catalog" task card in the project.
2. Activate "DP slaves" mode (I-slave) for F-CPU 2 and F-CPU 3 in the properties of their DP interfaces and assign these DP interfaces to a DP interface of F-CPU 1.
3. Select the DP interface of F-CPU 3 in the network view.
4. Select the "I/O communication" tab.

5. Use a drag-and-drop operation in the network view to move F-CPU 2 to the "Partner 2" column on the "I/O-communication" tab.

   This creates a line with "Direct data exchange" mode for sending to the I-slave (F-CPU 2) (→).

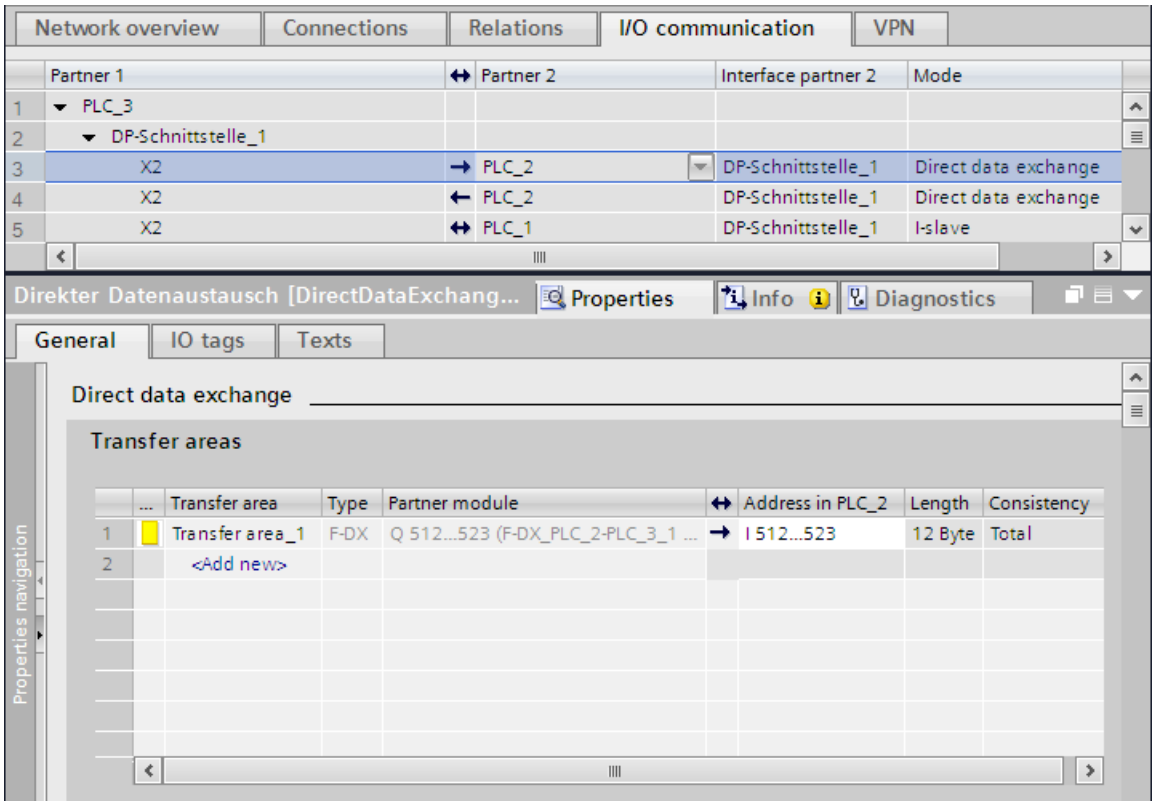


6. Click in the newly created line (→).
7. In "Transfer areas" ("Direct data exchange" table), create an F-DX connection (type "F-DX") for sending to the I-slave (F-CPU 2) (→). The F-DX connection is shown in yellow in the table and the transfer areas in the I-slaves assigned outside of the process image (PLC_2 and PLC_3) are displayed.

   In addition, a line with "Direct data exchange" mode for receiving from the I-slave (F-CPU 2) (←) is created automatically in the "I/O communication" tab, and an acknowledgment connection (→, transfer area x_Ack) is created automatically in the associated "Direct data exchange" table.

   One transfer area (type F-MS) for the master CPU is created in the "I-slave communication table" of each I-slave.

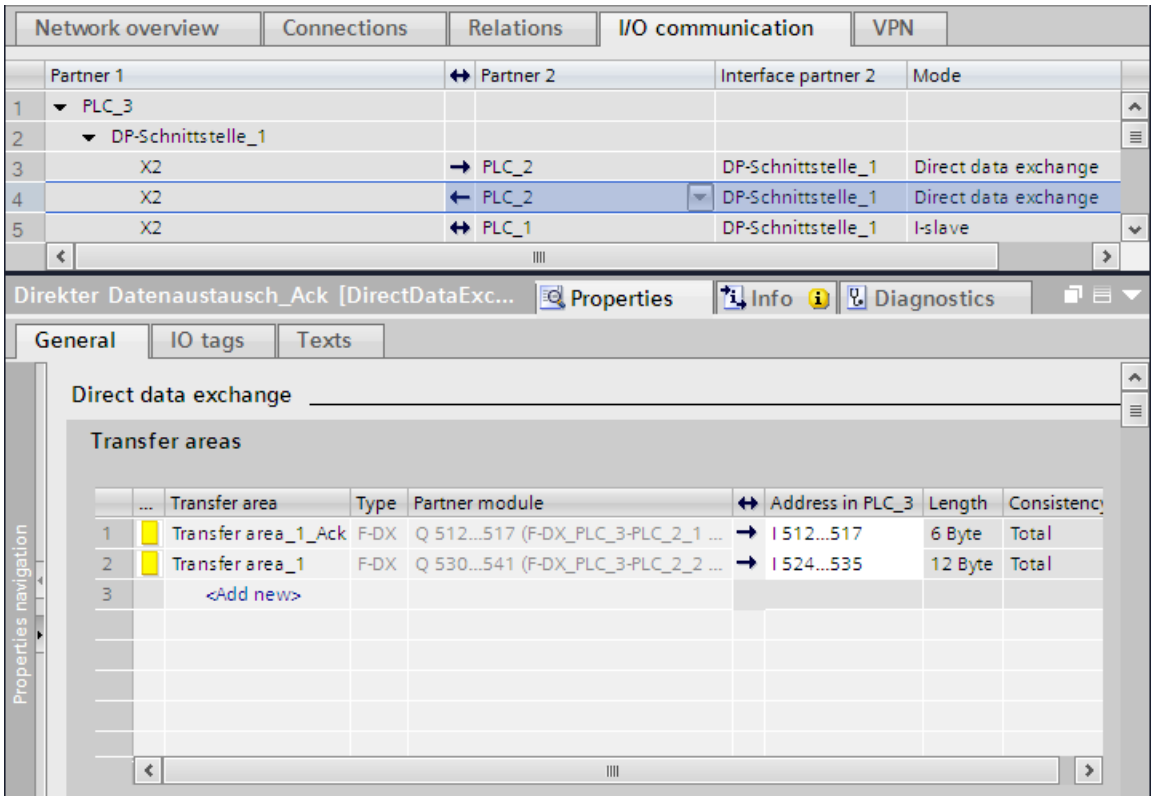
This completes the configuration for sending to F-CPU 2.



8. In the "I/O communication" tab, select the automatically created line with "Direct data exchange" mode for receiving from the I-slave (F-CPU 2) (←).
9. In "Transfer areas" ("Direct data exchange" table), create another F-DX connection for receiving from the I-slave (F-CPU 3).

   In this case, as well, an acknowledgment connection (→, transfer area x_Ack) is created automatically in the "Direct data exchange table" and two transfer areas (type F-MS) for the master CPU are created in the "I-slave communication" table of both I-slaves.

This completes the configuration for receiving from F-CPU 2.



Changing disabled local address areas of the transfer areas

To change the disabled local address area of the "Transfer area x", you must change the start address of the address area of the corresponding acknowledgment connection "Transfer area x_Ack".

1. In "I/O communication", select the line with the arrow pointing in the same direction as the arrow of "Transfer area x" in the "Direct data exchange" table.
2. Then select the line with "Transfer area x_Ack" in the "Direct data exchange" table.
3. Change the start address of the address range there.

9.1.6.2 Programming safety-related I-slave-I-slave communication

Reference

The description of the programming of safety-related I-slave-I-slave communication can be found in Program the safety-related master-I-slave or I-slave-I-slave communication (Page 228).

The assignment of the start addresses of the transfer areas to the LADDR input of the SENDDP/RCVDP instructions can be obtained from the following table.

Instruction	Start address LADDR	
	From row	From column
SENDDP in the 1st I-slave	→	Address in the <1st I-slave> (in example column "Address in PLC_2")
RCVDP in the 1st I-slave	←	Address in the <1st I-slave> (in example column "Address in PLC_2")
SENDDP in the 2nd I-slave	←	Address in the <2nd I-slave> (in example column "Address in PLC_3")
RCVDP in the 2nd I-slave	→	Address in the <2nd I-slave> (in example column "Address in PLC_3")

9.1.6.3 Limits for data transfer of safety-related I-slave-I-slave communication

Limits for data transfer

The description of the limits for the data transfer of safety-related I-slave-I-slave communication can be found in Limits for data transfer of safety-related master-I-slave or I-slave-I-slave communication (Page 230).

9.1.7 Safety-related IO controller-I-slave communication

Introduction

Safety-related communication between the safety program of the F-CPU of an IO controller and the safety program(s) of the F-CPU(s) of one or more I-slaves takes place via master-I-slave transfer areas (**F-**MS), as in standard systems.

IE/PB link

For the safety-related IO-controller-I-slave communication, the IE/PB link is mandatory. Each of the two F-CPUs is linked to the IE/PB link by means of its PROFIBUS DP or PROFINET-interface.

NOTE

If you are using an IE/PB link, you must take this into account when configuring the F-specific monitoring times and when calculating the maximum response time of your F-system (see also Monitoring and response times (Page 629)).

Note that the Excel file for calculating response times (https://support.industry.siemens.com/cs/ww/en/view/109783831) for S7-300/400 F-CPUs does not support all conceivable configurations.

Reference

The information on safety-related master-I-slave communication in Safety-related master-I-slave communication (Page 226) also applies.

9.1.8 Safety-related communication via S7 connections

9.1.8.1 Configuring safety-related communication via S7 connections

Introduction

Safety-related communication between the safety programs of F-CPUs via S7 connections takes place by means of established S7 connections that you create in the network view of the *hardware and network editor* - same as in standard programs.

Restrictions

NOTE

In SIMATIC Safety, S7 connections are generally permitted only via Industrial Ethernet.

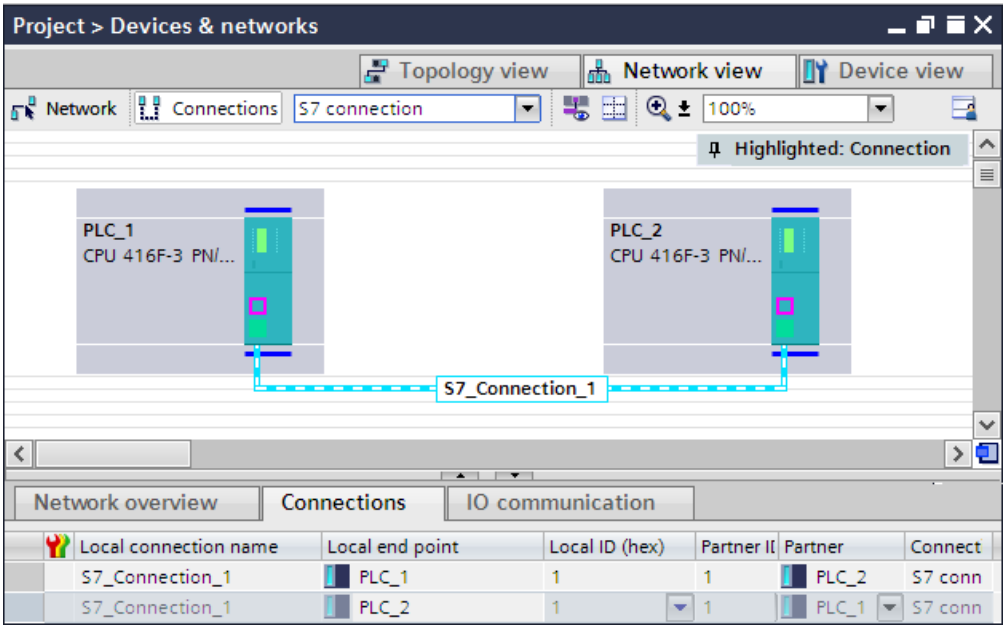
Safety-related communication via S7 connections is possible from and to the following CPUs:

- S7-300 F-CPUs via the integrated PROFINET interface
- S7-400 F-CPUs via the integrated PROFINET interface or a CP 443-1 or CP 443-1 Advanced-IT

Creating S7 connections

For each connection between two F-CPUs, you must create an S7 connection in the network view of the *hardware and network editor*.

For every end-point of a connection, a local and a partner ID is automatically assigned from the perspective of the end-point (the F-CPU). If necessary, you can change both IDs in the "Connections" tab. You assign the local ID to the "ID" input of the SENDS7 and RCVS7 instructions in the safety programs.

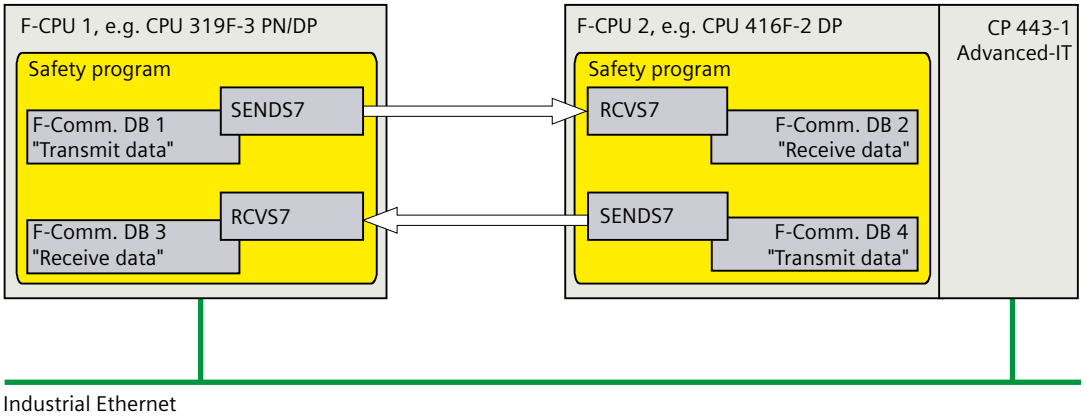


Procedure for configuring S7 connections

You configure the S7 connections for safety-related CPU-to-CPU communication in the same way as in *STEP 7 Professional* (see *Help on STEP 7 Professional* "S7 connections").

9.1.8.2 Programming safety-related communication via S7 connections

Communication via the SENDS7 and RCVS7 instructions



You use the **SENDS7 and RCVS7** instructions for fail-safe sending and receiving of data via S7 connections.

These instructions can be used to transmit a specified amount of fail-safe data of data types BOOL, INT, WORD, DINT, DWORD, and TIME in a fail-safe manner. The fail-safe data is stored in F-DBs (F-Communication DBs), which you create for each communication connection on the send and receive side.

You can find these instructions in the "Instructions" task card under "Communication". The RCVS7 instruction **must** be called at the start of the main safety block. The SENDS7 instruction **must** be called at the end of the main safety block.

Note that the send signals are sent only after calling the SENDS7 instruction at the end of the relevant F-runtime group execution.

A detailed description of the SENDS7 and RCVS7 instructions is found in SENDS7 and RCVS7: Communication via S7 connections (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 622).

Requirement for programming

The S7 connections between the relevant F-CPUs must be configured in the network view in the "Connections" tab of the *hardware and network editor*.

Creating and Editing an F-Communication DB

F-communication DBs are F-DBs that you create and edit in the same way as other F-DBs in the project tree. You can assign the F-communication DB numbers in the SENDS7 and RCVS7 instructions.

NOTE

The length and structure of the F-communication DB on the receiver side must match the length and structure of the associated F-communication DB on the sender side.

If the F-communication DBs do not match, the F-CPU can go to STOP mode. A diagnostics event is entered in the diagnostics buffer of the F-CPU.

For this reason, we recommend that you use the following procedure:

1. Create an F-communication DB in the project tree in or below the "Program blocks" folder of the F-CPU at the sender end.
2. Specify the appropriate structure of the F-communication DB, taking into account the data to be transferred.
3. Copy this F-communication DB to the project tree in or below the "Program blocks" folder of the F-CPU at the receiver end, and change the name, if necessary.

Other requirements for F-communication DBs

F-communication DBs must also conform to the following properties:

- They are not permitted to be instance DBs.
- Their length is not permitted to exceed 100 bytes.
- In F-communication DBs, only the following data types may be declared: BOOL, INT, WORD, DINT, DWORD, and TIME.

- The data types must be arranged block-wise and in the following order: BOOL, data types with bit length of 16 bits (INT, WORD), and data types with bit length of 32 bits (DINT, DWORD, and TIME). Within the data blocks with lengths of 16 bits and 32 bits, the data types can be arranged in any order.
- No more than 128 data elements of data type BOOL are permitted to be declared.
- The amount of data of data type BOOL must always be an integer multiple of 16 (word limit). Reserve data must be added, if necessary.

If these criteria are not fulfilled, *STEP 7 Safety Advanced* outputs an error message during compilation.

Assignment of fail-safe values

Fail-safe values are made available at the receiver end:

- While the connection between the communication partners is being established the first time after startup of the F-systems
- Whenever a communication error occurs

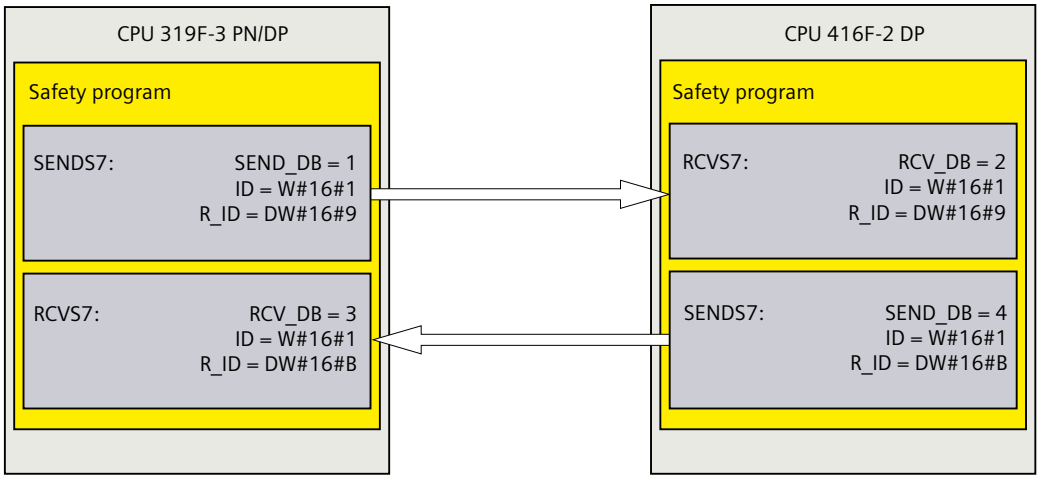
The values you specified as initial values in the F-communication DB at the receiver end are made available as initial values.

Programming procedure

You program safety-related communication via S7 connections as follows:

1. Supply the tags in the F-communication DB at the sender end with send signals using fully qualified access (e.g., "Name of F-communication DB".tag name).
2. Read the tags in the F-communication DB at the receiver end (receive signals) that you want to process further in other sections of the program using fully qualified access (e.g., "Name of F-communication DB".tag name).
3. In the safety program from which data is to be sent, call the SENDS7 instruction for sending at the end of the main safety block.
4. In the safety program in which data is to be received, call the instruction RCVS7 for receiving at the start of the main safety block.
5. Assign F-communication DB numbers to the SEND_DB input of SENDS7 and the RCV_DB input of RCVS7.
6. Assign the local ID of the S7 connection (data type: WORD) from the perspective of the F-CPU that was configured in the "Connections" tab of the network view to the ID input of SENDS7.
7. Assign the local ID of the S7 connection (data type: WORD) that was configured in the "Connections" tab of the network view to the ID input of RCVS7.

8. Assign an odd number (data type: DWORD) to the R_ID inputs of SENDS7 and RCVS7. This serves to specify that a SENDS7 instruction belongs to an RCVS7 instruction. The associated instructions receive the same value for R_ID.



WARNING

The value of the respective F-communication ID (input R_ID; data type: DWORD) can be freely selected; however, it must be odd and unique for all safety-related communication connections network-wide* and CPU-wide. The value R_ID + 1 is internally assigned and must not be used.

You must supply inputs ID and R_ID with constant values when calling the instruction. Direct read or write access to ID and R_ID in the associated instance DB is not permitted in the safety program. *(S020)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

9. Assign the TIMEOUT inputs of the SENDS7 and RCVS7 instructions with the required monitoring time.

WARNING

It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time. *(S018)*

Information on calculating the monitoring times can be found in Monitoring and response times (Page 629).

10. To reduce the bus load, you can temporarily shut down communication between the F-CPUs at input EN_SEND of the SENDS7 instruction. To do so, supply input EN_SEND (initial value = "TRUE") with 0. In this case, send data is no longer sent to the F-communication DB of the associated RCVS7 instruction and the receiver RCVS7 provides fail-safe values for this period (initial values in its F-communication DB). If communication was already established between the partners, a communication error is detected.
11. Optional: Evaluate the ACK_REQ output of RCVS7, for example, in the standard user program or on the operator control and monitoring system in order to query or to indicate whether user acknowledgment is required.
12. Supply the ACK_REI input of RCVS7 with the signal for the acknowledgment for reintegration.
13. Optional: evaluate the SUBS_ON output of RCVS7 or SENDS7 in order to query whether the RCVS7 instruction is outputting the fail-safe values you specified as initial values in the F-communication DB.
14. Optional: Evaluate the ERROR output of RCVS7 or SENDS7, for example, in the standard user program or on the operator control and monitoring system in order to query or to indicate whether a communication error has occurred.
15. Optional: evaluate the SENDMODE output of RCVS7 in order to query whether the F-CPU with the associated SENDS7 instruction is in disabled safety mode (Page 393).

Particularities for migrated projects

If you have migrated a project from *S7 Distributed Safety V5.4 SP5* to *STEP 7 Safety Advanced* in which safety-related communication via S7 connections is programmed, you must note the following:

- Do not delete migrated instance DBs for the SENDS7 and RCVS7 instructions in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks".

Otherwise, communication errors may occur in the relevant communication connections.

A migrated instance DB for the SENDS7 and RCVS7 instructions has been deleted if, after compiling the safety program, the "User-defined ID" in the newly generated instance DB is not identical to "FRCVS7CL" or "FSNDS7CL".

You can find the "User-defined ID" of a block in its properties in the "Information" area.

9.1.8.3 Safety-related communication via S7 connections - Limits of data transfer

NOTE

If the amount of data to be transmitted exceeds the permitted length for the F-communication DB (100 bytes), you can create a further F-communication DB that you transfer to additional SENDS7/RCVS7 instructions with a modified R_ID.

Note that USEND and URCV instructions are called internally at each SENDS7 or RCVS7 call and use connection resources in the F-CPU. This affects the maximum number of communication connections available (*see manuals for F-CPUs*).

More information can be found on the data transfer limits for S7 connections of individual F-CPUs is available on the Internet (https://support.industry.siemens.com/cs/ww/en/view/38549114).

9.1.9 Safety-related communication with other S7 F-systems

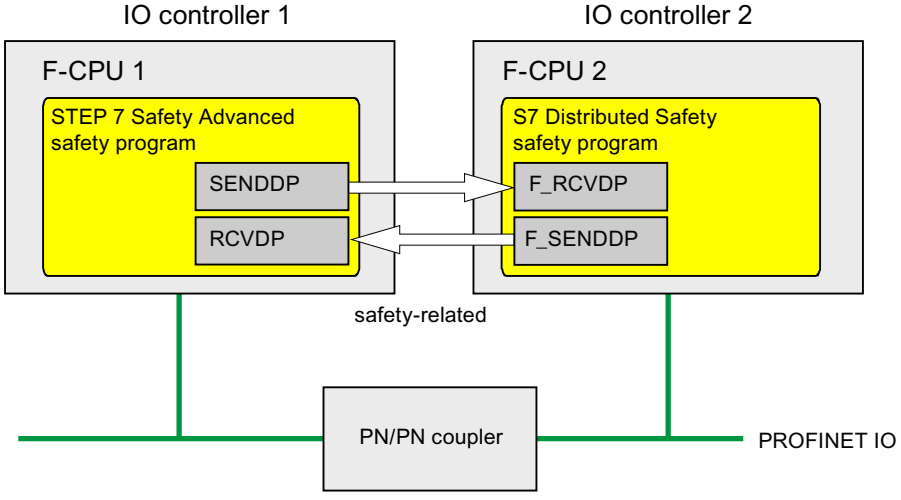
9.1.9.1 Introduction

Safety-related communication from F-CPUs in SIMATIC Safety to F-CPUs in S7 Distributed Safety F-systems is possible via a PN/PN coupler or DP/DP coupler that you use between the two F-CPUs as IO controller-IO controller communication, master-master communication or communication via established S7 connections.

Safety-related communication from F-CPUs in SIMATIC Safety to F-CPUs in S7 F/FH Systems F-systems is possible via established S7 connections.

9.1.9.2 Communication with S7 Distributed Safety via PN/PN coupler (IO controller-IO controller communication)

Communication functions between SENDDP/RCVDP instructions at the *STEP 7 Safety Advanced* end and F-application blocks F_SENDDP/F_RCVDP at the *S7 Distributed Safety* end:



Procedure at the *S7 Distributed Safety* end

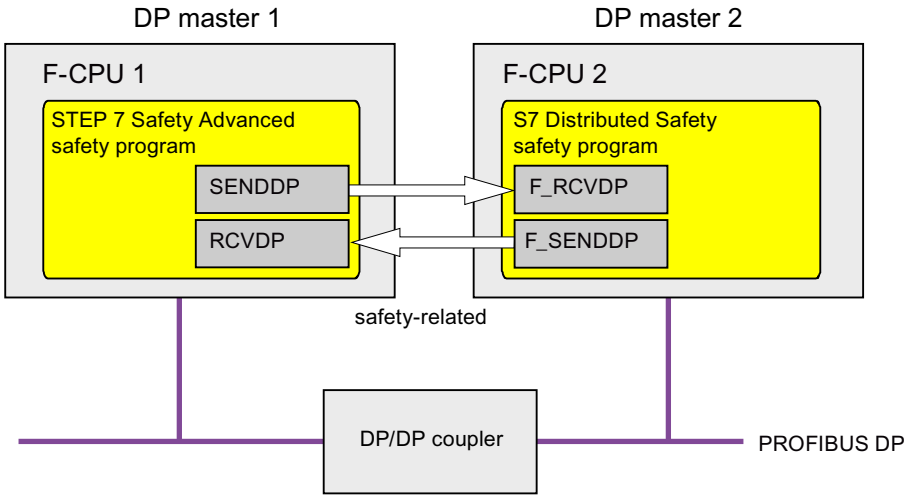
At the *S7 Distributed Safety* end, proceed as described in "Safety-related IO controller-IO controller communication" in the S7 Distributed Safety - Configuring and Programming (https://support.industry.siemens.com/cs/ww/en/view/22099875) manual.

Procedure at the *STEP 7 Safety Advanced* end

At the *STEP 7 Safety Advanced* end, proceed as described in Safety-related IO controller-IO controller communication (Page 205).

9.1.9.3 Communication with S7 Distributed Safety via DP/DP coupler (master-master communication)

Communication functions between SENDDP/RCVDP instructions at the *STEP 7 Safety Advanced* end and F-application blocks F_SENDDP/F_RCVDP at the *S7 Distributed Safety* end:



Procedure at the *S7 Distributed Safety* end

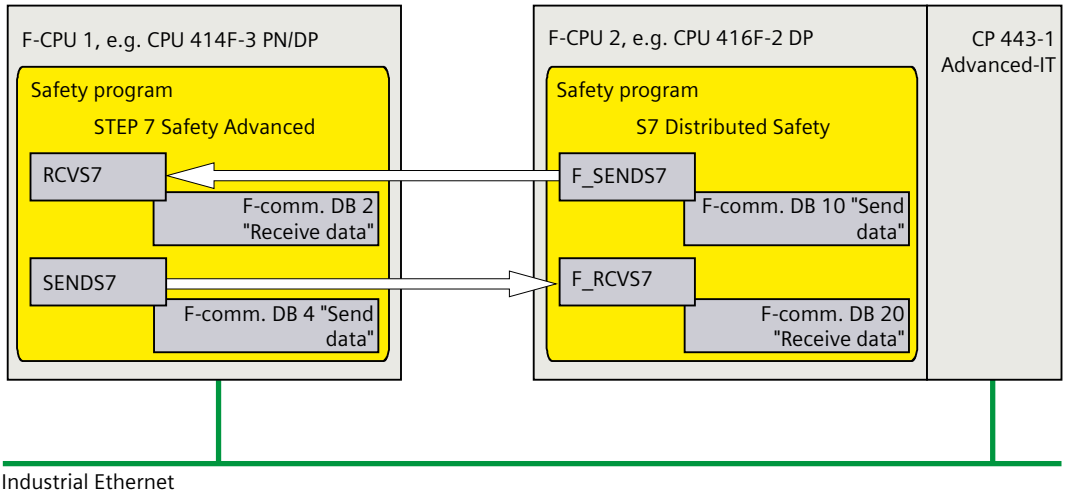
At the *S7 Distributed Safety* end, proceed as described in "Safety-related master-master communication" in the S7 Distributed Safety - Configuring and Programming (https://support.industry.siemens.com/cs/ww/en/view/22099875) manual.

Procedure at the *STEP 7 Safety Advanced* end

At the *STEP 7 Safety Advanced* end, proceed as described in Safety-related master-master communication (Page 212).

9.1.9.4 Communication with S7 Distributed Safety via S7 connections

Communication functions between SENDS7/RCVS7 instructions at the *STEP 7 Safety Advanced* end and F_SENDS7/F_RCVS7 F-application blocks at the *S7 Distributed Safety* end:



Procedure at the *S7 Distributed Safety* end

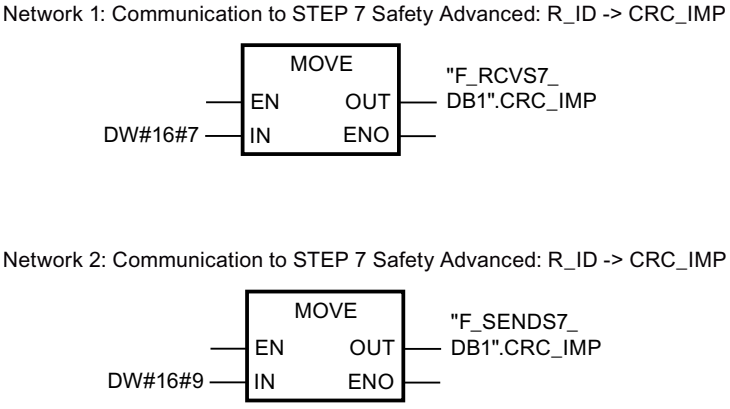
At the *S7 Distributed Safety* end, proceed as described in section "Safety-related communication via S7 communications" in the S7 Distributed Safety - Configuring and Programming (https://support.industry.siemens.com/cs/ww/en/view/22099875) manual.

Because safety-related communication via S7 connections is not possible with unspecified partners in S7 Distributed Safety, you must first create a "virtual" SIMATIC station in *S7 Distributed Safety* in which you configure an F-CPU as a proxy for the F-CPU in *STEP 7 Safety Advanced* with its IP address.

You then insert an S7 connection to this F-CPU in the connection table. Both the local connection and partner connection resources (hex) are thereby fixed. You must then set these in the associated, unspecified S7 connection that you created in *STEP 7 Professional*.

In addition, for all communication connections to this F-CPU, you must transfer the F-communication ID that you assigned in the R_ID input of the associated calls of the F_SENDS7 and F_RCVS7 F-application blocks additionally to the CRC_IMP tag in the instance DB of F_SENDS7 and F_RCVS7, respectively, in the standard user program immediately before calling the F-CALL.

Program example:



Procedure at the *STEP 7 Safety Advanced* end

At the *STEP 7 Safety Advanced* end, proceed as described in Safety-related communication via S7 connections (Page 237).

For the F-CPU in *S7 Distributed Safety*, you must create and specify an unspecified S7 connection. You can find information on this in the *STEP 7* help, under "Creating an unspecified connection" or "Specifying and unspecified connection".

For these you must set the local and partner connection resources (hex) that are fixed as a result of the associated S7 connection that you have created in *S7 Distributed Safety*.

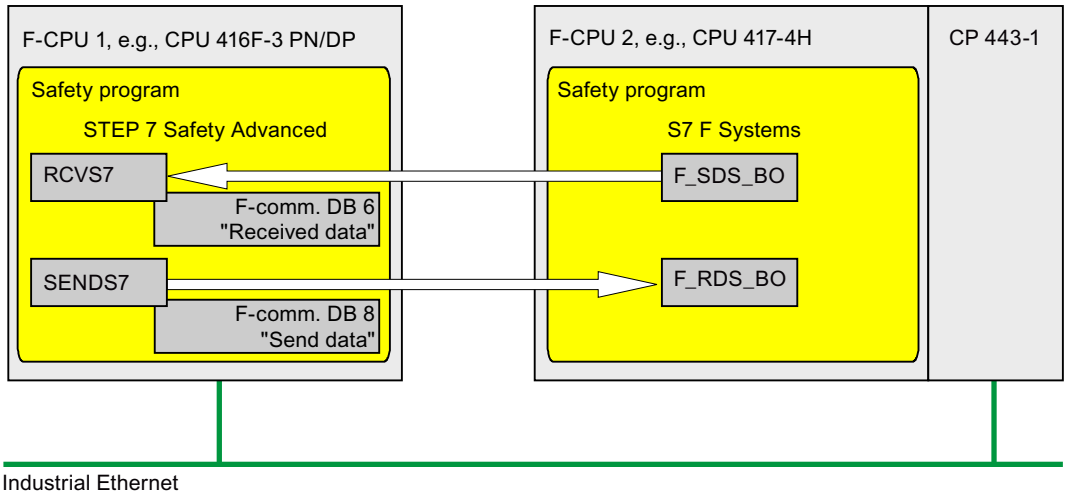
If the local connection resource (hex) is already occupied by an existing connection, you must change the connection resource (hex) for it.

If the instance DBs of the SENDS7 and RCVS7, instructions that you want to use for communication with *S7 Distributed Safety* were migrated from *S7 Distributed Safety*, you must delete them in the project tree in the "STEP 7 Safety" folder, under "Program blocks > System blocks" (contrary to the information in Programming safety-related communication via S7 connections (Page 238), section "Particularities for migrated projects").

9.1.9.5 Communication with S7 F/FH Systems via S7 connections

Communication functions between SENDS7/RCVS7 instructions at the *STEP 7 Safety Advanced* end and F_SDS_BO/F_RDS_BO F-blocks at the *S7 F Systems* end.

A maximum of 32 data elements of data type BOOL can be exchanged.



Procedure at the *S7 F Systems* end

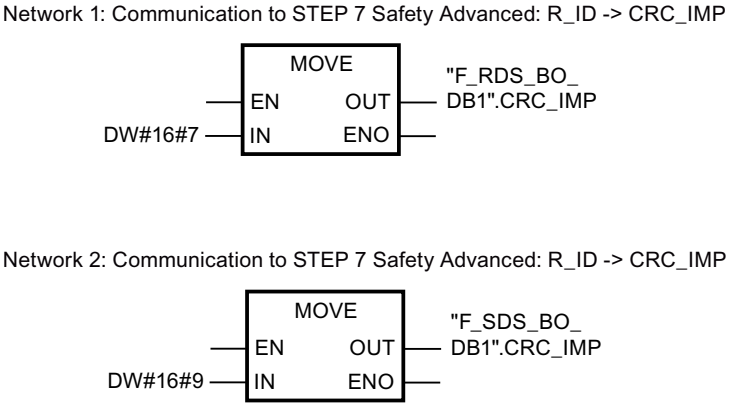
At the *S7 F systems* end, proceed as described in section "Safety-related communication between F-CPUs" in the S7 F/FH Systems - Configuring and Programming (https://support.industry.siemens.com/cs/ww/en/view/109802562) manual.

Because safety-related communication via S7 connections is not possible with unspecified partners in *S7 F/FH Systems*, you must first create a "virtual" SIMATIC station in *S7 F/FH Systems* in which you configure an F-CPU as a proxy for the F-CPU in *STEP 7 Safety Advanced* with its IP address.

You then insert an S7 connection to this F-CPU in the connection table. Both the local connection and partner connection resources (hex) are thereby fixed. You must then set these in the associated, unspecified S7 connection that you created in *STEP 7 Safety Advanced*.

In addition, you must insert a function in your S7 program (in the area reserved in CFC for other applications), in which, for all communication connections for this F-CPU, you transfer the F-communication ID that you assigned in the R_ID input of the associated calls of the F_SDS_BO and F_RDS_BO F-blocks additionally to the CRC_IMP tag in the instance DB of the F_SDS_BO and F_RDS_BO, respectively. You obtain the number of the instance DB from the object properties of the block in CFC. Assign descriptive names for these instance DBs. If you perform a compress operation in CFC, you must check whether the numbers of these instance DBs have changed.

Program example:



You must then import the function in CFC as block type and insert your standard user program in a chart. In the run sequence, make sure that the associated standard runtime group is processed before the F-runtime group.

Procedure at the *STEP 7 Safety Advanced* end

At the *STEP 7 Safety Advanced* end, proceed as described in "Safety-related communication via S7 connections" (Page 237).

Particularity: In *STEP 7 Safety Advanced*, you must create the F-communication DB with exactly 32 data elements of data type BOOL.

For the F-CPU in *S7 F/FH Systems*, you must create and specify an unspecified S7 connection. You can find information on this in the *STEP 7* help, under "Creating an unspecified connection" or "Specifying an unspecified connection".

For these you must set the local and partner connection resources (hex) that are fixed as a result of the associated S7 connection that you have created in *S7 F Systems*.

If the local connection resource (hex) is already occupied by an existing connection, you must change the connection resource (hex) for it.

9.2 Communication via SENDDP/RCVDP instructions (S7-1200, S7-1500):

9.2.1 Overview

Introduction

This section gives an overview of the options for safety-related communication using the SENDDP/RCVDP instructions in SIMATIC Safety F-systems.

Options for safety-related CPU-CPU communication

Safety-related CPU-CPU communication	On subnet	Additional hardware required	Required instructions
IO controller-IO controller communication	PROFINET IO	PN/PN coupler	SENDDP/RCVDP
Master-master communication	PROFIBUS DP	DP/DP coupler	SENDDP/RCVDP
IO controller-I-device communication	PROFINET IO	—	SENDDP/RCVDP
Master-I-slave communication	PROFIBUS DP	—	SENDDP/RCVDP
IO controller-I-slave communication	PROFINET IO and PROFIBUS DP	IE/PB link	SENDDP/RCVDP
IO controller-IO controller communication for *S7 Distributed Safety*	PROFINET IO	PN/PN coupler	SENDDP/RCVDP
Master-master communication for *S7 Distributed Safety*	PROFIBUS DP	DP/DP coupler	SENDDP/RCVDP

NOTE

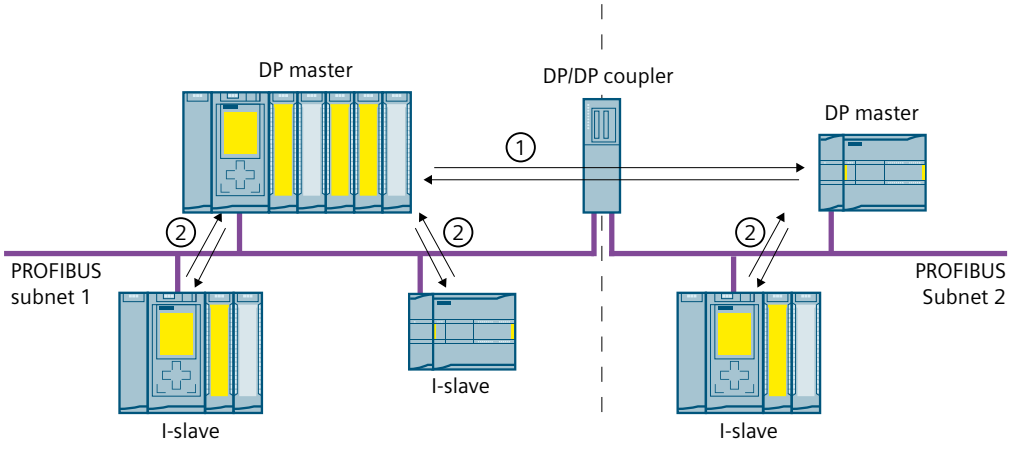
The options for safety-related communication depend on the F-CPUs used.

NOTE

Safety-related communication with S7-1200 G1 F-CPUs is only permitted as of firmware V4.1.2.

Overview of safety-related communication via PROFIBUS DP

The figure below provides an overview of the options for safety-related communication via PROFIBUS DP in SIMATIC Safety F-systems with S7-1200/1500 F-CPUs.

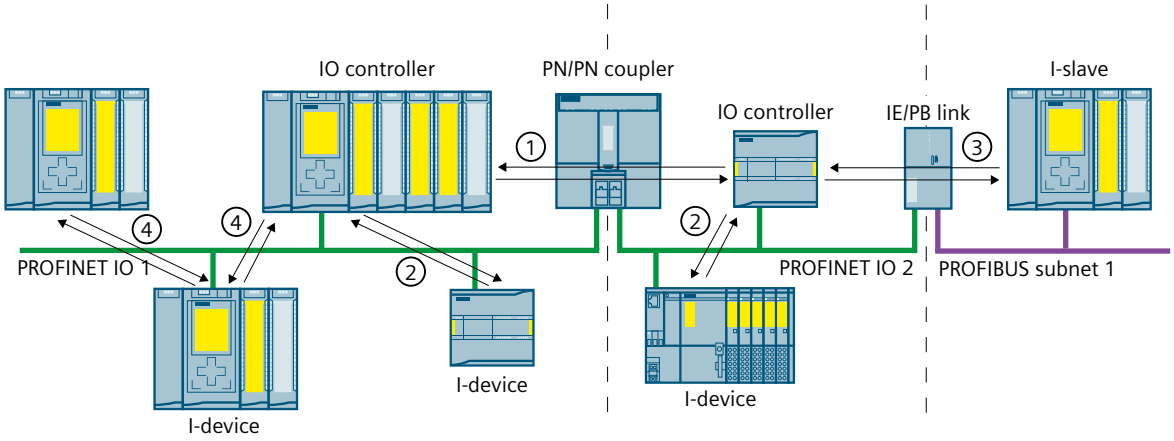


① Safety-related master-master communication

② Safety-related master-I-slave communication

Overview of safety-related communication via PROFINET IO

The figure below provides an overview of the options for safety-related communication via PROFINET IO in SIMATIC Safety F-systems with S7-1200/1500 F-CPUs.



① Safety-related IO controller-IO controller communication
② Safety-related IO controller-I-device communication
③ Safety-related IO controller-I-slave communication
④ Safety-related IO controller-I-device communication (shared I-device)

Safety-related CPU-CPU communication via PROFIBUS DP or PROFINET IO

In safety-related CPU-CPU communication, a fixed amount of data of the data type BOOL or INT (DINT as alternative) is transmitted fail-safe between the safety programs in F-CPUs of DP masters/I-slaves or IO controllers/I-devices.

The data are transferred using the SENDDP instruction for sending and the RCVDP instruction for receiving. The data are stored in configured transfer areas of the devices. The hardware identifier (HW identifier) defines the transfer areas configured.

Safety-related CPU-CPU communication for *S7 Distributed Safety*

Safety-related communication is possible from F-CPUs in *SIMATIC Safety* to F-CPUs in *S7 Distributed Safety*.

9.2.2 Safety-related IO controller-IO controller communication

9.2.2.1 Configure safety-related IO controller-IO controller communication

Introduction

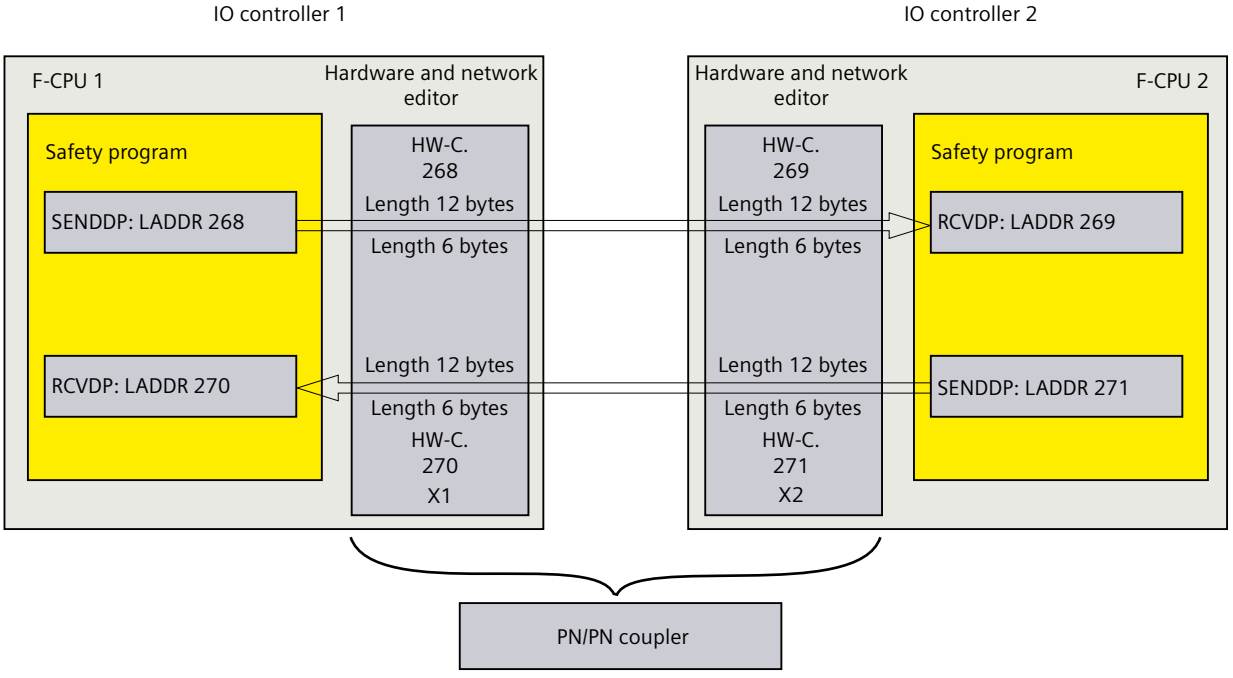
Safety-related communication between safety programs of the F-CPUs of IO controllers takes place over a PN/PN coupler that you set up between the F-CPUs.

NOTE

Deactivate the "Data validity display DIA" parameter in the properties for the PN/PN coupler in the *hardware and network editor*. This is the default setting. Otherwise, safety-related IO controller-IO controller communication is not possible.

Configuring transfer areas

You must configure one transfer area for output data and one transfer area for input data in the *hardware and network editor* for each safety-related communication connection between two F-CPUs in the PN/PN coupler. The figure below shows how both of the F-CPUs are able to send **and** receive data (bidirectional communication).



Rules for defining transfer areas

Data to be sent:

A total of 12 bytes (consistent) is required for the transfer area for output data; 6 bytes (consistent) are required for the transfer area for input data.

Data to be received:

A total of 12 bytes (consistent) is required for the transfer area for input data; 6 bytes (consistent) are required for the transfer area for output data.

NOTE

PN/PN Coupler article number 6ES7158-3AD10-0XA0

When configuring the transfer areas for the output and input data, proceed as described in the "SIMATIC bus links PN/PN coupler (https://support.industry.siemens.com/cs/ww/en/view/44319532)" manual, section "Configuring the PN/PN Coupler with STEP 7 TIA Portal".

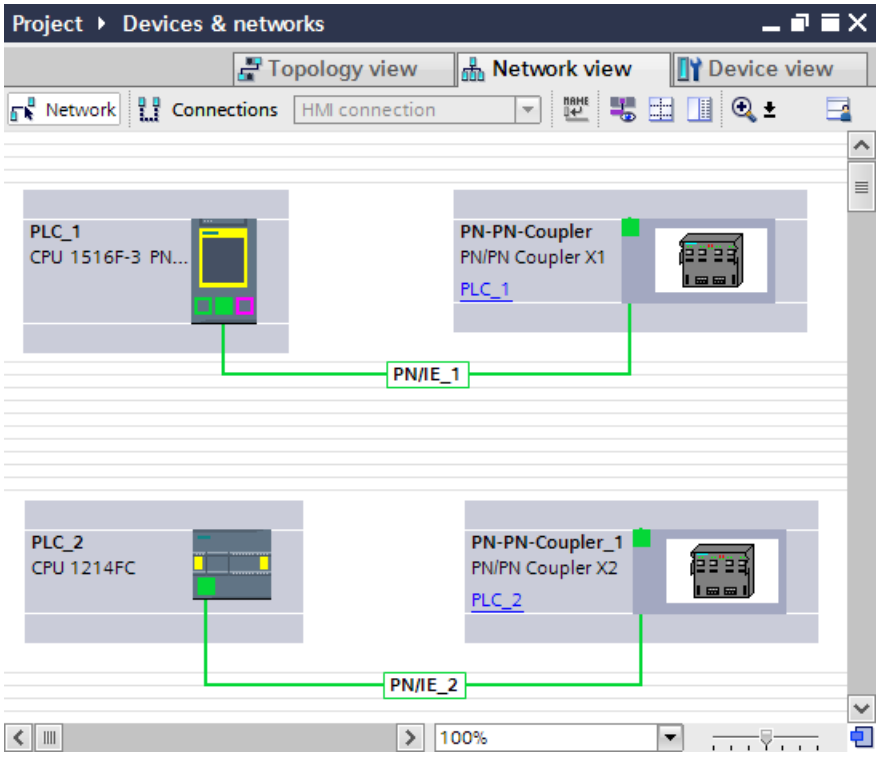
Procedure for configuration

The procedure for configuring safety-related IO controller-IO controller communication is identical to that in the standard system.

Proceed as follows:

1. Insert two F-CPUS from the "Hardware catalog" task card into the project.
2. Switch to the network view of the *hardware and network editor*.
3. Select a PN/PN Coupler X1 and a PN/PN Coupler X2 from "Other field devices\PROFINET IO\Gateway\Siemens AG\PN/PN Coupler" in the "Hardware catalog" task card and insert them into the network view of the hardware and network editor.

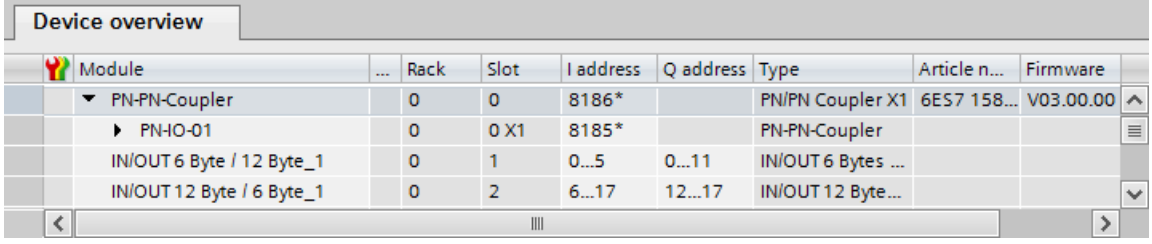
4. Connect the PN interface of the F-CPU 1 with the PN interface of the PN/PN Coupler X1 and the PN interface of the F-CPU 2 with the PN interface of PN/PN Coupler X2.



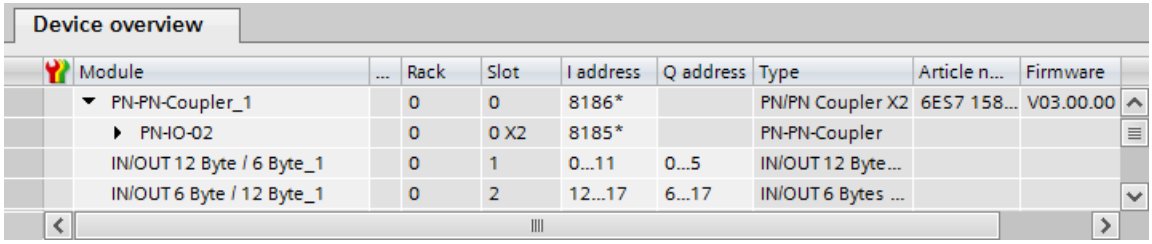
5. Switch to the device view of PN/PN Coupler X1 for bidirectional communication connections i.e. where each F-CPU is both to send and to receive data. Select the following modules from "IN/OUT" in the "Hardware catalog" task card (with filter activated), and insert them in the "Device overview" tab:
   - One "IN/OUT 6 bytes / 12 bytes" module and
   - One "IN/OUT 12 bytes / 6 bytes" module

NOTE

The transfer areas are assigned on the basis of the hardware identifier which is automatically assigned to the modules and devices. You need the HW identifier to program the SENDDP and RCVDP blocks (LADDR input). A system constant is created in the corresponding F-CPU for each hardware identifier of the transfer area. You can assign these system constants symbolically to the SENDDP and RCVDP blocks.

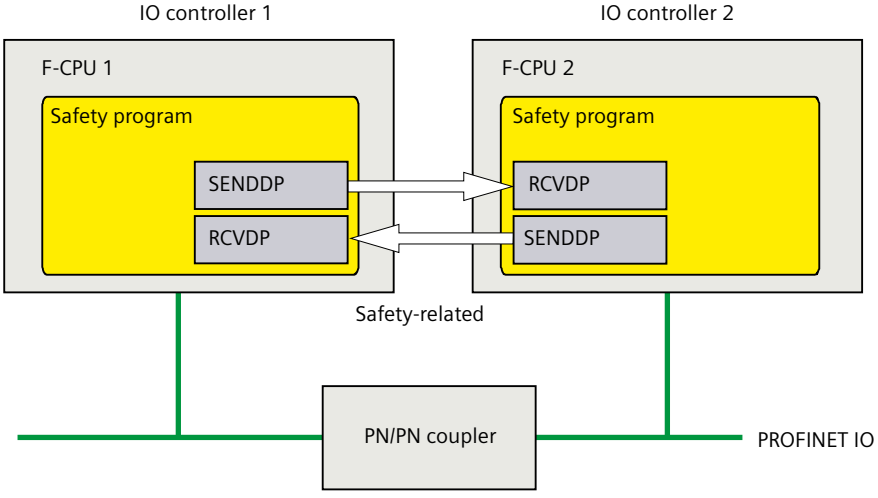


6. Select the following modules from "IN/OUT" in the device view of PN/PN coupler X2 and insert them in the "Device overview" tab:
   - One "IN/OUT 12 bytes / 6 bytes" module and
   - One "IN/OUT 6 bytes / 12 bytes" module



9.2.2.2 Program safety-related IO controller-IO controller communication

Communication via SENDDP and RCVDP instructions



Safety-related communication between the F-CPUs of the IO controllers uses the SENDDP and RCVDP instructions for sending and receiving, respectively. These can be used to perform a fail-safe transfer of a *fixed* amount of fail-safe data of the data type BOOL or INT (DINT as alternative).

You can find these instructions in the "Instructions" task card under "Communication". The RCVDP instruction **must** be called at the start of the main safety block. The SENDDP instruction **must** be called at the end of the main safety block.

You can also call the RCVDP and SENDDP instructions in separate F-FBs/F-FCs which you have to call up at the beginning or end of the main safety block.

Note that the send signals are not sent until after the SENDDP instruction call at the end of execution of the relevant F-runtime group.

You can find a detailed description of the SENDDP and RCVDP instructions in SENDDP and RCVDP: Send and receive data via PROFIBUS DP/PROFINET IO (STEP 7 Safety V21) (Page 612).

Requirement for programming

The transfer areas for input and output data for the PN/PN coupler must be configured.

Programming procedure

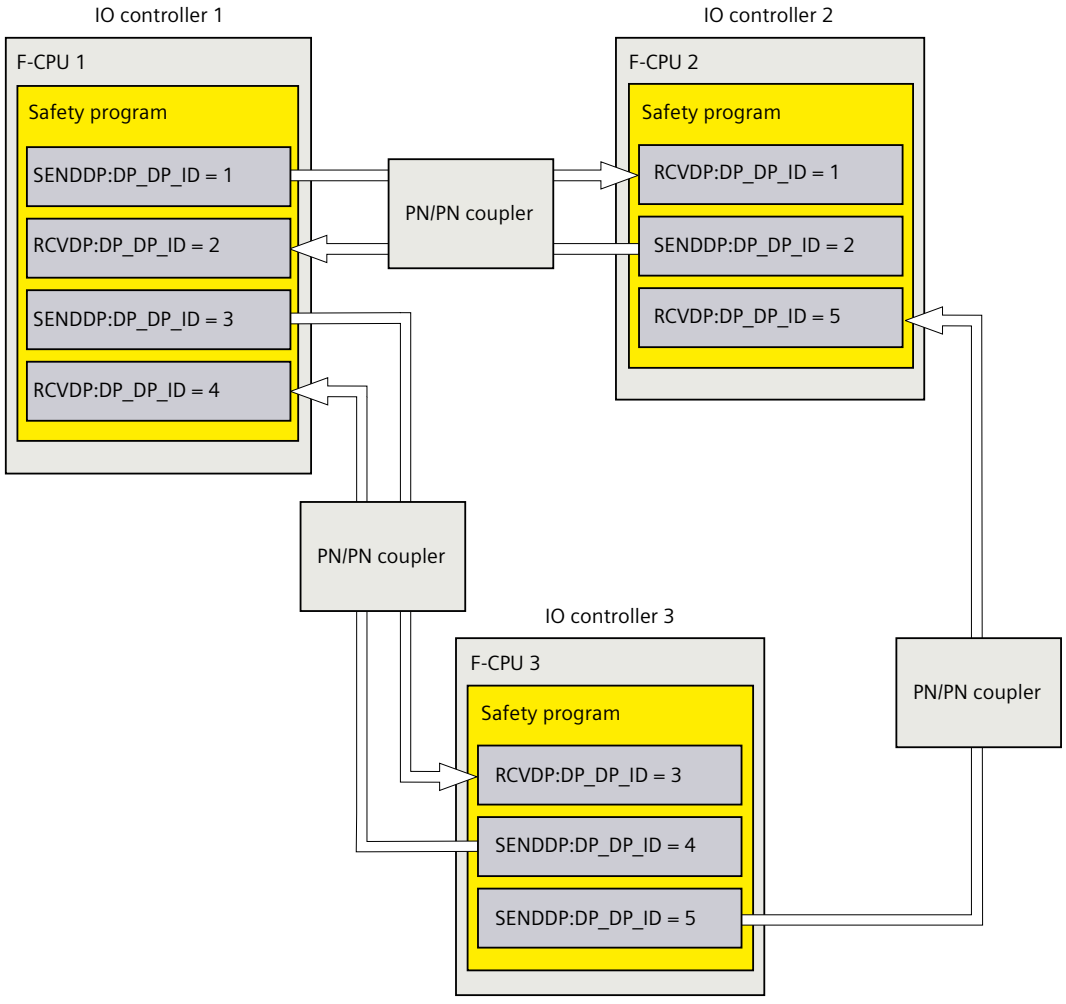
You program safety-related IO controller-IO controller communication as follows:

1. In the safety program from which data is to be sent, call the SENDDP instruction (Page 612) for sending at the end of the main safety block.
2. In the safety program in which data is to be received, call the RCVDP instruction (Page 612) for receiving at the start of the main safety block.
3. Assign the HW identifiers (system constants in the default tag table) of the transfer areas for output and input data of the PN/PN coupler that are configured in the hardware and network editor to the respective LADDR inputs.

   You must carry out this assignment for every communication connection for each of the F-CPUs involved.

4. Assign the value for the respective F-communication ID to the DP_DP_ID inputs. This establishes the communication relationship between the SENDDP instruction in one F-CPU and the RCVDP instruction in the other F-CPU: The associated instructions receive the same value for DP_DP_ID.

   The figure below contains an example of how to specify the F-communication IDs at the inputs of the SENDDP and RCVDP instructions for 5 safety-related IO controller-IO controller communication relationships.



WARNING

The value for the respective F-communication ID (input DP_DP_ID; data type: INT) can be freely selected**; however, it must be unique network-wide* and CPU-wide**** at all times for all safety-related communication connections. You must check the uniqueness in the safety summary during acceptance of the safety program.

You must supply constant values*** to the DP_DP_ID and LADDR inputs when calling the instruction. Direct write access to DP_DP_ID and LADDR in the associated instance DB is not permitted in the safety program! *(S016)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, no connection is established at the DP_DP_ID input for a F-communication ID "0".

*** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, the DP_DP_ID input can also be supplied with variable values from a global F-DB. In this case also, you have to check during acceptance of the safety program that uniqueness is ensured at all times. You need to check the algorithm for forming the variable value accordingly. If you cannot ensure a unique F-communication ID during startup of the safety program, because it is only specified after startup of the safety program, you must make sure that the value at the DP_DP_ID input is "0" during this phase.

**** With a redundant S7-1500HF system, both F-CPUs of the redundant S7-1500HF system are to be regarded as one single F-CPU with regard to the DP_DP_ID.

5. Supply the SD_BO_xx and SD_I_xx inputs (SD_DI_00 as alternative) of SENDDP with the send signals. To cut down on intermediate signals when transferring block parameters, you can write the value directly to the instance DB of SENDDP using fully qualified access (for example, "Name SENDDP_1".SD_BO_02) before calling SENDDP.
6. Supply the RD_BO_xx and RD_I_xx outputs (RD_DI_00 as alternative) of RCVDP with the signals that you want to process further in other program sections or use fully qualified access to read the received signals directly in the associated instance DB in the program sections to be processed further (e.g., "Name RCVDP_1".RD_BO_02).
7. If you want to send the data at the SD_DI_00 input instead of the data at the SD_I_00 and SD_I_01 inputs, supply the DINTMODE input (initial value = "FALSE") of SENDDP with TRUE.
8. Supply the SUBBO_xx and SUBI_xx or alternatively SUBDI_00 inputs of RCVDP with the fail-safe values that are to be output by RCVDP in place of the process data until communication is established for the first time after startup of the sending and receiving F-systems or in the event of an error in safety-related communication.
   - Specification of constant fail-safe values:

     For data of data type INT/DINT, you can enter constant fail-safe values directly as constants in the SUBI_xx or alternatively SUBDI_00 input (initial value = "0"). If you want to specify a constant fail-safe value "TRUE" for data of the data type BOOL, set TRUE for the SUBBO_xx input (initial value = "FALSE").

– Specification of variable substitute values:

  If you want to specify variable substitute values, define a tag that you calculate through your safety program in an F-DB and specify this tag (fully qualified) in the SUBBO_xx or SUBI_xx or alternatively SUBDI_00 input.

  ⚠ WARNING

  Note: The program logic for calculating variable substitute values can only be inserted after the RCVDP calls, because there must be no program logic before the RCVDP calls. This is why the initial values of the substitute values are active in all RCVDP instructions in the first cycle after a startup of the F-system. You must therefore assign appropriate initial values for these tags. *(S017)*

9. Configure the TIMEOUT inputs of the RCVDP and SENDDP instructions with the required monitoring time.

   ⚠ WARNING

   It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time. *(S018)*

   Information on calculating the monitoring times can be found in Monitoring and response times (Page 629).

10. Optional: Evaluate the ACK_REQ output of the RCVDP instruction, for example, in the standard user program or on the operator control and monitoring system in order to query or to indicate whether user acknowledgment is required.
11. Supply the ACK_REI input of the RCVDP instruction with the acknowledgment signal for reintegration.
12. Optional: Evaluate the SUBS_ON output of the RCVDP or SENDDP instruction in order to query whether the RCVDP instruction is outputting the fail-safe values assigned in the SUBBO_xx and SUBI_xx or alternatively SUBDI_00 inputs.
13. Optional: Evaluate the ERROR output of the RCVDP or SENDDP instruction, for example, in the standard user program or on the operator control and monitoring system in order to query or to indicate whether a communication error has occurred.
14. Optional: Evaluate the SENDMODE output of the RCVDP instruction in order to query whether the F-CPU with the associated SENDDP instruction is in disabled safety mode (Page 393).

9.2.2.3 Safety-related IO controller-IO controller communication - Limits for data transfer

NOTE

If the data quantities to be transmitted exceed the capacity of the SENDDP / RCVDP correlated instructions, a second (or third) SENDDP / RCVDP call can be used. This requires configuration of an additional communication connection via the PN/PN coupler. Whether or not this is possible with one single PN/PN coupler depends on the capacity restrictions of the PN/PN coupler.

9.2.3 Safety-related master-master communication

9.2.3.1 Configure safety-related master-master communication

Introduction

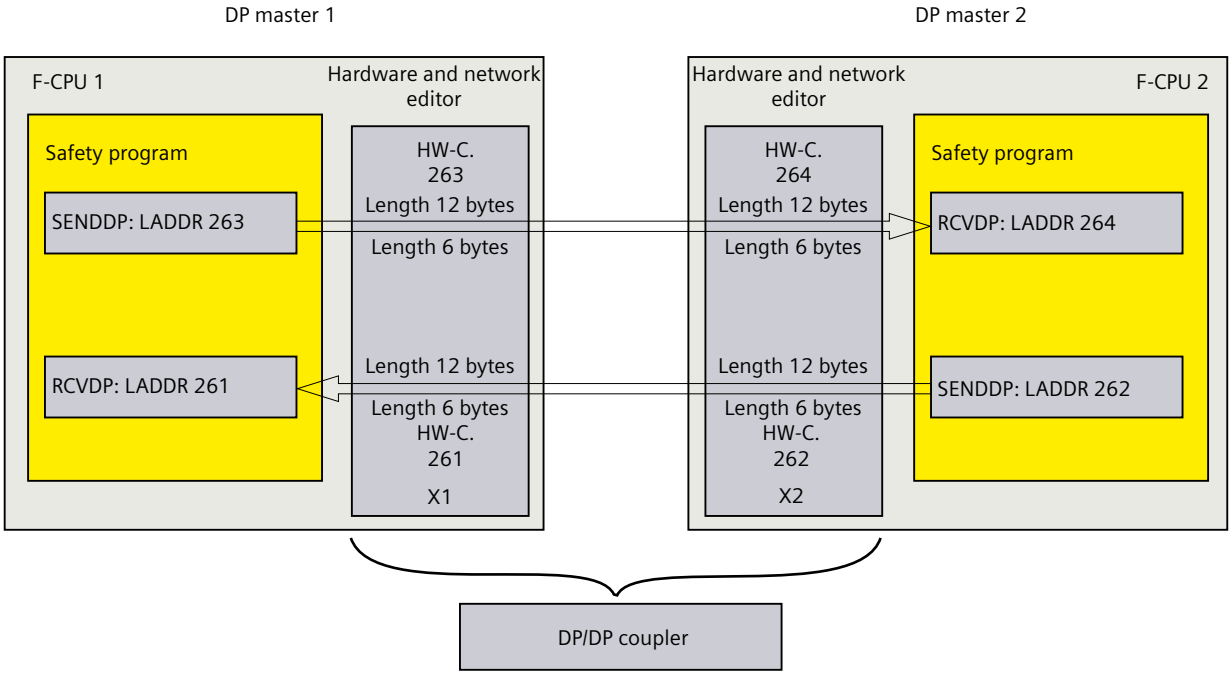
Safety-related communication between safety programs of the F-CPUs of DP masters takes place via a DP/DP coupler.

NOTE

Switch the data validity indicator "DIA" on the DIP switch of the DP/DP coupler to "OFF". Otherwise, safety-related CPU-CPU communication is not possible.

Configuring transfer areas

You must configure one transfer area for output data and one transfer area for input data in the *hardware and network editor* for each safety-related communication connection between two F-CPUs in the DP/DP coupler. The figure below shows how both of the F-CPUs are able to send and receive data (bidirectional communication).



Rules for defining transfer areas

Data to be sent:

A total of 12 bytes (consistent) is required for the transfer area for output data; 6 bytes (consistent) are required for the transfer area for input data.

Data to be received:

A total of 12 bytes (consistent) is required for the transfer area for input data; 6 bytes (consistent) are required for the transfer area for output data.

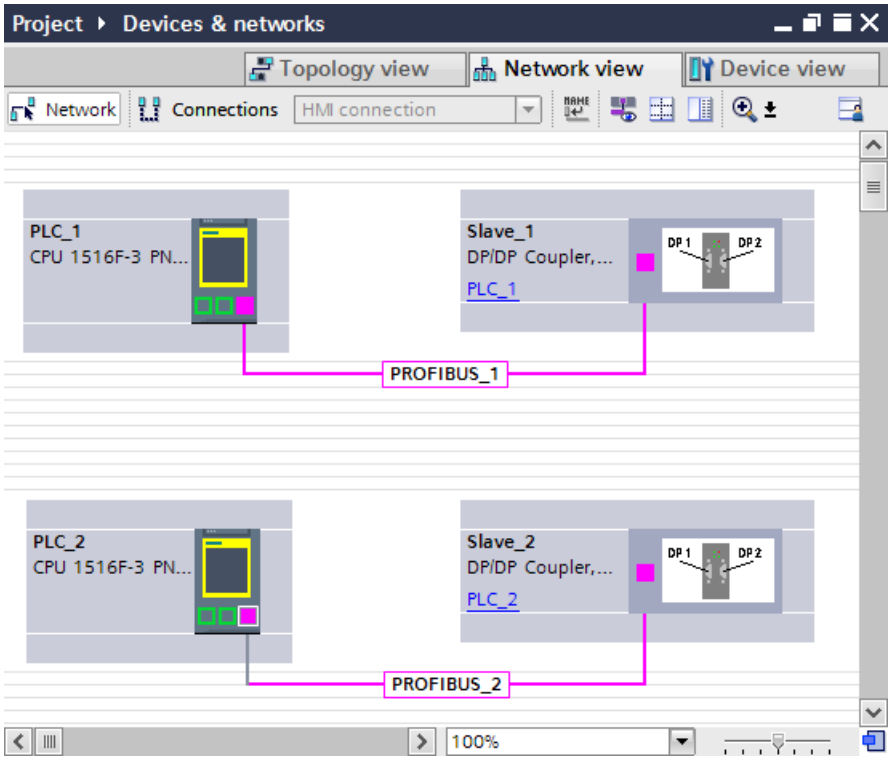
Procedure for configuration

The procedure for configuring safety-related master-master communication is identical to that in the standard system.

Proceed as follows:

1. Insert two F-CPUS from the "Hardware catalog" task card into the project.
2. Switch to the network view of the *hardware and network editor*.
3. Select a DP/DP coupler from "Other field devices\PROFIBUS DP\Gateways\Siemens AG\DP/DP Coupler" in the "Hardware catalog" task card and insert it into the network view of the hardware and network editor.

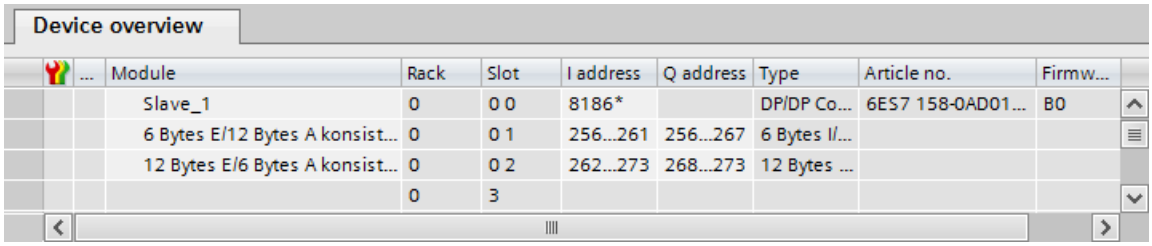
4. Insert a second DP/DP coupler.
5. Connect a DP interface of F-CPU 1 to the DP interface of a DP/DP coupler and a DP interface of F-CPU 2 to the DP interface of the other DP/DP coupler.



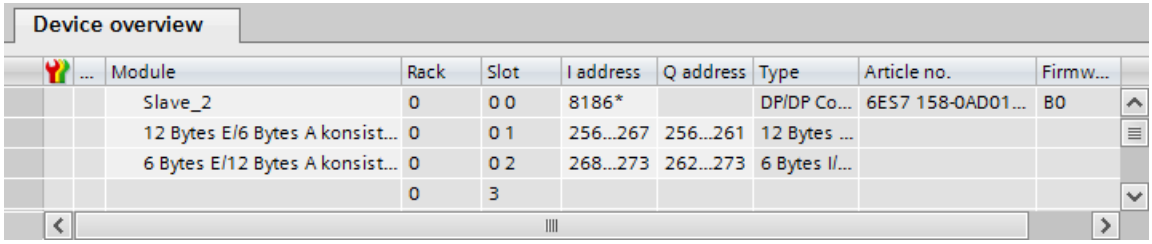
6. A free PROFIBUS address is assigned automatically in the properties of the DP/DP-coupler in the device view. You must set this address on the DP/DP coupler, either via the DIP switch on the device or in the configuration of the DP/DP coupler (see DP/DP Coupler (https://support.industry.siemens.com/cs/ww/en/view/1179382) manual).
7. Switch to the device view of the DP/DP coupler PLC1 for bidirectional communication connections i.e. where each F-CPU should both send and receive data. Select the following modules from the "Hardware catalog" task card (with filter activated), and insert them in the "Device overview" tab of the DP/DP coupler:
   - One "6 bytes I/12 bytes Q consistent" module, and
   - One "12 bytes I/6 bytes Q consistent" module

NOTE

The transfer areas are assigned on the basis of the hardware identifier which is automatically assigned to the modules and devices. You need the HW identifier to program the SENDDP and RCVDP blocks (LADDR input). A system constant is created in the corresponding F-CPU for each hardware identifier of the transfer area. You can assign these system constants symbolically to the SENDDP and RCVDP blocks.

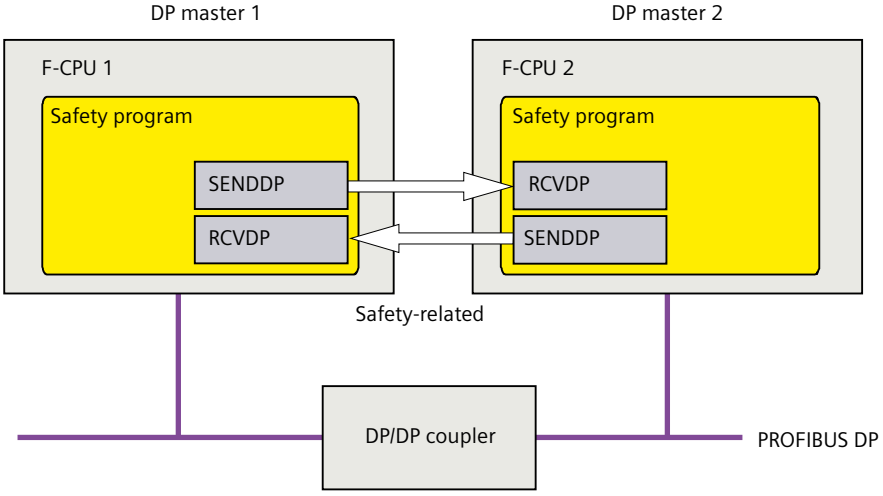


8. Select the following modules from the "Hardware catalog" task card (with filter activated) in the device view of DP/DP coupler PLC2, and insert them in the "Device overview" tab:
   - One "12 bytes I/6 bytes Q consistent" module, and
   - One "6 bytes I/12 bytes Q consistent" module



9.2.3.2 Program safety-related master-master communication

Communication via SENDDP and RCVDP instructions



Safety-related communication between the F-CPUs of the DP master uses the SENDDP and RCVDP instructions for sending and receiving, respectively. These can be used to perform a fail-safe transfer of a *fixed* amount of fail-safe data of the data type BOOL or INT (DINT as alternative).

You can find these instructions in the "Instructions" task card under "Communication". The RCVDP instruction **must** be called at the start of the main safety block. The SENDDP instruction **must** be called at the end of the main safety block.

You can also call the RCVDP and SENDDP instructions in separate F-FBs/F-FCs which you have to call up at the beginning or end of the main safety block.

Note that the send signals are not sent until after the SENDDP instruction call at the end of execution of the relevant F-runtime group.

You can find a detailed description of the SENDDP and RCVDP instructions in SENDDP and RCVDP: Send and receive data via PROFIBUS DP/PROFINET IO (STEP 7 Safety V21) (Page 612).

Requirement for programming

The address areas for input and output data for the DP/DP coupler must be configured.

Programming procedure

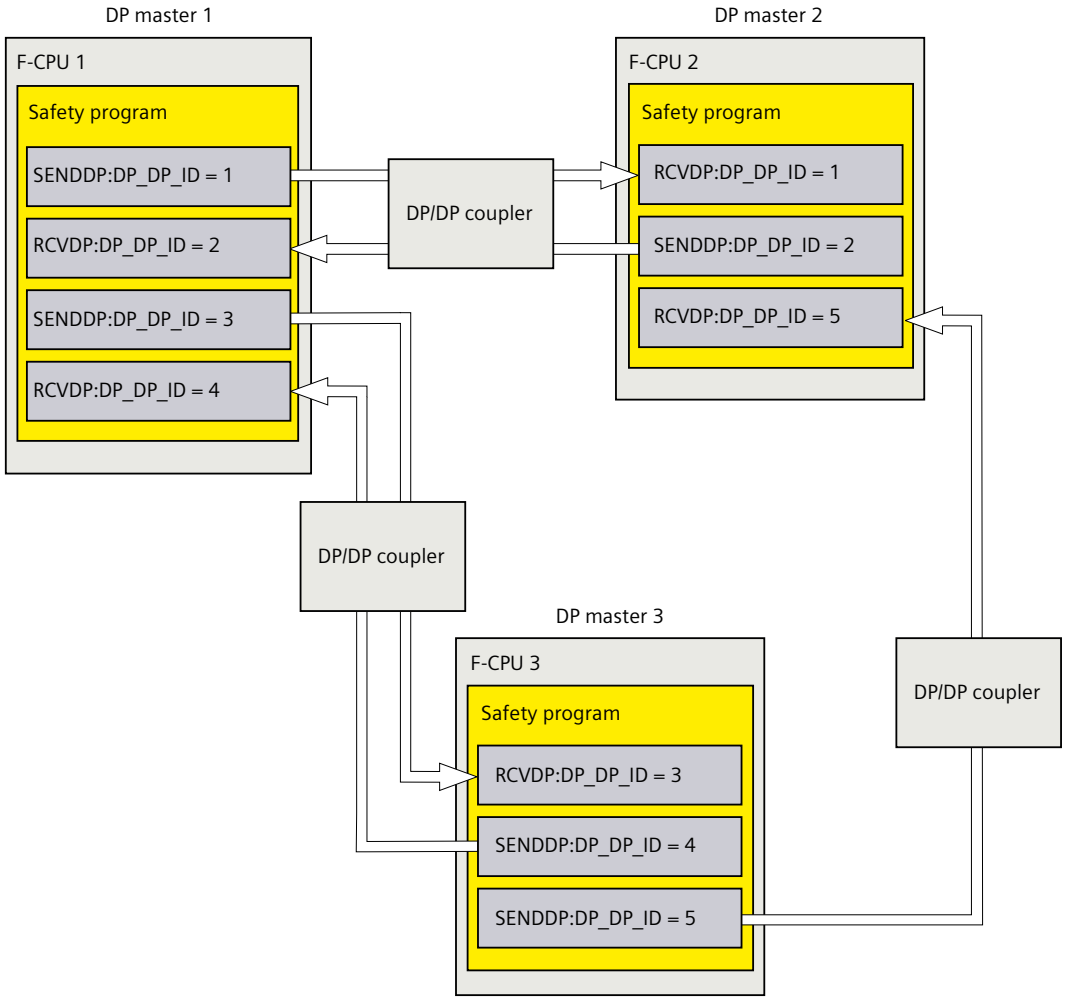
You program safety-related master-master communication as follows:

1. In the safety program from which data is to be sent, call the SENDDP instruction (Page 612) for sending at the end of the main safety block or a separate F-FC/F-FB.
2. In the safety program in which data is to be received, call the RCVDP instruction (Page 612) for receiving at the start of the main safety block or a separate F-FC/F-FB.
3. Assign the HW identifiers for the output and input data of the DP/DP coupler configured in the *hardware and network editor* (constant in the tag table) to the respective LADDR inputs.

   You must carry out this assignment for every communication connection for each of the F-CPUs involved.

4. Assign the value for the respective F-communication ID to the DP_DP_ID inputs. This establishes the communication relationship between the SENDDP instruction in one F-CPU and the RCVDP instruction in the other F-CPU: The associated instructions receive the same value for DP_DP_ID.

   The figure below contains an example of how to specify the F-communication IDs at the inputs of the SENDDP and RCVDP instructions for 5 safety-related master-master communications relationships.



WARNING

The value for the respective F-communication ID (input DP_DP_ID; data type: INT) can be freely selected**; however, it must be unique network-wide* and CPU-wide at all times for all safety-related communication connections. You must check the uniqueness in the safety summary during acceptance of the safety program.

You must supply constant values*** to the DP_DP_ID and LADDR inputs when calling the instruction. Direct write access to DP_DP_ID and LADDR in the associated instance DB is not permitted in the safety program! *(S016)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, no connection is established at the DP_DP_ID input for a F-communication ID "0".

*** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, the DP_DP_ID input can also be supplied with variable values from a global F-DB. In this case also, you have to check during acceptance of the safety program that uniqueness is ensured at all times. You need to check the algorithm for forming the variable value accordingly. If you cannot ensure a unique F-communication ID during startup of the safety program, because it is only specified after startup of the safety program, you must make sure that the value at the DP_DP_ID input is "0" during this phase.

5. Supply the SD_BO_xx and SD_I_xx inputs (SD_DI_00 as alternative) of SENDDP with the send signals. To cut down on intermediate signals when transferring block parameters, you can write the value directly to the instance DB of SENDDP using fully qualified access (for example, "Name SENDDP_1".SD_BO_02) before calling SENDDP.
6. Supply the RD_BO_xx and RD_I_xx outputs (RD_DI_00 as alternative) of RCVDP with the signals that you want to process further in other program sections or use fully qualified access to read the received signals directly in the associated instance DB in the program sections to be processed further (e.g., "Name RCVDP_1".RD_BO_02).
7. If you want to send the data at the SD_DI_00 input instead of the data at the SD_I_00 and SD_I_01 inputs, supply the DINTMODE input (initial value = "FALSE") of SENDDP with TRUE.
8. Supply the SUBBO_xx and SUBI_xx or alternatively SUBDI_00 inputs of RCVDP with the fail-safe values that are to be output by RCVDP in place of the process data until communication is established for the first time after startup of the sending and receiving F-systems or in the event of an error in safety-related communication.
   - Specification of constant fail-safe values:

     For data of data type INT/DINT, you can enter constant fail-safe values directly as constants in the SUBI_xx or alternatively SUBDI_00 input (initial value = "0"). If you want to specify a constant fail-safe value "TRUE" for data of the data type BOOL, set TRUE for the SUBBO_xx input (initial value = "FALSE").
   - Specification of variable substitute values:

     If you want to specify variable substitute values, define a tag that you calculate through your safety program in an F-DB and specify this tag (fully qualified) in the SUBBO_xx or SUBI_xx or alternatively SUBDI_00 input.

⚠ WARNING

Note: The program logic for calculating variable substitute values can only be inserted after the RCVDP calls, because there must be no program logic before the RCVDP calls. This is why the initial values of the substitute values are active in all RCVDP instructions in the first cycle after a startup of the F-system. You must therefore assign appropriate initial values for these tags. *(S017)*

9. Configure the TIMEOUT inputs of the RCVDP and SENDDP instructions with the required monitoring time.

   ⚠ WARNING

   It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time. *(S018)*

   Information on calculating the monitoring times can be found in Monitoring and response times (Page 629).

10. Optional: Evaluate the ACK_REQ output of the RCVDP instruction, for example, in the standard user program or on the operator control and monitoring system in order to query or to indicate whether user acknowledgment is required.
11. Supply the ACK_REI input of the RCVDP instruction with the acknowledgment signal for reintegration.
12. Optional: Evaluate the SUBS_ON output of the RCVDP or SENDDP instruction in order to query whether the RCVDP instruction is outputting the fail-safe values assigned in the SUBBO_xx and SUBI_xx or alternatively SUBDI_00 inputs.
13. Optional: Evaluate the ERROR output of the RCVDP or SENDDP instruction, for example, in the standard user program or on the operator control and monitoring system in order to query or to indicate whether a communication error has occurred.
14. Optional: Evaluate the SENDMODE output of the RCVDP instruction in order to query whether the F-CPU with the associated SENDDP instruction is in disabled safety mode (Page 393).

9.2.3.3 Safety-related master-master communication:Limits for data transfer

NOTE

If the data quantities to be transmitted exceed the capacity of the SENDDP / RCVDP correlated instructions, a second (or third) SENDDP / RCVDP call can be used. This requires configuration of an additional connection via the DP/DP coupler. Whether or not this is possible with one single DP/DP coupler depends on the capacity restrictions of the DP/DP coupler.

9.2.4 Safety-related IO controller-I-device communication

9.2.4.1 Configuring safety-related communication between IO controller and I-device

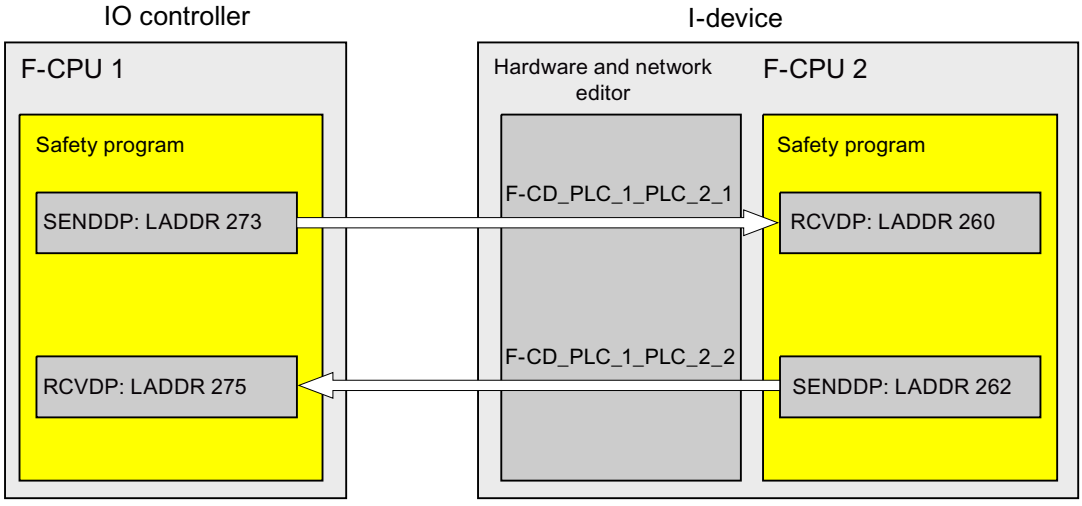
Introduction

Safety-related communication between the safety program of the F-CPU of an IO controller and the safety program(s) of the F-CPU(s) of one or more I-devices takes place via IO controller-I-device transfer areas (F-CD) in PROFINET IO, as in standard systems.

You do not need any additional hardware for IO controller-I-device communication.

Configuring transfer areas

For every safety-related communication connection between two F-CPUs, you must configure transfer areas in the *hardware and network editor*. The figure below shows how both of the F-CPUs are able to send and receive data (bidirectional communication).



The transfer area is assigned a label when it is created to identify it as the communication relationship. For example, "F-CD_PLC_2 PLC_1_1" for the first F-CD connection between IO controller F-CPU 1 and I-device F-CPU 2.

When you create a transfer area, a system constant with the name of the transfer area is created in the F-CPU of the IO controller and in the F-CPU of the I-device. The system constant contains the hardware identifier of the transfer area for the corresponding F-CPU.

You assign the hardware identifier (system constant from the default tag table) of the transfer areas symbolically to the LADDR input of the SENDDP and RCVDP instructions in the safety programs.

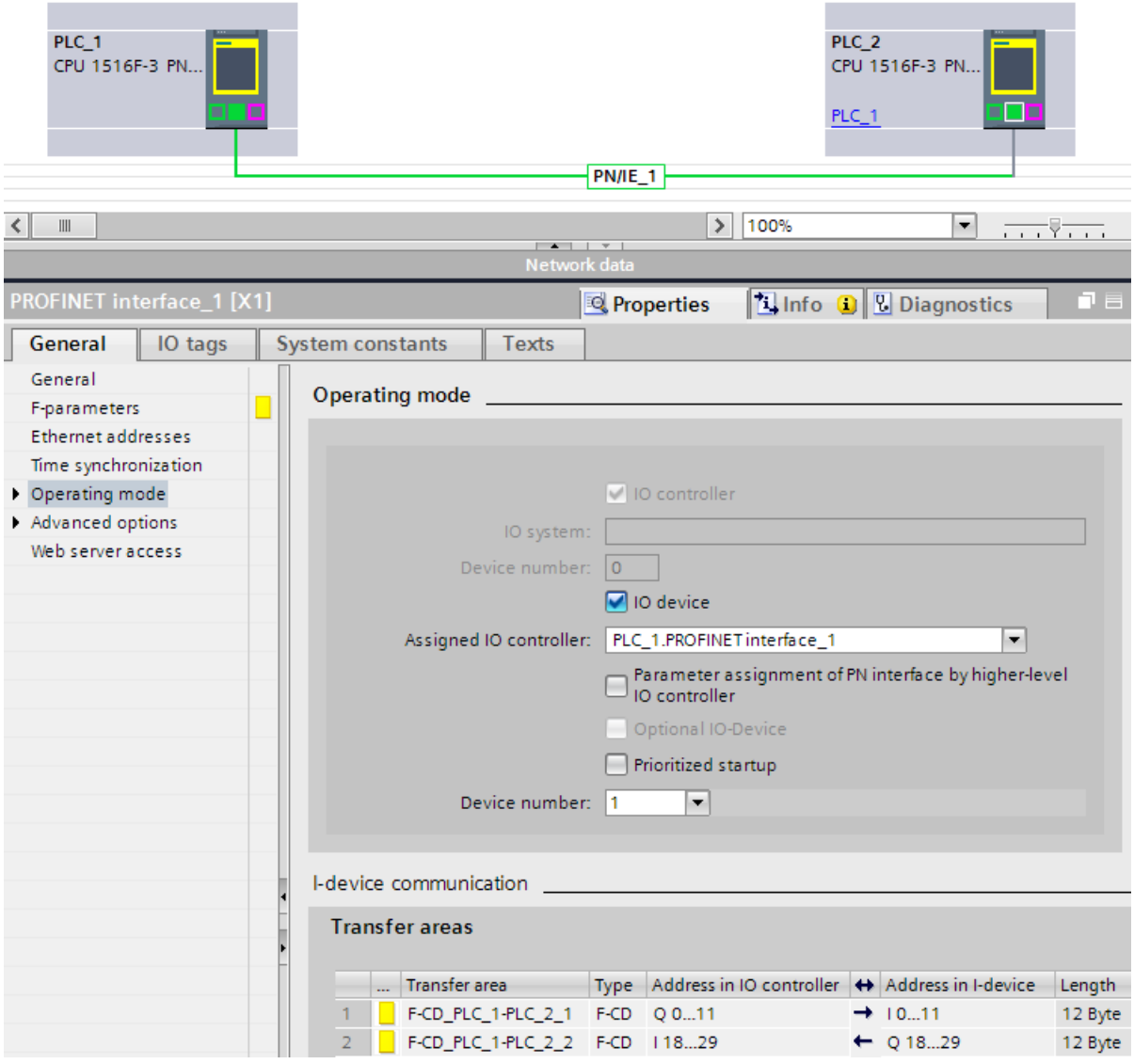
Procedure for configuration

The procedure for configuring safety-related IO controller-I-device communication is identical to that in the standard system.

Proceed as follows:

1. Insert two F-CPUS from the "Hardware catalog" task card into the project.
2. Enable the "IO Device" mode for F-CPU 2 in the properties of its PN interface and assign this PN interface to a PN interface of F-CPU 1.
3. Select the PROFINET interface of F-CPU 2. Under "Transfer areas", you create an F-CD connection (type "F-CD") for receiving from the IO controller (←). The F-CD connection is shown in yellow in the table and the address areas in the I-device and IO controller assigned are displayed.

   In addition, an acknowledgment connection is created automatically for each F-CD connection. (see "Transfer area details").
4. Create an additional F-CD connection for sending to the IO controller.
5. In the transfer area you just created, click on the arrow to change the transfer direction to sending to the IO controller (←).



NOTE

You can assign up to four IO controllers in separate projects to the shared I-device, as in standard systems. To do this, export the I-device as a GSD file and import the GSD file into the projects of the IO controllers. Proceed as described in the *STEP 7 help* under "Configuring an I-device as a shared device".

Configuration procedure for an S7-1500 F-CPU as of firmware V3.1 or S7-1200 G2 F-CPU as I-device

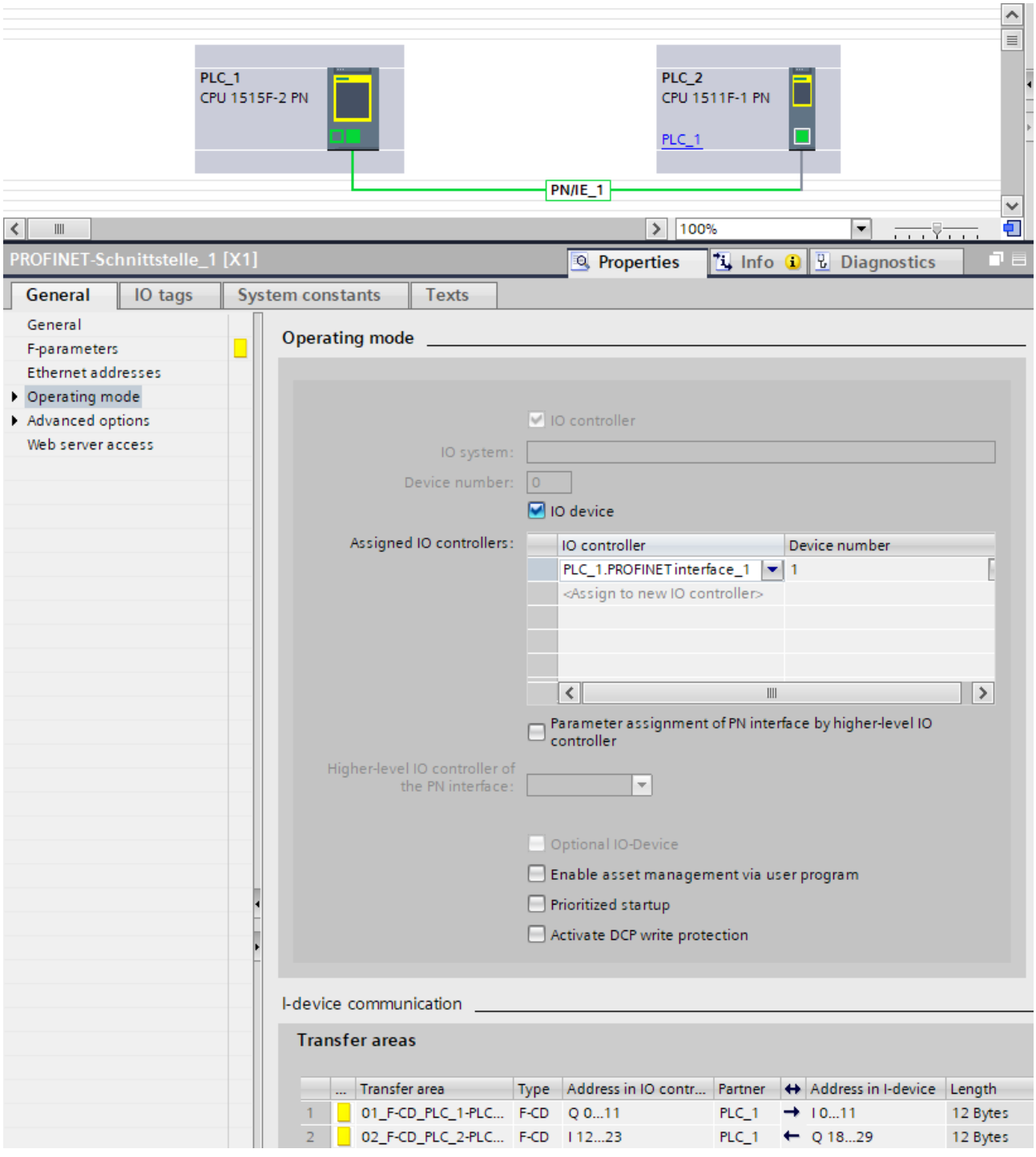
The procedure for configuring safety-related IO controller-I-device communication is identical to that in the standard system.

Proceed as follows:

1. From the "Hardware catalog" task card, insert any S7-1200/1500 F-CPU into the project as F-CPU 1.
2. From the "Hardware catalog" task card, insert an S7-1200 G2/S7-1500 F-CPU as of firmware V3.1 into the project as F-CPU 2 (I-device).
3. Enable the "IO Device" mode for F-CPU 2 in the properties of its PN interface and assign this PN interface to a PN interface of F-CPU 1.
4. Assign F-CPU 1 as an IO controller under "Assigned IO controllers".
5. Under "Transfer areas", you create an F-CD connection (type "F-CD") for receiving from the IO controller (←). The F-CD connection is shown in yellow in the table and the address areas in the I-device and IO controller assigned are displayed.

   In addition, an acknowledgment connection is created automatically for each F-CD connection. (see "Transfer area details").
6. Create an additional F-CD connection for sending to the IO controller.

7. In the transfer area you just created, click on the arrow to change the transfer direction to sending to the IO controller (←).



NOTE

Shared I-device

If you use S7-1500 F-CPUs as of firmware V3.1 or S7-1200 G2 F-CPUs as IO controller, you can divide different transfer areas of an I-device (S7-1500 F-CPU as of firmware V3.1 or S7-1200 G2 F-CPU) between two IO controllers within the project. As of *STEP 7 Safety V20*, you can divide between four IO controllers within the project, if the I-device is an S7-1500 F-CPU with firmware V3.1 or higher.

To do this, assign the I-device to a second IO controller under "Assigned IO controllers". Select the desired IO controller (partner) for each transfer area.

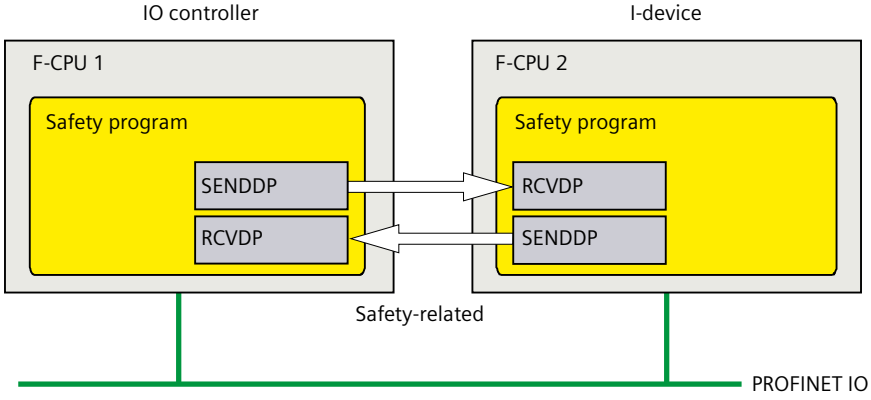
More information can be found in the *STEP 7 help*.

NOTE

You have the option to assign up to four additional IO controllers to the shared I-device, like in the standard. To do this, export the I-device as a GSD file and import the GSD file into the projects of the IO controllers. Proceed as described in the *STEP 7 help* under "Configuring an I-device as a shared device".

9.2.4.2 Programming safety-related IO controller I-device communication

Communication via SENDDP and RCVDP instructions



Safety-related communication between the F-CPUs of the IO controller and an I-device makes use of the SENDDP and RCVDP instructions for sending and receiving, respectively. These can be used to perform a fail-safe transfer of a *fixed* amount of data of the data type BOOL or INT (DINT as alternative).

You can find these instructions in the "Instructions" task card under "Communication". The RCVDP instruction **must** be called at the start of the main safety block. The SENDDP instruction **must** be called at the end of the main safety block.

You can also call the RCVDP and SENDDP instructions in separate F-FBs/F-FCs which you have to call up at the beginning or end of the main safety block.

Note that the send signals are not sent until after the SENDDP instruction call at the end of execution of the relevant F-runtime group.

You can find a detailed description of the SENDDP and RCVDP instructions in SENDDP and RCVDP: Send and receive data via PROFIBUS DP/PROFINET IO (STEP 7 Safety V21) (Page 612).

Requirement for programming

The transfer areas must be configured.

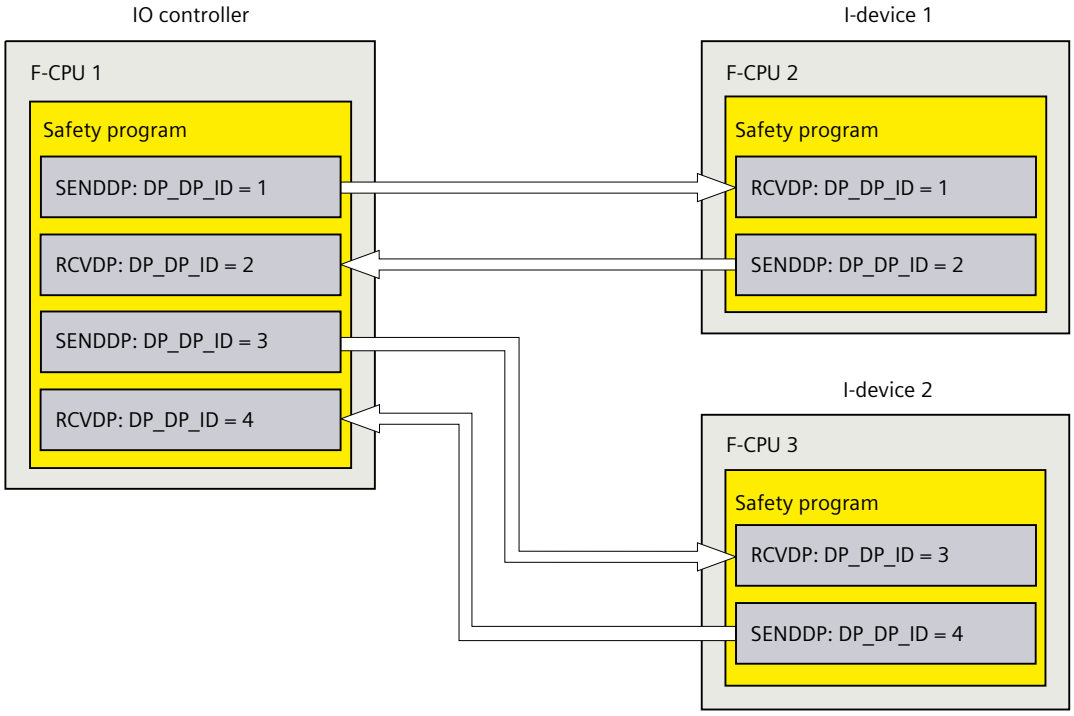
Programming procedure

The procedure for programming safety-related IO controller-I-device communication is the same as that for programming safety-related IO controller-IO controller communication (see Program safety-related IO controller-IO controller communication (Page 254)).

The assignment of the HW identifiers (system constants in the standard tag table) of the transfer areas to the LADDR input of the SENDDP/RCVDP instructions can be obtained from the following table:

Instruction	HW identifier
SENDDP in the IO controller	Hardware identifier of the transfer area in the IO controller
RCVDP in the IO controller	Hardware identifier of the transfer area in the IO controller
SENDDP in the I-device	Hardware identifier of the transfer area in the I-device
RCVDP in the I-device	Hardware identifier of the transfer area in the I-device

The figure below contains an example of how to specify the F-communication IDs for the inputs of the SENDDP and RCVDP instructions for 4 safety-related IO controller-I-device communication relationships.



⚠ WARNING

The value for the respective F-communication ID (input DP_DP_ID; data type: INT) can be freely selected**; however, it must be unique network-wide* and CPU-wide**** at all times for all safety-related communication connections. You must check the uniqueness in the safety summary during acceptance of the safety program.

You must supply constant values*** to the DP_DP_ID and LADDR inputs when calling the instruction. Direct write access to DP_DP_ID and LADDR in the associated instance DB is not permitted in the safety program! *(S016)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, no connection is established at the DP_DP_ID input for a F-communication ID "0".

*** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, the DP_DP_ID input can also be supplied with variable values from a global F-DB. In this case also, you have to check during acceptance of the safety program that uniqueness is ensured at all times. You need to check the algorithm for forming the variable value accordingly. If you cannot ensure a unique F-communication ID during startup of the safety program, because it is only specified after startup of the safety program, you must make sure that the value at the DP_DP_ID input is "0" during this phase.

**** With a redundant S7-1500HF system, both F-CPUs of the redundant S7-1500HF system are to be regarded as one single F-CPU with regard to the DP_DP_ID.

WARNING

It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time.*(S018)*

Information on calculating the monitoring times can be found in Monitoring and response times (Page 629).

9.2.4.3 Safety-related IO-Controller-IO-Device communication - Limits for data transfer

Limits for data transfer

If the amount of data to be transferred is greater than the capacity of related SENDDP/RCVDP instructions, you can use additional SENDDP/RCVDP instructions. Configure additional transfer areas for this purpose. Remember the maximum limit of 1440 bytes of input data or 1440 bytes of output data for transfer between an I-device and a IO controller.

The following table shows the amount of output and input data assigned in safety-related communication connections:

Safety-related communication	Communication connection	Assigned input and output data			
		In the IO controller		In the I-device	
		Output data	Input data	Output data	Input data
IO controller-I-Device	**Sending:** I-Device 1 to IO controller	6 bytes	12 bytes	12 bytes	6 bytes
	Receiving: I-Device 1 from IO controller	12 bytes	6 bytes	6 bytes	12 bytes

Consider all additional configured safety-related and standard communication connections (transfer areas of type F-CD and CD) for the maximum limit of 1440 bytes of input data or 1440 bytes of output data for transfer between an I-device and an IO controller. In addition, data are assigned for internal purposes such that the maximum limit may be reached sooner.

When the limit is exceeded, a corresponding error message is displayed.

You can find more information on the transfer areas in the *STEP 7 help* under "Rules for creating transfer areas".

9.2.5 Safety-related master-I-slave communication

9.2.5.1 Configuring safety-related master-I-slave communication

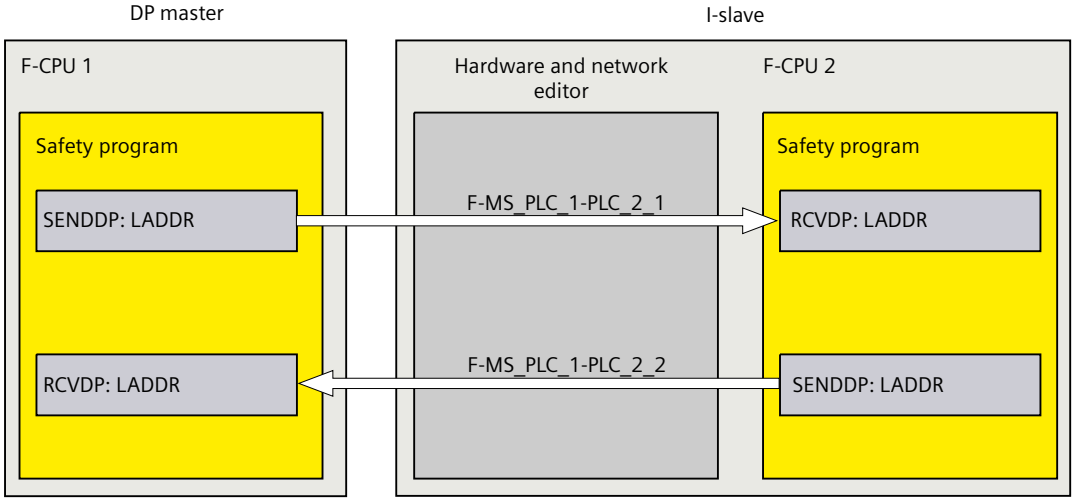
Introduction

Safety-related communication between the safety program of the F-CPU of a DP master and the safety program(s) of the F-CPU(s) of one or more I-slaves takes place via master-I-slave transfer areas (**F-**MS), as in standard systems.

You do not need a DP/DP coupler for master-I-slave communication.

Configuring transfer areas

For every safety-related communication connection between two F-CPUs, you must configure transfer areas in the *hardware and network editor*. The figure below shows how both of the F-CPUs are able to send and receive data (bidirectional communication).



The transfer area is assigned a label when it is created to identify it as the communication relationship. For example, "F-MS_PLC_2-PLC_1_1" for the first F-MS connection between DP master F-CPU 1 and I-slave F-CPU 2.

When you create a transfer area, a system constant with the name of the transfer area is created in the F-CPU of the DP master and in the F-CPU of the I-slave. The system constant contains the hardware identifier of the transfer area for the corresponding F-CPU.

You assign the hardware identifier (system constant from the default tag table) of the transfer areas symbolically to the LADDR input of the SENDDP and RCVDP instructions in the safety programs.

Procedure for configuration

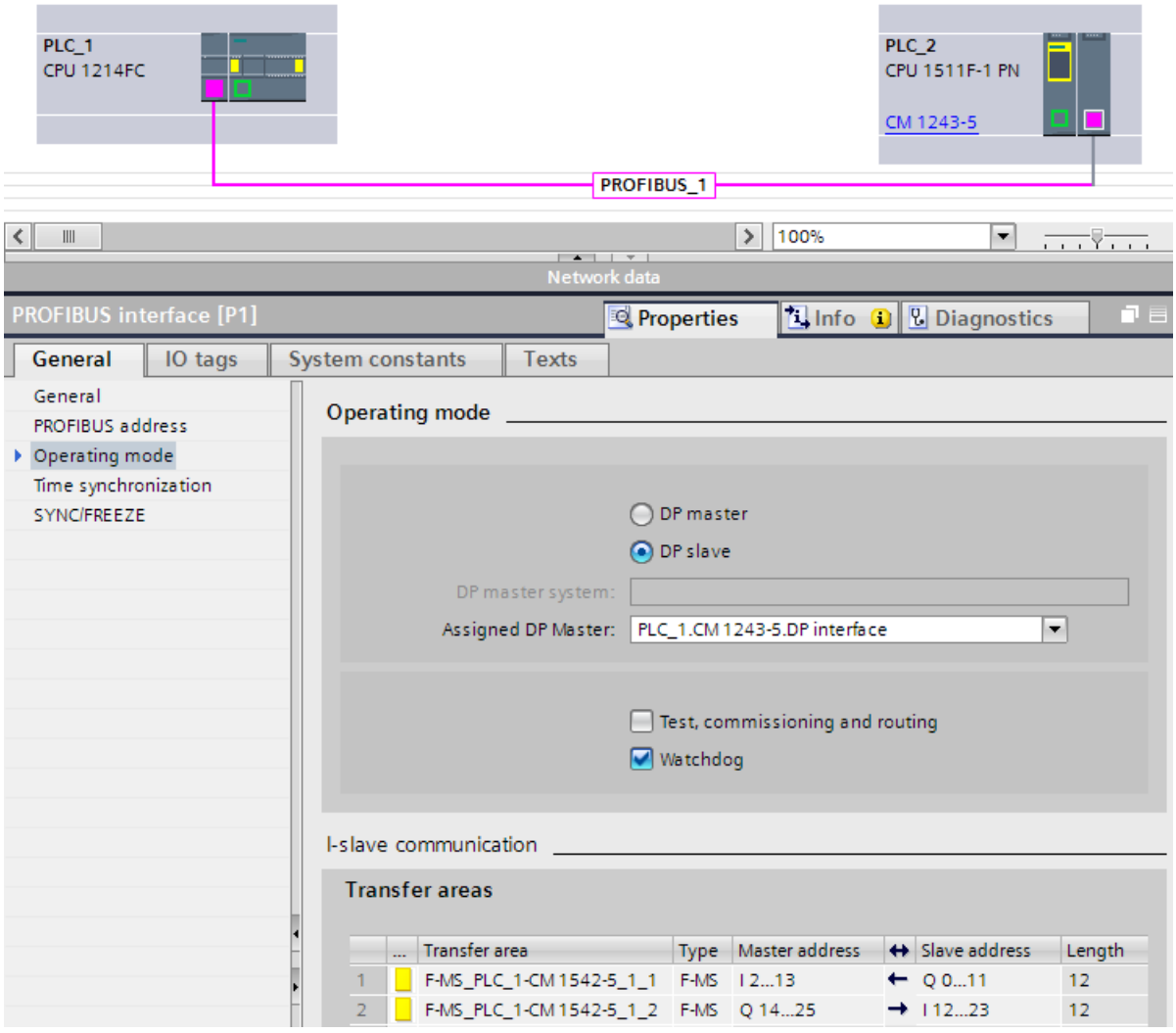
The procedure for configuring safety-related master-I-slave communication is identical to that in the standard system.

Proceed as follows:

1. Insert two F-CPUS from the "Hardware catalog" task card into the project.
2. If the F-CPU which is to be operated as DP master (F-CPU 1) does not have an integrated PROFIBUS DP interface, insert an PROFIBUS-CM, for example.
3. From the device view of the F-CPUs which are to be operated as I-slaves (F-CPU 2), insert a suitable CM DP module or CP DP module.
4. If necessary, enable "DP-slave" (I-slave) mode in the properties for the CM/CP DP module.
5. Assign the DP interface of the CM/CP to a DP interface of F-CPU 1.
6. Select the PROFIBUS interface of F-CPU 2 or of the CM. Under "Transfer areas", you create an F-MS connection (type "F-MS") for sending to the DP master (←). The F-MS connection is shown in yellow in the table and the assigned transfer areas in the I-slave and DP master are displayed.

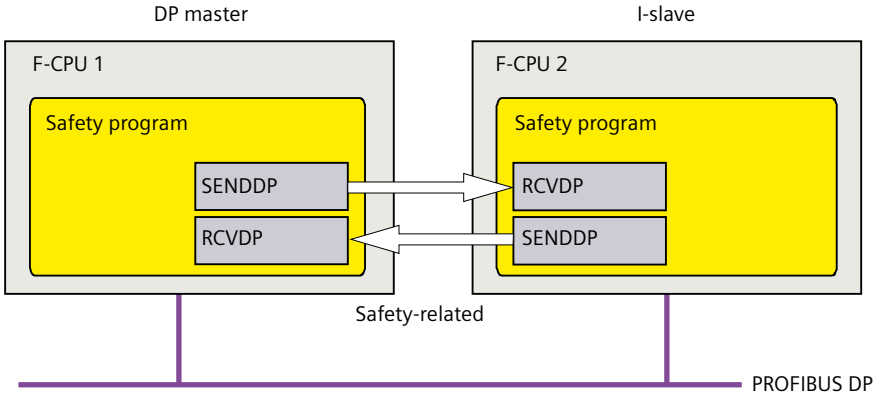
   In addition, an acknowledgment connection is created automatically for each F-MS connection. (see "Transfer area details").
7. Create an additional F-MS connection for receiving from the DP master.

8. In the transfer area you just created, click the arrow in order to change the transfer direction to receiving from DP master (→).



9.2.5.2 Programming safety-related master-I-slave communication

Communication via SENDDP and RCVDP instructions



Safety-related communication between the F-CPUs of the DP master and an I-slave makes use of the SENDDP and RCVDP instructions for sending and receiving, respectively. These can be used to perform a fail-safe transfer of a *fixed* amount of fail-safe data of the data type BOOL or INT (DINT as alternative).

You can find these instructions in the "Instructions" task card under "Communication". The RCVDP instruction **must** be called at the start of the main safety block. The SENDDP instruction **must** be called at the end of the main safety block.

You can also call the RCVDP and SENDDP instructions in separate F-FBs/F-FCs which you have to call up at the beginning or end of the main safety block.

Note that the send signals are not sent until after the SENDDP instruction call at the end of execution of the relevant F-runtime group.

You can find a detailed description of the SENDDP and RCVDP instructions in SENDDP and RCVDP: Send and receive data via PROFIBUS DP/PROFINET IO (STEP 7 Safety V21) (Page 612).

Requirements

The transfer areas must be configured.

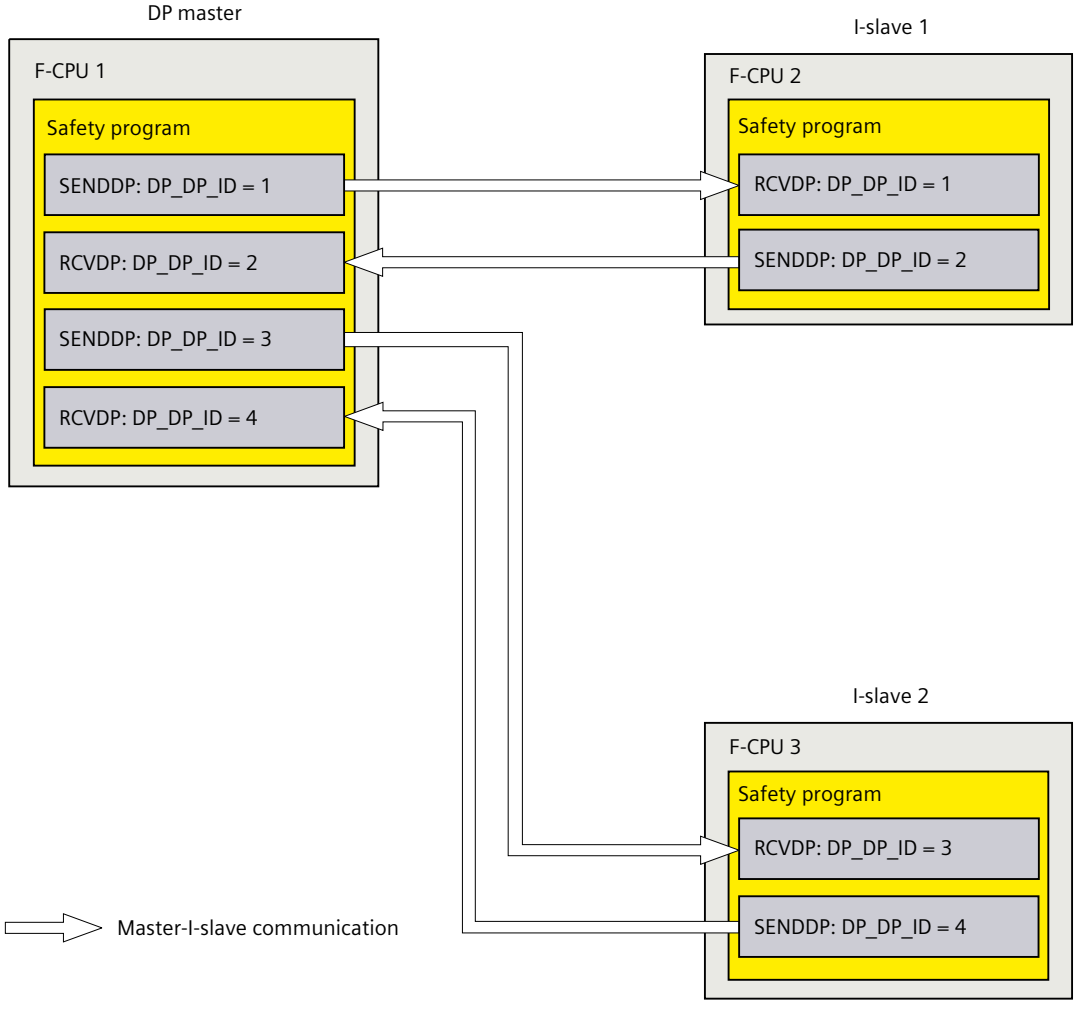
Programming procedure

The procedure for programming safety-related master-I-slave communication is the same as that for programming safety-related master-master communication (see Safety-related master-master communication (Page 259)).

The assignment of the HW identifiers of the transfer areas to the LADDR input of the SENDDP/RCVDP instructions can be obtained from the following table.

Instruction	HW identifier
SENDDP in the DP master	Hardware identifier of the respective transfer area in the DP master
RCVDP in the DP master	Hardware identifier of the respective transfer area in the DP master
SENDDP in the I-slave	Hardware identifier of the transfer area in the I-slave
RCVDP in the I-slave	Hardware identifier of the transfer area in the I-slave

The figure below contains an example of how to specify the F-communication IDs at the inputs of the SENDDP and RCVDP instructions for four safety-related master-I-slave communication relationships.



⚠ WARNING

The value for the respective F-communication ID (input DP_DP_ID; data type: INT) can be freely selected**; however, it must be unique network-wide* and CPU-wide at all times for all safety-related communication connections. You must check the uniqueness in the safety summary during acceptance of the safety program.

You must supply constant values*** to the DP_DP_ID and LADDR inputs when calling the instruction. Direct write access to DP_DP_ID and LADDR in the associated instance DB is not permitted in the safety program! *(S016)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, no connection is established at the DP_DP_ID input for a F-communication ID "0".

*** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, the DP_DP_ID input can also be supplied with variable values from a global F-DB. In this case also, you have to check during acceptance of the safety program that uniqueness is ensured at all times. You need to check the algorithm for forming the variable value accordingly. If you cannot ensure a unique F-communication ID during startup of the safety program, because it is only specified after startup of the safety program, you must make sure that the value at the DP_DP_ID input is "0" during this phase.

⚠ WARNING

It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time. *(S018)*

Information on calculating the monitoring times can be found in Monitoring and response times (Page 629).

9.2.5.3 Limits for data transfer of safety-related master-I-slave communication

Limits for data transfer

If the amount of data to be transferred is greater than the capacity of related SENDDP/RCVDP instructions, you can use additional SENDDP/RCVDP instructions. Configure additional transfer areas for this purpose. Note the maximum limit of 244 bytes of input data or 244 bytes of output data for transfer between an I-slave and a DP master.

The following table shows the amount of output and input data assigned in safety-related communication connections:

Safety-related communication	Communication connection	Assigned input and output data			
		DP master		I-slave	
		Output data	Input data	Output data	Input data
Master-I-slave	**Sending:** I-slave 1 to DP master	6 bytes	12 bytes	12 bytes	6 bytes
	Receiving: I-slave 1 from DP master	12 bytes	6 bytes	6 bytes	12 bytes

Consider all additional configured safety-related and standard communication connections (transfer areas of type F-MS-, F-DX-, F-DX-Mod., MS-, DX- and DX-Mod) for the maximum limit of 244 bytes of input data or 244 bytes of output data for transfer between an I-device and a DP master F-MS and MS). If the maximum limit of 244 bytes of input data or 244 bytes of output data is exceeded, you will receive a corresponding error message.

9.2.6 Safety-related IO controller-I-slave communication

Introduction

Safety-related communication between the safety program of the F-CPU of an IO controller and the safety program(s) of the F-CPU(s) of one or more I-slaves takes place via master-I-slave transfer areas (**F-**MS), as in standard systems.

IE/PB link

For the safety-related IO-controller-I-slave communication, the IE/PB link is mandatory. Each of the two F-CPUs is linked to the IE/PB link by means of its PROFIBUS DP or PROFINET-interface.

NOTE

If you are using an IE/PB link, you must take this into account when configuring the F-specific monitoring times and when calculating the maximum response time of your F-system (see also Monitoring and response times (Page 629)).

Note that the Excel file for calculating response times (https://support.industry.siemens.com/cs/ww/en/view/109783831) for S7-300/400 F-CPUs does not support all conceivable configurations.

Reference

The information on safety-related master-I-slave communication in Safety-related master-I-slave communication (Page 275) also applies.

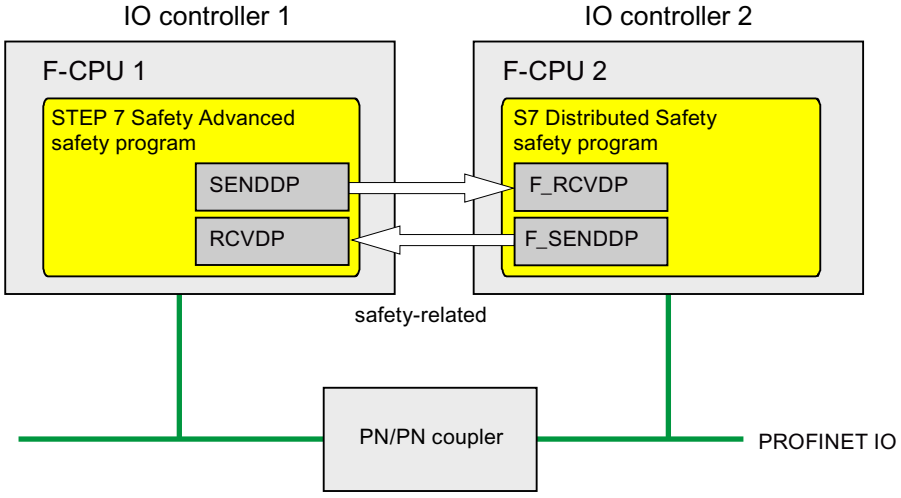
9.2.7 Safety-related communication to S7 F-System S7 Distributed Safety

Introduction

Safety-related communication from F-CPUs in SIMATIC Safety to F-CPUs in S7 Distributed Safety F-systems is possible, via a PN/PN coupler or DP/DP coupler that you use between the two F-CPUs, as IO controller-IO controller communication or master-master communication.

9.2.7.1 Communication with S7 Distributed Safety via PN/PN coupler (IO controller-IO controller communication)

Communication functions between SENDDP/RCVDP instructions at the *STEP 7 Safety* end and F-application blocks F_SENDDP/F_RCVDP at the *S7 Distributed Safety* end:



Procedure at the *S7 Distributed Safety* end

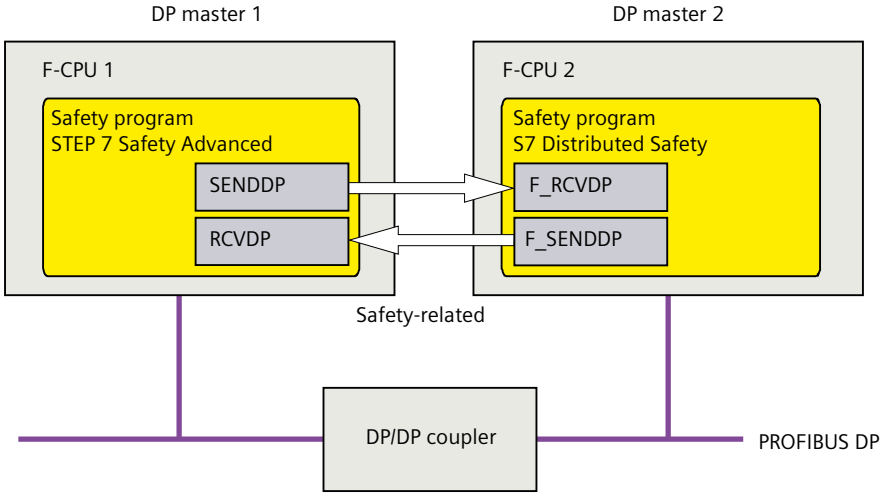
At the *S7 Distributed Safety* end, proceed as described in "Safety-related IO controller-IO controller communication" in the S7 Distributed Safety - Configuring and Programming (https://support.industry.siemens.com/cs/ww/en/view/22099875) manual.

Procedure at the *STEP 7 Safety* end

At the *STEP 7 Safety* end, proceed as described in Safety-related IO controller-IO controller communication (Page 251).

9.2.7.2 Communication with S7 Distributed Safety via DP/DP coupler (master-master communication)

Communication functions between SENDDP/RCVDP instructions at the *STEP 7 Safety* end and F-application blocks F_SENDDP/F_RCVDP at the *S7 Distributed Safety* end:



Procedure at the *S7 Distributed Safety* end

At the *S7 Distributed Safety* end, proceed as described in the section "Safety-related master-master communication" of the "S7 Distributed Safety - Configuring and Programming (https://support.industry.siemens.com/cs/ww/en/view/22099875)" manual.

Procedure at the *STEP 7 Safety* end

At the *STEP 7 Safety* end, proceed as described in Safety-related master-master communication (Page 259).

9.3 Communication via SENDDP/RCVDP instructions between S7-300/400 and S7-1200/1500 F-CPUs

Introduction

This section provides an overview of the options for safety-related communication using SENDDP/RCVDP instructions between S7-300/400 F-CPUs and S7-1200/1500 F-CPUs in SIMATIC Safety F-systems.

Options for safety-related CPU-CPU communication

Safety-related communication	On subnet	Additional hardware required	Required instructions
Master-master communication	PROFIBUS DP	DP/DP coupler	SENDDP/RCVDP
Master-I-slave communication	PROFIBUS DP	—	SENDDP/RCVDP
IO controller-IO controller communication	PROFINET IO	PN/PN coupler	SENDDP/RCVDP
IO controller-I-device communication	PROFINET IO	—	SENDDP/RCVDP
IO controller-I-slave communication	PROFINET IO and PROFIBUS DP	IE/PB link	SENDDP/RCVDP

Basic procedure for configuring and programming

Configure and program safety-related communication between S7-300/400 F-CPUs and S7-1200/1500 F-CPUs as described in Communication via SENDDP/RCVDP or SENDS7/RCVS7(S7-300, S7-400) instructions: (Page 202) and Communication via SENDDP/RCVDP instructions (S7-1200, S7-1500): (Page 248) for your application.

To program an S7-300/400 F-CPU, use the start addresses of the transfer areas. To program an S7-1200/1500 F-CPU, use the HW identifiers of the transfer areas.

9.4 Communication via SENDDP/RCVDP instructions: IO controller and I-device in different projects

9.4.1 Configuring safety-related communication between IO controller and I-device

Introduction

Safety-related communication between the safety program of the F-CPU of an IO controller and the safety program(s) of the F-CPU(s) of one or more I-devices takes place via IO controller-I-device transfer areas (F-CD) in PROFINET IO, as in standard systems.

The following section describes particular aspects when the IO Controller and the I-device are located in different projects.

Requirement

- The IO Controller is an S7-1200/1500 F-CPU that supports the IO Controller functionality.
- The I-device is an S7-300/400/1200/1500 F-CPU that supports the I-device functionality.
- The project in which the I-device is located, must have been created with *S7 Distributed Safety V5.4*, *STEP 7 Safety V13* or later.

Configuring

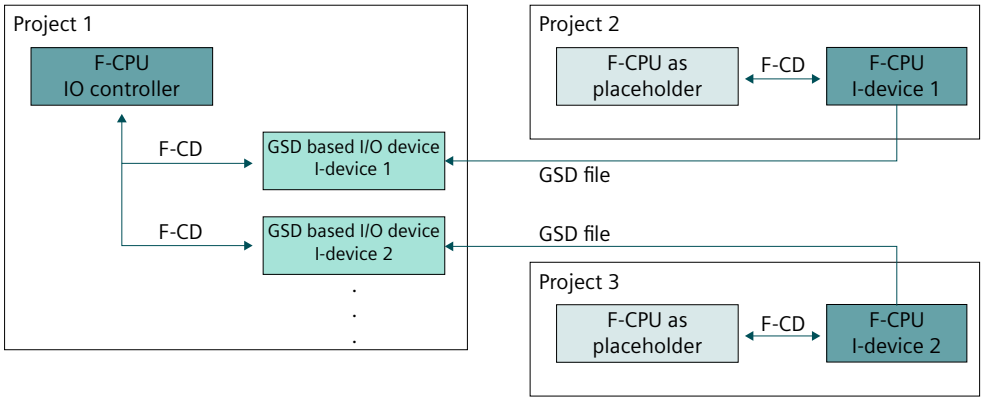
1. Configure the safety-related communication in the project with the I-device as described under "Configuring safety-related communication between IO controller and I-device (Page 220)" (S7-300, S7-400) or "Configuring safety-related communication between IO controller and I-device (Page 267)" (S7-1200, S7-1500) respectively. In this case the F-CPU 1 (IO Controller) is only a placeholder for the F-CPU in the project of the IO Controller.

   NOTE

   When creating with *STEP 7 Safety* < V14 SP1 avoid a subsequent change from the transfer areas from CD to F-CD.

   When creating with *S7 Distributed Safety V5.4* create the application transfer areas of the address type "Output" and "Input" directly after each other.

2. Export the I-device as a GSD file. Proceed as described in the *STEP 7* help under "Configuring an I-device".
3. Import the GSD file in the project with the IO Controller. Proceed as described in the *STEP 7* help under "Installing a GSD file".
4. Insert the I-device from the "Hardware catalog" task card into the project with the IO Controller.
5. Assign the F-CPU of the IO Controller to the I-device.



9.4.2 Programming safety-related IO Controller I-device communication

Programming procedure

To program the safety-related communication between the IO controller and I-device for an F-CPU S7-300/400, analogously follow the procedure described under "Programming safety-related IO controller I-device communication (Page 223)". To program an S7-300/400 F-CPU, use the start addresses of the transfer areas.

To program the safety-related communication between the IO controller and I-device for an F-CPU S7-1200/1500, analogously follow the procedure described under "Programming safety-related IO controller I-device communication (Page 271)". To program an S7-1200/1500 F-CPU, use the HW identifiers of the transfer areas.

9.5 Communication with OPC UA Safety (S7-1500)

With OPC UA Safety, you can exchange a defined data structure between an F-CPU S7-1500 and another F-CPU S7-1500 F-CPU or a device of a third-party manufacturer in a safety-related manner that supports OPC UA Safety. Optionally, you can also transfer non-safety-related data.

OPC UA Safety enables the configuration of static and dynamic connections. In the case of static connections, the communication relationship is already specified in the TIA Portal. In the case of dynamic connections, the communication relationship is only specified during runtime.

OPC UA Safety is specified by the OPC Foundation in OPC UA Specification OPC 10000-15 and is based on the mechanisms of OPC UA. SIMATIC Safety supports OPC UA client/server communication. More information can be found in the *STEP 7* help under "OPC UA communication".

For OPC UA Safety, the OPC Foundation provides a basic information model with the data types and methods needed for OPC UA Safety via a nodeset.

Based on this nodeset, you create the information model for your individual data structure as a nodeset as well.

OPC UA Safety designates a server interface that makes the safety-related data available as a SafetyProvider, and a corresponding client interface that reads the safety-oriented data from a SafetyProvider as a SafetyConsumer.

For a bidirectional communication connection, each communication partner must implement a SafetyProvider and a SafetyConsumer.

OPC UA Safety communication requires special parameter assignments, which you make using the Safety Parameter Interface (SPI). For communication with your Safety program or standard user program, the Safety Application Program Interface (SAPI) is available.

The following sections describe the Safety Parameter Interface (SPI) and the Safety Application Program Interface (SAPI) and the configuration and programming actions needed for an S7-1500 F-CPU for OPC UA Safety communication. All descriptions as well as the implementation of OPC UA Safety in the SIMATIC Safety F-system are based on the Version 1.05.04 of the OPC UA specification OPC 10000-15. For information on the procedure for devices of a third-party manufacturer, refer to the documentation of the device manufacturer.

NOTE

During simulation with *S7-PLCSIM*, the time monitoring that generates an error message after expiration when communication with real communication partners is interrupted (e.g. by setting a CPU to STOP) is not activated. This is why no error message is displayed in this case.

NOTE

The OPC UA Specification OPC 10000-15 specifies for the transfer of values of the BOOL data type that these be transferred in Byte format.

- "TRUE" is transferred as "0x01"
- "FALSE" is transferred as "0x00"

Since OPC UA Safety does not provide qualifier bits, they must be modeled, transferred and evaluated as part of the safety-related user data if necessary. The OPC UA Specification OPC 10000-15 prescribes the following semantics:

- "TRUE" or "0x01" corresponds to the "valid value" state
- "FALSE" or "0x00" corresponds to the "invalid value" state

Restrictions

The following *STEP 7 Safety* functions cannot be used together with OPC UA Safety:

- Safety program in RUN mode (S7-1200, S7-1500) (Page 406) (Fast Commissioning)
- Safety program in Software Unit (Safety Unit) (S7-1500) (Page 116)
- Consistent upload from the F-CPU (Page 91)

9.5.1 Configuration and programming interfaces

Introduction

A Safety Parameter Interface (SPI) and Safety Application Program Interface (SAPI) is available for each SafetyConsumer or SafetyProvider.

You must make special parameter assignments, which are required for the safety-related data transfer, in the SPI for each OPC UA Safety communication. To facilitate this, values are automatically assigned to the parameters whenever possible.

The SAPI forms the interface to your Safety program (or standard user program).

9.5.1.1 Safety Parameter Interface (SPI) of the SafetyProvider

The SPI of a SafetyProvider is shown in the properties of the SafetyProvider in the editor for server interfaces either during the creation of the information model with the special OPC UA modeling tool SiOME (Page 308) or later after import of the information model into TIA Portal (Page 313).

The following table shows the parameters of the SPI of a SafetyProvider.

Parameter	Data type	Changeable in TIA Portal	Description
SafetyProviderIDConfigured	UDINT	x	ID for authentication of a SafetyProvider
SafetyBaseIDConfigured	GUID	x	ID for authentication of a SafetyProvider
SafetyProviderLevel	BYTE	-	Safety Integrity Level (SIL) that can be achieved with the specific implementation of the SafetyProvider
SafetyStructureSignature	UDINT	-	Signature via the data structure of the OPC UA data type of the fail-safe data to be transferred
SafetyStructureSignatureVersion	UINT	-	Version of the algorithm for forming the SafetyStructureSignature
SafetyStructureIdentifier	STRING	-	Identifier of the OPC UA data types of the fail-safe data to be transferred (max. length 128 characters)
SafetyProviderDelay	UDINT	-	Maximum time in µs between the request of a SafetyConsumer and the response of a Safety-Provider
SafetyServerImplemented	BOOL	-	1=SafetyProvider supports OPC UA Client/Server communication
SafetyPubSubImplemented	BOOL	-	1=SafetyProvider supports OPC UA PubSub communication

The values of the SPI parameters of a SafetyProvider are always assigned automatically during import of the information model (Page 308) if a value has not yet been entered for a parameter in the information model. Previously entered values are applied if permissible. If there are impermissible values, an error message is output.

SafetyProviderIDConfigured/SafetyBaseIDConfigured

The SafetyProviderIDConfigured and SafetyBaseIDConfigured SPI parameters are used to uniquely and safely authenticate a SafetyProvider. For applications with changing communication partners (Page 332) or series machines (Page 331), you can overwrite the values of these SPI parameters using the SAPI parameters SafetyProviderID and SafetyBaseID (Page 291). The value that is actually in effect during runtime of the Safety program is referred to in this document as the "active SafetyBaseID" or "active SafetyProviderID".
Different combinations are conceivable with regard to the unique authentication:

- The active SafetyBaseID is identical for all SafetyProviders within a certain area of the system. The unique authentication within this plant area must take place through different active SafetyProviderIDs.
- The active SafetyProviderID is identical for all SafetyProviders. The unique authentication must then take place solely through different active SafetyBaseIDs.
- Neither the active SafetyProviderIDs nor the active SafetyBase IDs of all SafetyProviders within a specific area are identical. The unique authentication only takes place through different combinations of active SafetyProviderIDs and active SafetyBaseIDs.

In the case of automatic assignment, the following values are entered. The uniqueness is achieved solely using the SafetyBaseIDConfigured:

- **SafetyProviderIDConfigured:** If no local data mapping is available for the SAPI parameter SafetyProviderID, the value "1" is assigned. Otherwise, the value "0" is assigned.
- **SafetyBaseIDConfigured:** If no local data mapping is available for the SAPI parameter SafetyBaseID, a randomly generated value is assigned that fulfills the safety requirements of OPC UA Specification OPC 10000-15. Otherwise, the value "0" is assigned.

⚠ WARNING

For the unique and secure authentication of a SafetyProvider, the combination of active SafetyProviderID and active SafetyBaseID at any point in time must be unique for all OPC UA Safety communication connections network-wide. **

If your network is not safely separated from the outside world through physical separation, the active SafetyBaseID at any point in time must be a value that was generated randomly in accordance with the OPC UA Specification OPC 10000-15. ***

You must ensure the uniqueness in the safety summary during acceptance of the safety program. If the active SafetyProviderID or active SafetyBaseID is not formed until runtime of your Safety program, you must ensure the correct implementation of the algorithm for the generation of the values. *(S102)*

* If you cannot ensure a unique combination during startup of the Safety program because the active SafetyProviderID or active SafetyBaseID is not formed until runtime of your Safety program, you must ensure that the active SafetyProviderID or active SafetyBaseID is "0" during this phase.

** SafetyProviders to which the same local data element of your safety program is mapped (Safety Multicast (Page 330)) at the SafetyData SAPI parameter are permitted to use the same combination.

*** If you have entered a value for the SPI parameter SafetyBaseIDConfigured of a SafetyProvider (either in the NodeSet or in the properties of a SafetyProvider), you yourself must ensure that the value meets the safety requirements of the OPC UA Specification OPC 10000-15. Alternatively, generate the value of SafetyBaseIDConfigured again by opening the properties tab of the affected SafetyProvider and clicking on the "Generate" button.

⚠ WARNING

If you have not defined any value for the SPI parameter of a SafetyProvider in the NodeSet for the SPI parameter, a sufficiently clear UUIDV4 is made available from the system as a value for the parameter of the respective SafetyProvider when importing the NodeSet into the TIA Portal.

Through subsequent copying of a SafetyProvider (e.g. copying the NodeSet to another F-CPU, copying the F-CPU or copying the project), values provided by the system are not generated again and are therefore not unique anymore.

If such a copy is used to configure a new communication relationship, you yourself need to ensure the uniqueness. To do this, open the properties tab of all affected SafetyProvider and generate the value of SafetyBaseIDConfigured again by clicking on the "Generate" button. The uniqueness must be checked in the safety summary during acceptance. *(S110)*

SafetyProviderLevel

The SafetyProviderLevel SPI parameter is used to publicize the Safety Integrity Level (SIL) that can be achieved by a specific SafetyProvider implementation.

With automatic assignment, the value "3" is entered. This corresponds to the maximum SIL achievable with SIMATIC Safety and is the only permissible value.

SafetyStructureSignature

The SafetyStructureSignature SPI parameter is used to protect and identify the data structure at the SafetyData of the OPC UA data type of the SAPI parameters via a signature.

With automatic assignment, the value is calculated according to the algorithm prescribed in OPC UA Specification OPC 10000-15.

SafetyStructureSignatureVersion

The SafetyStructureSignatureVersion SPI parameter is used to specify the version of the algorithm used to calculate the SafetyStructureSignature.

With automatic assignment, the value "1" is entered. This corresponds to the only permissible value at the present time.

SafetyStructureIdentifier

With the SPI parameter SafetyStructureIdentifier the identifier of the OPC UA data types of the SAPI parameter SafetyData is publicized. The SafetyStructureIdentifier is included in the calculation of the SafetyStructureSignature.

With automatic assignment, the name of the OPC UA data type that describes the data structure at the SafetyData SAPI parameter is entered.

SafetyProviderDelay

The SafetyProviderDelay SPI parameter is used to publicize the maximum time (in µs) that can elapse between the reception of the request of a SafetyConsumer and the sending of the response of a SafetyProvider under worst-case conditions. For more details, refer to OPC UA Specification OPC 10000-15.

In the case of automatic assignment, the value "0" is entered at the present time.

SafetyServerImplemented

The SafetyServerImplemented SPI parameter is used to publicize whether the SafetyProvider supports OPC UA Client/Server communication mechanisms.

In the case of automatic assignment, the value "TRUE" is entered. This corresponds to the only permissible value since SIMATIC Safety supports only OPC UA client/server communication.

SafetyPubSubImplemented

The SafetyPubSubImplemented SPI parameter is used to publicize whether the SafetyProvider supports OPC UA PubSub communication mechanisms.

With automatic assignment, the value "FALSE" is entered. This corresponds to the only permissible value since SIMATIC Safety does not support OPC UA PubSub communication.

9.5.1.2 Safety Application Program Interface (SAPI) of the SafetyProvider

The SAPI of a SafetyProvider is shown in the properties of the SafetyProvider in the editor for server interfaces either when the information model is created in SiOME (Page 308) or after the information model is imported into TIA Portal (Page 313). These properties are stored in the internal F-OPC-UA-Provider-DB. The name is shown in the SAPI interface and listed in the safety summary.

The following table shows the parameters of the SAPI of a SafetyProvider.

SAPI parameters	Data type of the mapped local data element	Mapping to a local data element of the Safety program	Mapping to a local data element of the standard user program	Description
SafetyData	UDT	x	-	Data that is made available to a SafetyConsumer in a safety-related manner
NonSafetyData	UDT	-	x	Data that is made available to a SafetyConsumer in a non-safety-related manner
InputSignals	UDT			Summary of the following SAPI parameters
OperatorAck-Provider	BOOL	x	-	1=User acknowledgment at SafetyProvider end
ActivateFSV	BOOL	x	-	1=SafetyConsumer to output substitute values
EnableTest-Mode	BOOL	x	-	1=Current values at the SafetyData SAPI parameter are test data
OutputSignals	UDT			Summary of the following SAPI parameters
OperatorAck-Requested	BOOL	-	x	1=Acknowledgment request at the SafetyConsumer side is pending
Monitoring-Number	UDINT	-	x	Current MonitoringNumber
SafetyCon-sumerID	UDINT	-	x	ID for authentication of the connected SafetyConsumer
SafetyProviderID	UDINT	x	-	ID for authentication of a SafetyProvider
SafetyBaseID	GUID	-	x	ID for authentication of a SafetyProvider

To establish a connection of your Safety program (or standard user program) to a SafetyProvider, you must map all required SAPI parameters to a local data element of your Safety program (or standard user program) in SiOME (Page 308) ("Local data mapping").

SafetyData

The SAPI parameter SafetyData is used to transfer the data of your Safety program that you want to make available to a connected SafetyConsumer in a safety-related manner. The data structure of this data must be described in an F-compliant PLC data type (UDT) which agrees with the data structure of the OPC UA data type of SafetyData. This F-compliant PLC data type (UDT) may only contain the BOOL, INT and DINT data types.

⚠ WARNING

During communication with OPC UA Safety, it is only ensured that a signal state to be transferred is acquired and transferred to the receiver if it is pending for at least twice as long as the assigned monitoring time (SafetyConsumerTimeout). *(S109)*

⚠ WARNING

For each communication connection with OPC UA Safety, you must ensure when accepting the Safety program that the data structure of the OPC UA data types used by SafetyData matches on the SafetyProvider and SafetyConsumer side. In particular, the correct order of elements of the same data type is relevant. You can find the structures of the OPC UA data types used in the respective safety summary.

To identify the communication connection, note the following points:

- For static connections:

  To do this identify the communication connection by means of the combination of SafetyProviderIDConfigured and SafeBaseIDConfigured.
- For dynamic connections:

  To do this, identify the communication connection by means of the algorithm for the allocation of the values for SafetyProviderID and SafetyConsumerID.

(S111)

NonSafetyData

The SAPI parameter NonSafetyData is used to transfer the data of your standard user program that you want to make available to a connected SafetyConsumer in a non-safety-related manner synchronously with the SafetyData. The data structure of this data must be written in a PLC data type (UDT) which agrees with the data structure of the OPC UA data type of NonSafetyData.

This PLC data type (UDT) can only contain the data types BOOL, SINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL, WSTING, LDT and nested PLC data types (UDTs).

InputSignals

The SAPI parameter InputSignals is used to summarize the SAPI parameters OperatorAckProvider, ActivateFSV and EnableTestMode. These SAPI parameters enable you to trigger certain actions on the connected SafetyConsumer or communicate certain status information about the SafetyData to the SafetyConsumer.

An F-compliant PLC data type (UDT) that contains all the InputSignals listed in the above table is needed to use these SAPI parameters:

- **OperatorAckProvider**

  With the OperatorAckProvider SAPI parameter, you also have the option of performing user acknowledgment on the SafetyProvider side and informing the Safety program on the SafetyConsumer side (see also SAPI parameters OperatorAckConsumer and OperatorAckProvider (Page 299) of the SafetyConsumer).

  ⚠ WARNING

  A user acknowledgment (Page 190) must always take place with a signal that was generated by manual operation. Implementation with an automatically generated signal is not permitted. *(S011)*

- **ActivateFSV**

  With the ActivateFSV SAPI parameter, you can instruct a connected SafetyConsumer to output the designated substitute values from the F-OPC-UA-Consumer-DB (Page 304) instead of the current values at the SafetyData SAPI parameter. In order for the current values to be output at the SAPI parameter SafetyData again after ActivateFSV is reset, a user acknowledgment may be required.

- **EnableTestMode**

  With the SAPI parameter EnableTestMode, you can inform a connected SafetyConsumer that the current values at the SAPI parameter SafetyData are test data and, thus, were not formed in a safety-related manner.

OutputSignals

The SAPI parameter OutputSignals is used to summarize the SAPI parameters OperatorAckRequested, MonitoringNumber and SafetyConsumerID. These SAPI parameters allow you to evaluate certain information of the connected SafetyConsumer.

A PLC data type (UDT) that contains all the OutputSignals listed in the above table is needed to use these SAPI parameters:

- **OperatorAckRequested**

  You can use the OperatorAckRequested SAPI parameter to evaluate whether there is an acknowledgment request on the SafetyConsumer.

- **MonitoringNumber**

  With the SAPI parameter MonitoringNumber, you can evaluate the current MonitoringNumber for diagnostic purposes. More information can be found in the OPC UA Specification OPC 10000-15.

- **SafetyConsumerID**

  With the SAPI parameter SafetyConsumerID, you can evaluate the SafetyConsumerID of the SafetyConsumer for diagnostic purposes. More information can be found in the OPC UA Specification OPC 10000-15.

SafetyProviderID/SafetyBaseID

With the SAPI parameters SafetyProviderID and SafetyBaseID, you can uniquely and reliably authenticate a SafetyProvider at dynamic connections. For more information, refer to the sections "Series machines (Page 331)" and "Changing communication partners (Page 332)".

SafetyProviderID and/or SafetyBaseID are initialized with values matched to the specific situation in the system during runtime of the Safety program (or standard user program). As long as the relevant value <> 0, the value assigned to the SPI parameters SafetyProviderIDConfigured and SafetyBaseIDConfigured is overwritten.

If the active SafetyProviderID or SafetyBaseID is "0", no communication with a SafetyConsumer can take place.

NOTE

The instructions "Random", "Get_IM_Data", "RD_SYS_T" and "SHA2" are available for generating a SafetyBaseID. These instructions enable the safety requirements of OPC UA Specification OPC 10000-15 regarding randomly generated values to be fulfilled.

You can find application examples for concrete implementation on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109977450).

⚠ WARNING

For the unique and secure authentication of a SafetyProvider, the combination of active SafetyProviderID and active SafetyBaseID at any point in time must be unique for all OPC UA Safety communication connections network-wide. **

If your network is not safely separated from the outside world through physical separation, the active SafetyBaseID at any point in time must be a value that was generated randomly in accordance with the OPC UA Specification OPC 10000-15. ***

You must ensure the uniqueness in the safety summary during acceptance of the safety program. If the active SafetyProviderID or active SafetyBaseID is not formed until runtime of your Safety program, you must ensure the correct implementation of the algorithm for the generation of the values. *(S102)*

* If you cannot ensure a unique combination during startup of the Safety program because the active SafetyProviderID or active SafetyBaseID is not formed until runtime of your Safety program, you must ensure that the active SafetyProviderID or active SafetyBaseID is "0" during this phase.

** SafetyProviders to which the same local data element of your safety program is mapped (Safety Multicast (Page 330)) at the SafetyData SAPI parameter are permitted to use the same combination.

*** If you have entered a value for the SPI parameter SafetyBaseIDConfigured of a SafetyProvider (either in the NodeSet or in the properties of a SafetyProvider), you yourself must ensure that the value meets the safety requirements of the OPC UA Specification OPC 10000-15. Alternatively, generate the value of SafetyBaseIDConfigured again by opening the properties tab of the affected SafetyProvider and clicking on the "Generate" button.

⚠ WARNING

Due to system constraints, formation of a SafetyBaseID at runtime of your Safety program is only possible in your standard user program.

Note that these program sections are therefore not captured by the Collective F-Signature and are not protected by "Access protection for safety-related project data (Page 100)" or "F-CPU access protection (Page 103)".

For this reason, you must take organizational measures (Page 651) to ensure that only authorized persons can change these program sections.

Print the relevant program sections and supplement your safety summary with this printout. *(S103)*

9.5.1.3 Safety Parameter Interface (SPI) of the SafetyConsumer

The SPI of a SafetyConsumer is displayed to you after adding a SafetyProvider in the SafetyProvider list of the client interface (Page 320) in the properties of the SafetyConsumer in the editor for client interfaces (Page 324).

The following table shows the parameters of the SPI of a SafetyConsumer.

Parameter	Data type	Changeable in TIA Portal	Description
SafetyProviderIDConfigured	UDINT	x	ID for authentication of the intended SafetyProvider
SafetyBaseIDConfigured	GUID	x	ID for authentication of the intended SafetyProvider
SafetyConsumerIDConfigured	UDINT	x	ID for authentication of a SafetyConsumer
SafetyProviderLevel	BYTE	x	Safety Integrity Level (SIL) expected from the specific implementation of the connected SafetyProvider
SafetyStructureSignature	UDINT	-	Signature via the data structure of the OPC UA data type of the fail-safe data to be transferred
SafetyStructureSignatureVersion	UINT	-	Version of the algorithm for forming the SafetyStructureSignature
SafetyStructureIdentifier	STRING	-	Identifier of the OPC UA data types of the fail-safe data to be transferred (max. length 128 characters)
SafetyConsumerTimeout	UDINT	x	Monitoring time in µs: Maximum permissible time between the request from a SafetyConsumer and the receipt of the response from a SafetyProvider in the SafetyConsumer
SafetyOperatorAckNecessary	BOOL	x	1=Acknowledgment required
SafetyErrorIntervalLimit	INT	x	Minimum time in min. between two communication errors
SafetyClientImplemented	BOOL	-	1=SafetyConsumer supports OPC UA client/server communication mechanism
SafetyPubSubImplemented	BOOL	-	1=SafetyConsumer supports OPC UA PubSub communication mechanism

The values of individual SPI parameters of a SafetyConsumer are automatically assigned when the intended SafetyProvider is added to the SafetyProvider list of the client interface (Page 320) if no value has yet been entered for these parameters in the OPC UA SafetyProvider information model or if the parameters do not exist at the SafetyProvider. Previously entered values are applied if permissible. If necessary, you have to adjust the values in the properties

of the SafetyConsumer in the editor for client interfaces (Page 320). An error message is displayed for invalid values.

SafetyProviderIDConfigured/SafetyBaseIDConfigured

The SafetyProviderIDConfigured and SafetyBaseIDConfigured SPI parameters are used to uniquely and safely authenticate a SafetyProvider. For applications with changing communication partners (Page 332) or series machines (Page 331), you can overwrite the values of these SPI parameters in the F-OPC-UA-Consumer-DB (Page 304) using the SAPI parameters SafetyProviderID and SafetyBaseID. The value that is actually in effect during runtime of the Safety program is referred to hereinafter as the "active SafetyBaseID" or "active SafetyProviderID".

If the active SafetyProviderID or SafetyBaseID is "0", the SafetyConsumer outputs the designated substitute values from the F-OPC-UA-Consumer-DB (Page 304) to the Safety program instead of the current values at the SafetyData SAPI parameter.

If the correct values are not yet present in the SafetyProvider list of the client interface (Page 320) when adding the intended SafetyProvider, you must enter these in the properties of the SafetyConsumer in the editor for client interfaces (Page 320).

⚠ WARNING

For secure authentication of the SafetyProvider whose SafetyData is to be evaluated, the values of the active SafetyProviderID and active SafetyBaseID of the SafetyConsumer must correspond to the values of the active SafetyProviderID and active SafetyBaseID of this SafetyProvider at all times*.

When accepting the Safety programs, you must ensure compliance in the relevant safety summary. If the active SafetyProviderID or active SafetyBaseID is not generated until runtime of your Safety program, you must ensure the correct implementation of the algorithm for the generation of the values. *(S104)*

* If you cannot yet provide the values when starting the Safety program because they are only created during the runtime of your Safety program, you must ensure that the active SafetyProviderID or active SafetyBaseID or the Enable "0" SAPI parameter in this phase.

SafetyConsumerIDConfigured

The SafetyConsumerIDConfigured SPI parameter can be used to uniquely authenticate a SafetyConsumer, particularly for diagnostic purposes in applications with Safety Multicast (Page 330). For more details, refer to OPC UA Specification OPC 10000-15.

For applications with changing communication partners (Page 332) or series machines (Page 331), you can overwrite the value of this SPI parameter via the SafetyConsumerID (Page 299) SAPI parameter. The value that is actually in effect at runtime of the Safety program is referred to below as the "active SafetyConsumerID".

If the active SafetyConsumerID is "0", the SafetyConsumer outputs the designated substitute values from the F-OPC-UA-Consumer-DB (Page 304) to the Safety program instead of the current values at the SafetyData SAPI parameter.

When the foreseen SafetyProvider is added to the SafetyProvider list of the client interface (Page 320), a randomly generated value is assigned.

SafetyProviderLevel

The SafetyProviderLevel SPI parameter sets an expected value of the safety integrity level (SIL) achievable from the specific implementation of the connected SafetyProvider.

With automatic assignment, the value "3" is entered. This corresponds to the maximum SIL achievable with SIMATIC Safety. You may need to adapt this value to your individual application.

If the value does not match the value of the connected SafetyProvider, a communication error is detected (error code "0x14", see DIAG variable in the F-OPC-UA-Consumer-DB (Page 304)).

SafetyStructureSignature

The SafetyStructureSignature SPI parameter is used to protect the data structure of the OPC UA data type at the SafetyData SAPI parameter via a signature.

With automatic assignment, the value is calculated according to the algorithm prescribed in OPC UA Specification OPC 10000-15.

SafetyStructureSignatureVersion

The SafetyStructureSignatureVersion SPI parameter is used to specify the version of the algorithm used to calculate the SafetyStructureSignature.

With automatic assignment, the value "1" is entered. This corresponds to the only permissible value at the present time.

SafetyStructureIdentifier

With the SPI parameter SafetyStructureIdentifier the identifier of the OPC UA data types of the SAPI parameter SafetyData is publicized. The SafetyStructureIdentifier is included in the calculation of the SafetyStructureSignature.

With automatic assignment, the name of the OPC UA data type that describes the data structure at the SafetyData SAPI parameter is entered.

SafetyConsumerTimeout

The SafetyConsumerTimeout SPI parameter specifies the maximum time (in µs) that may elapse between the request from a SafetyConsumer and the response from the connected SafetyProvider.

If this monitoring time expires before a response from the connected SafetyProvider arrives, a communication error is detected (see DIAG variable in the F-OPC-UA-Consumer-DB (Page 304)) and the SafetyConsumer outputs the designated substitute values from the F-OPC-UA-Consumer-DB (Page 304) at the SafetyData SAPI parameter instead of the current values.

With automatic assignment, the value "2500000" (2.5 s) is currently entered as the default value. You must adapt this value to your individual application. More information can be found in the section "Response Times of Safety Functions (Page 635)".

⚠ WARNING

During communication with OPC UA Safety, it is only ensured that a signal state to be transferred is acquired and transferred to the receiver if it is pending for at least twice as long as the assigned monitoring time (SafetyConsumerTimeout). *(S109)*

SafetyOperatorAckNecessary

The SafetyOperatorAckNecessary SPI parameter is used to specify whether a manual user acknowledgment is required before the current values are output again at the SafetyData SAPI parameter instead of the substitute values from the F-OPC-UA-Consumer-DB (Page 304) after the following states have passed:

- Communication error "Timeout" (error code "0x08", see DIAG variable in the F-OPC-UA-Consumer-DB (Page 304)).
- ActivateFSV SPI parameter of the connected SafetyProvider is "1" (error code "0x20", see DIAG variable in the F-OPC-UA-Consumer-DB (Page 304)).

In the case of automatic assignment, the value "TRUE" is entered. You can adapt this value to your individual application if necessary.

SafetyErrorIntervalLimit

The SafetyErrorIntervalLimit SPI parameter specifies the minimum time (in minutes) that must elapse between two consecutive communication errors.

If this time is not reached, the SafetyConsumer outputs the substitute values from the F-OPC-UA-Consumer-DB (Page 304) at the SafetyData SAPI parameter instead of the current values. If this time is maintained, the SafetyConsumer tolerates a communication error by discarding the bad response from the SafetyProvider and waiting for the next error-free response. However, this can lead to a "Timeout" communication error if the monitoring time on the SafetyConsumerTimeout SPI parameter does not allow any reserves for repeats.

The value influences the probability of failure values (Page 446) (PFD_{avg} and PFH values) that you must take into account for communication with OPC UA Safety.

With automatic assignment, the value "600" is entered. This value achieves the lowest possible probability of failure values.

SafetyClientImplemented

The SafetyClientImplemented SPI parameter is used to indicate whether the SafetyConsumer supports OPC UA client/server communication mechanisms.

In the case of automatic assignment, the value "TRUE" is entered. This corresponds to the only permissible value since SIMATIC Safety supports only OPC UA client/server communication.

SafetyPubSubImplemented

The SafetyPubSubImplemented SPI parameter is used to indicate whether the SafetyConsumer supports OPC UA PubSub communication mechanisms.

With automatic assignment, the value "FALSE" is entered. This corresponds to the only permissible value since SIMATIC Safety does not support OPC UA PubSub communication.

9.5.1.4 Safety Application Program Interface (SAPI) of the SafetyConsumer in the F-OPC UA Consumer DB

The SAPI of a SafetyConsumer is available in the F-OPC-UA-Consumer-DB. The F-OPC-UA-Consumer-DB is automatically generated when the intended SafetyProvider is added to the SafetyProvider list of the client interface (Page 320). You can read or write directly to the corresponding variables in the F-OPC-UA-Consumer-DB.

You can find the name automatically assigned to the F-OPC-UA-Consumer-DB in the properties of the SafetyConsumer in the editor for client interfaces. If necessary, you can also adapt the name there. You can find the F-OPC-UA-Consumer-DB in the folder "Program blocks/System blocks/STEP 7 Safety/OPC UA Safety" folder in the project tree.

The following table shows the parameters of the SAPI of a SafetyConsumer.

SAPI parameter	Tag in the F-OPC-UA-Consumer-DB	Data type	Access to in the Safety program	Access in the standard user program	Description
Tags that you can or must write					
Enable	ENABLE	BOOL	x	-	1=Activation of communication to the connected SafetyProvider
OperatorAckConsumer	OPERATOR_ACK_CONSUMER	BOOL	x	-	1=User acknowledgment at SafetyConsumer end
SafetyProviderID	SAFETY_PROVIDER_ID	DINT	x	-	ID for authentication of the intended SafetyProvider
SafetyBaseID	SAFETY_BASE_ID	GUID	-	x	ID for authentication of the intended SafetyProvider
SafetyConsumerID	SAFETY_CONSUMER_ID	DINT	x	-	ID for authentication of a SafetyConsumer
Tags that you can/must evaluate					
SafetyData	SAFETY_DATA	UDT	x	(x)	Data provided by the connected SafetyProvider in a safety-related manner
NonSafetyData	NON_SAFETY_DATA	UDT	-	x	Data that is not made available by the connected SafetyProvider in a safety-related manner
FSV_Activated	FSV_ACTIVATED	BOOL	x	(x)	1=SafetyConsumer outputs substitute values

(x) can also be evaluated non-safety-related in the standard user program if required

SAPI parameter	Tag in the F-OPC-UA-Consumer-DB	Data type	Access to in the Safety program	Access in the standard user program	Description
OperatorAckRequested	OPERATOR_ACK_REQUESTED	BOOL	x	(x)	1=Acknowledgment request is pending
OperatorAckProvider	OPERATOR_ACK_PROVIDER	BOOL	x	(x)	1=User acknowledgment at SafetyProvider end
TestModeActivated	TEST_MODE_ACTIVATED	BOOL	x	(x)	1=Current values at the SafetyData SAPI parameter are test data

(x) can also be evaluated non-safety-related in the standard user program if required

Enable

The Enable SAPI parameter is used to activate communication with OPC UA Safety to the connected SafetyProvider. If Enable = 0, no SafetyData/NonSafetyData is requested from the connected SafetyProvider. Instead of the current values, the SafetyConsumer outputs the designated substitute values from the F-OPC-UA-Consumer-DB (Page 304) at the SafetyData SAPI parameter.

With Enable = 0, for example, you can delay the start of communication with OPC UA Safety until the underlying OPC UA communication is ready (e.g. after switching the power supply on). This allows you to avoid communication errors such as "timeout" in this phase (see SafetyConsumerTimeout (Page 295) SPI parameter).

NOTE

When changing from Enable = 1 to Enable = 0, the output of substitute values at the SafetyData SAPI parameter only take effect after the next data package has been received or after the monitoring time parameterized via SafetyConsumerTimeout.

OperatorAckConsumer

The OperatorAckConsumer SAPI parameter is used to perform user acknowledgment. This requires a positive edge after an acknowledgment request has been made at the OperatorAckRequested SAPI parameter.

Once the user acknowledgment has been carried out, the SafetyConsumer outputs the current values at the SafetyData SAPI parameter again instead of the substitute values from the F-OPC-UA-Consumer-DB (Page 304).

⚠ WARNING

A user acknowledgment (Page 190) must always take place with a signal that was generated by manual operation. Implementation with an automatically generated signal is not permitted. *(S011)*

SafetyProviderID/SafetyBaseID

With the SafetyProviderID and SafetyBaseID SAPI parameters, you can uniquely and reliably authenticate the foreseen SafetyProvider for dynamic connections. For more information, refer to the sections "Series machines (Page 331)" and "Changing communication partners (Page 332)".

SafetyProviderID and/or SafetyBaseID are initialized with values matched to the specific situation in the system during runtime of the Safety program (or standard user program). As long as both values <> 0, the values assigned to the SafetyProviderIDConfigured and SafetyBaseIDConfigured SPI parameters are overwritten.

If the active SafetyProviderID or SafetyBaseID is "0", the SafetyConsumer outputs the designated substitute values from the F-OPC-UA-Consumer-DB (Page 304) to the Safety program instead of the current values at the SafetyData SAPI parameter.

NOTE

A changed value at the SAPI parameters SafetyProviderID and SafetyBaseID only becomes effective after a positive edge at the Enable SAPI parameter.

⚠ WARNING

Due to system constraints, formation of a SafetyBaseID at runtime of your Safety program is only possible in your standard user program.

Note that these program sections are therefore not captured by the collective F-signature and are not protected by "Access protection for safety-related project data (Page 100)" or "F-CPU access protection (Page 103)".

For this reason, you must take organizational measures (Page 651) to ensure that only authorized persons can change these program sections.

Print the relevant program sections and supplement your safety summary with this printout. *(S103)*

⚠ WARNING

For secure authentication of the SafetyProvider whose SafetyData is to be evaluated, the values of the active SafetyProviderID and active SafetyBaseID of the SafetyConsumer must correspond to the values of the active SafetyProviderID and active SafetyBaseID of this SafetyProvider at all times*.

When accepting the Safety programs, you must ensure compliance in the relevant safety summary. If the active SafetyProviderID or active SafetyBaseID is not generated until runtime of your Safety program, you must ensure the correct implementation of the algorithm for the generation of the values. *(S104)*

* If you cannot yet provide the values when starting the Safety program because they are only created during the runtime of your Safety program, you must ensure that the active SafetyProviderID or active SafetyBaseID or the Enable "0" SAPI parameter in this phase.

SafetyConsumerID

You can also use the SafetyConsumerID SAPI parameter to uniquely authenticate a SafetyConsumer in use cases with changing communication partners (Page 332) or series machines (Page 331).

The SafetyConsumerID is then only supplied with a value adapted to the specific system situation during the runtime of the Safety program. As long as the relevant value <> 0, the value assigned to the SafetyProviderIDConfigured PI parameter is overwritten.

If the active SafetyConsumerID is "0", the SafetyConsumer outputs the designated substitute values from the F-OPC-UA-Consumer-DB (Page 304) to the Safety program instead of the current values at the SafetyData SAPI parameter.

NOTE

A changed value at the SAPI parameter SafetyConsumerID only becomes effective after a positive edge at the Enable SAPI parameter.

SafetyData

The SafetyData SAPI parameter provides the data that is made available by the connected SafetyProvider in a safety-related manner for evaluation in your Safety program.

⚠ WARNING

For each communication connection with OPC UA Safety, you must ensure when accepting the Safety program that the data structure of the OPC UA data types used by SafetyData matches on the SafetyProvider and SafetyConsumer side. In particular, the correct order of elements of the same data type is relevant. You can find the structures of the OPC UA data types used in the respective safety summary.

To identify the communication connection, note the following points:

- For static connections:

  To do this identify the communication connection by means of the combination of SafetyProviderIDConfigured and SafeBaseIDConfigured.
- For dynamic connections:

  To do this, identify the communication connection by means of the algorithm for the allocation of the values for SafetyProviderID and SafetyConsumerID.

(S111)

NonSafetyData

The NonSafetyData SAPI parameter provides the data that is not made available by the connected SafetyProvider in a safety-related manner for evaluation in your standard user program.

FSV_Activated

You can use the FSV_Activated SAPI parameter to evaluate whether the SafetyConsumer outputs the designated substitute values from the F-OPC-UA-Consumer-DB (Page 304) at the SafetyData SAPI parameter instead of the current values.

OperatorAckRequested

You can use the OperatorAckRequested SAPI parameter to evaluate whether there is an acknowledgment request on the SafetyConsumer.

OperatorAckProvider

You can use the OperatorAckProvider SAPI parameter to evaluate whether there is a user acknowledgment on the side of the connected SafetyProvider.

To use this as an alternative (or in addition) to your acknowledgment signal on the SafetyConsumer side, you must link the OperatorAckProvider SAPI parameter with the OperatorAckConsumer SAPI parameter (possibly OR-linked with the acknowledgment signal on the SafetyConsumer side).

TestModeActivated

You can use the TestModeActivated SAPI parameter to evaluate whether the connected SafetyProvider "only" provides test data, i.e. non-safety-related data, at its SafetyData SAPI parameter.

This is the case if the EnableTestMode SAPI parameter has been set to "1" on the connected SafetyProvider.

If the connected SafetyProvider is located on an F-CPU S7-1500, TestModeActivated is also set to "1" if the Safety program belonging to the connected SafetyProvider is running in disabled safety mode or on an *S7-PLCSIM*. For information on the behavior of devices of a third-party manufacturer, refer to the documentation of the device manufacturer.

⚠ WARNING

If the TEST_MODE_ACTIVATED variable in the F-OPC-UA-Consumer-DB is set to "1" during communication with OPC UA Safety, you can no longer assume that the data at the SAFETY_DATA variable is formed in a safety-related manner. This is the case when the connected SafetyProvider is in deactivated safety mode on a PLCSIM instance or the EnableTestMode SAPI parameter of the connected SafetyProvider has been set in the associated user program. During this time, you must take organizational measures (Page 651) to ensure the safety of the system components that are affected by this data. Alternatively, you can also use safe substitute values in your Safety program with TEST_MODE_ACTIVATED = 1 instead of the data in the SAFETY_DATA tag. *(S108)*

9.5.1.5 Other variables in the F-OPC UA Consumer DB

In addition to the SAPI of a SafetyConsumer, the F-OPC-UA-Consumer-DB also provides you with other variables that you need for communication with OPC UA Safety in the *SIMATIC Safety* F-system and the underlying OPC UA communication ("client instructions"):

Tag in the F-OPC-UA-Consumer-DB	Data type	Access in the Safety program:	Access in the standard user program	Description
Tags that you can or must write				
RESPONSE_SPDU	STRUCT	-	x	Tag for the in-out parameter (InOut) OutputArguments of the client instruction OPC_UA_MethodCall
METHOD_CALL_DONE	BOOL	-	x	Variable for the Done output of the client instruction OPC_UA_MethodCall
METHOD_CALL_BUSY	BOOL	-	x	Variable for the Busy output of the client instruction OPC_UA_MethodCall
METHOD_CALL_ERROR	BOOL	-	x	Variable for the Error output of the client instruction OPC_UA_MethodCall
METHOD_RESULT	DWORD	-	x	Tag for the in-out parameter (InOut) MethodResult of the client instruction OPC_UA_MethodCall
Tags that you can/must evaluate				
REQUEST_SPDU	STRUCT	-	x	Tag for the in-out parameter (InOut) InputArguments of the client instruction OPC_UA_MethodCall
METHOD_CALL_REQUEST	BOOL	-	x	Tag for the REQ input of the client instruction OPC_UA_MethodCall
DIAG	BYTE	-	x	Non-fail safe service information
SAFETY_PROVIDER_ID_ACTIVE	UDINT	-	x	Active SafetyProviderID
SAFETY_BASE_ID_ACTIVE	GUID	-	x	Active SafetyBaseID
SAFETY_CONSUMER_ID_ACTIVE	UDINT	-	x	Active SafetyConsumerID
Variables for which you can/must assign start values				
PASS_VALUES	UDT (analog SafetyData)	-	-	Substitute values for SafetyData

RESPONSE_SPDU

You need this variable to call up the client instruction OPC_UA_MethodCall, which is part of the underlying OPC UA communication.

When using the F-OPC-UA-Consumer-Com-FB (Page 328), this variable is automatically interconnected with the OPC_UA_MethodCall instruction.

METHOD_CALL_DONE, METHOD_CALL_BUSY, METHOD_CALL_ERROR, METHOD_RESULT

You need these variables to call up the client instruction OPC_UA_MethodCall, which is part of the underlying OPC UA communication.

When using the F-OPC-UA-Consumer-Com-FB (Page 328), these variables are automatically interconnected with the OPC_UA_MethodCall instruction.

REQUEST_SPDU

You need this variable to call up the client instruction OPC_UA_MethodCall, which is part of the underlying OPC UA communication.

When using the F-OPC-UA-Consumer-Com-FB (Page 328), this variable is automatically interconnected with the OPC_UA_MethodCall instruction.

METHOD_CALL_REQUEST

You need this variable to call up the client instruction OPC_UA_MethodCall, which is part of the underlying OPC UA communication.

When using the F-OPC-UA-Consumer-Com-FB (Page 328), this variable is automatically interconnected with the OPC_UA_MethodCall instruction.

DIAG

The DIAG variable is used to provide a non-fail-safe code for communication errors/states that have occurred for service purposes up to the user acknowledgement. You can read out this information by means of operator control and monitoring systems or, if applicable, you can evaluate it in your standard user program.

Error code	Cause	Bad response from the SafetyProvider discarded *	Output of substitute values FSV_Activated = 1	User acknowledgment required
0x01	Discrepancy detected with active SafetyProviderID or SafetyBaseID	x	-	-
0x11	Discrepancy with active SafetyBaseID detected	-	x	x
0x12	Discrepancy with active SafetyProviderID detected	-	x	x

(-): A user acknowledgment is required for the SafetyOperatorAckNecessary SPI parameter of the SafetyConsumer = 1.

0x13	Discrepancy with SafetyStructureSignature or SafetyStructureSignatureVersion or SafetyStructureIdentifier detected	-	x	x
0x14	Discrepancy detected with SafetyProviderLevel	-	x	x
0x05	CRC error detected	x	-	-
0x15	CRC error detected	-	x	x
0x06	Discrepancy with active SafetyConsumerID detected	x	-	-
0x16	Discrepancy with active SafetyConsumerID detected	-	x	x
0x07	MonitoringNumber error detected	x	-	-
0x17	MonitoringNumber error detected	-	x	x
0x08	Timeout error detected	-	x	(-)
0x09	Enable SAPI parameter of the SafetyConsumer = 0	-	x	-
0x0A	SafetyConsumer SPI parameters are invalid.	-	x	-
0x20	SAPI parameter ActivateFSV of the SafetyProvider = 1	-	x	(-)

(-): A user acknowledgment is required for the SafetyOperatorAckNecessary SPI parameter of the SafetyConsumer = 1.

* More information can be found in the section "Safety Parameter Interface (SPI) of the SafetyConsumer (Page 295)" under "SafetyErrorIntervalLimit".

SAFETY_PROVIDER_ID_ACTIVE/SAFETY_BASE_ID_ACTIVE/SAFETY_CONSUMER_ID_ACTIVE

You can use these variables to evaluate the active SafetyProviderID, SafetyBaseID or SafetyConsumerID for diagnostic purposes.

These variables are also required for the correct display of all data of the SafetyConsumerTypes (Page 335).

PASS_VALUES

You use this variable to define the substitute values as start values in the F-OPC-UA-Consumer-DB, which the SafetyConsumer outputs instead of the current values at the SafetyData SAPI parameter in the following cases (for details see DIAG variable):

- Communication error detected.
- Enable SAPI parameter of the SafetyConsumer is "0"
- ActivateFSV SAPI parameter of the connected SafetyProvider is "1"

NOTE

If you do not explicitly define any start values in the PASS_VALUES variable in the F-OPC-UA-Consumer-DB, the default values of the PLC data type (UDT) automatically created by *STEP 7 Safety* for this data structure are used as start values. If necessary, you can adapt these default values. For further details see "Configuring properties of a SafetyConsumer (Page 324)".

9.5.2 Configuring and programming a SafetyProvider

Requirements

- An F-CPU S7-1500 as of FW 4.0 (except HF-CPUs) or an S7-1500 F Software Controller as of firmware V40.0.
- The following function right is available for the "SafetyAdmin" role: OPC UA server access. More information can be found in the help for *STEP 7* under "Users and roles with OPC UA function rights (https://docs.tia.siemens.cloud/r/en-us/v20/configuring-automation-systems/using-opc-ua-communication-s7-1200-s7-1500-s7-1500t/using-the-s7-1500-as-an-opc-ua-server-s7-1500-s7-1500t/configuring-the-opc-ua-server-s7-1500-s7-1500t/users-and-roles-with-opc-ua-function-rights-s7-1500-s7-1500t)".
- The required Runtime license for OPC UA is available and is configured in *STEP 7*.
- The OPC UA server is activated.

Configuring and programming a SafetyProvider

Proceed as follows to configure and program a SafetyProvider. Repeat the following steps for each data structure that you want to make available to a SafetyConsumer.

9.5.2.1 Preparing the Safety program (or standard user program) for creation of an OPC UA SafetyProvider information model

You must make the following preparations in your Safety program (or standard user program) before creating an OPC UA SafetyProvider information model.

1. Create an F-compliant PLC data type (UDT) that describes the data structure of your safety message frame that you want to make available to a SafetyConsumer in a safety-related manner.
2. Create a tag with the previously created F-compliant PLC data type (UDT) in an F-shared DB.
3. Optional:

   Create a PLC data type (UDT) that describes the data structure of the data of your standard user program that you want to make available to a SafetyConsumer in a non-safety-related manner "synchronously" with the data of your Safety program.

   Create a tag with the previously created PLC data type (UDT) in a shared DB.
4. Optional:

   If you want to trigger certain actions on a SafetyConsumer or communicate certain status information to the SafetyConsumer, create an F-compliant PLC data type (UDT) that describes the data structure of the SAPI parameter InputSignals.

   Order, names and data types are specified as follows:

   – OperatorAckProvider: BOOL
   – ActivateFSV: BOOL
   – EnableTestMode: BOOL

   Create a tag with the previously created F-compliant PLC data type (UDT) in an F-shared DB.

5. Optional:

   If you want to evaluate certain information of a SafetyConsumer, create a PLC data type (UDT) that describes the data structure of the SAPI parameter OutputSignals.

   Order, names and data types are specified as follows:

   – OperatorAckRequested: BOOL
   – MonitoringNumber: UDINT
   – SafetyConsumerID: UDINT

   Create a tag with the previously created PLC data type (UDT) in a shared DB.

6. Optional:

   If you want to change the value of the active SafetyProviderID of a SafetyProvider at runtime, create a tag with the data type DINT in an F-shared DB.

7. Optional:

   If you want to change the value of the active SafetyBaseID of a SafetyProvider at runtime, create a tag with the GUID data type in a shared DB.

8. Compile the Safety program and, if applicable, the standard user program.

9.5.2.2 Creating an OPC UA SafetyProvider information model for an individual data structure

You create an OPC UA SafetyProvider information model for your individual data structure based on a nodeset file provided by the OPC Foundation for OPC UA Safety and using SiOME. For this purpose, you must map all the SAPI parameters you require in your Safety program (or standard user program) to the designated local data (local data mapping).

You then export the OPC UA SafetyProvider information model created in this way to an XML file (nodeset file).

Requirements

The individual data structures have been created as (F-compliant) PLC data types (UDTs).

The local data needed for the local data mapping of the SAPI parameters is available and compiled in your Safety program (or standard user program).

In order to connect SiOME (https://support.industry.siemens.com/cs/ww/en/view/109755133) to the TIA project, Openness must be installed and set up.

Creating an OPC UA SafetyProvider information model

Proceed as follows to create an OPC UA SafetyProvider XML file (nodeset file).

1. First, download the nodeset file "Opc.Ua.Safety.NodeSet2.xml" of the OPC UA Safety Specification V1.05.04 from the GitHub website of the OPC Foundation. You can find the download link here (https://github.com/OPCFoundation/UA-Nodeset/tree/Safety-1.05.04-2025-02-05/Safety).

   The "Opc.Ua.Safety.NodeSet2.xml" file serves as the basis for your individual nodeset file. It contains the type and method definitions you need to create an OPC UA SafetyProvider XML file.

2. Start SiOME.
3. First, import the namespace "http://opcfoundation.org/UA/Safety".

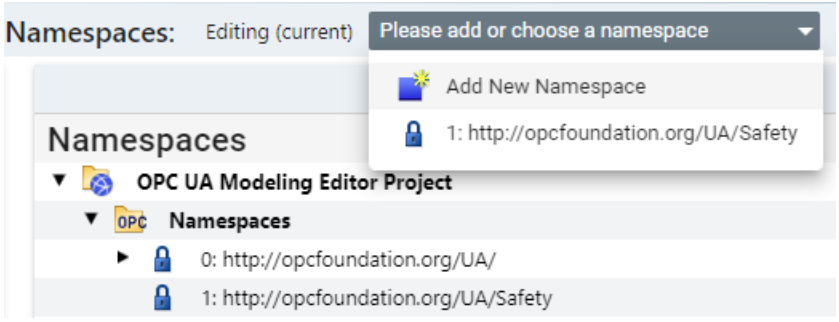
   To do so, click the "Import XML" button in the SiOME menu.

   SiOME displays the dialog for opening files.
4. Select the "Opc.Ua.Safety.NodeSet2.xml" file and click the "Open" button to import it.

   Result: SiOME imports the XML file and shows the namespace "http://opcfoundation.org/UA/Safety" in the "Namespaces" area.
5. Create your own namespace for your project.

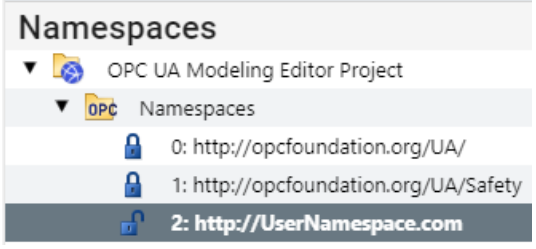
   To do so, click "Add New Namespace" in the "Namespaces: Editing (current)" area in the SiOME menu.



   SiOME opens the "Add Namespace" dialog.
6. Enter the name of the new namespace.

   The namespace "UserNamespace.com" is used in the example.

   SiOME additionally shows the new namespace:



7. Create an instance from the object type "SafetyProviderType" of the OPC UA Safety Specification.

   To do so, right-click the folder "Root/Objects/SafetyACSet" in the "Information model" area and select "Add Instance".

   NOTE

   Instances of the "SafetyProviderType" object type must always be located in the folder "SafetyACSet".

   SiOME displays the dialog "Add Instance".

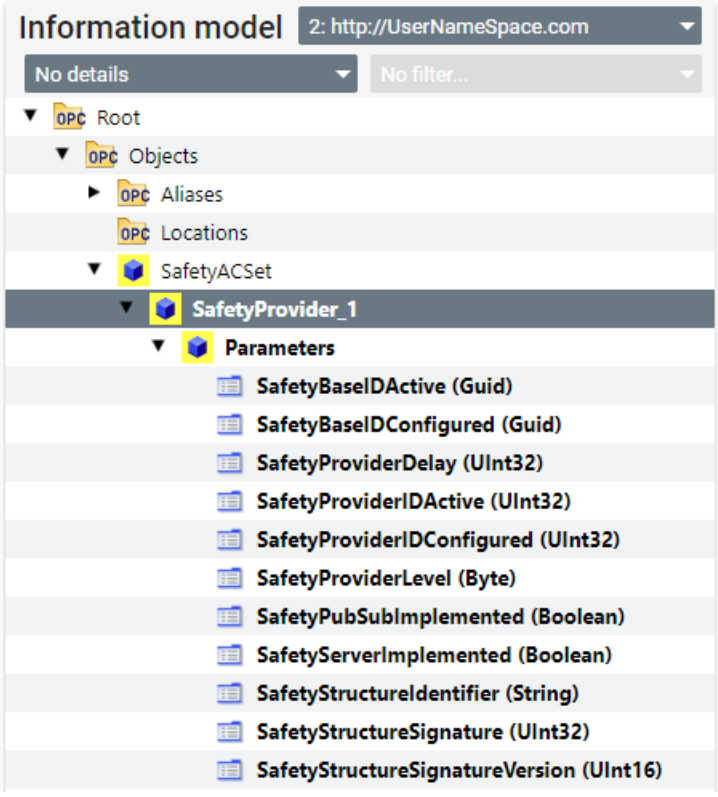
8. Enter a descriptive name for the new instance for "Name".

   The name "SafetyProvider_1" is used in the example.

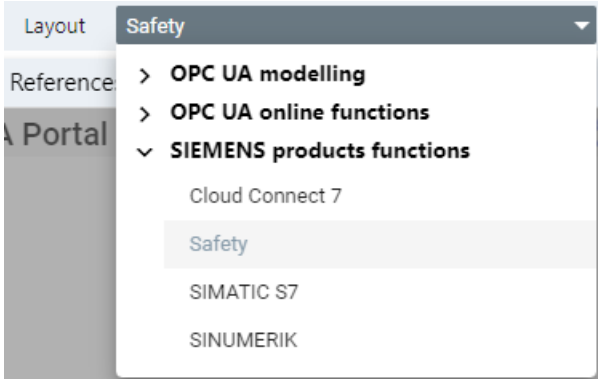
   For "TypeDefinition", select "SafetyProviderType". You can find the type under "BaseObjectTypes/SafetyObjectsType".

9. Click "OK".

   SiOME shows the new instance "SafetyProvider_1" under "SafetyACSet" in the "Information model" area. This instance already contains the SPI parameters required for communication with OPC UA Safety:



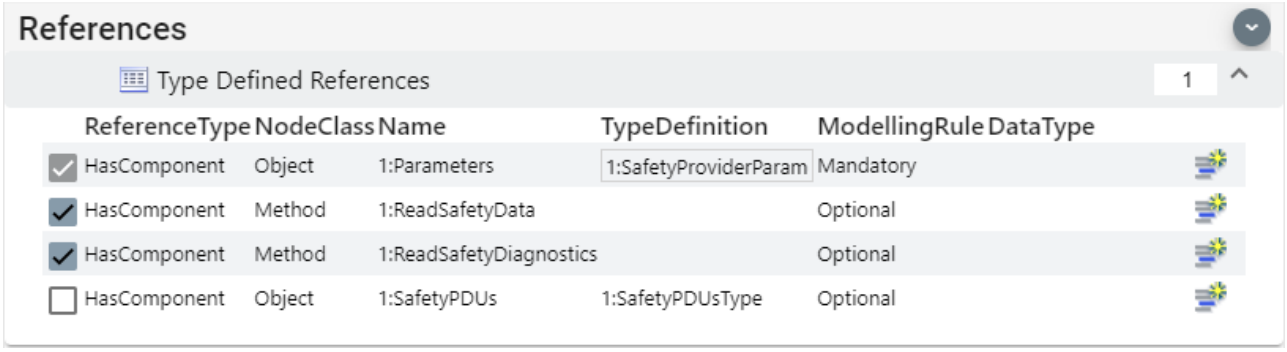
10. Select the Safety layout.
    To do so, select the "Safety" layout under "SIEMENS products functions" in the "Layout" area of the SiOME menu.



SiOME now shows the "Safety Configuration" area.

11. Add the "ReadSafetyData" and "ReadSafetyDataDiagnostics" methods. These methods are required for client access of the communication with OPC UA Safety.

    Select the corresponding check boxes for this in the "Attributes/References" area.



12. In SiOME, connect to the TIA project in which you created the data structures and local data.

    To do so, click the "Open TIA Portal project" button in the "Safety Configuration" area.

    – If a project is already opened in TIA Portal, SiOME shows the dialog "Attach to already opened TIA Portal project".
    – If no project is opened in TIA Portal, the dialog for opening of TIA Portal projects is displayed.

13. Select the project and click "Attach to selected project" or "Open". If required, enable the access in parallel in TIA Portal.

    SiOME displays the project tree of the project.

    Note: If the project contains more than one (F-)CPU, SiOME first displays the dialog "Select PLC". In this case, select the desired F-CPU.

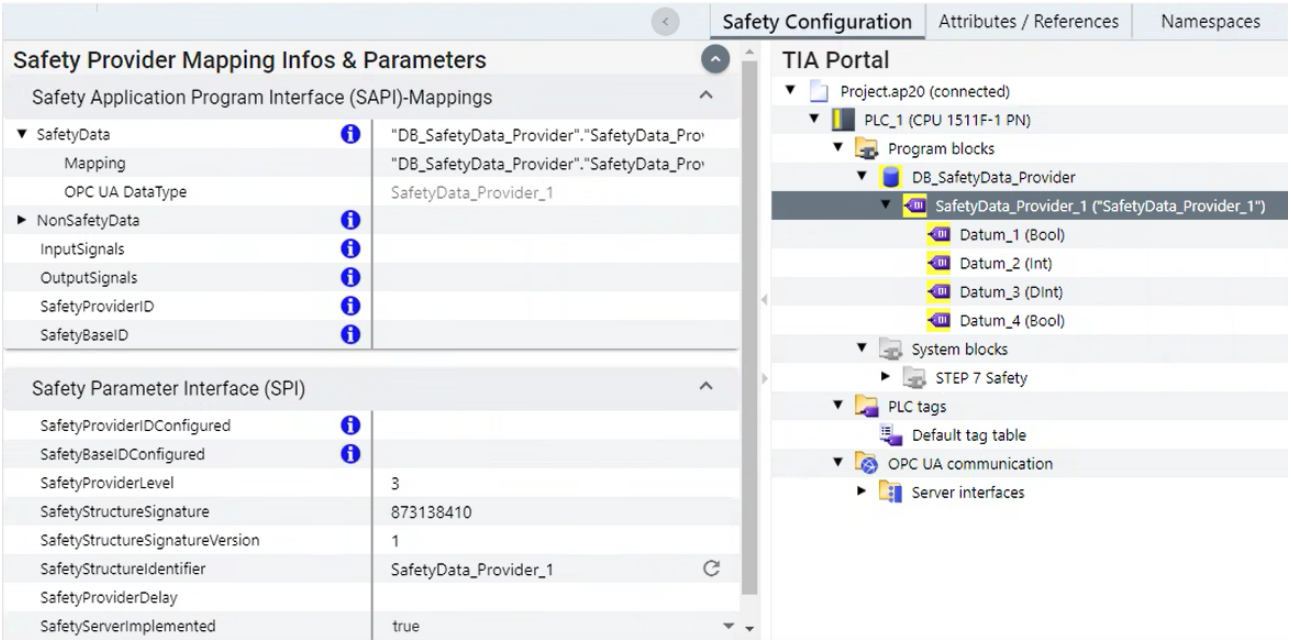
14. Perform the mapping for SafetyData in the Safety Application Program Interface (SAPI).

    To do so, use drag-and-drop to move the variable with the data structure of the data to be sent from the F-DB to the SafetyData parameter in the SAPI. Answer "Yes" to the displayed question "Do you want to generate the OPC UA DataTyp.".

    SiOME generates the associated OPC UA data type.

15. Optional:

    Perform the mapping for NonSafetyData, InputSignals, OutputSignals, SafetyProviderID and/or SafetyBaseID accordingly. The maximum length for strings is 128 characters.



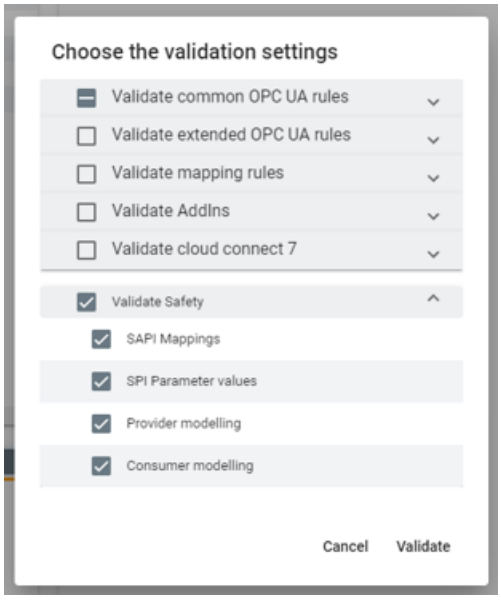
16. Optional:

    Validate the now available nodeset.
    To do so, click the "Validate information model" button in the SiOME menu.



    SiOME displays the dialog "Choose the validation settings".

17. Select the desired settings and click "Validate".



SiOME displays the result of the validation in the "Validation" area.

18. Export the XML file.

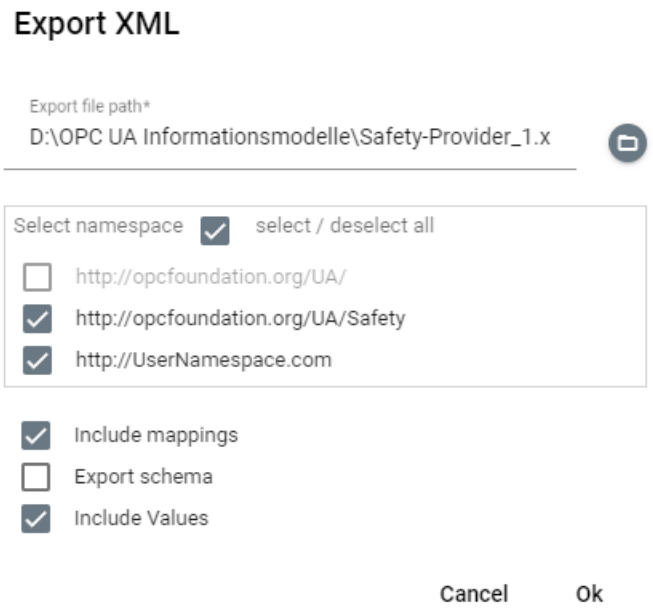
    To do so, click the "Export XML" button in the SiOME menu.



SiOME displays the dialog "Export XML".

Select the path for the export and a descriptive name for the file. The name "SafetyProvider_1" is used in the example.

Have all namespaces activated and click "OK".



9.5.2.3 Creating a server interface for an OPC UA SafetyProvider information model

You create a server interface for an OPC UA SafetyProvider information model in a similar way as a server interface for a companion specification. More information can be found in the *STEP 7* help under "Creating a server interface for a companion specification (https://docs.tia.siemens.cloud/r/en-us/v20/configuring-automation-systems/using-opc-ua-communication-s7-1200-s7-1500-s7-1500t/using-the-s7-1500-as-an-opc-ua-server-s7-1500-s7-1500t/opc-ua-server-interface-configuration-s7-1500-s7-1500t/creating-a-server-interface-for-companion-specification-s7-1500-s7-1500t)".

⚠ WARNING

Note that the data of a SafetyProvider can only be received safety-oriented if it is provided by a SafetyConsumer in the associated F-OPC UA Consumer DB. All other options for communication via OPC UA with a SafetyProvider or SafetyConsumer (OPC UA Read/Write access, OPC UA method calls) do not take place safety-oriented. *(S113)*

NOTICE

When making various changes in your user program, note that the OPC UA server of the F-CPU may need to be restarted. This can lead to an interruption of the safety-related communication with OPC UA Safety, which means that all connected SafetyConsumer output substitute values. In this case, the received process values are only output again after a user acknowledgement.

More information can be found in the *STEP 7* help under "Behavior of the OPC UA server during operation (https://docs.tia.siemens.cloud/r/en-us/v20/configuring-automation-systems/using-opc-ua-communication-s7-1200-s7-1500-s7-1500t/using-the-s7-1500-as-an-opc-ua-server-s7-1500-s7-1500t/interesting-information-about-the-opc-ua-server-of-the-s7-1500-cpus-s7-1500-s7-1500t/runtime-behavior-of-the-opc-ua-server-s7-1500-s7-1500t)".

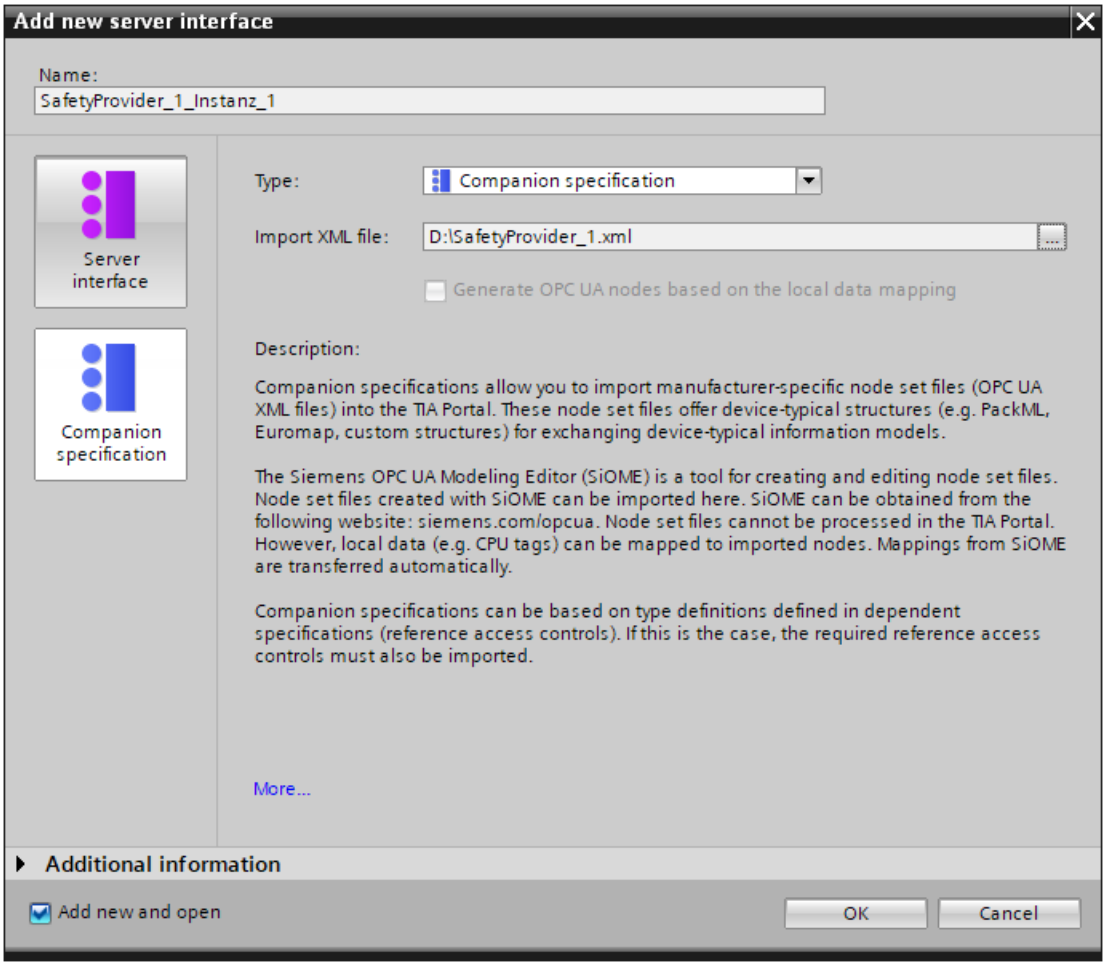
Creating a server interface for an OPC UA SafetyProvider information model

Proceed as follows to create a server interface for a companion specification with *STEP 7 (TIA Portal)*:

1. Select the F-CPU you are using as the OPC UA server.
2. In the project tree, click "OPC UA communication > Server interfaces".
3. Double-click "Add new server interface".
4. To select the type of this server interface, click the "Companion specification" button.

   A general name for the new interface, e.g. "Server_Interface_1", is entered in the dialog.

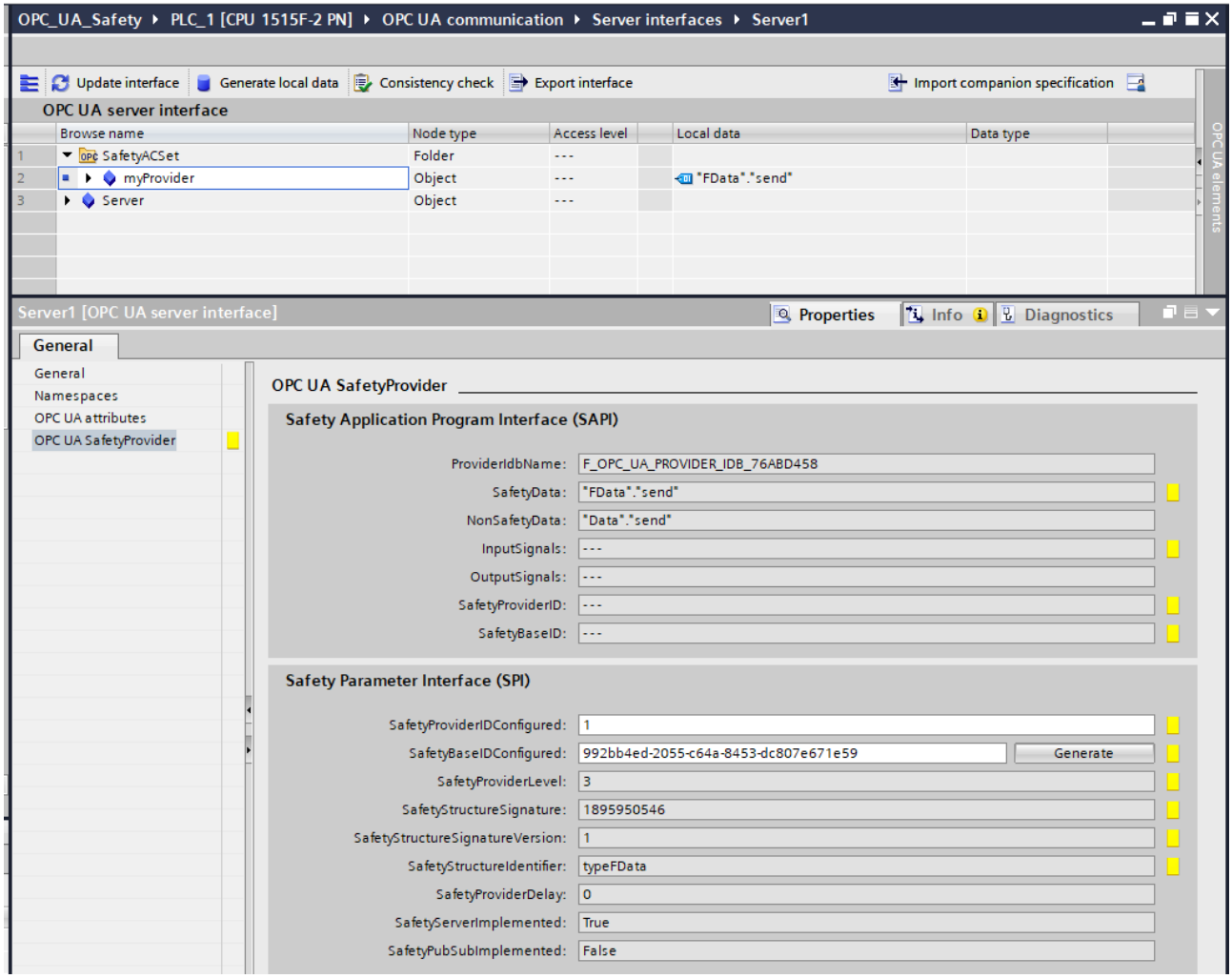
5. Change the name of the new server interface to a descriptive name in your project.
   The name "SafetyProvider_1_Instance_1" is used in the example.



6. In the "Import XML file" field, select the OPC UA SafetyProvider XML file that was created and exported in SiOME (Page 308) and contains the description of the information model for your individual data structure.
7. Click the "OK" button.
   STEP 7 (TIA Portal) imports the information model described in the selected XML file.
8. Select your SafetyProvider instance in the OPC UA server interface.
   Under "OPC UA Safety Provider" on the "Properties" tab, you see the local data of your Safety program (or standard user program) that was mapped to the SAPI parameters and the values of the SPI parameters that you assigned or that were assigned automatically during import.

NOTE

You performed the mapping of your local data previously when creating your OPC UA Safety XML file in SiOME. It is not necessary to create and map further local data with the "Generate local data" button.



9. Optional:

   Check the imported server interface for consistency.

   To do so, select your SafetyProvider instance in the OPC UA server interface and click the "Consistency check" button.

   STEP 7 displays the result of the consistency check on the "Info > General" tab.

10. Optional:

    Change the values of the SPI parameter SafetyProviderIDConfigured and/or SafetyBaseIDConfigured according to the requirements of your specific application.

    To do this, open the editor for the server interface and select "OPC UA Safety Provider" select the "Properties" tab.

    – SafetyProviderIDConfigured: Overwrite a previously entered value.
    – SafetyBaseIDConfigured: Click the "Generate" button.
      STEP 7 Safety generates a random value that fulfills the safety requirements of OPC UA Specification OPC 10000-15.

⚠ WARNING

For the unique and secure authentication of a SafetyProvider, the combination of active SafetyProviderID and active SafetyBaseID at any point in time must be unique for all OPC UA Safety communication connections network-wide. **

If your network is not safely separated from the outside world through physical separation, the active SafetyBaseID at any point in time must be a value that was generated randomly in accordance with the OPC UA Specification OPC 10000-15. ***

You must ensure the uniqueness in the safety summary during acceptance of the safety program. If the active SafetyProviderID or active SafetyBaseID is not formed until runtime of your Safety program, you must ensure the correct implementation of the algorithm for the generation of the values. *(S102)*

* If you cannot ensure a unique combination during startup of the Safety program because the active SafetyProviderID or active SafetyBaseID is not formed until runtime of your Safety program, you must ensure that the active SafetyProviderID or active SafetyBaseID is "0" during this phase.

** SafetyProviders to which the same local data element of your safety program is mapped (Safety Multicast (Page 330)) at the SafetyData SAPI parameter are permitted to use the same combination.

*** If you have entered a value for the SPI parameter SafetyBaseIDConfigured of a SafetyProvider (either in the NodeSet or in the properties of a SafetyProvider), you yourself must ensure that the value meets the safety requirements of the OPC UA Specification OPC 10000-15. Alternatively, generate the value of SafetyBaseIDConfigured again by opening the properties tab of the affected SafetyProvider and clicking on the "Generate" button.

9.5.2.4 Programming the initialization or evaluation of SAPI parameters in your safety program (or standard user program)

Programming procedure

You program the initialization or evaluation of SAPI parameters as follows:

1. Initialize the local data in your Safety program that you have mapped to the SafetyData SAPI parameter with the current values that you want to make available to a connected SafetyConsumer in a safety-related manner.
2. Optional:

   Initialize the local data in your standard user program that you have mapped to the NonSafetyData SAPI parameter with the current values that you want to make available to a connected SafetyConsumer in a non-safety-related manner.
3. Optional:

   If you want to perform user acknowledgment required on the connected SafetyConsumer on the SafetyProvider side:

   Initialize the local data element in your Safety program that you have mapped to the InputSignals.OperatorAckProvider SAPI parameter with your acknowledgment signal. For this purpose, you can also evaluate the local data element that you have mapped to the InputSignals.OperatorAckProvider SAPI parameter in your standard user program. You

can then, for example, have an operator control and monitoring system display that there is an acknowledgment on the connected SafetyConsumer.

⚠ WARNING

A user acknowledgment (Page 190) must always take place with a signal that was generated by manual operation. Implementation with an automatically generated signal is not permitted. *(S011)*

4. Optional:

   If a connected SafetyConsumer should use the designated substitute values from the F-OPC-UA-Consumer-DB instead of the current values at the SafetyData SAPI parameter:

   Initialize the local data element in your Safety program that you have mapped to the InputSignals.ActivateFSV SAPI parameter with "TRUE".

   NOTE

   Always set the date to "TRUE" if you cannot provide any safety-related values for the SafetyData SAPI parameter, while, for example, the F-CPU with the associated SafetyConsumer is already integrated in a productive operation.

   Non-safety-related values are present, for example, if your Safety program with the SafetyProvider is in disabled Safety mode during a test phase, is running on an *S7-PLCSIM* or is still incomplete.

5. Optional:

   If you want to inform a connected SafetyConsumer that the current values at the SAPI parameter SafetyData are test data and, thus, were not formed in a safety-related manner:

   Initialize the local data element in your safety program that you have mapped to the InputSignals.EnableTestMode SAPI parameter with "TRUE".

   NOTE

   The EnableTestMode SAPI parameter is also set internally to "TRUE" by the Safety program if it is in disabled Safety mode or is running on an *S7-PLCSIM*.

6. Optional:

   Applications with changing communication partners (Page 332) or series machines (Page 331):

   Initialize the local data element in your Safety program that you have mapped to the SafetyProviderID SAPI parameter with a SafetyProviderID matched to the specific situation in the system.

7. Optional:

   Applications with changing communication partners (Page 332) or series machines (Page 331):

   Initialize the local data element in your standard user program that you have mapped to the SafetyBaseID SAPI parameter with a SafetyBaseID matched to the specific situation in the system.

⚠ WARNING

For the unique and secure authentication of a SafetyProvider, the combination of active SafetyProviderID and active SafetyBaseID at any point in time must be unique for all OPC UA Safety communication connections network-wide. **

If your network is not safely separated from the outside world through physical separation, the active SafetyBaseID at any point in time must be a value that was generated randomly in accordance with the OPC UA Specification OPC 10000-15. ***

You must ensure the uniqueness in the safety summary during acceptance of the safety program. If the active SafetyProviderID or active SafetyBaseID is not generated until runtime of your Safety program, you must ensure the correct implementation of the algorithm for the generation of the values. *(S102)*

* If you cannot ensure a unique combination during startup of the Safety program because the active SafetyProviderID or active SafetyBaseID is not formed until runtime of your Safety program, you must ensure that the active SafetyProviderID or active SafetyBaseID is "0" during this phase.

** SafetyProviders to which the same local data element of your safety program is mapped (Safety Multicast (Page 330)) at the SafetyData SAPI parameter are permitted to use the same combination.

*** If you have entered a value for the SPI parameter SafetyBaseIDConfigured of a SafetyProvider (either in the NodeSet or in the properties of a SafetyProvider), you yourself must ensure that the value meets the safety requirements of the OPC UA Specification OPC 10000-15. Alternatively, generate the value of SafetyBaseIDConfigured again by opening the properties tab of the affected SafetyProvider and clicking on the "Generate" button.

⚠ WARNING

Due to system constraints, formation of a SafetyBaseID at runtime of your Safety program is only possible in your standard user program.

Note that these program sections are therefore not captured by the collective F-signature and are not protected by "Access protection for safety-related project data (Page 100)" or "F-CPU access protection (Page 103)".

For this reason, you must take organizational measures (Page 651) to ensure that only authorized persons can change these program sections.

Print the relevant program sections and supplement your safety summary with this printout. *(S103)*

9.5.3 Configuring and programming a SafetyConsumer

Requirements

- An F-CPU S7-1500 as of FW 4.0 (except HF-CPUs) or an S7-1500 F Software Controller as of firmware V40.0.
- The required Runtime license for OPC UA is available and is configured in *STEP 7*.
- The OPC UA client is activated.
- An OPC UA SafetyProvider information model for your individual data structure has been created or supplied for the device from a third-party manufacturer.

Configuring and programming a SafetyConsumer

Proceed as follows to configure and program a SafetyConsumer. Repeat the following steps for each SafetyProvider from which you want to evaluate data.

9.5.3.1 Creating a client interface for an OPC UA SafetyProvider information model

You create a client interface for an OPC UA SafetyProvider information model using the editor for client interfaces as in the standard. More information can be found in the *STEP 7* help under "Creating client interfaces (https://docs.tia.siemens.cloud/r/en-us/v20/configuring-automation-systems/using-opc-ua-communication-s7-1200-s7-1500-s7-1500t/using-the-s7-1500-cpu-as-an-opc-ua-client-s7-1500-s7-1500t/creating-client-interfaces-s7-1500-s7-1500t)".

Only the special features of OPC UA Safety are described below.

⚠ WARNING

Note that the data of a SafetyProvider can only be received safety-oriented if it is provided by a SafetyConsumer in the associated F-OPC UA Consumer DB. All other options for communication via OPC UA with a SafetyProvider or SafetyConsumer (OPC UA Read/Write access, OPC UA method calls) do not take place safety-oriented. *(S113)*

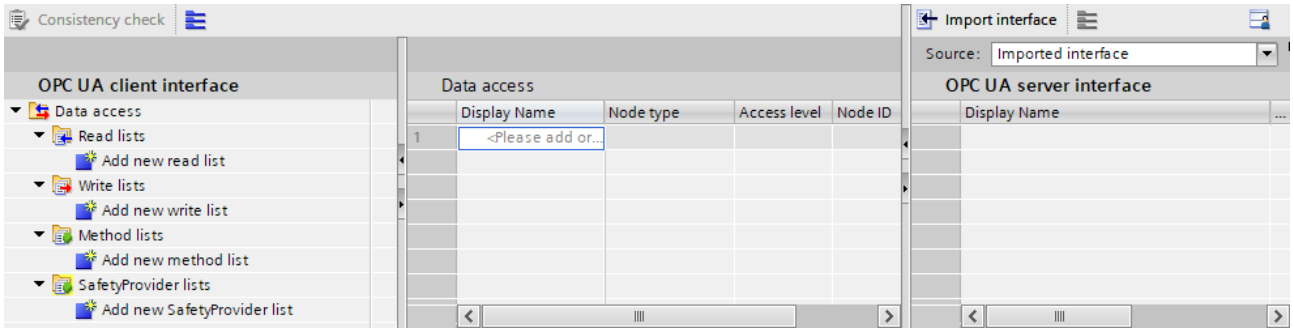
Creating a client interface for an OPC UA SafetyProvider information model

To create a client interface for an OPC UA SafetyProvider information model with *STEP 7 (TIA Portal)*, proceed as follows:

1. Select the F-CPU you are using as the OPC UA client.
2. In the project tree, click "OPC UA communication > Client interfaces".

3. Double-click on "Add new client interface".

   STEP 7 creates a new client interface and displays it in the editor:



   STEP 7 calls the new interface "Client interface_1".

   STEP 7 also generates the following data blocks:

   – **Client interface_1_Configuration**

     The data block contains variables that require to call the client instructions for the underlying OPC UA communication.

     When using the F-OPC-UA-Consumer-Com-FB (Page 328), these variables are automatically interconnected with the OPC_UA_MethodCall instruction.

   – **Client interface_1_Data**

     This data block is not required for OPC UA Safety.

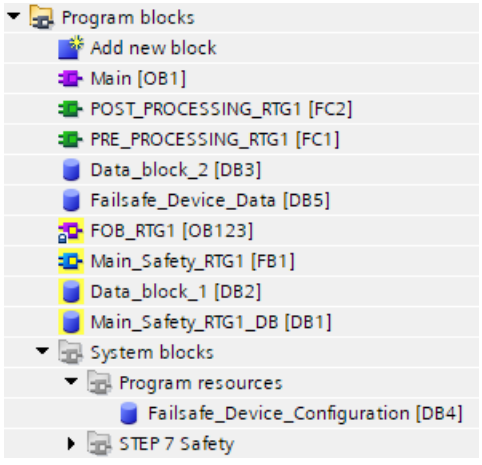
4. Change the name of the new client interface to a descriptive name in your project.

   The name "Failsafe_Device" is used in the example.

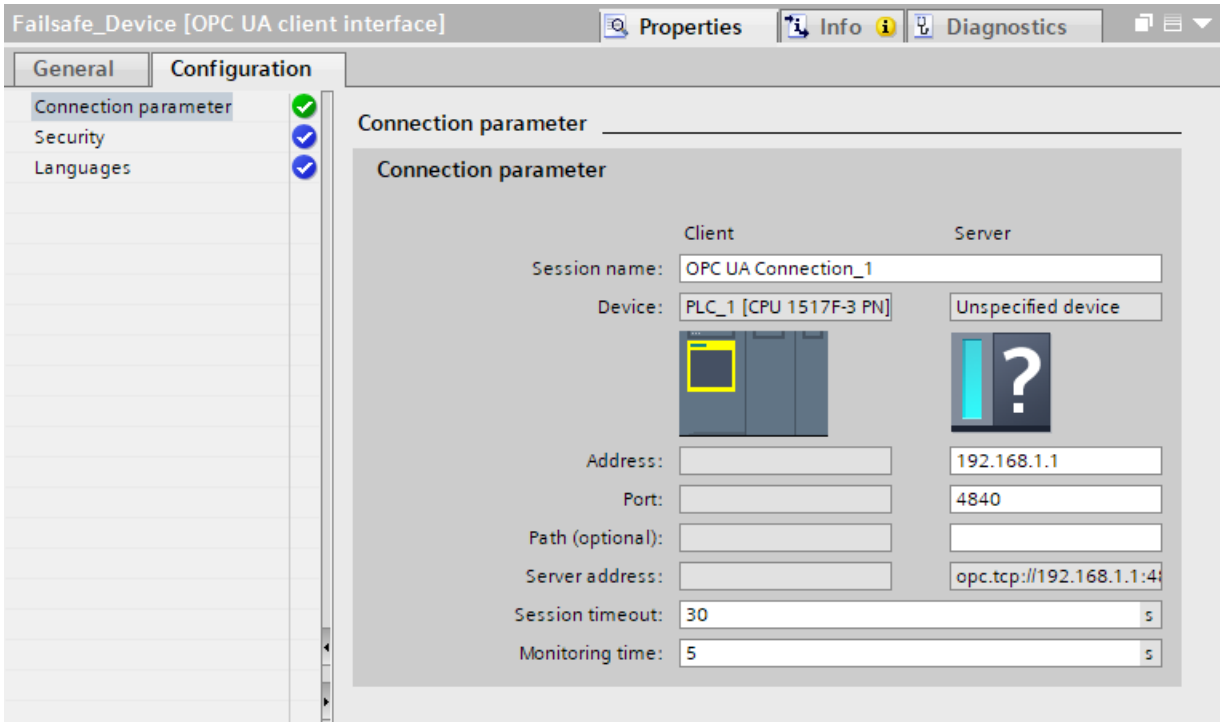
   This also changes the names of the associated data blocks to:

   – Failsafe_Device_Data

   – Failsafe_Device_Configuration



5. In the dialog that appears, specify the IP address of the desired OPC UA server in the Properties/Configuration tab under Connection parameters.



6. To import an OPC UA server interface, click the "Import interface" button at the top right of the editor when it is selected.

   Alternatively: You determine the server interface of a connected OPC UA server online. More information can be found in the STEP 7 online help under "Determining the server interface online (https://docs.tia.siemens.cloud/r/en-us/v20/configuring-automation-systems/using-opc-ua-communication-s7-1200-s7-1500-s7-1500t/using-the-s7-1500-cpu-as-an-opc-ua-client-s7-1500-s7-1500t/determine-server-interface-online-s7-1500-s7-1500t)".

7. In the "Import XML file" field of the dialog that appears, select the exported OPC UA SafetyProvider XML file (Page 308) created and exported in SiOME or the OPC UA SafetyProvider XML file supplied for a device by a third-party manufacturer, in which the information model of the intended SafetyProvider is described.

8. Generate a SafetyProvider list in this client interface.

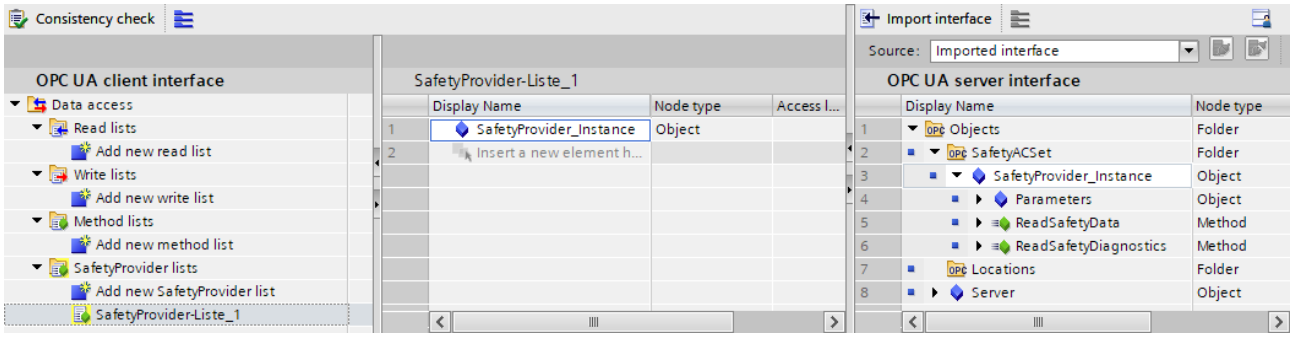
   To do this, click on "Add new SafetyProvider list" in the left section of the editor.

   STEP 7 adds a new list with the name "SafetyProvider list_1".

9. Add the intended SafetyProvider to the new SafetyProvider list.

   To do this, drag-and-drop the SafetyProvider from the OPC UA server interface to the "SafetyProvider list_1" SafetyProvider list in the middle field of the editor.

   In the example, the SafetyProvider has the name "SafetyProvider_Instance".



STEP 7 Safety generates the following program blocks and PLC data types:

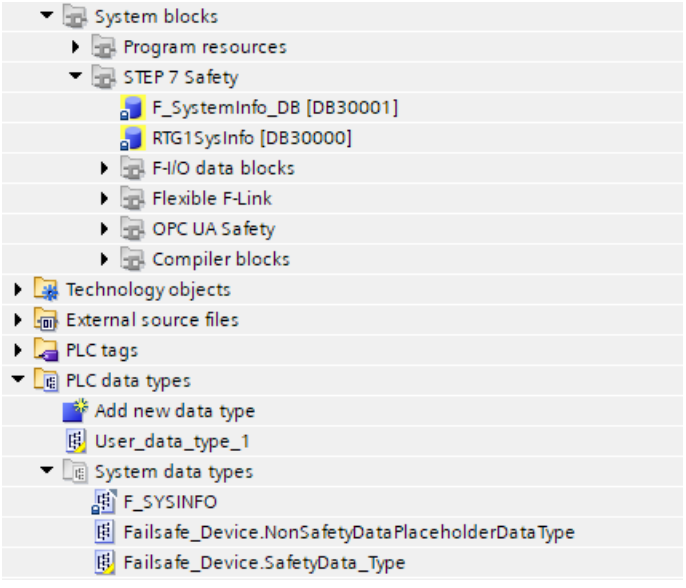
- F-OPC-UA-Consumer-DB (Page 304)

  (for example F_OPC_UA_CONSUMER)
- F-OPC-UA-Consumer-Com-FB (Page 328)

  (for example F_OPC_UA_CONSUMER_COM)
- UDT for the SafetyData SAPI parameter (Page 299) and the PASS_VALUES in the F-OPC-UA-Consumer-DB (Page 304)

  (for example Failsafe_Device.SafetyData_Type)
- UDT for the NonSafetyData SAPI parameter (Page 299)

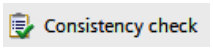
  (for example Failsafe_Device_NonSafetyDataPlaceholderDataType)



10. Optional:

    Check the imported client interface for consistency.

    To do this, select your SafetyProvider list in the OPC UA client interface and click the "Consistency check" button.

    STEP 7 displays the result of the consistency check on the "Info > General" tab.

9.5.3.2 Configuring properties of a SafetyConsumer

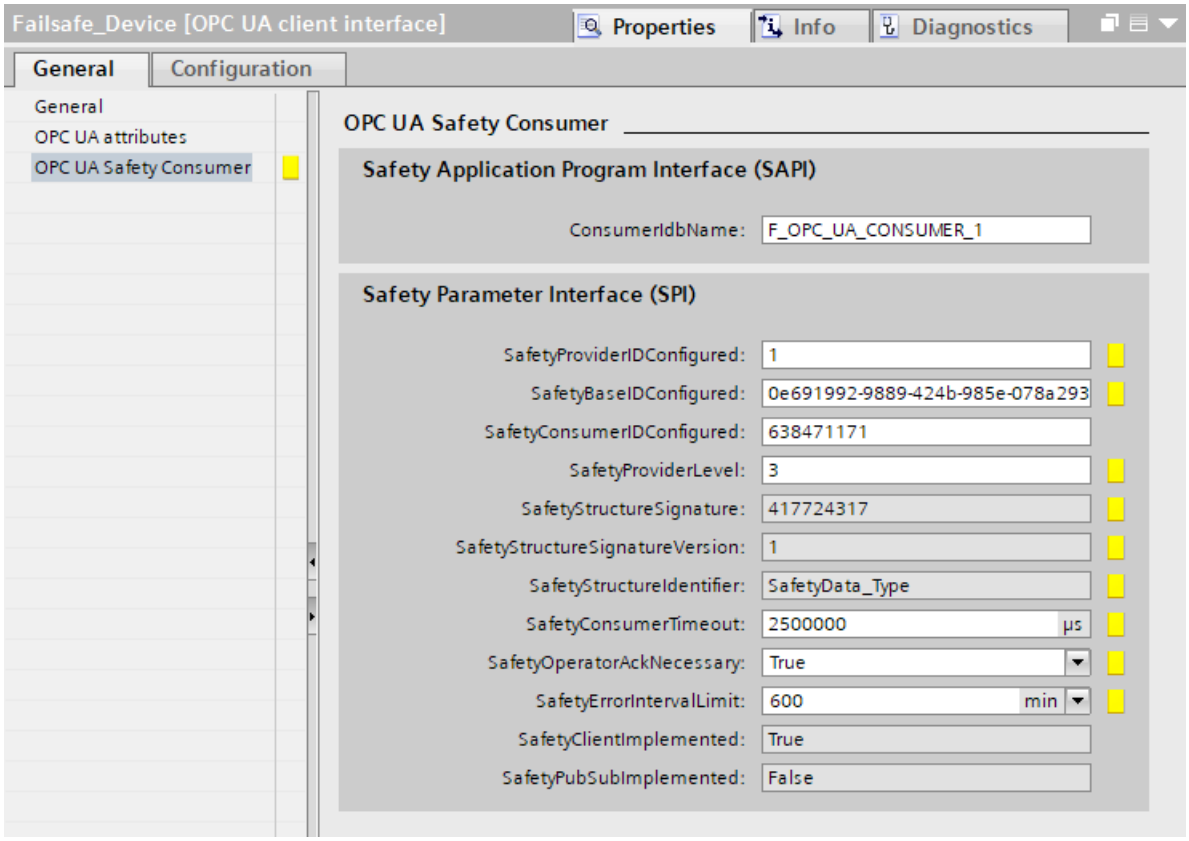
You configure the properties of a SafetyConsumer in the "Properties" tab in the editor for client interfaces or in the F-OPC UA Consumer DB. Note the requirements of your specific application.

Configuring properties of a SafetyConsumer

To configure the properties of a SafetyConsumer, proceed as follows:

1. Select the intended SafetyProvider in the SafetyProvider list.

   In the "Properties/General" tab, the F-OPC UA Consumer DB is displayed under "OPC UA Safety Consumer" as a substitute for the SAPI parameters and the SPI parameters.



2. Optional:

   Change the name of the F-OPC UA Consumer DB.

3. Specify the values of the SPI parameters SafetyProviderIDConfigured (Page 295) and SafetyBaseIDConfigured (Page 295). The values must match the corresponding values of the intended SafetyProvider.

   ⚠ WARNING

   For secure authentication of the SafetyProvider whose SafetyData is to be evaluated, the values of the active SafetyProviderID and active SafetyBaseID of the SafetyConsumer must correspond to the values of the active SafetyProviderID and active SafetyBaseID of this SafetyProvider at all times*.

   When accepting the Safety programs, you must ensure compliance in the relevant safety summary. If the active SafetyProviderID or active SafetyBaseID is not generated until runtime of your Safety program, you must ensure the correct implementation of the algorithm for the generation of the values. *(S104)*

   * If you cannot yet provide the values when starting the Safety program because they are only created during the runtime of your Safety program, you must ensure that the active SafetyProviderID or active SafetyBaseID or the Enable "0" SAPI parameter in this phase.

4. Optional:

   Change the value of the SPI parameter SafetyConsumerIDConfigured (Page 295).

5. Specify the value of the SafetyProviderLevel (Page 295) SPI parameter. The value must match the value on the connected SafetyProvider.

6. Change the value of the SPI parameter SafetyConsumerTimeout (Page 295).

   More information can be found in the section "Response Times of Safety Functions (Page 635)".

7. Optional:

   Change the value of the SafetyOperatorAckNecessary (Page 295) SPI parameter to "FALSE" if your specific system situation allows the values on the SafetyData SAPI parameter to be switched from substitute values to current values without manual user acknowledgment.

8. Optional:

   Change the value of the SafetyErrorIntervalLimit (Page 295) SPI parameter if your specific system situation requires this **as an exception**.

   Note the effects on the probability of failure values (PFDavg and PFH values) (Page 446).

9. Define the substitute values that are to be output instead of the current values at the SafetyData SAPI parameter in certain communication states.

   To do this, open the F-OPC UA Consumer DB belonging to your SafetyConsumer. You can find this in the project tree in the folder "Program blocks/System blocks/STEP 7 Safety/OPC UA Safety".

   Define the substitute values as start values in the PASS_VALUES (Page 304) variable. The data structure of the PASS_VALUES variable corresponds exactly to the data structure of your SafetyData and is specified by an automatically created PLC data type (UDT).

   Alternatively:

   If you use multiple SafetyConsumers with identical data structure (combination of SafetyData and NonSafetyData) and want to define identical substitute values for all locations of use, you can alternatively proceed as follows:

Open the automatically created PLC data type (UDT). You can find this in the project tree in the folder "PLC data types/system data types".

Define the substitute values as default values in this PLC data type (UDT).

9.5.3.3 Programming the initialization or evaluation of SAPI parameters in your Safety program (or standard user program)

Programming procedure

You program the initialization or evaluation of SAPI parameters as follows:

1. In your Safety program, set the ENABLE variable in the F-OPC-UA-Consumer-DB to "1" if you want to activate communication with OPC UA Safety to the connected SafetyProvider.

   Set the ENABLE variable to "0" if you want to deactivate an active communication to the connected SafetyProvider.

   NOTE

   When changing from Enable = 1 to Enable = 0, the output of substitute values at the SafetyData SAPI parameter only take effect after the next data package has been received or after the monitoring time parameterized via SafetyConsumerTimeout.

2. Optional:
   If you want to perform user acknowledgment on the SafetyConsumer side: In your Safety program, supply the OPERATOR_ACK_CONSUMER variable in the F-OPC-UA-Consumer-DB with your acknowledgment signal.
   You can evaluate the OPERATOR_ACK_REQUESTED variable for this. This allows you, for example, to display whether an acknowledgment request has been received on an operator control and monitoring system.

   ⚠ WARNING

   A user acknowledgment (Page 190) must always take place with a signal that was generated by manual operation. Implementation with an automatically generated signal is not permitted. *(S011)*

3. Applications with changing communication partners (Page 332) or series machines (Page 331):

   In your Safety program, supply the SAFETY_PROVIDER_ID variable in the F-OPC-UA-Consumer-DB with the corresponding value of the currently intended SafetyProvider.

4. Applications with changing communication partners (Page 332) or series machines (Page 331):

   In your standard user program, supply the SAFETY_BASE_ID variable in the F-OPC-UA-Consumer-DB with the corresponding value of the currently intended SafetyProvider.

WARNING

For secure authentication of the SafetyProvider whose SafetyData is to be evaluated, the values of the active SafetyProviderID and active SafetyBaseID of the SafetyConsumer must correspond to the values of the active SafetyProviderID and active SafetyBaseID of this SafetyProvider at all times*.

When accepting the Safety programs, you must ensure compliance in the relevant safety summary. If the active SafetyProviderID or active SafetyBaseID is not generated until runtime of your Safety program, you must ensure the correct implementation of the algorithm for the generation of the values. *(S104)*

* If you cannot yet provide the values when starting the Safety program because they are only created during the runtime of your Safety program, you must ensure that the active SafetyProviderID or active SafetyBaseID or the Enable "0" SAPI parameter in this phase.

WARNING

Due to system constraints, formation of a SafetyBaseID at runtime of your Safety program is only possible in your standard user program.

Note that these program sections are therefore not captured by the collective F-signature and are not protected by "Access protection for safety-related project data (Page 100)" or "F-CPU access protection (Page 103)".

For this reason, you must take organizational measures (Page 651) to ensure that only authorized persons can change these program sections.

Print the relevant program sections and supplement your safety summary with this printout. *(S103)*

5. Applications with changing communication partners (Page 332) or series machines (Page 331):

   In your Safety program, supply the SAFETY_CONSUMER_ID variable in the F-OPC-UA-Consumer-DB with a SafetyConsumerID that is adapted to the specific system situation.

6. In your Safety program, evaluate the SAFETY DATA variable in the F-OPC-UA-Consumer-DB. This supplies the SafetyData currently provided by the SafetyProvider in a safety-related manner or, in an error case, the intended substitute values from the PASS VALUES variable.

7. Optional:

   In your standard user program, evaluate the NON_SAFETY_DATA variable in the F-OPC-UA-Consumer-DB with the NonSafetyData currently not provided by the SafetyProvider in a safety-related manner.

8. Optional:

   Use the FSV_ACTIVATED variable in the F-OPC-UA-Consumer-DB to evaluate whether current values or substitute values are output at the SAFETY_DATA variable.

9. Optional:

   Use the TEST_MODE_ACTIVATED variable in the F-OPC UA consumer DB to evaluate whether "only" test data, i.e. data not created for safety-related purposes, is available at the SAFETY_DATA variable.

⚠ WARNING

If the TEST_MODE_ACTIVATED variable in the F-OPC-UA-Consumer-DB is set to "1" during communication with OPC UA Safety, you can no longer assume that the data at the SAFETY_DATA variable is formed in a safety-related manner. This is the case when the connected SafetyProvider is in deactivated safety mode on a PLCSIM instance or the EnableTestMode SAPI parameter of the connected SafetyProvider has been set in the associated user program. During this time, you must take organizational measures (Page 651) to ensure the safety of the system components that are affected by this data. Alternatively, you can also use safe substitute values in your Safety program with TEST_MODE_ACTIVATED = 1 instead of the data in the SAFETY_DATA tag. *(S108)*

9.5.3.4 Programming underlying OPC UA communication ("client instructions") using the F-OPC UA Consumer Com FB

OPC UA Safety is based on the mechanisms of OPC UA. SIMATIC Safety supports OPC UA client/server communication. To use OPC UA Safety communication, you first need to call the required client instructions in the standard user program for each SafetyConsumer (see *STEP 7* help "Useful information about the client instructions (https://docs.tia.siemens.cloud/r/en-us/v20/configuring-automation-systems/using-opc-ua-communication-s7-1200-s7-1500-s7-1500t/using-the-s7-1500-cpu-as-an-opc-ua-client-s7-1500-s7-1500t/useful-information-about-the-client-instructions-s7-1500-s7-1500t)" and "Client instructions (https://docs.tia.siemens.cloud/r/en-us/v20/communication-s7-1200-s7-1500/opc-ua-s7-1200-s7-1500/client-instructions-s7-1500)).

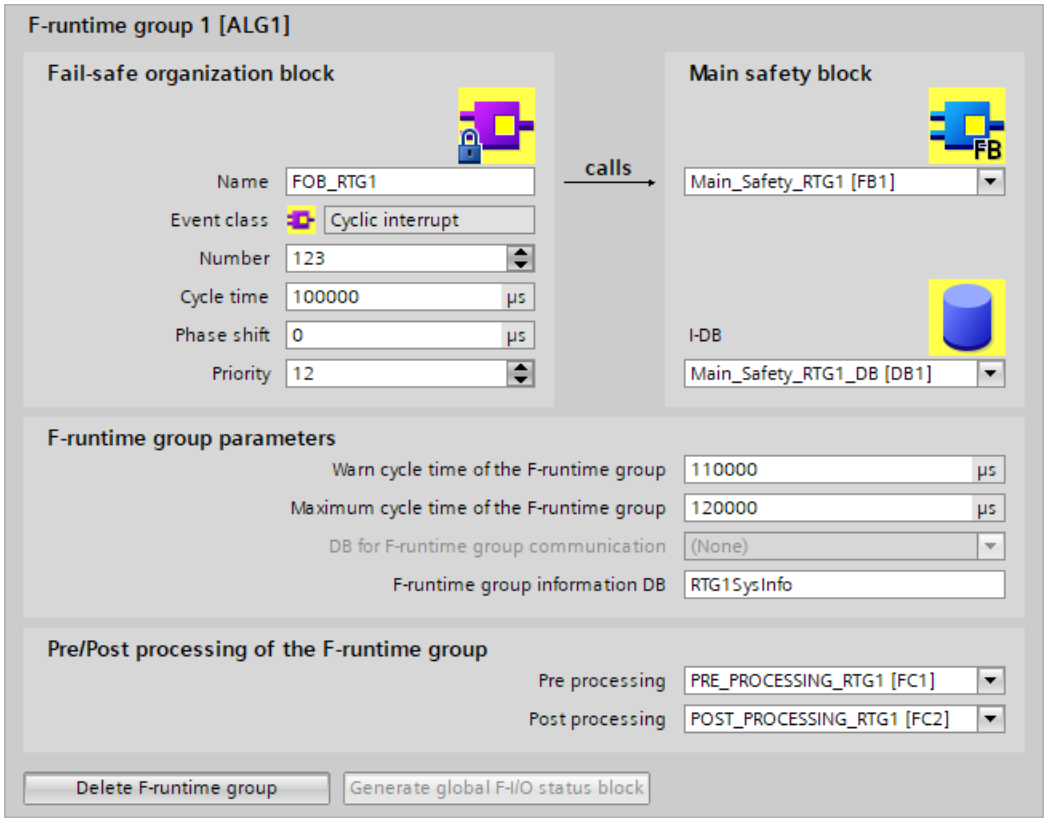
To make handling in the context of OPC UA Safety easier for you, *STEP 7 Safety* has already automatically generated an F-OPC UA Consumer-Com FB when adding the foreseen SafetyProvider in the SafetyProvider list of the client interface (Page 320). This already contains the call of the
client instruction OPC_UA_MethodCall required for a specific SafetyConsumer and can be used by you.

In the project tree, you will find the F-OPC-UA-Consumer-Com-FB in the "Program blocks/System blocks/STEP 7 Safety/OPC UA Safety" folder.

You can find application examples for concrete implementation on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109977450).

Programming calls of the F-OPC-UA-Consumer-Com-FB

1. If no pre-processing or post-processing module (Page 88) exists in your F-runtime group, first create the corresponding standard FCs and select them in the *Safety Administration Editor* for your F-runtime group.
   The FCs PRE_PROCESSING_RTG1 or POST_PROCESSING_RTG1 are used in the example.



2. Call the F-OPC-UA-Consumer-Com-FB in the pre-processing module. To do this, set the IsPre input to "TRUE".
3. Call the F-OPC-UA-Consumer-Com-FB in the post-processing module. To do this, set the IsPre input to "FALSE".

NOTE

If you also use OPC UA "Standard" communication, you may want to organize the call of the client statements differently. You can find examples on this in the application examples on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109977450).

WARNING

You may only use the Excel file for response time calculation or for timeout calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) when using OPC UA Safety communication, if you have observed the following rules with regard to the standard instruction OPC_UA_MethodCall:

You must call up the standard instruction OPC_UA_METHODCALL to receive the data in the pre-processing of the F-runtime group (Page 88) and to send the request in the post-processing of the F-runtime group (Page 88). The REQ input of the instruction must be supplied with a positive signal edge during preprocessing. You can use the F-OPC-UA-Consumer-Com-FB provided by the system. *(S112)*

9.5.4 Special use cases for communication with OPC UA Safety

9.5.4.1 Introduction

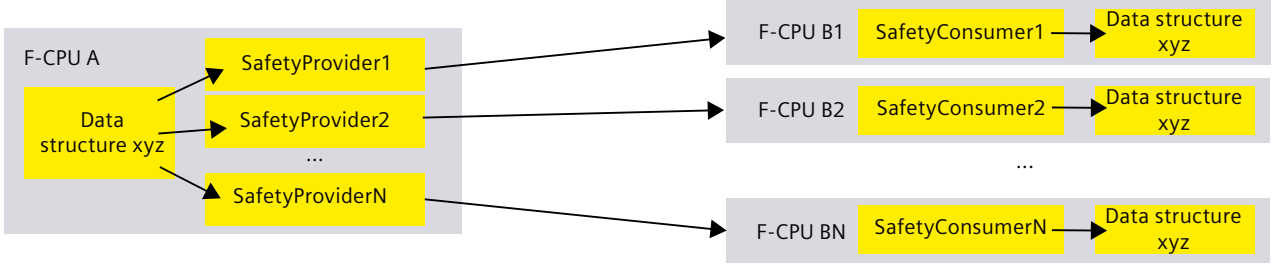
This section describes some special use cases in the context of communication with OPC UA Safety. However, the details of implementation depend on your specific application. The following descriptions are therefore to be understood as suggestions and do not claim to be exhaustive. When implementing your specific application, you must observe the corresponding warnings in this manual.

9.5.4.2 Safety Multicast

In a multicast, a communication partner provides data of several communication partners on different devices at the same time.

If you want to provide data on a F-CPU via Safety Multicast, you have to set up an individual SafetyProvider for each receiver (SafetyConsumer).

If all SafetyConsumers communicate with a shared SafetyProvider, timeout errors might occur because of competing SafetyConsumer requests.



Procedure

Proceed as follows:

1. Create a SafetyProvider by copying the underlying SafetyProviderType correspondingly often for each SafetyConsumer that evaluates it from the same associated OPC UA Safety information model on the F-CPU that provides the multicast data.

   NOTE

   As the underlying SafetyProvider already contains the mapping for SafetyData in the Safety Application Program Interface (SAPI), these SafetyProviders automatically use the same data from their Safety program.

2. Create a server interface (Page 313) on the basis of this OPC UA Safety information model.

   NOTE

   All SafetyProviders belonging to a multicast may use the same combination of active SafetyProviderID and active SafetyBaseID.

3. Create a client interface (Page 320) on each F-CPU that evaluates the multicast data and add the SafetyProvider from one of the created server interfaces (Page 313) ("SafetyConsumer_x") to a SafetyProvider list.

   NOTE

   Use a different one of the created SafetyProviders for each SafetyConsumer.

9.5.4.3 Series machines

In the context of series machines, there is usually a series machine project with a configured maximum configuration, from which different variants for a series machine can then be loaded. When a real machine is commissioned, the corresponding variant is then activated.

If the series machine project includes a Safety program, there is usually a requirement that the collective F-signature from the series machine project remains unchanged after the commissioning of a real machine.

In the context of communication with OPC UA Safety, however, it is necessary for each OPC UA Safety communication connection to have a unique combination of active SafetyProviderID and active SafetyBaseID. This means that the series machine project cannot yet contain the final values for a real machine, e.g. by defining them via the SPI parameters SafetyProviderIDConfigured and/or SafetyBaseIDConfigured.

Possible solutions:

1. Use the value "1" for the SafetyProviderIDConfigured SPI parameter in the series machine project for all OPC UA Safety communication connections.

   Note: The value is usually assigned automatically by *STEP 7 Safety*.

2. Use the value "0" for the SafetyBaseIDConfigured SPI parameter in the series machine project for all OPC UA Safety communication connections.

3. In the standard user program in the series machine project, program a block that generates a randomly generated value once for each OPC UA Safety communication connection during commissioning of a real machine and assigns it to the associated SAPI parameter SafetyBaseID.

   For this purpose, use the "Random", "Get_IM_Data", "RD_SYS_T" and "SHA2" instructions to ensure that the safety requirements of the OPC UA Specification OPC 10000-15 regarding randomly generated values are fulfilled.

You can find application examples for concrete implementation on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109977450).

9.5.4.4 Changing communication partners

There are changing communication partners when, during the runtime of a Safety program,

- a SafetyProvider is to communicate with changing SafetyConsumers
- a SafetyConsumer with changing SafetyProviders

with identical data structures of the parameters SafetyData and, if applicable, NonSafetyData.

In these situations, you must pay particular attention to the correct handling of the active SafetyProviderID, SafetyBaseID, SafetyConsumerID and the Enable SAPI parameter of the SafetyConsumer.

SafetyProvider with changing SafetyConsumers

SafetyProvider:

- Specify the SafetyProviderID and SafetyBaseID consistently via the SPI parameters SafetyProviderIDConfigured and SafetyBaseIDConfigured.

  Note: The values are usually assigned automatically by *STEP 7 Safety*.

SafetyConsumer:

- Specify the SafetyProviderID and SafetyBaseID consistently via the SPI parameters SafetyProviderIDConfigured and SafetyBaseIDConfigured. The values must match the corresponding values of the intended SafetyProvider.
- Specify the SafetyConsumerID consistently via the SafetyConsumerIDConfigured SPI parameter. Use different values for each SafetyConsumer.

  Note: *STEP 7 Safety* automatically assigns a randomly generated value.
- In your safety program, only set the ENABLE variable in the F-OPC-UA-Consumer-DB to "1" once the underlying OPC UA communication to the SafetyProvider has been established.
- In your safety program, set the ENABLE variable in the F-OPC-UA-Consumer-DB to "0" before terminating the underlying OPC UA communication to the SafetyProvider.

SafetyConsumer with changing SafetyProviders for a known number of SafetyProviders.

The SafetyConsumer knows all possible SafetyProviders and their SafetyProviderIDConfigured and SafetyBaseIDConfigured and can determine the system state of the foreseen SafetyProvider.

SafetyProvider:

- For each SafetyProvider, specify the SafetyProviderID and SafetyBaseID consistently via the SPI parameters SafetyProviderIDConfigured and SafetyBaseIDConfigured.

  Note: The values are usually assigned automatically by *STEP 7 Safety*.

SafetyConsumer:

- Specify the value "0" for the SPI parameters SafetyProviderIDConfigured and SafetyBaseIDConfigured.
- Specify the SafetyConsumerID consistently via the SafetyConsumerIDConfigured SPI parameter.

  Note: *STEP 7 Safety* automatically assigns a randomly generated value.
- Save the SafetyProviderIDConfigured and SafetyBaseIDConfigured of all possible SafetyProviders in an F-DB or Standard DB.
- On the basis of the current system state, select the value of the SafetyProviderIDConfigured of the foreseen SafetyProvider in your safety program and copy this into the variable SAFETY_PROVIDER_ID of the F-OPC-UA-Consumer-DB. Use the value 0 as long as no connection is to exist to a SafetyProvider.
- On the basis of the current system state, select the value of the SafetyBaseIDConfigured of the foreseen SafetyProvider in your safety program and copy this into the variable SAFETY_BASE_ID of the F-OPC-UA-Consumer-DB. Use the value 0 as long as no connection is to exist to a SafetyProvider.
- In your safety program, only set the ENABLE variable in the F-OPC-UA-Consumer-DB to "1" if the values of the foreseen SafetyProvider are pending and the underlying OPC UA communication to the SafetyProvider has been established at the variables SAFETY_PROVIDER_ID and SAFETY_BASE_ID of the F-OPC-UA-Consumer-DB.
- In your safety program, set the ENABLE variable in the F-OPC-UA-Consumer-DB to "0" before terminating the underlying OPC UA communication to the SafetyProvider.

SafetyConsumer with changing SafetyProvider for unknown number of SafetyProviders

Not all possible SafetyProviders and their active SafetyBase ID are known to the SafetyConsumer, since, for example, additional SafetyProviders are added during running system operation.

In this case, you must use a different method (e.g. via an RFID system) to make the active SafetyBaseID of the foreseen SafetyProvider, depending on the system status, known to the SafetyConsumer.

SafetyProvider:

- Constantly and identically specify the SafetyProviderID for each SafetyProvider via the SPI parameter SafetyProviderIDConfigured.

  Note: The value is usually assigned automatically by *STEP 7 Safety*.
- For each SafetyProvider specify the SafetyBaseID analogously to the use case for series machines (Page 331). Specify the value "0" at the SPI parameter SafetyBaseIDConfigured.

SafetyConsumer:

- Constantly specify SafetyProviderIDConfigured for the SafetyProviderID via the SPI parameters. The value must match the corresponding value of the SafetyProvider setting.
- Specify the value "0" for the SPI parameter SafetyBaseIDConfigured.
- Specify the SafetyConsumerID consistently via the SafetyConsumerIDConfigured SPI parameter.

  Note: *STEP 7 Safety* automatically assigns a randomly generated value.
- In your safety program, copy the value of the active SafetyBaseID of the intended SafetyProvider determined via the diversity path into the SAFETY_BASE_ID variable of the F-OPC-UA-Consumer-DB. Use the value 0 as long as no connection is to exist to a SafetyProvider.
- In your safety program, only set the ENABLE variable in the F-OPC-UA-Consumer-DB to "1" if the value of the foreseen SafetyProvider is pending and the underlying OPC UA communication to the SafetyProvider has been established at the variable SAFETY_BASE_ID of the F-OPC-UA-Consumer-DB.
- Set the ENABLE variable in the F-OPC-UA-Consumer-DB to "0" before closing the underlying OPC UA communication to the SafetyProvider.

You can find application examples for concrete implementation on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109977450).

9.5.5 Avoiding connection resource bottlenecks

Note that influences on the lower-level OPC UA communication may cause extended response times. As a result, "Timeout" communication errors can occur.

In particular, use of encrypted communication can lead to significantly extended response times on low and medium performance class F-CPUs.

The following factors can also affect the response times:

- Utilization of the F-CPU
- Number of OPC UA and OPC UA Safety connections
- Number of implemented server methods
- Open user communication
- Web server
- S7 connections

Take this into account when configuring the OPC UA Safety connection.

Also refer to the *STEP 7* help under "Details on OPC UA client/server connections (https://docs.tia.siemens.cloud/r/en-us/v20/editing-devices-and-networks/configure-networks/communication-via-connections/working-with-connections/connection-resources/connection-resources-s7-1500-s7-1500/details-about-opc-ua-client/server-connections-s7-1500)".

9.5.6 Using diagnostic options when communicating with OPC UA Safety

Introduction

You have all diagnostic options of OPC UA at your disposal. For example, you can diagnose the OPC UA server of the F-CPU online with common OPC UA clients, such as UaExpert. In addition, you can access nodes in the OPC UA address space and thus on SafetyProvider or SafetyConsumer in your standard user program in order to evaluate the content for diagnostic purposes.

⚠ WARNING

Note that the data of a SafetyProvider can only be received safety-oriented if it is provided by a SafetyConsumer in the associated F-OPC UA Consumer DB. All other options for communication via OPC UA with a SafetyProvider or SafetyConsumer (OPC UA Read/Write access, OPC UA method calls) do not take place safety-oriented. *(S113)*

NOTICE

Writing online access to nodes in the OPC UA address space, which have a connection to variables of your Safety program, are not permitted and can lead to a STOP of the F-CPU or to communication errors.

9.5.6.1 ReadSafetyDiagnostics

For diagnostic purposes, one OPC UA server method ReadSafetyDiagnostics is available for each SafetyProvider.

You can call the method via the OPC UA server interface. It transfers the input and output data of the last processing of the associated SafetyProvider.

More information can be found in the OPC UA Safety Specification OPC 10000-15.

9.5.6.2 SafetyConsumerType

Configuring and programming a SafetyConsumerType

In accordance with the OPC UA Safety specification OPC 10000-15 a SafetyConsumerType should be provided for each SafetyConsumer in an OPC UA information model. The SafetyConsumerType is not necessary for a functioning client-server-communication. More information can be found in the OPC UA Safety Specification OPC 10000-15.

Requirements

- An F-CPU S7-1500 as of FW 4.0 (except HF-CPUs) or an S7-1500 F Software Controller as of firmware V40.0.
- The following function right is available for the "SafetyAdmin" role: OPC UA server access. More information can be found in the help for *STEP 7* under "Users and roles with OPC UA function rights (https://docs.tia.siemens.cloud/r/en-us/v20/configuring-automation-systems/using-opc-ua-communication-s7-1200-s7-1500-s7-1500t/using-the-s7-1500-as-an-opc-ua-server-s7-1500-s7-1500t/configuring-the-opc-ua-server-s7-1500-s7-1500t/users-and-roles-with-opc-ua-function-rights-s7-1500-s7-1500t)".
- The required Runtime license for OPC UA is available and is configured in *STEP 7*.
- The OPC UA server is activated.

Configuring and programming a SafetyConsumerType

Proceed as follows to configure and program a SafetyConsumerType. Repeat the following steps for each SafetyConsumer that you want to make available to a SafetyConsumerType.

Creating an OPC UA SafetyConsumerType information model

You create an OPC UA SafetyConsumer information model for a SafetyConsumer on the basis of a nodeset file provided by the OPC Foundation for OPC UA Safety using SiOME. To do this you have to create the parameters SafetyBaseIDActive, SafetyConsumerIDActive and SafetyProviderIDActive of the SafetyConsumerType local data mappings on the SafetyConsumer previously created in TIA Portal and transfer the SPI parameters of each SafetyConsumer manually into the information model.

You then export the OPC UA SafetyConsumerType information model created in this way to an XML file (nodeset file).

Requirements

You require the name of the F-OPC-UA-Consumer-DB (Page 324) for the local data mapping of the parameters SafetyBaseIDActive, SafetyConsumerIDActive and SafetyProviderIDActive. It is assigned automatically during the creation of a SafetyConsumer and can be modified subsequently.

Creating an OPC UA SafetyConsumerType information model

Proceed as follows to create an OPC UA SafetyConsumerType XML file (nodeset file).

1. First, download the nodeset file "Opc.Ua.Safety.NodeSet2.xml" of the OPC UA Safety Specification V1.05.04 from the GitHub website of the OPC Foundation. You can find the download link here (https://github.com/OPCFoundation/UA-Nodeset/tree/Safety-1.05.04-2025-02-05/Safety).

   The "Opc.Ua.Safety.NodeSet2.xml" file serves as the basis for your individual nodeset file. It contains the type and method definitions you need to create an OPC UA SafetyConsumerType XML file.

2. Start SiOME.

3. First, import the namespace "http://opcfoundation.org/UA/Safety".

   To do so, click the "Import XML" button in the SiOME menu.

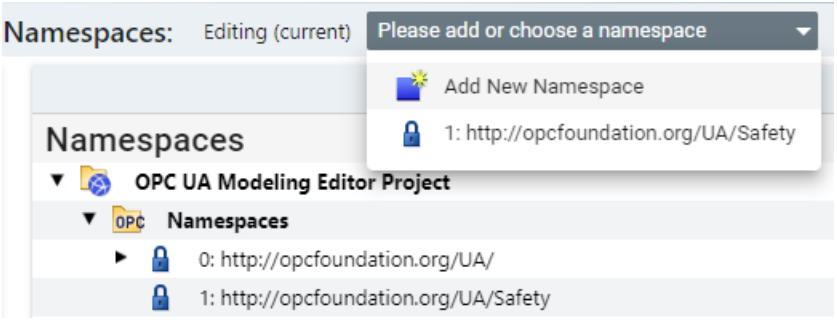
   SiOME displays the dialog for opening files.

4. Select the "Opc.Ua.Safety.NodeSet2.xml" file and click the "Open" button to import it.

   Result: SiOME imports the XML file and shows the namespace "http://opcfoundation.org/UA/Safety" in the "Namespaces" area.

5. Create your own namespace for your project.

   To do so, click "Add New Namespace" in the "Namespaces: Editing (current)" area in the SiOME menu.

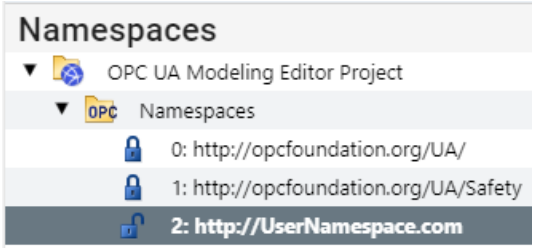
   

   SiOME opens the "Add Namespace" dialog.

6. Enter the name of the new namespace.

   The namespace "UserNamespace.com" is used in the example.

   SiOME additionally shows the new namespace:

7. Create an instance from the object type "SafetyConsumerType" of the OPC UA Safety Specification.

   To do so, right-click the folder "Root/Objects/SafetyACSet" in the "Information model" area and select "Add Instance".

   NOTE

   Instances of the "SafetyConsumerType" object type must always be located in the folder "SafetyACSet".

   SiOME displays the dialog "Add Instance".

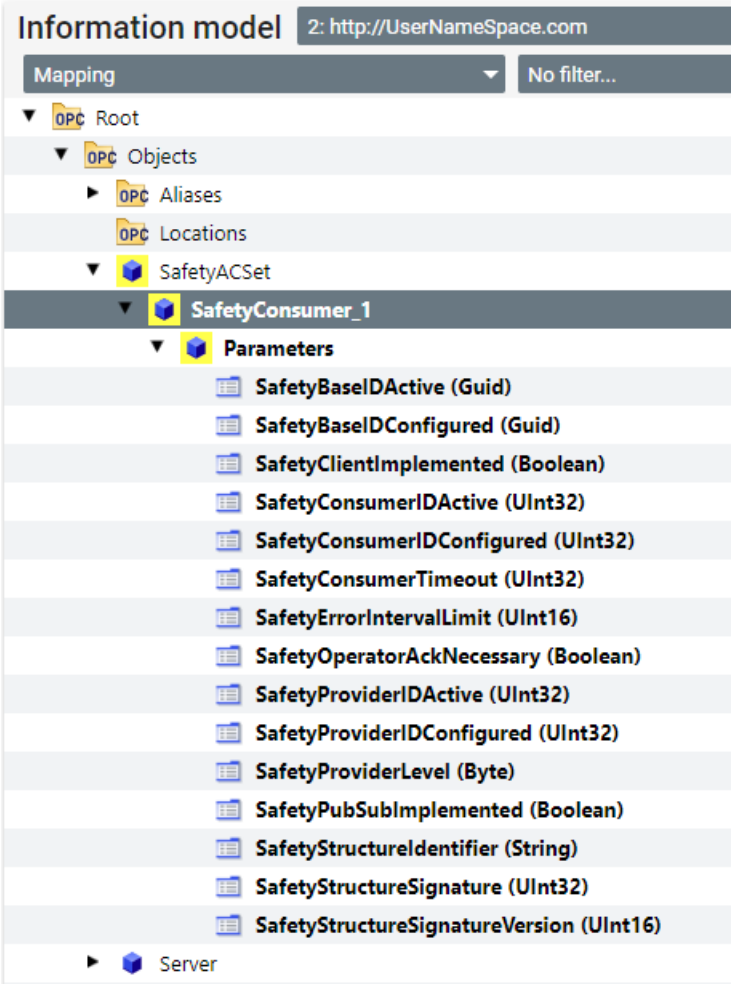
8. Enter a descriptive name for the new instance for "Name".

   The name "SafetyConsumer_1" is used in the example.

   For "TypeDefinition", select "SafetyConsumerType". You can find the type under "BaseObjectTypes/SafetyObjectsType".

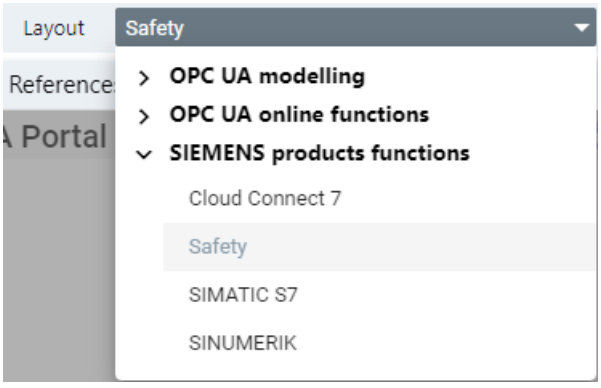
9. Click "OK".

   SiOME shows the new instance "SafetyConsumer_1" under "SafetyACSet" in the "Information model" area. This instance already contains the required parameters:



10. Select the safety layout.

    To do so, select the "Safety" layout in the SiOME menu in the "Layout" area under "SIEMENS products functions".



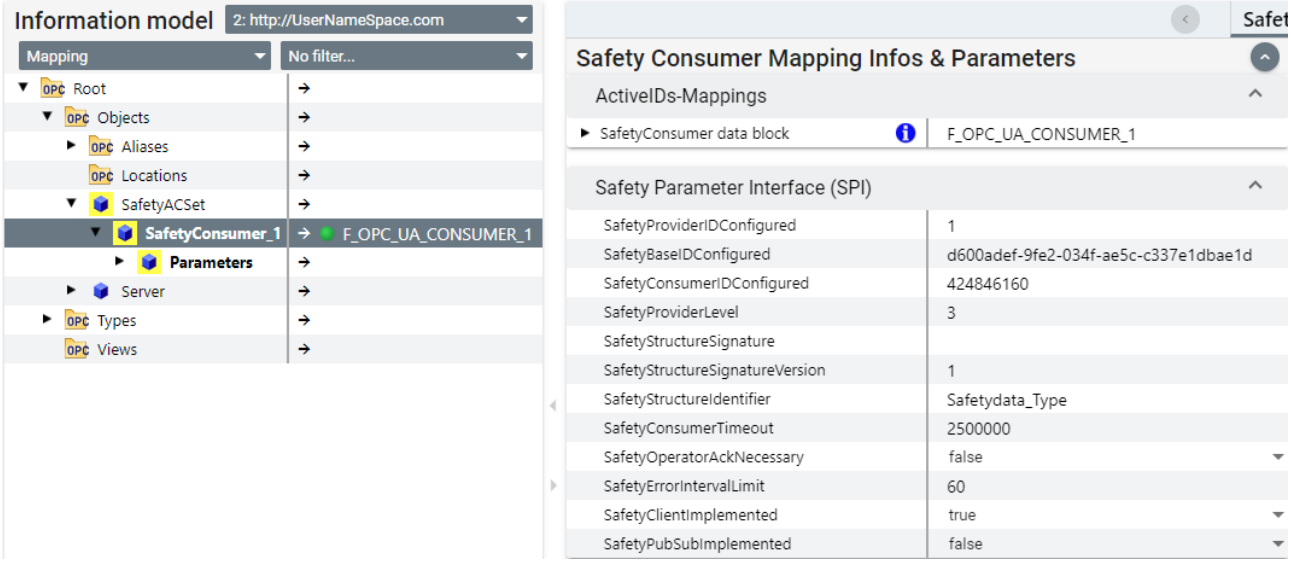
    SiOME now shows the "Safety Configuration" area.

11. Create the local data mapping for the ActiveIDs by entering the name of the F-OPC-UA-Consumer-DB in the designated field in the "ActiveIDs-Mappings" section.

    SiOME automatically generates the associated mappings for the parameters SafetyBaseIDActive, SafetyConsumerIDActive and SafeyProviderIDActive.

12. Optional:

    Transfer the values of the Safety Parameter Interface (SPI) from the SafetyConsumer from TIA Portal (Page 324) manually to the Safety Parameter Interface (SPI) of SiOME, so that this data can be queried via the OPC UA information model after completion of the configuration.



13. Export the XML file.

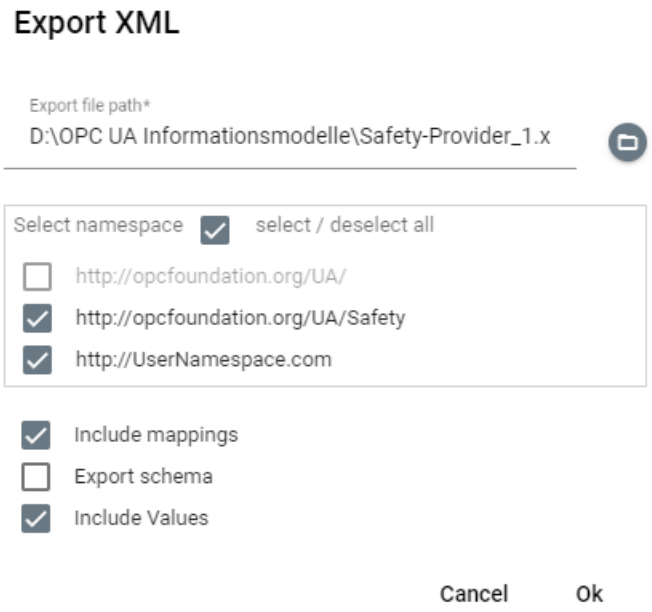
    To do so, click the "Export XML" button in the SiOME menu.



    SiOME displays the dialog "Export XML".

Select the path for the export and a descriptive name for the file. The name "SafetyConsumer_1" is used in the example.

Have all namespaces activated and click "OK".



Creating a server interface for an OPC UA SafetyConsumerType information model

You create a server interface for an OPC UA SafetyConsumerType information model in a similar way as a server interface for a companion specification. More information can be found in the *STEP 7* help under "Creating a server interface for a companion specification".

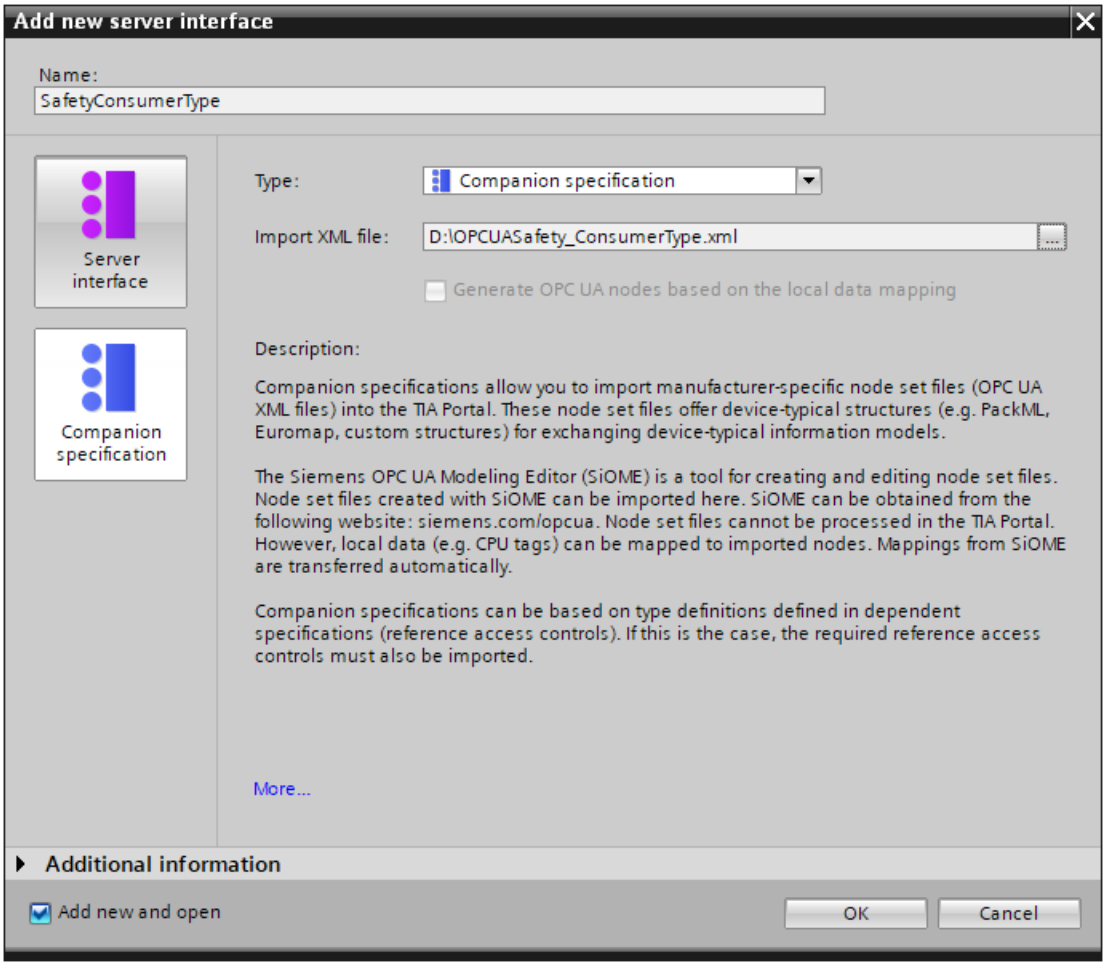
Creating a server interface for an OPC UA SafetyConsumerType information model

Proceed as follows to create a server interface for a companion specification with *STEP 7 (TIA Portal)*:

1. Select the F-CPU you are using as the OPC UA server.
2. In the project tree, click "OPC UA communication > Server interfaces".
3. Double-click "Add new server interface".
4. To select the type of this server interface, click the "Companion specification" button.

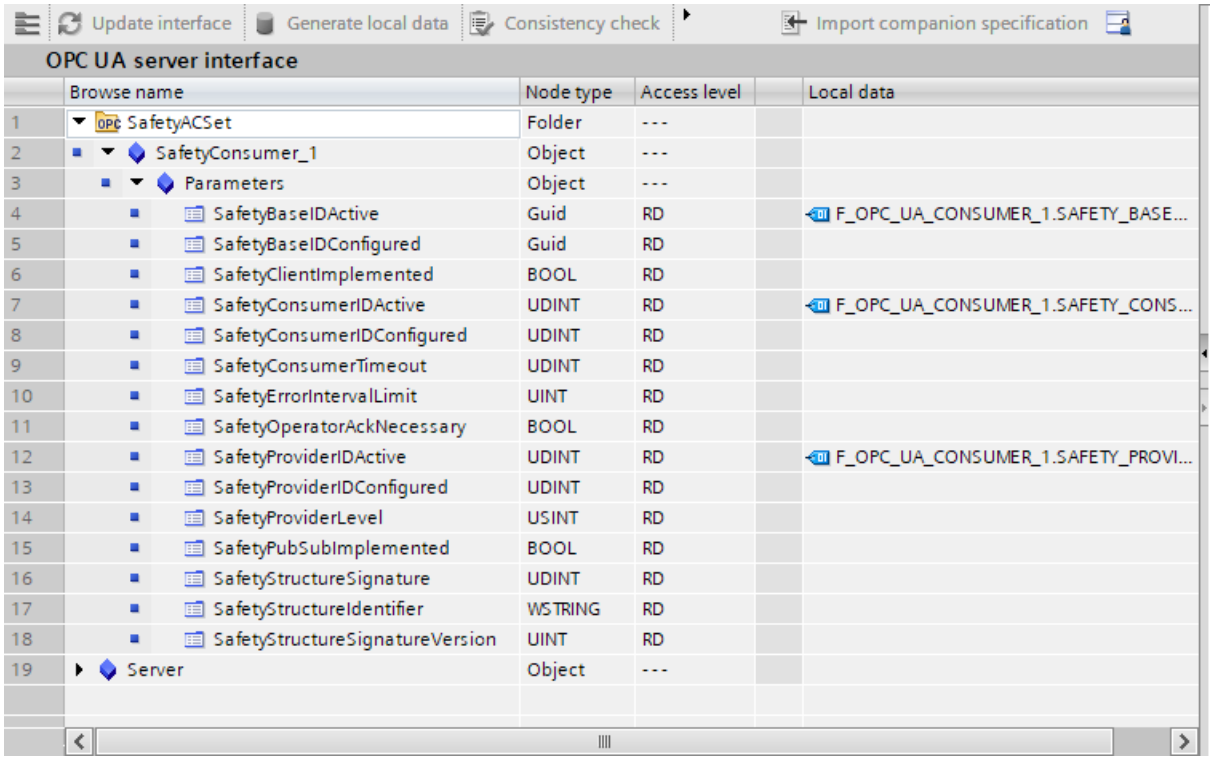
   A general name for the new interface, e.g. "Server_Interface_1", is entered in the dialog.

5. Change the name of the new server interface to a descriptive name in your project.
   In the example the name "SafetyConsumerType" is used.



6. In the "Import XML file" field, select the OPC UA SafetyConsumerType XML file that was created and exported in SiOME (Page 336) and contains the description of the information model for your individual data structure.

7. Click the "OK" button.
   STEP 7 (TIA Portal) imports the information model described in the selected XML file.



NOTE

You performed the mapping of your local data previously when creating your OPC UA Safety XML file in SiOME. It is not necessary to create and map further local data with the "Generate local data" button.

With this OPC UA server interface, you can request information on the associated SafetyConsumer with any OPC UA clients (e.g. UaExpert).

9.6 Communication with Flexible F-Link (S7-1200, S7-1500)

9.6.1 Flexible F-Link

Introduction

The fail-safe CPU-CPU communication "Flexible F-Link" is available for S7-1200 and S7-1500 F-CPUs. This means fail-safe data can be easily exchanged as fail-safe arrays with standard communication mechanisms between F-CPUs.

Flexible F-Link offers a series of advantages for exchanging fail-safe data:

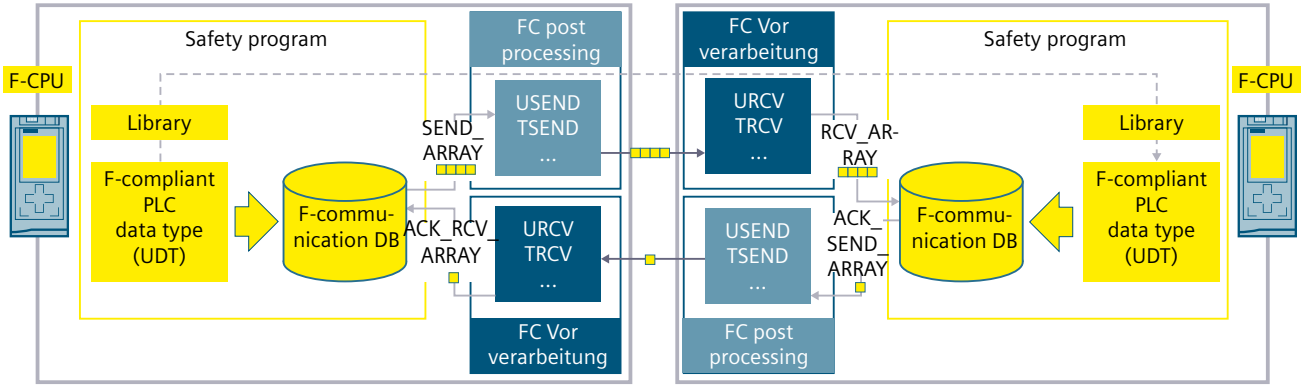
- Collection of fail-safe data to be transmitted in F-compliant PLC data types (UDTs)

  (Nested F-compliant PLC data types (UDT) are not supported.)
- Up to 1500 bytes of fail-safe data per UDT, including a maximum of 800 elements of data type BOOL.
- Easy parameter assignment and automatic generation of fail-safe communication DBs
- Transmission of fail-safe data with standard communication blocks also across network limits
- F-runtime group communication (Page 97) for F-CPUs 1200/1500
- System-integrated and globally sufficiently unique F-communication UUID
- Separate F-communication address signature for easy detection of changes to the F-communication UUID

Requirement

- S7-1500 F-CPUs as of firmware V2.0
- S7-1200 G1 F-CPUs as of firmware V4.2
- S7-1200 G2 F-CPUs as of firmware V1.0
- As of safety system version V2.2

Principles of communication via Flexible F-Link



Proceed as follows on the **send side**:

1. Create an F-compliant PLC data type (UDT) for the data to be sent. The size is up to 1500 bytes, including a maximum of 800 elements of data type BOOL.
2. Create safety-related communication with the "Send" direction in the *Safety Administration Editor* (Page 97).

   A new F-communication DB (Page 346) is created for the safety-related communication under "Program blocks\System blocks\STEP 7 Safety\F-communication DBs".
3. Set the F-monitoring time (Page 635) for the safety-related communication.

4. In the Safety program, supply the tags for the send data (SEND_DATA) in the F-communication DB (Page 346) of the safety-related communication.
5. Create standard blocks in which the fail-safe arrays created automatically in the F-communication DB are sent with the send data and received with the acknowledgment data. For processing the process values in the correct chronological order, you can make use of the F-OB Pre-/post-processing (Page 88). When using the communication instructions, ensure that the fail-safe arrays are available consistently at the time of evaluation and that the F-monitoring time (Page 635) is observed. Observe the note below.

Proceed as follows on the **receive side**:

1. Create an F-compliant PLC data type (UDT) with the same structure as on the send side.

   To do so, copy the F-compliant PLC data type (UDT) from the send side or use the project library or global library.
2. Create safety-related communication with the "Receive" direction in the *Safety Administration Editor* (Page 97).

   A new F-communication DB is created for the safety-related communication under "Program blocks\System blocks\STEP 7 Safety\F-communication DBs".
3. Copy the F-communication UUID of the safety-related communication from the send side.
4. Set the same F-monitoring time as for the send side.
5. In the Safety program, evaluate the tags for the receive data (RCV_DATA) at the F-communication DB (Page 346).
6. Create standard blocks in which the fail-safe arrays automatically created in the F-communication DB are received with the receive data and sent with the acknowledgment data. For processing the process values in the correct chronological order, you can make use of the F-OB Pre-/post-processing (Page 97). When using the communication instructions, ensure that the fail-safe arrays are available consistently at the time of evaluation and that the F-monitoring time (Page 635) is observed. Observe the note below.

NOTE

When using non-deterministic communication protocols (e.g. TCP/IP), you must take into account the following:

- Increased communication load can fundamentally impair the availability of your application (runtime of the F-monitoring time of the F-communication connection). This holds true especially when OPC UA and Secure Open User Communication (OUC) are used in parallel.
- Communication buffer overflows can adversely affect the availability of your application and should be avoided.

  Refer to the following application example for more helpful information: "Configuring Flexible F-Link Communication (https://support.industry.siemens.com/cs/ww/en/view/109768964)".

NOTE

The compilation time increases with increasing data frame size. The compilation times can rise significantly when the maximum frame length of 1500 bytes is fully used.

NOTE

During simulation with *S7-PLCSIM*, the time monitoring that generates an error message after expiration when communication with real communication partners is interrupted (e.g. by setting a CPU to STOP) is not activated. This is why no error message is displayed in this case.

⚠ WARNING

It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time. *(S018)*

⚠ WARNING

If the SENDMODE tag in the F-communication DB (Page 346) is set to "1" for receiving during communication with Flexible F-Link, you can no longer assume that the data at the RCV_DATA tag is formed in a safety-related manner.

During this time, you must take organizational measures (Page 651) to ensure the safety of the system components that are affected by this data. Alternatively, you can also use safe substitute values in your Safety program with SENDMODE = 1 instead of the data in the RCV_DATA tag. *(S086)*

⚠ WARNING

When a new Flexible F-Link communication is created in the Safety Administration Editor, a unique F-communication UUID for the communication is provided by the system. When communication is copied in the Safety Administration Editor within the parameterization table or copied to another F-CPU, the F-communication UUIDs are not regenerated and therefore not unique anymore. If the copy is used to configure a new communication relationship, you yourself must ensure the uniqueness. To do this, select the relevant UUIDs and generate them again via the "Generate UUID" context menu. You must ensure the uniqueness in the safety summary during acceptance, at the latest. *(S087)*

⚠ WARNING

For each communication connection via Flexible F-Link, you must check that the offsets of the elements of the F-compliant PLC data type (UDT) used for the communication connection match on the transmit and receive sides when you accept the Safety program.

You can find the offsets in the respective safety summary. To do this identify the communication connection by means of the UUID. *(S088)*

See also

F-runtime group communication (S7-1200, S7-1500) (Page 146)

9.6.2 F-communication DB (S7-1200, S7-1500)

F-communication DB for sending

The following table shows you the interface of the F-communication DB for the communication connection with the direction "Send":

Section	Name	Data type	Initial value	Description
Input	SEND_DATA	F-compliant PLC data type (UDT)	As in the F-compliant PLC data type (UDT).	User data to be sent.
	ACK_RCV_ARRAY	Array[0..n] of Byte	Each element with 16#0	Array with the received raw data.
Output	ERROR	BOOL	False	Signals currently pending communication errors or communication errors not acknowledged yet at the receiver (not in the initial start). 1=Communication error
	ACTIVATE_FV	BOOL	True	Communication passivated, in the initial start (for example receiver not started), or HOST sends ACTIVATE_FV. DEVICE sends status bit: FV_ACTVATED, but no 0-values. 1=The communication uses fail-safe values
	DIAG	Byte	16#0	Error bits (Timeout or CRC error currently still pending, or communication after error not depassivated yet) Bit 3: Acknowledgement request active at the receiver Bit 4: Timeout detected Bit 6: CRC error detected
	SEND_ARRAY	Array[0..n] of Byte	Each element with 16#0	Array with the raw data to be sent.
	ACK_RCV_LENGTH	UInt	0	Length information to ACK_RCV_ARRAY in bytes
	SEND_LENGTH	UInt	0	Length information to SEND_ARRAY in bytes
InOut	—	—	—	—
Static	—	—	—	—

F-communication DB for receiving

The following table shows you the interface of the F-communication DB for the communication connection with the direction "Receive":

Section	Name	Data type	Initial value	Description
Input	PASS_ON	BOOL	False	This way you can passivate the output data (output of the passivation values) 1=Enable passivation
	ACK_REI	BOOL	False	Reintegration (in case of reintegration request) by means of positive edge 1=Acknowledgment for reintegration
	RCV_ARRAY	Array[0..n] of Byte	Each element with 16#0	Array with the received raw data
Output	RCV_DATA	F-compliant PLC data type (UDT)	As in the F-compliant PLC data type (UDT).	Output data (PASS_VALUES or data received).
	ERROR	BOOL	False	Signals currently pending communication errors or communication errors not acknowledged yet (not in the initial start). 1=Communication error
	PASS_OUT	BOOL	True	At PASS_OUT=1 the PASS_VALUES are output Could be: ERROR, PASS_ON, in the initial start (e.g. sender not started), or ACK_REQ is pending (error not acknowledged)
	ACK_REQ	BOOL	False	Reintegration requirement (communication stable again after error, substitute values are still output) 1=Acknowledgment request for reintegration
	SENDMODE	BOOL	False	1=The Safety program on the sender side is in disabled Safety mode or is running on an S7-PLCSIM.
	DIAG	Byte	16#0	Error bits (Timeout or CRC error) Bit 0: Timeout detected by the sender Bit 1: Communication error currently pending in the sender Bit 2: CRC error detected by the sender Bit 4: Timeout detected by the receiver Bit 6: CRC error detected by the receiver
	ACK_SEND_ARRAY	Array[0..n] of Byte	Each element with 16#0	Array with the raw data to be sent.
	RCV_LENGTH	UInt	0	Length information of RCV_ARRAY in bytes

Section	Name	Data type	Initial value	Description
Output	ACK_SEND_LENGTH	UInt	0	Length information of ACK_SEND_ARRAY in bytes
InOut	—	—	—	—
Static	PASS_VALUES	F-compliant PLC data type (UDT)	Same as the F-compliant PLC data type (UDT) or in the I/O DB	Passivation or substitute values

Compiling and commissioning a safety program 10

10.1 Compiling the safety program

To get a consistent Safety program, follow these steps:

1. Select the "Program blocks" folder or the Safety Unit folder or a higher level.
2. Start the compile.

   To do this, proceed essentially in the same way as when compiling a standard user program. You can start at various points to accomplish this in *STEP 7*. The basics for compiling user programs can be found in the *Help on STEP 7*.

NOTE

Please note that following a safety-related change to the hardware configuration that not only this, but also the Safety program has to be recompiled and downloaded. This also applies for changes to the F-I/O which are not used in the Safety program.

NOTE

The following applies for S7-300/400 F-CPUs:

If you want to compile a know-how-protected F-block after a change, you must remove the know-how protection for this F-block before compiling.

Reporting compiling errors

You can recognize whether or not the compilation was successful based on the message in the inspector window under "Info > Compile", error messages and warnings are output.

For information on the procedure you must follow to eliminate compiling errors, see "Eliminating compiling errors" in the *Help on STEP 7*.

10.2 Safety program work memory requirements (S7-300, S7-400)

Estimation

You can estimate the work memory requirement for the safety program as follows:

Work memory required for the safety program

32 KB for F-system blocks

+ 4.4 KB for safety-related communication between F-runtime groups
+ 4.5 x work memory requirement for all F-FB/F-FCs/main safety blocks
+ 4.5 x work memory requirement of all utilized instructions, which are shown in the "Instructions" task card with the block icon.
  (except SENDDP, RCVDP, SENDS7, and RCVS7)
+ Work memory requirement of the utilized SENDDP and RCVDP instructions (4.3 KB each)
+ Work memory requirement of the utilized SENDS7 and RCVS7 instructions (8.5 KB each)

Work memory required for data

5 x work memory requirement for all F-DBs (including F-communication DB, but excluding DB for F-runtime group communication) and I-DBs for main safety block/F-FB

+ 24 x work memory requirement for all DBs for F-runtime group communication
+ 2.3 x work memory requirement for all I-DBs of instructions (except SENDDP, RCVDP, SENDS7 and RCVS7)
+ Work memory requirement of all I-DBs of instructions SENDDP (0.2 KB), RCVDP(0.3 KB), SENDS7 (0.6 KB), and RCVS7 (1.0 KB)
+ 0.7 KB per F-FC
+ 0.7 KB per F-I/O (for F-I/O DBs, etc.)
+ 4.5 KB

Block size of automatically generated F-blocks

Do not utilize the entire maximum size of an F-block, because the automatically generated F-blocks are larger and as a result, the maximum possible size may be exceeded in the F-CPU. If the block size is exceeded, this triggers a corresponding error message with information on which F-blocks are too large. These must be divided up, if necessary.

10.3 Downloading project data

Introduction

Once you have successfully compiled your safety program, you can download it together with the standard application program to the F-CPU.

(S7-1500) If the safety program is in a Safety Unit, you can load the Safety Unit independently into the F-CPU.

To download a safety program, you follow essentially the same approach as for downloading a standard user program, via different starting points in *STEP 7*.

- In the "Load preview" dialog, enter data (e.g. password for the F-CPU) and set the requirements for downloading (e.g. that the F-CPU is switched to STOP mode before downloading).
- The "Load results" dialog shows the results after loading.

We will show you the options for downloading the safety program later. For basic information on downloading, refer to the *Help on STEP 7*.

Rules for downloading the project data into an F-CPU

⚠ WARNING

(S7-300, S7-400) If **several F-CPUs** can be reached via a network (such as Industrial Ethernet) by **the same programming device or PC**, you must take the following additional measures to ensure that the safety-related project data is downloaded to the correct F-CPU:

Use passwords specific to each F-CPU, such as a uniform password for the F-CPUs with attached Ethernet address for each.

Note the following:

- A point-to-point connection must be used to activate the access protection of an F-CPU when the hardware configuration is loaded for the first time (similar to assigning an MPI address to an F-CPU for the first time).
- Before downloading the safety-related project data to an F-CPU, existing access permission for any other F-CPU must first be cancelled.
- The last download of the safety-related project data prior to switching to productive operation must be made with enabled access protection. *(S021)*

⚠ WARNING

(S7-1200, S7-1500) If **several F-CPUs** can be reached via a network (such as Industrial Ethernet) by the **same PG/PC**, you must take the following additional measures to ensure that the safety-related project data is downloaded to the correct F-CPU or that safety mode is deactivated in the correct F-CPU:

- Enter the serial number of the specific F-CPU in the *Safety Administration Editor*.

Depending on the respective CPU type (see documentation of the CPU), you have several possibilities for determining the serial number of an F-CPU:

- Print of the serial number on the housing of the F-CPU.
- Scanning of the ID link on the F-CPU.
- Display of the F-CPU
- Web server*
- Industrial Edge application user interface under /SIMATIC_MC/system/serialno* and via the UI or API of the Industrial Edge application
- TIA Portal (I&M data)*

Note the following when downloading:

- If the "Load preview" window does not contain a warning, the connection to the correct F-CPU has been established. You can confirm the download of the safety-related project data without an additional check.
- If a serial number has not yet been entered in the *Safety Administration Editor* or if the connected F-CPU has a different serial number, check the proposed serial number of the F-CPU in the "Load preview" window or "Disable safety mode" window. If you determine an error, cancel the download or the disabling of safety mode. Otherwise, you can confirm the procedure.
- When the serial number is confirmed (online), it is always applied in the *Safety Administration Editor* and any existing entry is overwritten.

  (S099)

* To make use of these possibilities, you must ensure a connection to the expected F-CPU. You achieve this through:

- A point-to-point connection to the respective F-CPU
- Verification of the flashing of the F-CPU after selecting the "Flash LED" check box in the online and diagnostics view of the F-CPU.
- Unplugging the network cable to the respective F-CPU
- For an S7-1500 F Virtual Controller, a unique connection can be achieved by a targeted switch to STOP of the Industrial Edge application.

NOTE

You can perform the downloading of a consistent safety program only in STOP mode.

NOTE

If *STEP 7 Safety* detects an inconsistent safety program during startup of the F-CPU, the F-CPU startup is prevented, provided the F-CPU supports this detection. (see product information for the respective S7-300/400 F-CPU. This is always supported with S7-1200/1500 F-CPUs.) A corresponding diagnostics event is entered in the diagnostics buffer of the F-CPU.

If the F-CPU does not support this detection function, the F-CPU can go to STOP mode if an inconsistent safety program is executed in activated safety mode.

The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

When downloading the safety program, ensure that the "Consistent download" action is set for the "Safety program" selection in the "Load preview" dialog.

Inconsistent downloading is only possible in disabled safety mode.

Password prompt before downloading to an F-CPU

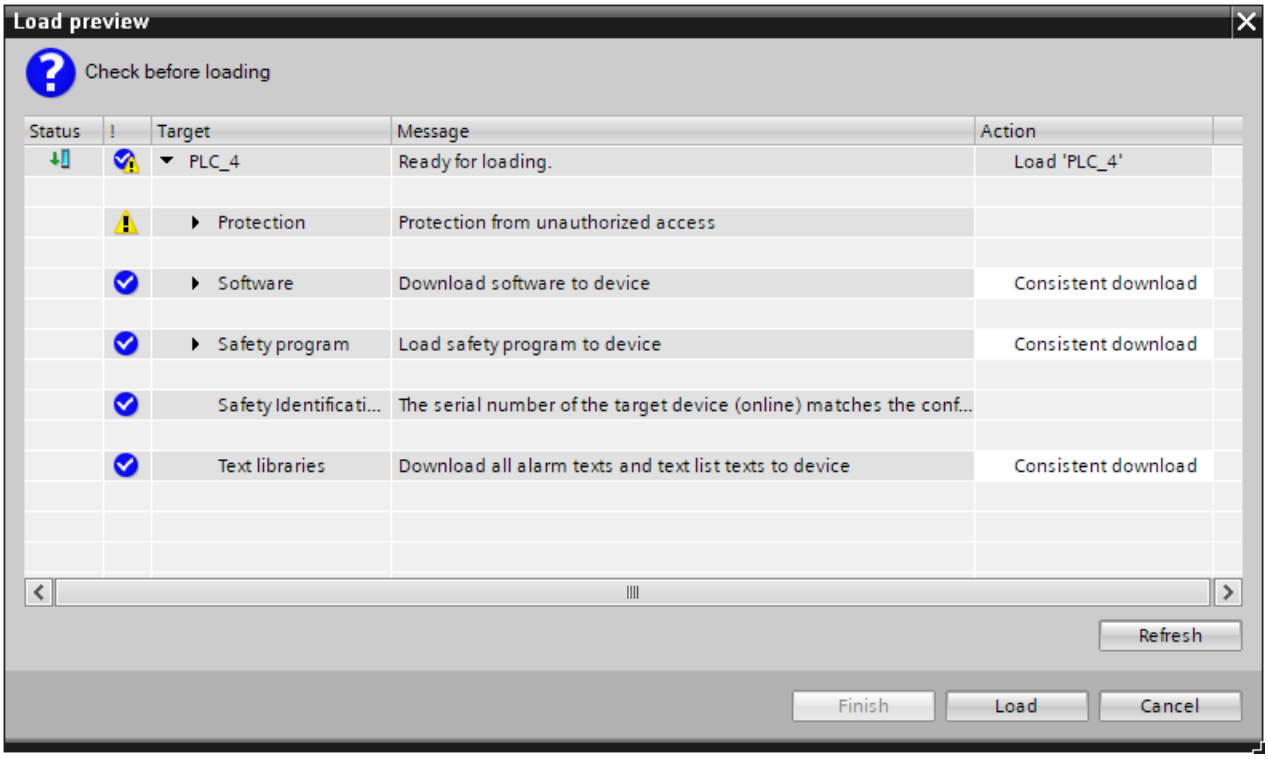
Before the download or, at the latest, during the download (in the "Load preview" dialog), you must gain access permission (Page 103) through legitimation with a user with F-right.

Check of serial number before the download to an S7-1200/1500 F-CPU

Before loading, the serial number of the F-CPU to be loaded is compared with the serial number entered in the *Safety Administration Editor*. If the serial numbers match, you can continue loading. If the serial numbers do not match or no serial number was entered in the *Safety Administration Editor*, you must check whether the serial number of the expected F-CPU is shown. If this is the case, confirm this with a check box in the "Load preview" dialog. The serial number is entered in the *Safety Administration Editor* for subsequent load processes.

"Load preview" dialog

For an F-CPU, the "Load preview" dialog also contains the section "Safety program".



Make the following selection:

- In order to download a consistent safety program, select the "Consistent download" action under target "Safety program".
- (S7-300, S7-400) To download individual F-blocks selectively (Page 357), select the "Download selection" action under the target "Safety program", and then select the required F-blocks. If necessary, you will be prompted to disable safety mode under "Disable safety mode". This setting is only suitable for the online test of individual F-blocks.
- (S7-300, S7-400) In order to download the safety program only, select the "Consistent download" action under target "Safety program" and the "Download selection" action under target" Standard software", and then select only the standard blocks that call the main safety block.
- (S7-300, S7-400) To download no safety program, for example, because you do not know the password of the F-CPU, select the "No action" action under target "Safety program".

For S7-1200/1500 F-CPUs, only the "Consistent download" value is possible as an action in the "Load preview" dialog. You cannot change the selection of program parts to be loaded in this dialog. The selection is made automatically based on the selection in the project tree when starting the loading process.

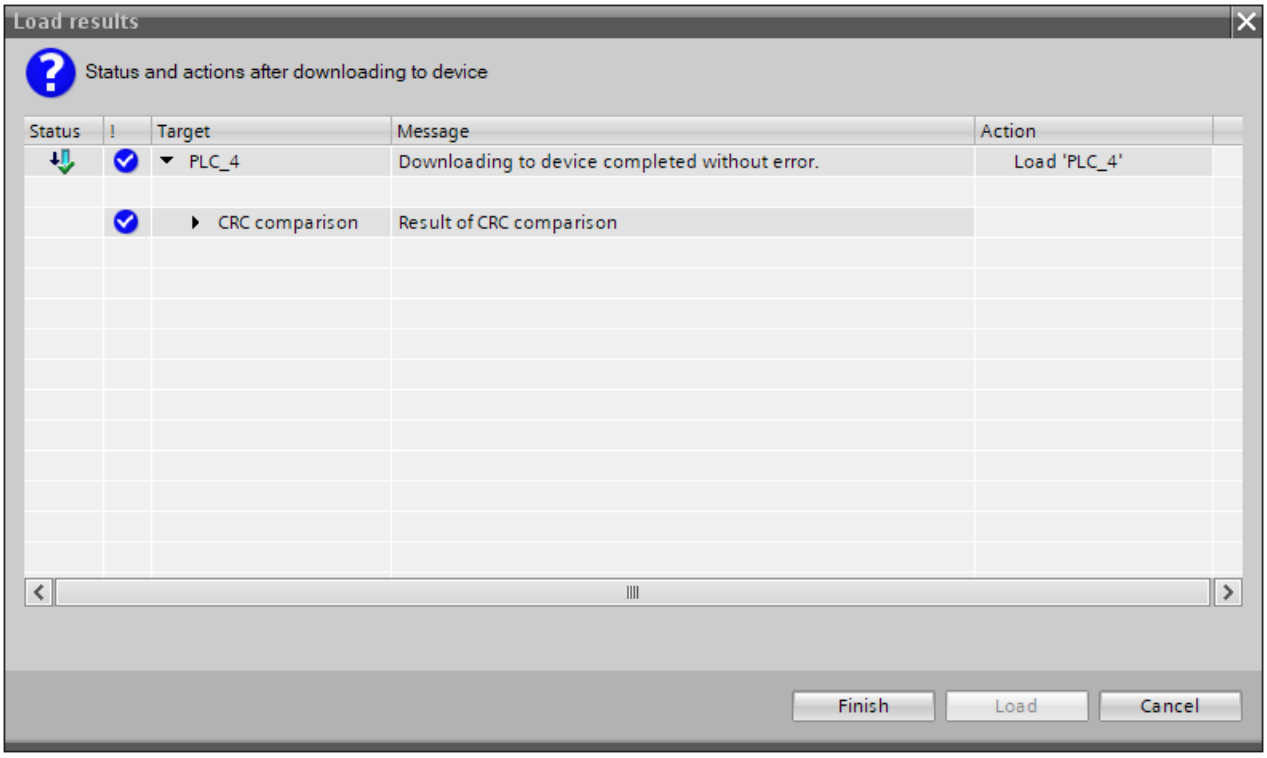
NOTE

If the function "Security Logging" is activated, the event "Download fail-safe program to a CPU" is entered as a log entry during the download.

For more information on the "Security Logging" function and its activation, refer to the help on STEP 7 under "Using Security Logging".

Dialog "Load results"

After downloading into the F-CPU, the dialog "Load results" is opened. This dialog shows you the status and the necessary actions after downloading.



Verify that the "Downloading to device completed without error" message appears in the "Load results" dialog. If not, repeat the download operation.

10.3.1 Downloading project data to an S7-300/400 F-CPU

10.3.1.1 Downloading project data to an S7-300/400 F-CPU with memory card inserted (SIMATIC Micro memory card or flash card)

When you download project data to an S7-300/400 F-CPU with memory card inserted (SIMATIC Micro Memory Card for S7-300 or flash card for S7-400), you must observe the following warning:

⚠ WARNING

If the function test of the safety program is not performed in the destination F-CPU, you must adhere to the following procedure when downloading the safety-related project data to the F-CPU with a **PG/PC** to ensure that no "old" safety-related project data is stored in the F-CPUs:

- Download the safety-related project data to the F-CPU.
- Perform a program identification (i.e. check whether the collective F-signatures match online and offline).
- Perform a memory reset of the F-CPU using the mode switch or via the PG/PC. During the reset, after the work memory has been deleted, the project data is again transferred from the load memory (memory card, SIMATIC Micro Memory Card for S7-300 F-CPUs, flash card for S7-400 F-CPUs) to the work memory. *(S022)*

10.3.1.2 Downloading project data to an S7-400 F-CPU without flash card inserted

When you download project data to an S7-400 F-CPU without flash card inserted, you must observe the following warning:

⚠ WARNING

If the function test of the safety program is not performed in the destination F-CPU, you must adhere to the following procedure when downloading the safety-related project data to the F-CPU with a **PG/PC** to ensure that the F-CPU does not contain any "old" safety-related project data:

- Perform a memory reset of the F-CPU using the **mode switch** or via the **PG/PC**.
- Download the safety-related project data to the F-CPU.
- Perform a program identification (i.e. check whether the collective F-signatures match online and offline). *(S023)*

10.3.1.3 Downloading individual F-blocks to an S7-300/400 F-CPU

Download F-blocks in disabled safety mode.

You can download F-blocks and standard blocks simultaneously to the F-CPU via the project tree. However, as soon as F-blocks are to be downloaded, a check is carried out to determine whether or not the F-CPU is in STOP mode or disabled safety mode. If not, you have the option of switching to disabled safety mode or placing the F-CPU in STOP mode.

If you want to load individual F-blocks into the F-CPU, e.g. to test changes, make sure that you have not selected the "Program blocks" folder or the F-CPU in the project navigation, but only the blocks to be loaded.

Only then will you be prompted in the "Load preview" dialog to disable Safety mode once you have changed the option from "Consistent download" to "Download selection" and have changed the option "Stop modules" to "No action".

If you fail to observe this prompt, the blocks will be downloaded without disabling safety mode, which can lead to STOP of the F-CPU.

You can also deactivate the safety mode explicitly in the *Safety Administration Editor* before you start the download.

Be aware that the consistency of the safety program in the F-CPU cannot be guaranteed when individual F-blocks are downloaded. For a consistent safety program, always download the entire safety program to the F-CPU.

Rules for downloading individual F-blocks

The following rules apply to downloading of individual F-blocks:

- Downloading is only possible in disabled safety mode or when the F-CPU is in STOP mode.
- F-blocks can only be downloaded to an F-CPU to which a safety program has already been downloaded.

Consequently, you have to download the entire safety program when initially downloading the safety program and after changing the password for the safety program.

NOTE

If you are downloading F-blocks only, the blocks in which the main safety blocks are called (e.g., cyclic interrupt OB 35) are not downloaded. To do so, select the "Selection" option under "Standard software" in the preview dialog, and select the necessary blocks.

NOTE

Downloading of individual F-blocks is only suitable for testing F-blocks. Prior to the transition to productive mode, you must download the safety program consistently to the F-CPU.

10.3.1.4 Downloading project data to a memory card and inserting at an F-CPU S7-300/400

Follow the same procedure as for a standard CPU to load project data of an F-CPU to a memory card (flash card for S7-400, SIMATIC Micro Memory Card for S7-300) or save it to a file folder. You must also observe the following warning:

⚠ WARNING

If the function test of the safety program is not performed in the destination CPU, you must ensure that the correct safety-related project data is on the memory card or in the file folder after loading or saving.

Follow these steps:

1. Always load or save the project data to an empty memory card or file folder. Existing project data must not be overwritten.
2. Label the memory card or file folder uniquely (e.g. with the Collective F-Signature).

You must limit access to the loaded or saved safety-related project data to authorized persons through organizational measures (Page 651). *(S043)*

When inserting a memory card (flash card for S7-400 or SIMATIC Micro Memory Card for S7-300) with project data of an F-CPU, you must observe the following warning:

⚠ WARNING

If the function test of the safety program is not carried out in the destination F-CPU, you must ensure through program identification (Page 384) or other suitable measures (for example, by checking the label of the memory card or removable data storage medium) that the inserted memory card or removable data storage medium contains the correct safety-related project data. *(S025)*

When you insert a memory card (flash card for S7-400 or SIMATIC Micro Memory Card for S7-300) into S7-300/400 F-CPUs, you must observe the following warning:

⚠ WARNING

If the function test of the safety program is not performed in the destination F-CPU, you must adhere to the following procedure when inserting the memory card to ensure that no "old" safety-related project data is stored in the F-CPUs:

- Switch off the power to the F-CPU. For F-CPUs with battery backup (e.g. CPU 416F-2), remove any battery. (To ensure that the F-CPU is de-energized, wait for the buffer time of the power supply you are using or, if this is not known, remove the F-CPU.)
- Remove the memory card with the old safety-related project data from the F-CPU.
- Insert the memory card with the new safety-related project data into the F-CPU.
- Switch on the F-CPU again and reinsert the battery, if it was removed.

(S026)

10.3.2 Downloading project data into an F-CPU S7-1200

10.3.2.1 Downloading project data to an S7-1200 F-CPU without program card inserted

When you download project data to an S7-1200 F-CPU without program card inserted, you must observe the following warning:

⚠ WARNING

If the function test of the safety program is not carried out in the destination F-CPU, you must adhere to the following procedure when downloading the safety-related project data to the F-CPU with a **PG/PC** to ensure that the F-CPU does not contain any "old" safety-related project data:

- Download the safety-related project data to the F-CPU.
- Perform a program identification (Page 384). *(S042)*

10.3.2.2 Downloading project data to an S7-1200 F-CPU with program card inserted

When you download project data to an S7-1200 F-CPU G2 with program card inserted, you must observe the following warning:

⚠ WARNING

You must observe the following procedure to ensure that there is no "old" safety-related project data in the internal load memory of the F-CPU when you insert a program card into an S7-1200 F-CPU G2:

1. Check whether the STOP/RUN LED (yellow) and the maintenance LED flash for 3 seconds during startup **when the card slot is empty.** If this is the case, the internal load memory of the F-CPU has already been deleted (for example, when the F-CPU has already been operated with a program card as external load memory), step 3 is skipped and you can skip step 5.
2. Insert the program card into the F-CPU.

   The F-CPU then performs an automatic restart, regardless of whether it is in RUN or STOP.
3. If the internal load memory contains project data, **the STOP/RUN LED (yellow) and the maintenance LED must flash after the restart**. In this case, the internal load memory of the F-CPU is deleted (therefore no longer contains any "old" safety-related project data) and step 5 can be skipped.

   Use one of the following variants to evaluate the program card:

   – Turn the F-CPU off and back on.
   – Remove and insert the program card.
   – Execute the "Memory reset" (MRES) function.

   The F-CPU then performs a restart.
4. After the restart, the program card is evaluated. The F-CPU then enters the startup mode (RUN or STOP) that was set up for the F-CPU (variant 1 of step 3) or remains in STOP (variant 2 or 3 of step 3 or if step 3 is skipped because the internal load memory was empty).
5. Perform a program identification. *(S106)*

When you download project data to an S7-1200 F-CPU G1 with program card inserted, you must observe the following warning:

⚠ WARNING

You must observe the following procedure to ensure that there is no "old" safety-related project data in the internal load memory of the F-CPU when you insert a program card into an S7-1200 G1 F-CPU:

1. Check to see whether **the STOP/RUN LED (yellow) and the maintenance LED flash during startup for 3 seconds on an F-CPU without memory card**.
   If this is the case, the internal load memory of the F-CPU has already been deleted (for example, when the F-CPU has already been operated with a program card as external load memory) and you can skip steps 3 and 5.
2. Insert the program card into the F-CPU.
   If the F-CPU is in RUN, it will change to STOP. The maintenance LED on the F-CPU is flashing to indicate that the program card has to be evaluated or that the internal load memory has to be deleted.
3. Use one of the following methods to delete the internal load memory:
   – Turn the F-CPU off and back on.
   – Switch the F-CPU from STOP to RUN.
   – Execute the "Memory reset" (MRES) function.

   After restart of the F-CPU and deletion of the internal load memory, **the STOP/RUN LED (yellow) and the maintenance LED must be flashing**. In this case, the internal load memory of the F-CPU is deleted (no longer contains any "old" safety-related project data) and step 5 can be skipped.
4. Use one of the following methods to trigger evaluation of the program card on the F-CPU:
   – Turn the F-CPU off and back on.
   – Switch the F-CPU from STOP to RUN.
   – Execute the "Memory reset" (MRES) function.

   The F-CPU restarts and evaluates the program card.
   The F-CPU then enters the startup mode (RUN or STOP) that has been set up for the F-CPU. *(S061)*
5. Perform a program identification.

For an S7-1200 F-CPU without inserted SIMATIC Memory Card and deleted internal load memory, the status LEDs have the status described in the table below.

Description	STOP/RUN Yellow/Green	ERROR Red	MAINT Yellow
Internal load memory deleted and SIMATIC Memory Card **not** inserted.	Flashing (yellow) (for 3 seconds during startup)	Off	Flashing (for 3 seconds during startup)

When you download project data to an S7-1200 F-CPU with program card inserted, you must observe the following warning:

⚠ WARNING

If you use a programming device/PC to download F-blocks to an S7-1200 F-CPU with inserted program card (external load memory), you must ensure that the transfer takes place to the external load memory. This can be accomplished by the following measure:

- Insert a program card whose memory size is different from the size of the internal load memory. Check in the project tree under "Online & Diagnostics > Diagnostics > Memory" if the memory size displayed for the load memory matches the memory size of the program card. *(S058)*

⚠ WARNING

If the function test of the safety program is not carried out in the destination F-CPU, you must adhere to the following procedure when downloading the safety-related project data to the F-CPU with a **PG/PC** to ensure that the F-CPU does not contain any "old" safety-related project data:

- Download the safety-related project data to the F-CPU.
- Perform a program identification (Page 384). *(S042)*

10.3.2.3 Loading project data of an F-CPU S7-1200 with a transfer card into the internal load memory

When you insert a transfer card into an S7-1200 F-CPU G2, you must observe the following warning:

⚠ WARNING

You must observe the following procedure to ensure that there is no "old" safety-related project data in the internal load memory when you copy the safety-related project data to an S7-1200 F-CPU G2 using a transfer card:

1. Check whether the STOP/RUN LED (yellow) and the maintenance LED flash for 3 seconds during startup **when the card slot is empty.** If this is the case, the internal load memory of the F-CPU has already been deleted, step 3 is skipped and you can skip step 7.
2. Insert the transfer card into the F-CPU.

   The F-CPU then performs an automatic restart, regardless of whether it is in RUN or STOP.
3. If the internal load memory contains project data, **the STOP/RUN LED (yellow) and the maintenance LED must flash after the restart**. In this case, the internal load memory of the F-CPU is deleted (therefore no longer contains any "old" safety-related project data) and step 7 can be skipped.

   Use one of the following variants to evaluate the transfer card (transfer from the transfer card to the internal load memory):

   – Turn the F-CPU off and back on.
   – Remove and insert the transfer card.
   – Execute the "Memory reset" (MRES) function.

   The F-CPU then performs a restart.
4. After restart of the F-CPU and evaluation of the SIMATIC Memory Card, the F-CPU copies the project data to the internal load memory of the F-CPU. Once the copy process is complete, the maintenance LED on the F-CPU is flashing to indicate that you can remove the transfer card.
5. Remove the transfer card from the F-CPU. The F-CPU then performs an automatic restart.
6. The F-CPU remains in STOP after the restart.
7. Perform a program identification. *(S105)*

When you insert a transfer card into an S7-1200 F-CPU G1, you must observe the following warning:

⚠ WARNING

You must observe the following procedure to ensure that there is no "old" safety-related project data in the internal load memory when you copy the safety-related project data to an S7-1200 F-CPU G1 using a transfer card:

1. Check whether **the STOP/RUN LED (yellow) and the maintenance LED flash during startup for 3 seconds on an F-CPU without memory card**.
   If this is the case, the internal load memory of the F-CPU has already been deleted and you can skip steps 3 and 6.
2. Insert the transfer card into the F-CPU.

   If the F-CPU is in RUN, it will change to STOP. The maintenance LED on the F-CPU is flashing to indicate that the transfer card has to be evaluated or that the internal load memory has to be deleted.
3. Use one of the following methods to delete the internal load memory:
   - Turn the F-CPU off and back on.
   - Switch the F-CPU from STOP to RUN.
   - Execute the "Memory reset" (MRES) function.

   After restart of the F-CPU and deletion of the internal load memory, **the STOP/RUN LED (yellow) and the maintenance LED must be flashing**. In this case, the internal load memory of the F-CPU is deleted (no longer contains any "old" safety-related project data) and step 6 can be skipped.
4. Use one of the following methods to evaluate the transfer card (transfer from the transfer card to the internal load memory):
   - Turn the F-CPU off and back on.
   - Switch the F-CPU from STOP to RUN.
   - Execute the "Memory reset" (MRES) function.

   After restart of the F-CPU and evaluation of the SIMATIC Memory Card, the F-CPU copies the project data to the internal load memory of the F-CPU. Once the copy process is complete, the maintenance LED on the F-CPU is flashing to indicate that you can remove the transfer card.
5. Remove the transfer card from the F-CPU.
6. Perform a program identification.
7. Use one of the following methods to initiate evaluation of the internal load memory:
   - Turn the F-CPU off and back on.
   - Switch the F-CPU from STOP to RUN.
   - Execute the "Memory reset" (MRES) function.

   The F-CPU then enters the startup mode (RUN or STOP) that has been set up for the F-CPU. *(S059)*

For an S7-1200 F-CPU without inserted SIMATIC Memory Card and deleted internal load memory, the status LEDs have the status described in the table below.

Description	STOP/RUN Yellow/Green	ERROR Red	MAINT Yellow
Internal load memory deleted and SIMATIC Memory Card **not** inserted.	Flashing (yellow) (for 3 seconds during startup)	Off	Flashing (for 3 seconds during startup)

When you want to update the project data on an S7-1200 F-CPU using a transfer card, you must observe the following warning:

WARNING

If you run an update of the safety-related project data on an S7-1200 F-CPU using a transfer card, you must ensure that the transfer to the internal load memory took place correctly by means of a subsequent program identification (Page 384). *(S060)*

10.3.2.4 Downloading project data of an S7-1200 F-CPU from the internal load memory to an empty SIMATIC Memory Card

When you download project data from the internal load memory of an S7-1200 F-CPU G2 to an empty SIMATIC Memory Card, you must observe the following warning:

⚠ WARNING

To ensure that the safety program is downloaded from the internal load memory of the F-CPU to the SIMATIC Memory Card when plugging an empty SIMATIC Memory Card into an S7-1200 F-CPU G2 and that the internal load memory of the F-CPU is deleted afterward, you must observe the following procedure:

1. Make sure that you are using an empty SIMATIC Memory Card, for example, by checking in the Windows Explorer that the "SIMATIC.S7S" folder and the "S7_JOB.S7S" file are deleted.
2. Insert the empty SIMATIC Memory Card into the F-CPU.

   The F-CPU then performs an automatic restart, regardless of whether it is in RUN or STOP.
3. If copying the project data from the internal to the external load memory is permitted, this will be performed automatically after the restart. The internal load memory is then deleted (therefore no longer contains any "old" safety-related project data) and **the STOP/RUN LED (yellow) and the maintenance LED must flash**. The SIMATIC Memory Card is now a program card. If copying the project data is not permitted, the error LED flashes and the project data remains in the internal load memory.
4. Use one of the following variants to evaluate the program card:
   - Turn the F-CPU off and back on.
   - Remove and insert the program card.
   - Execute the "Memory reset" (MRES) function.

   The F-CPU restarts and evaluates the program card.

   The F-CPU then enters the startup mode (RUN or STOP) that has been set up for the F-CPU (variant 1) or remains in STOP (variant 2 or 3). *(S107)*

When you download project data from the internal load memory of an S7-1200 F-CPU G1 to an empty SIMATIC Memory Card, you must observe the following warning:

⚠ WARNING

To ensure that the safety-related project data is transferred from the internal load memory of the F-CPU to the SIMATIC Memory Card when plugging an empty SIMATIC Memory Card into an S7-1200 F-CPU G1 and that the internal load memory of the F-CPU is deleted afterward, you must observe the following procedure:

1. Make sure that you are using an empty SIMATIC Memory Card, for example, by checking in the Windows Explorer that the "SIMATIC.S7S" folder and the "S7_JOB.S7S" file are deleted.
2. Insert the empty SIMATIC Memory Card into the F-CPU.

   If the F-CPU is in RUN, it will change to STOP. The maintenance LED on the F-CPU is flashing to indicate that the program can be copied from the internal load memory to the SIMATIC Memory Card and that the internal load memory is deleted afterward.
3. Use one of these methods to trigger copying from the internal load memory to the SIMATIC Memory Card and subsequent deletion of the internal load memory:
   - Turn the F-CPU off and back on.
   - Switch the F-CPU from STOP to RUN.
   - Execute the "Memory reset" (MRES) function.

   After restart of the F-CPU and copying of the project data from the internal load memory to the SIMATIC Memory Card and subsequent deletion of the internal load memory, **the STOP/RUN LED (yellow) and the maintenance LED must flash**. In this case, the internal load memory of the F-CPU is deleted and no longer contains any safety-related project data. The SIMATIC Memory Card is now a program card.
4. Use one of the following methods to trigger evaluation of the program card on the F-CPU:
   - Turn the F-CPU off and back on.
   - Switch the F-CPU from STOP to RUN.
   - Execute the "Memory reset" (MRES) function.

   The F-CPU restarts and evaluates the program card.

   The F-CPU then enters the startup mode (RUN or STOP) that has been set up for the F-CPU. *(S057)*

NOTE

Also observe the setting "Disable copying from internal load memory to external load memory" in the hardware configuration of your F-CPU.

10.3.2.5 Downloading project data to a SIMATIC Memory Card and inserting it at an F-CPU S7-1200

When loading project data of an F-CPU to a SIMATIC Memory Card or saving it to a file folder, follow the same procedure as for a standard CPU. You must also observe the following warning:

⚠ WARNING

If the function test of the safety program is not performed in the destination CPU, you must ensure that the correct safety-related project data is on the memory card or in the file folder after loading or saving.

Follow these steps:

1. Always load or save the project data to an empty memory card or file folder. Existing project data must not be overwritten.
2. Label the memory card or file folder uniquely (e.g. with the Collective F-Signature).

You must limit access to the loaded or saved safety-related project data to authorized persons through organizational measures (Page 651). *(S043)*

When inserting a SIMATIC Memory Card with project data from an F-CPU, you must observe the following warning:

⚠ WARNING

If the function test of the safety program is not carried out in the destination F-CPU, you must ensure through program identification (Page 384) or other suitable measures (for example, by checking the label of the memory card or removable data storage medium) that the inserted memory card or removable data storage medium contains the correct safety-related project data. *(S025)*

10.3.3 Downloading project data to an S7-1500 F-CPU

10.3.3.1 Downloading project data to an S7-1500 F-CPU

When you download project data to an S7-1500 F-CPU, you must observe the following warning:

⚠ WARNING

If the function test of the safety program is not carried out in the destination F-CPU, you must adhere to the following procedure when downloading the safety-related project data to the F-CPU with a **PG/PC** to ensure that the F-CPU does not contain any "old" safety-related project data:

- Download the safety-related project data to the F-CPU.
- Perform a program identification (Page 384). *(S042)*

10.3.3.2 Downloading project data into a redundant S7-1500HF system

With a redundant S7-1500HF system, a consistent safety program can only be downloaded in STOP mode of the redundant S7-1500HF system.

To do this, you must heed one of the following warnings:

- Loading the redundant S7-1500HF system in STOP.

⚠ WARNING

If the function test of the safety program is not performed in the redundant destination system S7-1500HF, you must adhere to the following procedure when downloading the safety-related project data to the HF-CPUs with a PG/PC. This way you ensure that no "old" safety-related project data is stored in the HF-CPUs:

1. Switch the two HF-CPUs to STOP.
2. If you did not program restart/startup protection in your previous safety program (Programming startup protection (Page 156)), you must format the SIMATIC Memory Card or delete the program, for example, via the display for both HF-CPUs.
3. Download the safety-related project data to one of the two HF-CPUs.
4. Switch the HF-CPU, to which you have downloaded the new safety-related project data, to RUN mode.
5. Switch the second HF-CPU to RUN mode.
6. As soon as the RUN-Redundant mode is reached, perform a program identification on one of the two HF-CPUs (i.e. check whether the Collective F-Signatures match online and offline).

   If you have programmed restart/startup protection in your new safety program, you may only acknowledge this after successful program identification.*(S090)*

- Loading the redundant S7-1500HF system with reduced STOP time.

⚠ WARNING

If the function test of the safety program is not performed in the redundant destination system S7-1500HF, you must adhere to the following procedure when downloading the safety-related project data to the HF-CPUs with a PG/PC. This way you ensure that no "old" safety-related project data is stored in the HF-CPUs:

1. Switch one of the two HF-CPUs to STOP.
2. If you did not program restart/startup protection in your previous safety program (Programming startup protection (Page 156)), you must format the SIMATIC Memory Card or delete the program, for example, via the display for the HF-CPU in STOP mode.
3. Download the safety-related project data to the HF-CPU, which is in STOP mode (to the backup CPU with the context menu command "Load into backup CPU").
4. Switch the HF-CPU that is still in RUN mode to STOP mode.
5. If you did not program restart/startup protection in your previous safety program, you must format the SIMATIC Memory Card or delete the program, for example, via the display for the HF-CPU recently switched in STOP mode.
6. Switch the HF-CPU with the newly downloaded safety-related project data to RUN mode.
7. Switch the second HF-CPU to RUN mode.
8. As soon as the RUN-Redundant mode is reached, perform a program identification on one of the two HF-CPUs (i.e. check whether the Collective F-Signatures match online and offline).

   If you have programmed restart/startup protection in your new safety program, you may only acknowledge this after successful program identification. *(S091)*

10.3.3.3 Downloading project data to an S7-1500 F Software Controller

When you download project data to an S7-1500 F Software Controller, you must observe the following warnings:

⚠ WARNING

If the function test of the safety program is not carried out in the destination F-CPU, you must adhere to the following procedure when downloading the safety-related project data to the F-CPU with a **PG/PC** to ensure that the F-CPU does not contain any "old" safety-related project data:

- Download the safety-related project data to the F-CPU.
- Perform a program identification (Page 384). *(S042)*

The following warning applies to an S7-1500 F Software Controller with firmware ≤ V21.9.x on an IPC 627E, IPC 647E, IPC 677E, or IPC 847E:

⚠ WARNING

When you download the safety program, you must perform a POWER OFF/POWER ON afterwards before you run a program identification for acceptance. *(S097)*

The following warning applies to an S7-1500 F Software Controller with firmware < V30.1

⚠ WARNING

For safety reasons, the password of an S7-1500 F Software Controller is stored in a separate memory in addition to the load memory.

Unlike the load memory, this separate memory is not deleted. This means the previous passwords are once again active after deleting the S7-1500 F Software Controller followed by a start-up.

For this reason, note the following:

- The S7-1500 F Software Controller is deleted in case of the following download scenarios of the PC station:
  - Downloading a PC station with revised interface assignment.
  - Downloading a PC station with revised storage location for retentive data.
- We recommend that you do not set up F-access protection until after commissioning. If you still have to change the interface assignment of the PC station or the storage location for retentive data, you do not have to enter the F-password during the subsequent mandatory download of the S7-1500 F Software Controller.
- We recommend that you remove the F-access protection from an S7-1500 F Software Controller that is no longer in use. If you use the S7-1500 F Software Controller later and don't remember the password, the F-password can only be removed by performing an uninstall/install/repair, installing an image file or importing a configuration file. *(S076)*

See also

Downloading project data (Page 350)
Software Controller (https://support.industry.siemens.com/cs/ww/en/view/109249299)

10.3.3.4 Downloading project data to a SIMATIC Memory Card or removable storage device and inserting it into an F-CPU S7-1500

When loading project data of an (H)F-CPU to a SIMATIC Memory Card or saving it to a file folder, follow the same procedure as for a standard CPU. You must also observe the following warning:

⚠ WARNING

If the function test of the safety program is not performed in the destination CPU, you must ensure that the correct safety-related project data is on the memory card or in the file folder after loading or saving.

Follow these steps:

1. Always load or save the project data to an empty memory card or file folder. Existing project data must not be overwritten.
2. Label the memory card or file folder uniquely (e.g. with the Collective F-Signature).

You must limit access to the loaded or saved safety-related project data to authorized persons through organizational measures (Page 651). *(S043)*

When inserting a SIMATIC Memory Card or an S7-1500 F Software Controller removable disk with project data of an (H)F-CPU, you must observe the following warning:

⚠ WARNING

If the function test of the safety program is not carried out in the destination F-CPU, you must ensure through program identification (Page 384) or other suitable measures (for example, by checking the label of the memory card or removable data storage medium) that the inserted memory card or removable data storage medium contains the correct safety-related project data. *(S025)*

Inserting SIMATIC Memory Cards in a redundant S7-1500HF system

⚠ WARNING

When inserting **two** new SIMATIC Memory Cards with new safety-related project data for a redundant S7-1500HF system, proceed as follows:

1. Switch the two HF-CPUs to STOP.
2. Replace the SIMATIC Memory Cards of both HF-CPUs with
   - Two SIMATIC Memory Cards, each with the new safety-related project data, or
   - One SIMATIC Memory Card with the new safety-related project data and one empty SIMATIC Memory Card

Procedure for ensuring correct safety-related project data through program identification:

1. Check on both HF CPUs in STOP mode, e.g. via the display, whether the correct safety-related project data is located on the SIMATIC Memory Card and that the second SIMATIC Memory Card is empty.
2. After a successful check, switch the HF-CPUs to RUN mode. If the new safety-related project data is only available on one SIMATIC Memory Card, you must first switch the HF-CPU with the new safety-related project data to RUN mode and then the HF-CPU with the empty SIMATIC Memory Card.

Procedure when it is ensured with clear identifiers that the correct safety-related project data is stored on the SIMATIC Memory Cards:

1. Switch the F-CPUs to RUN mode. If the new safety-related project data is only available on one SIMATIC Memory Card, you must first switch the HF-CPU with the new safety-related project data to RUN mode and then the HF-CPU with the empty SIMATIC Memory Card. *(S092)*

⚠ WARNING

When inserting **one** new SIMATIC Memory Card with new safety-related project data for a redundant S7-1500HF system, proceed as follows:

1. Switch the two HF-CPUs to STOP.
2. Remove the SIMATIC Memory Card on one of the two HF-CPUs.
3. Format the SIMATIC Memory Card or delete the program, for example via the display, for the HF-CPU that you did not remove.
4. Insert the SIMATIC memory card with the new safety-related project data into the HF-CPU without SIMATIC Memory Card.
5. If you cannot ensure by other suitable measures (e.g. by clear labeling of the SIMATIC Memory Card) that the correct safety-related project data is located on the SIMATIC Memory Card, you have to carry out a program identification on the HF-CPU with the new safety-related project data.
6. Switch the HF-CPU with the new safety-related project data to RUN mode.
7. Switch the second HF-CPU to RUN mode. *(S093)*

10.3.4 Restoring a backup of the safety program to an F-CPU

You have the option of backing up an F-CPU in the same way as a standard CPU and then restoring it. You can find information on backing up a CPU in the *help on STEP 7* under "Creating a backup of an S7-CPU". Saving an F-CPU S7-400 is only possible with an inserted flash card.

Note the following warnings when restoring the software and hardware configuration of an F-CPU:

⚠ WARNING

Once you have restored a backup of an F-CPU, you must perform a program identification (Page 384).

In a redundant S7-1500HF system, you must perform the program identification (Page 384) in RUN-Redundant mode. *(S055)*

NOTE

We recommend that you use the Collective F-Signature that is included in the name of the backup file for program identification. You must not change the Collective F-Signature in the name in this case.

⚠ WARNING

(S7-1200 G1, S7-1200 G2 as of FW 4.1, S7-1500) If several F-CPUs with activated web server can be accessed from a PG/PC, you must ensure through additional measures that the safety program is restored in the correct F-CPU when safety-relevant project data is restored via the web server. Use unique identifiers for this, such as the Collective F-Signature, serial number, station name or plant designation. *(S065)*

NOTE

When restoring, you must enter the previous password for the F-CPU.

10.3.5 Specifics on creating and installing an image file of an S7-1500 F Software Controller

Creating an image file

⚠ WARNING

When creating image files (for example, of an image) with safety-related project data, you must comply with the following points:

- You must take organizational measures (Page 651) to limit access to an S7-1500 F Software Controller to persons who are authorized to create image files.
- Before creating an image file, you must ensure through program identification (Page 384) that the correct safety-related project data is located in the S7-1500 F Software Controller.
- Image files with safety-related project data must be created on an empty data storage medium (deleted or formatted). Existing image files on a data storage medium or in a file system must be explicitly deleted; you are not permitted to overwrite an existing image file.
- After creating the image file, remove the data storage medium containing the image file or remove the image file from the file system.
- Clearly label the data storage medium or image file (e.g. with the Collective F-Signature). *(S073)*

Installing an image file

⚠ WARNING

When installing an image file (for example, of an image) with safety-related project data, you must comply with the following points:

- You must take organizational measures (Page 651) to limit access to an S7-1500 F Software Controller to persons who are authorized to install image files.
- When installing an image file via LAN, remote access or comparable access, you must ensure access protection (for example, using Windows administrator rights (ADMIN) or the Linux root user). Note, however, that only authorized persons are set up as users.
- To ensure that the image file is written to the correct S7-1500 F Software Controller, you must make sure when installing that the safety-related project data is installed in the correct device (for example, by comparing the serial number or the hardware identification of the device determined by the operating system).
- You must ensure that the image file contains the correct safety-related project data, for example, by checking the labeling of the data storage medium or image file.
- Remove the image file once you have installed it in the S7-1500 F Software Controller. Also remove any copies.

The following applies to firmware version as of V40.0 of the S7-1500 F Software Controller:

- After installing the image file, you must ensure through program identification (Page 384) that the correct safety-related project data is in the S7-1500 F Software Controller, for example with the display or with the command line command "GetCollectiveFSignature".

The following applies to firmware versions V30.0 and higher and < V40.0 of the S7-1500 F Software Controller:

- After installing the image file, you must ensure that the correct safety-related project data is in the S7-1500 F Software Controller through a program identification, for example, by reading out the Collective F-Signature with the command line command "GetCollectiveFSignature" or the display.
- Then you must enter and confirm the expected Collective F-Signature with the command line command "ConfirmCollectiveFSignature".

The following applies to firmware versions < V30.0 of the S7-1500 F Software Controller:

If the safety-related project data in the image file and the safety-related project data on the S7-1500 F Software Controller is different, the installed Safety program would not start up. In this case, you must download the project data to the F-CPU once again. For example, with TIA Portal. Therefore, you should always keep up-to-date backups of the image file.

In the same way that you can start up a Safety program from another CFast card, you can also install and run the image file created by another device or data storage medium.

(S074)

10.3.6 Loading project data (including safety-related project data) from an F-CPU into a PG/PC (S7-1500/S7-1200 G2 V4.1)

The "Upload from device (software)" or "Upload device as new station (hardware and software)" function is only possible for S7-1500/S7-1200 G2 V4.1 F-CPUs if the "Enable consistent upload from the F-CPU" option is activated for the F-CPU in the *Safety Administration Editor* and the project data is loaded to the F-CPU afterwards.

To load the project data (including safety-related project data) to a programming device or PC, proceed as in the standard scenario.

If "Delayed transfer" (internal value 0x7C6) was set for the F-BaseID in the original hardware configuration in the F-CPU, the actually assigned value is read when loading from the F-CPU and the option "Delayed transfer" is activated at the same time in the hardware configuration of the F-CPU. For connected F-I/Os of PROFIsafe address type 3, the setting "Delayed transfer" (internal value 0x7C6) remains in the hardware configuration.

When the data is downloaded to the F-CPU again, the actually assigned value for F-BaseID is thus retained and is transferred again to the F-I/O of PROFIsafe address type 3. If this behavior is not desired, you must deactivate the "Delayed transfer" option and generate the F-BaseID again, if necessary. The value for F-BaseID set in the hardware configuration then takes effect directly in the F-CPU and F-I/O of PROFIsafe address type 3 after a new compile and download of hardware and software.

If multiple F-CPUs can be reached over a network (e.g. Industrial Ethernet) by the programming device / PC, you have to ensure that the project data is uploaded from the correct F-CPU. For example, with "Online & diagnostics" > "Online access " > "Flash LED".

After successful loading from the device you can continue working as with a project that was created offline.

⚠ WARNING

If you want to carry out an acceptance with the project data loaded to the programming device / PC or want to carry out changes to the safety-related project data, you must ensure that the safety program is executable before loading.

This is fulfilled when loading from an F-CPU if the F-CPU is in RUN mode and in activated safety mode or can be set to RUN before loading. If the F-CPU remains in STOP, you are not allowed to perform acceptance or changes with the safety-related project data.

When loading from a memory card, it must have been in a suitable S7-1500/S7-1200 G2 V4.1 CPU beforehand. The same conditions apply as for loading from an F-CPU. *(S080)*

You can upload individual F-blocks into a PG/PC irrespective of the "Enable consistent upload from the F-CPU" option.

You cannot upload individual know-how protected F-blocks to a programming device / PC.

As of STEP 7 Safety V18, loading the project data from a memory card is possible as a new station (hardware and software) and device (software).

See also

"Settings" area of the Safety Administration Editor (Page 91)

10.3.7 Loading project data (including safety-related project data) from a memory card into a PG/PC (S7-1500/S7-1200 G2 V4.1)

The "Upload from device (software)" or "Upload device as new station (hardware and software)" function is only possible for S7-1500/S7-1200 G2 V4.1 F-CPUs if the "Enable consistent upload from the F-CPU" option is activated for the F-CPU in the *Safety Administration Editor* and the project data is loaded to the memory card afterwards.

To load the project data (including safety-related project data) to a programming device or PC, proceed as in the standard scenario.

If "Delayed transfer" (internal value 0x7C6) was set for the F-BaseID in the original hardware configuration in the F-CPU, the actually assigned value is read when loading from the F-CPU and the option "Delayed transfer" is activated at the same time in the hardware configuration of the F-CPU. For connected F-I/Os of PROFIsafe address type 3, the setting "Delayed transfer" (internal value 0x7C6) remains in the hardware configuration.

When the data is downloaded to the F-CPU again, the actually assigned value for F-BaseID is thus retained and is transferred again to the F-I/O of PROFIsafe address type 3. If this behavior is not desired, you must deactivate the "Delayed transfer" option and generate the F-BaseID again, if necessary. The value for F-BaseID set in the hardware configuration then takes effect directly in the F-CPU and F-I/O of PROFIsafe address type 3 after a new compile and download of hardware and software.

After successful loading from the memory card, you can continue working as with a project that was created offline.

⚠ WARNING

If you want to carry out an acceptance with the project data loaded to the programming device / PC or want to carry out changes to the safety-related project data, you must ensure that the safety program is executable before loading.

This is fulfilled when loading from an F-CPU if the F-CPU is in RUN mode and in activated safety mode or can be set to RUN before loading. If the F-CPU remains in STOP, you are not allowed to perform acceptance or changes with the safety-related project data.

When loading from a memory card, it must have been in a suitable S7-1500/S7-1200 G2 V4.1 CPU beforehand. The same conditions apply as for loading from an F-CPU. *(S080)*

See also

"Settings" area of the Safety Administration Editor (Page 91)

10.3.8 Loading a PC system configuration file for S7-1500 F software controllers

You can save the entire configuration of your S7-1500 F Software Controller in a configuration file, transport it and load it to a target system. The entire configuration of your S7-1500 F Software Controller is saved from the TIA Portal in a configuration file with the extension *.psc.

Saving and loading of the configuration file is supported as of:

- *STEP 7 Safety V15*
- S7-1500 F Software Controller V2.5

Example

You can find a detailed example on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109759142).

Identification parameters

The identification parameters include:

- File name
- Information in the project and the station that was stored from the TIA Portal in the PSC file in the metadata.

  For example:

  – Project version
  – Plant designation
  – Station comment

Save the identification parameters in a file, if necessary, that you store on the target system.

For evaluating and testing these identification parameters via script, you must store this information directly in the script or save the identification parameters in a separate file, if necessary, that you store on the target system.

10.3.8.1 Creating a configuration file

1. In TIA Portal create a new configuration file with "Project > Memory Card file > New > PC system configuration file (.psc)".

   The configuration file is created in the project tree under "Card Reader/USB memory".
2. Use the Collective F-Signature and, if necessary, the F-BaseID to check in the *Safety Administration Editor* that you have selected the correct project/station.
3. Use your mouse to drag the selected PC station to the configuration file.

   This loads the PC station to the configuration file.

⚠ WARNING

Instead of program identification, you can use a unique name for the PC system configuration file *.psc (**P**C **S**ystem **C**onfiguration) to ensure that the correct safety-related project data is located in the configuration file.

In addition, you have to observe the following when creating a configuration file:

When creating a configuration file with the safety-related project data, an existing file must not be used. You have to create a new file.

You should also delete configuration files with incorrect safety-related project data from the data storage.

You must limit access to the configuration file (*.psc) through organizational measures (Page 651) to persons who are authorized to import and modify the configuration file. *(S081)*

10.3.8.2 Importing the configuration file

You have the following options for importing the configuration file:

- Via the PC Station Panel menu (Import configuration file) **(only FW < V30.0)**
- By means of a script

Import via the PC Station Panel menu (only possible with firmware < V30.0)

Requirement

If you want to start the import of the configuration file via the menu in the PC Station Panel of an S7-150xS(P) F, the executing user must be in the Windows user group "Failsafe Operators".

Procedure

⚠ WARNING

If the import operation was successful, you will receive a positive feedback. If you do not receive a positive message, you must assume that the import operation was not successful and that the old safety-related project data is still present.

When a configuration file with safety-related project data is imported, you must adhere to the following points:

- Use the unique name of the configuration file to check that you have selected the required configuration file.
- To ensure that a configuration file imported via LAN is imported to the correct S7-1500 F Software Controller, use the following measures:
  - Remove the physical connections and possible routings to other S7-1500 F Software Controllers.
  - Use unique computer names and unique user logins or use other identification options. *(S084)*

Import by means of a script

⚠ WARNING

You must check in the script based on the specified identification parameters whether the import of the configuration file is permitted for the respective target system (e.g. by evaluating the F-CPU name, project name or using the plant designation).

In addition, it may be necessary or useful to check the respective instance of the target system. This means a diversity check of your addressing and/or checking the version of the configuration file, for example, only higher versions or the exclusion of specific versions (block list).

Example of checking the respective version for validity:

- The script evaluates the information about the version and only allows configuration files of a higher version, for example.

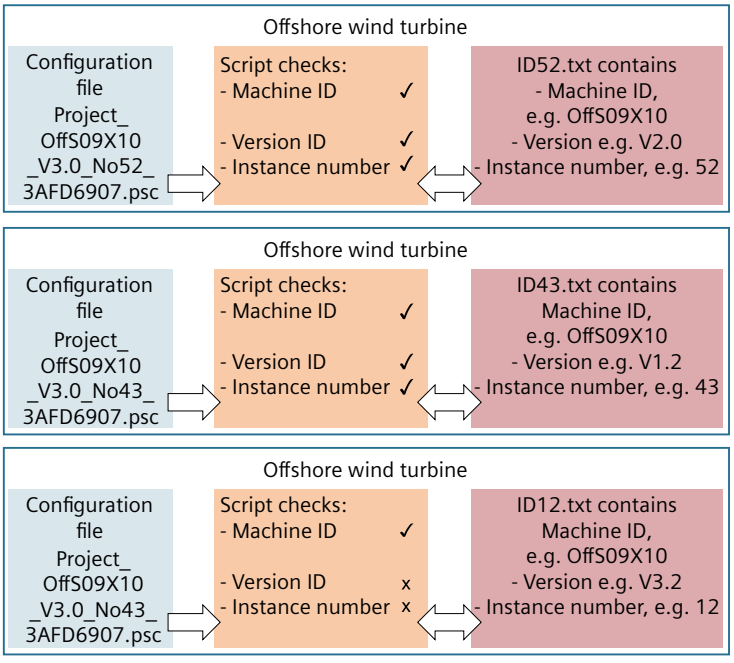
As a machine manufacturer, you must ensure that the script is protected against manipulation (unauthorized change of content or name).

If, as a machine manufacturer, you only make available the configuration files, you have to ensure that no incorrect configuration file is imported during import. You achieve this through technical measures (extended checks in the script) and training of the machine operators. *(S082)*

The script checks for:

- Matching machine ID
- Version ID greater than current one. If so, the new version is written to the txt file.
- Instance number

The figure below contains a systematic overview of checking the configuration file in the script with the help of an identification parameter stored in a separate file (shown in violet in the figure below):



⚠ WARNING

You must determine whether the safety-related project data was successfully imported via the script by evaluating the corresponding return value (0x51A3 or 20899). If the corresponding return value is not returned, the import has failed and the old safety-related project data may still be present.

If the import operation is triggered by a server, feedback about the positive return value must also be given.

For traceability we recommend that you document the import operation in a log file.

The following applies to firmware versions V40.0 and higher of the S7-1500 F Software Controller:

To have the return value 0x51A3 displayed after the import, you must call the import command with extension "/v" for verbose.

After a successful import, you must perform a program identification (for example, manually or via a script).

The following applies to firmware versions V30.0 and higher and < V40.0 of the S7-1500 F Software Controller:

To have the return value 0x51A3 displayed after the import, you must call the import command with extension "/v" for verbose.

After successful import, you must read out the Collective F-signature from the meta file (PSC file) and enter and confirm it with the command line command "ConfirmCollectiveFSignature". This confirmation is required to restart the safety program.

The following applies to firmware versions < V30.0 of the S7-1500 F Software Controller:

To ensure that the return value is not from the previous import, you must reset the return value to 0x3FF ("PCSystem_Control /ImportConfig" without entering a file name) before the import. Then check whether the return value was reset to 0x3FF (enter "PCSystem_Control/GetStatus /ImportConfig" and then enter "echo %errorlevel%". This instruction must deliver the return value 0x3FF).

If the configuration file is imported manually via the Windows command line (via script command), you need to do one of the following:

- Reset the return value to 0x3FF and check it before the import (see above).
  - Carry out the import.
  - Evaluate return value (enter "PCSystem_Control /GetStatus /ImportConfig" and then enter "echo %errorlevel%". This instruction must return the return value 0x51A3 or 20899).
- Carry out the import.
  - Perform a manual program identification, for example, using the display of the F-CPU.

(S083)

NOTE

The positive return value when importing a configuration file via script is "0x51A3" for an S7-1500 F Software Controller, in contrast to the S7-1500 Software Controller, in which case it is "0x0000".

When the file is imported via script, the authorization should be moved to the script. This means that the executing user does not need a higher authorization level, since the script that was made available to the user by the machine manufacturer contains the necessary authorizations (user group "Failsafe Operators").

The rights are assigned via script by assignment of the Windows service to the corresponding user group. This initial installation must be performed beforehand by the Windows administrator on every computer with S7-150xS(P) F. The Windows service can be called by the executing user and the Windows service executes the script.

10.4 Program identification

You use the program identification to determine that the correct safety-related project data has been downloaded to the F-CPU To do so, you compare the Collective F-Signature with an expected value. The expected value can be, for example, the offline Collective F-Signature from the *Safety Administration Editor* or from the safety summary.

For S7-1500 F-CPUs with firmware < V3.1 and S7-1200 G1 F-CPUs with firmware < V4.7, you also compare the "F-admin" right online and offline in the *Safety Administration Editor*.

F-I/Os of PROFIsafe address type 3 (with active F-BaseID or the value "Delayed transfer") can be used for S7-1500 and S7-1200 G2 F-CPUs with firmware V4.1 or higher, or S7-1500 F-Software Controllers with firmware V40.1 or higher, as well as Safety System Version V2.8. The F-BaseID is not part of the Collective F-Signature and may need to be checked separately. In general, this is not necessary because the F-I/Os passivate if inconsistencies occur. Checking the F-BaseID could be useful if, for example, you want to check which value is active online.

For information on the F-BaseID, see also sections "Configuring an F-CPU (Page 41)" and "PROFIsafe addresses for F-I/O of PROFIsafe address type 3 (Page 70)". To check the F-BaseID using the *Safety Administration Editor*, see the section "General" area of the *Safety Administration Editor* (Page 83).

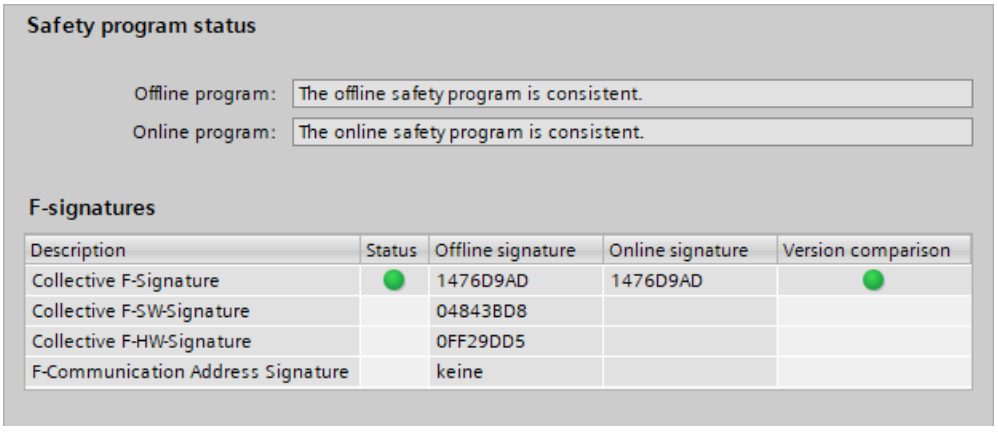
Use organizational measures (Page 651) to ensure that the safety-related project data is not downloaded by any other TIA Portal (on a separate programming device or PC) during the program identification.

With the *Safety Administration Editor*

For a program identification using the *Safety Administration Editor* (Page 80), follow these steps:

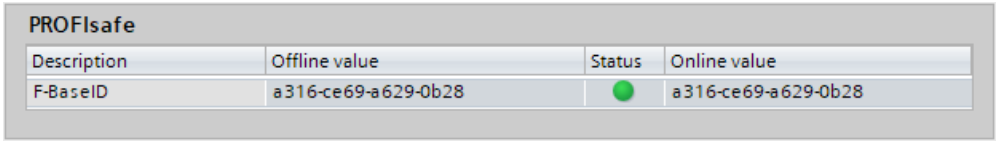
1. Open the *Safety Administration Editor* of the F-CPU you want to check.
2. Connect online with the F-CPU you want to check.
3. Compare the Collective F-Signature displayed online with the expected value in the "General" section.
4. Check whether the offline and online program are consistent (Page 430).

5. Check whether the green symbol is displayed in the column "status" and "Version comparison".



6. For S7-1500 F-CPUs with firmware < V3.1 and S7-1200 G1 F-CPUs with firmware < V4.7: Check in the "Web server F-admins" section whether only authorized users have the "F-admin" right offline and online.
7. If the check of the F-BaseID is required:

   Check in the "General" area of the *Safety Administration Editor* under "PROFIsafe" to see if the F-BaseID is the same offline and online. If it is the same, this is indicated by the green icon. (For information on checking the F-BaseID using the *Safety Administration Editor*, see also section "General" area of the Safety Administration Editor (Page 83).)



With HMI

For a program identification using the HMI, follow these steps:

1. Read the Collective F-Signature of the safety program from the F_PROG_SIG tag of the F-global DB (Page 154) (S7-300, S7-400) or the F_SYSINFO.F_PROG_SIG tag of the F-runtime group information DB (Page 154) (S7-1200, S7-1500).
2. Compare the value of the F_PROG_SIG tag with the expected value.
3. If the check of the F-BaseID is required:

   Compare the value of parameter BASEID_ACTIVE at the F-I/O DB of any F-I/O of PROFIsafe address type 3 of the relevant F-CPU with the expected value of the F-BaseID (e.g. from the safety summary). For information regarding BASEID_ACTIVE, see also section BASEID_ACTIVE (Page 172).

With the display of an S7-1500 F-CPU

For a program identification using the display of an F-CPU, follow these steps:

1. In the display menu, go to "Overview > Fail-safe".
2. Compare the displayed Collective F-Signature with the expected value.
3. If the check of the F-BaseID is required:
   Compare the displayed F-BaseID with the expected value (e.g. from the safety summary).

With the web server of an S7-1200 G1/S7-1200 G2 F-CPU as of V4.1/S7-1500

For a program identification with the web server of an S7-1200 G1/S7-1200 G2 F-CPU as of V4.1/S7-1500, follow these steps:

1. Read the Collective F-Signature on the homepage of the Web server.
2. Compare the displayed Collective F-Signature with the expected value.
3. If the check of the F-BaseID is required:
   Compare the displayed F-BaseID with the expected value (e.g. from the safety summary).

With the application UI of the S7-1500VF

For program identification using the UI of the S7-1500VF application, follow these steps:

1. Open the UI of the S7-1500VF by clicking on the app icon.
2. Compare the displayed Collective F-Signature with the expected value.

10.5 Comparing Safety Programs

Compare safety programs as in standard

You can use the *comparison editor* in *STEP 7* for offline-online or offline-offline comparison of safety programs. The procedure is the same as for standard user programs. The contents of F-blocks are also compared for the comparison of safety programs. As a result, an offline-offline comparison can also be used for an acceptance of changes (Page 433). You enable this comparison by selecting the "Safety" comparison criterion and disabling all other comparison criteria.

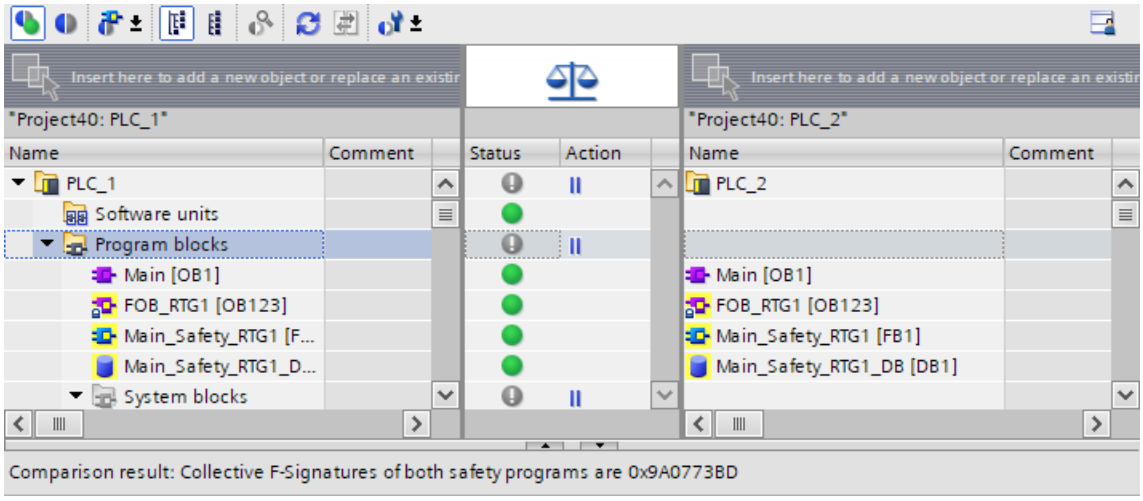
NOTE

You must not use the *Comparison editor* to detect changes offline-online in the safety program/configuration of the F-IOs when accepting changes. Only the offline-offline comparison is suitable for this purpose. To accept changes, proceed as described under Acceptance of Changes (Page 433).

Comparison result for safety programs

The representation of the comparison result corresponds to the representation of *STEP 7*.

Please note:
The status symbol of the "Program blocks" folder merely summarizes the comparison result of all lower-level program blocks, without taking into account the comparison values of program consistency and the collective F-signature.
To establish whether the two Safety programs are the same, you need to click the "Program blocks" folder on the left of the comparison editor. Then the collective F-signatures of both Safety programs are shown under "Comparison result". You also receive information about whether the safety program is consistent.



If you click on an F-block, you can see the respective signatures and interface signatures in addition to the standard information.

NOTE

If you interrupt the connection to the F-CPU during the online/offline comparison, the comparison result will be incorrect.

Comparison filter options

You can use filters in the *comparison editor* to limit the comparison result to the following block groups:

- Compare only F-blocks
- Compare only F-blocks relevant for certification
- Compare only standard blocks

You also have the *STEP 7* filter options "Show only objects with differences" and "Show identical and different objects".

For comparison of safety programs, F-blocks in the "System blocks" folder are also relevant.

Comparison criteria

Make sure that under only the comparison criterion "Safety" is enabled.

Classification of displayed changes

Regardless of whether you carried out an offline/online or offline/offline-comparison, the following changes could account for the indicated changes to the automatically generated F-blocks:

- Changing in the maximum cycle time of the F-runtime group and warning limit F-cycle time of the
  F-runtime group
- Change in F-parameters of the F-CPU
- (S7-1200/1500) Modified Safety system version
- (S7-300, S7-400) Change in the F-runtime group communication, for example, change in the number of a DB for F-runtime group communication
- Change in main safety block, F-FB, F-FC, F-DB
- Change of the hardware configuration for the F-I/O addressed in the safety program
- (S7-1200, S7-1500) Change in the communication with Flexible F-Link
- (S7-1500) Change in the communication with OPC UA Safety

It is possible that a block is displayed as changed, but no changes are displayed in the detailed comparison of the block content. This is not a display problem but means that changes of addresses in the tag table, for example, have an effect on this block. Test this block in case of doubt.

Document the result of the comparison

The comparison result can be printed via "Project > Print" in the menu bar or the "Print" icon in the toolbar or create a PDF file. Select "Print objects/area" "All" and "Properties" "All".

10.6 Creating a safety summary

The documentation of the safety-related project data (safety summary) serves, in addition to other system documentation, as the basis for checking the correctness of the individual components of the system. Correctness is a prerequisite for system acceptance.

The Collective F-Signature in the footer on the pages of the safety summary guarantees unambiguous assignment of the safety summary to a Safety program.

Bear in mind, however, that the F-BaseID does not enter into the Collective F-Signature. If this parameter is relevant to your archiving, you must separately check that the right F-BaseID was printed.

The safety summary can be created in printed or electronic form, for example, as a PDF file.

Procedure for creating the safety summary

To create the safety summary, follow these steps:

1. Open the *Safety Administration Editor* (Page 83) of the F-CPU whose safety summary you want to create.
2. Under "Safety summary" in the "General" section, press the "Create safety summary" button.

   In the displayed dialog, you can make layout settings for the printout and specify the scope of the printout (all/subset), among other things.

   Alternatively, you can select an F-CPU in the project tree and open the dialog using "Print" in the shortcut menu, "Project > Print" in the menu bar or the print button in the toolbar.
3. If you want to print the safety summary on paper, we recommend one of the ISO formats, e.g. "DocuInfo_Simple_A4_Portrait_Small_Footer" or "DocuInfo_ISO_A4_Portrait".

   In these formats, the summary contains page numbers with which you can verify the completeness of the summary.
4. Select the "All" option, if the F-blocks and F-compliant PLC data types (UDT) are to be shown in the printout. This is necessary, for example, to document the program code for the acceptance (see Acceptance of system (Page 413)). Select the "Compact" option to exclude the source code from the printout.
5. Click the "Print" button.

As a result, you receive the safety summary for the F-CPU.

After printing, make sure that all pages were printed. Incomplete printouts (e.g. lines cut off, insufficient toner when printing to paper) must not be used for an acceptance.

Overview of the contents of the safety summary

The information included in the safety summary is summarized below:

- General information on program identification, signatures, software versions, access protection, settings of the Safety program (from the "Settings" work area of the *Safety Administration Editor*), for example safety system version.
- System library elements used in Safety program (from the "Instructions" task card) along with their versions
- Information about the F-runtime groups (e.g. cycle time warning limit of the F-runtime group, maximum cycle time of the F-runtime group)
- List of F-blocks within the "Program blocks" folder (e.g. name, function, associated F-runtime group, signature)
- (S7-1200, S7-1500) List of the know-how protected F-blocks used in the Safety program (e.g., name, signature, used safety system version, used versioned instructions, named value types (NVT) or called F-blocks).
- (S7-1200, S7-1500) List of F-compliant PLC data types (UDT) used in the Safety program, if these exist in the Safety program.
- (S7-1200, S7-1500) List of the definitions of all named-value types (NVT) used in the safety program and all named-value types (NVT) that can be used in the safety program via a relation to another software unit.

- (S7-1200, S7-1500) List of all named-value types (NVT) used in the Safety program.
- (S7-1200, S7-1500) List of all system NVTs used in the safety program.
- List of all global constants used in the Safety program.
- Data from the standard user program that are evaluated in the Safety program
- Block parameters of the safety-related CPU-CPU communication by means of the instructions SENDDP/RCVDP or SENDS7/RCVS7
- (S7-300, S7-400) Absolute addresses and names of the F-shared DB tags that can be accessed from the standard user program.
- Information on the F-CPU used (e.g. firmware version, central F-source address, F-BaseID) Information on the F-I/O used (e.g. firmware version, general and specific parameters)
- Information on the printout (print date, number of pages)
- Parameters of the safety-oriented communication with OPC UA Safety
- Parameters of the safety-oriented communication with Flexible F-Link

Content of the footer of the safety summary

Based on the footer of the safety summary, you can see whether the printed safety-related project data is consistent and whether all pages of the safety summary belong to the same Safety program and the same version (the same Collective F-Signature in the footer of every page means that the safety summary belongs to the Safety program with this Collective F-Signature).

The footer is only added to the source code of the F-blocks if the "All" option was selected when creating the Safety program. If F-blocks are printed by other means, the footer is omitted, and you can no longer easily identify whether the block printout belongs to the current Safety program version.

Safety summary for a migrated project

You can only create the safety summary for a project migrated from *S7 Distributed Safety V5.4 SP5* if the project was compiled with *STEP 7 Safety Advanced* and the new program structure for safety programs (main safety block) was thereby applied. Otherwise, the safety documentation cannot be created and you will receive a corresponding error message.

We recommend that you create a safety summary for your project in *S7 Distributed Safety V5.4 SP5* before the migration.

10.7 Testing the safety program

10.7.1 Overview of Testing the Safety Program

Complete function test or test of changes

After creating a safety program, you must carry out a complete function test in accordance with your automation task.

For changes made to a safety program that has already undergone a complete function test, only the changes and that there is no effect on the parts of the safety program that were not changed need be tested.

Monitoring

Read-only test functions (such as monitoring tags of the safety program) are available for safety programs as in the standard.

Modifying

Read and write test functions (such as controlling tags of the safety program) are only available to a limited extent for safety programs and only in disabled safety mode.

Simulation via *S7-PLCSIM*

You can test the safety program using *S7-PLCSIM*. You use *S7-PLCSIM* in the same way as for standard user programs.

You start the simulation with *S7-PLCSIM* using menu item "Online > Simulation > Start"

Rules for testing

- Forcing of F-I/O inputs and F-I/O outputs is not possible.
- Controlling F-I/O outputs in connection with the function "Enabling F-I/O outputs" is not possible.
- Setting breakpoints in the standard user program will cause errors in the safety program (see also Testing the safety program (Page 397)).
- Changes in the configuration of F-I/O or safety-related CPU-CPU communication can only be tested after the hardware configuration has been saved and downloaded, and after the safety program has been compiled and downloaded to the F-CPU.

See also

Disabling safety mode (Page 393)

10.7.2 Disabled safety mode

The safety program generally runs in the F-CPU in safety mode after a STOP/RUN transition. This means that all fault control measures are activated. The safety program cannot be modified during operation (in RUN mode) in safety mode. You must disable safety mode of the safety program to, for example, modify tags in the safety program in RUN mode. Safety mode remains disabled until the F-CPU is next switched from STOP to RUN mode.

As of Safety system version V2.4, disabled safety mode for S7-1200/1500 CPUs is time-limited.

After the time limit has expired, the F-CPU or the redundant S7-1500HF system automatically goes to STOP. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

You can avoid an unplanned STOP of the F-CPU after the time limit has expired as follows:

- Ending the disabled safety mode through a targeted STOP/RUN transition.
- Resetting the remaining runtime. Resetting can be used as often as needed.

10.7.2.1 Configuring the time limit for disabled safety mode (S7-1200, S7-1500)

Introduction

You can configure the duration of the time limit.

Requirements

- In the *Safety Administration Editor*, the "Safety mode can be disabled" option is activated.
- The minimum Safety system version is set to V2.4.

Procedure for configuring the runtime

To configure the time limit, follow these steps:

1. Open the *Safety Administration Editor* of the F-CPU.
2. Open the "Settings (Page 91)" area in the area navigation.
3. Under "Runtime for the deactivated safety mode", configure the time limit for the safety mode. The default time is 4 hours. You have the option to configure the time limit to a maximum of 8 hours and a minimum of 1 minute.

NOTE

The value of the configured time is not included in the Collective F-Signature.

10.7.2.2 Disabling safety mode

Requirements

- In the *Safety Administration Editor*, the "Safety mode can be disabled" option is activated.
- (S7-1200, S7-1500) The serial number of the F-CPU (Page 83) has been entered in the *Safety Administration Editor*.
- The project is loaded to the F-CPU.
- The F-CPU is in RUN.
- The Safety program runs in safety mode.

Rules for disabling safety mode

⚠ WARNING

When safety mode is disabled, you must take the following into consideration:

- Disabling of safety mode is intended for test purposes, commissioning, etc. While safety mode is disabled, plant security must be ensured by organizational measures (Page 651).

  After the test or commissioning, safety mode must be enabled again. To do this, execute a STOP/RUN transition of the F-CPU.

  In a redundant S7-1500HF system you have to set both HF-CPUs or the redundant system S7-1500HF to STOP before you restart the HF-CPUs.
- A disabling of the safety mode must be displayed.

  Use the MODE tag in the F-shared DB ("F_GLOBDB".MODE) for S7-300/400 F-CPUs or in the F-runtime group information DB (e.g. RTG1SysInfo.F_SYSINFO.MODE) for S7-1200/1500 F-CPUs, which you can evaluate to read the operating mode (1 = Disabled safety mode).
- It must be possible to verify that safety mode has been disabled. A log is required, if possible by recording and, if applicable, archiving alarms to the operator control and monitoring system or, if need be, through organizational measures. In addition, it is recommended that a disabling of the safety mode be indicated on the operator control and monitoring system.
- With safety-related communication, you must also take into account a possible deactivation of Safety mode on the communication partner's side*. You can then no longer assume that the data provided by the communication partner is formed in a safety-related manner.
  During this time, you must take organizational measures (Page 651) to ensure the safety of the system components that are affected by this data. Alternatively, you can also use safe substitute values in your Safety program instead of the received data.

(S027)

*You can determine whether Safety mode has been deactivated on the communication partner's side as follows:

- Communication via SENDDP/RCVDP instructions: SENDMODE output at RCVDP (Page 612)
- Communication via SENDS//RCVS7 instructions: SENDMODE output at RCVS7 (Page 622)
- Communication with OPC UA Safety: TEST_MODE_ACTIVATED variable in the F-OPC UA Consumer DB
- Communication with Flexible F-Link: SENDMODE tag in the F-Communication DB (Page 346) for receiving

⚠ WARNING

(S7-1200, S7-1500) If **several F-CPUs** can be reached via a network (such as Industrial Ethernet) by the **same PG/PC**, you must take the following additional measures to ensure that the safety-related project data is downloaded to the correct F-CPU or that safety mode is deactivated in the correct F-CPU:

- Enter the serial number of the specific F-CPU in the *Safety Administration Editor*.

Depending on the respective CPU type (see documentation of the CPU), you have several possibilities for determining the serial number of an F-CPU:

- Print of the serial number on the housing of the F-CPU.
- Scanning of the ID link on the F-CPU.
- Display of the F-CPU
- Web server*
- Industrial Edge application user interface under /SIMATIC_MC/system/serialno*
- TIA Portal (I&M data)*

Note the following when downloading:

- If the "Load preview" window does not contain a warning, the connection to the correct F-CPU has been established. You can confirm the download of the safety-related project data without an additional check.
- If a serial number has not yet been entered in the *Safety Administration Editor* or if the connected F-CPU has a different serial number, check the proposed serial number of the F-CPU in the "Load preview" window or "Disable safety mode" window. If you determine an error, cancel the download or the disabling of safety mode. Otherwise, you can confirm the procedure.
- When the serial number is confirmed (online), it is always applied in the *Safety Administration Editor* and any existing entry is overwritten.

(S099)

* To make use of these possibilities, you must ensure a connection to the expected F-CPU. You achieve this through:

- A point-to-point connection to the respective F-CPU
- Verification of the flashing of the F-CPU after selecting the "Flash LED" check box in the online and diagnostics view of the F-CPU.
- Unplugging the network cable to the respective F-CPU
- For an S7-1500 F Virtual Controller, a unique connection can be achieved by a targeted switch to STOP of the Industrial Edge application.

Procedure for disabling safety mode

To disable safety mode, follow these steps:

1. Open the *Safety Administration Editor* of the corresponding F-CPU.
2. Open the "General (Page 83)" area in the area navigation.
3. Click the "Disable safety mode" button.
4. If access protection is set up for the F-CPU, enter the password for the F-CPU or legitimate yourself again as a user with F-right.
5. Only for S7-300/400 F-CPUs for which access protection is **not** set up:
   Make sure that you disable safety mode of the desired F-CPU by checking the Collective F-Signature in the dialog that is displayed next.
6. Confirm the deactivation.

Safety mode is then disabled.

The remaining runtime until an S7-1200/1500 F-CPU goes into STOP starts immediately after the safety mode is disabled. The remaining runtime is displayed in the *Safety Administration Editor* in the "General (Page 83)" area in online mode. In addition, the remaining runtime is displayed in the web server of an S7-1500 F-CPU as of firmware V2.9.

Reset remaining runtime until the F-CPU goes to STOP

To avoid the expiration of the remaining runtime and the associated STOP of the F-CPU, you can reset the remaining runtime in the *Safety Administration Editor*. This resets the remaining runtime to the configured value and starts it running again immediately after that.

Proceed as follows:

1. Open the *Safety Administration Editor* of the corresponding F-CPU.
2. Open the "General (Page 83)" area in the area navigation.
3. Click the "Reset remaining runtime" button.
4. If access protection is set up for the F-CPU, enter the password for the F-CPU or legitimate yourself again as a user with F-right.
5. Only for S7-300/400 F-CPUs for which access protection is **not** set up:
   Make sure that you disable the remaining runtime of the correct F-CPU by checking the Collective F-Signature in the dialog that is displayed next.
6. Confirm the reset of the remaining runtime.

NOTE

The remaining runtime is also displayed in the F-runtime group information DB (Page 154) of the respective F-runtime group in the MODE_REMAINING_TIME tag.

In case of two F-runtime groups, different values may be displayed caused by different update times.

The value "0" is not displayed after expiration of the remaining runtime, but the remaining runtime existing in the last cycle.

Enabling safety mode

To enable safety mode, the F-CPU must be switched from STOP to RUN mode.

In a redundant S7-1500HF system you have to set both HF-CPUs or the redundant system S7-1500HF to STOP before you restart the HF-CPUs.

Switching the F-CPU from STOP to RUN mode always enables safety mode, even if the Safety program has been modified or is not consistent.

If you have changed your Safety program, but have not recompiled and downloaded it, the F-CPU can revert to STOP mode.

Evaluating safety mode/disabled safety mode

If you wish to evaluate safety mode/disabled safety mode in the Safety program, you can evaluate the "MODE" tag in the F-shared DB (Page 154) for S7-300/400 F-CPUs or F-runtime group information DB for S7-1200/1500 F-CPUs (1 = Disabled safety mode). You use fully qualified access to access this tag (e.g. "F_GLOBDB".MODE or RTG1SysInfo.MODE).

You can use this evaluation, for example, to passivate F-I/O when the Safety program is in disabled safety mode. To do so, assign the "MODE" tag in the F-shared DB or F-runtime group information DB to all "PASS_ON" tags in the F-I/O DBs of the F-I/O that you wish to passivate.

⚠ WARNING

When the Safety program is in disabled safety mode, the "MODE" tag in the F-shared DB or F-runtime group information DB is also evaluated in disabled safety mode.

Even if the F-I/O is passivated in disabled safety mode by evaluating the "MODE" tag, the safety of the plant must be ensured during disabled safety mode by organizational measures (Page 651). *(S028)*

See also

Configuring the time limit for disabled safety mode (S7-1200, S7-1500) (Page 392)

10.7.3 Testing the safety program

Introduction

Tags of the safety program can be monitored at any time.

Controlling tags of the safety program is only possible in deactivated safety mode as some fault control measures of the safety program have to be disabled for this.

You can control the following tags of the safety program:

- Inputs and outputs of the F-I/O (channel values and value status (S7-1200, S7-1500))
- Tags in F-global DB (except DB for F-runtime group communication)
- Tags in instance DBs of F-FBs
- Tags in F-I/O DBs (for permitted tags see F-I/O DB (Page 164))

Procedure for monitoring tags of the safety program

Monitor the required tags of the safety program from an open watch table or from the *program editor* (program status).

1. Proceed as in the standard. More information can be found in the *STEP 7 help* in "Testing user programs".

Procedure for controlling tags of the safety program

Control the required tags of the safety program from an open watch table:

1. For modifying, deactivate the safety mode (Page 393) in the automatically shown dialog.
2. Terminate existing modify requests after testing is complete before activating safety mode.

Values in F-DBs can only be modified online in the F-CPU. If the value is also to be changed offline, you must do this by editing the start value offline and compiling the safety program.

Proceed as follows to control tags of F-I/O:

1. Create a separate row for each channel value and value status (S7-1200, S7-1500) to be modified. The control value must correspond to the channel value or value status.
2. Set "start of scan cycle" or "end of scan cycle" and "permanent" or "once".

   Regardless of the trigger point setting, requests to modify inputs (PII) of F-I/O always become effective before the main safety block is executed and requests to modify outputs (PIQ) always become effective after execution of the main safety block.
3. (S7-300, S7-400) Create an additional watch table if you want to control more than 5 inputs/outputs.

NOTE

F-I/O can only be modified in RUN mode of the F-CPU.

You cannot modify a configured F-I/O from which neither a channel value or a value status (S7-1200, S7-1500), nor any tag from the associated F-I/O DB has been used in the safety program. In your safety program, you should therefore always use at least one tag from the associated F-I/O DB or at least one channel value or value status (S7-1200, S7-1500) of the F-I/O to be modified.

For inputs (PII), modify requests take priority over fail-safe value output, while for outputs (PIQ), fail-safe value output takes priority over modify requests. For outputs (channels) that are not activated in the properties for the F-I/O, modify requests affect the PIQ only, and not the F-I/O.

NOTE

The following applies for S7-1200/1500 F-CPUs:

To avoid invalid combinations of channel value and status value:

- The value status is set by the F-system automatically to 1 when setting a channel value to a value <> fail-safe value 0
- The fail-safe value 0 is automatically output when setting the value status to 0 for the associated channel value

⚠ WARNING

You need to specifically reset constant modify requests in the watch table in disabled safety mode.

Note that constant modify requests that are not correctly reset can remain active in the background even after a STOP/RUN transition of the F-CPU.

Because the F-CPU is in safety mode again after a STOP/RUN transition, the constant modify requests are no longer effective and are not shown in the watch table.

The requests become active again as soon as you disable safety mode again.

With a memory reset of the F-CPU, you can make sure that no constant modify requests are active in the background on the F-CPU. *(S029)*

Wiring test using watch table

You can carry out a wiring test for an input by changing an input signal and verifying whether or not the new value arrives in the PII.

You can carry out a wiring test for an output by changing the output with the Modify function and verifying whether the required actuator responds.

For the wiring test, note that a safety program must be running on the F-CPU, in which at least one channel value or value status (S7-1200, S7-1500) of the F-I/O to be monitored or modified or one tag from the associated F-I/O DB has been used.

For F-I/O that can also be operated as standard I/O (e.g., S7-300 fail-safe signal modules), you can also carry out the wiring test for outputs using the Modify function in STOP mode by operating the F-I/O as standard I/O rather than in safety mode.

Additional rules for testing (S7-300, S7-400, S7-1500)

Setting breakpoints in the standard user program will cause the following errors in the safety program:

- F-cycle time monitoring has expired
- Error during communication with the F-I/O

  (S7-1500) Fail-safe modules switch to safe mode after the configured F-monitoring time has expired.
- Error during safety-related CPU-CPU communication
- Internal CPU faults

If you nevertheless want to use breakpoints for testing, you must first disable safety mode. This will result in the following errors:

- Error during communication with the F-I/O
- Error during safety-related CPU-CPU communication

Difference between S7-1500 F-CPUs and S7-300/400 F-CPUs:

- If a breakpoint is activated and reached, the F-CPU goes directly to STOP after HOLD.
- If you want to switch to RUN again after HOLD to test your standard user program further, you can simulate this with S7-PLCSIM.

No access protection is initially necessary for test purposes, commissioning, etc. This means you can execute all offline and online actions without access protection, that is, without password prompt.

See also

Changing the safety program in RUN mode (S7-300, S7-400) (Page 404)

Downloading project data (Page 350)

10.7.4 Testing the Safety program with S7-PLCSIM/S7-PLCSIM Advanced

You can test your Safety program together with your standard program on a simulated CPU with *S7-PLCSIM* and without the need for hardware. Also observe warning S030 in the section "Notes on Safety Mode of the Safety Program (Page 441)".

You use *S7-PLCSIM* for SIMATIC Safety F-systems as you would for S7 standard systems. Note the following special features:

Safety mode/disabled safety mode

We recommend that you test your Safety program in safety mode to detect whether the F-CPU goes into STOP as early as in the test phase of your Safety program in *S7-PLCSIM* as a result of, for example, that the results of instructions were outside the permitted range for the data type.

The following simulations can be run in *S7-PLCSIM*, just as on an actual F-CPU, in disabled safety mode only.

- Modifying tags in F-DBs and F-I/O DBs.

(S7-1200, S7-1500) To prevent unintentional modification of tags in F-DBs and F-I/O DBs in Safety mode, we recommend that you do not select the "Enable/disable modification of non-inputs" button in *S7-PLCSIM*

During the simulation with *S7-PLCSIM* , monitoring of the maximum cycle time of the F-runtime group and the cycle time warning limit of the F-runtime group (S7-1200, S7-1500) are disabled.

Differences between a simulated F-CPU and real F-CPU

Note that *S7-PLCSIM* does not fully behave like a real F-CPU down to the individual details. In particular, the startup behavior of an F-I/O cannot be simulated exactly.

Input simulation of F-I/O

Simulation of inputs (channel values) in *S7-PLCSIM:*

You simulate inputs (channel values) of F-I/O in *S7-PLCSIM* as you would inputs (channel values) of standard I/O. Keep in mind the following notes/restrictions:

During the transition of the F-CPU from "STOP" to "RUN" mode in *S7-PLCSIM*, all inputs (channel values) of F-I/O are initialized with 0 in the process image input (PII).

The inputs (channel values) can be simulated starting from the 2nd cycle and are then available in the PII.

Simulation of inputs (value status) in *S7-PLCSIM:*

(S7-1200 G1, S7-1500) By simulating inputs (value status) of F-I/O, you can simulate incoming and outgoing F-I/O / channel faults. Keep in mind the following notes/restrictions:

- To realistically simulate the behavior of the F-I/O, you must note the connection between channel value and value status on the real F-I/O. The combination value status = 0 and channel value <> fail-safe value (0) is invalid and can result in the simulation deviating from the behavior of the real F-CPU.
- During the transition of the F-CPU from "STOP" to "RUN" in *S7-PLCSIM*, all inputs (value status) of F-I/O are initialized with 1 in the process image input (PII). This means you can start with the simulation of inputs (channel values) immediately without simulation of the inputs (value status). For more information, refer to the sections "Addressing F-I/O (Page 157)" and "Value status (S7-1200, S7-1500) (Page 159)".
- The simulation of inputs (value status) in *S7-PLCSIM* does not have an effect on the QBAD and PASS_OUT tags in the F-I/O DB. Note that with real F-I/O, QBAD and PASS_OUT can be 1 as soon as the value status is 0 for at least one channel of the F-I/O (see tags of the F-I/O DB: PASS_OUT/QBAD/QBAD_I_xx/QBAD_O_xx and value status (Page 169)).
- For F-I/O configured with "Behavior after channel fault" = "Passivation of the complete F-I/O", use the tag PASS_ON in the F-I/O DB for simulation of the passivation of the complete F-I/O for F-I/O / channel faults. If you limit the simulation to passivation of individual inputs (channel value including value status), the behavior of the simulation will deviate from that of the real F-CPU.

- You can also use the PASS_ON tag in the F-I/O DB for F-I/O without value status to simulate the passivation of the entire F-I/O in case of F-I/O or channel faults.
- To simulate an F-I/O / channel fault of the SM 336; AI 6 x 13Bit or SM 336; F-AI 6 x 0/4...20 mA HART with configuration "Behavior after channel fault" = "Passivate channel", you must simulate the inputs (channel values) with $7FFF_H$ (for overflow) or 8000_H (for underflow).
- For F-I/O which does not support the "RIOforFA-Safety" profile, you must run a user acknowledgment with a positive edge at the ACK_REI tag of the F-I/O DB as with a real F-I/O for reintegration after the value status has changed from 0 to 1 or when the channel value has changed from $7FFF_H$/8000_H to unequal $7FFF_H$/8000_H (see above) when ACK_NEC = 1 of the F-I/O DB. Reintegration takes place automatically in all other cases possibly deviating from the real F-I/O.

Update times

Keep in mind that the status of the inputs (channel values or value status (S7-1200 G1, S7-1500)) that you are monitoring in the SIM table in *S7-PLCSIM* is only identical to the status being processed in the Safety program if there is no passivation of the associated F-I/O.

When the F-I/O is passivated, the Safety program operates with substitute values (channel value and value status (S7-1200 G1, S7-1500) =0).

Communication via SENDDP/RCVDP instructions

The following applies for SENDDP/RCVDP (S7-300, S7-400) instructions and SENDDP/RCVDP instructions with Version < 3.0 (S7-1200, S7-1500):

You cannot simulate communication between F-CPUs with the SENDDP and RCVDP instructions in *S7-PLCSIM*. You can, however, use the SENDDP and RCVDP instructions together with *S7-PLCSIM*. During simulation in *S7-PLCSIM*, the RCVDP instruction outputs the substitute values pending at its inputs SUBBO_xx and SUBI_xx ((S7-1200, S7-1500) or alternatively SUBDI_00). The SENDDP and RCVDP instructions signal this with 1 at output SUBS_ON.

For SENDDP/RCVDP instructions with Version >= 3.0 (S7-1200, S7-1500) the following applies:

During the simulation with *S7-PLCSIM* it is possible to simulate the received data and the information "Deactivated Safety Mode" (RCVDP) or respectively the information "Substitute value output" (SENDDP) in the corresponding transfer area for inputs. Note the following notes:

- The simulated values do not become active until you set the SIMULATION bit for the first time in the respective simulation control word (see the following table) after the F-system has started up. Before setting the SIMULATION bit, the RCVDP instruction outputs the fail-safe values that are pending at its inputs SUBBO_xx and SUBI_yy (or alternatively SUBDI_00).
- The setting of the SEND_MODE bit in the simulation control word causes a setting of the SENDMORE output for the RCVDP instruction.
- The setting of the STATUS_SUBS bit in the simulation control word causes a setting of the SUBS_ON output for the SENDDP instruction.

- Reserved bits in the simulation control word always have to be 0.
- During a STOP/RUN transition from *S7-PLCSIM* the most recently simulated values in the transfer area for inputs are kept.

The start address(es) of the configured transfer area for the input and output data can be found in the respective configuration (see also "Communication via SENDDP/RCVDP instructions (S7-1200, S7-1500): (Page 248)").

Table 10-1 Structure of the relevant transfer area for inputs of the simulation control word (instruction RCVDP)

Byte	Meaning	Comment
0	RD_BO_15 ... RD_BO_08	
1	RD_BO_07 ... RD_BO_00	
	DINTMODE=0:	
2	RD_I_00	Word RD_I_00, MSB[1] first
3		
4	RD_I_01	Word RD_I_01, MSB[1] first
5		
	Alternative DINTMODE=1:	
2	RD_DI_00	High Word from RD_DI_00, MSB[1] first
3		
4		Low Word from RD_DI_00 **XOR 0x8000**, MSB[1] first
5		
6	Simulation control word (High Byte)	Bit 0...6: Reserved Bit 7: SIMULATION: Activating RCVDP simulation
7	Simulation control word (Low Byte)	Bit 0: SEND_MODE: Set output SENDMODE Bit 1...7: Reserved
8 ... 11	Reserved	

[1] MSB: most significant bit

Table 10-2 Structure of the relevant transfer area for inputs of the simulation control word (instruction SENDDP)

Byte	Meaning	Comment
0	Simulation control word (High Byte)	Bit 0: STATUS_SUBS: Set output SUBS_ON Bit 1...6: Reserved Bit 7: SIMULATION: Activating SENDDP simulation
1	Simulation control word (Low Byte)	Bit 0...7: Reserved
2 ... 5	Reserved	

(S7-300, S7-400) Communication via the SENDS7/RCVS7 instructions

You cannot simulate communication between F-CPUs with the SENDS7 and RCVS7 instructions in *S7-PLCSIM*. You can, however, use the SENDS7 and RCVS7 instructions together with *S7-PLCSIM*.

During simulation in *S7-PLCSIM*, the RCVS7 instruction outputs the initial values specified in the communication DB as fail-safe values. The SENDS7 and RCVS7 instructions signal this with 1 at output SUBS_ON.

Communication with OPC UA Safety

You can simulate CPU-CPU communication with OPC UA Safety with S7-PLCSIM Advanced as of V7.0. Use the "TCP/IP Single Adapter" option under "Online Access" for this. Note the following warning if the Safety program associated with the connected SafetyProvider is running on an S7-PLCSIM:

⚠ WARNING

If the TEST_MODE_ACTIVATED variable in the F-OPC-UA-Consumer-DB is set to "1" during communication with OPC UA Safety, you can no longer assume that the data at the SAFETY_DATA variable is formed in a safety-related manner. This is the case when the connected SafetyProvider is in deactivated safety mode on a PLCSIM instance or the EnableTestMode SAPI parameter of the connected SafetyProvider has been set in the associated user program. During this time, you must take organizational measures (Page 651) to ensure the safety of the system components that are affected by this data. Alternatively, you can also use safe substitute values in your Safety program with TEST_MODE_ACTIVATED = 1 instead of the data in the SAFETY_DATA tag. *(S108)*

Communication with Flexible F-Link

Note the following warning if the Safety program on the sender side is running on an S7-PLCSIM:

⚠ WARNING

If the SENDMODE tag in the F-communication DB (Page 299) is set to "1" for receiving during communication with Flexible F-Link, you can no longer assume that the data at the RCV_DATA tag is formed in a safety-related manner.

During this time, you must take organizational measures (Page 651) to ensure the safety of the system components that are affected by this data. Alternatively, you can also use safe substitute values in your Safety program with SENDMODE = 1 instead of the data in the RCV_DATA tag. *(S086)*

Inconsistent Safety program (S7-1200, S7-1500)

If the CPU goes into STOP in *S7-PLCSIM* with the diagnostic entry "Safety program: inconsistent", the F-CPU is not initialized correctly in *S7-PLCSIM* yet. Perform a memory reset of the F-CPU in *S7-PLCSIM* and download the project data once again to the CPU in *S7-PLCSIM*.

See also

F-communication DB (S7-1200, S7-1500) (Page 346)

10.7.5 Changing the safety program in RUN mode (S7-300, S7-400)

Introduction

You make changes to F-blocks offline in the *program editor* in the same way as for a standard program. You download the changed F-blocks in RUN to the F-CPU in the "disabled safety mode (Page 393)".

NOTE

If you do **not** want to make changes to the safety program during operation, see Creating F-blocks in FBD/LAD (Page 150).

Procedure for changing the safety program in RUN mode

To change the safety program, follow these steps:

1. Change the main safety block or F-FB and its associated instance DB, F-FC, or F-DB in the *Program editor*.
2. Download the changed F-block(s) to the F-CPU (for procedure, see Downloading project data (Page 350)). The entire program is then automatically compiled.
3. If safety mode is active, the "Load preview" dialog will prompt you to deactivate it and to enter the password for the safety program.

NOTE

When downloading in disabled safety mode, you can only download the fail-safe blocks created by you (main safety blocks, F-FB, F-FC, or F-DB), F-application blocks, or standard blocks and their associated instance DBs. If you download automatically added F-blocks (F-SBs or automatically generated F-blocks and associated instance DBs, F-shared DB), the F-CPU can go to STOP mode or safety mode can be activated.

Therefore, always select individual blocks only when downloading in disabled safety mode.

Sequence for downloading changes

Changes in the safety program in RUN mode when safety mode is disabled can, for example, cause the status of an actuator to change as a result of program changes.

After changes, start by downloading the safety program and then the function of the standard user program monitored by the safety program.

Restrictions on safety-related CPU-CPU communication

During operation (in RUN mode), you cannot establish new safety-related CPU-CPU communication by means of new SENDDP/RCVDP or SENDS7/RCVS7 instructions.

To establish new safety-related CPU-CPU communication you must always download the relevant safety program consistently to the F-CPU while in STOP mode after inserting a new SENDDP, SENDS7, RCVDP, or RCVS7 instruction.

Restrictions on F-runtime group communication

You cannot make any changes to the safety-related communication between F-runtime groups in RUN mode. This means that you cannot assign, delete, or change any DBs for F-runtime group communication of an F-runtime group.

Following changes in the F-runtime group communication, you must always download the safety program consistently to the F-CPU while in STOP mode.

Restrictions on F-I/O access

If during operation (in RUN mode), you insert an F-I/O access to an F-I/O of which no single channel value or tag from the associated F-I/O DB has yet been used in the safety program, the F-I/O access only becomes effective when the safety program is downloaded consistently to the F-CPU.

Changing the standard user program

You can download changes in the standard user program when the F-CPU is in RUN mode, regardless of whether safety mode is enabled or disabled.

⚠ WARNING

(S7-300, S7-400) In productive operation, it must be prevented that changes to the standard user program are unintentionally made in the safety program as well. For this purpose, you must configure the protection level "Write protection for fail-safe blocks" and configure a password for the F-CPU.

With the "Write protection" or "Write/read protection" protection level, the password would apply to the standard and the safety program. *(S001)*

Procedure for applying changes to the safety program

If you download individual F-blocks to the F-CPU during operation (in RUN mode), the F-system blocks (F-SBs) and the automatically generated F-blocks are neither updated nor downloaded, resulting in an inconsistent safety program in the F-CPU. Use the following procedure to apply changes to the safety program:

1. Download the safety program consistently to the F-CPU, and activate safety mode by switching the F-CPU from STOP to RUN mode (for procedure, see Downloading project data (Page 350)).
2. Follow the steps described in Acceptance of Changes (Page 433).

10.7.6 Safety program in RUN mode (S7-1200, S7-1500)

Introduction

You make changes to F-blocks offline in the *program editor* in the same way as for a standard program. Loading the changed F-blocks in RUN to the F-CPU is done with the Fast Commissioning mode in deactivated safety mode (Page 393).

The Fast Commissioning mode distinguishes between "Fast Compile" (default setting) and "Consistent Compile".

Fast Commissioning mode with "Fast Compile"can be used for minor software changes to the Safety program, e.g. for testing and commissioning. It enables a fast compile, as only the F-blocks created by the user are compiled here.

In contrast, the Fast Commissioning mode with "Consistent Compile" can be used, for example, at the end of commissioning to compile the safety program consistently and then download it consistently to the F-CPU. In this case, you forego the advantage of the fast compiling, but after downloading the safety program to the F-CPU, you can enable safety mode again immediately with a short STOP-RUN transition of the F-CPU.

NOTE

Do not switch the F-CPU to STOP in Fast Commissioning mode with "Fast Compile". Otherwise, the F-CPU does not restart afterwards because the F-CPU contains an inconsistent safety program.

Solution:

- Download a consistent safety program to the F-CPU and then carry out a STOP-RUN transition. The deactivated safety mode is hereby terminated.
- If your F-CPU supports "Consistent Compile", select "Consistent Compile" and download the safety program consistently to your F-CPU. After that, a STOP-RUN transition is possible.

NOTE

Fast Commissioning mode with "Fast Compile" behaves differently with *S7-PLCSIM* than with a real F-CPU.

An *S7-PLCSIM* restarts after a STOP.

Requirements

To activate and use Fast Commissioning mode, the following requirements must be met:

- You can use Fast Commissioning mode with "Fast Compile" as of Safety System version V2.4. It is available for S7-1200 G2 F-CPUs as of firmware V1.0, S7-1200 G1 F-CPUs as of firmware V4.5 and S7-1500 F-CPUs as of firmware V2.0.
- You can use the Fast Commissioning mode with "Consistent Compile" from Safety System version V2.5. It is available for S7-1200 G2 F-CPUs as of firmware V1.0 and for S7-1500 F-CPUs as of firmware V3.0. With S7-1500 F-Software Controller, it is available as of Safety System version V2.5 only for approved IPCs as of firmware V30.0.

  NOTE

  S7-1500 F Software Controller

  The approved IPCs can be found in the table in the Product information for the F-CPUs (https://support.industry.siemens.com/cs/ww/en/view/109478599).

  If IPCs that have not been approved are used, an error or restart of the F Software Controller occurs. The download must then be repeated in STOP.

- The "Safety mode can be disabled" option must be selected in the "Settings" area of the *Safety Administration Editor*.
- The offline Safety program must be consistent and identical to the online Safety program.
- The F-CPU is in RUN.

Enabling Fast Commissioning mode

You enable Fast Commissioning mode by clicking the "Enable Fast Commissioning" button in the "General" area of the Safety Administration Editor.

Here, "Fast Compile" is selected as the default setting.

Once Fast Commissioning mode with "Fast Compile" is active, the behavior when compiling changes (shortcut menu "Compile > Software (only changes)" and "Compile > Hardware and software (only changes)") switches to the behavior when compiling changes in Fast Commissioning mode with "Fast Compile".

As soon as you change to the "Consistent Compile" setting, you can also compile the safety program consistently and download it to the F-CPU in RUN.

NOTE

After enabling Fast Commissioning mode, all other areas in the *Safety Administration Editor* are grayed out except the "General" area. Write access in the *Safety Administration Editor* is not permitted, also not via Openness. Read access is still possible.

While Fast Commissioning mode with "Fast Compile" is activated, the *Safety Administration Editor* displays the information "Fast Commissioning is activated" as the status of the safety program.

No signatures are calculated in Fast Commissioning mode with "Fast Compile". Therefore, no statement can be made in this mode about the status of the safety program and the F-blocks with regard to the offline-online comparison. The status of the signature comparisons is displayed with the icon in this case.

Downloading changes in Fast Commissioning mode

To download changes to the Safety program to the F-CPU in RUN in Fast Commissioning mode, the following 3 conditions must be met:

- The F-CPU is in RUN.
- Safety mode is deactivated.
- Fast Commissioning is enabled.

If at least one of these 3 conditions is not met, the "Download" dialog displays a note that download in RUN mode is not possible.

NOTE

In case of a connection abort between the PG/PC and the F-CPU during loading in RUN mode, loading to the F-CPU can **not** be completed properly. Even the indicators of the safety program on the display and in the web server of the F-CPU are no longer current.

A subsequent loading in RUN is no longer possible in this case, even though all requirements for Fast Commissioning mode are being met.

Solution: Switch the F-CPU to STOP mode and download a consistent safety program to the F-CPU.

NOTE

If the work memory utilization of the F-CPU is > 80%, a download in RUN mode with "Consistent Compile" may be aborted and the download to the F-CPU may not be properly completed. Even the indicators regarding the safety program on the display and in the web server of the F-CPU are no longer current.

A subsequent download in RUN is no longer possible, even though all requirements for Fast Commissioning mode are met.

Solution: Switch the F-CPU to STOP mode and download a consistent safety program to the F-CPU.

Supported changes in Fast Commissioning mode

While Fast Commissioning mode is enabled, you cannot arbitrarily change your Safety program. The following is an overview of the supported changes.

- Changes to F-FBs/F-FCs/F-DBs. See "Changes not supported" for exceptions.
- Changes to fail-safe-compliant PLC data types. See "Changes not supported" for exceptions.
- Changes to the standard user program that influence the safety program, for example, DBs used in both the standard user program and the safety program.
- Creating and using F-blocks. See "Changes not supported" for exceptions.

Unsupported changes in Fast Commissioning mode with "Fast Compile"

While Fast Commissioning mode with "Fast Compile" is activated, you cannot make changes to your safety program at will. The following is an overview of the changes **not** supported .

Type of change	System response
All changes to the hardware configuration.	A STOP is required for the download to the F-CPU.
Change in an F-FB/F-FC:	
• Insertion of a new call of a time-processing instruction (for example, TON, TOF, TP, MUT_P...) or an ACK_OP/ACK_GL instruction. (Time value parameters can be reassigned in existing calls).	Compiling errors
• Insertion of a new call of a SENDDP/RCVDP instruction.	
• Insertion of a new access to an F-I/O (access to a tag of the F-I/O DB or to the process image) that has not yet been used in the F-runtime group.	
• Insertion of a new access to a communication with Flexible F-Link (access to a tag of the F-communication DB) which has not yet been used in the F-runtime group.	
Deleting in an F-FB/F-FC:	
• Deletion of a call of a time-processing instruction (for example, TON, TOF, TP, MUT_P...) or an ACK_OP/ACK_GL instruction.	Compiling errors when the instances are also deleted
• Delete a call to a SENDDP/RCVDP instruction.	Compiling errors
• Deletion of the last access to an F-IO (access to a tag of the F-I/O DB or to the process image) so that it is no longer used in the F-runtime group.	
• Deletion of the last access to a communication with Flexible F-Link (access to a tag of the F-communication DB) so that it is no longer used in the F-runtime group.	
Deleting an F-FB or F-FC when it was called before Fast Commissioning mode was activated. However, a call can be deleted.	Compiling errors
Deleting an F-DB when tags of this F-DB were already being used before Fast Commissioning mode was activated.	Compiling errors
Deleting a standard DB when tags of this DB were already being used for reading before Fast Commissioning mode was activated.	Compiling errors
Deleting a tag from the standard user program when it was already being used for reading before Fast Commissioning mode was activated.	Compiling errors
Structural changes of fail-safe-compliant PLC data types that are referenced in a Flexible F-Link communication.	Compiling errors
Creating library types of blocks	Compiling errors
Adding/deleting of:	
• F-OBs	Compiling errors
• Flexible F-Link communications	Blocking in the Safety Administration

Type of change	System response
• F-runtime groups (including changes)	Editor
• F-CD/F-MS connections (including changes)	
Inserting an F-FB/F-FC (e.g. from a library or a different F-CPU) with global access to an F-DB.	Compiling errors
Changing the instruction version in the "Instructions" task card.	Compiling errors
Creating and changing instruction profiles	
Downloading a Safety program compiled in Fast Commissioning mode from the F-CPU to the PG/PC.	Interlocking by loading the station from the device
Changes in the "Settings" area of the *Safety Administration Editor*.	Blocking in the *Safety Administration Editor*
Changes to the password for the safety program.	
Renaming of the F-OB	Errors during download
All changes with Openness. (Name area *SafetyAdministration*)	Are rejected with an exception.

NOTE

For the changes marked with "Compiling errors", the compiling of the safety program is aborted in Fast Commissioning mode. An additional note is displayed in the inspector window under "Info > Compile" indicating the F-block containing the unsupported change.

NOTE

The table "Unsupported changes in Fast Commissioning mode with Fast Compile" serves as an aid. When you see errors during compiling that refer to blocks that you have not programmed yourself, an unsupported change was most likely made. In this case, undo the previous changes or exit Fast Commissioning mode and compile the entire safety program.

NOTE

Logging in the F-change history is incomplete when Fast Commissioning mode with "Fast Compile" is activated, the following entries are not recorded:

- Collective F-Signature
- Compiling time stamp
- Compiled F-blocks with signature and time stamp

Unsupported changes in Fast Commissioning mode with "Consistent Compile"

In Fast Commissioning mode with "Consistent Compile", you also cannot change your safety program at will. Compared to "Fast Compile", however, many more changes to the safety program are possible. The following is an overview of the changes **not** supported .

NOTE

Note that the changes mentioned here must not be present when enabling Fast Commissioning mode with "Consistent Compile". Therefore, make sure that the offline safety program is consistent and identical to the online safety program. If you do not observe this, the F-CPU may STOP.

Type of change	System response
All changes to the hardware configuration.	A STOP is required for the download to the F-CPU.
Adding/deleting of:	
• Flexible F-Link communications	Blocking in the *Safety Administration Editor*
• F-runtime groups (including changes)	
Changes in the "Settings" area of the *Safety Administration Editor*.	
Changes to the password for the safety program.	
Renaming of the F-OB	CPU goes to STOP
All changes with Openness. (Name area *SafetyAdministration*)	Are rejected with an exception.
Insertion of a new call of an ACK_GL instruction.	Possible, but results in passivation of all F-I/Os. Reintegration is only possible with a STOP-RUN transition.
Insertion of a new access to an F-I/O (access to a tag of the F-I/O DB or to the process image) that has not yet been used in the F-runtime group.	Possible, but results in passivation of the affected F-I/O. Reintegration is only possible with a STOP-RUN transition.
Deletion of a call to an ACK_GL instruction.	Possible, but results in passivation of all F-I/Os. Reintegration is only possible with a STOP-RUN transition.
Deletion of the last access to an F-IO (access to a tag of the F-I/O DB or to the process image) so that it is no longer used in the F-runtime group.	Possible, but results in a diagnostic buffer entry via a "Profisafe communication error".

Exiting Fast Commissioning mode

To exit Fast Commissioning mode with "Fast Compile" and return to Safety mode, proceed as follows:

1. Click on the "Deactivate Fast Commissioning" button.
2. Compile the Safety program with Software (rebuild all).
3. Download the safety program to the F-CPU. The F-CPU is set to STOP mode by this.

   In a redundant S7-1500HF system, you have to select "Stop R/H system" in the "Load preview" dialog.

After that, the online Safety program is consistent again and the F-CPU can be switched to RUN with Safety mode enabled.

To exit Fast Commissioning mode with "Consistent Compile" and return to Safety mode, proceed as follows:

1. Click on the "Deactivate Fast Commissioning" button.
2. Perform a STOP-RUN transition to enable safety mode again.

10.8 F-change history

Enable the logging of changes of the safety-related project data by using the option "Enable F-change history" in the *Safety Administration Editor*.

An F-change history is created for each F-CPU in the project navigation under "Common data/logs".

The following is logged in the F-change history:

- Collective F-Signature
- User name
- Compile time stamp
- Download of the safety-related project data with time stamp
- Compiled F-blocks with signature and time stamp

The F-change history can contain a maximum of 5000 entries per F-CPU. When the 5000 entries are exceeded, a new F-change history is created using the naming scheme "F-change history <CPU name> YYYY-MM-DD hh:mm:ss".

After a project upgrade, the "Go to" function is not supported anymore for the F-change history of the project for the entries which were created before *STEP 7 Safety* V15.1.

NOTICE
The connection between the F-CPU and the associated F-change history is made through the name of the F-change history. Therefore, do not rename the F-CPU and the F-change history. If you rename the F-CPU or the F-change history, a new F-change history with the current name of the F-CPU is started.

NOTE

You are not permitted to use the F-change history to identify changes of the safety-related project data for acceptance of changes.

To accept changes, proceed as described under Acceptance of Changes (Page 433).

NOTE

We recommend activating the F-change history before changing over to productive operation.

System acceptance

11

11.1 Overview of System Acceptance

Introduction

When performing a system acceptance test, all the standards and guidelines (for example PROFINET Installation Guidelines) relevant to the specific application must be complied with. This also applies to systems that are not "subject to acceptance". For the acceptance, you must consider the requirements in the Certification Report (https://support.industry.siemens.com/cs/ww/en/view/109825198).

As a general rule, the acceptance of an F-System is performed by an independent expert. The independence required of the expert must be defined in the safety plan and depends on the required PL/SIL.

Observe all warnings in this manual.

⚠ WARNING

The configuration of F-CPUs and F-I/Os as well as the programming of F-blocks must be carried out in TIA Portal as described in this documentation. To ensure safe operation with the SIMATIC Safety system, you must comply with the requirements of the relevant standards, such as IEC 61508. The section System acceptance (Page 413) describes procedures and tools of STEP 7 Safety to meet these requirements. *(S056)*

Proof of the correct implementation of the safety-related project data

In order for a system acceptance to be granted, you must assess and document the correctness of the individual components. For documentation of the component characteristics, you must create the safety summary.

The following characteristics must be covered:

- Correctness of the safety program including hardware configuration (including testing) (Page 414)
- Completeness of the safety summary (Page 415)
- Compliance of the system library elements used in the safety program with Annex 1 of the Report for the TÜV certificate (Page 416)
- Compliance of the know-how protected F-blocks used in the Safety program with their safety summary. (Page 417)
- Completeness and correctness of the hardware configuration (Page 418)
- Completeness and correctness of safety-related communication (Page 424)

- Identity of online and offline program (Page 430)
- Other characteristics (Page 431) such as software version, use of data from the standard user program

After the acceptance, you should archive all relevant documents and also the project data so as to make the accepted project available as a reference for a subsequent acceptance.

Safety summary

The safety summary (Page 388) is the documentation of the project that is required for system acceptance

11.2 Correctness of the safety program including hardware configuration (including testing)

The correctness of software cannot only be ensure through tests and verifications during commissioning, but rather already requires the observance of a wide variety of measures during creation. Also see warning S062 on this in chapter "Overview (Page 23)".

Verification/function test

Already during the creation, you will test (Page 391) your safety program and the associated hardware configuration. You must carry out these tests with regard to the specification of your safety functions and document them before you perform the system acceptance.

In order to perform a code review of your safety program and/or document the accepted program code, you can include the source code of all F-blocks in the safety summary (Page 388). To do this, select the "All" option when printing.

If you want to carry out a function test after loading, you have to carry out program identification. More information is available in "Downloading project data (Page 350)".

The correct function of the safety program must be guaranteed by complying with all steps from the "Overview of System Acceptance (Page 413)" chapter before it can be used productively. When using configuration control (option handling), you must ensure correct operation of the safety program for all possible station options by performing appropriate functional tests. You should archive the test reports along with the safety summary and the acceptance documents.

Times, for example monitoring times (Page 629) and delay times, can only be verified to a limited extent with functional tests (Page 350). For this reason, you should check these times selectively to determine whether they are dimensioned correctly, for example, using the safety summary.

Some of these times are itemized specifically in the safety summary, for example, the F-monitoring time (for communication between F-CPU and F-I/O) and the monitoring time of the safety-related CPU-CPU communication (TIMEOUT). To determine the monitoring times derived under normative conditions, the Excel file for response time calculation is available on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109783831). These have to be considered together with the practically determined conditions of the application. Note that these monitoring times have an impact on the response times of your safety functions.

Consistency of the safety program

Check in the "General information" section of the safety summary to determine whether the safety program was recognized as "consistent".

This is the case for S7-300/400 F-CPUs only if the following signatures are also identical:

- Collective F-Signature ("General information" section, "Collective F-Signature")
- "Signature of F-blocks with F-attribute" ("General information" section, "Current compilation")

Consistency of the safety program is required for the acceptance. If the signatures are not identical, you have the possibility to establish consistency by recompiling the safety program and re-creating the safety summary.

11.3 Completeness of the safety summary

Introduction

When you have an acceptance-ready version of your safety-related project data, you must perform and document additional checks of the safety summary to prove that the safety summary is complete and belongs to the safety program to be accepted.

Creating a safety summary

Follow the procedures described in Creating a safety summary (Page 388) to create the safety summary. In so doing, select "All" under Properties in the "Print" dialog to include the source code of your F-blocks and their utilized named value data types and PLC data types in the documentation.

Completeness of the safety summary

The safety summary of *STEP 7 Safety* is always created in its entirety. When performing subsequent processing steps (for example, creating copies, inserting (electronic) signatures, comments, etc.), the completeness must continue to be maintained.

If you created the safety summary with the "All" option, the source code of all F-blocks will also be printed. The printout of this source code also includes the footer, which contains the collective F-signature of the associated safety program.

The information "Number of pages in the safety summary" does not include the pages on which the source code of the F-blocks is output.

Association with the safety program

In the "General information" section of the safety summary, check whether the Collective F-Signature corresponds online and offline to the Collective F-Signature of the safety program to be accepted in the work area of the *Safety Administration Editor* under "General". If they are not the same, then the summary and safety program do not match. Bear in mind that the

F-BaseID is not included in the Collective F-Signature and, if relevant, must be checked separately.

11.4 Compliance of the system library elements used in the safety program with Annex 1 of the Report for the TÜV certificate

Introduction

STEP 7 Safety contains LAD/FBD instructions for programming of your safety program as well as F-system blocks for creating an executable safety program that have been created and tested by SIEMENS and certified by TÜV. The F-system blocks used are automatically inserted by the F-system based on the set safety system version (see section "Settings" area of the Safety Administration Editor (Page 91)).

To allow you to check whether the used versioned LAD/FBD instructions and F-system blocks correspond to Annex 1 of the Report for the TÜV certificate and to the versions you intend to use, these are listed in the safety summary.

Procedure

To check, download the current Annex 1 of the report for the TÜV certificate "SIMATIC Safety" from the Internet (https://support.industry.siemens.com/cs/ww/en/view/109825198).

Proceed as follows for the check:

⚠ WARNING

- (S7-1200, S7-1500) The versions of the versioned LAD/FBD instructions listed in the safety summary in the section "System library elements used in safety program" must correspond to the versions in Annex 1 of the Report for the TÜV certificate.
- (S7-300, S7-400) The versions, signatures and initial value signatures of the versioned LAD/FBD instructions and F-system blocks listed in the safety summary in the section "System library elements used in safety program" must correspond to the versions, signatures and initial value signatures in Annex 1 of the Report for the TÜV certificate.
- The versions of the versioned LAD/FBD instructions listed in the safety summary must meet the safety requirements of your application.

  Keep in mind possible differences in functionality of different versions specified in the section for the respective instruction.
- The safety system version listed in the safety summary under "Safety program settings" must match the versions in Annex 1 of the Report for the TÜV certificate. *(S054)*

In case of discrepancies, recheck whether you have the correct versions.

(S7-300, S7-400) Differences can also arise when there are F-blocks / instructions in your safety program that are not used.

11.5 Compliance of the know-how protected F-blocks used in the Safety program with their safety summary.

If you use know-how protected F-blocks (for example, from libraries) to program your Safety program, the source code for these is not output in the safety summary.

Therefore, the author of the know-how protected F-block must already carry out acceptance of the F-block and provide the following information:

S7-300/400 F-CPUs

- Signature and initial value signature of the know-how protected F-block
- Versions of all the used versioned LAD/FBD instructions
- Signatures and initial value signatures of all called F-blocks
- Signatures of all F-DBs accessed by the fail-safe block to be reused.

For system acceptance, you must perform the following checks using the safety summary:

- The signature and initial value signature of each know-how protected F-block listed in the safety summary in the section "F-blocks in Safety program" must correspond to the signature and initial value signature documented by the author.
- The versions of the versioned LAD/FBD instructions listed in the safety summary in the section "System library elements used in Safety program" must correspond to the versions of each know-how protected F-block documented by the author or must be functionally identical with them.
- The signatures and initial value signatures of the F-blocks called in each know-how protected F-block listed in the safety summary in the section "F-blocks in Safety program" must correspond to the signatures and initial value signatures (of the called F-blocks) documented by the author.

In case of differences, set the documented (or functionally identical) versions and use the F-blocks with the documented signatures and initial value signatures. If the version conflicts cannot be eliminated due to other dependencies, contact the author of the know-how protected block in order to obtain a compatible approved version.

S7-1200/1500 F-CPUs

- Signature of the know-how protected F-block
- Safety system version that was set while setting up the know-how protection
- Versions of all the used versioned LAD/FBD instructions
- Signatures of all called F-FBs/F-FCs in the know-how protected F-block.
- Signatures of all F-DBs accessed by the know-how protected F-block.

For system acceptance, you must perform the following checks using the safety summary:

- The signature of each know-how protected F-block listed in the safety summary in the section "Know-how protected F-blocks in the Safety program" must correspond to the signature documented by the author.
- The safety system version of each know-how protected F-block listed in the safety summary under "Know-how protected F-blocks in the Safety program" must correspond to one of the versions listed in Annex 1 of the Report for the TÜV certificate.
- The versions of the versioned LAD/FBD instructions of each know-how protected F-block listed in the safety summary in the section "Know-how protected F-blocks in the Safety program" must correspond to the versions documented by the author or must be functionally identical with them.
- The signatures of the F-blocks called in each know-how protected F-block listed in the safety summary in the section "Know-how protected F-blocks in the Safety program" must correspond to the signatures (of the called F-blocks) documented by the author.

In case of differences, set the documented (or functionally identical) versions and use the F-blocks with the documented signatures. If the version conflicts cannot be eliminated due to other dependencies, contact the author of the know-how protected block in order to obtain a compatible approved version.

11.6 Completeness and correctness of the hardware configuration

Introduction

The hardware configuration is an essential component of the project to be accepted. With the configuration of the hardware, you have set properties that can influence the safety of signals. You must document these settings with the safety summary to prove that you fulfill the safety requirements of your application.

The section "Hardware configuration of F-I/O" is available in the safety summary for this. This section consists of several tables:

- A table with information about the F-CPU and about the ranges of F-destination addresses used, the "Central F-source address" and the F-BaseID of the F-CPU.
- Overview tables with the F-I/O used.
- A table for each F-I/O with information on the F-I/O and all parameters of the F-I/O with the configured values.

NOTE

Note that you will find F-I/O that you address via safety-related I-slave-slave communication in the safety summary of the F-CPU of the I-slave and not in the safety summary of the F-CPU of the assigned DP master.

The safety summary of the F-CPU of the DP master contains a note in the overview table for this F-I/O indicating that it is not assigned to this F-CPU.

NOTE

When using shared devices:

Note that F-I/O that you address in a shared device can be found in the safety summary of the F-CPU of the IO controller to which it is assigned.

The safety summary of the other F-CPUs (IO controllers) that share the cross-project shared device contains a note in the overview table for this F-I/O indicating that it is not assigned to this F-CPU.

Procedure for checking that the hardware configuration is complete

Check whether the safety summary contains all the configured F-I/O. Also make sure that there is no F-I/O that you have not configured as belonging to this F-CPU.

NOTE

If configuration control (option handling) is used, the safety summary must contain all the F-I/Os of the maximum configuration. The following checks are to be carried out for al the F-I/Os of the maximum configuration.

Procedure for checking the correctness of the hardware configuration using the safety summary

To check the hardware configuration for correctness, proceed as follows:

1. Check in the "Hardware configuration of F-IO" section to verify the uniqueness of the PROFIsafe addresses.

   Observe the information in sections

   PROFIsafe addresses for F-I/O of PROFIsafe address type 1 (Page 67)

   PROFIsafe addresses for F-I/O of PROFIsafe address type 2 (Page 68)

   PROFIsafe addresses for F-I/O of PROFIsafe address type 3 (Page 70)

   Peculiarities when configuring fail-safe GSD based DP slaves and fail-safe GSD based I/O devices (Page 39)

   Recommendation for PROFIsafe address assignment (Page 63)

   – For F-CPUs to which F-I/Os of PROFIsafe address type 2 are assigned:

     Check if the "Central F-source address" parameter of the individual F-CPUs differs network-wide.

     (This also applies to F CPUs that have F-I/O of PROFIsafe address type 3 assigned to them, unless an individual active F-BaseID is generated for each F-CPU.)

   – Check that the setting of the F-BaseID parameter is as expected, i.e. inactive (value 0), "Delayed transfer" (value 0x7C6) or an individually generated value for the F-CPU. Observe also warnings S115 and S116.

– For F-I/Os of PROFIsafe address type 1 check whether the F-destination addresses comply with the following warning:

⚠ WARNING

F-I/Os of PROFIsafe address type 1 are uniquely addressed by their F-destination address (e.g. with the switch setting on the address switch).

The F-destination address (and therefore also the switch position on the address switch) of the F-I/O of PROFIsafe address type 1 must be unique network-wide* and CPU-wide**/*** (system-wide) **for the entire** F-I/O. For this, the F-I/O of PROFIsafe address type 2 and PROFIsafe address type 3 when F-BaseID is inactive (value = 0) must also be taken into account. *(S051)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

** "CPU-wide" means all F-I/O assigned to an F-CPU: Central F-I/O of this F-CPU, F-I/O for which the F-CPU is DP master/IO controller and assigned F-I/O in a shared device. An F-I/O that is addressed using I-slave-slave communication is assigned to the F-CPU of the I-slave and not to the F-CPU of the DP master / IO controller.

*** In a redundant S7-1500HF system, both F-CPUs of the redundant S7-1500HF system are to be considered as one F-CPU with regard to the PROFIsafe addresses. The "Central F-source address" and, if present, the F-BaseID are therefore set identically for both F-CPUs by the system.

NOTE

For more information on the assignment of PROFIsafe addresses that are unique for the CPU and across the network, see this FAQ (https://support.industry.siemens.com/cs/ww/en/view/109740240).

– For F-I/Os of PROFIsafe address type 2 check whether the F-destination addresses comply with the following warning:

⚠ WARNING

F-I/O of PROFIsafe address type 2 is uniquely addressed using a combination of F-source address ("Central F-source address" parameter of the assigned F-CPU) and F-destination address.

The combination of F-source address and F-destination address for each F-I/O of PROFIsafe address type 2 must be unique network-wide* and CPU-wide**/*** (system-wide). In addition, the F-destination address must not be occupied by F-I/O of PROFIsafe address type 1.

If you use a supported configuration (Page 64), you only have to ensure that the "Central F-source address" parameter of all F-CPUs is unique network-wide*. *(S052)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

** "CPU-wide" means all F-I/O assigned to an F-CPU: Central F-I/O of this F-CPU, F-I/O for which the F-CPU is DP master/IO controller and assigned F-I/O in a shared device. An F-I/O that is addressed using I-slave-slave communication is assigned to the F-CPU of the I-slave and not to the F-CPU of the DP master / IO controller.

*** In a redundant S7-1500HF system, both F-CPUs of the redundant S7-1500HF system are to be considered as one F-CPU with regard to the PROFIsafe addresses. The "Central F-source address" and, if present, the F-BaseID are therefore set identically for both F-CPUs by the system.

NOTE

For more information on the assignment of PROFIsafe addresses that are unique for the CPU and across the network, see this FAQ (https://support.industry.siemens.com/cs/ww/en/view/109740240).

– For F-I/O of PROFIsafe address type 3, check whether the F-destination addresses comply with the following warning:

⚠ WARNING

The F-BaseID may only be assigned manually in exceptional cases, when you are sure that the PROFIsafe addresses of the F-I/Os of the F-CPU are unique (worldwide). F-I/Os with the same F-BaseID must differ in the combination of F-source and F-destination address. We explicitly advise against manual assignment and recommend generation with the tools offered in the TIA Portal.

With the manual assignment of value "0" as F-BaseID, you lose the advantages of PROFIsafe address type 3 and must adhere to the specifications for PROFIsafe address type 2 (see warning S052). *(S115)*

– Check whether the PROFIsafe addresses of fail-safe GSD based DP slaves / GSD based I/O devices comply with the following warning:

⚠ WARNING

Check the documentation for your fail-safe GSD based DP slaves / fail-safe GSD based I/O devices to find out the valid PROFIsafe address type. If you do not find the necessary information, assume PROFIsafe address type 1. For configuring, proceed as described under PROFIsafe addresses for F-I/O of PROFIsafe address type 1 (Page 67), PROFIsafe addresses for F-I/O of PROFIsafe address type 2 (Page 68) or PROFIsafe addresses for F-I/O of PROFIsafe address type 3 (Page 70).

Set the F-source address for fail-safe GSD based DP slaves / fail-safe GSD based I/O devices according to the manufacturer's specifications. If the F-source address needs to correspond to the "Central F-source address" parameter of the F-CPU (PROFIsafe address type 2), you will find the latter in the "Properties" tab of the F-CPU. In this case, check the safety summary to verify that the value for the "Central F-source address" parameter corresponds to the value of the F-source address of the fail-safe GSD based DP slave/fail-safe GSD based I/O device. *(S053)*

2. Check the safety-related parameters (including F-monitoring time or F_WD_Time) of all configured F-I/O.

   You can find these parameters in the "Hardware configuration of F-I/O" section in the detailed tables for the F-I/O.

   The table consists of two parts:

   – The left part contains the parameters which refer to the F-I/O itself ("Module data").
   – The right part contains the parameters of the individual channels ("Channel parameters")

   These parameters must be set as prescribed by the safety requirements of your application.

   When using fail-safe GSD based DP slaves/GSD based I/O devices, note the relevant documents for the possible additional safety-related (technological) parameters.

NOTE

F-I/O that are to be assigned the same safety-related parameters (except for PROFIsafe addresses) can be copied during configuration. Except for the PROFIsafe addresses, you no longer have to check the safety-related parameters individually. It is sufficient to compare the "F-parameter signature (without addresses)" in the "Hardware configuration of the F-I/O" section in the overview table. This also applies to fail-safe GSD based DP slaves/GSD based I/O devices without i-parameters. For GSD based DP slaves/GSD based I/O devices with i-parameters, it might be that the "F-parameter signature (without addresses)" does not match, even though all safety-related parameters, except for the PROFIsafe addresses, do match. In this case, you need to compare all safety-related parameters.

Exception:

For F-I/Os that do not support the "RIOforFA-Safety" profile, you need to compare the "Behavior after channel fault" parameter, if any, in addition to the "F-parameter signature (without addresses)".

3. Check whether the article numbers of the F-I/O in the safety summary correspond to the article numbers of the F-I/O actually present in the system. If the article numbers are different, the F-I/O present must be compatible as a replacement part for the F-I/O listed in the safety summary.
4. For non-supported configuration, see Configurations supported by the SIMATIC Safety F-system (Page 64).

⚠ WARNING

If you create a configuration that is not one of the supported configurations, you must observe the following in addition to the warnings S051, S052, S053 and S115:

- Make sure that the F-I/O of this configuration appears in the safety summary and that an F-I/O DB has been created for it. Otherwise, you cannot use the F-I/O in this configuration. (Contact Customer Support.)
- For F-I/O in the PROFINET IO environment[1], you must check the safety summary for correctness of the PROFIsafe operating mode parameter (F_Par_Version). V2 mode must be set in the PROFINET IO environment. F-I/O that support only V1 mode must not be used in the PROFINET IO environment.
- Based on the safety summary, you must check whether the components of the PROFIsafe address centrally assigned for the F-CPU were transferred correctly into the respective F-I/Os:
  - PROFIsafe address type 3: F-BaseID from the "F-BaseID" parameter of the F-CPU.
  - PROFIsafe address type 2 and 3: F-source addresses from the "Central F-source address" parameter of the F-CPU.
- You must ensure the CPU-wide[2,3] and/or network-wide[4] unique assignment of the F-destination address for F-I/Os:
  - For F-I/O of PROFIsafe address type 1 or 2 (if you cannot set the F-source address to match the "Central F-source address" parameter of the F-CPU), you must ensure the network-wide[4] uniqueness of the PROFIsafe address through the unique assignment of the F-destination address alone.
  - In an unsupported configuration, you must check the uniqueness of the F-destination address individually for each F-I/O using the safety summary, regardless of the PROFIsafe address type.

    PROFIsafe address type 1: Network-wide[4] and CPU-wide[2,3]

    PROFIsafe address type 2 and 3: Only CPU-wide[2,3].

(S050)

1) The F-I/O is located in the "PROFINET IO environment" if at least part of safety-related communication with the F-CPU takes place via PROFINET IO. If the F-I/O is connected via I-slave-slave communication, also keep in mind the communication line to the DP master/IO controller.

2) "CPU-wide" means all F-I/Os assigned to an F-CPU: Central F-I/O of this F-CPU, F-I/O for which the F-CPU is DP master/IO controller and assigned F-I/O in a shared device. An F-I/O that is addressed using I-slave-slave communication is assigned to the F-CPU of the I-slave and not to the F-CPU of the DP master / IO controller.

3) In a redundant S7-1500HF system, both F-CPUs of the redundant S7-1500HF system are to be considered as one F-CPU with regard to the PROFIsafe addresses. The "Central F-source address" and, if present, the F-BaseID are therefore set identically for both F-CPUs by the system.

4) A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

NOTE

For more information on the assignment of PROFIsafe addresses that are unique for the CPU and across the network, see this FAQ (https://support.industry.siemens.com/cs/ww/en/view/109740240).

11.7 Completeness and correctness of safety-related communication

Introduction

Safety-related communication is based on the mechanisms of the standard communication of *STEP 7*.

Safety-related communication must also detect and uncover communication errors, which remain undetected in the lower-level standard communication. Further parameters are required for this, which you have to document and check on acceptance.

For this purpose, the "Block parameters for safety-related CPU-CPU communication", "Communications via Flexible F-Link Overview" and "Communications via OPC UA Safety Overview" sections are available in the safety summary.

The section "Block parameters for safety-related CPU-CPU-communication" contains up to two tables (for communication via PROFIBUS DP or PROFINET IO and for communication via S7 connections).

The section "Communications via Flexible F-Link Overview" contains a table with an overview of the connection configurations and a "Communications via Flexible F-Link for UDT" table for each used F-compliant PLC data type (UDT).

The section "Communications via OPC UA Safety Overview" contains multiple tables with an overview of all OPC UA data types used by SafetyProviders or SafetyConsumers, as well as a detailed view of all communication parameters for every SafetyProvider and every SafetyConsumer.

Not all safety-related communication is available for all F-CPUs. More information can be found in the section "Safety-related communication (Page 202)".

NOTE

The statements made regarding communication via Flexible F-Link in this section also apply when this communication is used between F-runtime groups.

11.7.1 Completeness and correctness of safety-related communication

(S7-300, S7-400) Observe the following warning for safety-related communication with the SENDS7/RCVS7 instructions:

⚠ WARNING

The value of the respective F-communication ID (input R_ID; data type: DWORD) can be freely selected; however, it must be odd and unique for all safety-related communication connections network-wide* and CPU-wide. The value R_ID + 1 is internally assigned and must not be used.

You must supply inputs ID and R_ID with constant values when calling the instruction. Direct read or write access to ID and R_ID in the associated instance DB is not permitted in the Safety program. *(S020)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

Observe the following warning for safety-related communication with the SENDDP/RCVDP instructions:

⚠ WARNING

The value for the respective F-communication ID (input DP_DP_ID; data type: INT) can be freely selected**; however, it must be unique network-wide* and CPU-wide**** at all times for all safety-related communication connections. You must check the uniqueness in the safety summary during acceptance of the safety program.

You must supply constant values*** to the DP_DP_ID and LADDR inputs when calling the instruction. Direct write access to DP_DP_ID and LADDR in the associated instance DB is not permitted in the safety program! *(S016)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, no connection is established at the DP_DP_ID input for a F-communication ID "0".

*** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, the DP_DP_ID input can also be supplied with variable values from a global F-DB. In this case also, you have to check during acceptance of the safety program that uniqueness is ensured at all times. You need to check the algorithm for forming the variable value accordingly. If you cannot ensure a unique F-communication ID during startup of the safety program, because it is only specified after startup of the safety program, you must make sure that the value at the DP_DP_ID input is "0" during this phase.

**** With a redundant S7-1500HF system, both F-CPUs of the redundant S7-1500HF system are to be regarded as one single F-CPU with regard to the DP_DP_ID.

Observe the following warnings for communication via Flexible F-Link:

⚠ WARNING

When a new Flexible F-Link communication is created in the Safety Administration Editor, a unique F-communication UUID for the communication is provided by the system. When communication is copied in the Safety Administration Editor within the parameterization table or copied to another F-CPU, the F-communication UUIDs are not regenerated and therefore not unique anymore. If the copy is used to configure a new communication relationship, you yourself must ensure the uniqueness. To do this, select the relevant UUIDs and generate them again via the "Generate UUID" context menu. You must ensure the uniqueness in the safety summary during acceptance, at the latest. *(S087)*

⚠ WARNING

For each communication connection via Flexible F-Link, you must ensure that the offsets of the elements of the F-compliant PLC data type (UDT) used for the communication connection match on the transmit and receive sides during acceptance of the safety program.

You can find the offsets in the respective safety summary. To do this identify the communication connection by means of the UUID. *(S088)*

11.7.2 Procedure to check the correctness of the configuration of communication with OPC UA Safety

Procedure to check the correctness of the project planning for every SafetyProvider

You must carry out the following verifications for each SafetyProvider in the respective SafetyProvider detail view of your safety summary.

- Check the correctness of the values of the SPI parameters.
- Check the correctness of the assignments of the local data on the SAPI parameters.
- Check whether the necessary uniqueness of the combination of active SafetyProviderID and active SafetyBaseID of all SafetyProviders is available at all times.
  - For static connections:

    The set values can be found in the respective SafetyProvider detail view.

    If the value for SafetyBaseIDConfigured has been generated by the system, you do not have to undergo the SafetyBaseID, since the sufficient uniqueness is ensured by the STEP 7 Safety Engineering. If you have entered the value yourself, you must verify

sufficient uniqueness. You will be shown whether this is necessary in the respective SafetyProvider detail view.

– For dynamic connections:

  Ensure correct implementation of the algorithm for generating the values of the local data that is mapped to the SafetyProviderID and SafetyBaseID SAPI parameters.

Note the following warnings:

WARNING

For the unique and secure authentication of a SafetyProvider, the combination of active SafetyProviderID and active SafetyBaseID at any point in time must be unique for all OPC UA Safety communication connections network-wide. **

If your network is not safely separated from the outside world through physical separation, the active SafetyBaseID at any point in time must be a value that was generated randomly in accordance with the OPC UA Specification OPC 10000-15. ***

You must ensure the uniqueness in the safety summary during acceptance of the safety program. If the active SafetyProviderID or active SafetyBaseID is not generated until runtime of your Safety program, you must ensure the correct implementation of the algorithm for the generation of the values. *(S102)*

* If you cannot ensure a unique combination during startup of the Safety program because the active SafetyProviderID or active SafetyBaseID is not formed until runtime of your Safety program, you must ensure that the active SafetyProviderID or active SafetyBaseID is "0" during this phase.

** SafetyProviders to which the same local data element of your safety program is mapped (Safety Multicast (Page 330)) at the SafetyData SAPI parameter are permitted to use the same combination.

*** If you have entered a value for the SPI parameter SafetyBaseIDConfigured of a SafetyProvider (either in the NodeSet or in the properties of a SafetyProvider), you yourself must ensure that the value meets the safety requirements of the OPC UA Specification OPC 10000-15. Alternatively, generate the value of SafetyBaseIDConfigured again by opening the properties tab of the affected SafetyProvider and clicking on the "Generate" button.

WARNING

Due to system constraints, formation of a SafetyBaseID at runtime of your Safety program is only possible in your standard user program.

Note that these program sections are therefore not captured by the collective F-signature and are not protected by "Access protection for safety-related project data (Page 100)" or "F-CPU access protection (Page 103)".

For this reason, you must take organizational measures (Page 651) to ensure that only authorized persons can change these program sections.

Print the relevant program sections and supplement your safety summary with this printout. *(S103)*

⚠ WARNING

If you have not defined any value for the SPI parameter of a SafetyProvider in the NodeSet for the SPI parameter, a sufficiently clear UUIDV4 is made available from the system as a value for the parameter of the respective SafetyProvider when importing the NodeSet into the TIA Portal.

Through subsequent copying of a SafetyProvider (e.g. copying the NodeSet to another F-CPU, copying the F-CPU or copying the project), values provided by the system are not generated again and are therefore not unique anymore.

If such a copy is used to configure a new communication relationship, you yourself need to ensure the uniqueness. To do this, open the properties tab of all affected SafetyProvider and generate the value of SafetyBaseIDConfigured again by clicking on the "Generate" button. The uniqueness must be checked in the safety summary during acceptance. *(S110)*

Procedure to check the correctness of the configuration for every SafetyConsumer

You must carry out the following verifications for each SafetyConsumer in the respective SafetyConsumer detail view of your safety summary.

- Check the correctness of the values of the SPI parameters.
- Check for compliance of the active SafetyProviderID and active SafetyBaseID with the SafetyProvider intended for communication.
  - For static connections:

    The set values can be found in the respective SafetyConsumer detail view.
  - For dynamic connections:

    Ensure correct implementation of the algorithm for generating the values of the local data that is assigned to the SAFETY_PROVIDER_ID and SAFETY_BASE_ID variables of the associated F-OPC-UA-Consumer-DB.
- Check the correctness of the substitute value.
- Check that the data structure of SafetyData of the SafetyConsumer matches the data structure of SafetyData of the intended SafetyProvider. This means that at each position of the data structure, the data type must match the data type on the corresponding position on the side of the SafetyProvider on the side of the SafetyConsumer. Different names may be permitted if it is ensured that there are no confusion of the meanings of the individual elements of the data structure. In particular, the correct order of elements of the same data type is relevant.

Note the following warnings:

⚠ WARNING

Due to system constraints, formation of a SafetyBaseID at runtime of your Safety program is only possible in your standard user program.

Note that these program sections are therefore not captured by the collective F-signature and are not protected by "Access protection for safety-related project data (Page 100)" or "F-CPU access protection (Page 103)".

For this reason, you must take organizational measures (Page 651) to ensure that only authorized persons can change these program sections.

Print the relevant program sections and supplement your safety summary with this printout. *(S103)*

⚠ WARNING

For secure authentication of the SafetyProvider whose SafetyData is to be evaluated, the values of the active SafetyProviderID and active SafetyBaseID of the SafetyConsumer must correspond to the values of the active SafetyProviderID and active SafetyBaseID of this SafetyProvider at all times*.

When accepting the Safety programs, you must ensure compliance in the relevant safety summary. If the active SafetyProviderID or active SafetyBaseID is not generated until runtime of your Safety program, you must ensure the correct implementation of the algorithm for the generation of the values. *(S104)*

* If you cannot yet provide the values when starting the Safety program because they are only created during the runtime of your Safety program, you must ensure that the active SafetyProviderID or active SafetyBaseID or the Enable "0" SAPI parameter in this phase.

⚠ WARNING

For each communication connection with OPC UA Safety, you must ensure when accepting the Safety program that the data structure of the OPC UA data types used by SafetyData matches on the SafetyProvider and SafetyConsumer side. In particular, the correct order of elements of the same data type is relevant. You can find the structures of the OPC UA data types used in the respective safety summary.

To identify the communication connection, note the following points:

- For static connections:

  To do this identify the communication connection by means of the combination of SafetyProviderIDConfigured and SafeBaseIDConfigured.
- For dynamic connections:

  To do this, identify the communication connection by means of the algorithm for the allocation of the values for SafetyProviderID and SafetyConsumerID.

(S111)

11.8 Identity of online and offline program

Once you have checked all properties of the offline safety program you must ensure that the safety program is identical on the F-CPU on which it is supposed to be run.

WARNING

In a redundant S7-1500HF system, you must perform the verification of identity of the online and offline program in RUN-Redundant mode. In so doing, it is sufficient to perform the verification only for one S7-1500 HF-CPU. *(S098)*

1. Connect online with the F-CPU. If multiple F-CPUs can be reached over a network (e.g. Industrial Ethernet) by the programming device / PC, you have to ensure that you are connected with the correct F-CPU. For example with "Online & diagnostics" > "Online accesses " > "Flash LED".
2. Open the *Safety Administration Editor*.
3. Check whether the online and offline Collective F-Signatures correspond to the Collective F-Signature from the safety summary.
4. Now check in the "General" area under "Safety program status" whether the safety programs are identical online and offline.

   Use the "Status" and "Version comparison" display to check which situation you are dealing with and, if necessary, execute the recommended measure:

Status	Version comparison	Statement	Measure
	Not relevant	The safety programs are different.	– Ensure that you are connected to the desired F-CPU. – Download the safety program to the F-CPU.
		The safety programs are identical but different versions of F-blocks are used.	The safety program must be downloaded to the F-CPU for the latest versions to become effective.
		The safety programs are identical.	None

5. If relevant:
   Check in the "General" area under "PROFIsafe" to see if the values for the F-BaseID match offline and online. See also warning S116.

Status	Meaning
	The online and offline F-BaseIDs match.
	The online and offline F-BaseIDs do *not* match.
	The F-BaseID could not be determined online or is not supported.

11.9 Other characteristics

Introduction

In addition, you must check a few more characteristics that are also relevant for the acceptance of the project.

Plausibility check for data transfer from the standard program to the safety program

Check to determine whether a plausibility check was programmed for all data transferred from the standard user program to the safety program. For this purpose, the "Data from the standard user program" section lists all tags of the standard user program that you are reading in the safety program. Tags of the standard user program that you are writing in the safety program are not listed here because a plausibility check is not required for them. For more information, refer to the S015 warning in section "Data Transfer from Standard User Program to Safety Program (Page 200)".

Multiple call of an F-FB or an instruction with the same instance (single instance or multi-instance)

Check whether you used a separate instance for each call of an F-FB or an instruction.

Multiple calls with the same instance are usually not worthwhile and require special care:

For example, if you explicitly transfer an operand (for example, as a constant) in the 1st call to the instance, and there is no explicit transfer in the 2nd call (so that the start value is valid), you must verify which value in the 2nd call is valid for this operand. Also consider startup processes in this regard; higher priority OBs or similar in which the 1st call may be present.

Checking the program version

Check whether the version of *STEP 7 Safety* used to create the summary (in the footer of the printout) is as least as high as the version used to compile the safety program. You can find the latter version in the section "General information" of the safety summary under "Used versions". Both versions must be listed in Annex 1 of the Report for the TÜV certificate.

Ability to disable safety mode

Make sure that safety mode cannot be disabled. For information about this, refer to section "General information" under "Safety program settings". This setting ensures that the safety mode of the safety program cannot be disabled inadvertently. For more information, refer to the S027 warning in section "Disabling safety mode (Page 393)".

Safety system version with an S7-1500 HF-CPU

Check whether the safety system version is V2.4 or higher when using an S7-1500 HF-CPU.

For information about this, refer to section "General information" under "Safety program settings".

Access protection

Check in the "General information" section under "Access protection" to determine whether the setting for access protection is permitted. Note the following warnings.

Otherwise, the project must not be accepted, because the safety program in the F-CPU is not protected against unauthorized accesses.

⚠ WARNING

(S7-300, S7-400) In productive operation, it must be prevented that changes to the standard user program are unintentionally made in the safety program as well. For this purpose, you must configure the protection level "Write protection for fail-safe blocks" and configure a password for the F-CPU.

With the "Write protection" or "Write/read protection" protection level, the password would apply to the standard and the safety program. *(S001)*

⚠ WARNING

(S7-1200, S7-1500) In productive operation, the safety-related project data must be protected against unintentional changes.

If you use user management, the Runtime rights "Change parameter of the F-System (F-Admin)" and "Full access including fail-safe" and the engineering right "Manage users and roles" may only be granted to authorized users. You must also limit the assignment of user rights to trained personnel. Before the transition into productive operation, you must remove the "Change parameter of the F-System (F-Admin)" and "Full access including fail-safe" rights from the "Anonymous" user.

If you are not using user management or the "Access control via access levels" parameter is activated, then configure at least the "Full access (no protection)" protection level and assign a password under "Full access incl. fail-safe (no protection)". This protection level allows full access only to the standard user program and not the safety program.

If you select a higher protection level, for example to protect the standard user program, you must assign an additional password for "Full access (no protection)".

Assign different passwords for the individual access levels. *(S041)*

F-Admin right of the web server

For S7-1200 G1 F-CPUs with firmware < V4.7 and S7-1500 F-CPUs with firmware < V3.1, the user management of the web server is part of the hardware configuration.

For S7-1500 F-CPUs as of firmware V3.1 and S7-1200 G1 as of firmware V4.7 or S7-1200 G2 as of firmware V1.0, this right is set in the Runtime rights of the respective F-CPU in the "Users and roles" area.

When downloading for the last time before the transition to productive operation, ensure that only authorized persons have the "F-admin" right for the web server.

After the download, the "F-Admin" right setting can only be checked online in the TIA Portal in the *Safety Administration Editor* (S7-1200 G1 F-CPUs with firmware < V4.7 and S7-1500 F-CPUs with firmware < V3.1) or in the "Security settings\Users and roles" tab (F-CPUs S7-1500 as of firmware V3.1 and S7-1200 G1 as of firmware V4.7 or S71200 G2 as of firmware V1.0).

⚠ WARNING

The "Change parameter of the F-System (F-Admin)" authorization for the web server (including web server API) without password protection (user "Everybody", or user "Anonymous" when using the user management) is intended only for testing, commissioning, etc. This means only when the system is not in productive operation. In this case, you must ensure the safety of the system through other organizational measures (Page 651).

Before the transition to productive operation, you must remove the rights "Change parameter of the F-System (F-Admin)" and "Full access including fail-safe" from the "Everybody" or "Anonymous" user.

The password of the web server user with "Change parameter of the F-System (F-Admin)" right may only be disclosed to authorized persons. After downloading the hardware configuration, check whether only permitted users of the web server have the "Change parameter of the F-System (F-Admin)" right on the F-CPU. To do so, use the online mode of the *Safety Administration Editor* for an F-CPU without user management or the "Security settings\Users and roles" tab for an F-CPU with user management.

Saving the login file and the password of the web server in the browser is only permitted when usage by unauthorized persons is prevented through other organizational measures (Page 651).

If you use the ticket mechanism, handling of the tickets must be just as restrictive as handling of the user password of the web server with the "Change parameter of the F-System (F-Admin)" right. *(S064)*

11.10 Acceptance of Changes

Introduction

In general, you can adopt the same approach for the acceptance of changes as the initial system acceptance (see Overview of System Acceptance (Page 413)).

You have to check the entire safety-related project data (Safety program and safety-related hardware configuration) for changes and determine the safety-related project data to be validated and approved as part of an impact analysis.

⚠ WARNING

In the case of acceptance of changes, you must check whether the intended changes were made correctly and completely.

You must also check whether unintentional changes may have been made at another location (for example, F-I/O or instructions that were added). *(S072)*

To avoid the acceptance of the entire system in case of negligible changes, *STEP 7 Safety* helps you to identify those parts of your Safety program that have changed.

To accept changes, proceed as follows:

1. Localize the changes in the safety-related project data:
   - changed or newly added F-blocks
   - changed or newly added instructions and F-system blocks
   - safety-related parameters of the changed or newly added F-I/O
   - structure of the safety-related hardware configuration (e.g. slot positions or start addresses of the F-I/Os)
   - changed safety-related communication via Flexible F-Link
   - changed safety-related communication via OPC UA Safety
2. From these changes, determine the safety-related project data affected by them (impact analysis). This can also be safety-related project data that has not been changed.
3. Then you carry out a validation and acceptance of the safety-related project data affected by the changes.

NOTE

Acceptance of changes is not possible after CPU migration.

⚠ WARNING

When you make changes in which the assignment of input/output addresses and wiring can change, then you must perform a wiring test (Page 397).

Examples for such changes are:

- Adding F-I/Os
- Changing the start address of F-I/Os
- Changing the slot position of F-I/Os
- Changing
  - the rack
  - the slave/device address
  - the PROFIBUS DP/PROFINET IO subnet
  - the IP address
  - the device name

(S071)

Localization of changes in the safety-related project data

To be able to localize relevant changes, you need two TIA projects:

- Reference project: Contains the initially accepted project data. They are the starting point for the upcoming comparison.
- Project to be accepted: Contains the current safety-related project data. It is the result of the reference project and the changes made in it.

To localize changes you have to compare the safety-related project data from the reference project with the safety-related data of the project to be accepted (see Comparing Safety Programs (Page 386)). Use filter setting "Compare only F-blocks relevant for certification". This limits the output of the comparison to the relevant F-blocks.

We recommend adding the summary (possibly as PDF) of the compare result to the acceptance documents.

The Collective F-Signature is a quick first step to determine whether relevant changes have been made. If the signature has changed, relevant changes are present in the safety-related project data.

(S7-1200, S7-1500) You can now use the Collective F-SW-Signature, the Collective F-HW-Signature and the F-Communication Address Signature to narrow down whether these changes are contained in the Safety program (Collective F-HW-Signature changed) and/or in the safety-related hardware configuration (Collective F-HW-Signature) and/or in the communication data (with Flexible F-Link, F-Communication Address Signature).

Bear in mind that the F-BaseID does not enter into any of the specified signatures and, if relevant, must be checked separately (see "Completeness and correctness of the hardware configuration (Page 418)").

Localization of changes in the Safety program

(S7-1200, S7-1500) A quick possibility to detect changes in the Safety program is the comparison of the Collective F-SW-Signature of the safety-related project data in the reference project with the Collective F-SW-Signature in the safety-related project data in the project to be accepted. If they differ from each other, this means that there are changes in the Safety program which need to be validated and, if necessary, accepted.

To localize changes in the Safety program, perform an offline-offline comparison between the Safety program to be accepted of the project to be accepted and the Safety program of the reference project (see Comparing Safety Programs (Page 386)).

⚠ WARNING

Make sure that the comparison criterion "Safety" is enabled and all other comparison criteria are disabled so that the criteria relevant for acceptance of changes are taken into consideration in the comparison. *(S069)*

If additional comparison criteria are enabled, you will see additional changes not relevant for the acceptance of changes (e.g. time stamp).

The status of the comparison helps you to identify which F-blocks were changed.

(S7-1200, S7-1500) As an alternative to the offline-offline comparison, you can localize the changes in the Safety program by comparing the group signatures. Use the safety summary of the reference project and of the project to be accepted for this purpose. Groups with an unchanged signature indicate that no F-blocks in them or lower-level groups have been changed. In groups with a changed group signature, you can localize the changed F-blocks using their changed block signature. Note that the assignment of F-blocks to groups is not recorded by the Collective F-Signature.

Changes to the maximum cycle time of an F-runtime group and F-cycle time warning limit of an F-runtime group are visible during comparison output through changes to the system block F-runtime group info DB.

The following changes do not influence the behavior of the Safety program directly and neither lead to a change of the Collective F-Signature nor are they shown during an offline-offline comparison. In the event of a change acceptance test, you should be able to identify these specifically using the safety summaries:

- Changes to the runtime for the deactivated safety mode
- Changes of the Safety System version
- Changes in the group structuring of the program blocks

Localization of changes in the safety-related hardware configuration

(S7-1200, S7-1500) A quick possibility to detect changes in the safety-related hardware configuration is the comparison of the Collective F-HW-Signature of the safety-related project data in the reference project with the Collective F-HW-Signature of the safety-related project data in the project to be accepted. If they differ from each other, this means that there are changes in the safety-related hardware configuration which need to be validated and, if necessary, accepted.

(S7-1200, S7-1500) If the Collective F-HW-Signature has changed and all F-IOs are unchanged, this indicates that safety-related parameters of the F-CPU have changed, or that the structure of the safety-related hardware configuration has changed, for example, slot positions.

Bear in mind that the F-BaseID does not enter into the Collective F-HW-Signature.

There are two possible ways for localizing safety-related changes in the safety-related hardware configuration:

- Localization by offline-offline comparison
- Localization by comparison of two safety summaries

The procedure is described below.

Localization by offline-offline comparison

The reference project and the project to be accepted must be consistent and compiled for a comparison. To perform the comparison, see Comparing Safety Programs (Page 386).

1. Navigate in the comparison result to the "System blocks > STEP 7 Safety > F-I/O DBs" folder. All data blocks listed in this folder are F-I/O-DBs and are each assigned to an F-I/O.
   - If the F-I/O-DBs in the comparison result are identical, this means that the safety-related configuration of the assigned F-I/O was also not changed. Standard-parameters might have changed.
   - If the F-I/O-DBs in the comparison result are not identical, this means that the safety-related configuration of the assigned F-I/O was also changed.
   - If F-IO DBs in the comparison result are marked as "not existing", associated F-IOs might have been deleted or added or the name or start addresses of the F-IOs have been changed. In this case, you can find the assignment of an F-I/O DB to a specific F-I/O in the safety summary under "Hardware configuration of F-I/O".
2. If you have found changed F-I/O, you can check the changed parameters in the safety summary as described below.

Note that this method is not suitable for identifying differences for the following parameters: HW ID, slot address, F-BaseID.

Localization by comparison of two safety summaries

Carry out a comparison based on two safety summaries as follows:

1. In the section "Hardware configuration of the F-I/Os" compare the start addresses (I/O addresses), the parameter "Behavior after channel fault" and the slot of the F-I/Os.
2. In the "Hardware configuration of F-I/O" section in the overview table of the F-I/O used, compare the parameter CRCs of the F-I/O with those in the safety summary of the corresponding F-CPU from the reference project.
   - If the "Parameter signature (without addresses)" is different for an F-I/O, this indicates the existence of a change in the safety-related parameters of the F-I/O.

     In this case, check the corresponding detail table of the safety-related parameters of the F-I/O and verify the uniqueness of the PROFIsafe addresses.
   - If the "Parameter signature (w/o addresses)" is identical, only the PROFIsafe addresses have been changed.

     In this case it is sufficient to verify the uniqueness of the PROFIsafe addresses.

Check as described in section "Completeness and correctness of the hardware configuration (Page 418)".

(S7-1200, S7-1500) Detecting changes in the communication with Flexible F-Link

A quick possibility to detect changes in the configuration of the communication with Flexible F-Link is the comparison of the F-communication address signature of the safety-related project data in the reference project with the F-communication address signature of the safety-related project data in the project to be accepted. If they differ from each other, this means that there are changes in the configuration of the communication (UUID only) with Flexible F-Link that must be validated and, if necessary, accepted. Other communication parameters such as timeout or transmission direction are covered by the Collective F-SW-Signature (see "Localization of changes in the Safety program" above).

There are two ways to localize safety-related changes in the communication with Flexible F-Link:

- Localization by offline-offline comparison
- Localization by comparison of two safety summaries

The procedure is described below.

Localization by offline-offline comparison

The reference project and the project to be accepted must be consistent and compiled for a comparison. For information on performing the comparison, see Comparing Safety Programs (Page 386).

1. Navigate in the comparison result to the "System blocks > STEP 7 Safety > Flexible F-Link" folder. All data blocks listed in this folder are F-communication DBs, and they are each assigned to a Flexible F-Link communication.
   - If F-communication DBs in the comparison result are identical, this means that the safety-related configuration of the assigned Flexible F-Link communication was not changed.
   - If F-communication DBs in the comparison result are different, this means that the safety-related configuration of the assigned Flexible F-Link communication was changed.
   - If F-communication DBs in the comparison result are marked as "not available", associated Flexible F-Link communications might have been deleted or added. In this case, you can find the assignment of an F-communication DB to a specific Flexible F-Link communication in the safety summary in the "Overview of communication via Flexible F-Link" table.
2. If you have identified a changed Flexible F-Link communication, you can check the changed parameters in the safety summary as described below.

Localization by comparison of two safety summaries

To localize changes in the configuration of the communication with Flexible F-Link, compare the section "Communications via Flexible F-Link Overview" of the reference project with that of the project to be accepted in the respective safety summary.

(S7-1500) Detecting changes in the communication with OPC UA Safety

There are two ways to localize safety-related changes in communication with OPC UA Safety:

- Localization by offline-offline comparison
- Localization by comparison of two safety summaries

The procedure is described below.

Localization by offline-offline comparison

The reference project and the project to be accepted must be consistent and compiled for a comparison. For information on performing the comparison, see Comparing Safety Programs (Page 386).

- For a comparison of SafetyConsumers, follow these steps:

  Navigate in the comparison result to the "System blocks > STEP 7 Safety > OPC UA Safety" folder. All data blocks listed in this folder are F-OPC-UA-Consumer-DBs, and they are each assigned to a SafetyConsumer. You can determine the specific SafetyConsumer based on the safety summary under "OPC UA Safety communication".

  – If F-OPC-UA-Consumer-DBs in the comparison result are identical, this means that the safety-related configuration of the associated SafetyConsumer was not changed.
  – If F-OPC-UA-Consumer-DBs in the comparison result are different, this means that the safety-related configuration of the associated SafetyConsumer was changed.
  – If F-OPC-UA-Consumer-DBs in the comparison result are marked as "not available", associated SafetyConsumers might have been deleted or added.

- You can perform the comparison of SafetyProviders using the F-OPC-UA-Provider-DBs. Follow these steps:

  Navigate in the comparison result to the "System blocks > STEP 7 Safety > Compiler blocks" folder. You can recognize the F-OPC-UA-Provider-DBs from the prefix "F_OPC_UA_PROVIDER_IDB_". Each F-OPC-UA-Provider-DB has been assigned to a SafetyProvider. You can determine the specific SafetyProvider based on the safety summary under "OPC UA Safety communication".

  – If F-OPC-UA-Provider-DBs in the comparison result are identical, this means that the safety-related configuration of the associated SafetyProvider was not changed.
  – If F-OPC-UA-Provider-DBs in the comparison result are different, this means that the safety-related configuration of the associated SafetyProvider was changed.
  – If F-OPC-UA-Provider-DBs in the comparison result are marked as "not available", associated SafetyProviders might have been deleted or added.

If you have identified a changed OPC UA Safety communication, you can check the changed parameters in the safety summary as described below.

Localization by comparison of two safety summaries

To localize changes in the configuration of the communication with OPC UA Safety, compare the detailed views of all SafetyProviders and all SafetyConsumers in the "Communications with OPC UA Safety Overview" area of the reference project with that of the project to be accepted in the respective safety summary.

See also

Accessing tags of the F-I/O DB (Page 172)

Operation and maintenance 12

12.1 Notes on startup of the F-system

In "STARTUP" mode of an F-CPU, the startup of the standard user program occurs as usual.

In the safety program, all DB contents of the F-DBs are generally reset to their start values from the load memory.

An automatic reintegration of the F-I/O (Page 172) takes place starting from the 2nd cycle of the F-runtime group.

NOTE

Resetting the F-DBs to start values from the load memory also triggers the following:

- Any stored error information is reset.
- Edge memory bits in instructions or F-FBs with edge evaluations (e.g. TON instruction) are reset, so that for a signal that already has the "TRUE" state in the 1st cycle of an F-runtime group, a positive signal edge is detected immediately at a query for positive signal edge.

12.2 Notes on Safety Mode of the Safety Program

Introduction

Pay attention to the following important notes on safety mode of the Safety program.

Use of simulation devices/simulation programs

⚠ WARNING

Use of simulation devices/simulation programs in plants

If you operate simulation devices or simulation programs that generate safety message frames, for example, based on PROFIsafe, and make them available to the SIMATIC Safety F-system via the bus system (such as PROFIBUS DP or PROFINET IO), you must ensure the safety of the plant through organizational measures (Page 651).
Note, for example, that a protocol analyzer is not permitted to perform any function that reproduces recorded message frame sequences with correct time behavior.

- **S7-PLCSIM version < 15.1 or S7-PLCSIM Advanced version < 2.0 SP1 and Safety System version < V2.2**

  If you use *S7-PLCSIM* (Page 391) to simulate Safety programs, the measures mentioned above are not necessary because S7-PLCSIM cannot establish an online connection to a real component.
- **S7-PLCSIM version ≥ 15.1 or S7-PLCSIM Advanced version ≥ 2.0 SP1 or Safety System version ≥ V2.2**

  You must ensure the safety of the plant through organizational measures.

  In addition, the loading of the safety-related project data with Safety System version V2.2 and higher onto an *S7-PLCSIM* is only permissible as of *S7-PLCSIM* V15.1 or *S7-PLCSIM Advanced* V2.0 SP1 and higher.

(S030)

NOTE

For an *S7-PLCSIM* before V15.1 or *S7-PLCSIM Advanced* before V2.0 SP1 and a Safety system version V2.2 and higher, the Safety program changes to STOP. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

Switching F-CPU to STOP mode

⚠ WARNING

Unintentional restart

In contrast to the standard user program, when the Safety program starts up, all tags of the F-DBs are generally initialized with their start values. This means that saved error information is lost. The F-system performs automatic reintegration of the F-I/O.

If an unintentional startup of your process is not allowed, you must program restart/startup protection in the Safety program. For this, the output of process values must be blocked until user acknowledgment takes place (see "Implementation of user acknowledgment (Page 190)"). This user acknowledgment must not take place until safe output of process values is possible (see "Programming startup protection (Page 156)").

You must expect an unintentional restart in the following cases:

- After a STOP generated by PG/PC operation, mode switch, communication function or "STP" instruction
- After an operating error
- After a STOP caused by a fault reaction function
- The following applies to the S7-1500 F Virtual Controller:

  After a STOP or a restart of the S7-1500 F Virtual Controller via Industrial Edge Management.

(S008)

CRC error in F-I/O or safety-related communication

If you observe that a manual acknowledgment is requested more than once every 100 hours due to a CRC error and this occurs repeatedly, this indicates that the underlying communication infrastructure (cabling, wireless connections, network components, etc.) does not have the required reliability.

CRC errors are present:

- For F-I/O access, the ACK_REQ tag of the F-I/O DB is set and the DIAG tag of the F-I/O DB indicates CRC errors with bit 2 or bit 6.
- For communication with the SENDDP/RCVDP instructions, the ACK_REQ output of the RCVDP instruction is set and the DIAG output of the SENDDP/RCVDP instruction indicates CRC errors with bit 6.
- For communication with the SENDS7/RCVS7 instructions, the ACK_REQ output of the RCVS7 instruction is set and the DIAG output of the SENDS7/RCVS7 instruction indicates CRC errors with bit 6.
- For communication with Flexible F-Link, the ACK_REQ tag of the F-communication DB is set for receiving and the DIAG tag of the F-communication DB for sending/receiving indicates CRC errors with bit 6.

  or

- A CRC error is entered in the diagnostic buffer of the F-CPU.

In this case, the failure probability values (Page 446) (PFD_{avg}/PFH) for safety-related communication no longer apply.

Errors in safety-related communication with OPC UA Safety

The OPC UA Safety communication protocol contains a mechanism that tolerates a communication error within a configurable period of time (see description SafetyErrorIntervalLimit (Page 295) in the SPI of the SafetyConsumer).

If you observe that, despite this mechanism, a manual acknowledgment is requested more than once within 24 hours due to a communication error, this indicates that the underlying communication infrastructure (cabling, wireless connections, network components, etc.) does not have the required reliability.

You can recognize communication errors by the fact that the FSV_ACTIVATED output of an F-OPC UA Consumer DB is set and the DIAG output is set according to the type of error.

In this case, the failure probability values (Page 446) (PFDavg/PFH) for safety-related communication no longer apply.

12.3 Replacing Software and Hardware Components

Introduction

Proceed as in the standard. Also note the following sections.

Replacement of software components

When replacing software components on your programming device or PC, e.g. with a new version of *STEP 7*, you must adhere to the information regarding upward and downward compatibility in the documentation and readme files for these products (e.g. *STEP 7 Safety*).

When replacing *STEP 7 Safety*, check whether the version of *STEP 7 Safety* is listed in Annex 1 of the Report for the TÜV certificate.

Replacement of hardware components

Hardware components for SIMATIC Safety (F-CPU, F-I/O, batteries, etc.) are replaced in the same way as in Standard automation systems.

If you replace an F-I/O in the TIA Portal with an F-I/O with a newer article number that occupies more I/O addresses, the now larger I/O address range may already be occupied by another F-I/O.

For example, if you replace an F module F-DI 8x24VDC HF with part number 6ES7136-6BA00-0CA0 by an F module with 6ES7136-6BA01-0CA0. The end address of the input and output addresses is increased by 1 byte each by the exchange. In this case, *STEP 7* assigns a new free I/O address range to the F module. The corresponding tag in the tag table must then be manually adapted to the new I/O addresses.

Replacement of an HF-CPU in the redundant S7-1500 HF system

Note the following warning when replacing an HF-CPU of a redundant S7-1500HF system:

WARNING

If you replace an F-CPU of a redundant S7-1500HF system (e.g. due to CPU failure, defect, etc.), you must put into operation the replacement F-CPU with the SIMATIC Memory Card of the previous F-CPU or with an empty SIMATIC Memory Card. This ensures that after successful SYNCUP of the replacement F-CPU, only the previous safety program is executed. *(S094)*

Replacement of S7-1500 F Software Controllers

WARNING

The following applies to firmware versions V30.0 and higher and < V40.0 of the S7-1500 F Software Controller:

After a CPU module replacement (for example, new PC with data storage medium from old PC), a data storage medium replacement (for example, data storage medium with safety-related project data 1 is replaced by data storage medium with safety-related project data 2) or a UEFI update (if no NVRAM is present), you must ensure that the correct safety program is located in the S7-1500 F Software Controller by reading out the Collective F-signature with the command line command "GetCollectiveFSignature" or the display.

Then you must enter and confirm the expected Collective F-signature with the command line command "ConfirmCollectiveFSignature". This confirmation is required to restart the safety program.

For both command line commands, you must be a member of the "Software Controller Operators" (Windows) or "software_controller_operators" (Linux) user group and the "Failsafe Operators" (Windows) or "failsafe_operators" (Linux) user group.

The following applies to firmware versions < V30.0 of the S7-1500 F Software Controller:

After a CPU module replacement (e.g. new PC with data storage medium from old PC), a data storage medium replacement (e.g. data storage medium containing safety program 1 is replaced with data storage medium containing safety program 2) or a UEFI update, you must check via the display whether the correct collective F-signature is displayed or perform a program identification. *(S066)*

Removing and inserting F-I/O during operation

If removing and inserting is possible for standard I/O during operation, it is also possible for the respective F-I/O. However, be aware that replacing an F-I/O module during operation can cause a communication error in the F-CPU.

You must acknowledge the communication error in your safety program in the ACK_REI tag of the F-I/O DB (Page 164) or, alternatively, by using the "ACK_GL (Page 530)" instruction. Without an acknowledgment, the F-I/O will remain passivated.

CPU firmware update

Check of the CPU operating system for F-approval: When using a new CPU operating system (firmware update), you must check to see if the CPU operating system you are using is approved for use in an F-system.

The minimum CPU operating system versions with guaranteed F-capability are specified in the appendix of the Certificate. This information and any notes on the new CPU operating system must be taken into account.

Firmware update for interface module

When using a new operating system for an interface module, such as IM 151-1 HIGH FEATURE for ET 200S (firmware update), you must observe the following:

If you have selected the "Activate firmware after update" option for the firmware update (see *Help on STEP 7*, "Online & Diagnostics"), the interface module is automatically reset following a successful download operation and then runs on the new operating system. Note that the firmware update for interface modules during operation causes a communication error in the F-CPU.

You must acknowledge the communication error in your safety program in the ACK_REI tag of the F-I/O DB (Page 164) or, alternatively, by using the "ACK_GL (Page 530)" instruction. Without an acknowledgment, the F-I/O will remain passivated.

Preventive maintenance (proof test)

Proof test for complex electronic components generally means replacement with new, unused components.

12.4 PFDavg and PFH values

PFD_{avg} and PFH values for S7-300/400 F-CPUs and F-I/O

You will find a list of the failure probability values (PFD_{avg}, PFH values) for components that can be used in SIMATIC Safety on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109481784).

PFD_{avg} and PFH values for S7-1200/1500 F-CPUs

Below are the probability of failure values (PFD_{avg}, PFH values) for S7-1200/1500 F-CPUs with a service life of 20 years and a mission time of 100 hours:

Operation in low demand mode low demand mode According to IEC 61508:2010: PFD_{avg} = Average probability of dangerous failure on demand	**Operation in high demand or continuous mode high demand/continuous mode According to IEC 61508:2010: PFH = Average frequency of a dangerous failure [h^{-1}]**
< 2E-05	< 1E-09 up to an operating altitude of 3 000 m or < 2E-09 at an operating altitude above 3 000 m and up to 5 000 m

More information can be found on the operating conditions of the F-CPUs in the PI F-CPUs (https://support.industry.siemens.com/cs/ww/en/view/109478599) and in the PI of the PFD_{avg}, PFH values (https://support.industry.siemens.com/cs/ww/en/view/109481784).

PFD_{avg} and PFH values for safety-related communication without OPC UA Safety

Below you will find the failure probability values (PFD_{avg}, PFH values) for safety-related communication:

Operation in low demand mode low demand mode According to IEC 61508:2010: PFD_{avg} = Average probability of dangerous failure on demand	**Operation in high demand or continuous mode high demand/continuous mode According to IEC 61508:2010: PFH = Average frequency of a dangerous failure [h^{-1}]**
< 1E-05*	< 1E-09*

*** Note on S7-300/400 F-CPUs:**
The PFH value is valid under the assumption that a maximum of 100 F-I/Os are involved in a safety function. If you use more than 100 F-I/Os, you have to also add 4E-12 per F-I/O for the safety function.
The PFD_{avg} value is valid for a mission time of 20 years and under the assumption that a maximum of 25 F-I/Os are involved in a safety function. If more than 25 F-I/Os are used, you need to add 3.5E-7 per F-I/O for this safety function.

More information can be found on the operating conditions of the F-CPUs in the PI of the PFD_{avg} and PFH values (https://support.industry.siemens.com/cs/ww/en/view/109478599).

PFD_{avg} and PFH values for safety-related communication with OPC UA Safety (S7-1500)

Below you will find the probability of failure values (PFD_{avg} and PFH values) for safety-related communication via OPC UA Safety:

Operation in low demand mode low demand mode According to IEC 61508:2010: PFD_{avg} = Average probability of dangerous failure on demand	**Operation in high demand or continuous mode high demand/continuous mode According to IEC 61508:2010: PFH = Average frequency of a dangerous failure [h^{-1}]**
< 8E-08*	< 4E-11*

*** Note**:
The values listed in the table above apply when configuring the SafetyErrorIntervalLimit (Page 295) SPI parameter of a SafetyConsumer with the value 600.
In exceptional cases, you can consider a lower setting (60 or 6). A lower setting may reduce the risk of false tripping. A lower setting is appropriate if the unnecessary triggering of the safety function or the output of fail-safe substitute values would have particularly serious consequences. However, this also worsens the PFD_{avg}/PFH values as follows:

SafetyErrorIntervalLimit	PFD_{avg} value	PFH value
60	< 2.5E-07	< 4E-10
6 (not permitted for SIL3)	< 1E-06	< 4E-09

12.5 Guide to Diagnostics

Introduction

Here you find a compilation of diagnostic capabilities that can be evaluated for your system when an error occurs. Most of the diagnostic capabilities are the same as those in standard automation systems. The sequence of steps represents a recommendation.

Steps for evaluating diagnostic capabilities

The following table shows the steps you take to evaluate diagnostic capabilities.

Step	Procedure	Reference
1	**Evaluate LEDs on the hardware (F-CPU, F-I/O):** • BUSF LED on the F-CPU: Flashes when a communication error occurs on PROFIBUS DP/PROFINET IO; On if a programming error occurs when OB 85 and OB 121 are programmed (e.g. instance DB is not loaded) • STOP LED on the F-CPU: illuminates when the F-CPU is in STOP mode • Fault LEDs on the F-I/O: e.g. SF-LED (group error LED) on if any fault occurs in the individual F-I/O	*Manuals for F-CPU and F-I/O*
2	**Evaluate diagnostic buffer of the modules:** You read the diagnostic buffer of a module (F-CPU, F-I/O, CP) in its online and diagnostic view in the "Diagnostic buffer" group under the "Online & Diagnostics" folder.	*Help on STEP 7 and manuals for the F-CPU and F-I/O*
3	**Evaluate stacks of the F-CPU:** when the F-CPU is in STOP mode, read the following successively: • Block stack: Check whether STOP mode of the F-CPU was triggered by an F-block of the Safety program • Interruption stack • Local data stack	*STEP 7 help, Diagnostics Function Manual, system manuals*
4	**Evaluate diagnostic tag of the F-I/O DB using testing and commissioning functions, by means of an operator control and monitoring system, or in the standard user program:** Evaluate the DIAG tag in the F-I/O DB	DIAG (Page 171)

Step	Procedure	Reference
5	**Evaluate diagnostic tag of the F-OPC UA Consumer DB using testing and commissioning functions, by means of an operator control and monitoring system, or in the standard user program:** in the F-OPC UA Consumer DB of the variable DIAG	Other variables in the F-OPC UA Consumer DB (Page 304) OPC UA access to SafetyProvider and SafetyConsumer information models (Page 335)
6	**Evaluate diagnostic tag of the F-communication DB using testing and commissioning functions, by means of an operator control and monitoring system, or in the standard user program:** Evaluate the DIAG tag in the F-communication DB	F-communication DB (S7-1200, S7-1500) (Page 346)
7	**Evaluate diagnostic outputs of the instance DBs of instructions using testing and commissioning functions, using an operator control and monitoring system, or in the standard user program:** • Evaluate the following for MUTING, EV1oo2DI, TWO_H_EN, MUT_P, ESTOP1, FDBACK, SFDOOR in the assigned instance DB: – Output DIAG • Evaluate the following for SENDDP or RCVDP in the assigned instance DB: – Output RET_DPRD/RET_DPWR – Output DIAG • Evaluate the following for SENDS7 or RCVS7 in the assigned instance DB: – Output STAT_RCV – Output STAT_SND – Output DIAG	Instructions

Tip on RET_DPRD/RET_DPWR

The diagnostic information of the RET_DPRD/RET_DPWR outputs of the SENDDP or RCVDP instructions corresponds to the diagnostic information of the RETVAL return value of the "DPRD_DAT" and "DPWR_DAT" instructions. You can find the description in the help on *STEP 7* for the "DPRD_DAT" and "DPWR_DAT" instructions.

Tip: STAT_RCV and STAT_SND

The diagnostic information of the STAT_RCV output of the SENDS7 or RCVS7 instructions corresponds to the diagnostic information of the STATUS output of the "URCV" instruction. The diagnostic information of the STAT_SND output of the SENDS7 or RCVS7 instructions corresponds to the diagnostic information of the STATUS output of the "USEND" instruction. You can find the description in the help on *STEP 7* for the instruction "UCRV" or "USEND" .

STEP 7 Safety V21 instructions

13

Overview of instructions for the safety program

When programming an F-block, you can find all instructions available for programming an F-block in LAD or FBD with the configured F-CPU in the "Instructions" task card.

In addition to the instructions that are familiar to you from programming a standard block, there are also special safety functions, e.g., for two-hand monitoring, discrepancy analysis, muting, emergency STOP/emergency OFF, safety door monitoring, and feedback monitoring and instructions for safety-related CPU-CPU communication.

13.1 Connection of the Enable input EN and Enable output ENO

The enable input EN cannot be connected.

(S7 -300, S7-400) It is not possible to connect the ENO output.

(S7-1200, S7-1500) The ENO output can only be wired with the following instructions:

- ADD: Add (Page 553)
- SUB: Subtract (Page 556)
- MUL: Multiply (Page 559)
- DIV: Divide (Page 562)
- NEG: Create two's complement (Page 565)
- ABS: Form absolute value (Page 568)
- CONVERT: Convert value (Page 578)

13.2 Temporal blurring for instructions with time processing

Observe the following warning for the following instructions with time processing:

- ESTOP1: Emergency STOP/emergency OFF up to stop category 1 (STEP 7 Safety V21) (Page 486)
- TWO_HAND: Two-hand monitoring (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 490)
- TWO_H_EN: Two-hand monitoring with enable (STEP 7 Safety V21) (Page 492)
- MUTING: Muting (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 495)
- MUT_P: Parallel muting (STEP 7 Safety V21) (Page 504)
- EV1oo2DI: 1oo2 evaluation with discrepancy analysis (STEP 7 Safety V21) (Page 515)
- FDBACK: Feedback monitoring (STEP 7 Safety V21) (Page 520)
- TP: Generate pulse (STEP 7 Safety V21) (Page 532)
- TON: Generate on-delay (STEP 7 Safety V21) (Page 535)

- TOF: Generate off-delay (STEP 7 Safety V21) (Page 537)
- ACK_OP: Fail-safe acknowledgment (STEP 7 Safety V21) (Page 604)

WARNING

When using an instruction with time processing, take the following timing imprecision sources into account when determining your response times:

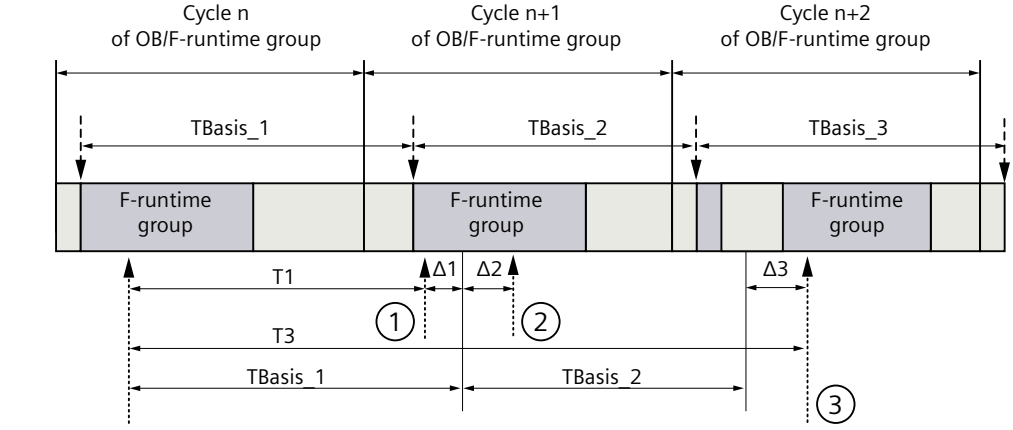
- The temporal blurring known from the standard user program, which is caused by the cyclic processing
- Timing imprecision resulting from the update time of the time base used in the instruction (see figure in section "Timing imprecision resulting from the update time of the time base used in the instruction")
- Tolerance of internal time monitoring in the F-CPU
  - For time values up to 200 ms, a maximum of 4 ms
  - For time values greater than or equal to 200 ms, a maximum of 2% of the (assigned) time value
- Tolerance of internal time monitoring in the S7-1500 HF-CPU
  - For time values up to 500 ms, a maximum of 10 ms
  - For time values greater than or equal to 500 ms, a maximum of 2% of the (assigned) time value
- Tolerance of internal time monitoring in the S7-1500 F Virtual Controller

  Without setup of Virtual Time Service:
  - For time values up to 200 ms, a maximum of 5 ms
  - For time values greater than or equal to 200 ms, a maximum of 2% of the (assigned) time value plus 1 ms

  With setup of Virtual Time Service:
  - For time values up to 200 ms, a maximum of 10 ms
  - For time values greater than or equal to 200 ms, a maximum of 2% of the (assigned) time value plus 6 ms

You must choose the interval between two calls of an instruction with time processing in such a way that the required response times are achieved, taking into account the possible timing imprecision. *(S034)*

Timing imprecision resulting from the update time of the time base used in the instruction:



= Update of the time basis TBasis_n

= Productive operation of an instruction with time processing

① For the first call in cycle n+1, the call time of the instruction relative to the start of the F-runtime group is earlier than that in cycle n by the amount of Δ_1, e.g. because parts of the Safety program of the F-runtime group before the call time of the instruction in cycle n+1 are skipped. For the time update, the instruction takes account of time T_{Base_1} instead of the time T_1 that has actually elapsed in cycle n since the call.

② The instruction is called a second time in cycle n+1. This does not involve another time update (by Δ_2).

③ For the call in cycle n+2, the call time of the instruction relative to the start of the F-runtime group is later than that in cycle n by the amount of Δ_3, e.g. because the F-runtime group was interrupted by a higher priority interrupt before the call time of the instruction in cycle n+2. The instruction took into account time $T_{Base_1} + T_{Base_2}$ instead of the time T_3 that has actually elapsed in cycle n since the call. This would also be the case if no call occurred in cycle n+1.

13.3 General

13.3.1 LAD

13.3.1.1 New network (STEP 7 Safety V21)

Requirement

An F-block is open.

Procedure

To insert a new network, follow these steps:

1. Select the network after which you want to insert a new network.
2. Select the "Insert network" command in the shortcut menu.

NOTE

If you insert an element into the last empty network of the F-block in an LAD program, a new empty network is automatically inserted below it.

Result

A new empty network is inserted into the F-block.

13.3.1.2 Empty box (STEP 7 Safety V21)

Requirement

A network is available.

Procedure

To insert an LAD instruction into a network using an empty box, follow these steps:

1. Open the "Instructions" task card.
2. Navigate to "Basic instructions > General > Empty box".
3. Use a drag-and-drop operation to move the "Empty box" element to the desired place in the network.
4. Hover the cursor over the yellow triangle in the top right corner of the empty box.

   A drop-down list is displayed.
5. Select the required instruction from the drop-down list.

If the instruction acts as a function block (FB) within the system, the "Call options" dialog opens. In this dialog, you can create an instance data block for the function block, either as a single instance or, if necessary, multi-instance, in which data of the inserted instruction are stored. Once it is created, the new instance data block can be found in the "Program resources" folder in the project tree under "Program blocks > System blocks". If you have selected "multi-instance", you can find it in the block interface in the "Static" section.

Result

The empty box is changed to the appropriate instruction. Placeholders are inserted for the parameters.

13.3.1.3 Open branch (STEP 7 Safety V21)

Description

Use branches to program parallel connections with the Ladder Logic (LAD) programming language. Branches are inserted into the main current path. You can insert several contacts into the branch, thereby creating a parallel connection from series connections. You can program complex ladder diagrams in this way.

Requirement

- A network is available.
- The network contains elements.

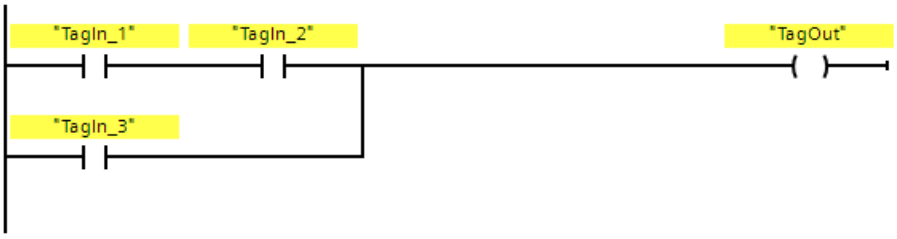
Procedure

To insert a new branch in a network, follow these steps:

1. Open the "Instructions" task card.
2. Navigate to "Basic instructions > General > Open branch".
3. Use a drag-and-drop operation to move the element to the desired place in the network.
4. If you want to connect the new branch directly to the power rail, drag the element to the power rail.

Example

The following figure provides an example of how to use branches:



13.3.1.4 Close branch (STEP 7 Safety V21)

Description

Branches must be closed again at suitable places. If necessary, branches will be arranged so that they do not cross each other.

Requirement

A branch is available.

Procedure

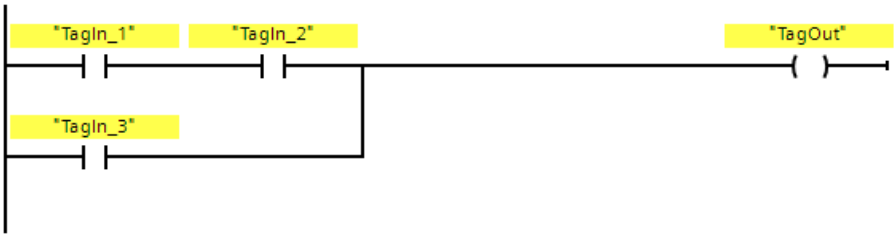
To close an open branch, follow these steps:

1. Select the open branch.
2. Press and hold down the left mouse button.

   A dashed line will appear as soon as the cursor is moved.
3. Drag the dashed line to a suitable place on the network. Permissible connections are indicated by green lines.
4. Release the left mouse button.

Example

The figure below provides an example of how to use branches:



13.3.2 FBD

13.3.2.1 New network (STEP 7 Safety V21)

Requirement

An F-block is open.

Procedure

To insert a new network, follow these steps:

1. Select the network after which you want to insert a new network.
2. Select the "Insert network" command in the shortcut menu.

NOTE

If you insert an element into the last empty network of the F-block in an FBD program, a new empty network is automatically inserted below it.

Result

A new empty network is inserted into the F-block.

13.3.2.2 Empty box (STEP 7 Safety V21)

Requirement

A network is available.

Procedure

To insert FBD elements into a network using an empty box, follow these steps:

1. Open the "Instructions" task card.
2. Navigate to "Basic instructions > General > Empty box".
3. Use a drag-and-drop operation to move the "Empty box" element to the desired place in the network.
4. Hover the cursor over the yellow triangle in the top right corner of the empty box.

   A drop-down list is displayed.
5. Select the desired FBD element from the drop-down list.

If the instruction acts as a function block (FB) within the system, the "Call options" dialog opens. In this dialog, you can create an instance data block for the function block, either as a single instance or, if necessary, multi-instance, in which data of the inserted instruction are stored. Once it is created, the new instance data block can be found in the "Program resources" folder in the project tree under "Program blocks > System blocks". If you have selected "multi-instance", you can find it in the block interface in the "Static" section.

Result

The empty box is changed to the appropriate instruction. Placeholders are inserted for the parameters.

13.3.2.3 Open branch (STEP 7 Safety V21)

Description

You use branches, which you insert between the boxes, to program parallel connections with the Function Block Diagram (FBD) programming language You can insert additional boxes in the branch, thereby programming complex function block diagrams.

Requirement

A network is available.

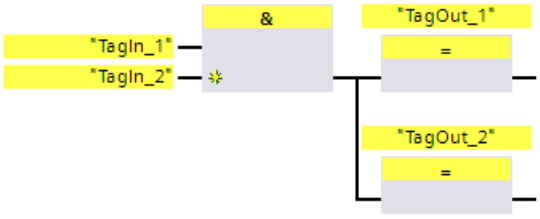
Procedure

To insert a new branch in a network, follow these steps:

1. Open the "Instructions" task card.
2. Navigate in the pane to "Basic instructions > General > Branch".
3. Use a drag-and-drop operation to move the element to the desired place to a connecting line between two boxes.

Example

The following figure provides an example of how to use branches:



13.3.2.4 Insert binary input (STEP 7 Safety V21)

Description

Use the "Insert binary input" instruction to expand the box of one of the following instructions by a binary input:

- "AND logic operation"
- "OR logic operation"
- "EXCLUSIVE OR logic operation"

You can query the signal state of several operands by expanding the instruction box.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand>	Input	BOOL	The operand indicates the bit whose signal state will be queried.

Example

The following example shows how the instruction works:



The box of instruction "AND logic operation" has been expanded by an additional binary input at which the signal state of operand "TagIn_3" is queried. Output "TagOut" is set when the signal state of operands "TagIn_1", "TagIn_2", and "TagIn_3" is "1".

See also

AND logic operation (STEP 7 Safety V21) (Page 472)

OR logic operation (STEP 7 Safety V21) (Page 473)

X: EXCLUSIVE OR logic operation (STEP 7 Safety V21) (Page 474)

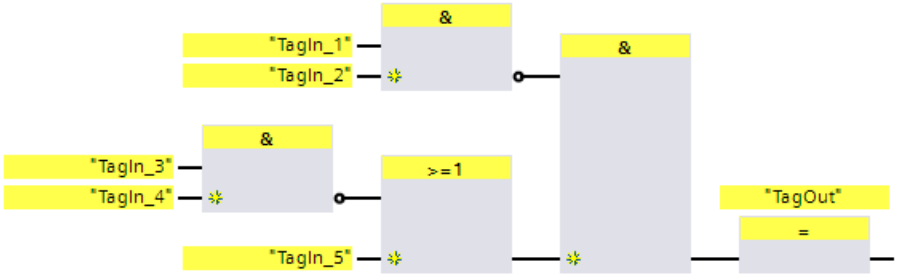
13.3.2.5 Invert RLO (STEP 7 Safety V21)

Description

You can use the "Invert RLO" instruction to invert the result of logic operation (RLO).

Example

The following example shows how the instruction works:



Output "TagOut" is set when the following conditions are fulfilled:

- The "TagIn_1" and/or "TagIn_2" input has the signal state "0".
- The "TagIn_3" or "TagIn_4" input has the signal state "0" or the "TagIn_5" input has the signal state "1".

13.4 Bit logic operations

13.4.1 LAD

13.4.1.1 ---| |---: Normally open contact (STEP 7 Safety V21)

Description

The activation of the normally open contact depends on the signal state of the associated operand. If the operand has signal state "1," the normally open contact is closed. Power flows from the left power rail through the normally open contact into the right power rail and the signal state at the output of the instruction is set to "1".

If the operand has signal state "0," the normally open contact is not activated. The power flow to the right power rail is interrupted and the signal state at the output of the instruction is reset to "0".

Two or more normally open contacts are linked bit-by-bit by AND when connected in series. With a series connection, power flows when all contacts are closed.

The normally open contacts are linked by OR when connected in parallel. With a parallel connection, power flows when one of the contacts is closed.

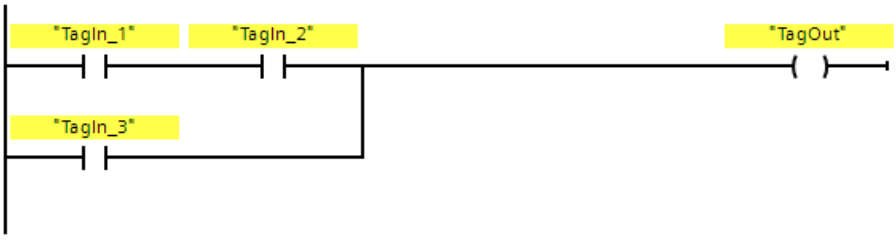
Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand>	Input	BOOL	Operand whose signal state is queried.

Example

The following example shows how the instruction works:



Operand "TagOut" is set when one of the following conditions is fulfilled:

- Operands "TagIn_1" and "TagIn_2" have signal state "1".
- The signal state at operand "TagIn_3" is "1".

13.4.1.2 ---| / |---: Normally closed contact (STEP 7 Safety V21)

Description

The activation of the normally closed contact depends on the signal state of the associated operand. If the operand has signal state "1", the normally closed contact is opened and the signal state at the output of the instruction is reset to "0".

If the operand has signal state "0", the normally closed contact is not activated and the signal state at the output of the instruction is set to "1".

Two or more normally closed contacts are linked bit-by-bit by AND when connected in series. With a series connection, power flows when all contacts are closed.

The normally closed contacts are linked by OR when connected in parallel. With a parallel connection, power flows when one of the contacts is closed.

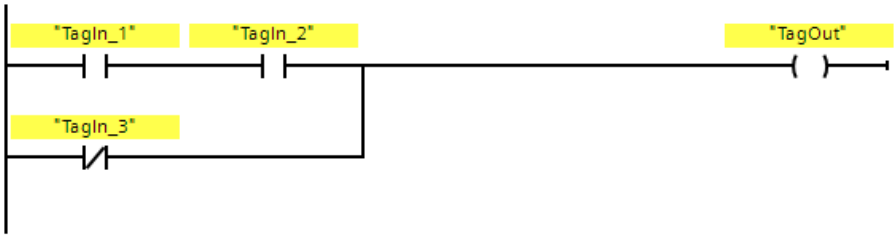
Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand>	Input	BOOL	Operand whose signal state is queried.

Example

The following example shows how the instruction works:



Operand "TagOut" is set when one of the following conditions is fulfilled:

- Operands "TagIn_1" and "TagIn_2" have signal state "1".
- The signal state at operand "TagIn_3" is "0".

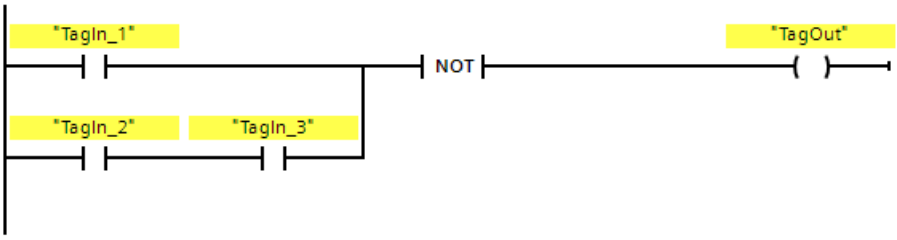
13.4.1.3 --|NOT|--: Invert RLO (STEP 7 Safety V21)

Description

You can use the "Invert RLO" instruction to invert the signal state of the result of logic operation (RLO). When the signal state is "1" at the input of the instruction, the output of the instruction has the signal state "0". When the signal state is "0" at the input of the instruction, the output has the signal state "1".

Example

The following example shows how the instruction works:



Operand "TagOut" is reset when one of the following conditions is fulfilled:

- Operand "TagIn_1" has signal state "1".
- Operands "TagIn_2" and "TagIn_3" have signal state "1".

13.4.1.4 ---( )---: Assignment (STEP 7 Safety V21)

Description

You can use the "Assignment" instruction to set the bit of a specified operand. When the result of logic operation (RLO) at the input of the coil is "1," the specified operand is set to signal state "1". When the signal state is "0" at the input of the coil, the bit of the specified operand is reset to "0".

The instruction does not influence the RLO. The RLO at the input of the coil is sent immediately to the output.

The "Assignment" instruction can be placed at any position in the network.

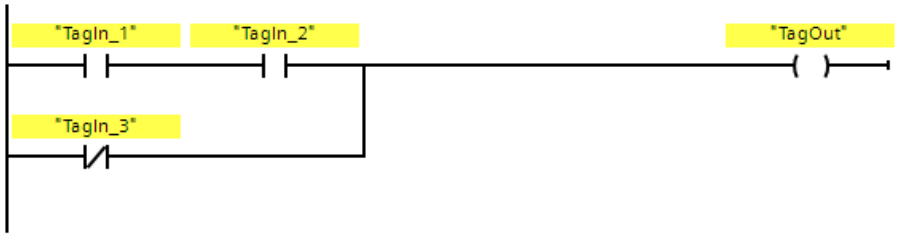
Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand>	Output	BOOL	Operand to which the RLO is assigned.

Example

The following example shows how the instruction works:



Operand "TagOut" is set when one of the following conditions is fulfilled:

- Operands "TagIn_1" and "TagIn_2" have signal state "1".
- The signal state at operand "TagIn_3" is "0".

13.4.1.5 ---(R)---: Reset output (STEP 7 Safety V21)

Description

You can use the "Reset output" instruction to reset the signal state of a specified operand to "0".

If power flows to the coil (RLO is "1"), the specified operand is set to "0". If the result of logic operation at the input of the coil is "0" (no signal flow to the coil), the signal state of the specified operand remains unchanged.

The instruction does not influence the RLO. The RLO at the input of the coil is sent directly to the output of the coil.

NOTE

If you want to use a formal parameter of an F-FC for the operand of the instruction, it must be declared as an InOut parameter.

NOTE

If the operand area "local data (temp)" is used for the operands of the instruction, the local data bit used must be initialized beforehand.

NOTE

You cannot use the "process image of the inputs", "process image of the outputs" from standard I/O and "standard DB" and "bit memory" operand areas for the operands of the instruction.

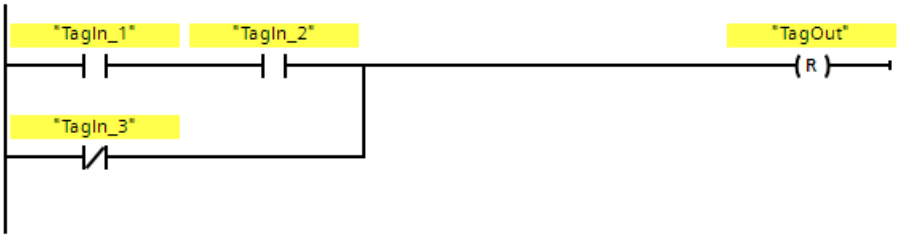
Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand>	Output	BOOL	Operand that is reset when RLO = "1".

Example

The following example shows how the instruction works:



Operand "TagOut" is reset when one of the following conditions is fulfilled:

- Operands "TagIn_1" and "TagIn_2" have signal state "1".
- The signal state of operand "TagIn_3" is "0".

13.4.1.6 ---(S)---: Set output (STEP 7 Safety V21)

Description

You can use the "Set output" instruction to set the signal state of a specified operand to "1".

If power flows to the coil (RLO is "1"), the specified operand is set to "1". If the result of logic operation at the input of the coil is "0" (no signal flow to the coil), the signal state of the specified operand remains unchanged.

The instruction does not influence the RLO. The RLO at the input of the coil is sent directly to the output of the coil.

NOTE

The instruction is not executed if it is applied to an output of an F-I/O that is passivated (e.g., during startup of the F-system). Therefore, it is preferable to access outputs of the F-I/O using only the "Assignment" instruction.

An F-I/O output is passivated if QBAD or QBAD_O_xx = 1 or value status = 0 is set in the corresponding F-I/O DB.

NOTE

If you want to use a formal parameter of an F-FC for the operand of the instruction, it must be declared as an InOut parameter.

NOTE

If the operand area "local data (temp)" is used for the operands of the instruction, the local data bit used must be initialized beforehand.

NOTE

You cannot use the "process image of the inputs", "process image of the outputs" from standard I/O and "standard DB" and "bit memory" operand areas for the operands of the instruction.

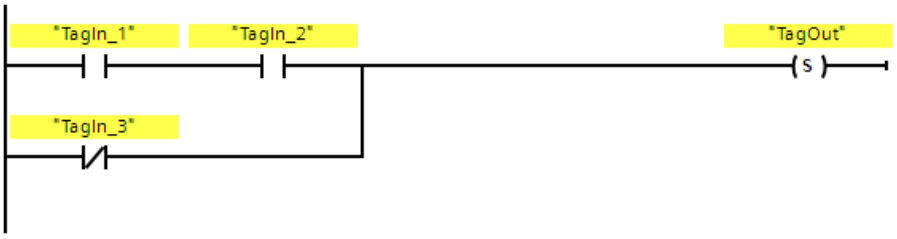
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand>	Output	BOOL	Operand that is set when RLO = "1".

Example

The following example shows how the instruction works:



Operand "TagOut" is set when one of the following conditions is fulfilled:

- Operands "TagIn_1" and "TagIn_2" have signal state "1".
- The signal state of operand "TagIn_3" is "0".

13.4.1.7 SR: Set/reset flip-flop (STEP 7 Safety V21)

Description

You can use the "Set/reset flip-flop" instruction to set or reset the bit of the specified operand based on the signal state of inputs S and R1. If the signal state at input S is "1" and the signal state at input R1 is "0", the specified operand is set to "1". If the signal state at input S is "0" and the signal state at input R1 is "1", the specified operand is reset to "0".

Input R1 takes priority over input S. If the signal state is "1" at the two inputs S and R1, the signal state of the specified operand is reset to "0".

The instruction is not executed if the signal state at the two inputs S and R1 is "0". The signal state of the operand then remains unchanged.

The current signal state of the operand is transferred to output Q and can be queried there.

NOTE

If you want to use a formal parameter of an F-FC for the operand of the instruction, it must be declared as an InOut parameter.

NOTE

You cannot use the "process image", "standard DB" and "bit memory" operand areas for the operands of the instruction.

If the operand area "local data (temp)" is used for the operands of the instruction, the local data bit used must be initialized beforehand.

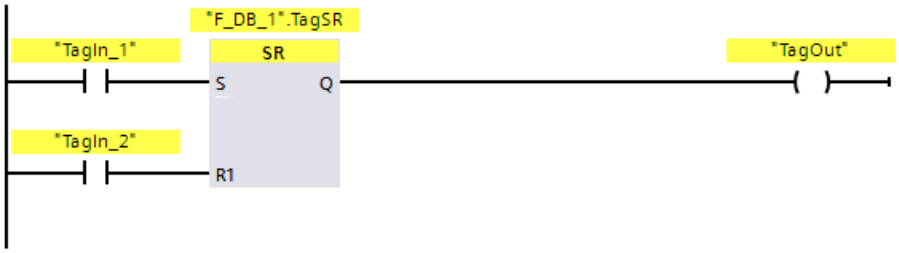
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
S	Input	BOOL	Enable set
R1	Input	BOOL	Enable reset
<Operand>	Output	BOOL	Operand that is set or reset.
Q	Output	BOOL	Signal state of the operand

Example

The following example shows how the instruction works:



Operands ""F_DB_1".TagSR" and "TagOut" are set when the following conditions are fulfilled:

- Operand "TagIn_1" has signal state "1".
- Operand "TagIn_2" has signal state "0".

Operands ""F_DB_1".TagSR" and "TagOut" are reset when the following conditions are fulfilled:

- Operand "TagIn_1" has signal state "0" and operand "TagIn_2" has signal state "1".
- Both operands "TagIn_1" and "TagIn_2" have signal state "1".

13.4.1.8 RS: Reset/set flip-flop (STEP 7 Safety V21)

Description

You can use the "Reset/set flip-flop" instruction to reset or set the bit of the specified operand based on the signal state of inputs R and S1. When the signal state is "1" at input R and "0" at input S1, the specified operand is reset to "0". When the signal state is "0" at input R and "1" at input S1, the specified operand is set to "1".

Input S1 takes priority over input R. If the signal state is "1" at the two inputs R and S1, the signal state of the specified operand is set to "1".

The instruction is not executed if the signal state at the two inputs R and S1 is "0". The signal state of the operand then remains unchanged.

The current signal state of the operand is transferred to output Q and can be queried there.

NOTE

If you want to use a formal parameter of an F-FC for the operand of the instruction, it must be declared as an InOut parameter.

NOTE

You cannot use the "process image", "standard DB" and "bit memory" operand areas for the operands of the instruction.

If the operand area "local data (temp)" is used for the operands of the instruction, the local data bit used must be initialized beforehand.

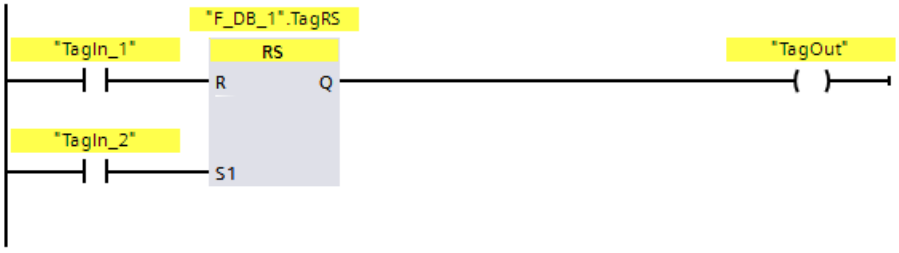
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
R	Input	BOOL	Enable reset
S1	Input	BOOL	Enable set
<Operand>	Output	BOOL	Operand that is reset or set.
Q	Output	BOOL	Signal state of the operand

Example

The following example shows how the instruction works:



Operands ""F_DB_1".TagRS" and "TagOut" are reset when the following conditions are fulfilled:

- Operand "TagIn_1" has signal state "1".
- Operand "TagIn_2" has signal state "0".

Operands ""F_DB_1".TagRS" and "TagOut" are set when the following conditions are fulfilled:

- Operand "TagIn_1" has signal state "0" and operand "TagIn_2" has signal state "1".
- Operands "TagIn_1" and "TagIn_2" have signal state "1".

13.4.1.9 --|P|--: Scan operand for positive signal edge (STEP 7 Safety V21)

Description

You can use the "Scan operand for positive signal edge" instruction to determine if there is a change from "0" to "1" in the signal state of a specified operand (<Operand1>). The instruction compares the current signal state of <Operand1> with the signal state of the previous query saved in <Operand2>. If the instruction detects a change in the result of logic operation from "0" to "1", there is a positive, rising edge.

If a rising edge is detected, the output of the instruction has signal state "1". In all other cases, the signal state at the output of the instruction is "0".

Enter the operand to be queried (<Operand1>) in the operand placeholder above the instruction. Enter the edge memory bit (<Operand2>) in the operand placeholder below the instruction.

NOTE

The address of the edge memory bit must not be used more than once in the program, otherwise the edge memory bit would be overwritten. This would influence edge evaluation and the result would no longer be unequivocal.

NOTE

If you want to use a formal parameter of an F-FC for the edge memory bit <Operand2> of the instruction, it must be declared as an InOut parameter.

NOTE

You cannot use the "process image", "standard DB" and "bit memory" operand areas for the edge memory bit <Operand2> of the instruction.

If operand area "local data (temp)" is used for the edge memory bit <Operand2> of the instruction, the local data bit used must be initialized beforehand.

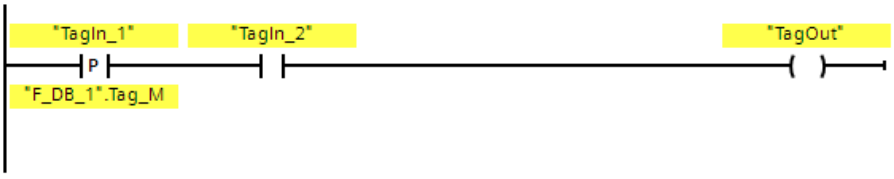
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand1>	Input	BOOL	Signal to be queried
<Operand2>	InOut	BOOL	Edge memory bit in which the signal state of the previous query is saved.

Example

The following example shows how the instruction works:



Operand "TagOut" is set when the following conditions are fulfilled:

- There is a rising edge at input "TagIn_1". The signal state of the previous query is saved at edge memory bit ""F_DB_1".Tag_M".
- The signal state of operand "TagIn_2" is "1".

13.4.1.10 --|N|--: Scan operand for negative signal edge (STEP 7 Safety V21)

Description

You can use the "Scan operand for negative signal edge" instruction to determine if there is a change form "1" to "0" in the signal state of a specified operand. The instruction compares the current signal state of <Operand1> with the signal state of the previous query saved in <Operand2>. If the instruction detects a change in the result of logic operation from "1" to "0", there is a negative, falling edge.

If a falling edge is detected, the output of the instruction has signal state "1". In all other cases, the signal state at the output of the instruction is "0".

Enter the operand to be queried (<Operand1>) in the operand placeholder above the instruction. Enter the edge memory bit (<Operand2>) in the operand placeholder below the instruction.

NOTE

The address of the edge memory bit must not be used more than once in the program, otherwise the edge memory bit would be overwritten. This would influence edge evaluation and the result would no longer be unequivocal.

NOTE

If you want to use a formal parameter of an F-FC for the edge memory bit <Operand2> of the instruction, it must be declared as an InOut parameter.

NOTE

You cannot use the "process image", "standard DB" and "bit memory" operand areas for the edge memory bit <Operand2> of the instruction.

If operand area "local data (temp)" is used for the edge memory bit <Operand2> of the instruction, the local data bit used must be initialized beforehand.

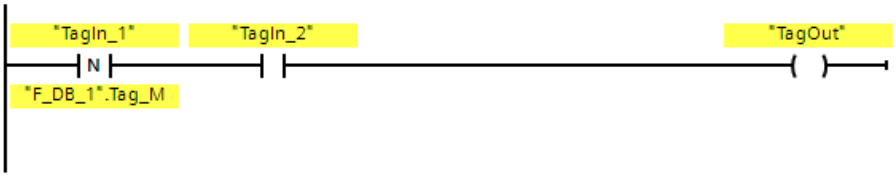
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand1>	Input	BOOL	Signal to be queried
<Operand2>	InOut	BOOL	Edge memory bit in which the signal state of the previous query is saved.

Example

The following example shows how the instruction works:



Operand "TagOut" is set when the following conditions are fulfilled:

- There is a falling edge at operand "TagIn_1". The signal state of the previous query is saved at edge memory bit ""F_DB_1".Tag_M".
- The signal state of operand "TagIn_2" is "1".

13.4.1.11 P_TRIG: Scan RLO for positive signal edge (STEP 7 Safety V21)

Description

You can use the "Scan RLO for positive signal edge" instruction to query a change in the signal state of the result of logic operation from "0" to "1". The instruction compares the current signal state of the result of logic operation (RLO) with the signal state of the previous query, which is saved in the edge bit memory (<Operand>). If the instruction detects a change in the RLO from "0" to "1", there is a positive, rising edge.

If a rising edge is detected, the output of the instruction has signal state "1". In all other cases, the signal state at the output of the instruction is "0".

NOTE

The address of the edge memory bit must not be used more than once in the program, otherwise the edge memory bit would be overwritten. This would influence edge evaluation and the result would no longer be unequivocal.

NOTE

If you want to use a formal parameter of an F-FC for the edge memory bit <Operand> of the instruction, it must be declared as an iInOut parameter.

NOTE

You cannot use the "process image", "standard DB" and "bit memory" operand areas for the edge memory bit <Operand> of the instruction.

If operand area "local data (temp)" is used for the edge memory bit <Operand> of the instruction, the local data bit used must be initialized beforehand.

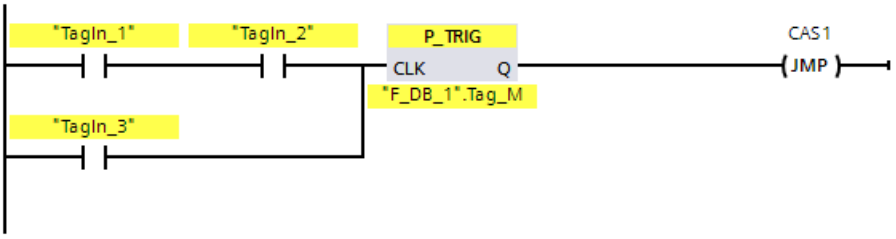
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
CLK	Input	BOOL	Current RLO
<Operand>	InOut	BOOL	Edge memory bit in which the RLO of the previous query is saved.
Q	Output	BOOL	Result of edge evaluation

Example

The following example shows how the instruction works:



The RLO from the previous bit logic operation is saved in edge memory bit ""F_DB_1".Tag_M". If a change in the RLO signal state from "0" to "1" is detected, the program jumps to jump label CAS1.

13.4.1.12 N_TRIG: Scan RLO for negative signal edge (STEP 7 Safety V21)

Description

You can use the "Scan RLO for negative signal edge" instruction to query a change in the signal state of the result of logic operation from "1" to "0". The instruction compares the current signal state of the result of logic operation with the signal state from the previous query, which is saved in the edge memory bit (<Operand>). If the instruction detects a change in the RLO from "1" to "0", there is a negative, falling edge.

If a falling edge is detected, the output of the instruction has signal state "1". In all other cases, the signal state at the output of the instruction is "0".

NOTE

The address of the edge memory bit must not be used more than once in the program, otherwise the edge memory bit would be overwritten. This would influence edge evaluation and the result would no longer be unequivocal.

NOTE

If you want to use a formal parameter of an F-FC for the edge memory bit <Operand> of the instruction, it must be declared as an iInOut parameter.

NOTE

You cannot use the "process image", "standard DB" and "bit memory" operand areas for the edge memory bit <Operand> of the instruction.

If operand area "local data (temp)" is used for the edge memory bit <Operand> of the instruction, the local data bit used must be initialized beforehand.

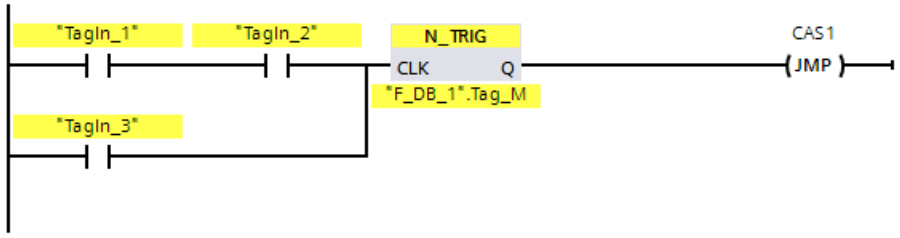
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
CLK	Input	BOOL	Current RLO
<Operand>	InOut	BOOL	Edge memory bit in which the RLO of the previous query is saved.
Q	Output	BOOL	Result of edge evaluation

Example

The following example shows how the instruction works:



The RLO of the previous bit logic operation is saved in edge bit memory ""F_DB_1".Tag_M". If a change in the RLO signal state from "1" to "0" is detected, the program jumps to jump label CAS1.

13.4.2 FBD

13.4.2.1 AND logic operation (STEP 7 Safety V21)

Description

You can use the "AND logic operation" instruction to query the signal states of two or more specified operands and evaluate them according to the AND truth table.

If the signal state of all the operands is "1", then the conditions are fulfilled and the instruction returns the result "1". If the signal state of one of the operands is "0", then the conditions are not fulfilled and the instruction generates the result "0".

If the "AND logic operation" instruction is the first instruction in a logic string, it saves the result of its signal state query in the RLO bit.

Each "AND logic operation" instruction that is not the first instruction in the logic string logically combines the result of its signal state query with the value saved in the RLO bit. This logical combination is performed according to the AND truth table.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand>	Input	BOOL	The operand indicates the bit whose signal state will be queried.

Example

The following example shows how the instruction works:



Output "TagOut" is set when the signal state of operands "TagIn_1" and "TagIn_2" is "1".

AND truth table

The following table shows the results when linking two operands by an AND logic operation:

Signal state of the first operand	Signal state of the second operand	Result of logic operation
1	1	1
0	1	0
1	0	0
0	0	0

See also

Insert binary input (STEP 7 Safety V21) (Page 457)

13.4.2.2 OR logic operation (STEP 7 Safety V21)

Description

You can use the "OR logic operation" instruction to get the signal states of two or more specified operands and evaluate them according to the OR truth table.

If the signal state of at least one of the operands is "1", then the conditions are fulfilled and the instruction returns the result "1". If the signal state of all of the operands is "0", then the conditions are not fulfilled and the instruction generates the result "0".

If the "OR logic operation" instruction is the first instruction in a logic string, it saves the result of its signal state query in the RLO bit.

Each "OR logic operation" instruction that is not the first instruction in the logic string, logically combines the result of its signal state query with the value saved in the RLO bit. This logical combination is performed according to the OR truth table.

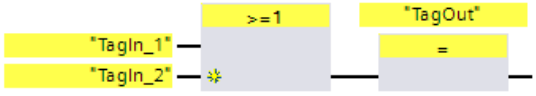
Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand>	Input	BOOL	The operand indicates the bit whose signal state will be queried.

Example

The following example shows how the instruction works:



Output "TagOut" is set when the signal state of operand "TagIn_1" or "TagIn_2" is "1".

OR truth table

The following table shows the results when linking two operands by an OR logic operation:

Signal state of the first operand	Signal state of the second operand	Result of logic operation
1	0	1
0	1	1
1	1	1
0	0	0

See also

Insert binary input (STEP 7 Safety V21) (Page 457)

13.4.2.3 X: EXCLUSIVE OR logic operation (STEP 7 Safety V21)

Description

You can use the "EXCLUSIVE OR logic operation" instruction to get the result of a signal state query according to the the EXCLUSIVE OR truth table.

With an "EXCLUSIVE OR logic operation" instruction, the signal state is "1" when the signal state of one of the two specified operands is "1". When more than two operands are queried, the overall result of logic operation is "1" if an odd-numbered quantity of queried operands returns the result "1".

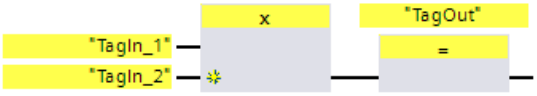
Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand>	Input	BOOL	The operand indicates the bit whose signal state will be queried.

Example

The following example shows how the instruction works:



Output "TagOut" is set when the signal state of one of the two operands "TagIn_1" and "TagIn_2" is "1". When both operands have signal state "1" or "0" then output "TagOut" is reset.

EXCLUSIVE OR truth table

The following table shows the results when two operands are linked by an EXCLUSIVE OR:

Signal state of the first operand	Signal state of the second operand	Result of logic operation
1	0	1
0	1	1
1	1	0
0	0	0

The following table shows the results when three operands are linked by an EXCLUSIVE OR:

Signal state of the first operand	Signal state of the second operand	Signal state of the third operand	Result of logic operation
1	0	0	1
0	1	1	0
0	1	0	1
1	0	1	0
0	0	1	1
1	1	0	0
1	1	1	1
0	0	0	0

See also

Insert binary input (STEP 7 Safety V21) (Page 457)

13.4.2.4 =: Assignment (STEP 7 Safety V21)

Description

You can use the "Assignment" instruction to set the bit of a specified operand. If the result of logic operation (RLO) at the box input has signal state "1" or the box input for S7-1200/1500 F-CPUs is not connected, the specified operand is set to signal state "1". If the signal state at the box input is "0", the bit of the specified operand is reset to "0".

The instruction does not influence the RLO. The RLO at the box input is assigned directly to the operand located above the Assign box.

The "Assignment" instruction can be placed at any position in the logic operation sequence.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand>	Output	BOOL	Operand to which the RLO is assigned.

Example

The following example shows how the instruction works:



Operand "TagOut" at the output of the "Assignment" instruction is set when one of the following conditions is fulfilled:

- Inputs "TagIn_1" and "TagIn_2" have signal state "1".
- The signal state at input "TagIn_3" is "0".

13.4.2.5 R: Reset output (STEP 7 Safety V21)

Description

You can use the "Reset output" instruction to reset the signal state of a specified operand to "0".

If the box input has signal state "1" or the box input for S7-1200/1500 F-CPUs is not connected, the specified operand is reset to "0". If there is a result of logic operation of "0" at the box input, the signal state of the specified operand remains unchanged.

The instruction does not influence the RLO. The RLO at the box input is transferred directly to the box output.

NOTE

If you want to use a formal parameter of an F-FC for the operand of the instruction, it must be declared as an InOut parameter.

NOTE

If the operand area "local data (temp)" is used for the operands of the instruction, the local data bit used must be initialized beforehand.

NOTE

You cannot use the "process image of the inputs", "process image of the outputs" from standard I/O and "standard DB" and "bit memory" operand areas for the operands of the instruction.

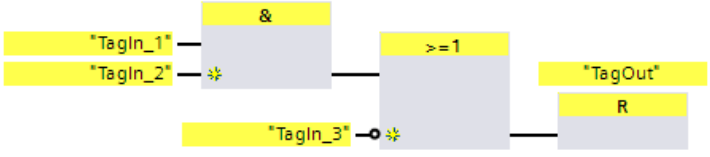
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand>	Output	BOOL	Operand that is reset with RLO = "1".

Example

The following example shows how the instruction works:



Operand "TagOut" is reset when one of the following conditions is fulfilled:

- Operands "TagIn_1" and "TagIn_2" have signal state "1".
- The signal state of operand "TagIn_3" is "0".

13.4.2.6 S: Set output (STEP 7 Safety V21)

Description

You can use the "Set output" instruction to set the signal state of a specified operand to "1".

If the box input has signal state "1" or the box input for S7-1200/1500 F-CPUs is not connected, the specified operand is set to "1". If there is a result of logic operation of "0" at the box input, the signal state of the specified operand remains unchanged.

The instruction does not influence the RLO. The RLO at the box input is transferred directly to the box output.

NOTE

The instruction is not executed if it is applied to an output of an F-I/O that is passivated (e.g., during startup of the F-system). Therefore, it is preferable to access outputs of the F-I/O using only the "Assignment" instruction.

An F-I/O output is passivated if QBAD or QBAD_O_xx = 1 or value status = 0 is set in the corresponding F-I/O DB.

NOTE

If you want to use a formal parameter of an F-FC for the operand of the instruction, it must be declared as an InOut parameter.

NOTE

If the operand area "local data (temp)" is used for the operands of the instruction, the local data bit used must be initialized beforehand.

NOTE

You cannot use the "process image of the inputs", "process image of the outputs" from standard I/O and "standard DB" and "bit memory" operand areas for the operands of the instruction.

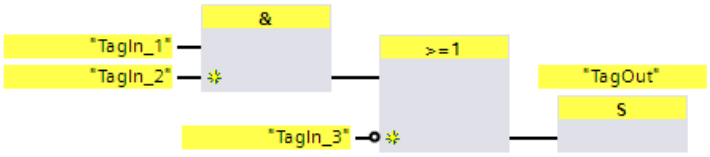
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand>	Output	BOOL	Operand that is set when RLO = "1".

Example

The following example shows how the instruction works:



Operand "TagOut" is set when one of the following conditions is fulfilled:

- Operands "TagIn_1" and "TagIn_2" have signal state "1".
- The signal state of operand "TagIn_3" is "0".

13.4.2.7 SR: Set/reset flip-flop (STEP 7 Safety V21)

Description

You can use the "Set/reset flip-flop" instruction to set or reset the bit of the specified operand based on the signal state of inputs S and R1. If the signal state at input S is "1" and the signal state at input R1 is "0", the specified operand is set to "1". If the signal state at input S is "0" and the signal state at input R1 is "1", the specified operand is reset to "0".

Input R1 takes priority over input S. If the signal state is "1" at the two inputs S and R1, the signal state of the specified operand is reset to "0".

The instruction is not executed if the signal state at the two inputs S and R1 is "0". The signal state of the operand then remains unchanged.

The current signal state of the operand is transferred to output Q and can be queried there.

NOTE

If you want to use a formal parameter of an F-FC for the operand of the instruction, it must be declared as an InOut parameter.

NOTE

You cannot use the "process image", "standard DB" and "bit memory" operand areas for the operands of the instruction.

If the operand area "local data (temp)" is used for the edge bit memory of the instruction, the local data bit used must be initialized beforehand.

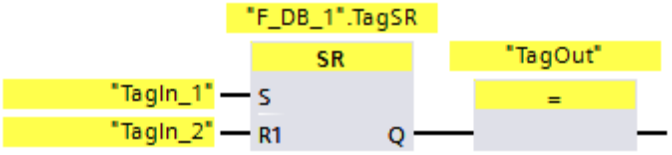
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
S	Input	BOOL	Enable set
R1	Input	BOOL	Enable reset
<Operand>	Output	BOOL	Operand that is set or reset.
Q	Output	BOOL	Signal state of the operand

Example

The following example shows how the instruction works:



Operands ""F_DB_1".TagSR" and "TagOut" are set when the following conditions are fulfilled:

- Operand "TagIn_1" has signal state "1".
- Operand "TagIn_2" has signal state "0".

Operands ""F_DB_1".TagSR" and "TagOut" are reset when the following conditions are fulfilled:

- Operand "TagIn_1" has signal state "0" and operand "TagIn_2" has signal state "1".
- Both operands "TagIn_1" and "TagIn_2" have signal state "1".

13.4.2.8 RS: Reset/set flip-flop (STEP 7 Safety V21)

Description

You can use the "Reset/set flip-flop" instruction to reset or set the bit of the specified operand based on the signal state of inputs R and S1. When the signal state is "1" at input R and "0" at input S1, the specified operand is reset to "0". When the signal state is "0" at input R and "1" at input S1, the specified operand is set to "1".

Input S1 takes priority over input R. If the signal state is "1" at the two inputs R and S1, the signal state of the specified operand is set to "1".

The instruction is not executed if the signal state at the two inputs R and S1 is "0". The signal state of the operand then remains unchanged.

The current signal state of the operand is transferred to output Q and can be queried there.

NOTE

If you want to use a formal parameter of an F-FC for the operand of the instruction, it must be declared as an InOut parameter.

NOTE

You cannot use the "process image", "standard DB" and "bit memory" operand areas for the operands of the instruction.

If the operand area "local data (temp)" is used for the edge bit memory of the instruction, the local data bit used must be initialized beforehand.

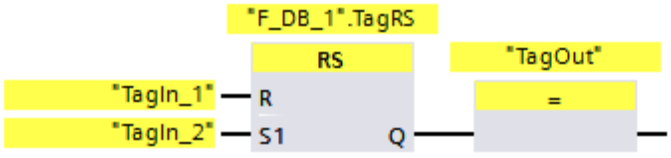
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
R	Input	BOOL	Enable reset
S1	Input	BOOL	Enable set
<Operand>	Output	BOOL	Operand that is reset or set.
Q	Output	BOOL	Signal state of the operand

Example

The following example shows how the instruction works:



Operands ""F_DB_1".TagRS" and "TagOut" are reset when the following conditions are fulfilled:

- Operand "TagIn_1" has signal state "1".
- Operand "TagIn_2" has signal state "0".

Operands ""F_DB_1".TagRS" and "TagOut" are set when the following conditions are fulfilled:

- Operand "TagIn_1" has signal state "0" and operand "TagIn_2" has signal state "1".
- Operands "TagIn_1" and "TagIn_2" have signal state "1".

13.4.2.9 P: Scan operand for positive signal edge (STEP 7 Safety V21)

Description

You can use the "Scan operand for positive signal edge" instruction to determine if there is a change from "0" to "1" in the signal state of a specified operand (<Operand1>). The instruction compares the current signal state of <Operand1> with the signal state of the previous query saved in <Operand2>. If the instruction detects a change in the result of logic operation from "0" to "1", there is a positive, rising edge.

If a rising edge is detected, the output of the instruction has signal state "1". In all other cases, the signal state at the output of the instruction is "0".

Enter the operand to be queried (<Operand1>) in the operand placeholder above the instruction. Enter the edge memory bit (<Operand2>) in the operand placeholder below the instruction.

NOTE

The address of the edge memory bit must not be used more than once in the program, otherwise the edge memory bit would be overwritten. This would influence edge evaluation and the result would no longer be unequivocal.

NOTE

If you want to use a formal parameter of an F-FC for the edge memory bit <Operand2> of the instruction, it must be declared as an InOut parameter.

NOTE

You cannot use the "process image", "standard DB" and "bit memory" operand areas for the edge memory bit <Operand2> of the instruction.

If operand area "local data (temp)" is used for the edge memory bit <Operand2> of the instruction, the local data bit used must be initialized beforehand.

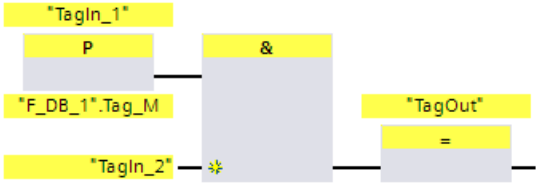
Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand1>	Input	BOOL	Signal to be queried
<Operand2>	InOut	BOOL	Edge memory bit in which the signal state of the previous query is saved.

Example

The following example shows how the instruction works:



"TagOut" is set when the following conditions are fulfilled:

- There is a rising edge at input "TagIn_1".
- The signal state of operand "TagIn_2" is "1".

13.4.2.10 N: Scan operand for negative signal edge (STEP 7 Safety V21)

Description

You can use the "Scan operand for negative signal edge" instruction to determine if there is a change form "1" to "0" in the signal state of a specified operand. The instruction compares the current signal state of <Operand1> with the signal state of the previous query saved in <Operand2>. If the instruction detects a change in the result of logic operation from "1" to "0", there is a negative, falling edge.

If a falling edge is detected, the output of the instruction has signal state "1". In all other cases, the signal state at the output of the instruction is "0".

Enter the operand to be queried (<Operand1>) in the operand placeholder above the instruction. Enter the edge memory bit (<Operand2>) in the operand placeholder below the instruction.

NOTE

The address of the edge memory bit must not be used more than once in the program, otherwise the edge memory bit would be overwritten. This would influence edge evaluation and the result would no longer be unequivocal.

NOTE

If you want to use a formal parameter of an F-FC for the edge memory bit <Operand2> of the instruction, it must be declared as an InOut parameter.

NOTE

You cannot use the "process image", "standard DB" and "bit memory" operand areas for the edge memory bit <Operand2> of the instruction.

If operand area "local data (temp)" is used for the edge memory bit <Operand2> of the instruction, the local data bit used must be initialized beforehand.

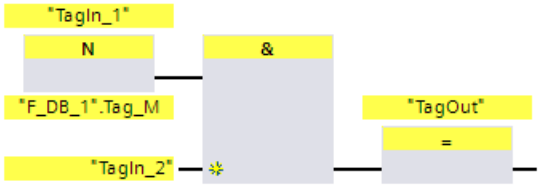
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Operand1>	Input	BOOL	Signal to be queried
<Operand2>	InOut	BOOL	Edge memory bit in which the signal state of the previous query is saved.

Example

The following example shows how the instruction works:



Output "TagOut" is set when the following conditions are fulfilled:

- There is a falling edge at input "TagIn_1".
- The signal state of operand "TagIn_2" is "1".

13.4.2.11 P_TRIG: Scan RLO for positive signal edge (STEP 7 Safety V21)

Description

You can use the "Scan RLO for positive signal edge" instruction to query a change in the signal state of the result of logic operation from "0" to "1". The instruction compares the current signal state of the result of logic operation with the signal state from the previous query, which is saved in the edge memory bit (<Operand>). If the instruction detects a change in the RLO from "0" to "1", there is a positive, rising edge.

If a rising edge is detected, the output of the instruction has signal state "1". In all other cases, the signal state at the output of the instruction is "0".

NOTE

The address of the edge memory bit must not be used more than once in the program, otherwise the edge memory bit would be overwritten. This would influence edge evaluation and the result would no longer be unequivocal.

NOTE

If you want to use a formal parameter of an F-FC for the edge memory bit <Operand> of the instruction, it must be declared as an iInOut parameter.

NOTE

You cannot use the "process image", "standard DB" and "bit memory" operand areas for the edge memory bit <Operand> of the instruction.

If operand area "local data (temp)" is used for the edge memory bit <Operand> of the instruction, the local data bit used must be initialized beforehand.

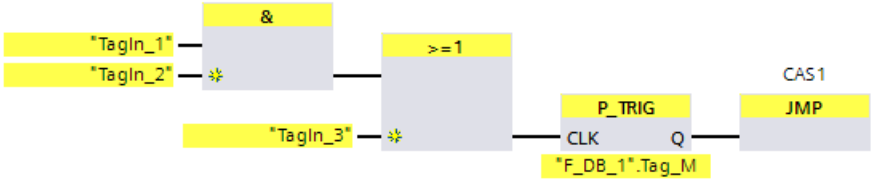
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
CLK	Input	BOOL	Current RLO
<Operand>	InOut	BOOL	Edge memory bit in which the RLO of the previous query is saved.
Q	Output	BOOL	Result of edge evaluation

Example

The following example shows how the instruction works:



The RLO from the previous bit logic operation is saved in edge memory bit ""F_DB_1".Tag_M". If a change in the RLO signal state from "0" to "1" is detected, the program jumps to jump label CAS1.

13.4.2.12 N_TRIG: Scan RLO for negative signal edge (STEP 7 Safety V21)

Description

You can use the "Scan RLO for negative signal edge" instruction to query a change in the signal state of the result of logic operation from "1" to "0". The instruction compares the current signal state of the result of logic operation with the signal state from the previous query, which is saved in the edge memory bit (<Operand>). If the instruction detects a change in the RLO from "1" to "0", there is a negative, falling edge.

If a falling edge is detected, the output of the instruction has signal state "1". In all other cases, the signal state at the output of the instruction is "0".

NOTE

The address of the edge memory bit must not be used more than once in the program, otherwise the edge memory bit would be overwritten. This would influence edge evaluation and the result would no longer be unequivocal.

NOTE

If you want to use a formal parameter of an F-FC for the edge memory bit <Operand> of the instruction, it must be declared as an iInOut parameter.

NOTE

You cannot use the "process image", "standard DB" and "bit memory" operand areas for the edge memory bit <Operand> of the instruction.

If operand area "local data (temp)" is used for the edge memory bit <Operand> of the instruction, the local data bit used must be initialized beforehand.

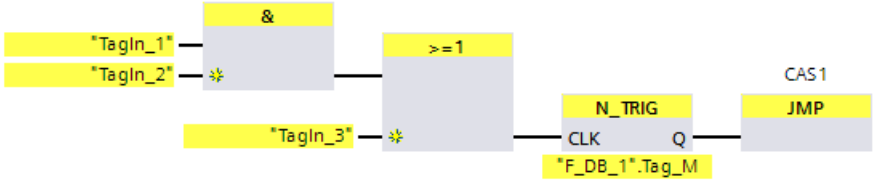
Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
CLK	Input	BOOL	Current RLO
<Operand>	InOut	BOOL	Edge memory bit in which the RLO of the previous query is saved.
Q	Output	BOOL	Result of edge evaluation

Example

The following example shows how the instruction works:



The RLO of the previous bit logic operation is saved in edge bit memory ""F_DB_1".Tag_M". If a change in the RLO signal state from "1" to "0" is detected, the program jumps to jump label CAS1.

13.5 Safety functions

13.5.1 ESTOP1: Emergency STOP/emergency OFF up to stop category 1 (STEP 7 Safety V21)

Description

This instruction supports the implementation of an EMERGENCY STOP/EMERGENCY OFF shutdown with acknowledgment for STOP categories 0 and 1 for the purpose of complying with requirements of IEC 60204 or IEC 61800-5-2. However, additional measures described in the respective standard may have to be taken in order to fully comply with the requirements of the respective standard.

Enable signal Q is reset to 0, as soon as input E_STOP takes a signal state of 0 (Stop category 0). Enable signal Q_DELAY is reset to 0 after the time delay set at input TIME_DEL (Stop Category 1).

Enable signal Q is reset to 1 not before input E_STOP takes a signal state of 1 and an acknowledgment occurs. The acknowledgment for the enable takes place according to the parameter assignment at input ACK_NEC:

- If ACK_NEC = 0, the acknowledgment is automatic.
- If ACK_NEC = 1, you must use a rising edge at input ACK for acknowledging the enable.

Output ACK_REQ is used to signal that a user acknowledgment is required at input ACK for the acknowledgment. The instruction sets output ACK_REQ to 1, as soon as input E_STOP = 1.

Following an acknowledgment, the instruction resets ACK_REQ to 0.

Every call of the "Emergency STOP/Emergency OFF up to Stop Category 1" instruction must be assigned a data area in which the instruction data are stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., ESTOP1_DB_1) or a multi-instance (e.g., ESTOP1_Instance_1) for the "Emergency STOP/Emergency OFF up to Stop Category 1" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

⚠ WARNING

The ACK_NEC tag must not be assigned a value of "0" unless an automatic restart of the affected process is otherwise excluded. *(S033)*

Note: In case of two channels according to category 3.4 of ISO 13849-1:2015 or EN ISO 13849-1:2015, the discrepancy monitoring of the two NC contacts of the EMERGENCY STOP/EMERGENCY OFF must already take place in the F-I/O. The F-I/O has to be configured accordingly (sensor evaluation: two-channel, equivalent), and the result of the evaluation interconnected with the E_STOP input. In order to keep the discrepancy time from influencing the response time, you must assign "Supply value 0" for the behavior of discrepancy during configuration.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
E_STOP	Input	BOOL	EMERGENCY STOP/EMERGENCY OFF
ACK_NEC	Input	BOOL	1=Acknowledgment necessary
ACK	Input	BOOL	1=Acknowledgment
TIME_DEL	Input	TIME	Time delay
Q	Output	BOOL	1=Enable
Q_DELAY	Output	BOOL	Enable is OFF delayed
ACK_REQ	Output	BOOL	1=Acknowledgment necessary
DIAG	Output	BYTE	Non-fail safe service information

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	Version 1.0 requires that the F_TOF block with the number FB 186 is available in the project tree in the "Program blocks/System blocks/STEP 7 Safety" folder.
1.1	x	—	—	—	These versions have identical functions to version 1.0, but do not require the F_TOF block to have a particular number.
1.2	x	—	—	o	
1.3	x	o	—	o	
1.4	x	o	—	o	
1.5	x	x	—	x	
1.6	x	x	x	x	The reaction of the delay time TIME_DEL for F-CPUs S7-1200/1500 was adapted to the reaction of F-CPUs S7-300/400: If the input ESTOP (0 -> 1 (including acknowledgment) -> 0) is changed while the delay time is running, the delay time is restarted.

o This version is no longer supported.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Startup behavior

After an F-system startup, when ACK_NEC = 1, you must acknowledge the instruction using a rising edge at input ACK.

Output DIAG

The DIAG output provides non-fail-safe information on errors for service purposes. You can read out this information by means of operator control and monitoring systems or, if applicable, you can evaluate it in your standard user program. DIAG bits 4 and 5 are saved until you acknowledge at the ACK input.

Structure of DIAG

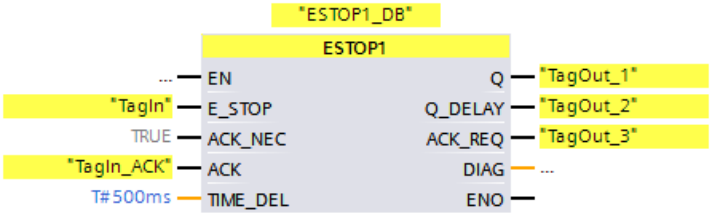
Bit no.	Assignment	Possible error causes	Remedies
Bit 0	Incorrect TIM_DEL setting	Time delay setting < 0	Set time delay > 0
Bit 1	Reserved	—	—
Bit 2	Reserved	—	—
Bit 3	Reserved	—	—
Bit 4	Acknowledgment not possible because emergency STOP/emergency OFF is still active	Emergency STOP/Emergency OFF pushbutton is locked	Release Emergency STOP/Emergency OFF pushbutton
		F-I/O fault, channel fault, or communication error, or passivation by means of PASS_ON of F-I/O of emergency STOP/emergency OFF pushbutton	For a solution, see the section "Structure of DIAG", bits 0 to 6 in DIAG (Page 171)
		Emergency STOP/Emergency OFF pushbutton is defective	Check emergency STOP/emergency OFF pushbutton
		Wiring fault	Check wiring of the emergency STOP/emergency OFF pushbutton
Bit 5	If enable is missing: input ACK has a permanent signal state of 1	Acknowledgment button defective	Check acknowledgment button
		Wiring fault	Check wiring of acknowledgment button
Bit 6	Acknowledgment required (= state of ACK_REQ)	—	—
Bit 7	State of output Q	—	—

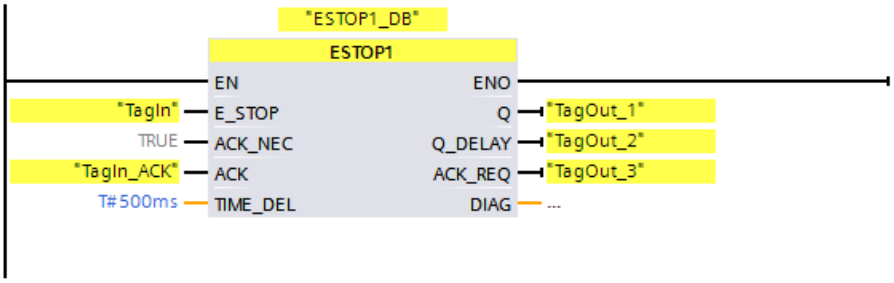
Timing imprecision resulting from the update time and the tolerance of the time base used in the instruction

Refer to section "Temporal blurring for instructions with time processing (Page 450)" for more on this.

Example

The following example shows how the instruction works:





13.5.2 TWO_HAND: Two-hand monitoring (STEP 7 Safety Advanced V21) (S7-300, S7-400)

Description

This instruction supports the implementation of two-hand monitoring for the purpose of complying with requirements of ISO 13851. However, additional measures described in the standard may have to be taken in order to fully comply with the requirements of the standard.

NOTE

This instruction is only available for S7-300 and S7-400 F-CPUs. For S7-1200/1500 F-CPUs, you use the instruction "Two-hand monitoring with enable". The application "Two-hand monitoring with enable" replaces the instruction "Two-hand monitoring" with compatible functions.

If pushbuttons IN1 and IN2 are activated within the permitted discrepancy time DISCTIME ≤ 500 ms (IN1/IN2 = 1) (synchronous activation), output signal Q is set to 1. If the time difference between activation of pushbutton IN1 and pushbutton IN2 is greater than DISCTIME, then the pushbuttons must be released and reactivated.

Q is reset to 0 as soon as one of the pushbuttons is released (IN1/IN2 = 0). Enable signal Q can be reset to 1 only if the other pushbutton has been released, and if both pushbuttons are then reactivated within the discrepancy time. Enable signal Q can never be set to 1 if the discrepancy time is set to values less than 0 or greater than 500 ms.

Every call of the "Two-hand monitoring" instruction must be assigned a data area in which the instruction data are stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., TWO_HAND_DB_1) or a multi-instance (e.g., TWO_HAND_Instance_1) for the "Two-hand monitoring" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Note: Only one signal per pushbutton can be evaluated in the instruction. Discrepancy monitoring of the NC and NO contacts of pushbuttons IN1 and IN2 is performed directly during suitable configuration (sensor evaluation: 1oo2 evaluation, non-equivalent) directly through the F-I/O with inputs. The normally open contact must be wired in such a way that it supplies the useful signal (see manual for the F-I/O you are using). In order to keep the

discrepancy time from influencing the response time, you must assign "Supply value 0" for the behavior of discrepancy during configuration. If a discrepancy is detected, a fail-safe value of 0 is entered in the process image of the inputs (PII) for the pushbutton and QBAD or QBAD_I_xx = 1 is set in the relevant F-I/O DB. (See also F-I/O access (Page 157))

Parameters

The following table shows the parameters of the instruction:

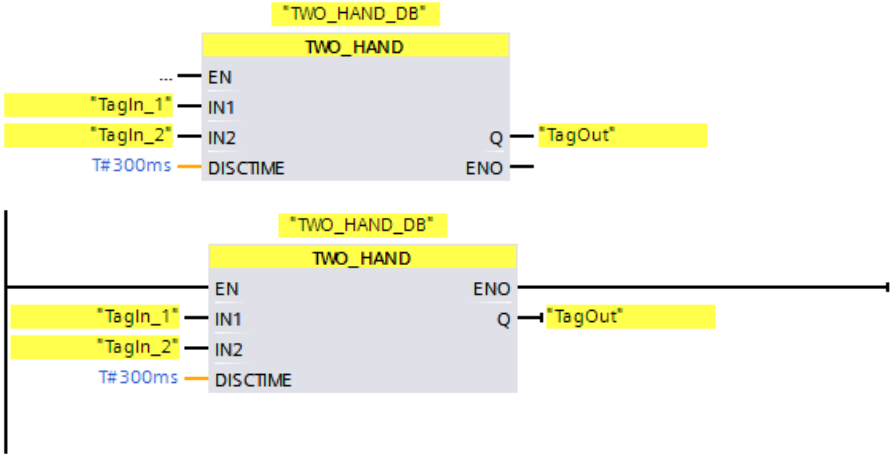
Parameter	Declaration	Data type	Description
IN1	Input	BOOL	Pushbutton 1
IN2	Input	BOOL	Pushbutton 2
DISCTIME	Input	TIME	Discrepancy time (0 to 500 ms)
Q	Output	BOOL	1=Enable

Timing imprecision resulting from the update time and the tolerance of the time base used in the instruction:

Refer to section "Temporal blurring for instructions with time processing (Page 450)" for more on this.

Example

The following example shows how the instruction works:



13.5.3 TWO_H_EN: Two-hand monitoring with enable (STEP 7 Safety V21)

Description

This instruction supports the implementation of two-hand monitoring with enable for the purpose of complying with requirements of ISO 13851. However, additional measures described in the standard may have to be taken in order to fully comply with the requirements of the standard.

If pushbuttons IN1 and IN2 are activated within the permitted discrepancy time DISCTIME ≤ 500 ms (IN1/IN2 = 1) (synchronous activation), output signal Q is set to 1 when existing ENABLE = 1. If the time difference between activation of pushbutton IN1 and pushbutton IN2 is greater than DISCTIME, then the pushbuttons must be released and reactivated.

Q is reset to 0 as soon as one of the pushbuttons is released (IN1/IN 2 = 0) or ENABLE = 0. Enable signal Q can be reset to 1 only if the other pushbutton has been released, and if both pushbuttons are then reactivated within the discrepancy time when existing ENABLE = 1.

Every call of the "Two-hand monitoring with enable" instruction must be assigned a data area in which the instruction data are stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., TWO_H_EN_DB_1) or a multi-instance (e.g., TWO_H_EN_Instance_1) for the "Two-hand monitoring with enable" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Note: Only one signal per pushbutton can be evaluated in the instruction. Discrepancy monitoring of the NC and NO contacts of pushbuttons IN1 and IN2 is performed directly during corresponding configuration (sensor evaluation: 1oo2 evaluation, non-equivalent) directly through the F-I/O with inputs. The normally open contact must be wired in such a way that it supplies the useful signal (see manual for the F-I/O you are using). In order to keep the discrepancy time from influencing the response time, during the configuration of discrepancy behavior, you must configure "Supply value 0".

If a discrepancy is detected, a fail-safe value of 0 is entered in the process image of the inputs (PII) for the pushbutton and QBAD or QBAD_I_xx = 1 or respectively value status = 0 is set in the relevant F-I/O DB.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN1	Input	BOOL	Pushbutton 1
IN2	Input	BOOL	Pushbutton 2
ENABLE	Input	BOOL	Enable input
DISCTIME	Input	TIME	Discrepancy time (0 to 500 ms) For productive use, select a discrepancy time > 0 ms.
Q	Output	BOOL	1=Enable
DIAG	Output	BYTE	Non-fail safe service information

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	x	—	—	o	These versions have identical functions to version 1.0.
1.2	x	o	—	o	
1.3	x	x	x	x	

o This version is no longer supported.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Output DIAG

The DIAG output provides non-fail-safe information on errors for service purposes. You can read out this information by means of operator control and monitoring systems or, if applicable, you can evaluate it in your standard user program. DIAG bits 0 to 5 are saved until the cause of the error has been eliminated.

Structure of DIAG

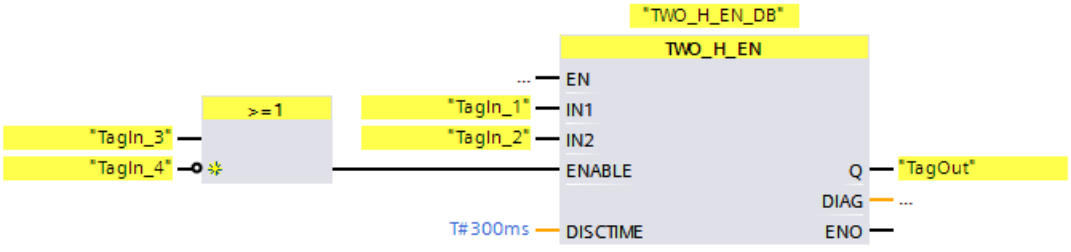
Bit no.	Assignment	Possible error causes	Remedies
Bit 0	Incorrect discrepancy time DISCTIME setting	Discrepancy time setting is <0 or > 500 ms	Set discrepancy time in range of 0 to 500 ms
Bit 1	Discrepancy time elapsed	Discrepancy time setting is too low	If necessary, set a higher discrepancy time
		Pushbuttons were not activated within the discrepancy time	Release pushbuttons and activate them within the discrepancy time
		Wiring fault	Check wiring of pushbuttons
		Pushbuttons defective	Check pushbuttons
		Pushbuttons are wired to different F-I/O, and F-I/O fault, channel fault, or communication error, or passivation by means of PASS_ON on an F-I/O	For a solution, see the section "Structure of DIAG", bits 0 to 6 in DIAG (Page 171)
Bit 2	Reserved	—	—
Bit 3	Reserved	—	—
Bit 4	Incorrect activation sequence	One pushbutton was not released	Release pushbuttons and activate them within the discrepancy time
		Pushbuttons defective	Check pushbuttons
Bit 5	ENABLE does not exist	ENABLE = 0	Set ENABLE = 1, release pushbutton and activate it within the discrepancy time
Bit 6	Reserved	—	—
Bit 7	State of output Q	—	—

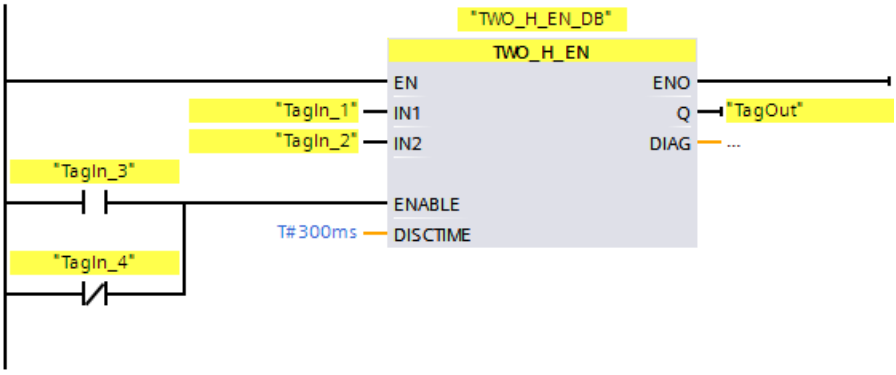
Timing imprecision resulting from the update time and the tolerance of the time base used in the instruction

Refer to section "Temporal blurring for instructions with time processing (Page 450)" for more on this.

Example

The following example shows how the instruction works:





13.5.4 MUTING: Muting (STEP 7 Safety Advanced V21) (S7-300, S7-400)

Description

This instruction performs parallel muting with two or four muting sensors.

NOTE

This instruction is only available for S7-300 and S7-400 F-CPUs. For S7-1200/1500 F-CPUs, you use the instruction "Parallel muting (Page 504)". The instruction "Parallel muting" replaces the instruction "Muting" with compatible functions.

Muting is a defined suppression of the protective function of light curtains. Light curtain muting can be used to introduce goods or objects into the danger area monitored by the light curtain without causing the machine to stop.

To utilize the muting function, at least two independently wired muting sensors must be present. The use of two or four muting sensors and correct integration into the production sequence must ensure that no persons enter the danger area while the light curtain is muted.

Every call of the "Muting" instruction must be assigned a data area in which the instruction data are stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., MUTING_DB_1) or a multi-instance (e.g., MUTING_Instance_1) for the "Muting" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

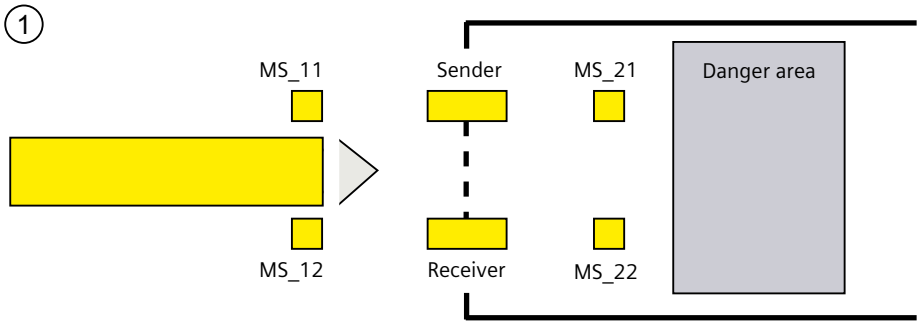
Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
MS_11	Input	BOOL	Muting sensor 1 of sensor pair 1
MS_12	Input	BOOL	Muting sensor 2 of sensor pair 1
MS_21	Input	BOOL	Muting sensor 1 of sensor pair 2
MS_22	Input	BOOL	Muting sensor 2 of sensor pair 2
STOP	Input	BOOL	1=Conveyor system stopped
FREE	Input	BOOL	1=Light curtain uninterrupted
QBAD_MUT	Input	BOOL	QBAD signal of the F-I/O or QBAD_O_xx signal of the muting lamp channel
DISCTIM1	Input	TIME	Discrepancy time of sensor pair 1 (0 to 3 s)
DISCTIM2	Input	TIME	Discrepancy time of sensor pair 2 (0 to 3 s)
TIME_MAX	Input	TIME	Maximum muting time (0 to 10 min)
ACK	Input	BOOL	Acknowledgment of restart inhibit
Q	Output	BOOL	1= Enable, not off
MUTING	Output	BOOL	Display of muting is active
ACK_REQ	Output	BOOL	Acknowledgment necessary
FAULT	Output	BOOL	Group error
DIAG	Output	BYTE	Non-fail safe service information

Schematic sequence of error-free muting procedure with 4 muting sensors (MS_11, MS_12, MS_21, MS_22)



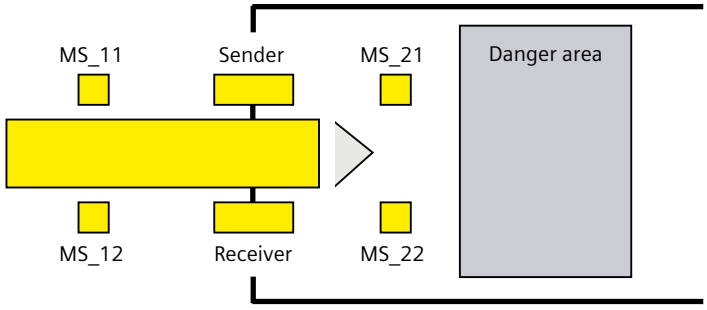
- If both muting sensors MS_11 and MS_12 are activated by the product within DISCTIM1 (apply signal state = 1), the instruction starts the MUTING function. Enable signal Q remains 1, even when input FREE = 0 (light curtain interrupted by product). The MUTING output for setting the muting lamp switches to 1.

NOTE

The muting lamp can be monitored using the QBAD_MUT input. To do this, you must wire the muting lamp to an output with wire break monitoring of an F-I/O and supply the QBAD_MUT input with the QBAD signal of the associated F-I/O or the QBAD_O_xx signal of the associated channel. If QBAD_MUT = 1, muting is terminated by the instruction. If monitoring of the muting lamp is not necessary, you do not have to supply input QBAD_MUT.

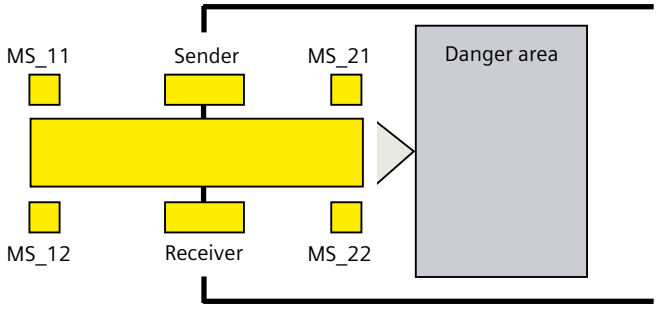
F-I/O that can promptly detect a wire break after activation of the muting operation must be used (*see manual for specific F-I/O*).

②



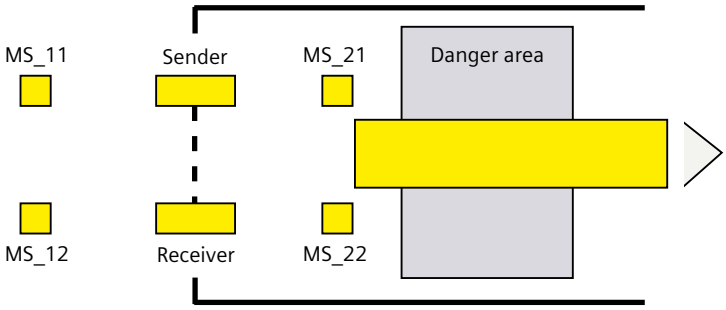
- As long as both muting sensors MS_11 and MS_12 continue to be activated, the MUTING function of the instruction causes Q to remain 1 and MUTING to remain 1 (so that the product can pass through the light curtain without causing the machine to stop).

③



- The two muting sensors MS_21 and MS_22 must be activated (within DISCTIM2) before muting sensors MS_11 and MS_12 are switched to inactive (apply signal state 0). In this way, the instruction retains the MUTING function. (Q = 1, MUTING = 1).

④

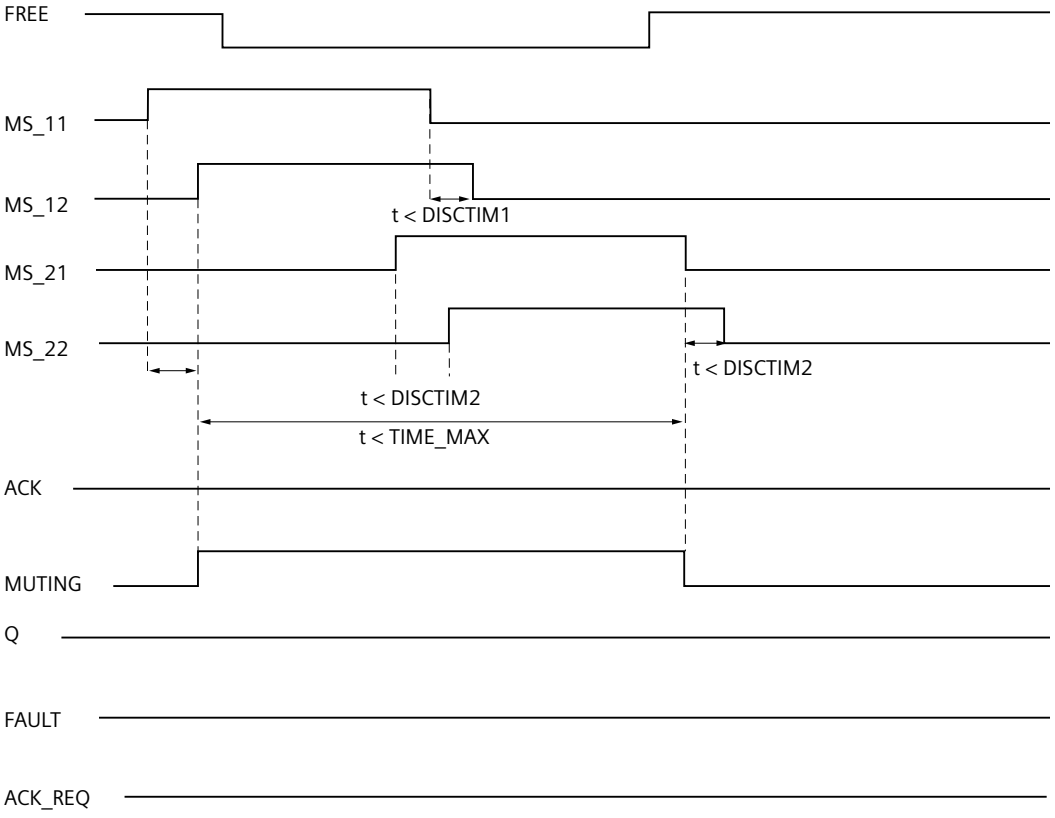


- Only if one of the two muting sensors MS_21 and MS_22 is switched to inactive (product enables sensors) is the MUTING function terminated (Q = 1, MUTING = 0). The maximum activation time for the MUTING function is the time set at input TIME_MAX.

NOTE

The MUTING function is also started if the product passes the light curtain in the reverse direction and the muting sensors are thus activated by the product in reverse order.

Timing diagrams for error-free muting procedure with 4 muting sensors

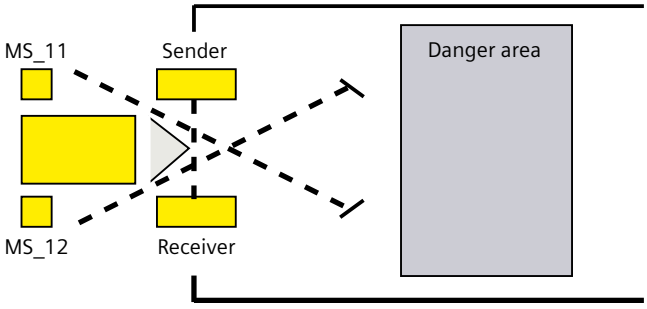


Schematic sequence of muting procedure with reflection light barriers

If reflection light barriers are used as muting sensors, they are generally arranged diagonally.

In general, this arrangement of reflection light barriers as muting sensors requires only two light barriers, and only MS_11 and MS_12 are interconnected.

The sequence is similar to that of the muting procedure with 4 muting sensors. Step 3 is omitted. In step 4, replace MS_21 and MS_22 with MS_11 and MS_12, respectively.



Restart inhibit upon interruption of light curtain (if MUTING is not active), when errors occur, and during F-system startup

Enable signal Q cannot be set to 1 or becomes 0, if:

- Light curtain is interrupted (e.g., by a person or material transport) while the MUTING function is not active
- The muting lamp monitoring function responds at input QBAD_MUT
- Sensor pair 1 (MS_11 and MS_12) or sensor pair 2 (MS_21 and MS_22) is not activated or deactivated during discrepancy time DISCTIM1 or DISCTIM2, respectively
- The MUTING function is active longer than the maximum muting time TIME_MAX
- Discrepancy times DISCTIM1 and DISCTIM2 have been set to values < 0 or > 3 s
- Maximum muting time TIME_MAX has been set to a value< 0 or > 10 min

In the identified cases, output FAULT (group error) is set to 1 (restart inhibit). If the MUTING function is started, it will be terminated and the Muting output becomes 0.

⚠ WARNING

When a valid combination of signal states of muting sensors is immediately detected at startup of the F-system (for example, because the muting sensors are interconnected to inputs of a standard I/O that immediately provide process values during the F-system startup), the MUTING function is immediately started and the MUTING output and enable signal Q are set to 1. The FAULT output (group error) is not set to 1 (no restart inhibit!). *(S035)*

Acknowledgment of restart inhibit

Enable signal Q becomes 1 again, when:

- The light curtain is no longer interrupted
- If present, errors are eliminated (see output DIAG)

  and

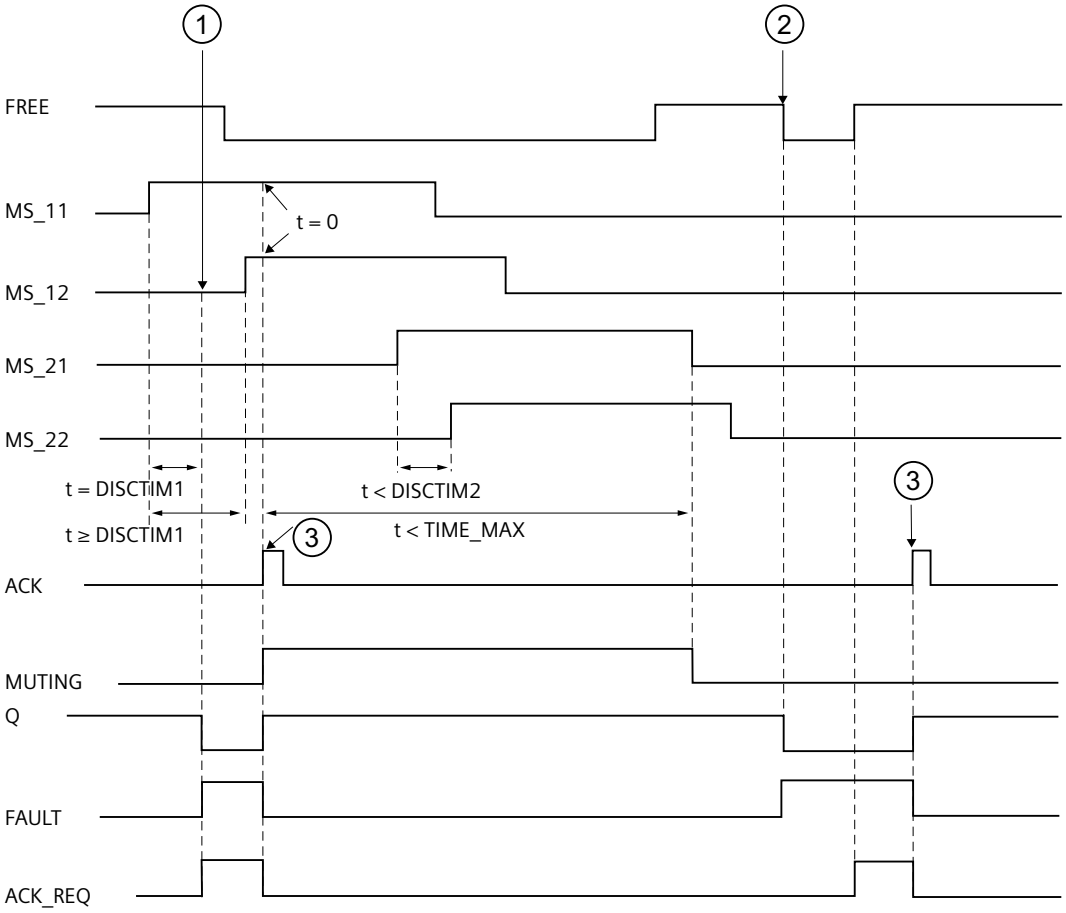
- A user acknowledgment with positive edge occurs at input ACK (see also Implementation of user acknowledgment (Page 190)).

The FAULT output is set to 0. Output ACK_REQ = 1 signals that user acknowledgment at input ACK is required to eliminate the restart inhibit. The instruction sets ACK_REQ = 1 as soon as the light curtain is no longer interrupted or the errors have been eliminated. Once acknowledgment has occurred, the instruction resets ACK_REQ to 0.

NOTE

Following discrepancy errors and once the maximum muting time has been exceeded, ACK_REQ is immediately set to 1. As soon as a user acknowledgment has taken place at input ACK, discrepancy times DISCTIM1 and DISCTIM2 and maximum muting time TIME_MAX are reset.

Timing diagrams for discrepancy errors at sensor pair 1 or interruption of the light curtain (if MUTING is not active)



① Sensor pair 1 (MS_11 and MS_12) is not activated within discrepancy time DISCTIM1.

② The light curtain is interrupted even though the MUTING function is not active.

③ Acknowledgment

Behavior with stopped conveyor equipment

If, while the conveyor equipment has stopped, the monitoring for one of the following reasons is to be deactivated:

- To comply with discrepancy time DISCTIM1 or DISCTIM2
- To comply with maximum muting time TIME_MAX

You must supply input STOP with a "1" signal for as long as the conveyor equipment is stopped. As soon as the conveyor equipment is running again (STOP = 0), discrepancy times DISCTIM1 and DISCTIM2 and maximum muting time TIME_MAX are reset.

⚠ WARNING

When STOP = 1, the discrepancy monitoring is shut down. During this time, if inputs MSx1/MSx2 of a sensor pair both take a signal state of 1 due to an undetected error, for example, because both muting sensors have adopted signal state 1 in the event of an error, the fault is not detected and the MUTING function can be started unintentionally. *(S036)*

Output DIAG

The DIAG output provides non-fail-safe information on errors for service purposes. You can read out this information by means of operator control and monitoring systems or, if applicable, you can evaluate it in your standard user program. DIAG bits are saved until acknowledgment at input ACK.

Structure of DIAG

Bit no.	Assignment	Possible error causes	Remedies
Bit 0	Discrepancy error or incorrect discrepancy time DISCTIM 1 setting for sensor pair 1	Malfunction in production sequence	Malfunction in production sequence eliminated
		Sensor defective	Check sensors
		Wiring fault	Check wiring of sensors
		Sensors are wired to different F-I/O, and F-I/O fault, channel fault, or communication error, or passivation by means of PASS_ON on an F-I/O	For a solution, see the section "Structure of DIAG", bits 0 to 6 in DIAG (Page 171)
		Discrepancy time setting is too low	If necessary, set a higher discrepancy time
		Discrepancy time setting is < 0 s or > 3 s	Set discrepancy time in range between 0 s and 3 s
Bit 1	Discrepancy error or incorrect discrepancy time DISCTIM 2 setting for sensor pair 2	Same as Bit 0	Same as Bit 0
Bit 2	Maximum muting time exceeded or incorrect muting time TIME_MAX setting	Malfunction in production sequence	Malfunction in production sequence eliminated
		Maximum muting time setting is too low	If necessary, set a higher maximum muting time
		Muting time setting is < 0 s or > 10 min	Set muting time in range from 0 s to 10 min
Bit 3	Light curtain interrupted and muting not active	Light curtain is defective	Check light curtain
		Wiring fault	Check wiring of light curtain (FREE input)

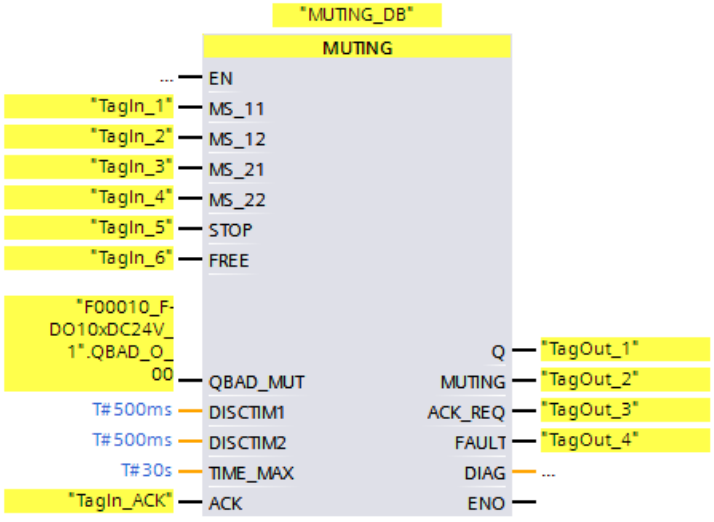
Bit no.	Assignment	Possible error causes	Remedies
Bit 3	Light curtain interrupted and muting not active	F-I/O fault, channel fault, or communication error, or passivation by means of PASS_ON of F-I/O of light curtain (FREE input)	For a solution, see the section "Structure of DIAG", bits 0 to 6 in DIAG (Page 171)
		See other DIAG bits	
Bit 4	Muting lamp is defective or cannot be set	Muting lamp is defective	Replace muting lamp
		Wiring fault	Check wiring of muting lamp
		F-I/O fault, channel fault, or communication error, or passivation by means of PASS_ON of F-I/O of muting lamp	For a solution, see the section "Structure of DIAG", bits 0 to 6 in DIAG (Page 171)
Bit 5	Reserved	—	—
Bit 6	Reserved	—	—
Bit 7	Reserved	—	—

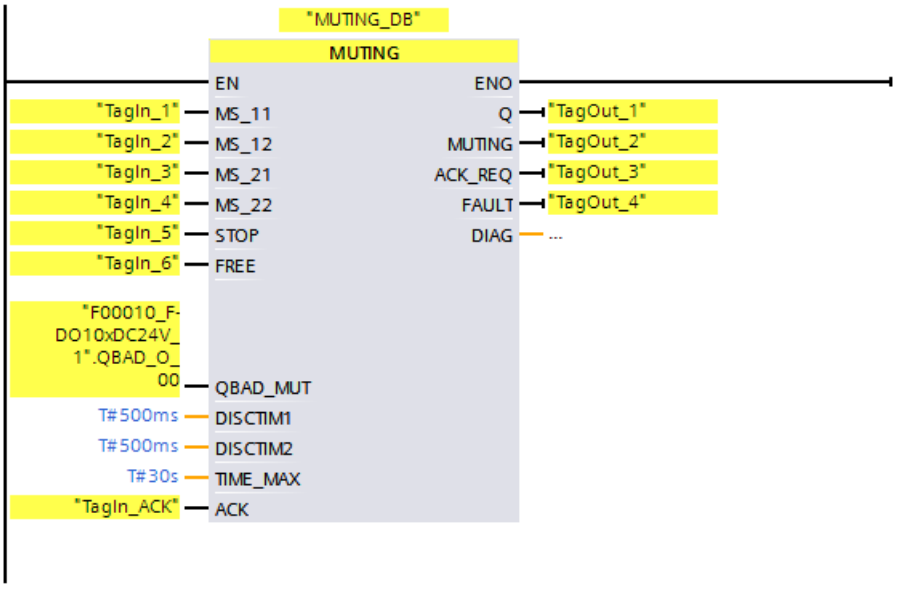
Timing imprecision resulting from the update time and the tolerance of the time base used in the instruction:

Refer to section "Temporal blurring for instructions with time processing (Page 450)" for more on this.

Example

The following example shows how the instruction works:





13.5.5 MUT_P: Parallel muting (STEP 7 Safety V21)

Description

This instruction performs parallel muting with two or four muting sensors.

Muting is a defined suppression of the protective function of light curtains. Light curtain muting can be used to introduce goods or objects into the danger area monitored by the light curtain without causing the machine to stop.

To utilize the muting function, at least two independently wired muting sensors must be present. The use of two or four muting sensors and correct integration into the production sequence must ensure that no persons enter the danger area while the light curtain is muted.

Every call of the "Parallel muting" instruction must be assigned a data area in which the instruction data are stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., MUT_P_DB_1) or a multi-instance (e.g., MUT_P_Instance_1) for the "Parallel muting" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
MS_11	Input	BOOL	Muting sensor 11
MS_12	Input	BOOL	Muting sensor 12
MS_21	Input	BOOL	Muting sensor 21
MS_22	Input	BOOL	Muting sensor 22
STOP	Input	BOOL	1=Conveyor system stopped
FREE	Input	BOOL	1=Light curtain uninterrupted
ENABLE	Input	BOOL	1=Enable MUTING
QBAD_MUT	Input	BOOL	QBAD signal of the F-I/O or QBAD_O_xx signal / inverted value status of the muting lamp channel
ACK	Input	BOOL	Acknowledgment of restart inhibit
DISCTIM1	Input	TIME	Discrepancy time of sensor pair 1 (0 to 3 s)
DISCTIM2	Input	TIME	Discrepancy time of sensor pair 2 (0 to 3 s)
TIME_MAX	Input	TIME	Maximum muting time (0 to 10 min)
Q	Output	BOOL	1= Enable, not off
MUTING	Output	BOOL	Display of muting is active
ACK_REQ	Output	BOOL	Acknowledgment necessary
FAULT	Output	BOOL	Group error
DIAG	Output	WORD	Non-fail safe service information

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x*	—	—	—	
1.1	x*	—	—	—	These versions have identical functions to version 1.0. The output DIAG can now be correctly interconnected with the operand of data type WORD.
1.2	x*	—	—	o	
1.3	x*	o	—	o	
1.4	x	x	x	x	

o This version is no longer supported.

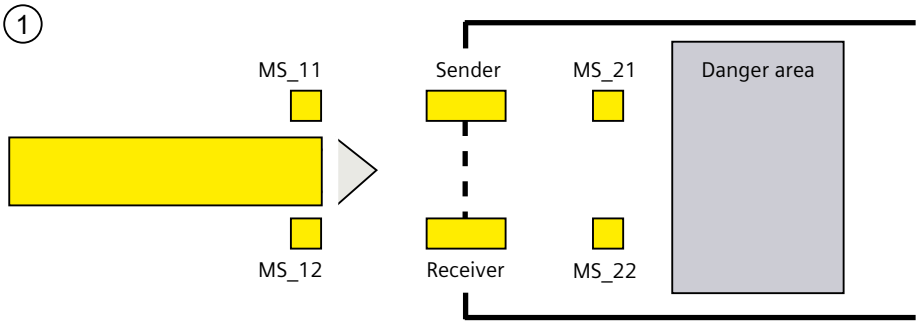
* S7-300/400: When a restart inhibit (output FAULT = 1) and ENABLE = 1 is present, output ACK_REQ is set to 1 even if not at least one muting sensor is activated. Use the DIAG bits 5 and 6 for more information.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Schematic sequence of error-free muting procedure with 4 muting sensors (MS_11, MS_12, MS_21, MS_22)

①



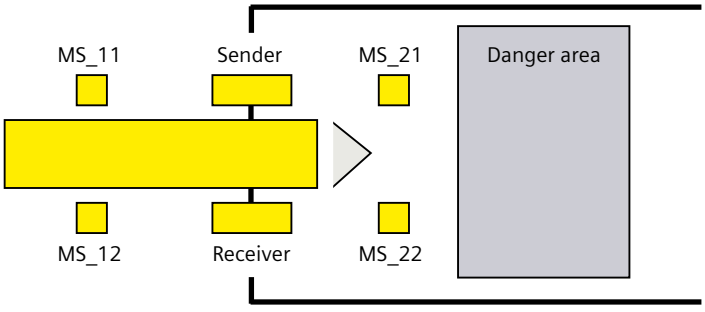
- If muting sensors MS_11 and MS_12 are both activated by the product within DISCTIM1 (apply signal state = 1) and MUTING is enabled by setting the ENABLE input to 1, the instruction starts the MUTING function. Enable signal Q remains 1, even when input FREE = 0 (light curtain interrupted by product). The MUTING output for setting the muting lamp switches to 1.

NOTE

The muting lamp can be monitored using the QBAD_MUT input. To do this, you must wire the muting lamp to an output with wire break monitoring of an F-I/O and supply the QBAD_MUT input with the QBAD signal of the associated F-I/O or the QBAD_O_xx signal / with inverted value statues of the associated channel. If QBAD_MUT = 1, muting is terminated by the instruction. If monitoring of the muting lamp is not necessary, you do not have to supply input QBAD_MUT.

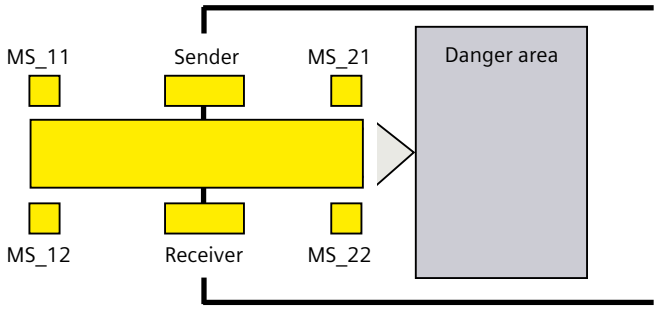
F-I/O that can promptly detect a wire break after activation of the muting operation must be used (*see manual for specific F-I/O*).

②



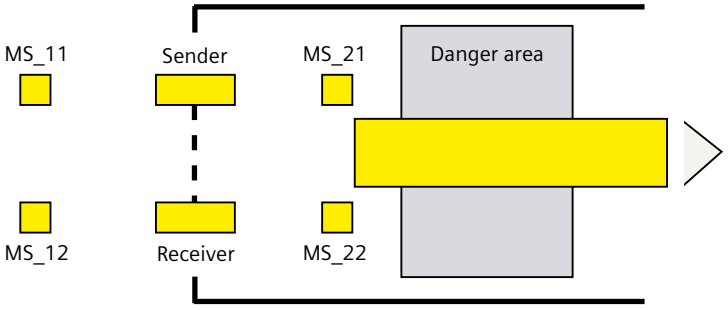
- As long as both muting sensors MS_11 and MS_12 continue to be activated, the MUTING function of the instruction causes Q to remain 1 and MUTING to remain 1 (so that the product can pass through the light curtain without causing the machine to stop). Each of the two muting sensors MS_11 and MS_12 may be switched to inactive (t < DISCTIM1) for a short time (apply signal state 0).

③



- Muting sensors MS_21 and MS_22 must both be activated (within DISCTIM2) before muting sensors MS_11 and MS_12 are switched to inactive (apply signal state 0). In this way, the instruction retains the MUTING function. (Q = 1, MUTING = 1).

④

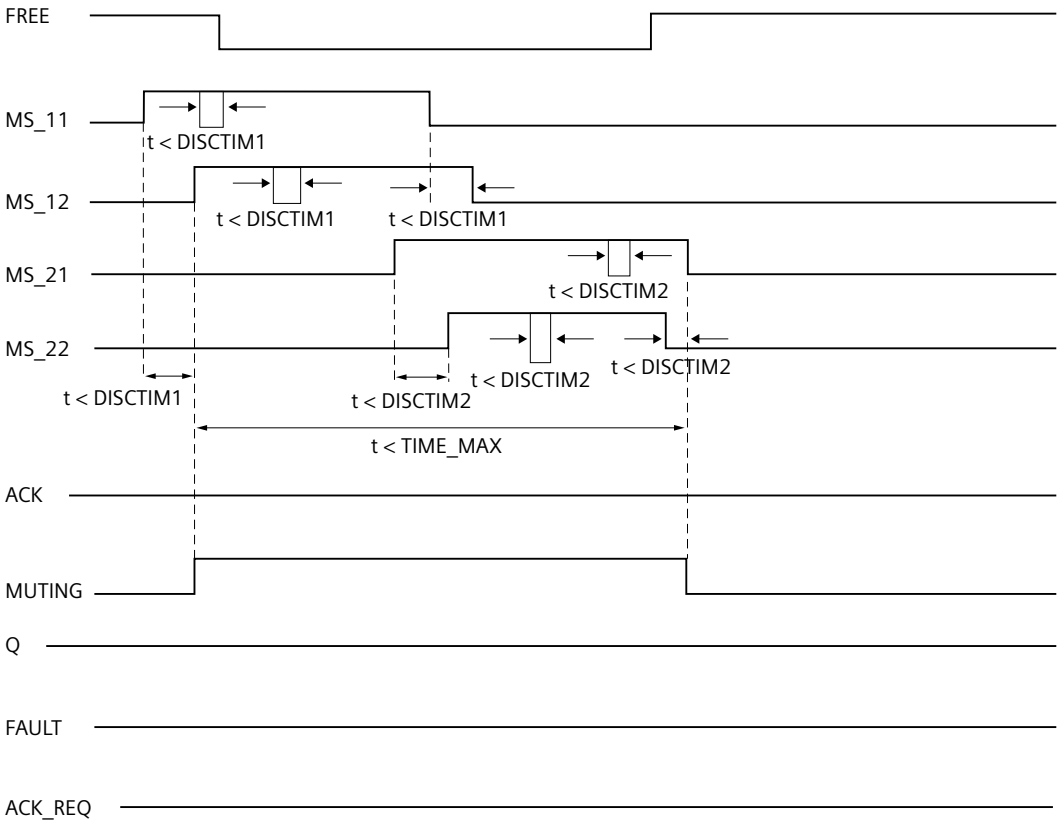


Only if muting sensors MS_21 and MS_22 are both switched to inactive (product enables sensors) is the MUTING function terminated (Q = 1, MUTING = 0). The maximum activation time for the MUTING function is the time set at input TIME_MAX.

NOTE

The MUTING function is also started if the product passes the light curtain in the reverse direction and the muting sensors are thus activated by the product in reverse order.

Timing diagrams for error-free muting procedure with 4 muting sensors

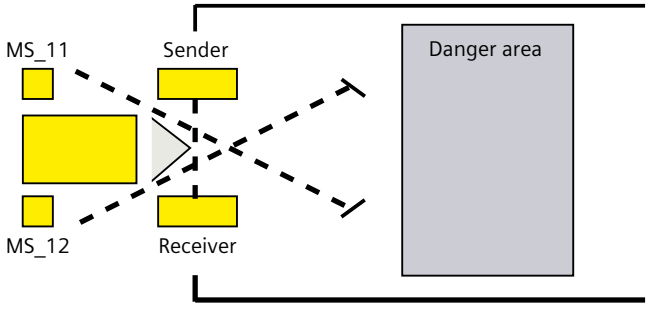


Schematic sequence of muting procedure with reflection light barriers

If reflection light barriers are used as muting sensors, they are generally arranged diagonally.

In general, this arrangement of reflection light barriers as muting sensors requires only two light barriers, and only MS_11 and MS_12 are interconnected.

The sequence is similar to that of the muting procedure with 4 muting sensors. Step 3 is omitted. In step 4, replace MS_21 and MS_22 with MS_11 and MS_12, respectively.



Restart inhibit upon interruption of light curtain (MUTING is not active), as well as when errors occur and during F-system startup

Enable signal Q cannot be set to 1 or becomes 0, if:

- Light curtain is interrupted (e.g., by a person or material transport) while the MUTING function is not active
- Light curtain is (being) interrupted and the muting lamp monitoring at input QBAD_MUT is set to 1
- Light curtain is (being) interrupted and the MUTING function is not enabled by setting input ENABLE to 1
- Sensor pair 1 (MS_11 and MS_12) or sensor pair 2 (MS_21 and MS_22) is not activated or deactivated during discrepancy time DISCTIM1 or DISCTIM2, respectively
- The MUTING function is active longer than the maximum muting time TIME_MAX
- Discrepancy times DISCTIM1 and DISCTIM2 have been set to values < 0 or > 3 s
- Maximum muting time TIME_MAX has been set to a value< 0 or > 10 min
- The F-system starts up (regardless of whether or not the light curtain is interrupted, because the F-I/O is passivated after F-system startup and, thus, the FREE input is initially supplied with 0)

In the identified cases, output FAULT (group error) is set to 1 (restart inhibit). If the MUTING function is started, it will be terminated and the Muting output becomes 0.

User acknowledgment of restart inhibit (no muting sensor is activated or ENABLE = 0)

Enable signal Q becomes 1 again, when:

- The light curtain is no longer interrupted
- If present, errors are eliminated (see output DIAG)

  and

- A user acknowledgment with positive edge occurs at input ACK (see also Implementation of user acknowledgment (Page 190)).

The FAULT output is set to 0. Output ACK_REQ = 1 (and DIAG bit 6) signals that user acknowledgment at input ACK is required to eliminate the restart inhibit. The instruction sets ACK_REQ = 1 as soon as the light curtain is no longer interrupted or the errors have been eliminated. Once acknowledgment has occurred, the instruction resets ACK_REQ to 0.

User Acknowledgment of restart inhibit (at least one muting sensor is activated and ENABLE = 1)

Enable signal Q becomes 1 again, when:

- If present, errors are eliminated (see output DIAG)
- FREE function occurs until a valid combination of muting sensors is detected

The FAULT output is set to 0. The MUTING function is restarted, if necessary, and the MUTING output becomes 1 if a valid combination of muting sensors is detected. When ENABLE = 1, output ACK_REQ = 1 (and DIAG-Bit 5) signals that FREE function is necessary for error

elimination and for removal of the restart inhibit. Once FREE function has occurred successfully, the instruction resets ACK_REQ to 0.

NOTE

Once the maximum muting time is exceeded, TIME_MAX is reset as soon as the MUTING function is restarted.

FREE function

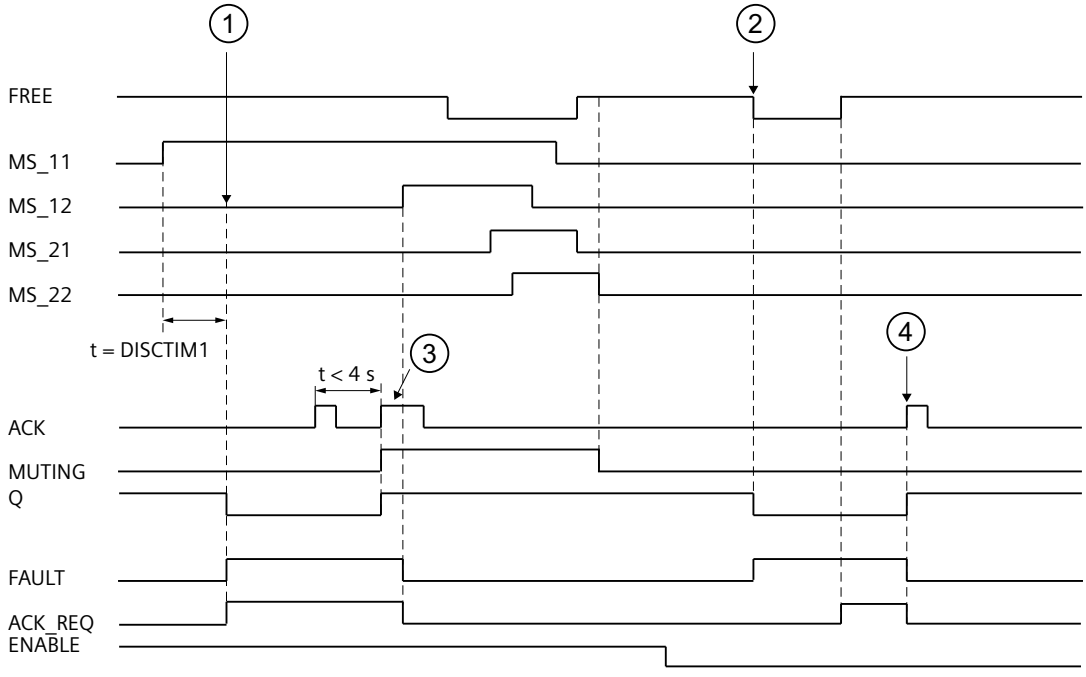
If an error cannot be corrected immediately, the FREE function can be used to free the muting range. Enable signal Q and output MUTING =1 temporarily. The FREE function can be used if:

- ENABLE = 1
- At least one muting sensor is activated
- A user acknowledgment with rising edge at input ACK occurs twice within 4 s, and the second user acknowledgment at input ACK remains at a signal state of 1 (acknowledgment button remains activated)

⚠ WARNING

When using the FREE function, the action must be observed. A dangerous situation must be able to be interrupted at any time by releasing the acknowledgment button. The acknowledgment button must be mounted in such a way that the entire danger area can be observed by the operator. *(S037)*

Timing diagrams for discrepancy errors at sensor pair 1 or interruption of the light curtain (MUTING is not active)



① Sensor pair 1 (MS_11 and MS_22) is not activated within discrepancy time DISCTIM1.

② The light curtain is interrupted even though there is no enable (ENABLE=0)

③ FREE function

④ Acknowledgment

Behavior with stopped conveyor equipment

If, while the conveyor equipment has stopped, the monitoring for one of the following reasons is to be deactivated:

- To comply with discrepancy time DISCTIM1 or DISCTIM2
- To comply with maximum muting time TIME_MAX

You must supply input STOP with a "1" signal for as long as the conveyor equipment is stopped. As soon as the conveyor equipment is running again (STOP = 0), discrepancy times DISCTIM1 and DISCTIM2 and maximum muting time TIME_MAX are reset.

⚠ WARNING

When STOP = 1 or ENABLE = 0, discrepancy monitoring is shut down. During this time, if inputs MSx1/MSx2 of a sensor pair both take a signal state of 1 due to an undetected error, for example, because both muting sensors have adopted signal state 1 in the event of an error, the fault is not detected and the MUTING function can be started unintentionally (when ENABLE =1). *(S038)*

Output DIAG

The DIAG output provides non-fail-safe information on errors for service purposes. You can read out this information by means of operator control and monitoring systems or, if applicable, you can evaluate it in your standard user program. DIAG bits 0 to 6 are saved until acknowledgment at input ACK.

Structure of DIAG

Bit no.	Assignment	Possible error causes	Remedies
Bit 0	Discrepancy error or incorrect discrepancy time DISCTIM 1 setting for sensor pair 1	Malfunction in production sequence	Malfunction in production sequence eliminated
		Sensor defective	Check sensors
		Wiring fault	Check wiring of sensors
		Sensors are wired to different F-I/O, and F-I/O fault, channel fault, or communication error, or passivation by means of PASS_ON on an F-I/O	For a solution, see the section "Structure of DIAG", bits 0 to 6 in DIAG (Page 171)
		Discrepancy time setting is too low	If necessary, set a higher discrepancy time
		Discrepancy time setting is < 0 s or > 3 s	Set discrepancy time in range between 0 s and 3 s
Bit 1	Discrepancy error or incorrect discrepancy time DISCTIM 2 setting for sensor pair 2	Same as Bit 0	Same as Bit 0
Bit 2	Maximum muting time exceeded or incorrect muting time TIME_MAX setting	Malfunction in production sequence	Malfunction in production sequence eliminated
		Maximum muting time setting is too low	If necessary, set a higher maximum muting time
		Muting time setting is < 0 s or > 10 min	Set muting time in range from 0 s to 10 min
Bit 3	Light curtain interrupted and muting not active	ENABLE = 0	Set ENABLE = 1
		Light curtain is defective	Check light curtain
		Wiring fault	Check wiring of light curtain (FREE input)

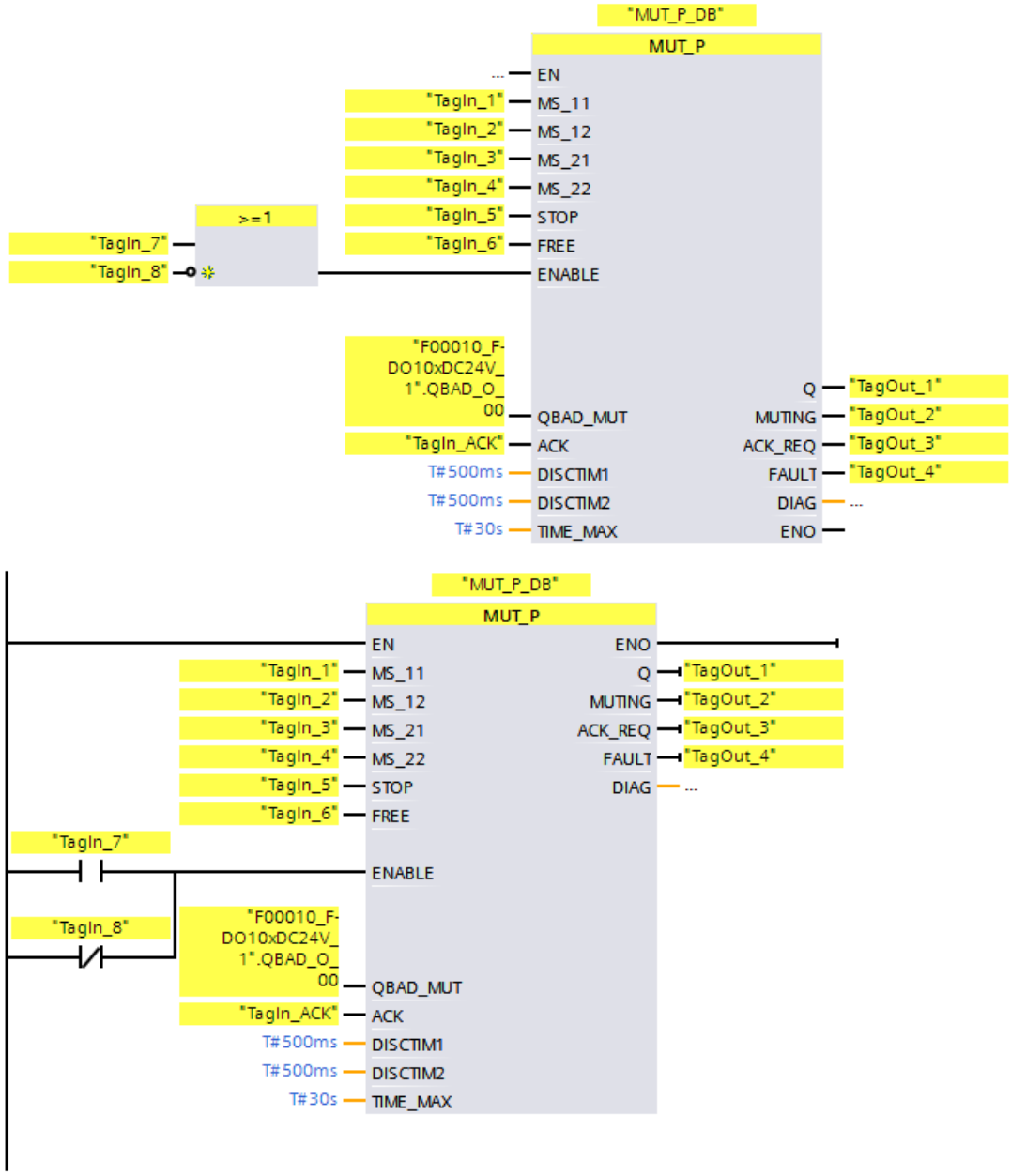
Bit no.	Assignment	Possible error causes	Remedies
Bit 3	Light curtain interrupted and muting not active	F-I/O fault, channel fault, or communication error, or passivation by means of PASS_ON of F-I/O of light curtain (FREE input)	For a solution, see the section "Structure of DIAG", bits 0 to 6 in DIAG (Page 171)
		Startup of F-system	For FREE function, see DIAG Bit 5
		See other DIAG bits	
Bit 4	Muting lamp is defective or cannot be set	Muting lamp is defective	Replace muting lamp
		Wiring fault	Check wiring of muting lamp
		F-I/O fault, channel fault, or communication error, or passivation by means of PASS_ON of F-I/O of muting lamp	For a solution, see the section "Structure of DIAG", bits 0 to 6 in DIAG (Page 171)
Bit 5	FREE function is necessary	See other DIAG bits	Two rising edges at ACK within 4 s, and activate acknowledgment button until ACK_REQ = 0
Bit 6	Acknowledgment necessary	—	—
Bit 7	State of output Q	—	—
Bit 8	State of output MUTING	—	—
Bit 9	FREE function active	—	—
Bit 10	Reserved	—	—
...			
Bit 15	Reserved	—	—

Timing imprecision resulting from the update time and the tolerance of the time base used in the instruction

Refer to section "Temporal blurring for instructions with time processing (Page 450)" for more on this.

Example

The following example shows how the instruction works:



13.5.6 EV1oo2DI: 1oo2 evaluation with discrepancy analysis (STEP 7 Safety V21)

Description

This instruction implements a 1oo2 evaluation of two single-channel sensors combined with a discrepancy analysis.

Output Q is set to 1, if the signal states of inputs IN1 and IN2 both equal 1 and no discrepancy error DISC_FLT is stored. if the signal state of one or both inputs is 0, output Q is set to 0.

As soon as the signal states of inputs IN1 and IN2 are different, the discrepancy time DISCTIME is started. If the signal states of the two inputs are still different once the discrepancy time expires, a discrepancy error is detected and DISC_FLT is set to 1 (restart inhibit).

If the discrepancy between inputs IN1 and IN2 is no longer detected, the discrepancy error is acknowledged according to the parameter assignment of ACK_NEC:

- If ACK_NEC = 0, the acknowledgment is automatic.
- If ACK_NEC = 1, you must use a rising edge at input ACK to acknowledge the discrepancy error.

The output ACK_REQ = 1 signals that a user acknowledgment at input ACK is necessary to acknowledge the discrepancy error (cancel the restart inhibit). The instruction sets ACK_REQ = 1 as soon as discrepancy is no longer detected. After acknowledgment or if, prior to acknowledgment, there is once again a discrepancy between inputs IN1 and IN2, the instruction resets ACK_REQ to 0.

Output Q can never be set to 1 if the discrepancy time setting is < 0 or > 60 s. In this case, output DISC_FLT is also set to 1 (restart inhibit). The call interval of the safety program (e.g., OB 35) must be less than the discrepancy time setting.

Every call of the "1oo2 evaluation with discrepancy analysis" instruction must be assigned a data area in which the instruction data is stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., EV1oo2DI_DB_1) or a multi-instance (e.g., EV1oo2DI_Instance_1) for the "1oo2 evaluation with discrepancy analysis" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

⚠ WARNING

The ACK_NEC tag must not be assigned a value of "0" unless an automatic restart of the affected process is otherwise excluded. *(S033)*

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN1	Input	BOOL	Sensor 1
IN2	Input	BOOL	Sensor 2
DISCTIME	Input	TIME	Discrepancy time (0 to 60 s)
ACK_NEC	Input	BOOL	1 = acknowledgment necessary for discrepancy error
ACK	Input	BOOL	Acknowledgment of discrepancy error
Q	Output	BOOL	Output
ACK_REQ	Output	BOOL	1 = acknowledgment required
DISC_FLT	Output	BOOL	1 = discrepancy error
DIAG	Output	BYTE	Non-fail safe service information

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	x	—	—	o	These versions have identical functions to version 1.0.
1.2	x	o	—	o	
1.3	x	x	x	x	

o This version is no longer supported.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

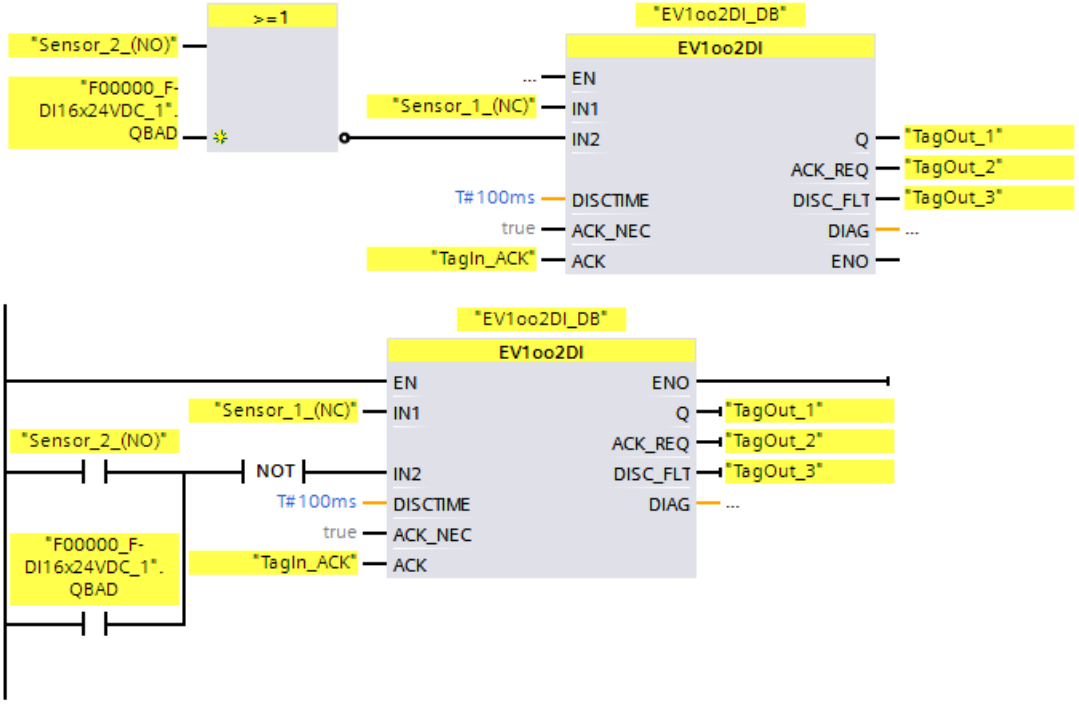
For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Activating inputs IN1 and IN2

Inputs IN1 and IN2 must both be activated in such a way that their safe state is 0.

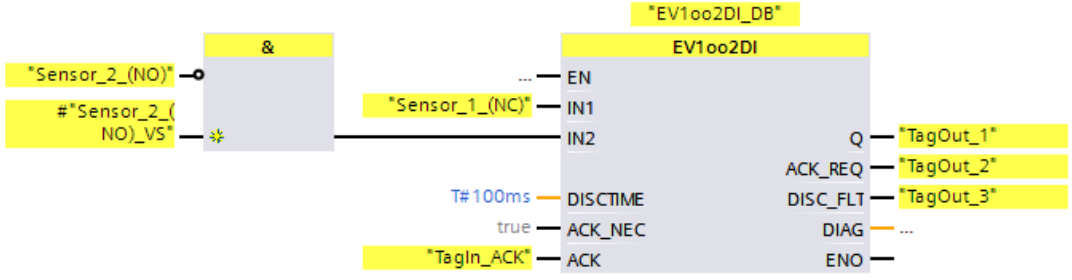
Example with QBAD or QBAD_I_xx signal

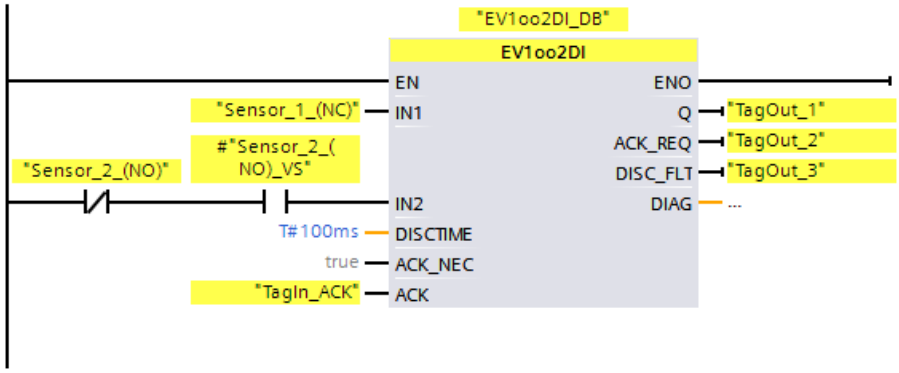
For non-equivalent signals you need to OR the input (IN1 and IN2) with which you assign the encoder signal to the safe state 1, with the QBAD signal of the associated F-I/O or the QBAD_I_xx signal of the associated channel (with S7-300/400 F-CPUs) and negate the result. Signal state 0 is then at input IN1 or IN2 when fail-safe values are output.



Example with value status

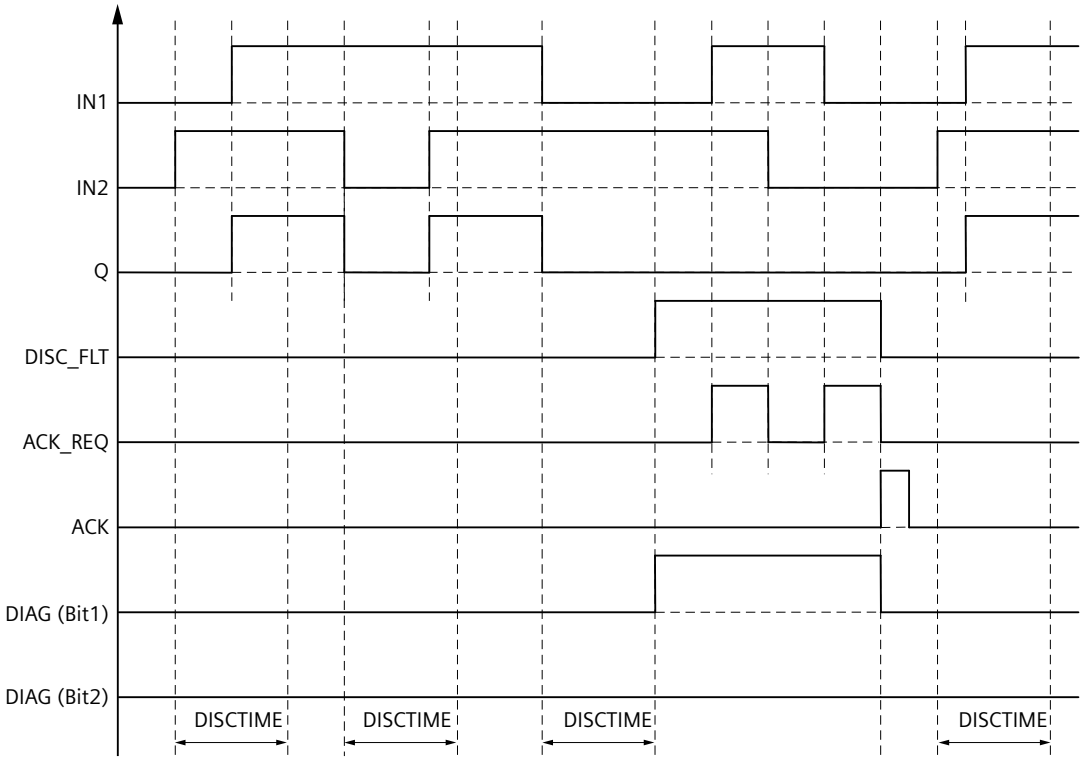
For non-equivalent signals, you have to negate the input (IN1 or IN2) with which you have assigned the encoder signal to a safe state of 1 and AND it with the value status of the associated channel. Signal state 0 is then at input IN1 or IN2 when fail-safe values are output.





Timing diagrams EV1oo2DI

If ACK_NEC = 1:



Startup behavior

NOTE

If the sensors at inputs IN1 and IN2 are assigned to different F-I/O, it is possible that the fail-safe values are output for different lengths of time following startup of the F-system due to different startup behavior of the F-I/O. If the signal states of inputs IN1 and IN2 remain different after the discrepancy time DISCTIME has expired, a discrepancy error is detected after the F-system starts up.

If ACK_NEC = 1 you must acknowledge the discrepancy error with a rising edge at input ACK.

Output DIAG

The DIAG output provides non-fail-safe information on errors for service purposes. You can read out this information by means of operator control and monitoring systems or, if applicable, you can evaluate it in your standard user program. DIAG bits are saved until acknowledgment at input ACK.

Structure of DIAG

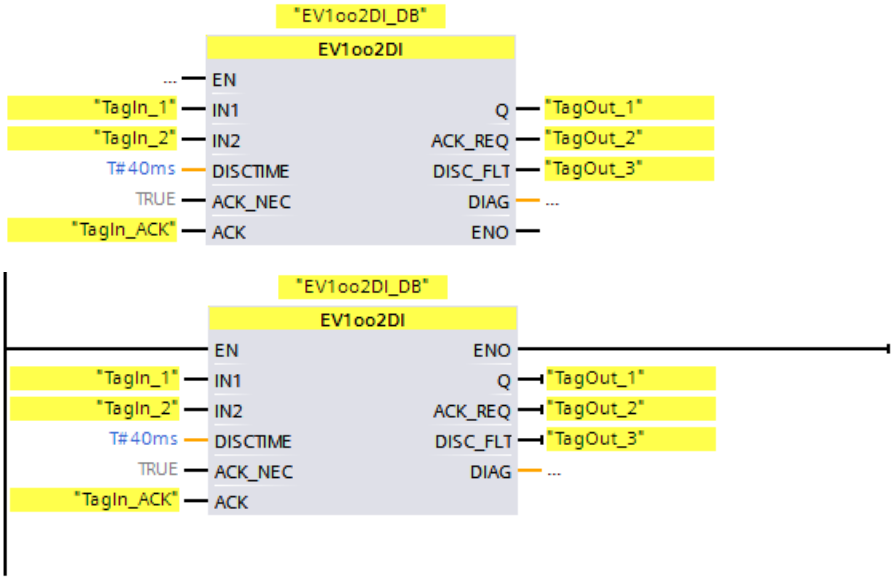
Bit no.	Assignment	Possible error causes	Remedies
Bit 0	Discrepancy error or incorrect discrepancy time setting (= status of DISC_FLT)	Sensor defective	Check sensors
		Wiring fault	Check wiring of sensors
		Sensors are wired to different F-I/O, and F-I/O fault, channel fault, or communication error, or passivation by means of PASS_ON on an F-I/O	For a solution, see the section "Structure of DIAG", bits 0 to 6 in DIAG (Page 171)
		Discrepancy time setting is too low	If necessary, set a higher discrepancy time
		Discrepancy time setting is < 0 s or > 60 s	Set discrepancy time in range between 0 s and 60 s
Bit 1	For discrepancy errors: last signal state change was at input IN1	—	—
Bit 2	For discrepancy errors: last signal state change was at input IN2	—	—
Bit 3	Reserved	—	—
Bit 4	Reserved	—	—
Bit 5	For discrepancy errors: input ACK has a permanent signal state of 1	Acknowledgment button defective	Replace acknowledgment button
		Wiring fault	Check wiring of acknowledgment button
Bit 6	Acknowledgment necessary	—	—
Bit 7	State of output Q	—	—

Timing imprecision resulting from the update time and the tolerance of the time base used in the instruction

Refer to section "Temporal blurring for instructions with time processing (Page 450)" for more on this.

Example

The following example shows how the instruction works:



13.5.7 FDBACK: Feedback monitoring (STEP 7 Safety V21)

Description

This instruction implements feedback monitoring.

The signal state of output Q is checked to see whether it corresponds to the inverse signal state of the feedback input FEEDBACK.

Output Q is set to 1 as soon as input ON = 1. Requirement for this is that the feedback input FEEDBACK = 1 and no feedback error is saved.

Output Q is reset to 0, as soon as input ON = 0 or if a feedback error is detected.

A feedback error ERROR = 1 is detected if the inverse signal state of the feedback input FEEDBACK (to input Q) does not follow the signal state of output Q within the maximum tolerable feedback time. The feedback error is saved.

If a discrepancy is detected between the feedback input FEEDBACK and the output Q after a feedback error, the feedback error is acknowledged in accordance with the parameter assignment of ACK_NEC:

- If ACK_NEC = 0, the acknowledgment is automatic.
- If ACK_NEC = 1, you must acknowledge the feedback error with a rising edge at input ACK.

The ACK_REQ = 1 output then signals that a user acknowledgment is necessary at input ACK to acknowledge the feedback error. Following an acknowledgment, the instruction resets ACK_REQ to 0.

To avoid a feedback errors from being detected and acknowledgment from being required when the F-I/O controlled by the Q output are passivated, you need to supply input QBAD_FIO with the QBAD signal of the associated F-I/O or the QBAD_O_xx signal / inverted value status of the associated channel.

Every call of the "Feedback monitoring" instruction must be assigned a data area in which the instruction data is stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., FDBACK_DB_1) or a multi-instance (e.g., FDBACK_Instance_1) for the "Feedback monitoring" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

⚠ WARNING

The ACK_NEC tag must not be assigned a value of "0" unless an automatic restart of the affected process is otherwise excluded. *(S033)*

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
ON	Input	BOOL	1= Enable output
FEEDBACK	Input	BOOL	Feedback input
QBAD_FIO	Input	BOOL	QBAD signal of the F-I/O or QBAD_O_xx signal / inverted value status of the Q output
ACK_NEC	Input	BOOL	1=Acknowledgment necessary
ACK	Input	BOOL	Acknowledgment
FDB_TIME	Input	TIME	Feedback time
Q	Output	BOOL	Output
ERROR	Output	BOOL	Feedback error
ACK_REQ	Output	BOOL	Acknowledgment request
DIAG	Output	BYTE	Non-fail safe service information

Instruction versions

A number of versions are available for this instruction:

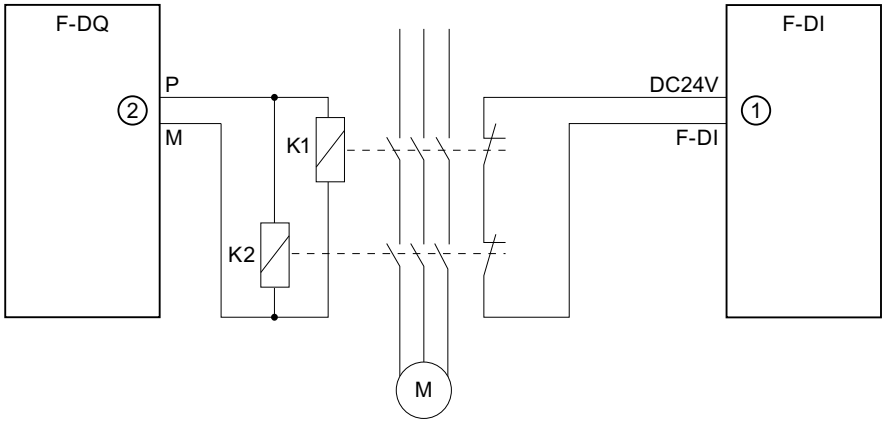
Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	Version 1.0 requires that the F_TOF block with the number FB 186 is available in the project tree in the "Program blocks/System blocks/STEP 7 Safety" folder.
1.1	x	—	—	—	These versions have identical functions to version 1.0, but do not require the F_TOF block to have a particular number.
1.2	x	—	—	o	
1.3	x	o	—	o	
1.4	x	o	—	o	
1.5	x	x	x	x	

o This version is no longer supported.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Interconnection example



① Sent to the FEEDBACK input of the instruction

② from output Q of the instruction

Output DIAG

The DIAG output provides non-fail-safe information on errors for service purposes. You can read out this information by means of operator control and monitoring systems or, if applicable, you can evaluate it in your standard user program. DIAG bits 0, 2, and 5 are saved until acknowledgment at input ACK.

Structure of DIAG

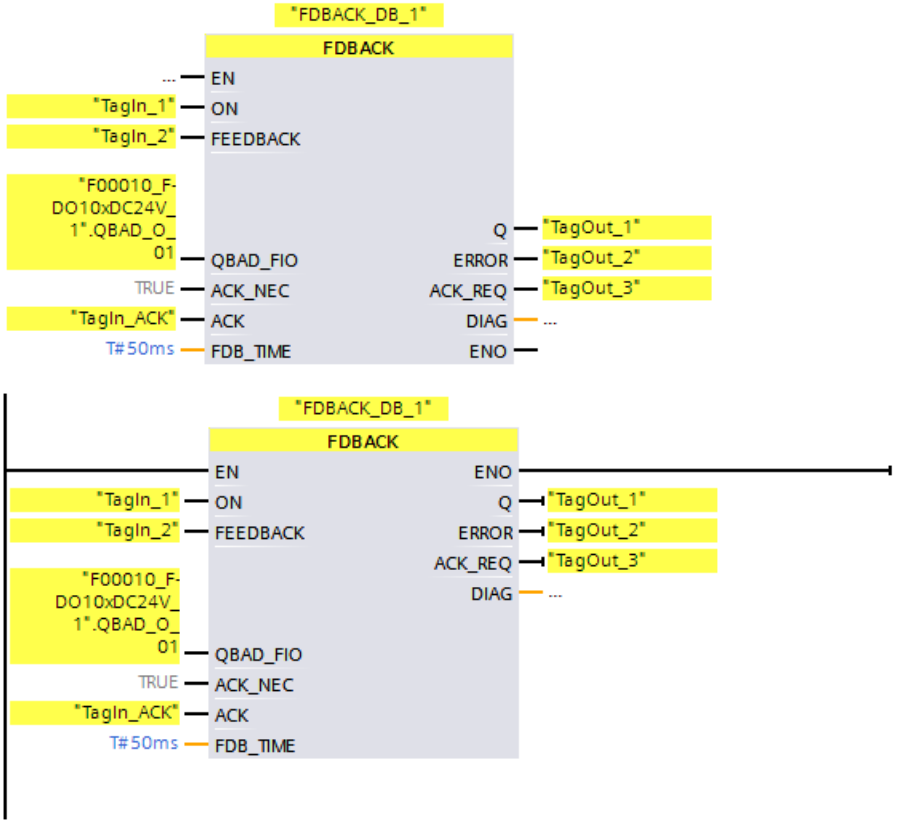
Bit no.	Assignment	Possible error causes	Remedies
Bit 0	Feedback error or incorrect feedback time setting (= state of ERROR)	Feedback time setting < 0	Set feedback time > 0
		Feedback time setting is too low	If necessary, set a higher feedback time
		Wiring fault	Check wiring of actuator and feedback contact
		Actuator or feedback contact is defective	Check actuator and feedback contact
		I/O fault or channel fault of feedback input	Check I/O
Bit 1	Passivation of F-I/O/channel controlled by output Q (= state of QBAD_FIO)	F-I/O fault, channel fault, or communication error, or passivation by means of PASS_ON of F-I/O	For a solution, see the section "Structure of DIAG", bits 0 to 6 in DIAG (Page 171)
Bit 2	After feedback error: feedback input has permanent signal state of 0	F-I/O fault or channel fault of feedback input	Check I/O
		Feedback contact is defective	Check feedback contact
		F-I/O fault, channel fault, or communication error, or passivation by means of PASS_ON of F-I/O of feedback input	For a solution, see the section "Structure of DIAG", bits 0 to 6 in DIAG (Page 171)
Bit 3	Reserved	—	—
Bit 4	Reserved	—	—
Bit 5	For feedback error: input ACK has a permanent signal state of 1	Acknowledgment button defective	Check acknowledgment button
		Wiring fault	Check wiring of acknowledgment button
Bit 6	Acknowledgment required (= state of ACK_REQ)	—	—
Bit 7	State of output Q	—	—

Timing imprecision resulting from the update time and the tolerance of the time base used in the instruction

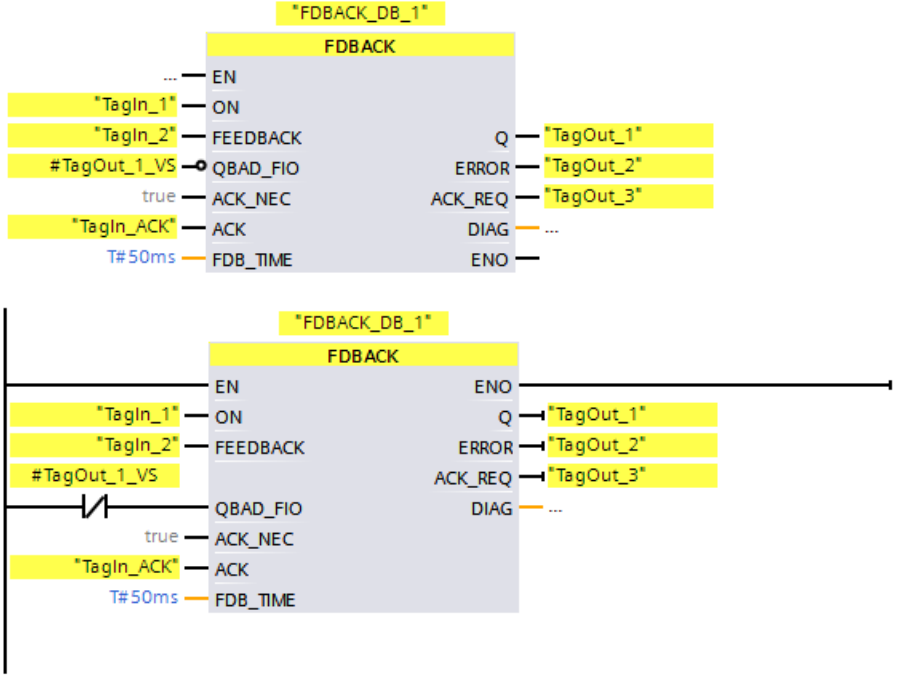
Refer to section "Temporal blurring for instructions with time processing (Page 450)" for more on this.

Example

The following example shows how the instruction for S7-300/400 F-CPUs works:



The following example shows how the instruction for S7-1200/1500 F-CPUs works:



13.5.8 SFDOOR: Safety door monitoring (STEP 7 Safety V21)

Description

This instruction implements safety door monitoring.

Enable signal Q is reset to 0 as soon as one of the inputs IN1 or IN2 take a signal state of 0 (safety door is opened). The enable signal can be reset to 1, only if:

- Inputs IN1 and IN2 both take a signal state of 0 prior to opening the door (safety door has been completely opened)
- Inputs IN1 and IN2 then both take a signal state of 1 (safety door is closed)
- An acknowledgment occurs

The acknowledgment for the enable takes place according to the parameter assignment at input ACK_NEC:

- If ACK_NEC = 0, the acknowledgment is automatic.
- If ACK_NEC = 1, you must use a rising edge at input ACK for acknowledging the enable.

Output ACK_REQ = 1 is used to signal that a user acknowledgment is required at input ACK for the acknowledgment. The instruction sets ACK_REQ = 1 as soon as the door is closed. Following an acknowledgment, the instruction resets ACK_REQ to 0.

In order for the instruction to recognize whether inputs IN1 and IN2 are 0 merely due to passivation of the associated F-I/O, you need to supply inputs QBAD_IN1 or QBAD_IN2 with the QBAD signal of the associated F-I/O or QBAD_I_xx signal / inverted value status of the associated channel. Among other things, this will prevent you from having to open the safety door completely prior to an acknowledgment in the event the F-I/O are passivated.

Every call of the "Safety door monitoring" instruction must be assigned a data area in which the instruction data are stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., SFDOOR_DB_1) or a multi-instance (e.g., SFDOOR_Instance_1) for the "Safety door monitoring" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

⚠ WARNING

The ACK_NEC tag must not be assigned a value of "0" unless an automatic restart of the affected process is otherwise excluded. *(S033)*

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN1	Input	BOOL	Input 1
IN2	Input	BOOL	Input 2
QBAD_IN1	Input	BOOL	QBAD signal of the F-I/O or QBAD_O_xx signal / inverted value status of the channel of input IN1
QBAD_IN2	Input	BOOL	QBAD signal of the F-I/O or QBAD_O_xx signal / inverted value status of the channel of input IN2
OPEN_NEC	Input	BOOL	1= Open necessary at startup
ACK_NEC	Input	BOOL	1=Acknowledgment necessary
ACK	Input	BOOL	Acknowledgment
Q	Output	BOOL	1= Enable, safety door closed
ACK_REQ	Output	BOOL	Acknowledgment request
DIAG	Output	BYTE	Non-fail safe service information

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	x	—	—	o	These versions have identical functions to version 1.0.
1.2	x	o	—	o	
1.3	x	x	x	x	

o This version is no longer supported.

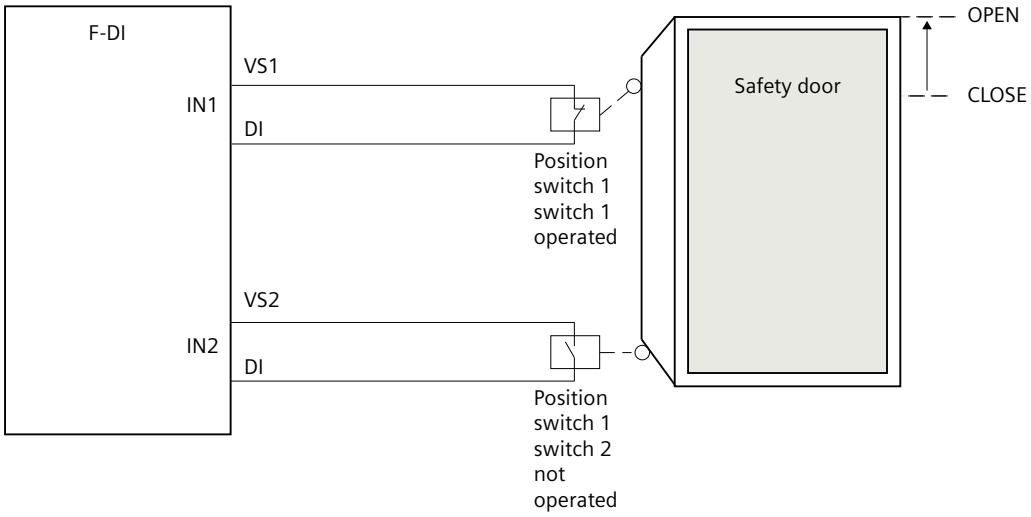
When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

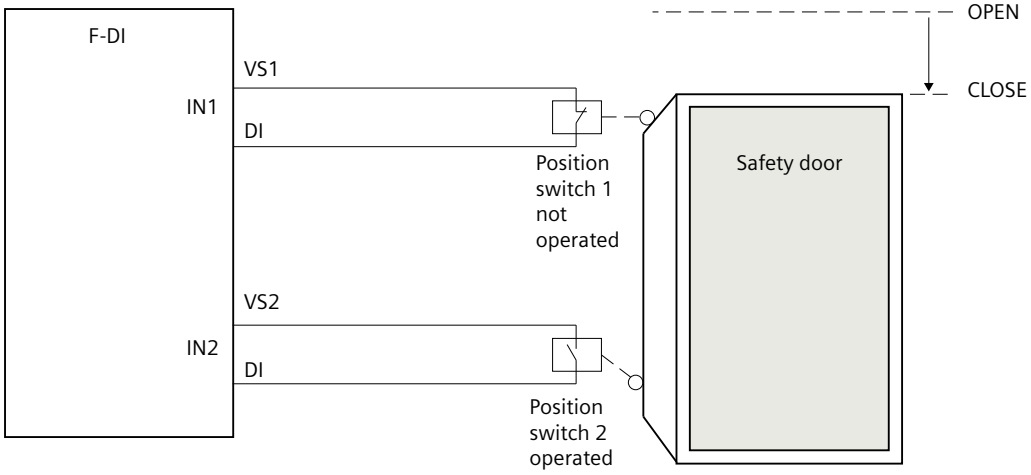
Interconnection example

You must interconnect the NC contact of position switch 1 of the safety door at input IN1 and the NO contact of position switch 2 at input IN2. Position switch 1 must be mounted in such a way that it is positively operated when the safety door is open. Position switch 2 must be mounted in such a way that it is operated when the safety door is closed.

Safety door open:



Safety door closed:



Startup behavior

After an F-system startup, enable signal Q is reset to 0. The acknowledgment for the enable takes place according to the parameter assignment at inputs OPEN_NEC and ACK_NEC:

- When OPEN_NEC = 0, an automatic acknowledgment occurs **independently** of ACK_NEC, as soon as the two inputs IN1 and IN2 take signal state 1 for the first time following reintegration of the associated F-I/O (safety door is closed).
- When OPEN_NEC = 1 **or** if at least one of the IN1 and IN2 inputs still has a signal state of 0 after reintegration of the associated F-I/O, an automatic acknowledgment occurs **according** to ACK_NEC or you have to use a rising edge at input ACK for the enable. Prior to acknowledgment, inputs IN1 and IN2 both have to take a signal state of 0 (safety door has been completely opened) followed by a signal state of 1 (safety door is closed).

⚠ WARNING

The OPEN_NEC tag must not be assigned a value of "0" unless an automatic restart of the affected process is otherwise excluded. *(S039)*

Output DIAG

The DIAG output provides non-fail-safe information on errors for service purposes. You can read out this information by means of operator control and monitoring systems or, if applicable, you can evaluate it in your standard user program.

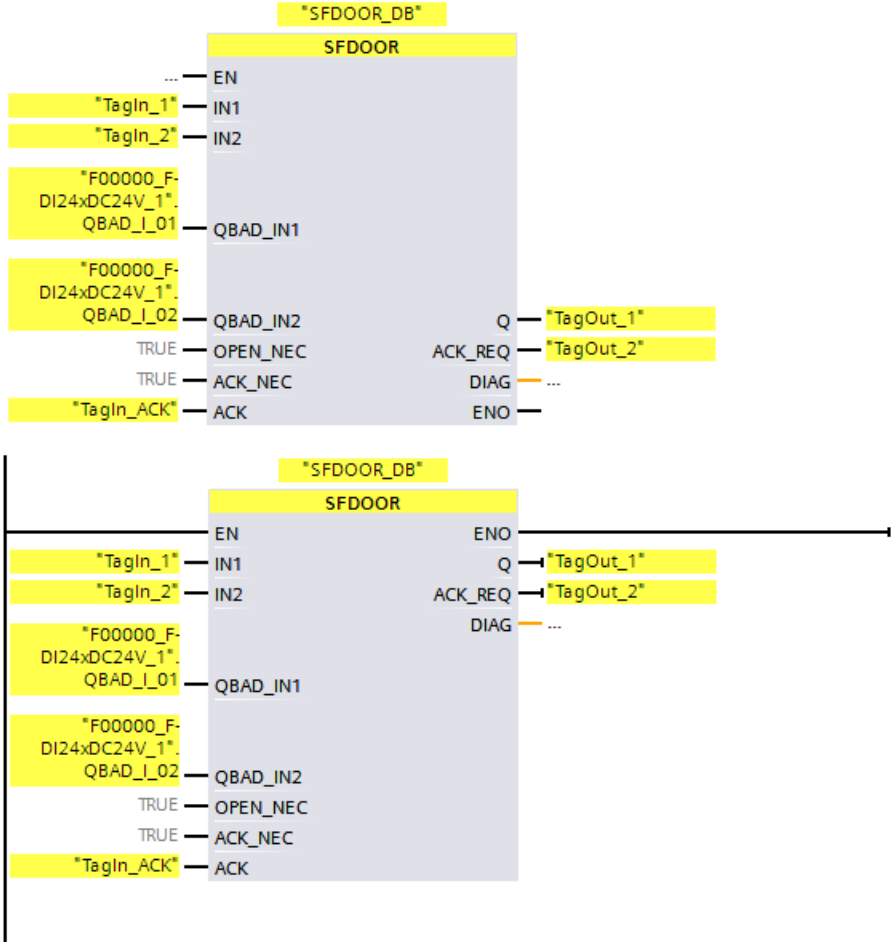
Structure of DIAG

Bit no.	Assignment	Possible error causes	Remedies
Bit 0	Reserved	—	—
Bit 1	Signal state 0 is missing at both IN1 and IN2 inputs	Safety door was not completely opened when OPEN_NEC = 1 after F-system startup	Open safety door completely
		Open safety door was not completely opened	Open safety door completely
		Wiring fault	Check wiring of position switch
		Position switch is defective	Check position switch
		Position switch is incorrectly adjusted	Adjust position switch properly
Bit 2	Signal state 1 is missing at both IN1 and IN2 inputs	Safety door was not closed	Close safety door
		Wiring fault	Check wiring of position switch
		Position switch is defective	Check position switch
		Position switch is incorrectly adjusted	Adjust position switch properly

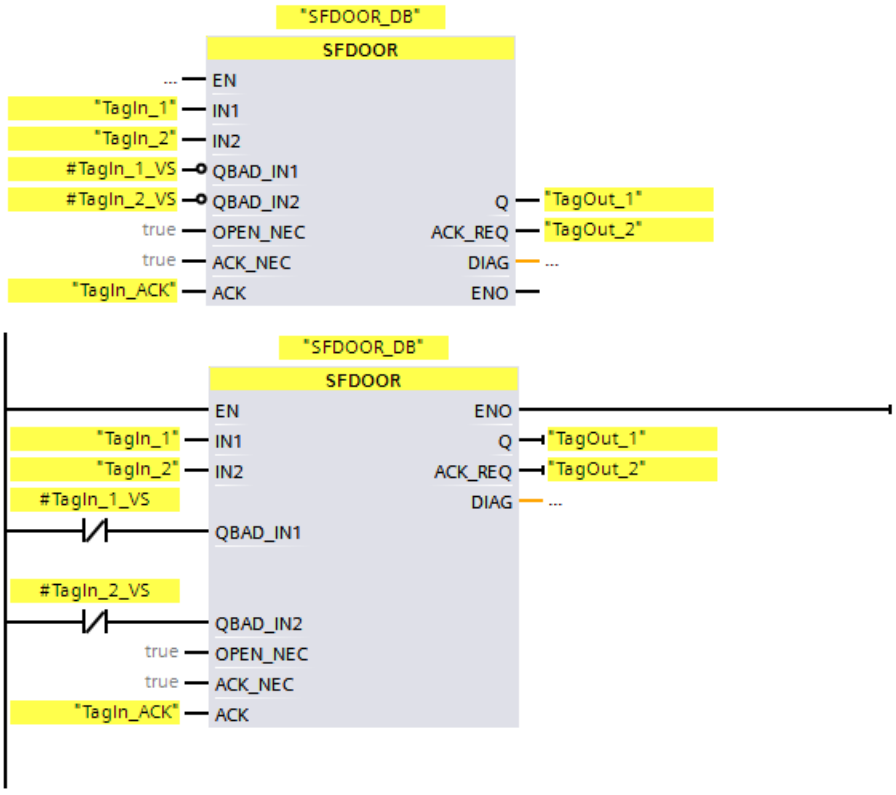
Bit no.	Assignment	Possible error causes	Remedies
Bit 3	QBAD_IN1 and/or QBAD_IN2 = 1	F-I/O fault, channel fault, or communication error, or passivation by means of PASS_ON of F-I/O or channel of IN1 and/or IN2	For a solution, see the section "Structure of DIAG", bits 0 to 6 in DIAG (Page 171)
Bit 4	Reserved	—	—
Bit 5	If enable is missing: input ACK has a permanent signal state of 1	Acknowledgment button defective	Check acknowledgment button
		Wiring fault	Check wiring of acknowledgment button
Bit 6	Acknowledgment required (= state of ACK_REQ)	—	—
Bit 7	State of output Q	—	—

Example

The following example shows how the instruction for S7-300/400 F-CPUs works:



The following example shows how the instruction for S7-1200/1500 F-CPUs works:



13.5.9 ACK_GL: Global acknowledgment of all F-I/O in an F-runtime group (STEP 7 Safety V21)

Description

This instruction creates an acknowledgment for the simultaneous reintegration of all F-I/O or channels of the F-I/O of an F-runtime group after communication errors, F-I/O errors, or channel faults.

A user acknowledgment (Page 190) with a positive edge at input ACK_GLOB is required for reintegration. The acknowledgment occurs analogously to the user acknowledgment via the ACK_REI tag of the F-I/O DB (Page 167), but it acts simultaneously on all F-I/O of the F-runtime group in which the instruction is called.

If you use the instruction ACK_GL, you do not have to provide a user acknowledgment for each F-I/O of the F-runtime group via the ACK_REI tag of the F-I/O DB.

Every call of the "Global acknowledgment of all F-I/O of a runtime group" instruction must be assigned a data area in which the instruction data are stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., ACK_GL_DB_1) or a multi-instance (e.g., ACK_GL_Instance_1) for the "Global acknowledgment of all F-I/O of a runtime group" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

NOTE

An acknowledgment via the ACK_GL instruction is only possible if the tag ACK_REI of the F-I/O DB = 0. Accordingly, an acknowledgment via the tag ACK_REI of the F-I/O DB is only possible if the input ACK_GLOB of the instruction = 0.

The instruction is only allowed to be called once per F-runtime group.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
ACK_GLOB	Input	BOOL	1=acknowledgment for reintegration

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	x	—	—	o	These versions have identical functions to version 1.0.
1.2	x	o	—	o	
1.3	x	x	x	x	

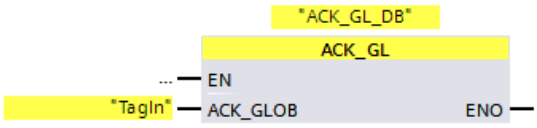
o This version is no longer supported.

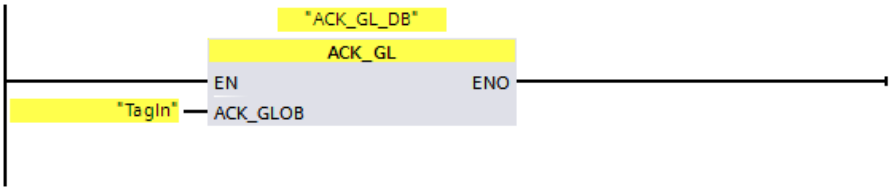
When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Example

The following example shows how the instruction works:





13.6 Timer operations

13.6.1 TP: Generate pulse (STEP 7 Safety V21)

Description

You can use the "Generate pulse" instruction to set output Q for a programmed period. The instruction is started if the result of logic operation (RLO) changes from "0" to "1" (positive edge) at input IN. The programmed period PT starts running when the instruction starts. Output Q is set for period PT, regardless of the subsequent sequence of the input signal. Also the detection of a new positive signal edge does not influence the signal state at output Q as long as period PT runs.

You can query the current time value at the output ET. The time value begins at T#0s and ends when the value of period PT is reached. If period PT is reached and the signal state at input IN is "0", output ET is reset.

Every call of the "Generate pulse" instruction must be assigned a data area in which the instruction data is stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., F_IEC_Timer_DB_1) or a multi-instance (e.g., F_IEC_Timer_Instance_1) for the "Generate pulse" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the *help on STEP 7*.

The operating system resets the instances of the "Generate pulse" instruction on a startup of the F-system.

NOTE

The functionality of this instruction differs from the corresponding standard TP instruction in the following points:

- If the instruction is called while the time is running with PT = 0 ms, the outputs Q and ET are reset.
- If the instruction is called with PT < 0 ms, the outputs Q and ET are reset.

To restart the pulse, a new rising signal edge at input IN is required once PT is greater than 0 again.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN	Input	BOOL	Start input
PT	Input	TIME	Duration of pulse; must be positive.
Q	Output	BOOL	Pulse output
ET	Output	TIME	Current time value

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	x	—	—	o	These versions have identical functions to version 1.0.
1.2	x	o	—	o	
1.3	x	o	—	o	
1.4	x	x	x	x	

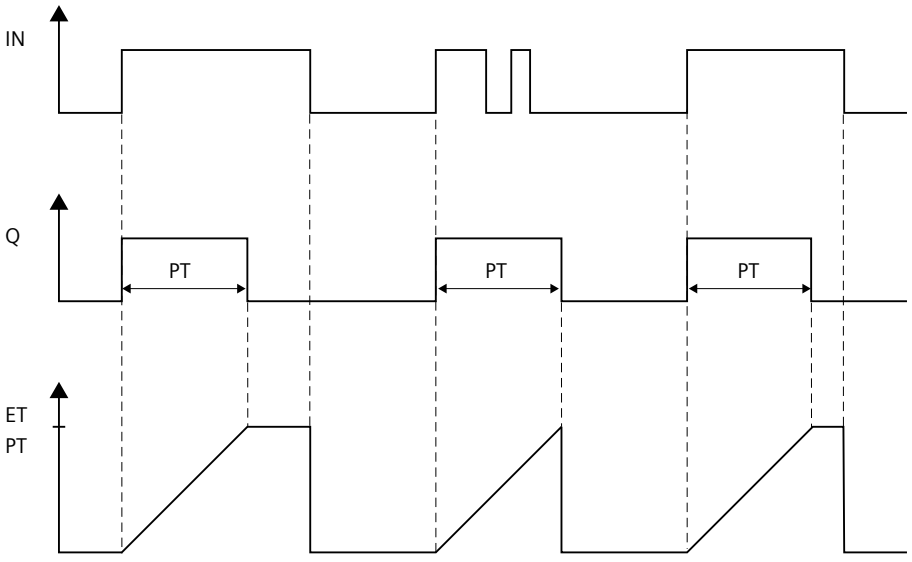
o This version is no longer supported.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Pulse diagram

The following figure shows the pulse diagram of the instruction "Generate pulse":

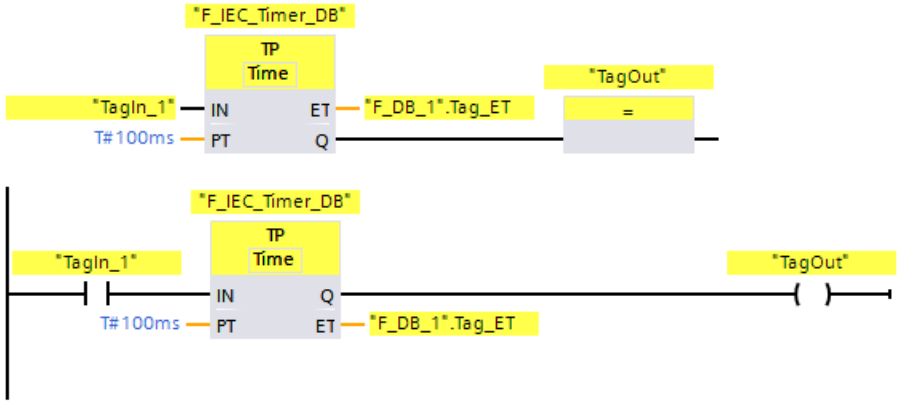


Timing imprecision resulting from the update time and the tolerance of the time base used in the instruction

Refer to section "Temporal blurring for instructions with time processing (Page 450)" for more on this.

Example

The following example shows how the instruction works:



If the signal state of operand "TagIn_1" changes from "0" to "1", the "Generate pulse" instruction is started and the period assigned at input PT (100 ms) runs, regardless of the further course of operand "TagIn_1".

Operand "TagOut" at output Q has signal state "1" as long as the period is running. Operand ""F_DB_1".Tag_ET" contains the current time value.

13.6.2 TON: Generate on-delay (STEP 7 Safety V21)

Description

You use the "Generate on-delay" instruction to delay the setting of output Q by the assigned period PT. The instruction is started if the result of logic operation (RLO) changes from "0" to "1" (positive edge) at input IN. The programmed period PT starts running when the instruction starts. When period PT has expired, output Q is set to signal state "1". Output Q remains set as long as the start input is set to "1". When the signal state at the start input changes from "1" to "0", output Q is reset. The time function is restarted when a new positive signal edge is detected at the start input.

You can query the current time value at the output ET. The time value begins at T#0s and ends when the value of period PT is reached. Output ET is reset, as soon as the signal state at input IN changes to "0".

Every call of the "Generate on-delay" instruction must be assigned a data area in which the instruction data are stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., F_IEC_Timer_DB_1) or a multi-instance (e.g., F_IEC_Timer_Instance_1) for the "Generate on-delay" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the *help on STEP 7*.

The operating system resets the instances of the "Generate on-delay" instruction on a startup of the F-system.

NOTE

The functionality of this instruction differs from the corresponding standard TON instruction in the following points:

- If the instruction is called while the time is running with PT = 0 ms, the output ET is reset.
- If the instruction is called with PT < 0 ms, the outputs Q and ET are reset.

To restart the on-delay, a new rising signal edge at input IN is required once PT is greater than 0 again.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN	Input	BOOL	Start input
PT	Input	TIME	Duration of on-delay; must be positive.
Q	Output	BOOL	Output that is set after expiration of time PT.
ET	Output	TIME	Current time value

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	x	—	—	o	These versions have identical functions to version 1.0.
1.2	x	o	—	o	
1.3	x	o	—	o	
1.4	x	x	x	x	

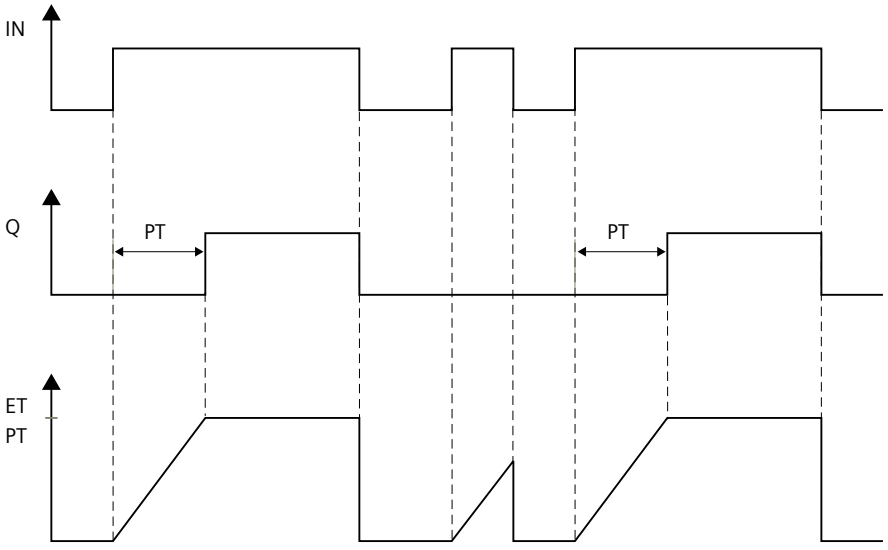
o This version is no longer supported.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Pulse diagram

The following figure shows the pulse diagram of the instruction "Generate on-delay":

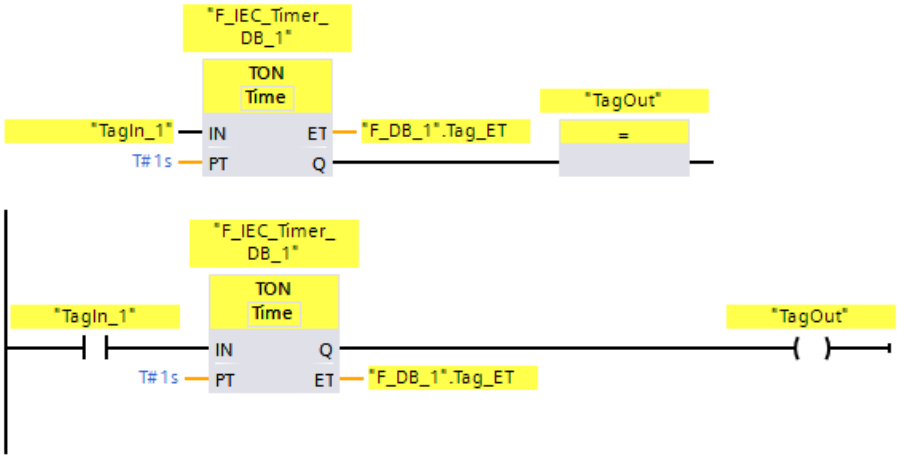


Timing imprecision resulting from the update time and the tolerance of the time base used in the instruction

Refer to section "Temporal blurring for instructions with time processing (Page 450)" for more on this.

Example

The following example shows how the instruction works:



When the signal state of operand "TagIn_1" changes from "0" to "1", the "Generate on-delay" instruction is started and the period assigned at input PT (1 s) runs.

Operand "TagOut" at output Q feeds signal state "1" when the period has elapsed and remains set as long as operand "TagIn_1" still feeds signal state "1". Operand ""F_DB_1".Tag_ET" contains the current time value.

13.6.3 TOF: Generate off-delay (STEP 7 Safety V21)

Description

You use the "Generate off-delay" instruction to delay the resetting of output Q by the assigned period PT. Output Q is set if the result of logic operation (RLO) changes from "0" to "1" (positive edge) at input IN. When the signal state at input IN changes back to "0", the programmed period PT starts. Output Q remains set as long as period PT runs. After period PT expires, output Q is reset. If the signal state at input IN changes to "1" before period PT has expired, then the time is reset. The signal state at output Q remains at "1".

You can query the current time value at the output ET. The time value begins at T#0s and ends when the value of period PT is reached. After time PT has elapsed, output ET remains at its current value until input IN changes back to "1". If input IN changes to "1" before time PT has expired, the output ET is reset to value T#0.

Every call of the "Generate off-delay" instruction must be assigned a data area in which the instruction data are stored. The "Call options" dialog is automatically opened when the instruction is in inserted in the program for this reason. There you can create a data block (single instance) (e.g., F_IEC_Timer_DB_1) or a multi-instance (e.g., F_IEC_Timer_Instance_1) for the "Generate off-delay" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the *help on STEP 7*.

The operating system resets the instances of the "Generate off-delay" instruction on a startup of the F-system.

NOTE

The functionality of this instruction differs from the corresponding standard TOF instruction in the following points:

- If the instruction is called while the time is running with PT = 0 ms, the outputs Q and ET are reset.
- If the instruction is called with PT < 0 ms, the outputs Q and ET are reset.

To restart the off-delay, another falling signal edge at input IN is required once PT is greater than 0 again.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN	Input	BOOL	Start input
PT	Input	TIME	Duration of off delay; must be positive.
Q	Output	BOOL	Output that is reset after expiration of time PT.
ET	Output	TIME	Current time value

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	x	—	—	o	These versions have identical functions to version 1.0.
1.2	x	o	—	o	
1.3	x	o	—	o	
1.4	x	x	x	x	

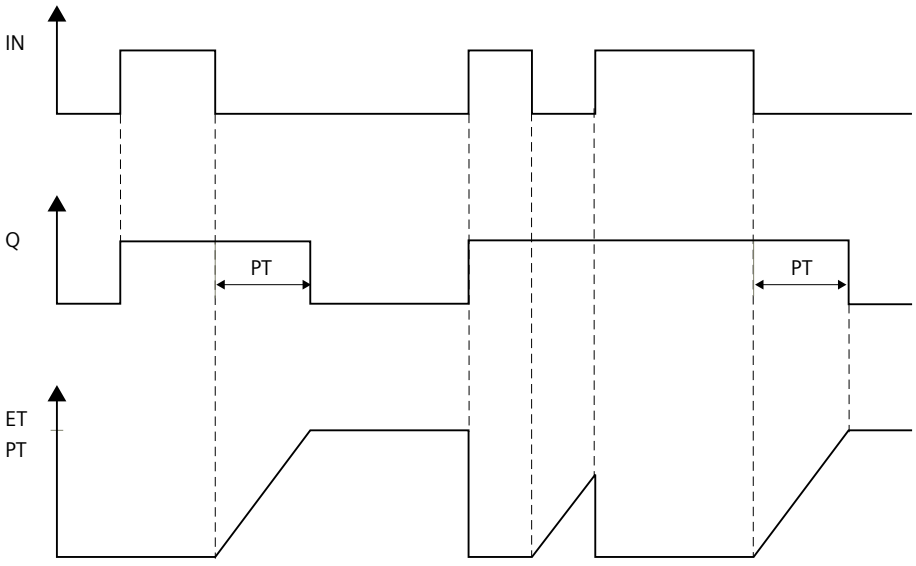
o This version is no longer supported.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Pulse diagram

The following figure shows the pulse diagram of the instruction "Generate off-delay":

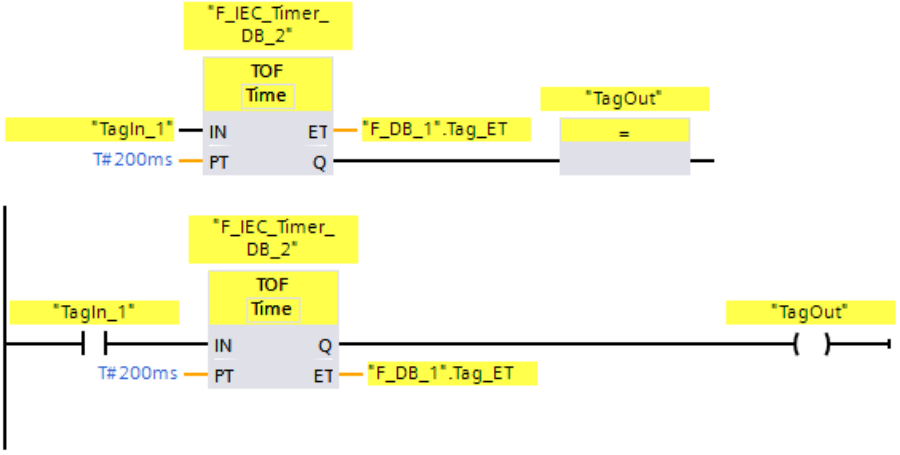


Timing imprecision resulting from the update time and the tolerance of the time base used in the instruction

Refer to section "Temporal blurring for instructions with time processing (Page 450)" for more on this.

Example

The following example shows how the instruction works:



If the signal state of operand "TagIn_1" changes from "0" to "1", the signal state of operand"TagOut" at output Q is set to "1".

If the signal state of operand "TagIn_1" changes back to "0", the period assigned at input PT (200 ms) runs.

The "TagOut" operand at output Q is set back to "0" when the period expires. Operand ""F_DB_1".Tag_ET" contains the current time value.

13.7 Counter operations

13.7.1 CTU: Count up (STEP 7 Safety V21)

Description

You can use the "Count up" instruction to increment the value at output CV. When the signal state at the CU input changes from "0" to "1" (positive signal edge), the instruction is executed and the current count at the CV output increases by one. The count value is increased on each detection of a positive signal edge until it reaches the high limit of the data type specified at the CV output. When the high limit is reached, the signal state at the CU input no longer affects the instruction.

The counter status can be queried at output Q. The signal state at output Q is determined by parameter PV. When the current count value is greater than or equal to the value of parameter PV, output Q is set to signal state "1". In all other cases, the signal state at output Q is "0".

The value at output CV is reset to zero when the signal state at input R changes to "1". As long as signal state "1" exists at input R, the signal state at input CU has no effect on the instruction.

Every call of the "Count up" instruction must be assigned a data area in which the instruction data are stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., F_IEC_Counter_DB_1) or a multi-instance (e.g., F_IEC_Counter_Instance_1) for the "Count up" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the *help on STEP 7*.

The operating system resets the instances of the "Count up" instruction on a startup of the F-system.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
CU	Input	BOOL	Counter input
R	Input	BOOL	Reset input
PV	Input	INT	Value for which output Q is set
Q	Output	BOOL	Counter status
CV	Output	INT	Current count value

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	x	—	—	o	These versions have identical functions to version 1.0.
1.2	x	o	—	o	
1.3	x	x	x	x	

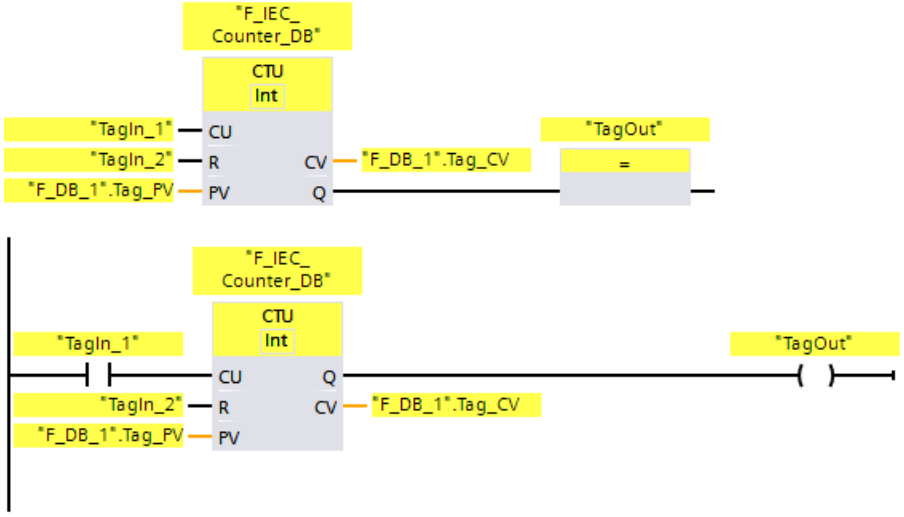
o This version is no longer supported.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Example

The following example shows how the instruction works:



When the signal state of the "CU" input changes from "0" to "1", the "Count up" instruction is executed and the current count of the "CV" output increases by one. The count value is increased on every additional positive signal edge until the high limit of the specified data type (32767) is reached.

The value at the PV parameter is applied as the limit for determining the "TagOut" operands at the Q output. Output "Q" returns the signal state "1" as long as the current count is greater than or equal to the value of operand "PV". In all other cases, the "Q" output has the signal state "0".

13.7.2 CTD: Count down (STEP 7 Safety V21)

Description

You can use the "Count down" instruction to decrement the value at output CV. When the signal state at input CD changes from "0" to "1" (positive signal edge), the instruction is executed and the current count value at output CV is decreased by one. The count value is decreased on each detection of a positive signal edge until it reaches the low limit of the specified data type. When the low limit is reached, the signal state at input CD no longer affects the instruction.

The counter status can be queried at output Q. When the current count value is less than or equal to zero, output Q is set to signal state "1". In all other cases, the signal state at output Q is "0".

The value at output CV is set to the value of parameter "PV" when the signal state at input LD changes to "1". As long as signal state "1" exists at input LD, the signal state at input CD has no effect on the instruction.

Every call of the "Count down" instruction must be assigned a data area in which the instruction data are stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., F_IEC_Counter_DB_1) or a multi-instance (e.g., F_IEC_Counter_Instance_1) for the "Count down" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the *help on STEP 7*.

The operating system resets the instances of the "Count down" instruction on a startup of the F-system.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
CD	Input	BOOL	Counter input
LD	Input	BOOL	Load input
PV	Input	INT	Value at the output CV when LD = 1 is set
Q	Output	BOOL	Counter status
CV	Output	INT	Current count value

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	x	—	—	o	These versions have identical functions to version 1.0.
1.2	x	o	—	o	
1.3	x	x	x	x	

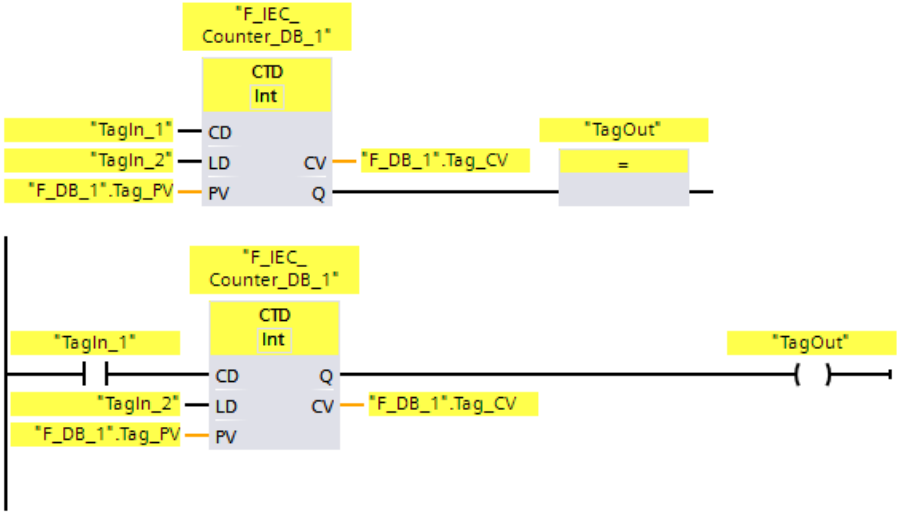
o This version is no longer supported.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Example

The following example shows how the instruction works:



When the signal state of the "CD" input changes from "0" to "1", the "Count down" instruction is executed and the current count at "CV" output decreases by one. The count value is decreased on each additional positive signal edge until the low limit of the specified data type (-32768) is reached.

Output "Q" returns the signal state "1" as long as the current count is less than or equal to zero. In all other cases, output "Q" has signal state "0".

13.7.3 CTUD: Count up and down (STEP 7 Safety V21)

Description

You can use the "Count up and down" instruction to increment and decrement the count value at output CV. If the signal state at the CU input changes from "0" to "1" (positive signal edge), the current count at the CV output increases by one. If the signal state at input CD changes from "0" to "1" (positive signal edge), the count value at output CV is decreased by one. If a positive signal edge is present at inputs CU and CD in one program cycle, the current count value at output CV remains unchanged.

The count value can be increased until it reaches the high limit of the data type specified at output CV. When the high limit is reached, the count value is no longer incremented on a positive signal edge. When the low limit of the specified data type is reached, the count value is no longer decreased.

When the signal state at input LD changes to "1", the count value at output CV is set to the value of parameter PV. As long as signal state "1" exists at input LD, the signal state at inputs CU and CD has no effect on the instruction.

The count value is set to zero, when the signal state at input R changes to "1". As long as signal state "1" exists at input R, the signal state at inputs CU, CD, and LD has no effect on the "Count up and down" instruction.

The status of the up counter can be queried at output QU. When the current count value is greater than or equal to the value of parameter PV, output QU delivers signal state "1". In all other cases, the signal state at output QU is "0".

The status of the down counter can be queried at output QD. When the current count value is lesser than or equal to zero, output QD delivers signal state "1". In all other cases, the signal state at output QD is "0".

Every call of the "Count up and down" instruction must be assigned a data area in which the instruction data are stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., F_IEC_Counter_DB_1) or a multi-instance (e.g., F_IEC_Counter_Instance_1) for the "Count up and down" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the *help on STEP 7*.

The operating system resets the instances of the "Count up and down" instruction when the F-system is started up.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
CU	Input	BOOL	Count up input
CD	Input	BOOL	Count down input
R	Input	BOOL	Reset input
LD	Input	BOOL	Load input
PV	Input	INT	Value at which the output QU is set / to which the output CV is set when LD = 1
QU	Output	BOOL	Status of up counter
QD	Output	BOOL	Status of down counter
CV	Output	INT	Current count value

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	x	—	—	o	These versions have identical functions to version 1.0.
1.2	x	o	—	o	
1.3	x	x	x	x	

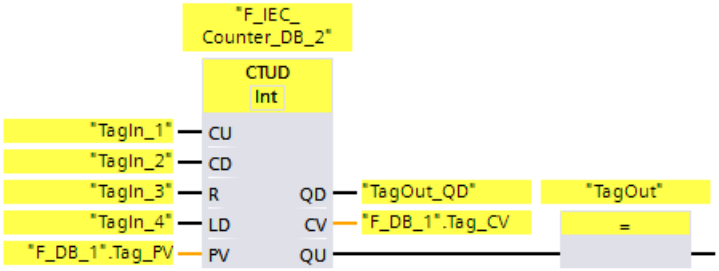
o This version is no longer supported.

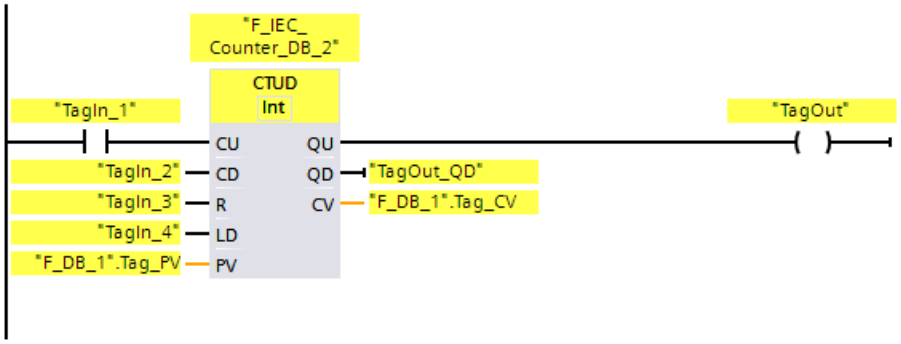
When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Example

The following example shows how the instruction works:





When the signal state at the "CU" input or at the "CD" input changes from "0" to "1" (positive signal edge), the "Count up and down" instruction is executed. When a positive signal edge is present at the "CU" input, the current count of the "CV" output increases by one. When a positive signal edge is present at the "CD" input, the current count at the "CV" output decreases by one. The count value is increased on each positive signal edge at the CU input until it reaches the high limit of 32767. The count value is decreased on each positive signal edge at the CD input until it reaches the low limit of -32768.

Output "QU" returns the signal state "1" as long as the current count is greater than or equal to the value at the "PV" input. In all other cases, output "QU" has signal state "0".

Output "QD" returns the signal state "1" as long as the current count is less than or equal to zero. In all other cases, output "QD" has signal state "0".

13.8 Comparator operations

13.8.1 CMP ==: Equal (STEP 7 Safety V21)

Description

You can use the "Equal" instruction to determine if the first comparison value (IN1 or <Operand1>) is equal to the second comparison value (IN2 or <Operand2>).

If the condition of the comparison is satisfied, the instruction returns result of logic operation (RLO) "1". If the condition of the comparison is not satisfied, the instruction returns RLO "0".

For LAD:

The RLO of the instruction is linked to the RLO of the entire current path as follows:

- By AND, when the comparison instruction is connected in series.
- By OR, when the comparison instruction is connected in parallel.

Enter the first comparison value (<Operand1>) in the operand placeholder above the instruction. Enter the second comparison value (<Operand2>) in the operand placeholder below the instruction.

Parameters

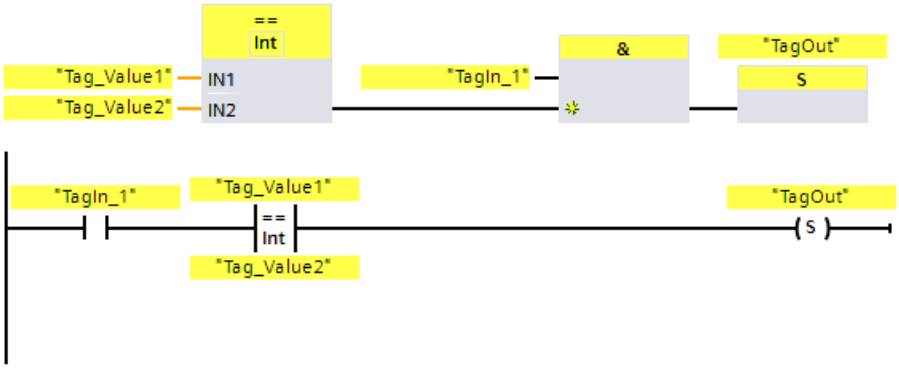
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
FBD: IN1 LAD: <Operand1>	Input	INT, DINT, TIME, WORD, (S7-300, S7-400) DWORD	First value to compare
FBD: IN2 LAD: <Operand2>	Input	INT, DINT, TIME, WORD, (S7-300, S7-400) DWORD	Second value to compare

You can select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example

The following example shows how the instruction works:



Output "TagOut" is set when the following conditions are fulfilled:

- "Tag_In1" has signal state "1".
- The condition of the comparison instruction is fulfilled ("Tag_Value1" = "Tag_Value2").

13.8.2 CMP <>: Not equal (STEP 7 Safety V21)

Description

You can use the "Not equal" instruction to determine if the first comparison value (IN1 or <Operand1>) is not equal to the second comparison value (IN2 or <Operand2>).

If the condition of the comparison is satisfied, the instruction returns result of logic operation (RLO) "1". If the condition of the comparison is not satisfied, the instruction returns RLO "0".

For LAD:

The RLO of the instruction is linked to the RLO of the entire current path as follows:

- By AND, when the comparison instruction is connected in series.
- By OR, when the comparison instruction is connected in parallel.

Enter the first comparison value (<Operand1>) in the operand placeholder above the instruction. Enter the second comparison value (<Operand2>) in the operand placeholder below the instruction.

Parameters

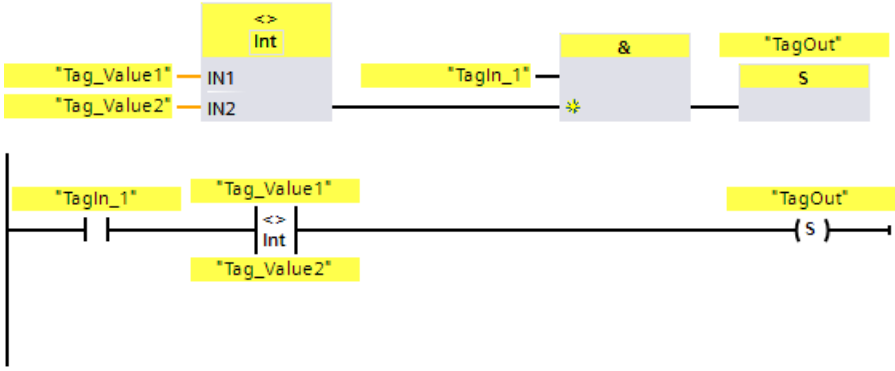
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
FBD: IN1 LAD: <Operand1>	Input	INT, DINT, TIME, WORD, (S7-300, S7-400) DWORD	First value to compare
FBD: IN2 LAD: <Operand2>	Input	INT, DINT, TIME, WORD, (S7-300, S7-400) DWORD	Second value to compare

You can select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example

The following example shows how the instruction works:



Output "TagOut" is set when the following conditions are fulfilled:

- "Tag_In1" has signal state "1".
- The condition of the comparison instruction is fulfilled ("Tag_Value1" <> "Tag_Value2").

13.8.3 CMP >=: Greater or equal (STEP 7 Safety V21)

Description

You can use the "Greater or equal" instruction to determine if the first comparison value (IN1 or <Operand1>) is greater than or equal to the second comparison value (IN2 or <Operand2>). Both comparison values must be of the same data type.

If the condition of the comparison is satisfied, the instruction returns result of logic operation (RLO) "1". If the condition of the comparison is not satisfied, the instruction returns RLO "0".

For LAD:

The RLO of the instruction is linked to the RLO of the entire current path as follows:

- By AND, when the comparison instruction is connected in series.
- By OR, when the comparison instruction is connected in parallel.

Enter the first comparison value (<Operand1>) in the operand placeholder above the instruction. Enter the second comparison value (<Operand2>) in the operand placeholder below the instruction.

Parameters

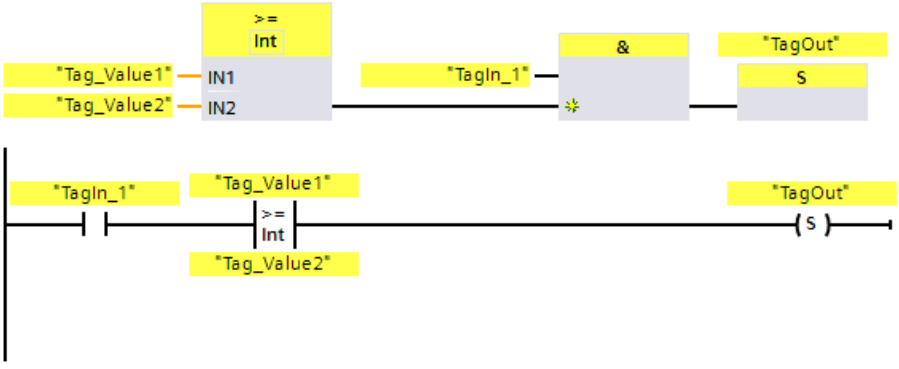
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
FBD: IN1 LAD: <Operand1>	Input	INT, DINT, TIME	First value to compare
FBD: IN2 LAD: <Operand2>	Input	INT, DINT, TIME	Second value to compare

You can select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example

The following example shows how the instruction works:



Output "TagOut" is set when the following conditions are fulfilled:

- "Tag_In1" has signal state "1".
- The condition of the comparison instruction is fulfilled ("Tag_Value1" >= "Tag_Value2").

13.8.4 CMP <=: Less or equal (STEP 7 Safety V21)

Description

You can use the "Less or equal" instruction to determine if the first comparison value (IN1 or <Operand1>) is less than or equal to the second comparison value (IN2 or <Operand2>). Both comparison values must be of the same data type.

If the condition of the comparison is satisfied, the instruction returns result of logic operation (RLO) "1". If the condition of the comparison is not satisfied, the instruction returns RLO "0".

For LAD:

The RLO of the instruction is linked to the RLO of the entire current path as follows:

- By AND, when the comparison instruction is connected in series.
- By OR, when the comparison instruction is connected in parallel.

Enter the first comparison value (<Operand1>) in the operand placeholder above the instruction. Enter the second comparison value (<Operand2>) in the operand placeholder below the instruction.

Parameters

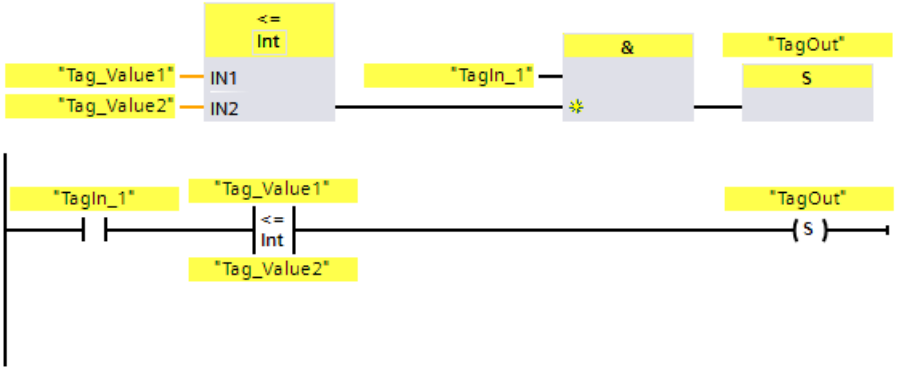
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
FBD: IN1 LAD: <Operand1>	Input	INT, DINT, TIME	First value to compare
FBD: IN2 LAD: <Operand2>	Input	INT, DINT, TIME	Second value to compare

You can select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example

The following example shows how the instruction works:



Output "TagOut" is set when the following conditions are fulfilled:

- "Tag_In1" has signal state "1".
- The condition of the comparison instruction is fulfilled ("Tag_Value1" <= "Tag_Value2").

13.8.5 CMP >: Greater than (STEP 7 Safety V21)

Description

You can use the "Greater than" instruction to determine if the first comparison value (IN1 or <Operand1>) is greater than the second comparison value (IN2 or <Operand2>). Both comparison values must be of the same data type.

If the condition of the comparison is satisfied, the instruction returns result of logic operation (RLO) "1". If the condition of the comparison is not satisfied, the instruction returns RLO "0".

For LAD:

The RLO of the instruction is linked to the RLO of the entire current path as follows:

- By AND, when the comparison instruction is connected in series.
- By OR, when the comparison instruction is connected in parallel.

Enter the first comparison value (<Operand1>) in the operand placeholder above the instruction. Enter the second comparison value (<Operand2>) in the operand placeholder below the instruction.

Parameters

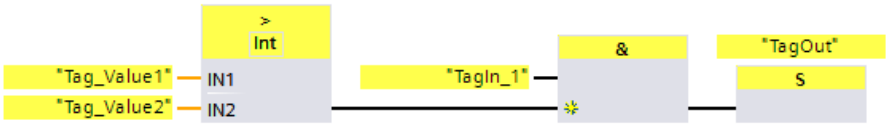
The following table shows the parameters of the instruction:

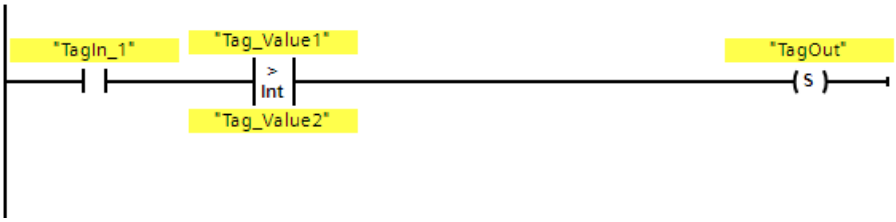
Parameter	Declaration	Data type	Description
FBD: IN1 LAD: <Operand1>	Input	INT, DINT, TIME	First value to compare
FBD: IN2 LAD: <Operand2>	Input	INT, DINT, TIME	Second value to compare

You can select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example

The following example shows how the instruction works:





Output "TagOut" is set when the following conditions are fulfilled:

- "Tag_In1" has signal state "1".
- The condition of the comparison instruction is fulfilled ("Tag_Value1" > "Tag_Value2").

13.8.6 CMP <: Less than (STEP 7 Safety V21)

Description

You can use the "Less than" instruction to determine if the first comparison value (IN1 or <Operand1>) is less than the second comparison value (IN2 or <Operand2>). Both comparison values must be of the same data type.

If the condition of the comparison is satisfied, the instruction returns result of logic operation (RLO) "1". If the condition of the comparison is not satisfied, the instruction returns RLO "0".

For LAD:

The RLO of the instruction is linked to the RLO of the entire current path as follows:

- By AND, when the comparison instruction is connected in series.
- By OR, when the comparison instruction is connected in parallel.

Enter the first comparison value (<Operand1>) in the operand placeholder above the instruction. Enter the second comparison value (<Operand2>) in the operand placeholder below the instruction.

Parameters

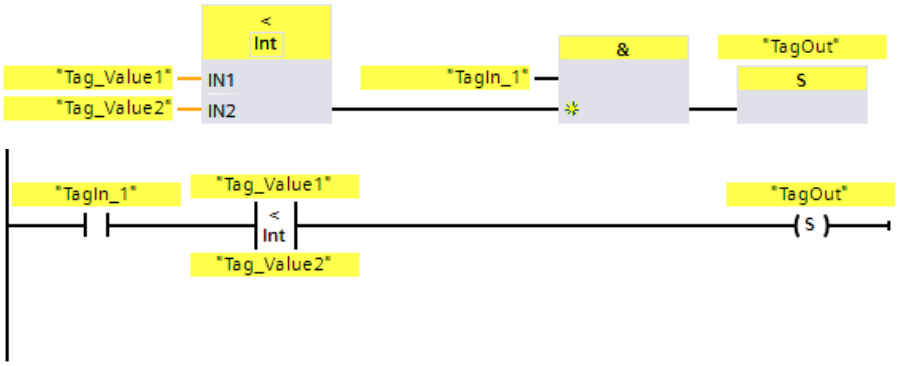
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
FBD: IN1 LAD: <Operand1>	Input	INT, DINT, TIME	First value to compare
FBD: IN2 LAD: <Operand2>	Input	INT, DINT, TIME	Second value to compare

You can select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example

The following example shows how the instruction works:



Output "TagOut" is set when the following conditions are fulfilled:

- "Tag_In1" has signal state "1".
- The condition of the comparison instruction is fulfilled ("Tag_Value1" < "Tag_Value2").

13.9 Math functions

13.9.1 ADD: Add (STEP 7 Safety V21)

Description

You can use the "Add" instruction to add the value at input IN1 and the value at input IN2 and query the sum at the OUT output (OUT = IN1 + IN2).

Enable input "EN" or (S7-300, S7-400) enable output "ENO" cannot be connected. The instruction is therefore always executed regardless of the signal state at enable input "EN".

NOTE

When the result of the instruction is located outside the permitted range for this data type, the F-CPU may switch to STOP. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

You must therefore ensure that the permitted range for the data type is observed when creating the program!

(S7-1200, S7-1500) You can avoid a STOP of the F-CPU by connecting the ENO enable output, thereby programming overflow detection.

Note the following:

- If the result of the instruction is located outside the permitted range for this data type, the enable output ENO returns the signal state "0".
- The result of the instruction reacts like the analogous instruction in a standard block.
- The execution time of the instruction is extended (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).
- Work memory requirement of safety program is increased.

(S7-300, S7-400) You can avoid a STOP of the F-CPU by inserting a "Get status bit OV" instruction in the next network, thereby programming overflow detection.

Note the following:

- The result of the instruction reacts like the analogous instruction in a standard block.
- The network with the "Get status bit OV" instruction must not contain any jump labels.
- The execution time of the instruction is extended (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).
- A warning is issued if you do not insert a "Get status bit OV" instruction.
- Work memory requirement of safety program is increased.

Parameters

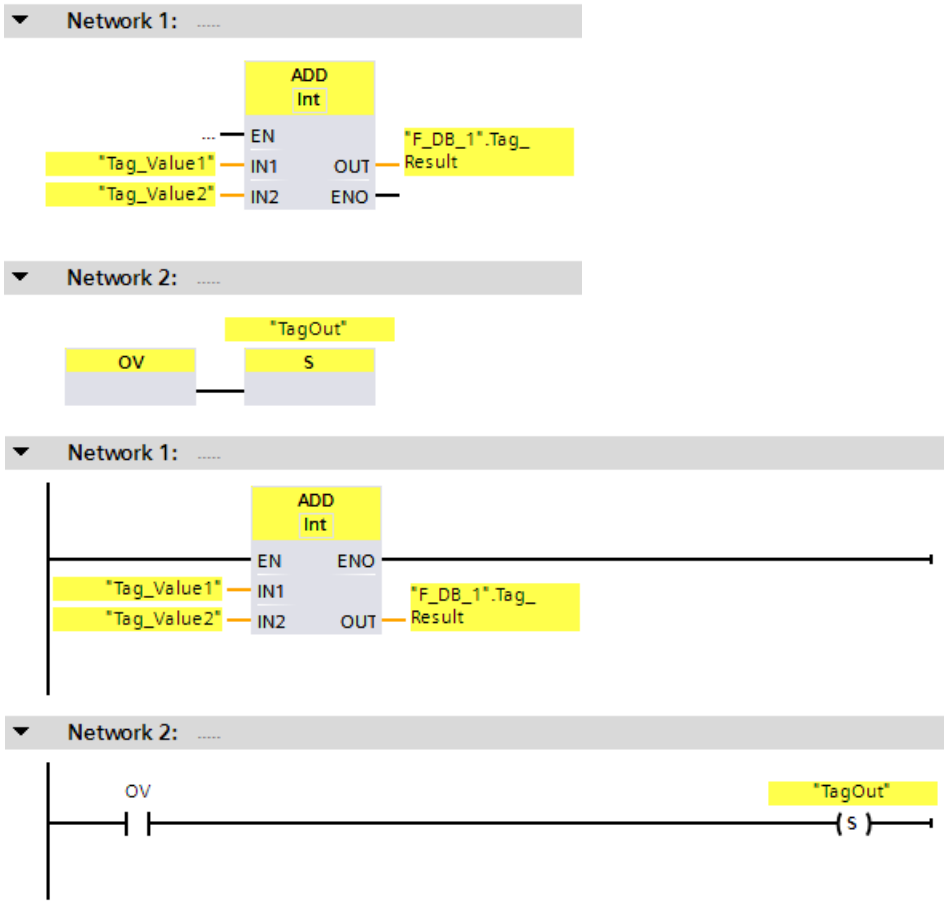
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
ENO	Output	BOOL	(S7-1200, S7-1500) Enable output
IN1	Input	INT, DINT	First addend
IN2	Input	INT, DINT	Second addend
OUT	Output	INT, DINT	Total

You select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example for S7-300/400 F-CPUs

The following example shows how the instruction works:



The "Add" instruction is always executed regardless of the signal state at enable input EN.

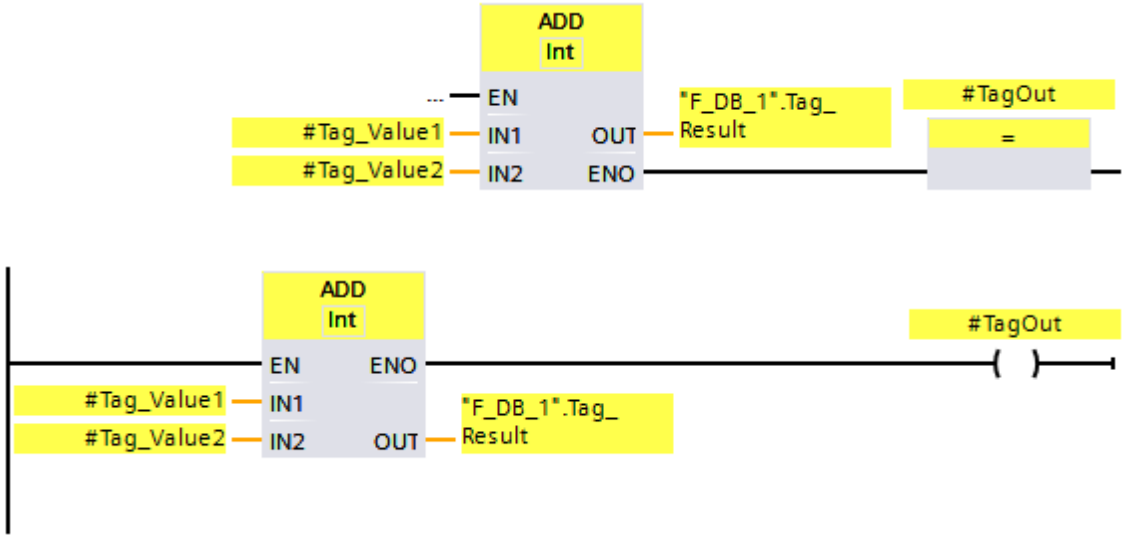
The value of the "Tag_Value1" operand is added to value of the "Tag_Value2" operand. The result of the addition is stored in the ""F_DB_1".Tag_Result" operand.

If needed, you can also store the signal status of the ENO enable output in an (F-)DB, and centrally evaluate whether overflows have occurred for all or one group of instructions with overflow detection.

When an overflow occurs during execution of the "Add" instruction, the status bit OV is set to "1". In network 2, following the query of the status bit OV, the "Set output" (S) instruction is executed and the "TagOut" operand is set.

Example for S7-1200/1500 F-CPUs

The following example shows how the instruction works:



The "Add" instruction is always executed regardless of the signal state at enable input EN.

The value of the "#Tag_Value1" operand is added to value of the "#Tag_Value2" operand. The result of the addition is stored in the ""F_DB_1".Tag_Result" operand.

When no overflow occurs during execution of the "Add" instruction, the ENO enable output has the signal state "1" and the "#TagOut" operand is set.

If needed, you can also store the signal status of the ENO enable output in an (F-)DB, and centrally evaluate whether overflows have occurred for all or one group of instructions with overflow detection.

See also

---| |--- OV: Get status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 609)

---| / |--- OV: Get negated status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 610)

13.9.2 SUB: Subtract (STEP 7 Safety V21)

Description

You can use the "Subtract" instruction to subtract the value at input IN2 from the value at input IN1 and query the difference at the OUT output (OUT = IN1 – IN2).

Enable input "EN" or (S7-300, S7-400) enable output "ENO" cannot be connected. The instruction is therefore always executed regardless of the signal state at enable input "EN".

NOTE

When the result of the instruction is located outside the permitted range for this data type, the F-CPU may switch to STOP. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

You must therefore ensure that the permitted range for the data type is observed when creating the program!

(S7-1200, S7-1500) You can avoid a STOP of the F-CPU by connecting the ENO enable output, thereby programming overflow detection.

Note the following:

- If the result of the instruction is located outside the permitted range for this data type, the enable output ENO returns the signal state "0".
- The result of the instruction reacts like the analogous instruction in a standard block.
- The execution time of the instruction is extended (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).
- Work memory requirement of safety program is increased.

(S7-300, S7-400) You can avoid a STOP of the F-CPU by inserting a "Get status bit OV" instruction in the next network, thereby programming overflow detection.

Note the following:

- The result of the instruction reacts like the analogous instruction in a standard block.
- The network with the "Get status bit OV" instruction must not contain any jump labels.
- The execution time of the instruction is extended (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).
- A warning is issued if you do not insert a "Get status bit OV" instruction.
- Work memory requirement of safety program is increased.

Parameters

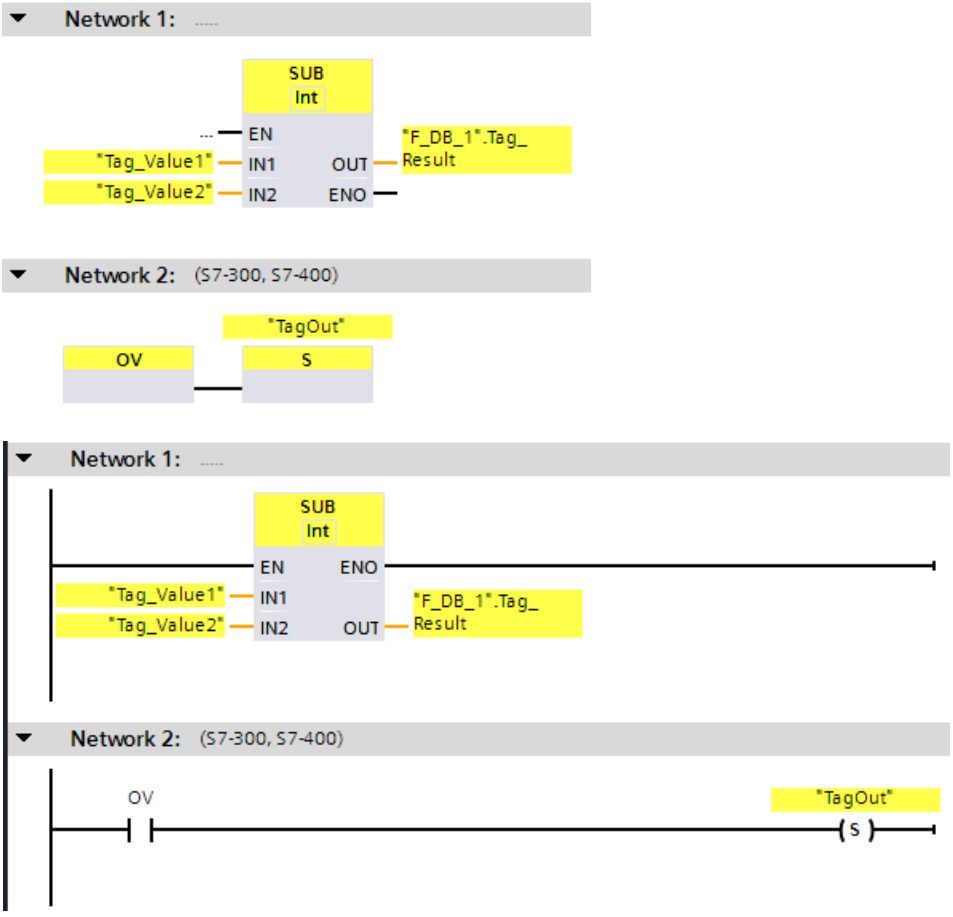
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
ENO	Output	BOOL	(S7-1200, S7-1500) Enable output
IN1	Input	INT, DINT	Minuend
IN2	Input	INT, DINT	Subtrahend
OUT	Output	INT, DINT	Difference

You select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example for S7-300/400 F-CPUs

The following example shows how the instruction works:



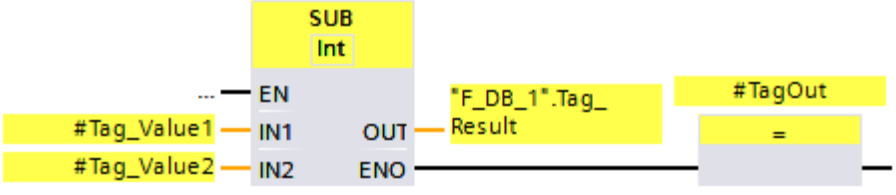
The "Subtract" instruction is always executed regardless of the signal state at enable input EN.

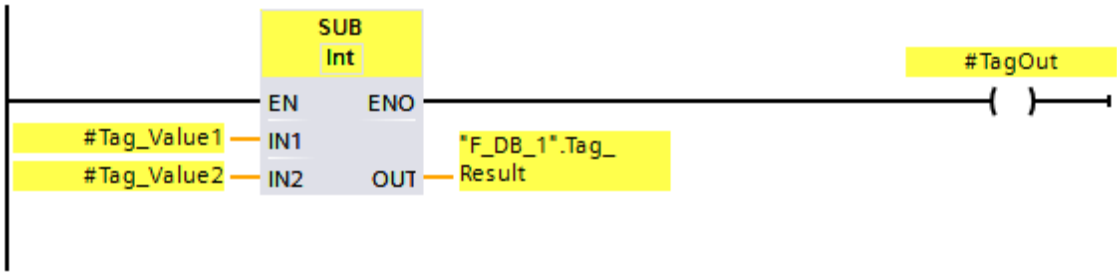
The value of the "Tag_Value2" operand is subtracted from the value of the "Tag_Value1" operand. The result of the addition is stored in the ""F_DB_1".Tag_Result" operand.

When an overflow occurs during execution of the "Subtract" instruction, the status bit OV is set to "1". In network 2, following the query of the status bit OV, the "Set output" (S) instruction is executed and the "TagOut" operand is set.

Example for S7-1200/1500 F-CPUs

The following example shows how the instruction works:





The "Subtract" instruction is always executed regardless of the signal state at enable input EN.

The value of the "#Tag_Value2" operand is subtracted from the value of the "#Tag_Value1" operand. The result of the addition is stored in the ""F_DB_1".Tag_Result" operand.

When no overflow occurs during execution of the "Subtract" instruction, the ENO enable output has the signal state "1" and the "#TagOut" operand is set.

See also

---| |--- OV: Get status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 609)

---| / |--- OV: Get negated status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 610)

13.9.3 MUL: Multiply (STEP 7 Safety V21)

Description

You can use the "Multiply" instruction to multiply the value at input IN1 by the value at input IN2 and query the product at output OUT (OUT = IN1 × IN2).

Enable input "EN" or (S7-300, S7-400) enable output "ENO" cannot be connected. The instruction is therefore always executed regardless of the signal state at enable input "EN".

NOTE

When the result of the instruction is located outside the permitted range for this data type, the F-CPU may switch to STOP. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

You must therefore ensure that the permitted range for the data type is observed when creating the program!

(S7-1200, S7-1500) You can avoid a STOP of the F-CPU by connecting the ENO enable output, thereby programming overflow detection.

Note the following:

- If the result of the instruction is located outside the permitted range for this data type, the enable output ENO returns the signal state "0".
- The result of the instruction reacts like the analogous instruction in a standard block.
- The execution time of the instruction is extended (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).
- Work memory requirement of safety program is increased.

(S7-300, S7-400) You can avoid a STOP of the F-CPU by inserting a "Get status bit OV" instruction in the next network, thereby programming overflow detection.

Note the following:

- The result of the instruction reacts like the analogous instruction in a standard block.
- The network with the "Get status bit OV" instruction must not contain any jump labels.
- The execution time of the instruction is extended (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).
- A warning is issued if you do not insert a "Get status bit OV" instruction.
- Work memory requirement of safety program is increased.

Parameters

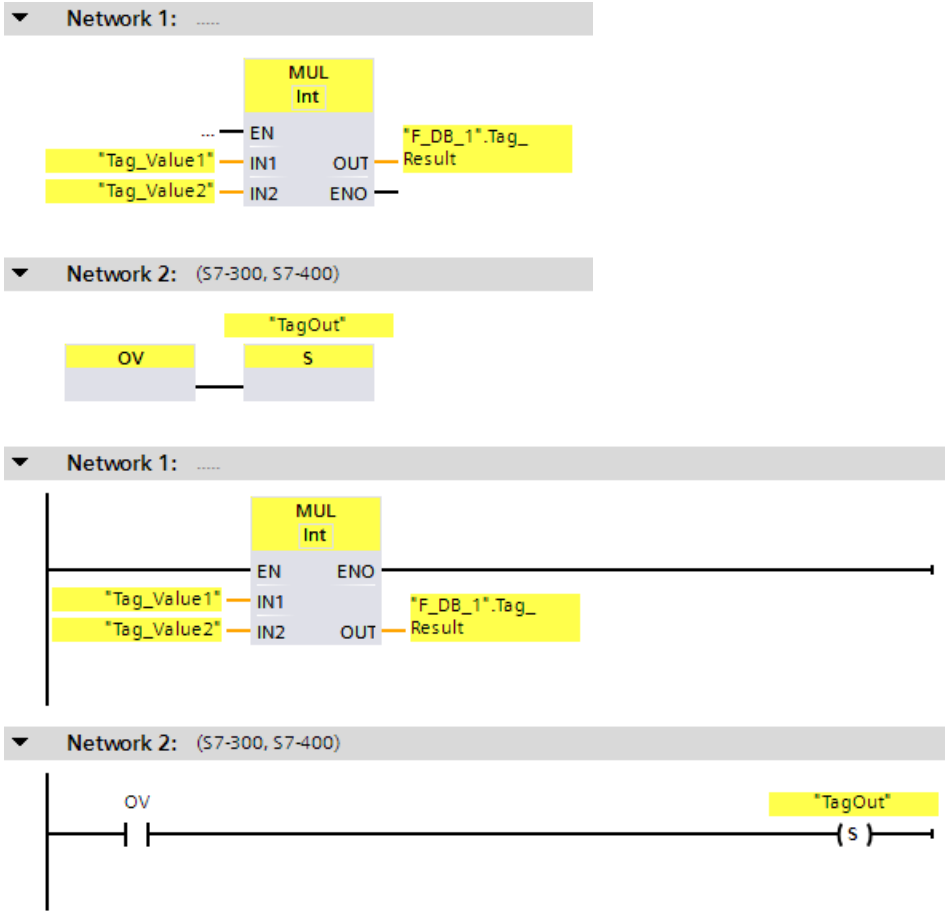
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
ENO	Output	BOOL	(S7-1200, S7-1500) Enable output
IN1	Input	INT, DINT	Multiplier
IN2	Input	INT, DINT	Multiplicand
OUT	Output	INT, DINT	Product

You select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example for S7-300/400 F-CPUs

The following example shows how the instruction works:



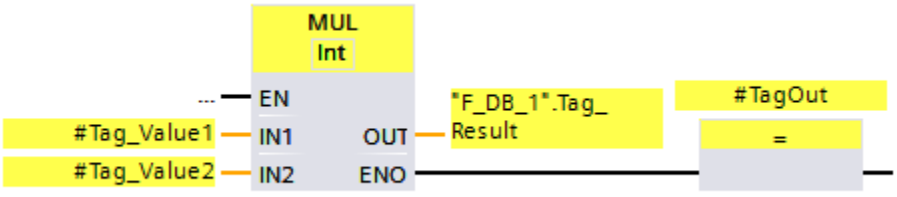
The "Multiply" instruction is always executed regardless of the signal state at enable input EN.

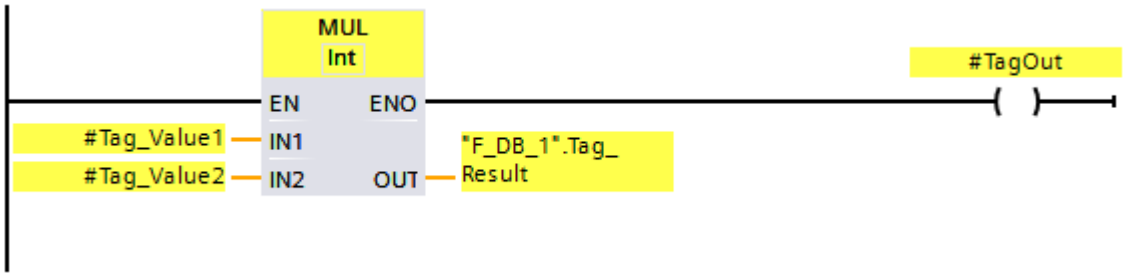
The value of the "Tag_Value1" operand is multiplied by the value of the "Tag_Value2" operand. The result of the multiplication is stored in the ""F_DB_1".Tag_Result" operand.

When an overflow occurs during execution of the "Multiply" instruction, the status bit OV is set to "1". In network 2, following the query of the status bit OV, the "Set output" (S) instruction is executed and the "TagOut" operand is set.

Example for S7-1200/1500 F-CPUs

The following example shows how the instruction works:





The "Multiply" instruction is always executed regardless of the signal state at enable input EN.

The value of the "#Tag_Value1" operand is multiplied by the value of the "#Tag_Value2" operand. The result of the multiplication is stored in the ""F_DB_1".Tag_Result" operand.

When no overflow occurs during execution of the "Multiply" instruction, the ENO enable output has the signal state "1" and the "#TagOut" operand is set.

See also

---| |--- OV: Get status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 609)

---| / |--- OV: Get negated status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 610)

13.9.4 DIV: Divide (STEP 7 Safety V21)

Description

You can use the "Divide" instruction to divide the value at input IN1 by the value at input IN2 and query the quotient at the OUT output (OUT = IN1 / IN2).

Enable input "EN" or (S7-300, S7-400) enable output "ENO" cannot be connected. The instruction is therefore always executed regardless of the signal state at enable input "EN".

NOTE

When the result of the instruction is located outside the permitted range for this data type, the F-CPU may switch to STOP. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

You must therefore ensure that the permitted range for the data type is observed when creating the program!

(S7-1200, S7-1500) You can avoid a STOP of the F-CPU by connecting the ENO enable output, thereby programming overflow detection.

Note the following:

- If the result of the instruction is located outside the permitted range for this data type, the enable output ENO returns the signal state "0".
- The result of the instruction reacts like the analogous instruction in a standard block.
- The execution time of the instruction is extended (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).
- Work memory requirement of safety program is increased.

(S7-300, S7-400) You can avoid a STOP of the F-CPU by inserting a "Get status bit OV" instruction in the next network, thereby programming overflow detection.

Note the following:

- The result of the instruction reacts like the analogous instruction in a standard block.
- The network with the "Get status bit OV" instruction must not contain any jump labels.
- The execution time of the instruction is extended (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).
- A warning is issued if you do not insert a "Get status bit OV" instruction.
- Work memory requirement of safety program is increased.

NOTE

S7-300/400, S7-1200/1500 (enable output ENO connected):

If the divisor (input IN2) of a DIV instruction = 0, the quotient of the division (result of division at output OUT) = 0. The result reacts like the corresponding instruction in a standard block. The F-CPU does *not* go to STOP mode.

S7-1200/1500 (enable output ENO not connected):

If the divisor (input IN2) of a DIV instruction = 0, the F-CPU goes to STOP. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU. We recommend that you rule out a divisor (input IN2) = 0 when creating the program.

Parameters

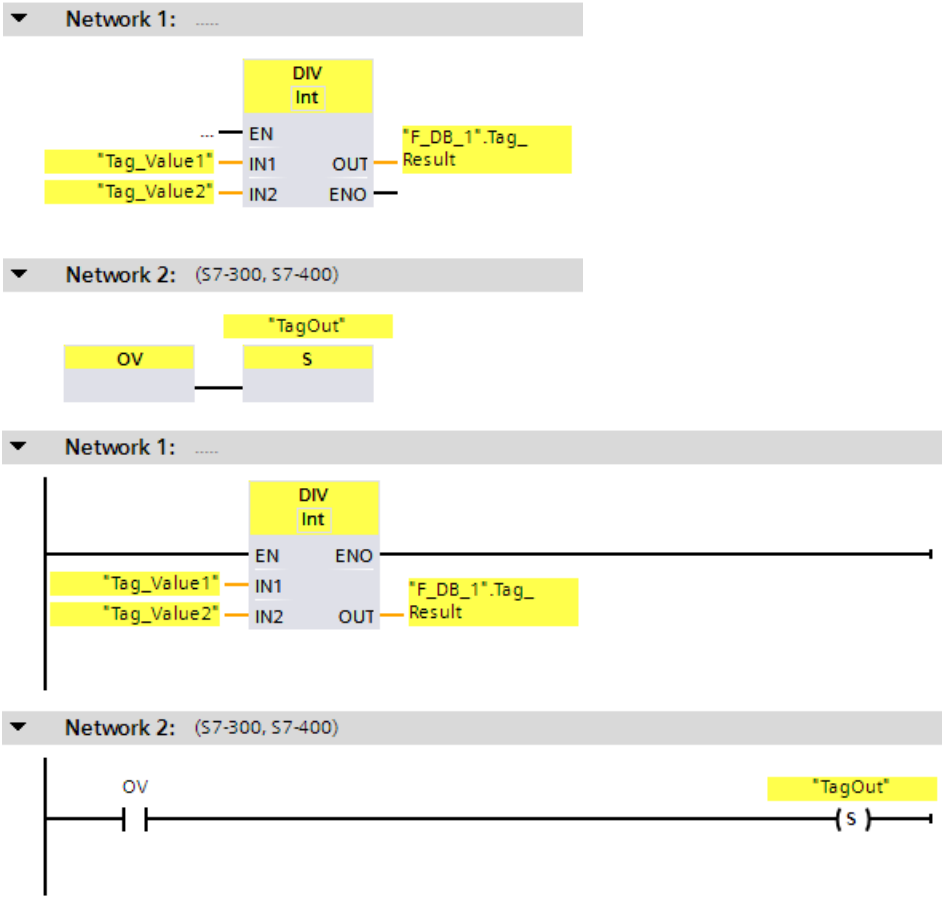
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
ENO	Output	BOOL	(S7-1200, S7-1500) Enable output
IN1	Input	INT, DINT	Dividend
IN2	Input	INT, DINT	Divisor
OUT	Output	INT, DINT	Quotient

You select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example for S7-300/400 F-CPUs

The following example shows how the instruction works:



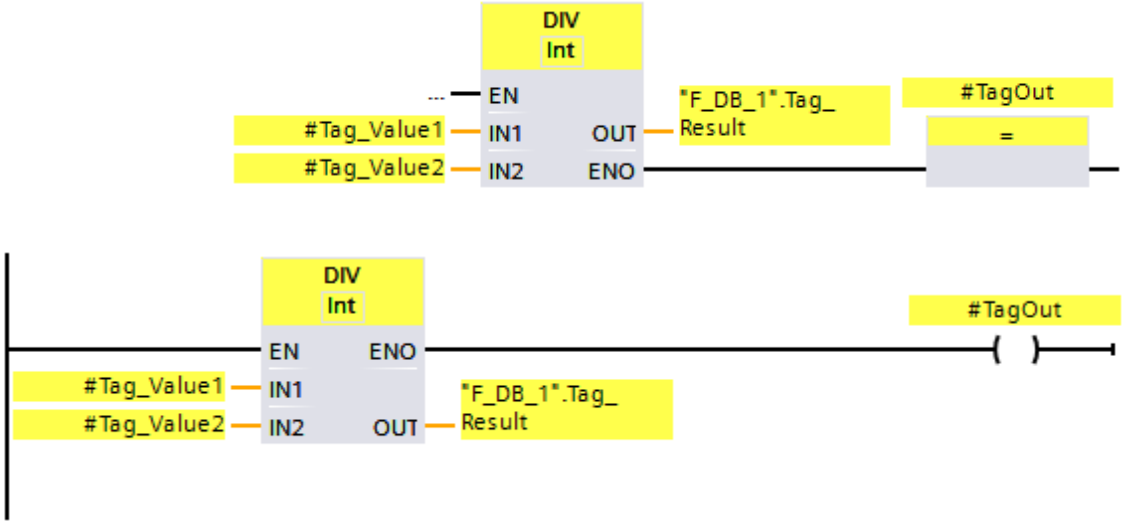
The "Divide" instruction is always execute regardless of the signal state at enable input EN.

The value of the "Tag_Value1" operand is divided by the value of the "Tag_Value2" operand. The result of the division is stored in the ""F_DB_1".Tag_Result" operand.

When an overflow occurs during execution of the "Divide" instruction, the status bit OV is set to "1". In network 2, following the query of the status bit OV, the "Set output" (S) instruction is executed and the "TagOut" operand is set.

Example for S7-1200/1500 F-CPUs

The following example shows how the instruction works:



The "Divide" instruction is always execute regardless of the signal state at enable input EN.

The value of the "#Tag_Value1" operand is divided by the value of the "#Tag_Value2" operand. The result of the division is stored in the ""F_DB_1".Tag_Result" operand.

When no overflow occurs during execution of the "Divide" instruction, the ENO enable output has the signal state "1" and the "#TagOut" operand is set.

See also

---| |--- OV: Get status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 609)

---| / |--- OV: Get negated status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 610)

13.9.5 NEG: Create twos complement (STEP 7 Safety V21)

Description

You can use the "Create twos complement" instruction to change the sign of the value at input IN input and query the result at output OUT. If there is a positive value at input IN, for example, the negative equivalent of this value is sent to output OUT.

Enable input "EN" or (S7-300, S7-400) enable output "ENO" cannot be connected. The instruction is therefore always executed regardless of the signal state at enable input "EN".

NOTE

When the result of the instruction is located outside the permitted range for this data type, the F-CPU may switch to STOP. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

You must therefore ensure that the permitted range for the data type is observed when creating the program!

(S7-1200, S7-1500) You can avoid a STOP of the F-CPU by connecting the ENO enable output, thereby programming overflow detection.

Note the following:

- If the result of the instruction is located outside the permitted range for this data type, the enable output ENO returns the signal state "0".
- The result of the instruction reacts like the analogous instruction in a standard block.
- The execution time of the instruction is extended (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).
- Work memory requirement of safety program is increased.

(S7-300, S7-400) You can avoid a STOP of the F-CPU by inserting a "Get status bit OV" instruction in the next network, thereby programming overflow detection.

Note the following:

- The result of the instruction reacts like the analogous instruction in a standard block.
- The network with the "Get status bit OV" instruction must not contain any jump labels.
- The execution time of the instruction is extended (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).
- A warning is issued if you do not insert a "Get status bit OV" instruction.
- Work memory requirement of safety program is increased.

Parameters

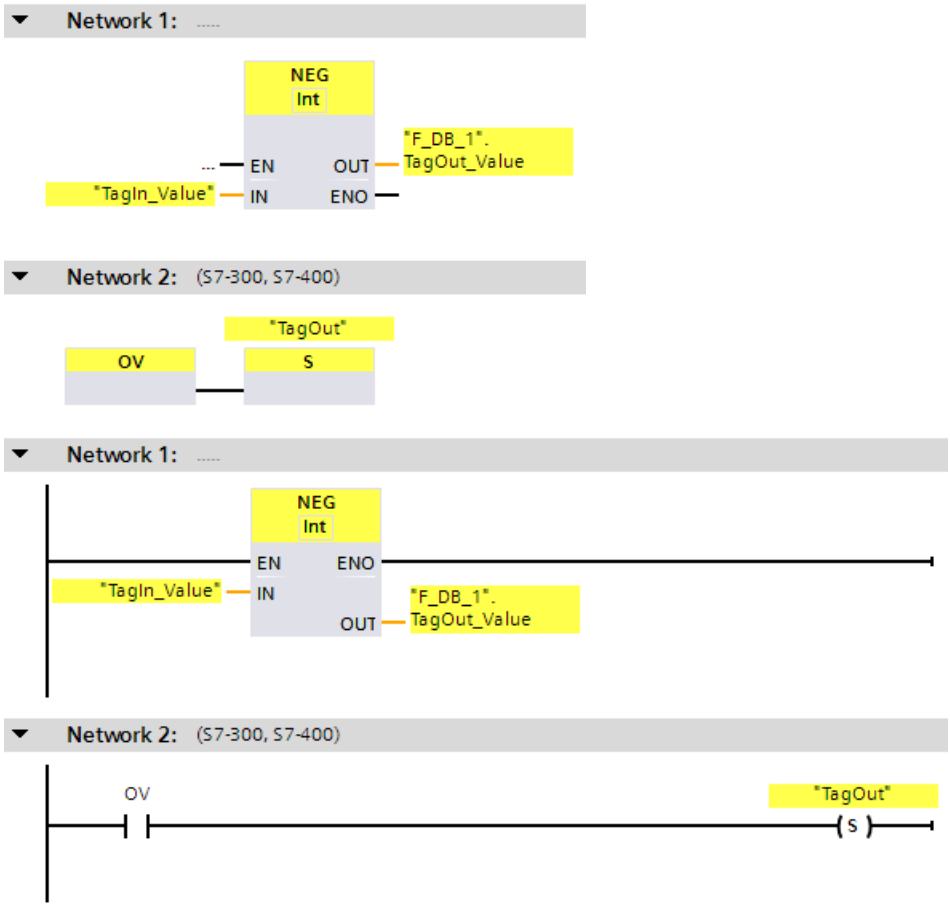
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
ENO	Output	BOOL	(S7-1200, S7-1500) Enable output
IN	Input	INT, DINT	Input value
OUT	Output	INT, DINT	Twos complement of the input value

You select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example for S7-300/400 F-CPUs

The following example shows how the instruction works:



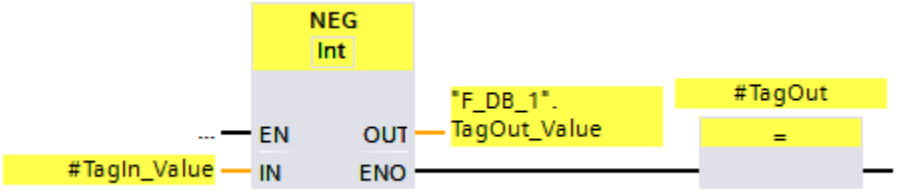
The "Create two's complement" instruction is always executed regardless of the signal state at enable input EN.

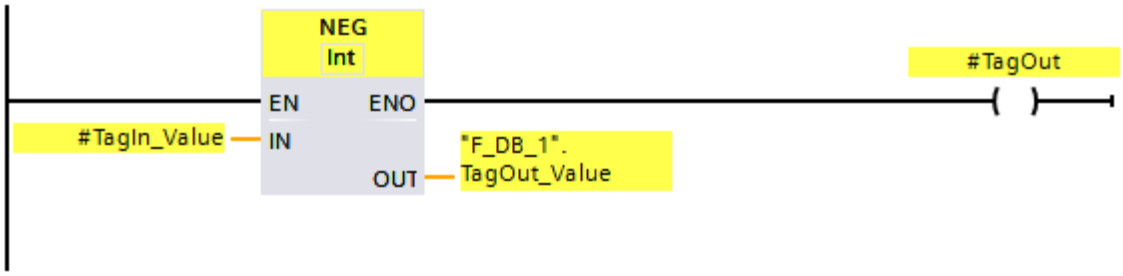
The sign of the "TagIn_Value" operand is changed and the result is stored in the ""F_DB_1".TagOut_Value" operand.

When an overflow occurs during execution of the "Create two's complement" instruction, the status bit OV is set to "1". In network 2, following the query of the status bit OV, the "Set output" (S) instruction is executed and the "TagOut" operand is set.

Example for S7-1200/1500 F-CPUs

The following example shows how the instruction works:





The "Create two's complement" instruction is always executed regardless of the signal state at enable input EN.

The sign of the "#TagIn_Value" operand is changed and the result is stored in the ""F_DB_1".TagOut_Value" operand.

When no overflow occurs during execution of the "Create two's complement" instruction, the ENO enable output has the signal state "1" and the "TagOut" operand is set.

See also

---| |--- OV: Get status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 609)

---| / |--- OV: Get negated status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 610)

13.9.6 ABS: Form absolute value (STEP 7 Safety V21) (S7-1200, S7-1500)

Description

You use the "Form absolute value" instruction to calculate the absolute amount of the value which is specified at the input "IN". The result of the instruction is output at the OUT output and can be queried there.

Enable input "EN" or (S7-300, S7-400) enable output "ENO" cannot be connected. The instruction is therefore always executed regardless of the signal state at enable input "EN".

NOTE

When the result of the instruction is located outside the permitted range for this data type, the F-CPU may switch to STOP. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

You must therefore ensure that the permitted range for the data type is observed when creating the program!

You can avoid a STOP of the F-CPU by connecting the ENO enable output, thereby programming overflow detection.

Note the following:

- If the result of the instruction is located outside the permitted range for this data type, the enable output ENO returns the signal state "0".
- The result of the instruction reacts like the analogous instruction in a standard block.
- The execution time of the instruction is extended (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).
- Work memory requirement of safety program is increased.

Parameters

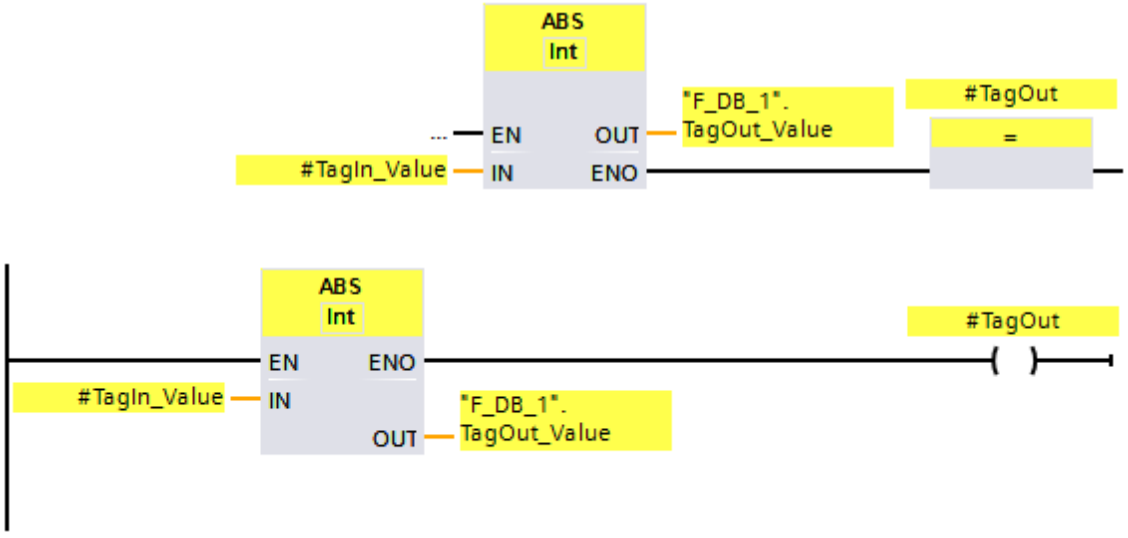
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
ENO	Output	BOOL	(S7-1200, S7-1500) Enable output
IN	Input	INT, DINT	Input value
OUT	Output	INT, DINT	Absolute value of the input value

You select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example

The following example shows how the instruction works:



The instruction "Form absolute value"is always executed regardless of the signal state at enable input "EN".

The absolute amount of the value at the "TagIn_Value" operand is calculated and the result is stored in the ""F_DB_1".TagOut_Value" operand.

When no overflow occurs during execution of the "Form absolute value" instruction, the enable output ENO has the signal state "1" and the "#TagOut" operand is set.

13.10 Move operations

13.10.1 MOVE: Move value (STEP 7 Safety V21)

Description

You can use the "Move value" instruction to transfer the content of the operand at input IN to the operand at output OUT1.

Only operands with identical operand width and identical data structure can be specified for input IN and output OUT1.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

(S7-1200, S7-1500) In the basic state, the instruction box contains an output (OUT1). The number of outputs can be expanded. The inserted outputs are numbered in ascending order on the box. During execution, the content of the operand at the IN input is transferred to all available outputs. The instruction box cannot be expanded if operands with F-compliant PLC data types (UDT) are transferred.

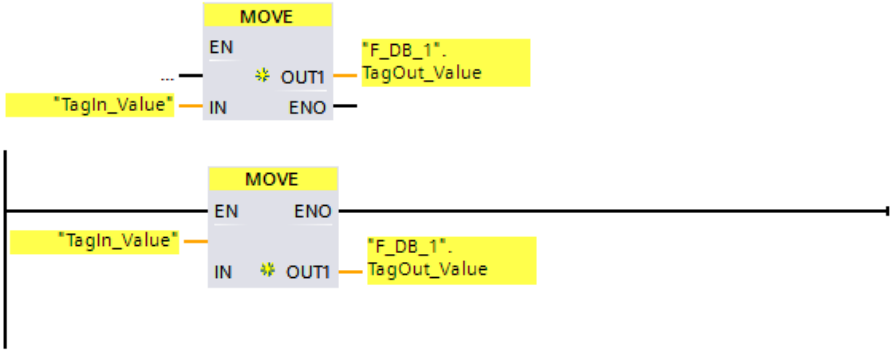
Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN	Input	INT, DINT, WORD, (S7-300, S7-400) DWORD, TIME, F-compliant PLC data type (UDT)	Source value
OUT1	Output	INT, DINT, WORD, (S7-300, S7-400) DWORD, TIME, F-compliant PLC data type (UDT)	Destination address

Example

The following example shows how the instruction works:



The instruction is always executed regardless of the signal state at enable input "EN". The instruction copies the content of operand "TagIn_Value" to operand ""F_DB_1".TagOut_Value".

13.10.2 RD_ARRAY_I: Read value from INT F-array (STEP 7 Safety V21) (S7-1200 G2, S7-1500)

Description

You use the "Read value from INT F-array" instruction to read an element from the array at the ARRAY input, which refers to the index at the INDEX input, and write its value at the OUT output. If an error occurs while accessing the array during runtime, this is displayed at the output ERROR.

The array must be created in an F-DB and can contain only one dimension. The elements of the ARRAY must be data type INT. The following applies for the array limits in contrast:

- The low limit must be 0.
- The high limit can be 10000 maximum.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
ARRAY	Input	VARIANT	Array from which is read
INDEX	Input	DINT	Element in the array which is read. The specification may be a constant or a tag.
OUT	Output	INT	Value which is read and output.
ERROR	Output	BOOL	Error information The parameter ERROR is set if an error occurs during the processing of the instruction.

ARRAY parameter

In addition to the direct connection with an array within a fail-safe global DB, this input can also be interconnected with an INOUT of data type ARRAY[*] of INT. This enables the decoupling of data and program logic in order, for example, to create a library function without any connection existing to a dedicated data block.

ERROR parameter

The table below shows the meaning of the value of the ERROR parameter:

Value	Description
FALSE	No error
TRUE	Value at the INDEX parameter is outside the limit value of the ARRAY.

Instruction versions

One version is available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	—	—	x	x[1]	

[1] supported as of firmware V2.0

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

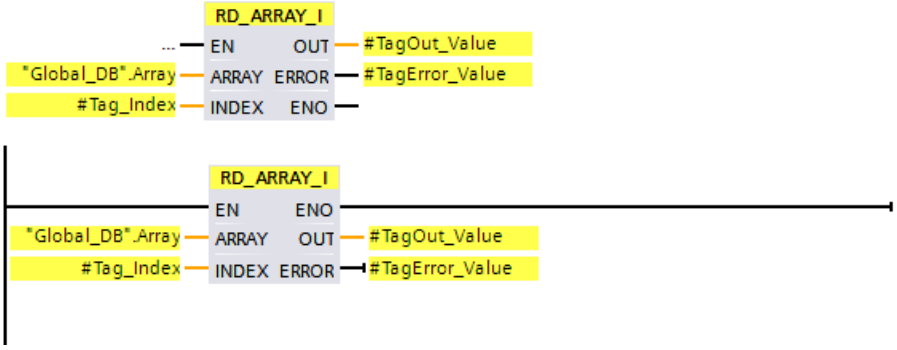
Reaction to errors

If the value at the INDEX input is outside the array limits, the output ERROR = 1 is set, and the array value of the element with index = 0 is output at OUT output, regardless of the value passed at the INDEX input.

Therefore, set the value of the element with index = 0 as a fail-safe substitute value.

Example

The following example shows how the instruction works:



The following table shows how the instruction works using specific operand values:

Parameter	Operand	Value
ARRAY	"Global_DB".Array	The "Global_DB".Array operand is an ARRAY of data type Array[0..10] of INT
INDEX	#Tag_Index	2
OUT	#TagOut_Value	Value of the element at the location array[2]
ERROR	#TagError_Value	False

The instruction "Read value from INT F-array" is always executed regardless of the signal state at enable input "EN".

The content of the 2nd element of the operand "Global_DB.Array" is output at the "#TagOut_Value" output.

13.10.3 RD_ARRAY_DI: Read value from DINT F-array (STEP 7 Safety V21) (S7-1200 G2, S7-1500)

Description

You use the "Read value from DINT F-array" instruction to read an element from the array at the ARRAY input, which refers to the index at the INDEX input, and write its value at the OUT output. If an error occurs while accessing the array during runtime, this is displayed at the output ERROR.

The array must be created in an F-DB and can contain only one dimension. The elements of the array must be of the DINT data type. The following applies for the array limits in contrast:

- The low limit must be 0.
- The high limit can be 10000 maximum.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
ARRAY	Input	VARIANT	Array from which is read
INDEX	Input	DINT	Element in the array which is read. The specification may be a constant or a tag.
OUT	Output	DINT	Value which is read and output.
ERROR	Output	BOOL	Error information The parameter ERROR is set if an error occurs during the processing of the instruction.

ARRAY parameter

In addition to the direct connection with an array within a fail-safe global DB, this input can also be interconnected with an INOUT of data type ARRAY[*] of DINT. This enables the decoupling of data and program logic in order, for example, to create a library function without any connection existing to a dedicated data block.

ERROR parameter

The table below shows the meaning of the value of the ERROR parameter:

Value	Description
FALSE	No error
TRUE	Value at the INDEX parameter is outside the limit value of the ARRAY.

Instruction versions

One version is available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	—	—	x	x[1]	

[1] supported as of firmware V2.0

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

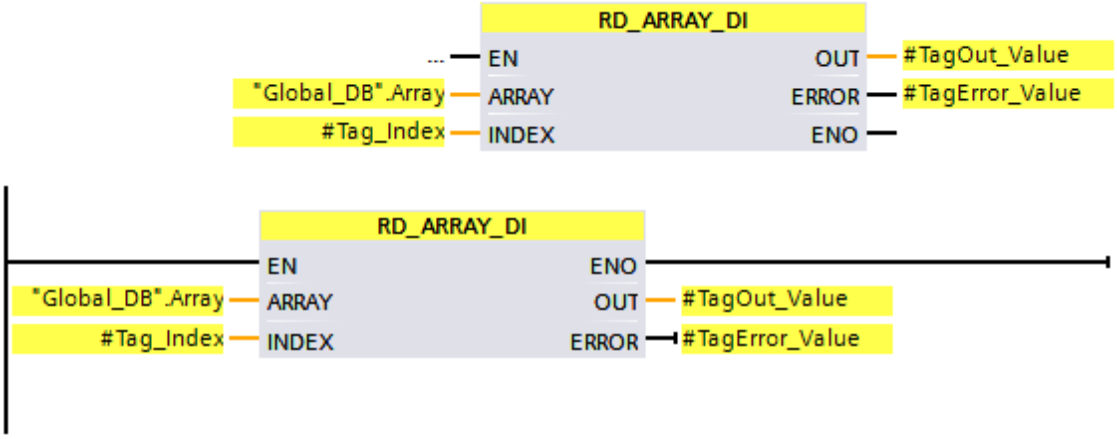
Reaction to errors

If the value at the INDEX input is outside the array limits, the output ERROR = 1 is set, and the array value of the element with index = 0 is output at OUT output, regardless of the value passed at the INDEX input.

Therefore, set the value of the element with index = 0 as a fail-safe substitute value.

Example

The following example shows how the instruction works:



The following table shows how the instruction works using specific operand values:

Parameter	Operand	Value
ARRAY	"Global_DB".Array	The operand "Global_DB".Array is an ARRAY of data type Array[0..10] of DINT
INDEX	#Tag_Index	2
OUT	#TagOut_Value	Value of the element at the location array[2]
ERROR	#TagError_Value	False

The instruction "Read value from DINT F-array" is always executed regardless of the signal state at enable input "EN".

The content of the 2nd element of the operand "Global_DB.Array" is output at the "#TagOut_Value" output.

13.10.4 WR_FDB: Write value indirectly to an F-DB (STEP 7 Safety Advanced V21) (S7-300, S7-400)

Description

This instruction writes the value specified in input IN to the tag addressed by INI_ADDR and OFFSET in an F-DB.

The address of the tags addressed using INI_ADDR and OFFSET must be within the address range defined by addresses INI_ADDR and END_ADDR.

If the F-CPU has gone to STOP mode with diagnostics event ID 75E2, check to determine if this condition is satisfied.

The start address of the area in an F-DB to which the value at input IN is to be written is transferred using input INI_ADDR. The associated offset in this area is transferred using input OFFSET.

The addresses transferred in input INI_ADDR or END_ADDR must point to a tag of the selected data type in an F-DB. Only tags of the selected data type are permitted between the INI_ADDR and END_ADDR addresses. The INI_ADDR address must be smaller than the END_ADDR address.

As shown in the following example, the INI_ADDR and END_ADDR addresses must be transferred fully-qualified as "DBx".DBWy or in the corresponding symbolic representation. Transfers in other forms are not permitted.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN	Input	INT, DINT	Value to be written to the F-DB
INI_ADDR	Input	POINTER	Start address of the area in an F-DB
END_ADDR	Input	POINTER	End address of the area in an F-DB
OFFSET	Input	INT	Offset

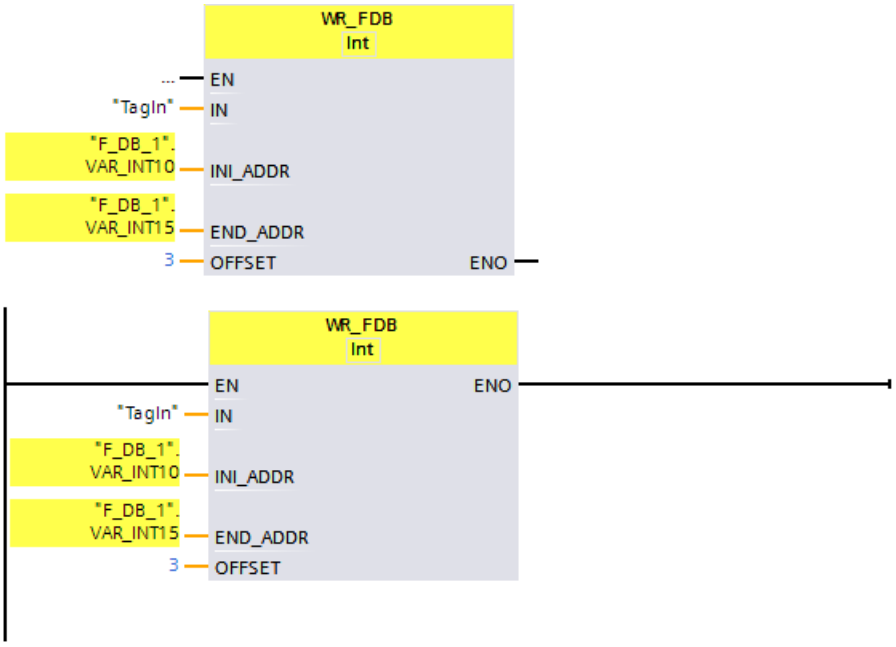
You can select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Examples of parameter assignment of INI_ADDR, END_ADDR, and OFFS

Name	Data type	Initial value	Comment
Static			
VAR_BOOL10	BOOL	false	
VAR_BOOL11	BOOL	false	
VAR_BOOL12	BOOL	false	
VAR_BOOL13	BOOL	false	
VAR_TIME10	TIME	T#0MS	
VAR_TIME11	TIME	T#0MS	
VAR_INT10	INT	0	<- INI_ADDR = "F-DB_1".VAR_INT10 Example 1
VAR_INT11	INT	0	
VAR_INT12	INT	0	
VAR_INT13	INT	0	<- OFFSET = 3
VAR_INT14	INT	0	
VAR_INT15	INT	0	<- END_ADDR = "F-DB_1".VAR_INT15
VAR_BOOL20	BOOL	false	
VAR_BOOL21	BOOL	false	
VAR_BOOL22	BOOL	false	
VAR_BOOL23	BOOL	false	
VAR_INT20	INT	0	<- INI_ADDR = "F-DB_1".VAR_INT20 Example 2
VAR_INT21	INT	0	
VAR_INT22	INT	0	
VAR_INT23	INT	0	<- END_ADDR = "F-DB_1".VAR_INT23
VAR_INT30	INT	0	<- INI_ADDR = "F-DB_1".VAR_INT30 Example 3
VAR_INT31	INT	0	<- OFFSET = 1
VAR_INT32	INT	0	
VAR_INT33	INT	0	
VAR_INT34	INT	0	<- END_ADDR = "F-DB".VAR_INT34
VAR_TIME20	TIME	T#0MS	
VAR_DINT10	DINT	0	<- INI_ADDR = "F-DB_1".VAR_DINT10 Example 4
VAR_DINT11	DINT	0	
VAR_DINT12	DINT	0	<- OFFSET = 2
VAR_DINT13	DINT	0	<- END_ADDR = "F-DB_1".VAR_DINT13

Example

The following example shows how the instruction works:



13.10.5 RD_FDB: Read value indirectly from an F-DB (STEP 7 Safety Advanced V21) (S7-300, S7-400)

Description

This instruction reads the tag addressed via INI_ADDR and OFFSET in an F-DB and provides it at output OUT.

The address of the tags addressed using INI_ADDR and OFFSET must be within the address range defined by addresses INI_ADDR and END_ADDR.

If the F-CPU has gone to STOP mode with diagnostics event ID 75E2, check to determine if this condition is satisfied.

The start address of the area in an F-DB from which the tag is to be read is transferred using input INI_ADDR. The associated offset in this area is transferred using input OFFSET.

The addresses transferred in input INI_ADDR or END_ADDR must point to a tag of the selected data type in an F-DB. Only tags of the selected data type are permitted between the INI_ADDR and END_ADDR addresses. The INI_ADDR address must be smaller than the END_ADDR address.

The INI_ADDR and END_ADDR addresses must be transferred fully-qualified as "DBx".DBWy or in the corresponding symbolic representation. Transfers in other forms are not permitted. Examples of parameter assignment of INI_ADDR, END_ADDR, and OFFSET are contained in WR_FDB: Write value indirectly to an F-DB (STEP 7 Safety Advanced V21) (S7-300, S7-400) (Page 575).

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Parameters

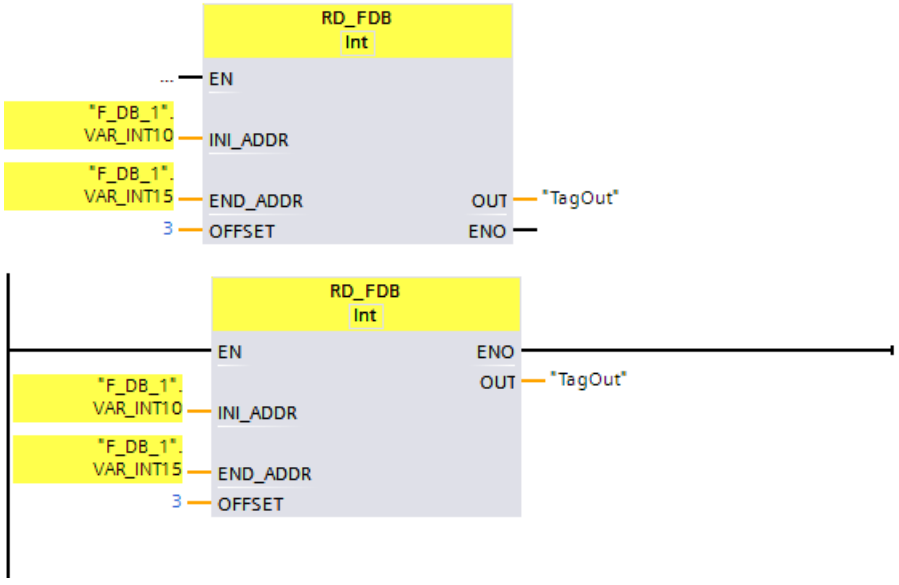
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
INI_ADDR	Input	POINTER	Start address of the area in an F-DB
END_ADDR	Input	POINTER	End address of the area in an F-DB
OFFSET	Input	INT	Offset
OUT	Output	INT, DINT	Value to be read from the F-DB

You can select the data type of the instruction in the "<???>" drop-down list in the instruction box.

Example

The following example shows how the instruction works:



13.11 Conversion operations

13.11.1 CONVERT: Convert value (STEP 7 Safety V21)

Description

The "Convert value" instruction reads the content of parameter IN and converts it according to the data types selected in the instruction box. The converted value is output at the OUT output .

Enable input "EN" cannot be connected. The instruction is therefore always executed (regardless of the signal state at the "EN" enable input). The connection of the "ENO" enable output is only possible and required when converting from the "DINT" to the "INT" data type.

NOTE

When converting from "DINT" to the "INT" data type, you need to connect the ENO enable output and thereby programming overflow detection.

Note the following:

- If the value at the input is outside the INT range, ENO returns 0.
- The result of the instruction reacts like the analogous instruction in a standard block.

Parameters

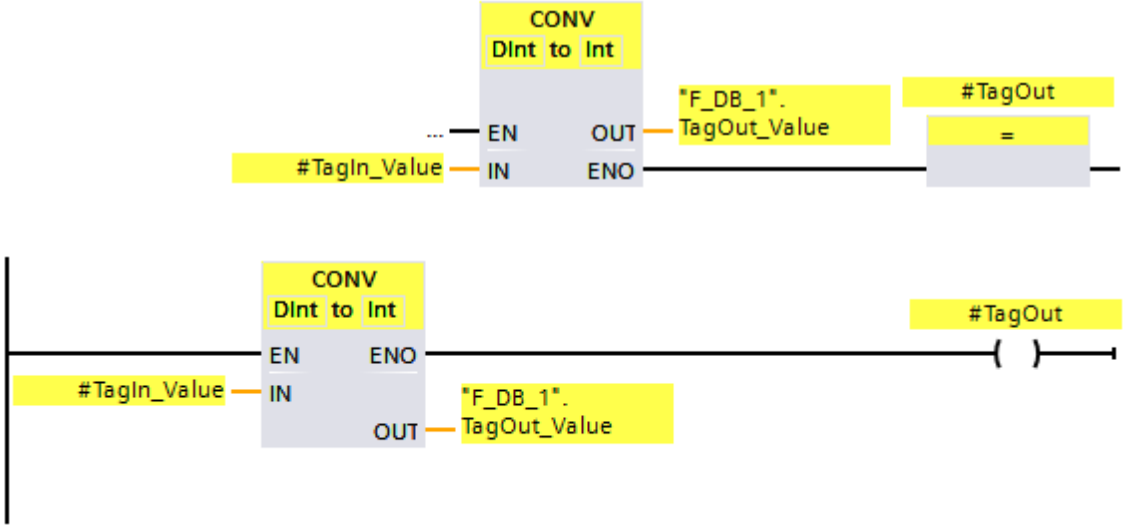
The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
ENO	Output	BOOL	Enable output
IN	Input	INT, DINT	Value to be converted.
OUT	Output	INT, DINT	Result of the conversion

You can select the data types of the instruction in the "<???>" drop-down lists of the instruction box. The supported conversions are from "INT to DINT" and "DINT to INT".

Example

The following example shows how the "Convert value "DINT to INT"" instruction for S7-1200/1500 F-CPUs works:



The instruction is always executed regardless of the signal state at enable input EN.

The value of the "TagIn_Value" operand is converted to an integer (16-bit) and the result is stored in the ""F_DB_1".TagOut_Value" operand.

When no overflow occurs during execution of the "Convert value "DINT to INT"" instruction, the ENO enable output has the signal state "1" and the "TagOut" operand is set.

You can also store the signal status of the ENO enable output in an (F-)DB, and centrally evaluate whether overflows have occurred for all or one group of instructions with overflow detection.

13.11.2 BO_W: Convert 16 data elements of data type BOOL to a data element of data type WORD (STEP 7 Safety V21)

Description

This instruction converts the 16 values of data type BOOL at inputs IN0 to IN15 to a value of data type WORD, which is made available at output OUT. The conversion takes place as follows: The i-th bit of the WORD value is set to 0 (or 1), if the value at input INi = 0 (or 1).

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN0	Input	BOOL	Bit 0 of WORD value
IN1	Input	BOOL	Bit 1 of WORD value
...			...
IN15	Input	BOOL	Bit 15 of WORD value
OUT	Output	WORD	WORD value consisting of IN0 to IN15

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	o	—	—	o	These versions have identical functions to version 1.0.
1.2	x	—	—	o	
1.3	x	o	—	o	
1.4	x	x	—	x	
2.0	x	x[1]	x	x[2]	

o This version is no longer supported.

[1] supported as of firmware V4.2

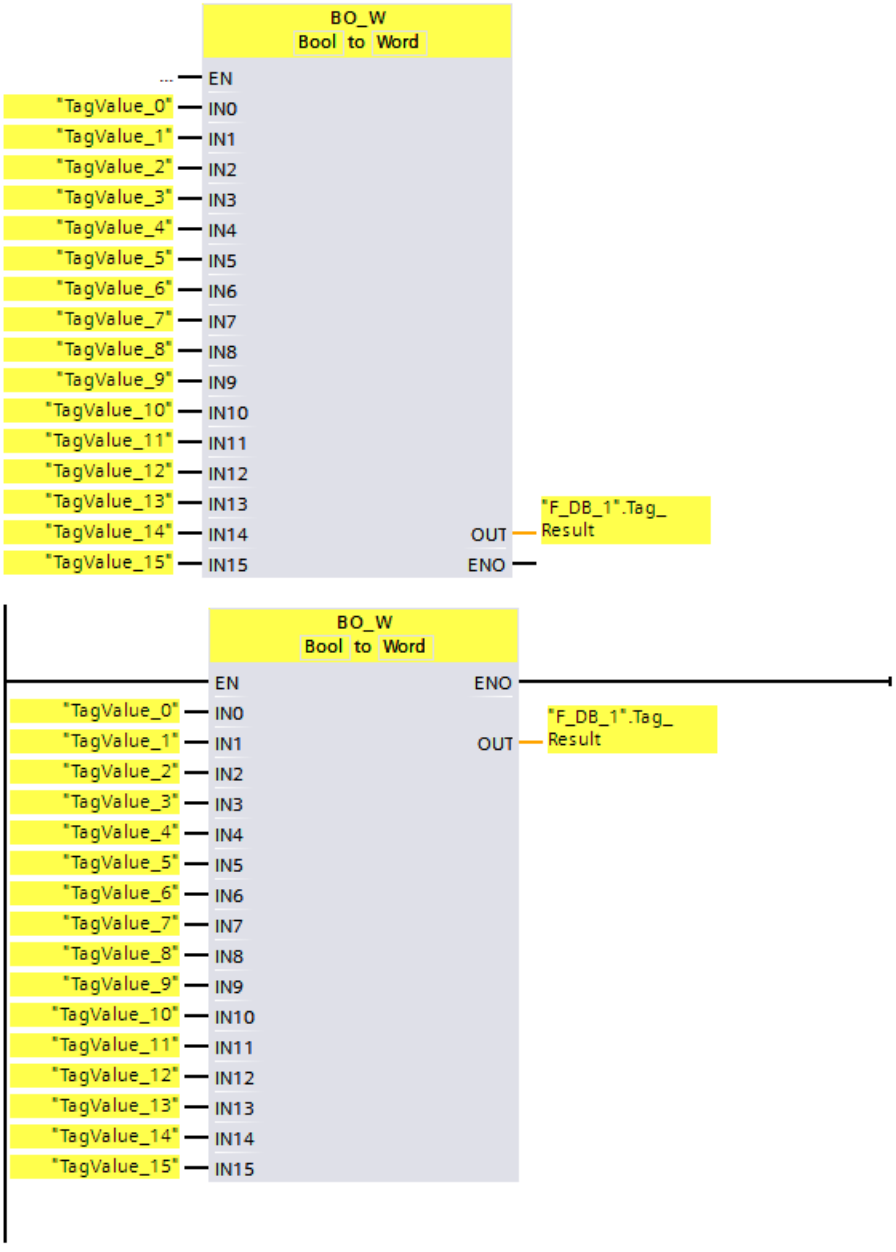
[2] supported as of firmware V2.0

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Example

The following example shows how the instruction works:



The following table shows how the instruction works using specific operand values:

Parameter	Operand	Value
IN0	TagValue_0	FALSE
IN1	TagValue_1	FALSE
...		...
IN13	TagValue_13	FALSE
IN14	TagValue_14	TRUE
IN15	TagValue_15	TRUE
OUT	"F_DB_1".Result	W#16#C000

The values of operands "TagValue_0" to "TagValue_15" are combined into a value of the data type WORD and assigned to operand ""F_DB_1".TagResult".

13.11.3 W_BO: Convert a data element of data type WORD to 16 data elements of data type BOOL (STEP 7 Safety V21)

Description

This instruction converts the value of data type WORD at input IN to 16 values of data type BOOL, which are provided at outputs OUT0 to OUT15. The conversion takes place as follows: Output OUTi is set to 0 (or 1), if the i-th bit of the WORD value is 0 (or 1).

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN	Input	WORD	WORD value
OUT0	Output	BOOL	Bit 0 of WORD value
OUT1	Output	BOOL	Bit 1 of WORD value
...			...
OUT15	Output	BOOL	Bit 15 of WORD value

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	o	—	—	o	These versions have identical functions to version 1.0.
1.2	x	—	—	o	
1.3	x	o	—	o	
1.4	x	x	—	x	
2.0	x	x[1]	x	x[2]	

o This version is no longer supported.

[1] supported as of firmware V4.2

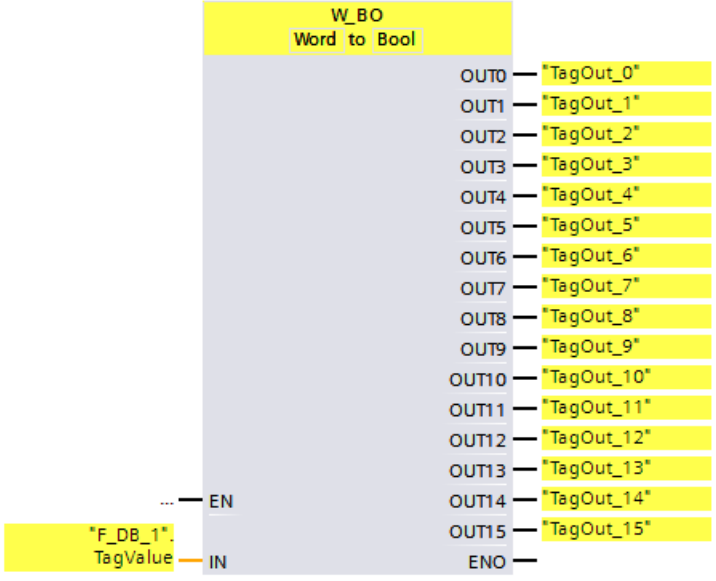
[2] supported as of firmware V2.0

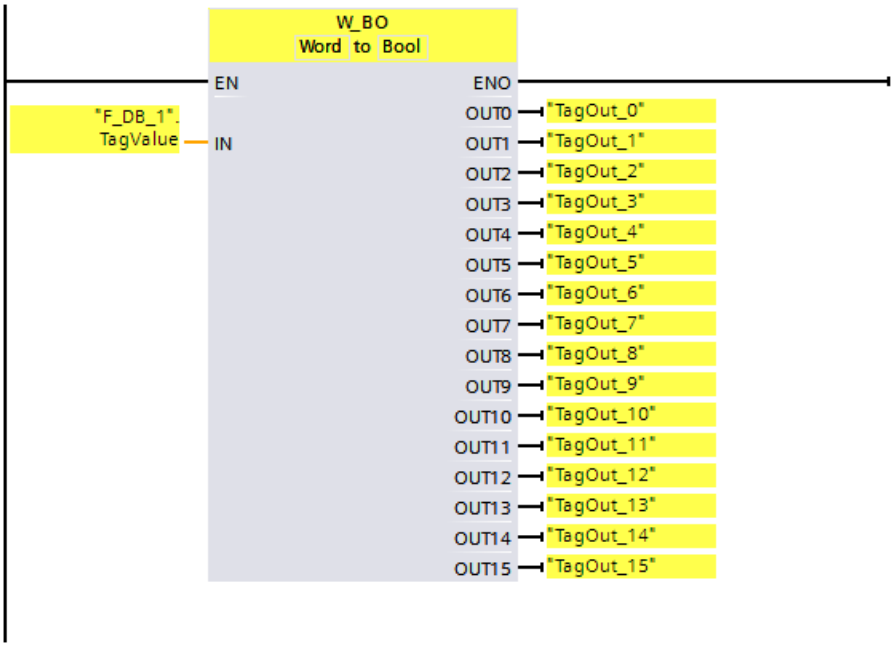
When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Example

The following example shows how the instruction works:





The following table shows how the instruction works using specific operand values:

Parameter	Operand	Value
IN	"F_DB_1".TagValue	W#16#C000
OUT0	TagOUT_0	FALSE
OUT1	TagOUT_1	FALSE
...		...
OUT13	TagOUT_13	FALSE
OUT14	TagOUT_14	TRUE
OUT15	TagOUT_15	TRUE

The value of operand ""F_DB_1".TagValue" of data type WORD is converted to the 16 values "TagOUT_0" to "TagOUT_15" of data type BOOL.

13.11.4 SCALE: Scale (STEP 7 Safety V21)

Description

This instruction scales the value at input IN in physical units between the low limit value at input LO_LIM and the high limit value at input HI_LIM. It is assumed that the value at input IN is between 0 and 27648. The scaling result is provided at output OUT.

The instruction uses the following equation:

OUT = [IN × (HI_LIM – LO_LIM)] / 27648 + LO_LIM

As long as the value at input IN is greater than 27648, output OUT is linked to HI_LIM and OUT_HI is set to 1.

As long as the value at input IN is less than 0, output OUT is linked to LO_LIM and OUT_LO is set to 1.

For inverse scaling, you must assign LO_LIM > HI_LIM. With inverse scaling, the output value at output OUT decreases while the input value at input IN increases.

Every call of the "Scale value" instruction must be assigned a data area in which the instruction data is stored. In addition, when the instruction is inserted in the program, the "Call options" dialog is automatically opened, where you can create a data block (single instance) (e.g., SCALE_DB_1) or a multi-instance (e.g., SCALE_Instance_1) for the "Scale value" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN	Input	INT	Input value to be scaled in physical units
HI_LIM	Input	INT	High limit value of value range of OUT
LO_LIM	Input	INT	Low limit value of value range of OUT
OUT	Output	INT	Result of scaling
OUT_HI	Output	BOOL	1 = Input value > 27648: OUT = HI_LIM
OUT_LO	Output	BOOL	1 = Input value < 0: OUT = LO_LIM

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	x	—	—	o	These versions have identical functions to version 1.0.
1.2	x	x	x	x	

o This version is no longer supported.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Behavior in the event of overflow or underflow of analog values and fail-safe value output

NOTE

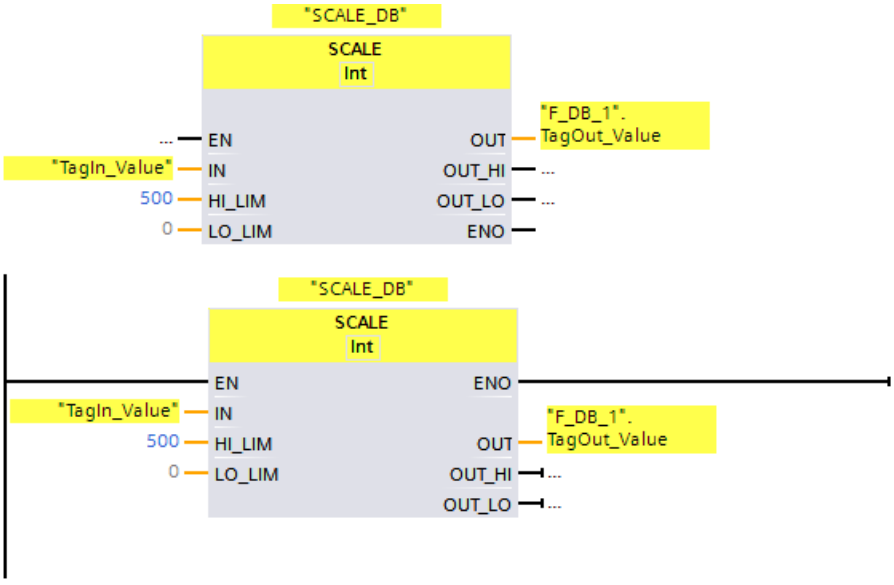
If inputs from the PII of an SM 336; AI 6 x 13Bit or SM 336; F-AI 6 x 0/4 ... 20 mA HART are used as input values, note that the F-system detects an overflow or underflow of a channel of this F-SM as an F-I/O fault or channel fault. The fail-safe value 0 is provided in place of $7FFF_H$ (for overflow) or 8000_H (for underflow) in the PII for the safety program.

If other fail-safe values should be output in this case, you need to evaluate the QBAD signal of the associated F-I/O or QBAD_I_xx signal / value status of the corresponding channel.

If the value in the PII of the F-SM is within the overrange or underrange, but is > 27648 or < 0, you can likewise branch to the output of an individual fail-safe value by evaluating outputs OUT_HI and OUT_LO, respectively.

Example

The following example shows how the instruction works:



When operand "TagIn_Value" = 20000, the result for ""F_DB_1".TagOut_Value" is 361.

13.11.5 SCALE_D: Scale to data type DINT (STEP 7 Safety V21) (S7-1200, S7-1500)

Description

This instruction scales the value at input IN in physical units between the low limit value at input LO_LIM and the high limit value at input HI_LIM. It is assumed that the value at input IN is between 0 and 27648. The scaling result is provided at output OUT.

The instruction uses the following equation:

OUT = [IN × (HI_LIM – LO_LIM)] / 27648 + LO_LIM

As long as the value at input IN is greater than 27648, output OUT is linked to HI_LIM and OUT_HI is set to 1.

As long as the value at input IN is less than 0, output OUT is linked to LO_LIM and OUT_LO is set to 1.

For inverse scaling, you must assign LO_LIM > HI_LIM. With inverse scaling, the output value at output OUT decreases while the input value at input IN increases.

Every call of the "Scale value to data type DINT" instruction must be assigned a data area in which the instruction data is stored. In addition, when the instruction is inserted in the program, the "Call options" dialog is automatically opened, where you can create a data block (single instance) (e.g., SCALE_D_DB_1) or a multi-instance (e.g., SCALE_D_Instance_1) for the "Scale value to data type DINT" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Parameter

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN	Input	INT	Input value to be scaled in physical units
HI_LIM	Input	DINT	High limit value of value range of OUT
LO_LIM	Input	DINT	Low limit value of value range of OUT
OUT	Output	DINT	Result of scaling
OUT_HI	Output	BOOL	1 = Input value > 27648: OUT = HI_LIM
OUT_LO	Output	BOOL	1 = Input value < 0: OUT = LO_LIM

Instruction versions

One version is available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
2.0	—	x[1]	x	x[2]	

[1] supported as of firmware V4.2

[2] supported as of firmware V2.0

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Behavior in the event of overflow or underflow of analog values and fail-safe value output

NOTE

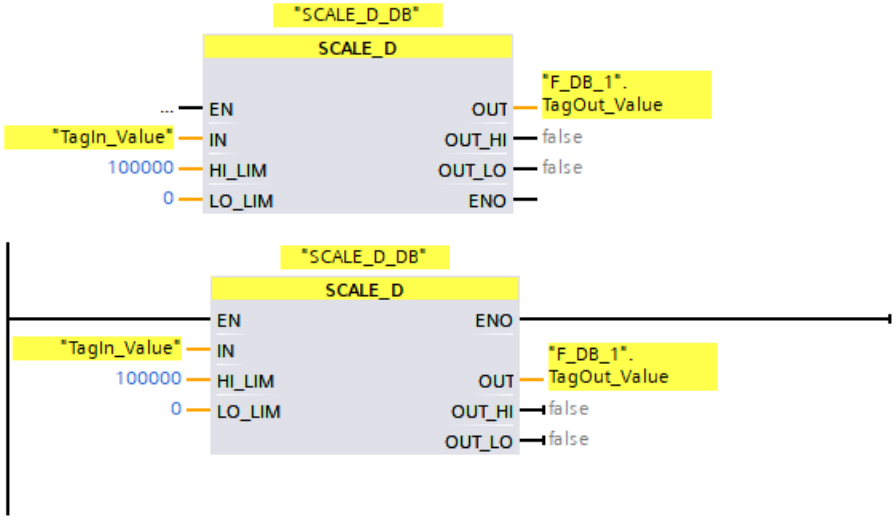
If inputs from the PII of an SM 336; AI 6 x 13Bit or SM 336; F-AI 6 x 0/4 ... 20 mA HART are used as input values, note that the F-system detects an overflow or underflow of a channel of this F-SM as an F-I/O fault or channel fault. The fail-safe value 0 is provided in place of $7FFF_H$ (for overflow) or 8000_H (for underflow) in the PII for the safety program.

If other fail-safe values should be output in this case, you need to evaluate the QBAD signal of the associated F-I/O or QBAD_I_xx signal / value status of the corresponding channel.

If the value in the PII of the F-SM is within the overrange or underrange, but is > 27648 or < 0, you can likewise branch to the output of an individual fail-safe value by evaluating outputs OUT_HI and OUT_LO, respectively.

Example

The following example shows how the instruction works:



When the operand "TagIn_Value" = 20000, the result for ""F_DB_1".TagOut_Value" is 72337.

13.12 Program control operations

13.12.1 JMP: Jump if RLO = 1 (STEP 7 Safety V21)

Description

You can use the "Jump if RLO = 1" instruction to interrupt the linear execution of the program and resume it in another network. The destination network must be identified by a jump label (Page 592) (LABEL). The description of the jump label is specified in the placeholder above the instruction.

The specified jump label must be in the same block in which the instruction is executed. The name you specify can only occur once in the block.

If the result of logic operation (RLO) at the input of the instruction is "1" or the input is not connected, the jump to the network identified by the jump label is executed. The jump direction can be towards higher or lower network numbers.

If the result of logic operation (RLO) at the input of the instruction is "0", the program continues executing in the next network.

NOTE

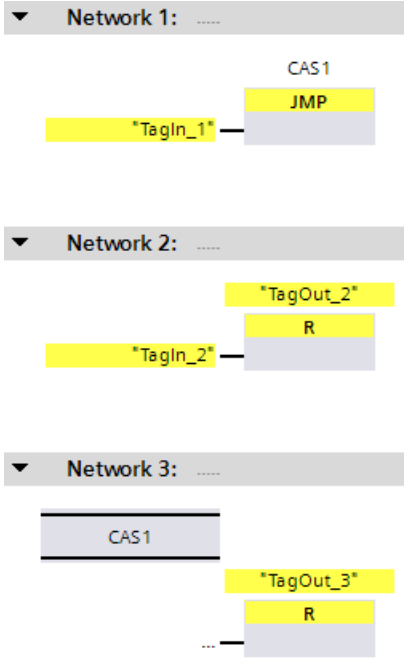
(S7-1200, S7-1500)
If the jump destination (jump label) for an instruction "JMP" or "JMPN" is above the associated instruction "JMP" or "JMPN" (backwards jump), you cannot insert any other instructions for program control (JMP, JMPN, RET, jump label) between them.
Exception: You can insert an instruction "JMP" or "JMPN" between them if you also insert the associated jump destination in between as well as below the associated instruction "JMP" or "JMPN" (forward jump).
Non-compliance can lead to compilation errors or to the F-CPU going to STOP.

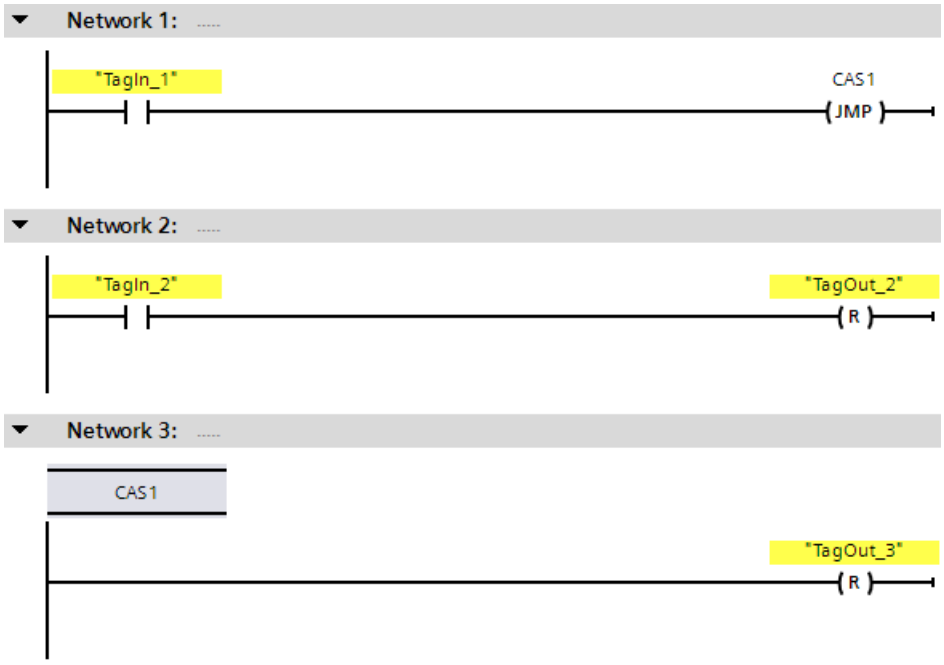
NOTE

You are not permitted to insert any SENDDP or SENDS7 instructions between an instruction JMP or JMPN and the associated jump destination (jump label).

Example

The following example shows how the instruction works:





When operand "TagIn_1" has signal state "1", the "Jump if RLO = 1" instruction is executed. The linear execution of the program is interrupted and continues in Network 3, which is identified by the jump label CAS1. When input "TagIn_3" has signal state "1", output "TagOut_3" is reset.

13.12.2 JMPN: Jump if RLO = 0 (STEP 7 Safety V21)

Description

You can use the "Jump if RLO = 0" instruction to interrupt the linear execution of the program and resume it in another network, when the result of logic operation at the input of the instruction is "0". The destination network must be identified by a jump label (Page 592) (LABEL). The description of the jump label is specified in the placeholder above the instruction.

The specified jump label must be in the same block in which the instruction is executed. The name you specify can only occur once in the block.

If the result of logic operation (RLO) at the input of the instruction is "0", the jump to the network identified by the jump label is executed. The jump direction can be towards higher or lower network numbers.

If the result of logic operation (RLO) at the input of the instruction is "1", the program continues executing in the next network.

NOTE

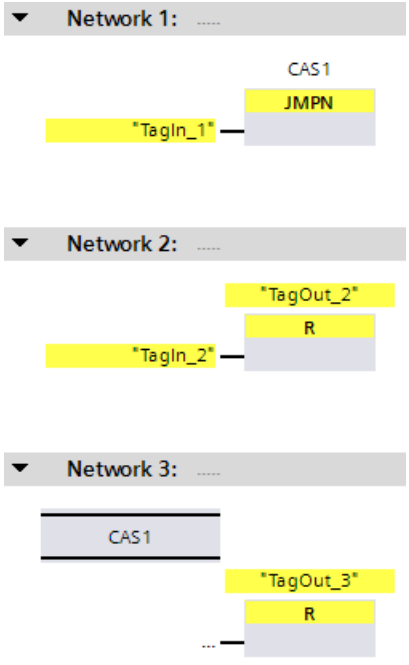
(S7-1200, S7-1500)
If the jump destination (jump label) for an instruction "JMP" or "JMPN" is above the associated instruction "JMP" or "JMPN" (backwards jump), you cannot insert any other instructions for program control (JMP, JMPN, RET, jump label) between them.
Exception: You can insert an instruction "JMP" or "JMPN" between them if you also insert the associated jump destination in between as well as below the associated instruction "JMP" or "JMPN" (forward jump).
Non-compliance can lead to compilation errors or to the F-CPU going to STOP.

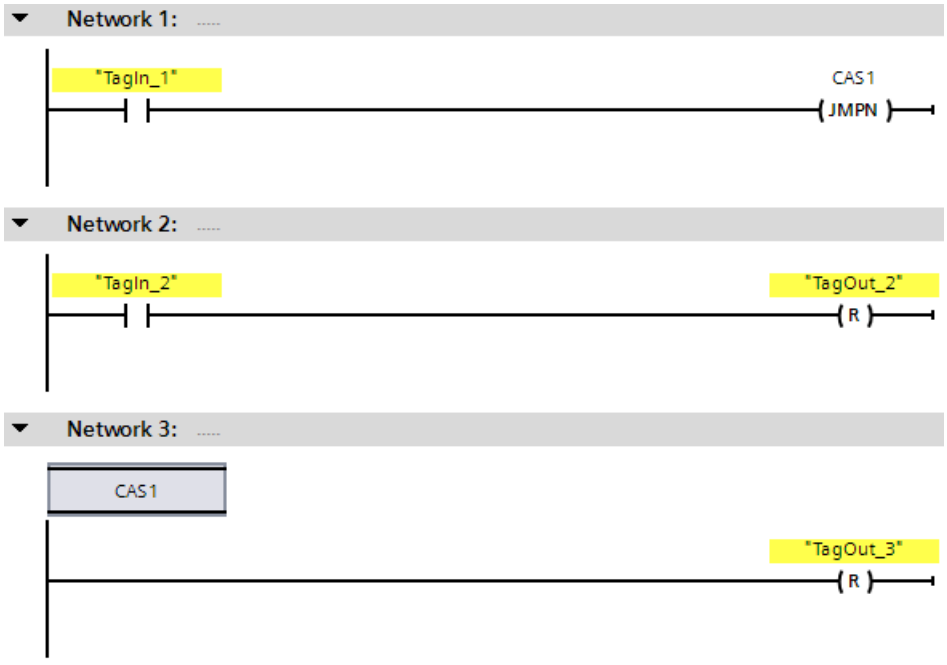
NOTE

You are not permitted to insert any SENDDP or SENDS7 instructions between an instruction JMP or JMPN and the associated jump destination (jump label).

Example

The following example shows how the instruction works:





When operand "TagIn_1" has signal state "0", the "Jump if RLO = 0" instruction is executed. The linear execution of the program is interrupted and continues in Network 3, which is identified by the jump label CAS1. When input "TagIn_3" has signal state "1", output "TagOut_3" is reset.

13.12.3 LABEL: Jump label (STEP 7 Safety V21)

Description

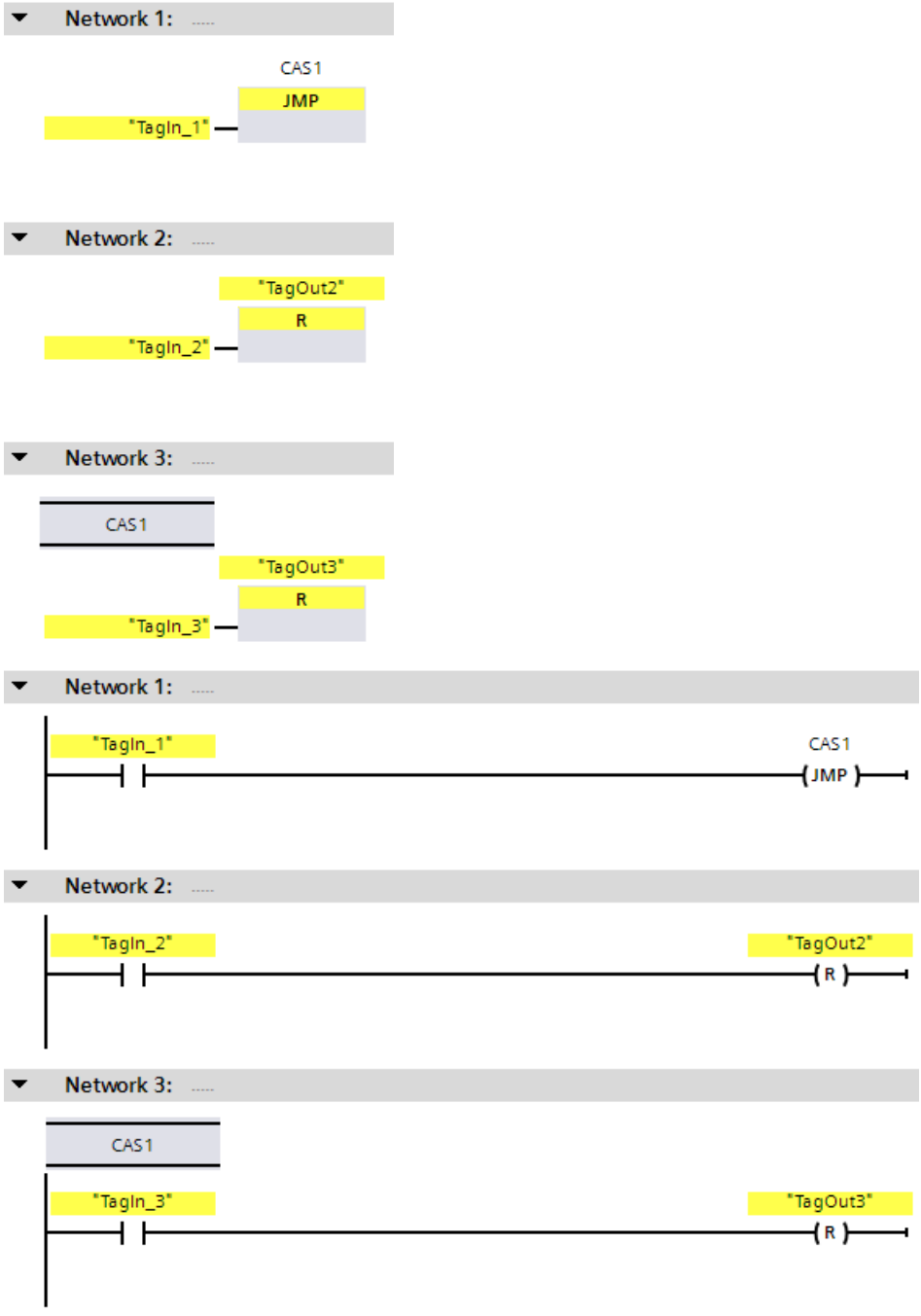
You can use a jump label to specify a destination network, in which the program execution should resume after a jump.

The jump label and the instruction in which the jump label is specified must be located in the same block. The name of a jump label can only be assigned once in a block.

Only one jump label can be placed in a network. To each jump label can be jumped from several locations.

Example

The following example shows how the instruction works:



When operand "TagIn_1" has signal state "1", the "Jump if RLO = 1" instruction is executed. The linear execution of the program is interrupted and continues in Network 3, which is identified by the jump label CAS1. When input "TagIn_3" has signal state "1", output "TagOut_3" is reset.

See also

JMP: Jump if RLO = 1 (STEP 7 Safety V21) (Page 588)
JMPN: Jump if RLO = 0 (STEP 7 Safety V21) (Page 590)
RET: Return (STEP 7 Safety V21) (Page 594)

13.12.4 RET: Return (STEP 7 Safety V21)

Description

You can use the "Return" instruction to stop the processing of a block.

If the result of logic operation (RLO) at the input of the "Return" instruction is "1" or the box input of the S7-1200/1500 F-CPUs is not connected in FBD, program execution is terminated in the currently called block and continued in the calling block (for example, in the main safety block) after the call function. If the RLO at the input of the "Return" instruction is "0", the instruction is not executed. Program execution continues in the next network of the called block.

Influencing the status of the call function (ENO) is irrelevant, because the enable output "ENO" cannot be connected.

NOTE

(S7-300, S7-400) You may not program a RET instruction in the main safety block.

Example

The following example shows how the instruction works:



When the "TagIn" operand delivers signal state "1", the "Return" instruction is executed. Program execution is terminated in the called block and continues in the calling block.

See also

JMP: Jump if RLO = 1 (STEP 7 Safety V21) (Page 588)
JMPN: Jump if RLO = 0 (STEP 7 Safety V21) (Page 590)
LABEL: Jump label (STEP 7 Safety V21) (Page 592)

13.12.5 OPN: Open global data block (STEP 7 Safety Advanced V21) (S7-300, S7-400)

Description

You can use the "Open global data block" instruction to open a data block. The number of the data block is transferred to the DB register. Subsequent DB commands access the relevant blocks depending on the register contents.

NOTE

Note when using the "Open global data block" instruction that the content of the DB register can be changed after calls of F-FB/F-FC and "fully qualified DB accesses", such that there is no guarantee that the last data block you opened with "Open global data block" is still open.

You should therefore use the following method for addressing data to avoid errors when accessing data of the DB register:

- Use symbolic addressing.
- Use only fully qualified DB accesses.

If you still want to use the "Open global data block" operation, you must ensure that the DB register is restored by repeating the "Open global data block" instruction following calls of F-FB/F-FC and "fully qualified DB accesses." Otherwise, a malfunction could result.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
<Data block>	Input	BLOCK_DB	Data block that is opened

"Fully qualified DB access"

The initial access to data of a data block in an F-FB/F-FC must always be a "fully qualified DB access," or it must be preceded by the "Open global data block" instruction. This also applies to the initial access to data of a data block after a jump label.

An example of "fully qualified DB access" and "non-fully qualified DB access" is provided in Restrictions in the programming languages FBD/LAD (Page 119).

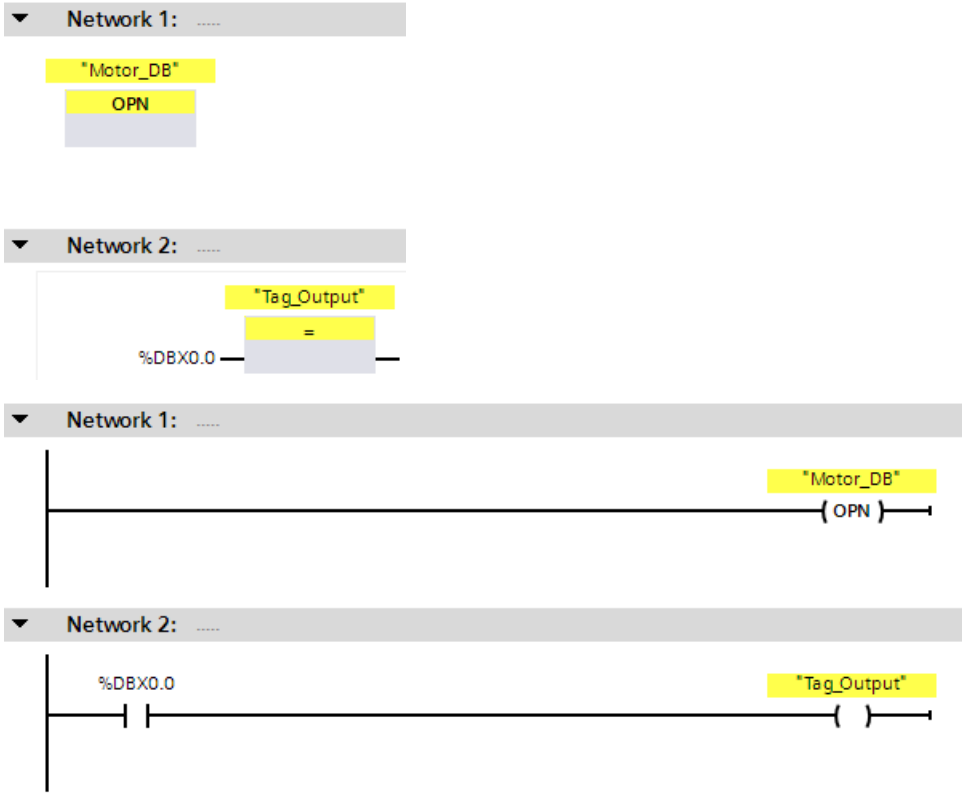
Access to instance DBs

You can also access instance DBs of F-FBs with fully qualified access, e.g., for transfer of block parameters. It is not possible to access static local data in single/multi-instances of other F-FBs.

Note that accessing instance DBs of F-FBs that are not called in the safety program can cause the F-CPU to go to STOP mode.

Example

The following example shows how the instruction works:



The "Motor_DB" data block is called in network 1. The number of the data block is transferred to the DB register. The "DBX0.0" operand is queried in network 2. The signal state of the "DBX0.0" operand is assigned to the "Tag_Output" operand.

13.13 Word logic operations

13.13.1 AND: AND logic operation (STEP 7 Safety V21)

Description

You can use the "AND logic operation" instruction to combine the value at input IN1 to the value at input IN2 bit-by-bit by AND logic and query the result at output OUT.

When the instruction is executed, bit 0 of the value at input IN1 and bit 0 of the value at input IN2 are ANDed. The result is stored in bit 0 of output OUT. The same logic operation is executed for all other bits of the specified values.

The result bit has signal state "1" only when both of the bits in the logic operation also have signal state "1". If one of the two bits of the logic operation has signal state "0", the corresponding result bit is reset.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

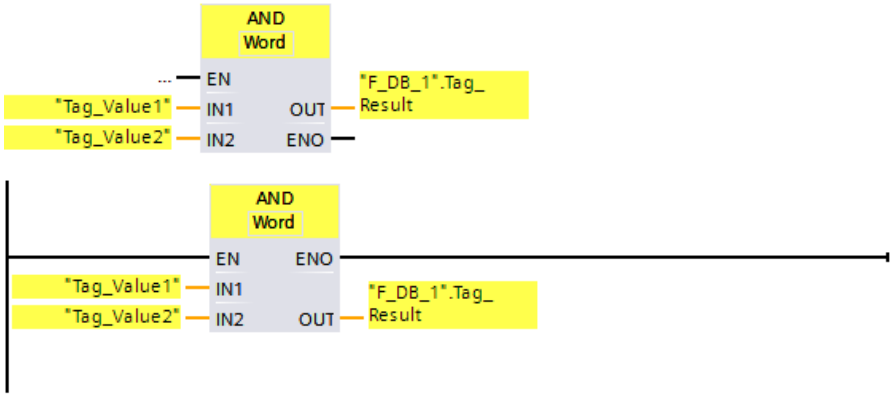
Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN1	Input	WORD	First value of logic operation
IN2	Input	WORD	Second value of logic operation
OUT	Output	WORD	Result of the instruction

Example

The following example shows how the instruction works:



IN1	"Tag_Value1" = 01010101 01010101
IN2	"Tag_Value2" = 00000000 00001111
OUT	"F_DB_1"."Tag_Result" = 00000000 00000101

The instruction is always executed regardless of the signal state at enable input "EN". The value of the "Tag_Value1" operand and the value of the "Tag_Value2" operand are ANDed. The result is mapped bit-by-bit and output in the ""F_DB_1".Tag_Result" operand.

13.13.2 OR: OR logic operation (STEP 7 Safety V21)

Description

You can use the "OR logic operation" instruction to connect the value at input IN1 input to the value at input IN2 bit-by-bit by OR logic and query the result at output OR.

When the instruction is executed, bit 0 of the value at input IN1 and bit 0 of the value at input IN2 are ORed. The result is stored in bit 0 of output OUT. The same logic operation is executed for all bits of the specified tags.

The result bit has signal state "1" when at least one of the two bits in the logic operation has signal state "1". If both of the bits of the logic operation have signal state "0", the corresponding result bit is reset.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

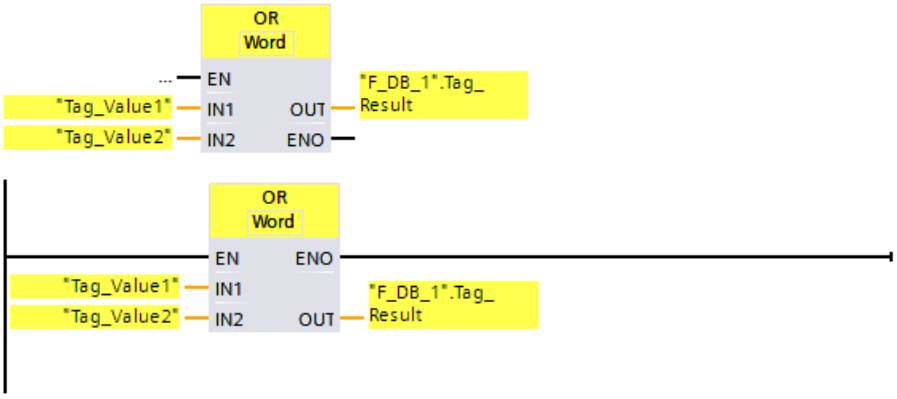
Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN1	Input	WORD	First value of logic operation
IN2	Input	WORD	Second value of logic operation
OUT	Output	WORD	Result of the instruction

Example

The following example shows how the instruction works:



IN1	"Tag_Value1" = 01010101 01010101
IN2	"Tag_Value2" = 00000000 00001111
OUT	"F_DB_1"."Tag_Result" = 01010101 01011111

The instruction is always executed regardless of the signal state at enable input "EN". The value of the "Tag_Value1" operand and the value of the "Tag_Value2" operand are ORed. The result is mapped bit-by-bit and output in the ""F_DB_1".Tag_Result" operand.

13.13.3 XOR: EXCLUSIVE OR logic operation (STEP 7 Safety V21)

Description

You can use the "EXCLUSIVE OR logic operation" instruction to combine the value at input IN1 and the value at input IN2 bit-by-bit by EXCLUSIVE OR logic and query the result at output OUT.

When the instruction is executed, bit 0 of the value at input IN1 input and bit 0 of the value at input IN2 are logically combined by EXCLUSIVE OR. The result is stored in bit 0 of output OUT. The same logic operation is executed for all other bits of the specified value.

The result bit has signal state "1" when one of the two bits in the logic operation has signal state "1". If both of the bits of the logic operation have signal state "1" or "0", the corresponding result bit is reset.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

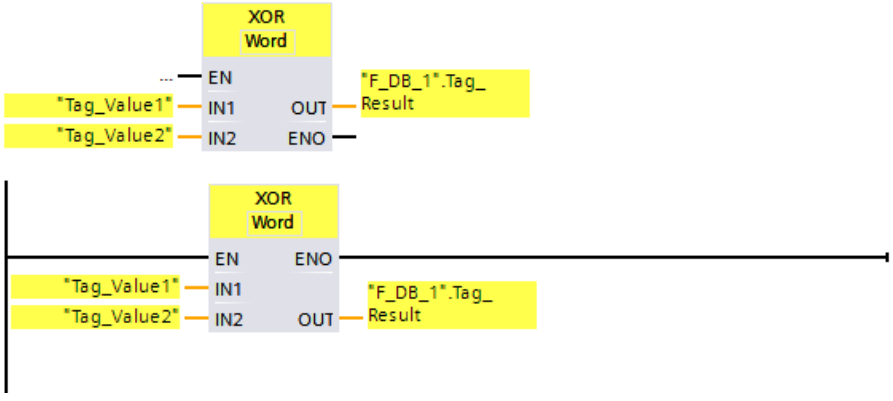
Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN1	Input	WORD	First value of logic operation
IN2	Input	WORD	Second value of logic operation
OUT	Output	WORD	Result of the instruction

Example

The following example shows how the instruction works:



IN1	"Tag_Value1" = 01010101 01010101
IN2	"Tag_Value2" = 00000000 00001111
OUT	"F_DB_1"."Tag_Result" = 01010101 01011010

The instruction is always executed regardless of the signal state at enable input "EN". The value of the "Tag_Value1" operand and the value of the "Tag_Value2" operand are logically combined by EXCLUSIVE OR. The result is mapped bit-by-bit and output in the ""F_DB_1".Tag_Result" operand.

13.14 Shift and rotate

13.14.1 SHR: Shift right (STEP 7 Safety V21)

Description

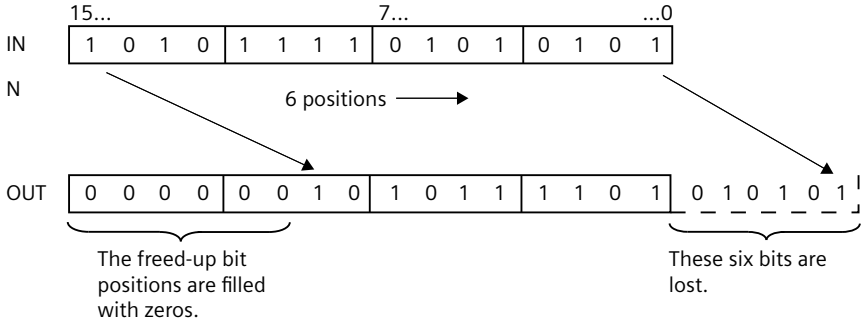
Use the "Shift right" instruction to shift the content of the operand at input IN bit-by-bit to the right and query the result at output OUT. Use input N to specify the number of bit positions by which the specified value is to be moved.

If the value at input N is "0", the value at input IN is copied to the operand at output OUT.

If the value at input N is greater than the number of available bit positions, the operand value at input IN is shifted to the right by the available number of bit positions.

The bit locations that are freed up in the left area of the operand during the shift operation are filled with zeros.

The following figure shows how the content of an operand of data type WORD is moved by 6 bit positions to the right:



Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

NOTE

S7-300/400:

Only the low-byte is evaluated from input N.

S7-1200/1500:

If the value at input N < 0, the output OUT is set to 0.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN	Input	WORD	Value that is shifted
N	Input	INT	Number of bit positions by which the value is shifted
OUT	Output	WORD	Result of the instruction

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	o	—	—	o	These versions have identical functions to version 1.0.
1.2	x	—	—	o	
1.3	x	o	—	o	
1.4	x	x	—	x	
2.0	x	x[1]	x	x[2]	

o This version is no longer supported.

[1] supported as of firmware V4.2

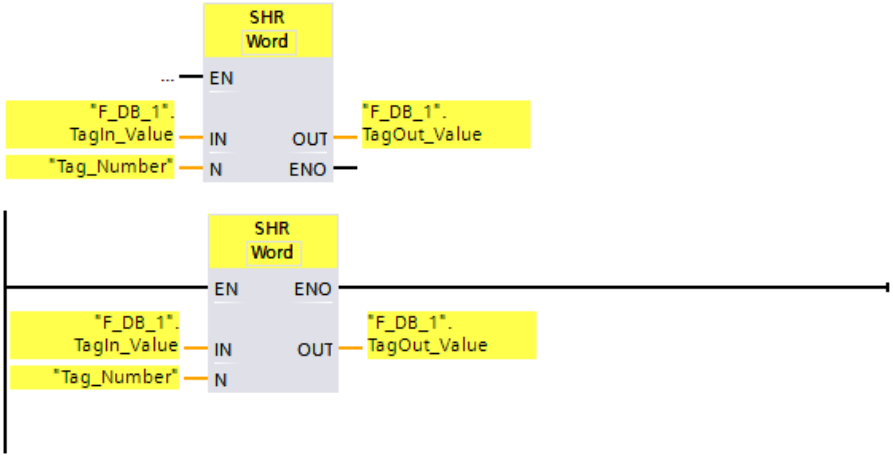
[2] supported as of firmware V2.0

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Example

The following example shows how the instruction works:



The following table shows how the instruction works using specific operand values:

Parameter	Operand	Value
IN	"F_DB_1".TagIn_Value	0011 1111 1010 1111
N	Tag_Number	3
OUT	"F_DB_1".TagOut_Value	0000 0111 1111 0101

The instruction is always executed regardless of the signal state at enable input "EN". The content of the operand ""F_DB_1".TagIn_Value" is moved three bit positions to the right. The result is output at output ""F_DB_1".TagOut_Value".

13.14.2 SHL: Shift left (STEP 7 Safety V21)

Description

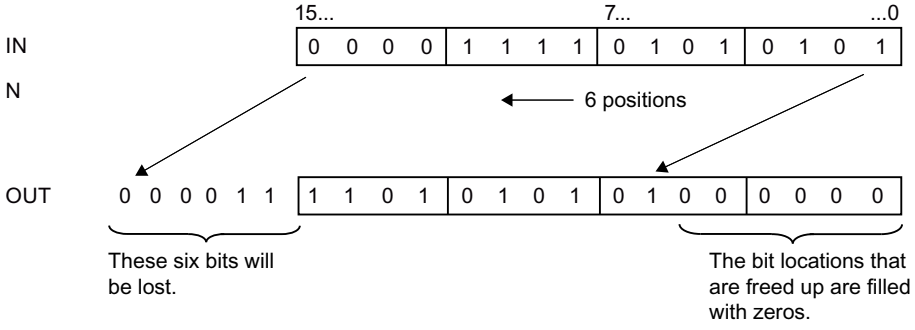
Use the "Shift left" instruction to shift the content of the operand at input IN bit-by-bit to the left and query the result at output OUT. Use input N to specify the number of bit positions by which the specified value is to be moved.

If the value at input N is "0", the value at input IN is copied to the operand at output OUT.

If the value at input N is greater than the number of available bit positions, the operand value at input IN is shifted to the left by the available number of bit positions.

The bit positions that are freed up in the right area of the operand during the shift operation are filled with zeros.

The following figure shows how the content of an operand of data type WORD is moved by 6 bit positions to the left:



Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

NOTE

S7-300/400:

Only the low-byte is evaluated from input N.

S7-1200/1500:

If the value at input N < 0, the output OUT is set to 0.

Parameters

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN	Input	WORD	Value that is shifted
N	Input	INT	Number of bit positions by which the value is shifted
OUT	Output	WORD	Result of the instruction

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	o	—	—	o	These versions have identical functions to version 1.0.
1.2	x	—	—	o	
1.3	x	o	—	o	
1.4	x	x	—	x	
2.0	x	x[1]	x	x[2]	

o This version is no longer supported.

[1] supported as of firmware V4.2

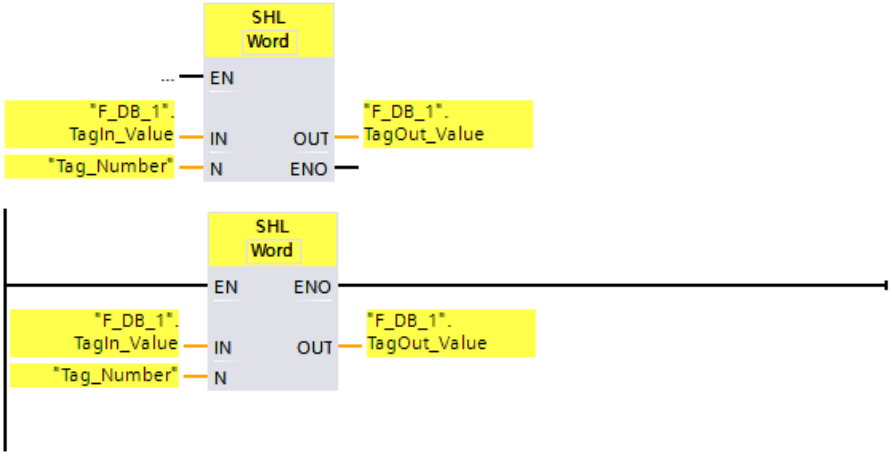
[2] supported as of firmware V2.0

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Example

The following example shows how the instruction works:



The following table shows how the instruction works using specific operand values:

Parameter	Operand	Value
IN	"F_DB_1".TagIn_Value	0011 1111 1010 1111
N	Tag_Number	4
OUT	"F_DB_1".TagOut_Value	1111 1010 1111 0000

The instruction is always executed regardless of the signal state at enable input "EN". The content of the operand ""F_DB_1".TagIn_Value" is moved four bit positions to the left. The result is output at output ""F_DB_1".TagOut_Value".

13.15 Operating

13.15.1 ACK_OP: Fail-safe acknowledgment (STEP 7 Safety V21)

Description (S7-300, S7-400)

This instruction enables fail-safe acknowledgment from an operator control and monitoring system. It allows, for example, reintegration of F-I/O to be controlled from the operator control and monitoring system. Acknowledgment takes place in two steps:

- InOut IN changes to a value of 6 for exactly one cycle
- InOut IN changes to a value of 9 within a minute for exactly one cycle

Once InOut IN has changed to a value of 6, the instruction evaluates whether this parameter has changed to a value of 9 after 1 second, at the earliest, or one minute, at the latest. Output OUT (output for acknowledgment) is then set to 1 for one cycle.

If an invalid value is input or if InOut IN has not changed to the value 9 within one minute or the change occurred before one second has elapsed, then InOut IN is reset to 0, and both steps listed above must be repeated.

During the time in which InOut IN must change from 6 to the value 9, output Q is set to 1. Otherwise, Q has a value of 0.

Every call of the "Fail-safe acknowledgment" instruction must be assigned a data area in which the instruction data is stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., ACK_OP_DB_1) or a multi-instance (e.g., ACK_OP_Instance_1) for the "Fail-safe acknowledgment" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

NOTE

A separate data area must be used for each call of ACK_OP. Each call can be processed only once in an F-runtime group cycle.

If this is not observed, the behavior described under "Description" is no longer ensured.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

⚠ WARNING

The two acknowledgment steps must **not** be triggered by one single operation, for example, by programming them including the time conditions and using one function key to trigger them.

The two separate acknowledgment steps also prevent erroneous triggering of an acknowledgment, for example, by your non-fail-safe operator control and monitoring system. *(S013)*

⚠ WARNING

If you have operator control and monitoring systems and F-CPUs that are interconnected and use the ACK_OP instruction for fail-safe acknowledgment, you need to ensure that the intended F-CPU will be addressed **before** you perform the two acknowledgment steps.

- Store a network-wide* unique name for the F-CPU in a DB of your standard user program in each F-CPU.
- In your operator control and monitoring system, set up a text box from which you can read out the F-CPU name from the DB online before executing the two acknowledgment steps.
- Optional:

  In your operator control and monitoring system, set up a text box to permanently store the F-CPU name. Then, you can determine whether the intended F-CPU is being addressed by simply comparing the F-CPU name read out online with the permanently stored designation. *(S014)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet.

NOTE

You can read out output Q by means of operator control and monitoring systems or, if applicable, evaluate it in your standard user program.

You can supply InOut IN with a separate memory word or DBW of a DB of the standard user program supply for each call of the ACK_OP instruction.

NOTE

The configuration of your operator control and monitoring system does not have any effect on the Collective F-Signature.

Description (S7-1200, S7-1500)

This instruction enables fail-safe acknowledgment from an operator control and monitoring system. It allows, for example, reintegration of F-I/O to be controlled from the operator control and monitoring system. Acknowledgment takes place in two steps:

- InOut IN changes to a value of 6 for exactly one cycle
- InOut IN changes to the value at the ACK_ID input within a minute for exactly one cycle

Once InOut IN has changed to a value of 6, the instruction evaluates whether this parameter has changed to a value at the ACK_ID input after 1 second, at the earliest, or one minute, at the latest. Output OUT (output for acknowledgment) is then set to 1 for one cycle.

If an invalid value is input or if InOut IN has not changed to the value at the ACK_ID input within one minute or the change occurred before one second has elapsed, then iInOut IN is reset to 0, and both steps listed above must be repeated.

During the time in which InOut IN must change from 6 to the value at the ACK_ID input, output Q is set to 1. Otherwise, Q has a value of 0.

Every call of the "Fail-safe acknowledgment" instruction must be assigned a data area in which the instruction data is stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., ACK_OP_DB_1) or a multi-instance (e.g., ACK_OP_Instance_1) for the "Fail-safe acknowledgment" instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

NOTE

A separate data area must be used for each call of ACK_OP. Each call can be processed only once in an F-runtime group cycle.

If this is not observed, the behavior described under "Description" is no longer ensured.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

⚠ WARNING

The two acknowledgment steps must **not** be triggered by one single operation, for example, by programming them including the time conditions and using one function key to trigger them.

The two separate acknowledgment steps also prevent erroneous triggering of an acknowledgment, for example, by your non-fail-safe operator control and monitoring system. *(S013)*

⚠ WARNING

If you have operator control and monitoring systems and F-CPUs that are interconnected and use the ACK_OP instruction for fail-safe acknowledgment, you need to ensure that the intended F-CPU will be addressed **before** you perform the two acknowledgment steps.

Alternative 1:

- The value for each identifier of the acknowledgment (ACK_ID input; data type: INT) can be freely selected in the range from 9 to 30000, but must be unique network-wide* for all instances of the ACK_OP instruction.
  You must supply the ACK_ID input with constant values when calling the instruction. Direct read or write access to the ACK_ID tag in the associated instance DB is not permitted!

Alternative 2:

- Store a network-wide* unique name for the F-CPU in a DB of your standard user program in each F-CPU.
- In your operator control and monitoring system, set up a text box from which you can read out the F-CPU name from the DB online before executing the two acknowledgment steps.
- Optional:

  In your operator control and monitoring system, set up a text box to permanently store the F-CPU name. Then, you can determine whether the intended F-CPU is being addressed by simply comparing the F-CPU name read out online with the permanently stored designation. *(S047)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet.

NOTE

You can read out output Q by means of operator control and monitoring systems or, if applicable, evaluate it in your standard user program.

You must supply InOut IN with a separate memory word or DBW of a DB of the standard user program for each call of the ACK_OP instruction.

NOTE

The supply of the IN iInOut of the ACK_OP instruction as well as the configuration of your operator control and monitoring system do not have any effect on the Collective F-Signature, the Collective F-SW-Signature or the signature of the block that calls the ACK_OP instruction.

Changes to the supply of the IN InOut or to the configuration of your operator control and monitoring system therefore do not result in a changed Collective F-Signature/Collective F-SW-Signature/signature of the calling block.

Parameters (S7-300, S7-400)

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
IN	InOut	INT	Input variable from operator control and monitoring system
OUT	Output	BOOL	Output for acknowledgment
Q	Output	BOOL	Time status

Parameters (S7-1200, S7-1500)

The following table shows the parameters of the instruction:

Parameter	Declaration	Data type	Description
ACK_ID	Input	INT	Identifier of the acknowledgment (9 to 30000)
IN	InOut	INT	Input variable from operator control and monitoring system
OUT	Output	BOOL	Output for acknowledgment
Q	Output	BOOL	Time status

Instruction versions

A number of versions are available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	x	—	—	o	These versions have identical functions to version 1.0 for S7-300/400 F-CPUs. The input ACK_ID must also be taken into consideration for S7-1200/1500 F-CPUs.
1.2	x	o	—	o	
1.3	x	x	x	x	

o This version is no longer supported.

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Timing imprecision resulting from the update time and the tolerance of the time base used in the instruction

Refer to section "Temporal blurring for instructions with time processing (Page 450)" for more on this.

Example

An example how the instruction is used is available under Implementing User Acknowledgment in the Safety Program of the F-CPU of a DP Master or IO controller (Page 190).

See also

Implementing user acknowledgment in the safety program of the F-CPU of a I-slave or I-device (Page 194)

13.16 Additional instructions

13.16.1 LAD

13.16.1.1 ---| |--- OV: Get status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400)

Description

You can use the "Get status bit OV" instruction to detect whether a number range overflow occurred in the last arithmetic instruction processed.

The "Get status bit OV" instruction functions like a normally open contact. If the query is fulfilled, the instruction has signal state "1". If the query is not fulfilled, the instruction has signal state "0".

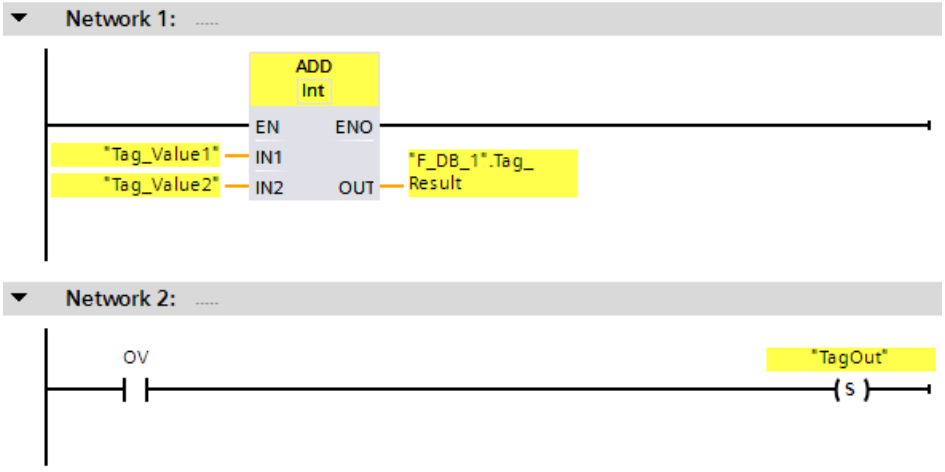
The "Get status bit OV" evaluation must be inserted in the network that follows the instruction that influences the OV. This network must not contain any jump labels.

NOTE

The execution time of the OV-affecting instruction is extended when the "Get status bit OV " instruction is used (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).

Example

The following example shows how the instruction works:



The "Add" instruction is always executed (regardless of the signal state at enable input EN).

The value of the "Tag_Value1" operand is added to value of the Tag_Value2 operand. The result of the addition is stored in the ""F_DB_1".Tag_Result" operand.

If an overflow occurs during execution of the "Add" instruction, the status bit OV is set to "1". In network 2, following the query of the status bit OV, the "Set output" (S) instruction is executed and the "TagOut" operand is set.

13.16.1.2 ---| / |--- OV: Get negated status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400)

Description

You can use the "Get negated status bit OV" instruction to detect whether a number range overflow occurred in the last arithmetic instruction processed. This instruction is only available in LAD.

The "Get negated status bit OV" instruction functions like a normally closed contact. If the query is satisfied, the instruction has signal state "0". If the query is not satisfied, the instruction has signal state "1".

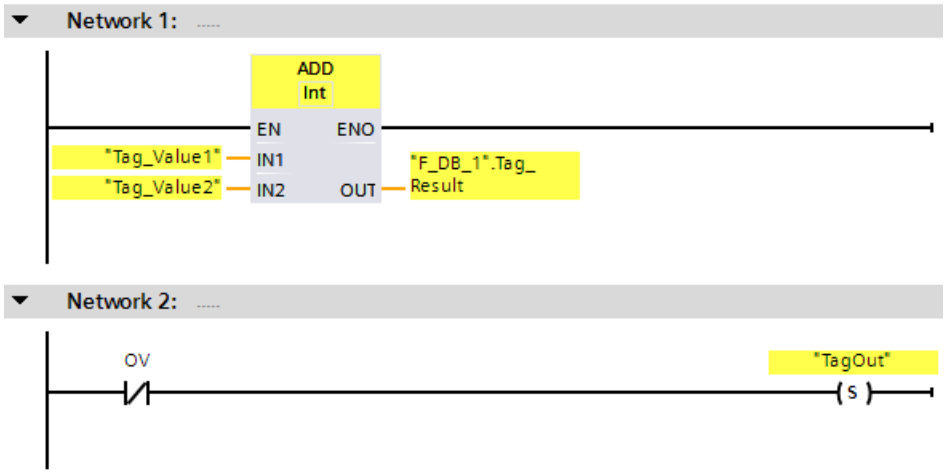
The "Get negated status bit OV" evaluation must be inserted in the network following the instruction that influences the OV. This network must not contain any jump labels.

NOTE

The execution time of the OV-affecting instruction is extended when the "Get negated status bit OV " instruction is used (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).

Example

The following example shows how the instruction works:



The "Add" instruction is always executed (regardless of the signal state at enable input EN).

The value of the "Tag_Value1" operand is added to value of the Tag_Value2 operand. The result of the addition is stored in the ""F_DB_1".Tag_Result" operand.

If an overflow does *not* occur during execution of the "Add" instruction, the status bit OV is reset to "0". In network 2, following the query of the status bit OV, the "Set output" (S) instruction is executed and the "TagOut" operand is set.

13.16.2 FBD

13.16.2.1 OV: Get status bit OV (STEP 7 Safety Advanced V21) (S7-300, S7-400)

Description

You can use the "Get status bit OV" instruction to detect whether a number range overflow occurred in the last arithmetic instruction processed.

The "Get status bit OV" evaluation must be inserted in the network that follows the instruction that influences the OV. This network must not contain any jump labels.

If the query is fulfilled, the instruction has signal state "1". If the query is not fulfilled, the instruction has signal state "0".

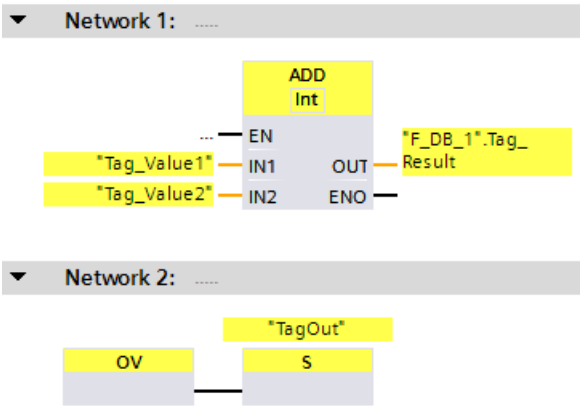
You can program a query of status bit OV for "0" with the "Invert RLO" instruction.

NOTE

The execution time of the OV-affecting instruction is extended when the "Get status bit OV " instruction is used (see also Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831)).

Example

The following example shows how the instruction works:



The value of the "Tag_Value1" operand is added to value of the Tag_Value2 operand. The result of the addition is stored in the ""F_DB_1".Tag_Result" operand.

If an overflow occurs during execution of the "Add" instruction, the status bit OV is set to "1". In network 2, following the query of the status bit OV, the "Set output" (S) instruction is executed and the "TagOut" operand is set.

13.17 Communication

13.17.1 PROFIBUS/PROFINET

13.17.1.1 SENDDP and RCVDP: Send and receive data via PROFIBUS DP/PROFINET IO (STEP 7 Safety V21)

Introduction

You use the SENDDP and RCVDP instructions for fail-safe sending and receiving of data using:

- Safety-related master-master communication
- Safety-related master-master communication for S7 Distributed Safety
- Safety-related master-I-slave communication
- Safety-related I-slave-I-slave communication
- Safety-related IO controller-IO controller communication
- Safety-related IO controller-IO controller communication for S7 Distributed Safety
- Safety-related IO controller-I-device communication
- Safety-related IO controller-I-slave communication

Description

The SENDDP instruction sends 16 data elements of data type BOOL and 2 data elements of data type INT or one data element of the data type DINT (S7-1200, S7-1500) in a fail-safe manner to another F-CPU via PROFIBUS DP/PROFINET IO. The data can be received there by the related RCVDP instruction.

Every call of this instruction must be assigned a data area in which the instruction data is stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g. RCVDP_DB_1) for these instructions. Once it is created, the new data block can be found in the "STEP 7 Safety" folder in the project tree under "Program blocks > System blocks". For more information, refer to the help on *STEP 7*.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

With the SENDDP instruction, the data to be sent (for example, outputs of other F-blocks/instructions) are available at input SD_BO_xx or SD_I_xx or SD_DI_00 as alternative.

With the RCVDP instruction, the data received are available at outputs RD_BO_xx and RD_I_xx or RD_DI_00 as alternative for additional processing by other F-blocks/instructions.

(S7-1200, S7-1500) At the DINTMODE input of the SENDDP instruction you specify if the data at the inputs SD_I_00 and SD_I_01 or the data at the input SD_DI_00 is sent.

The operating mode of the F-CPU with the SENDDP instruction is provided at output SENDMODE. If the F-CPU with the SENDDP instruction is in disabled safety mode, output SENDMODE = 1.

Communication between F-CPUs takes place in the background by means of a special safety protocol. You must define the communication relationship between a SENDDP instruction in one F-CPU and a RCVDP instruction in the other F-CPU by specifying an F-communication ID at the DP_DP_ID inputs of the SENDDP and RCVDP instructions. Associated SENDDP and RCVDP instructions are assigned the same value for DP_DP_ID.

⚠ WARNING

The value for the respective F-communication ID (input DP_DP_ID; data type: INT) can be freely selected**; however, it must be unique network-wide* and CPU-wide**** at all times for all safety-related communication connections. You must check the uniqueness in the safety summary during acceptance of the safety program.

You must supply constant values*** to the DP_DP_ID and LADDR inputs when calling the instruction. Direct write access to DP_DP_ID and LADDR in the associated instance DB is not permitted in the safety program! *(S016)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, no connection is established at the DP_DP_ID input for a F-communication ID "0".

*** S7-1200/1500: As of version V3.0 of the SENDDP and RCVDP instructions, the DP_DP_ID input can also be supplied with variable values from a global F-DB. In this case also, you have to check during acceptance of the safety program that uniqueness is ensured at all times. You need to check the algorithm for forming the variable value accordingly. If you cannot ensure

a unique F-communication ID during startup of the safety program, because it is only specified after startup of the safety program, you must make sure that the value at the DP_DP_ID input is "0" during this phase.

**** With a redundant S7-1500HF system, both F-CPUs of the redundant S7-1500HF system are to be regarded as one single F-CPU with regard to the DP_DP_ID.

NOTE

Within a safety program, you must assign a different start address (S7-300, S7-400) or HW identifier (S7-1200, S7-1500) for every call of the SENDDP and RCVDP instructions at input LADDR.

A separate instance DB must be used for each call of the SENDDP and RCVDP instructions. You must not declare and call these instructions as multi-instances.

(S7-300, S7-400) The inputs of the RCVDP and RCVS7 instructions may not have preceding logic operations (for example "AND logic operation").

The inputs of the RCVDP instruction cannot be initialized using fully qualified DB accesses with outputs of a RCVDP or RCVS7 instruction called in an upstream network.

(S7-1200, S7-1500) The RD_D_00 output must not be evaluated for DINTMODE = 0; the RD_I_xx outputs of the RCVDP instruction must not be evaluated for DINTMODE = 1.

(S7-1200, S7-1500) The outputs of the SENDDP and RCVDP instructions must not be supplied with tags from the standard user program. Exception: RET_DPRD, RET_DPWR and DIAG outputs.

Fully qualified access to DP_DP_ID and LADDR is not possible in the safety program.

You cannot use an actual parameter for an output of an RCVDP instruction, if it is already being used for an input of the same or another RCVDP or RCVS7 instruction.

The F-CPU can go to STOP if this is not observed. The cause of the diagnostics event is entered in the diagnostics buffer of the F-CPU.

NOTE

You are not permitted to insert any SENDDP/RCVDP instructions between a JMP or JMPN instruction and the associated jump destination (jump label).

You cannot insert a RET instruction prior to a SENDDP instruction.

SENDDP parameter

The following table shows the parameters of the SENDDP instruction:

Parameter	Declaration	Data type	Description
SD_BO_00	Input	BOOL	Send data BOOL 00
...			...
SD_BO_15	Input	BOOL	Send data BOOL 15
SD_I_00	Input	INT	Send data INT 00
SD_I_01	Input	INT	Send data INT 01
SD_DI_00	Input	DINT	(S7-1200, S7-1500) (hidden) Send data DINT 00
DINTMODE	Input	DINT	(S7-1200, S7-1500) (hidden) 0=SD_I_00 u. SD_I_01 are sent 1=SD_DI_00 is sent
DP_DP_ID	Input	INT	F-communication ID between SENDDP and RCVDP
TIMEOUT	Input	TIME	Monitoring time in ms for safety-related communication (see also Monitoring and response times (Page 629))
LADDR	Input	INT (S7-300, S7-400) HW_SUBMODULE (S7-1200, S7-1500)	The start address (S7-300, S7-400) or HW identifier (S7-1200, S7-1500) of the address area/transfer area: • Of DP/DP coupler for safety-related master-master communication • For safety-related master-I-slave communication • For safety-related I-slave-I-slave communication • Of PN/PN coupler for safety-related IO controller-IO controller communication • For safety-related IO controller-I-device communication • For safety-related IO controller-I-slave communication
ERROR	Output	BOOL	1=Communication error
SUBS_ON	Output	BOOL	1=RCVDP outputs fail-safe values
RET_DPRD	Output	WORD	Non-fail-safe error code RET_VAL of the DPRD_DAT instruction (for a description of error codes, refer to the help for the DPRD_DAT instruction ("Extended instructions > Distributed I/O > Other")).
RET_DPWR	Output	WORD	Non-fail-safe error code RET_VAL of the DPWR_DAT instruction (for a description of error codes, refer to the help for the DPWR_DAT instruction ("Extended instructions > Distributed I/O > Other")).
DIAG	Output	BYTE	Non-fail safe service information

RCVDP parameter:

The following table shows the parameters of the RCVDP instruction:

Parameter	Declaration	Data type	Description
ACK_REI	Input	BOOL	1=Acknowledgment for reintegration of send data following communication error
SUBBO_00	Input	BOOL	Fail-safe value for receive data BOOL 00
...			...
SUBBO_15	Input	BOOL	Fail-safe value for receive data BOOL 15
SUBI_00	Input	INT	Fail-safe value for receive data INT 00
SUBI_01	Input	INT	Fail-safe value for receive data INT 01
SUBDI_00	Input	DINT	(S7-1200, S7-1500) (hidden) Fail-safe value for receive data DINT 00
DP_DP_ID	Input	INT	F-communication ID between SENDDP and RCVDP
TIMEOUT	Input	TIME	Monitoring time in ms for safety-related communication (see also Monitoring and response times (Page 629))
LADDR	Input	INT (S7-300, S7-400) HW_SUBMODULE (S7-1200, S7-1500)	The start address (S7-300, S7-400) or HW identifier (S7-1200, S7-1500) of the address area/transfer area: • Of DP/DP coupler for safety-related master-master communication • For safety-related master-I-slave communication • For safety-related I-slave-I-slave communication • Of PN/PN coupler for safety-related IO controller-IO controller communication • For safety-related IO controller-I-device communication • For safety-related IO controller-I-slave communication
ERROR	Output	BOOL	1=Communication error
SUBS_ON	Output	BOOL	1=Fail-safe values are output
ACK_REQ	Output	BOOL	1=Acknowledgment for reintegration of send data required
SENDMODE	Output	BOOL	1=F-CPU with SENDDP instruction in disabled safety mode
RD_BO_00	Output	BOOL	Receive data BOOL 00
...			...
RD_BO_15	Output	BOOL	Receive data BOOL 15
RD_I_00	Output	INT	Receive data INT 00
RD_I_01	Output	INT	Receive data INT 01
RD_DI_00	Output	DINT	(S7-1200, S7-1500) (hidden) Receive data DINT 00
RET_DPRD	Output	WORD	Non-fail-safe error code RET_VAL of the DPRD_DAT instruction (for a description of error codes, refer to the help for the DPRD_DAT instruction ("Extended instructions > Distributed I/O > Other")).
RET_DPWR	Output	WORD	Non-fail-safe error code RET_VAL of the DPWR_DAT instruction (for a description of error codes, refer to the help for the DPWR_DAT instruction ("Extended instructions > Distributed I/O > Other")).
DIAG	Output	BYTE	Non-fail safe service information

Instruction versions

A number of versions are available for these instructions:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	x	—	—	—	
1.1	o	—	—	o	These versions have identical functions to version 1.0.
1.2	x	—	—	o	
1.4	x	—	—	x	
1.3	x	—	—	o	S7-300/400: These versions have identical functions to version 1.0. S7-1200/1500: Instead of 2 data of data type INT, one data of data type DINT can be sent/received as alternative. Otherwise functionally identical to version 1.0.
1.5	x	x	—	x	
2.0	x	x[1]	—	x[2]	
3.0	x	x[1]	—	x[2]	S7-300/400: This version has identical functions to version 2.0. S7-1200/1500: • The DP_DP_ID input can also be supplied with tags of a global F-DB. In case of DP_DP_ID = 0 no connection is established. • Supports the data status byte of the PN/PN coupler as of firmware V4.0 • Supports the simulation of the communication in S7-PLCISM operation • Use for communication with S7-1200/1500 CPUs is not recommended Otherwise functionally identical to version 2.0
3.1	x	x[1]	x	x[2]	S7-300/400: This version has identical functions to version 3.0. S7-1200/1500: • This version has identical functions to version 3.0. If the communication partner also uses a version > V3.0, a communication protocol with improved safety features becomes active. Upgrading to version V3.1 is recommended before new acceptance of an existing system. • Incompatible with S7-300/400 F-CPUs.

o This version is no longer supported.

[1] supported as of firmware V4.2

[2] supported as of firmware V2.0

When a new F-CPU is created with *STEP 7 Safety*, the latest available version for the F-CPU created is automatically preset.

Check if all communication partners of this F-CPU with SENDDP/RCVDP are S7-1200/1500 F-CPUs. In this case, use version 3.1. Otherwise, only version 3.0 or lower is possible.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Placement

You must insert the RCVDP instruction *either* at the start of the main safety block *or* (for S7-1200/1500 F-CPUs) in an F-FB/F-FC called directly at the start of the main safety block. No other instructions can be located before in the main safety block and no other instructions can be located before or afterwards in the F-FB/F-FC.

You need to insert the SENDDP instruction *either* at the end of the main safety block *or* (with S7-1200/1500 F-CPUs) in an F-FB/F-FC called directly at the end of the main safety block. No other instructions can be located afterwards in the main safety block and no other instructions can be located before or afterwards in the F-FB/F-FC.

Startup behavior

After the sending and receiving F-systems are started up, communication must be established between the connection partners for the fist time (SENDDP and RCVDP instructions). During this time, the receiver (RCVDP instruction) outputs the fail-safe values present at its inputs SUBBO_xx and SUBI_xx or alternatively SUBDI_00.

The SENDDP and RCVDP instructions signal this with 1 at output SUBS_ON. The SENDMODE output has default setting "0" and is not updated as long as output SUBS_ON = 1.

As of version V3.0 of the SENDDP and RCVDP instructions, communication is only established when DP_DP_ID <> 0.

Behavior in the event of communication errors

If a communication error occurs, for example, due to a signature error (CRC) or when monitoring time TIMEOUT expires or for S7-1200/1500 F-CPUs as of V3.0 due to a change of DP_DP_ID to 0 after establishment of communication, the outputs ERROR and SUBS_ON = 1 are set. The receiver (RCVDP instruction) then outputs the fail-safe values assigned at its inputs SUBBO_xx and SUBI_xx or alternatively SUBDI_00 inputs. Output SENDMODE is not updated while output SUBS_ON = 1.

The send data of the SENDDP instruction present at inputs SD_BO_xx and SD_I_xx, SD_DI_00 as alternative, are only output again when communication errors are no longer detected (ACK_REQ = 1) and you acknowledge (Page 190) the RCVDP instruction with a positive edge at input ACK_REI.

Communication errors also occur if the values of the DP_DP_IDs between associated SENDDP and RCVDP are temporarily different on a change of the values of variable DP_DP_IDs after communication establishment.

⚠ WARNING

A user acknowledgment (Page 190) must always take place with a signal that was generated by manual operation. Implementation with an automatically generated signal is not permitted.* *(S011)*

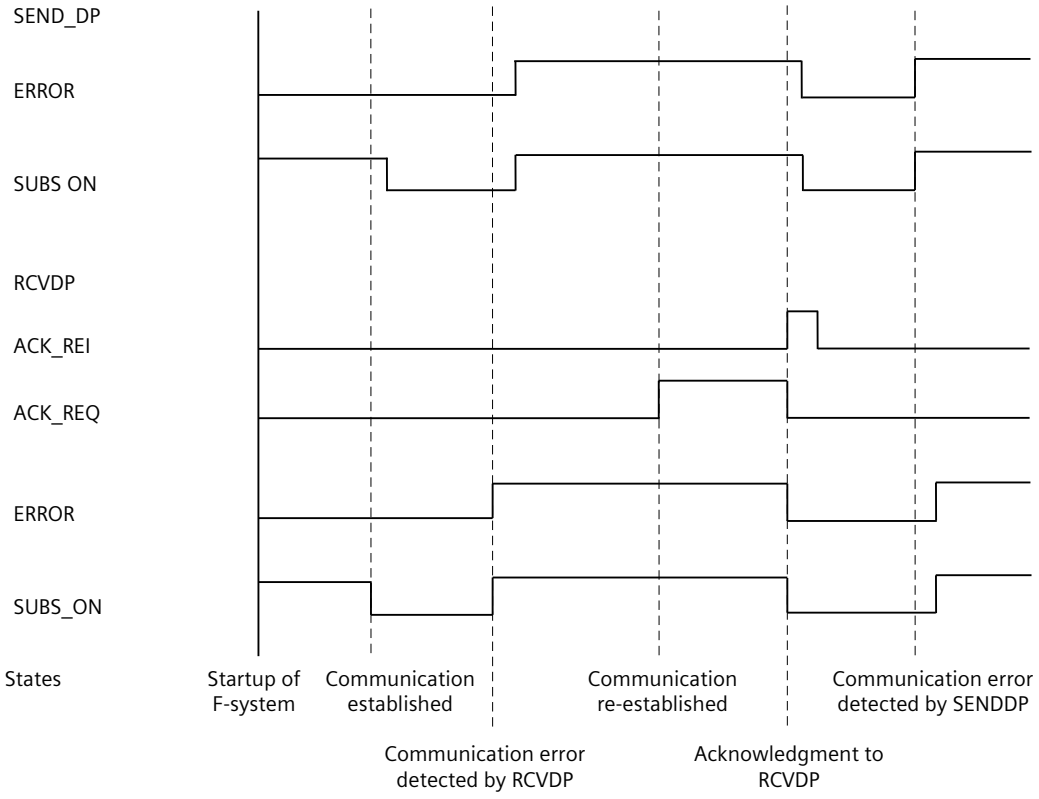
* If variable F-communication IDs are used, the communication partner of the SENDDP or RCVDP instructions can be changed during running operation. The resultant communication errors may only be acknowledged with an automatically generated signal at the ACK_REI input under the following conditions:

- The safety program reliably forms a signal "Communication partner change is in progress" with the RCVDP instruction on the basis of the process state.
- The signal "Communication partner change is in progress" is only formed if there is no communication error.
- While the signal "Communication partner change is in progress" is active, no evaluation of the received process values is carried out at the RCVDP instruction.
- Automatic acknowledgment takes place only if the "Communication partner change is in progress" signal is present.
- From a safety standpoint automatic reintegration is permitted for the relevant process.

Note that output ERROR (1=communication error) for a communication error will not be set unless communication between the connection partners (SENDDP and RCVDP instructions) has been previously established. If communication cannot be established after startup of the sending and receiving F-Systems, check the configuration of the safety-related CPU-CPU communication, the parameter assignment of the SENDDP and the RCVDP instructions, and the bus connection. You also obtain information on possible error causes by evaluating outputs DIAG, RET_DPRD and RETDP_WR.

In general, always evaluate RET_DPRD and RETDP_WR, since it is possible that only one of the two outputs will contain error information.

Timing diagrams SENDDP/RCVDP



Output DIAG

In addition, non-fail-safe information about the communication states that occurred is provided at output DIAG of the SENDDP and RCVDP instructions for service purposes.

You can read out this information by means of operator control and monitoring systems or, if applicable, you can evaluate it in your standard user program. DIAG bits 0-6 remain saved until you acknowledge the RCVDP instruction at the ACK_REI input.

Structure of DIAG of the SENDDP/RCVDP instruction

Bit no.	Assignment	Possible error causes	Remedies
Bit 0	Reserved	—	—
Bit 1	Reserved	—	—
Bit 2	Reserved	—	—
Bit 3	Invalid DP_DP_ID	The DP_DP_ID is 0.	Check the DP_DP_ID of SENDDP or RCVDP.
Bit 4	Timeout of SENDDP/RCVDP detected	The standard user program overwrites transfer areas of SENDDP and RCVDP.	Check the standard user program for write access to the transfer areas of SENDDP and RCVDP. Also take indirect accesses into account.

Bit no.	Assignment	Possible error causes	Remedies
Bit 4	Timeout of SENDDP/RCVDP detected	DP_DP_ID of SENDDP and RCVDP different.	Check the DP_DP_ID of SENDDP and RCVDP.
		For variable F-communication IDs the values are changed at the DP_DP_ID input.	When the DP_DP_ID of SENDDP and RCVD is consistent again, perform an acknowledgment at the ACK_REI input.
		Interference in bus connection to partner F-CPU.	Check the bus connection and ensure that there are no external sources of interference.
		The values at the TIMEOUT input are different for SENDDP and RCVDP.	Configure identical values at the TIMEOUT input of SENDDP and RCVDP.
		Monitoring time setting for F-CPU and partner F-CPU is too low.	Check the parameterized monitoring time TIMEOUT at SENDDP and RCVDP of both CPUs. Set a higher value if necessary. Recompile the safety program.
		Configuration of the DP/DP coupler or PN/PN coupler is invalid.	Check the configuration of the DP/DP coupler or of the PN/PN coupler.
		Data validity indicator "DIA" of the DP/DP coupler is "ON".	Set the data validity indicator "DIA" at the DIL switch of the DP/DP coupler to "OFF".
		Parameter "Data validity indicator DIA" of the PN/PN coupler is activated.	Deactivate the "Data validity display DIA" parameter in the properties for the PN/PN coupler.
		Parameter "Activate data status" of the PN/PN coupler (as of V4.0) is activated.	Deactivate the "Activate data status" parameter in the properties for the PN/PN coupler (as of V4.0) or S7-1200/1500: Use Version V3.0 of the SENDDP and RCVDP instructions.
		Internal fault of DP/DP coupler or PN/PN coupler	Replace the DP/DP coupler or PN/PN coupler
		CP in STOP mode, or internal fault in CP	Switch the CP to RUN. Check the diagnostics buffer of the CP. If necessary, replace the CP.
		F-CPU/partner F-CPU in STOP mode, or internal fault in F-CPU/partner F-CPU	Switch the F-CPUs to RUN. Check the diagnostics buffer of the F-CPUs. If necessary, replace the F-CPUs.
Bit 5	Sequence number error of SENDDP/RCVDP detected	See description for bit 4	See description for bit 4
Bit 6	CRC-error of SENDDP/RCVDP detected	See description for bit 4	See description for bit 4
Bit 7	Communication protocol with improved safety features is not active.	Communication partner uses an instruction version < V3.1	Upgrade the communication partner to an instruction version > V3.0 at the next opportunity.

See also

Communication with S7 Distributed Safety via PN/PN coupler (IO controller-IO controller communication) (Page 243)

Communication with S7 Distributed Safety via DP/DP coupler (master-master communication) (Page 244)

Communication via SENDDP/RCVDP or SENDS7/RCVS7(S7-300, S7-400) instructions: (Page 202)

Safety-related IO controller-IO controller communication (Page 205)

Safety-related master-master communication (Page 212)

Safety-related IO controller-I-device communication (Page 220)

Safety-related master-I-slave communication (Page 226)

Safety-related IO controller-I-slave communication (Page 236)

13.17.2 S7 communication

13.17.2.1 SENDS7 and RCVS7: Communication via S7 connections (STEP 7 Safety Advanced V21) (S7-300, S7-400)

Introduction

You use the SENDS7 and RCVS7 instructions for fail-safe sending and receiving of data using S7 connections.

NOTE

In *STEP 7 Safety Advanced*, S7 connections are generally permitted over Industrial Ethernet only.

Safety-related communication via S7 connections is possible from and to F-CPUs with PROFINET interface or S7-400 F-CPUs with PROFINET-capable CPs. See also Safety-related communication via S7 connections (Page 237).

Description

The SENDS7 instruction sends the send data contained in an F-communication DB to the F-communication DB of the associated RCVS7 instruction of another F-CPU in a fail-safe manner using an S7 connection.

Every call of this instruction must be assigned a data area in which the instruction data is stored. The "Call options" dialog is automatically opened when the instruction is inserted in the program for this reason. There you can create a data block (single instance) (e.g., SENDS7_DB_1) or a multi-instance (e.g., SENDS7_Instance_1) for this instruction. Once it is created, you can find the new data block in the project tree in the "STEP 7 Safety" folder under "Program blocks > System blocks" or the multi-instance as a local tag in the "Static" section of the block interface. For more information, refer to the help on *STEP 7*.

Enable input "EN" and enable output "ENO" cannot be connected. The instruction is therefore always executed (regardless of the signal state at enable input "EN").

Information on the F-communication DB is contained in "Programming safety-related communication via S7 connections (Page 238)".

An F-communication DB is an F-DB for safety-related CPU-CPU communication with special properties. You must specify the numbers of the F-communication DBs at inputs SEND_DB and RCV_DB of instructions SENDS7 and RCVS7.

The operating mode of the F-CPU with the SENDS7 instruction is provided at output SENDMODE of the RCVS7 instruction. If the F-CPU with the SENDS7 instruction is in disabled safety mode, then output SENDMODE = 1.

To reduce the bus load, you can temporarily shut down communication between the F-CPUs at input EN_SEND of the SENDS7 instruction. To do so, supply input EN_SEND with "0" (default = "1"). In this case, send data is no longer sent to the F-communication DB of the associated RCVS7 instruction, and the receiver provides fail-safe values for this period of time (initial values in its F-communication DB). If communication was already established between the partners, a communication error is detected.

You must specify the local ID - from the perspective of the F-CPU - of the S7 connection (from the connection table in the network view) at input ID of the SENDS7 instruction (see also Configuring (Page 35)).

Communication between F-CPUs takes place in the background by means of a special safety protocol. You must define a communication relationship between an SENDS7 instruction in one F-CPU and a communication relationship between an RCVS7 instruction and the other F-CPU by assigning an odd number at input R_ID (of the SENDS7 and RCVS7 instructions). Associated SENDS7 and RCVS7 instructions receive the same value for R_ID.

⚠ WARNING

The value of the respective F-communication ID (input R_ID; data type: DWORD) can be freely selected; however, it must be odd and unique for all safety-related communication connections network-wide* and CPU-wide. The value R_ID + 1 is internally assigned and must not be used.

You must supply inputs ID and R_ID with constant values when calling the instruction. Direct read or write access to ID and R_ID in the associated instance DB is not permitted in the safety program. *(S020)*

* A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

NOTE

A separate instance DB must be used for each call of the SENDS7 and RCVS7 instructions within a safety program. You must not declare and call these instructions as multi-instances.

The inputs of the RCVS7 instruction cannot be initialized with outputs (using fully qualified DB accesses) of a RCVS7 or RCVDP instruction called in an upstream network.

You cannot use an actual parameter for an output of an RCVS7 instruction, if it is already being used for an input of the same or another RCVS7 or RCVDP instruction.

The F-CPU can go to STOP if this is not observed. A diagnostics event is entered in the diagnostics buffer of the F-CPU.

NOTE

You must not program any SENDS7/RCVS7 instruction between a JMP or JMPN instruction and the associated destination network of the JMP or JMPN instruction.

You cannot program a RET instruction prior to a SENDS7 instruction.

SENDS7 parameter

The following table shows the parameters of the SENDS7 instruction:

Parameter	Declaration	Data type	Description
SEND_DB	Input	BLOCK_DB	Number of F-communication DB
TIMEOUT	Input	TIME	Monitoring time in ms for safety-related communication (see also Monitoring and response times (Page 629))
EN_SEND	Input	BOOL	1= Send enable
ID	Input	WORD	Local ID of the S7 connection
R_ID	Input	DWORD	Network-wide unique value for a F-communication ID between a SENDS7 instruction and a RCVS7 instruction
ERROR	Output	BOOL	1=Communication error
SUBS_ON	Output	BOOL	1=Receiving block outputs fail-safe values
STAT_RCV	Output	WORD	Non-fail-safe status parameter STATUS of the URCV instruction (You can find a description of error codes in the help for the URCV instruction ("Communication > S7 Communication"))
STAT_SND	Output	WORD	Non-fail-safe status parameter STATUS of the USEND instruction (You can find a description of error codes in the help for the USEND instruction ("Communication > S7 Communication"))
DIAG	Output	BYTE	Non-fail-safe service information

RCVS7 parameter

The following table shows the parameters of the RCVS7 instruction.

Parameter	Declaration	Data type	Description
ACK_REI	Input	BOOL	Acknowledgment for reintegration of send data after communication error
RCV_DB	Input	BLOCK_DB	Number of F-communication DB
TIMEOUT	Input	TIME	Monitoring time in ms for safety-related communication (see also Monitoring and response times (Page 629))
ID	Input	WORD	Local ID of the S7 connection
R_ID	Input	DWORD	Network-wide unique value for a F-communication ID between a SENDS7 instruction and a RCVS7 instruction
ERROR	Output	BOOL	1=Communication error
SUBS_ON	Output	BOOL	1=Fail-safe values are output
ACK_REQ	Output	BOOL	1=Acknowledgment for reintegration of send data required

Parameter	Declaration	Data type	Description
SENDMODE	Output	BOOL	1=F-CPU with the SENDS7 instruction in disabled safety mode
STAT_RCV	Output	WORD	Non-fail-safe status parameter STATUS of the URCV instruction (You can find a description of error codes in the help for the URCV instruction ("Communication > S7 Communication"))
STAT_SND	Output	WORD	Non-fail-safe status parameter STATUS of the USEND instruction (You can find a description of error codes in the help for the USEND instruction ("Communication > S7 Communication"))
DIAG	Output	BYTE	Non-fail-safe service information

Instruction versions

A number of versions are available for these instructions:

Version	S7-300/400	S7-1500	Function
1.0	x	—	
1.1	x	—	This version has identical functions to version 1.0. It also supports later versions of internally called instructions.
1.2	x	—	This version has identical functions to version 1.0/1.1. It also supports later versions of internally called instructions.

When a new F-CPU is created with *STEP 7 Safety Advanced*, the latest available version for the F-CPU created is automatically preset.

For more information on the use of instruction versions, refer to the help on *STEP 7* under "Using instruction versions".

Placement

You must insert the RCVS7 instruction at the start of the main safety block. No other instructions may be located before it in the main safety block.

You must insert the SENDS7 instruction at the end of the main safety block. No other instructions may be located after it in the main safety block.

Startup behavior

After the sending and receiving F-systems are started up, communication must be established between the connection partners for the fist time (SENDS7 and RCVS7 instructions). The receiver (RCVS7 instruction) provides fail-safe values for this time period (initial values in its F-communication DB).

The SENDS7 and RCVS7 instructions signal this with 1 at output SUBS_ON. The SENDMODE output (RCVS7 instruction) has default setting "0" and is not updated as long as output SUBS_ON = 1.

Behavior in the event of communication errors

If a communication error occurs, for example, due to a signature error (CRC) or when monitoring time TIMEOUT expires, outputs ERROR and SUBS_ON = 1 are set. The receiver (RCVS7 instruction) then provides fail-safe values (initial values in its F-communication DB). Output SENDMODE is not updated while output SUBS_ON = 1.

The send data present in the F-communication DB (SENDS7 instruction) are not output before the communication error is no longer detected (ACK_REQ = 1) and you acknowledge (Page 190) with a positive edge at input ACK_REI of the RCVS7 instruction.

⚠ WARNING

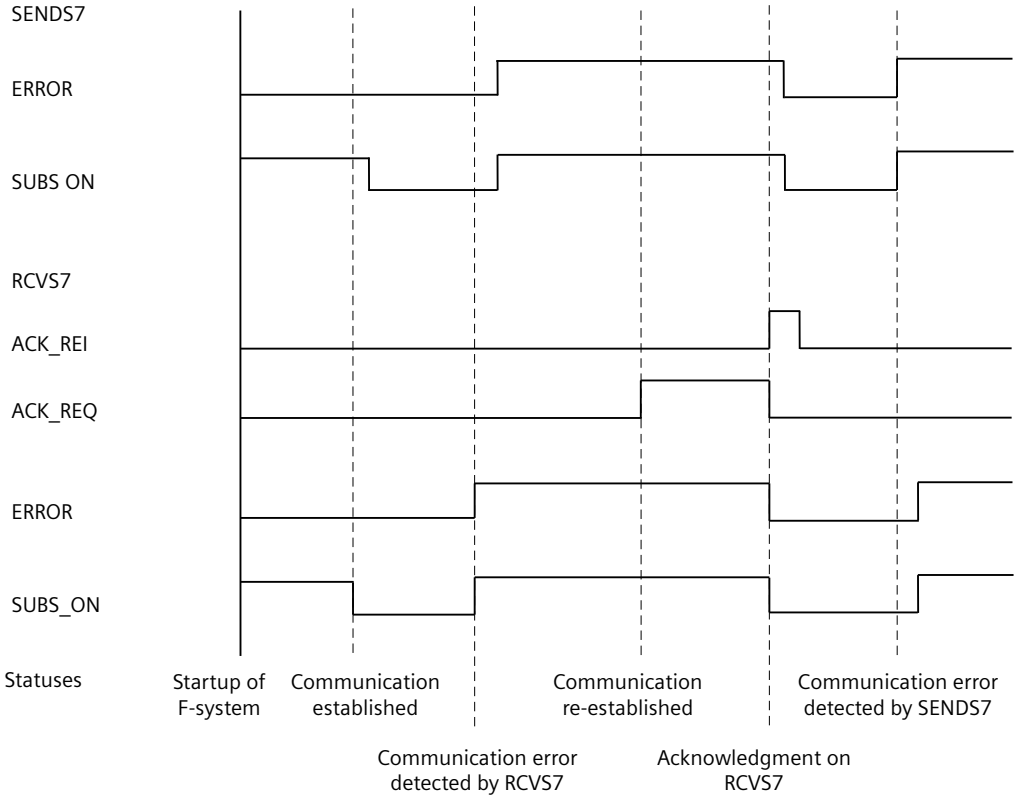
A user acknowledgment (Page 190) must always take place with a signal that was generated by manual operation. Implementation with an automatically generated signal is not permitted. *(S011)*

Note that output ERROR (1=communication error) will be set for the first time on a communication error if communication has already been established between the connection partners (SENDS7 and RCVS7 instructions). If communication cannot be established after startup of the sending and receiving F-Systems, check the configuration of the safety-related CPU-CPU communication, the parameter assignment of the SENDS7 and the RCVS7 instructions, and the bus connection. You can also receive information on possible error causes by evaluating the STAT_RCV and STAT_SND outputs.

In general, always evaluate STAT_RCV and STAT_SND, since it is possible that only one of the two outputs will contain error information.

If one of the DIAG bits is set at output DIAG, also check whether the length and structure of the associated F-communication DB on both the sending and receiving ends match.

Timing diagrams SENDS7 and RCVS7



Output DIAG

Non-fail-safe information on the type of communication errors that have occurred is made available at output DIAG for service purposes. You can read out this information by means of operator control and monitoring systems or, if applicable, you can evaluate it in your standard user program. DIAG bits are saved until you acknowledge them at input ACK_REI of the associated RCVS7 instruction.

Structure of DIAG

Bit no.	Assignment of SENDS7 and RCVS7	Possible error causes	Remedies
Bit 0	Reserved	—	—
Bit 1	Reserved	—	—
Bit 2	Reserved	—	—
Bit 3	Reserved	—	—
Bit 4	Timeout detected by SENDS7 and RCVS7	Fault in bus connection to partner F-CPU	Check bus connection and ensure that no external fault sources are present.

Bit no.	Assignment of SENDS7 and RCVS7	Possible error causes	Remedies
Bit 4	Timeout detected by SENDS7 and RCVS7	Monitoring time setting for F-CPU and partner F-CPU is too low	Check assigned monitoring time TIMEOUT for SENDS7 and RCVS7 of both F-CPUs. If possible, set a higher value. Recompile safety program
		CPs in STOP mode, or internal fault in CPs	• Switch CPs to RUN mode • Check diagnostic buffer of CPs • Replace CPs, if necessary
		F-CPU/partner F-CPU in STOP mode, or internal fault in F-CPU/partner F-CPU	• Switch F-CPUs to RUN mode • Check diagnostic buffer of F-CPUs • Replace F-CPUs, if necessary
		Communication was shut down with EN_SEND = 0.	Enable communication again at the associated SENDS7 with EN_SEND = 1
		S7 connection has changed, the IP address of the CP has changed, for example	Recompile the safety programs and download them to the F-CPUs
Bit 5	Sequence number error detected by SENDS7 and RCVS7	See description for bit 4	See description for bit 4
Bit 6	CRC-error detected by SENDS7 and RCVS7	See description for bit 4	See description for bit 4
Bit 7	RCVS7: Communication cannot be established	Configuration of the safety-related CPU-CPU communication is incorrect, parameter assignment of the SENDS7 and RCVS7 instructions is incorrect See also description for Bit 4	Check configuration of the safety-related CPU-CPU communication, the parameter assignment of the SENDS7 and the RCVS7 instructions is incorrect See also description for Bit 4
	SENDS7: Reserved	—	—

Monitoring and response times

A

Introduction

In the following, you will learn:

- Which F-specific monitoring times you must configure
- Which rules must be followed when specifying monitoring times
- Where you enter the F-specific monitoring times
- Which rules must be followed with regard to the maximum response time of a safety function

Support for calculations

An Excel file is available on the Internet (https://support.industry.siemens.com/cs/ww/en/view/109783831) to support you in the approximative calculation of the runtimes of the F-runtime groups, the F-specific minimum monitoring times and the maximum response times for your F-system.

Additional information

The monitoring and response times for the standard part are calculated in SIMATIC Safety in exactly the same way as for standard S7-300, S7-400, S7-1200 and S7-1500 automation systems and are not addressed here. For a description of this calculation, refer to the *hardware manuals for the CPUs*. For redundant S7-1500HF systems, also observe the system manual "Redundant System S7-1500R/H (https://support.industry.siemens.com/cs/ww/en/view/109754833)".

A.1 Configuring monitoring times

Monitoring times to be configured

You must configure the following monitoring times:

Monitoring...	Setting...	Parameter	See
F-cycle time or cycle time warning limit of the F-runtime groups that contain the Safety program	in *Safety Administration Editor*: • Dialog for definition of an F-runtime group	Maximum cycle time of the F-runtime group	• Procedure for defining an F-runtime group (S7-300, S7-400) (Page 134) • Procedure for defining an F-runtime group (S7-1200, S7-1500) (Page 138)
of the safety-related communication between F-CPU and F-I/O via PROFIsafe (PROFIsafe monitoring time)	In the *hardware and network editor*: • Centrally when configuring the F-CPU; properties of the F-CPU; or • when configuring the F-I/O; properties of the F-I/O	F-monitoring time F_WD_TIME	• Configuring an F-CPU (Page 41) • Configuring F-I/O (Page 53) • Peculiarities when configuring fail-safe GSD based DP slaves and fail-safe GSD based I/O devices (Page 39)
Communication via SENDDP/RCVDP or SENDS7/RCVS7 instructions:	At the TIMEOUT input of the instructions: • SENDDP; RCVDP; SENDS7; RCVS7	TIMEOUT	• Communication (Page 612)
(S7-1500) Communication via OPC UA Safety	After adding a SafetyProvider to a SafetyProvider list of a client interface (Page 320) in the properties of the SafetyConsumer in the editor for client interfaces (Page 324)	SafetyConsumer-Timeout	• Safety Parameter Interface (SPI) of the SafetyConsumer (Page 295)
(S7-1200, S7-1500) Communication with Flexible F-Link	in Safety Administration Editor: • "Flexible F-Link" area	F-monitoring time of F-communication	• "Flexible F-Link" area of the Safety Administration Editor (S7-1200, S7-1500) (Page 97) • F-runtime group communication (S7-1200, S7-1500) (Page 146) • Communication with Flexible F-Link (S7-1200, S7-1500) (Page 342)

(S7-300, S7-400) You do not have to configure the monitoring time for safety-related communication between F-runtime groups.

Rules for configuring monitoring times

When configuring monitoring times, you must take into account the availability as well as the safety of the F-system as follows:

- Availability: Sufficiently large monitoring times must be selected so that time monitoring is not exceeded in an error-free scenario.
- Safety: Sufficiently small monitoring times must be selected so that the process safety time of the process is not exceeded.

⚠ WARNING

It can only be ensured (from a fail-safe standpoint) that a signal state to be transferred will be acquired and transferred to the receiver if the signal level is pending for at least as long as the assigned monitoring time. *(S018)*

General procedure for configuring monitoring times

Use the following procedure for configuring monitoring times:

1. Configure the standard system.

   Refer to the applicable *hardware manuals* and *Help on STEP 7* for the necessary information.

2. Configure the specific monitoring times of the F-System with respect to availability. You use the Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) to calculate the approximate minimum monitoring time.

   NOTE

   To avoid a triggering of the time monitoring in the special operating and system states of the redundant S7-1500HF system in case of error-free operation, higher minimum monitoring times are required for a redundant S7-1500HF system with regard to availability. This is taken into account accordingly in the Excel file for the response time calculation.

3. Use the Excel file for response time calculation to calculate the maximum response time and then verify that the process safety time of the process is not exceeded. If necessary, you must reduce the specific monitoring times of the F-System.

See also

Redundant System S7-1500R/H
(https://support.industry.siemens.com/cs/ww/en/view/109754833)

A.1.1 Minimum monitoring time for the F-runtime group cycle time

Parameter "Maximum cycle time of the F-runtime group"

You configure the monitoring time for the F-runtime group cycle time in the *Safety Administration Editor* in the work area for definition of the F-runtime group (Page 133).

To prevent F-runtime group cycle time monitoring from being triggered when there are no pending faults and causing the F-CPU to go to STOP, you must set the maximum cycle time of the F-runtime group high enough.

Use the Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) to determine the minimum monitoring time for the F-runtime group cycle time. Note also the comments in the Excel file.

For S7-1200/1500 F-CPUs, you can also use the tags TCYC_CURR (Page 154) and TCYC_LONG (Page 154) to determine the current and longest cycle time of the F-runtime group that occurred since the last STOP-RUN transition for dimensioning.

A.1.2 Minimum monitoring time of the F-I/O access

Parameter "F-monitoring time"

You have two options for configuring the monitoring time of safety-related communication between the F-CPU and F-I/O:

- Centrally in the *hardware and network editor* during parameter assignment of the F-CPU (Page 41); in the properties of the F-CPU, or
- During parameter assignment of the F-I/O (Page 53) in the *hardware and network editor*; in the properties of the F-I/O

NOTE

Note that in a redundant S7-1500HF system and use of an IO device that does not support system redundancy S2, there is an interruption of PROFINET communication with this device when the primary CPU fails and thus a communication error in the F-I/Os of this IO device, which requires reintegration of the F-I/Os (see section "After communication errors (Page 175)").

For IO devices with system redundancy S2, the redundant S7-1500HF system switches to the backup CPU without interruption if the primary CPU fails. During the primary backup switchover, the fail-safe outputs remain active.

"F-monitoring time" = PROFIsafe monitoring time T_{PSTO}

The specified PROFIsafe monitoring time T_{PSTO} must be high enough to prevent tripping the F-cycle time monitoring when no faults are present.

Use the Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) available for SIMATIC Safety to calculate the minimum monitoring time for safety-related communication between the F-CPU and F-I/O.

Note also the comments in the Excel file.

Check to determine whether configured PROFIsafe monitoring time is too short

NOTE

During commissioning of the F-system, you can perform a check while safety mode is active to determine whether the configured PROFIsafe monitoring time is too short.

This check of the PROFIsafe monitoring time is useful if you want to ensure that the configured monitoring time exceeds the minimum monitoring time by a sufficient amount. In this way, you can avoid the possible occurrence of sporadic monitoring time errors.

Procedure:

1. Insert an F-I/O (one that will not be needed later for system operation).
2. Assign a shorter monitoring time for this F-I/O than for the F-I/O of the system.
3. If the additional F-I/O fails and the "Monitoring time for safety message frame exceeded" diagnostic is signaled, you have fallen below the minimum possible PROFIsafe monitoring time.
4. Increase the monitoring time for the added F-I/O just to the point where it no longer fails. This monitoring time corresponds approximately to the minimum possible monitoring time.

Conditions:

The F-I/O to be inserted additionally and the F-I/O whose PROFIsafe monitoring time is to be checked must have the following properties in common:

- They must be inserted in the same rack
- They must be nodes in the same subnet

Tip:

It may be useful to leave the additional F-I/O in place for systems that will be modified or expanded during operation after commissioning. This F-I/O will then provide an early warning in the event of changes in the time behavior, enabling you to avoid a process shutdown triggered by the F-I/O in the process.

In a redundant S7-1500HF system, the check as to whether the configured PROFIsafe monitoring time is too short should be carried out in all operating and system states, because the response times there also depend on the respective operating and system state.

A.1.3 Minimum monitoring time for communication via the SENDDP/RCVDP or SENDS7/RCVS7 instructions

TIMEOUT Input at SENDDP and RCVDP or SENDS7 and RCVS7

The time monitoring is performed in the SENDDP and RCVDP (Page 612) or SENDS7 and RCVS7 (Page 622) instructions of the communication partners. You must assign the time monitoring with identical monitoring time for both instructions at the TIMEOUT input.

You must specify a monitoring time TIMEOUT that is large enough so that monitoring is not initiated when no faults are present.

Use the Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) that is available for SIMATIC Safety to determine the minimum value for TIMEOUT.

Note also the comments in the Excel file.

A.1.4 Minimum monitoring time of communication with OPC UA Safety

"SafetyConsumerTimeout" SAPI parameter

For communication with OPC UA Safety, specify the F-monitoring time for this safety-related communication after adding a SafetyProvider to a SafetyProvider list of a client interface (Page 326) in the properties of the SafetyConsumer in the editor for client interfaces (Page 320).

You must specify a monitoring time that is high enough so that SafetyConsumerTimeout is not triggered when no error has occurred.

Use the Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) that is available for SIMATIC Safety to determine the minimum value for SafetyConsumerTimeout.

Note also the comments in the Excel file.

A.1.5 Minimum monitoring time for communication with Flexible F-Link

"F-monitoring time" parameter in the settings for Flexible F-Link

For communication with Flexible F-Link, you specify the F-monitoring time for the safety-related communication when establishing safety-related communication (Page 97).

You must specify a monitoring time that is large enough so that F-monitoring is not initiated when no faults are present.

Use the Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) that is available for SIMATIC Safety to determine the minimum value for the F-monitoring time.

Note also the comments in the Excel file.

A.1.6 Monitoring Time for Safety-Related Communication between F-Runtime Groups

Monitoring time for safety-related communication between F-runtime groups (S7-300, S7-400)

The monitoring time for safety-related communication between F-runtime groups with Flexible F-Link is determined automatically from the values of the "Maximum cycle time of F-runtime group" (work area for Definition of the F-runtime group (Page 133) in the *Safety Administration Editor*).

Monitoring time = (maximum cycle time of the 1st F-runtime group) + (maximum cycle time of the 2nd F-runtime group)

Monitoring time for safety-related communication between F-runtime groups (S7-1200, S7-1500)

You can calculate the monitoring time for safety-related communication between F-runtime groups from the values for the "Maximum cycle time of the F-runtime group" (area for Definition of the F-runtime group (Page 133) in the *Safety Administration Editor*), if you place the default user program for the F-runtime group communication into pre-/postprocessing (Page 88).

Monitoring time = (Maximum cycle time of the 1st F-runtime group) + (Maximum cycle time of the 2nd F-runtime group).

A.2 Response Times of Safety Functions

Definition of response time

The response time is the time from detection of an input signal until the linked output signal changes.

Fluctuation range

The actual response time lies between a minimum response time and maximum response time. You must always take the maximum response time into account in your system configuration.

Also note the comments in the Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831).

NOTE

In a redundant S7-1500HF system, the current operating and system state influences the actual response time.

Rules for maximum response time of a safety function

The maximum response time of a safety function must be shorter than the process safety time of the process.

Definition for process safety time of a process

The process safety time of a process is the time interval between the occurrence of an error, within which the process can be left on its own without causing injury to operating personnel or damage to the environment, and the point in time the response is completed.

The controlling F-system can perform any control during the process safety time, this includes incorrectly or not at all. The process safety time of a process depends on the process type and must be determined on a case-by-case basis.

Procedure for response time calculation

The Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) is available for calculating the maximum response time of a safety function.

Use the Excel file to calculate the approximate maximum response time of the safety function and then check that the process safety time of the process is not exceeded.

If necessary, you must reduce the specific monitoring times of the F-system (see Minimum monitoring time of the F-I/O access (Page 632)).

⚠ WARNING

The response time of your safety function depends, among other things, on the cycle time of the F-OB, the runtime of the F-runtime group and, when distributed F-I/O is used, the parameter assignment of PROFINET/PROFIBUS.

Therefore, the configuration/parameter assignment of the standard system influences the response time of your safety function.

Examples:

- Increasing the priority of a standard OB compared to an F-OB can extend the cycle time of the F-OB or the runtime of the F-runtime group due to the higher-priority processing of the standard OB. Note that during the creation of technology objects, OBs with very high priority may be created automatically.
- The change of the send clock cycle of PROFINET changes the cycle time of an F-OB with event class "Synchronous cycle".

Note that the configuration / parameter assignment of the standard system is not subject to access protection for the safety-related project data and does not lead to a modification of the collective F-signature.

If you do not take organizational measures to prevent changes in the configuration / parameter assignment of the standard system with influence on the response time, you must always use the monitoring times for the calculation of the maximum response time of a safety function (see Configuring monitoring times (Page 630)).

The monitoring times are protected against change with the access protection of the safety-related project data and are captured by the Collective F-Signature as well as the collective F-SW-signature.

When calculating with the Excel file for response time calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) you need to consider the value that is specified for "Any standard system runtimes" as value for the maximum response time. *(S085)*

⚠ WARNING

You may only use the Excel file for response time calculation or for timeout calculation when using Flexible F-Link communication, if you have observed the following rules with regard to the standard instructions for consistent transfer of data:

CPU-CPU communication (Page 342)

You must call the standard instruction for consistent sending of data and acknowledgment in the post-processing of the F-runtime group (Page 88). With the standard instructions for receiving data and acknowledgments consistently, you must differentiate whether or not the standard communication connection is deterministic. For deterministic connections (such as DPRD_DAT / DPWR_DAT), you need to call the standard instruction in the pre-processing of the F-runtime group (Page 88). If the connection is non-deterministic (e.g. S7 connection, TCP connections), you must call the standard instruction in a cyclic interrupt OB. This cyclic interrupt OB must be called at shorter intervals than the F-runtime group. A ratio of 1:5 is recommended for this.

F-runtime group communication (Page 146)

You have to call up the standard instruction UMOVE_BLK for transferring the data to be sent in the postprocessing of the sending F-runtime group. You have to call the standard instruction UMOVE_BLK for transferring the acknowledgment to be sent in the postprocessing of the receiving F-runtime group. *(S089)*

⚠ WARNING

You may only use the Excel file for response time calculation or for timeout calculation (https://support.industry.siemens.com/cs/ww/en/view/109783831) when using OPC UA Safety communication, if you have observed the following rules with regard to the standard instruction OPC_UA_MethodCall:

You must call up the standard instruction OPC_UA_METHODCALL to receive the data in the pre-processing of the F-runtime group (Page 88) and to send the request in the post-processing of the F-runtime group (Page 88). The REQ input of the instruction must be supplied with a positive signal edge during preprocessing. You can use the F-OPC-UA-Consumer-Com-FB provided by the system for this. *(S112)*

Checklist

B

Life cycle of fail-safe automation systems

The following table contains a checklist summarizing all activities in the life cycle of a fail-safe SIMATIC Safety system, including requirements and rules that must be observed for each activity.

Checklist

Legend:

- Stand-alone section references refer to this documentation.
- *"F-SMs manual"* refers to the Automation System S7-300, ET 200M Distributed I/O System, Fail-Safe Signal Modules manual (https://support.industry.siemens.com/cs/ww/en/view/19026151)
- *"F-Modules manual"* refers to the ET 200S Distributed I/O System, Fail-Safe Modules manual (https://support.industry.siemens.com/cs/ww/en/view/27235629)
- *"ET 200eco manual"* refers to the ET 200eco Distributed I/O Station, Fail-Safe I/O Module manual (https://support.industry.siemens.com/cs/ww/en/view19033850)
- *"ET 200eco PN manual"* refers to the ET 200eco PN F-DI 8 x 24 VDC, 4xM12 / F-DQ 3 x 24 VDC/2.0A PM, 3xM12 equipment manual (https://support.industry.siemens.com/cs/ww/en/view/109765611)
- *"ET 200pro manual"* refers to the ET 200pro Distributed I/O System, Fail-Safe I/O Module manual (https://support.industry.siemens.com/cs/ww/en/view22098524)
- *"ET 200iSP manual"* refers to the ET 200iSP Distributed I/O Device, Fail-Safe Modules manual (https://support.industry.siemens.com/cs/ww/en/view/47357221)
- *"ET 200SP manual"* refers to the ET 200SP system manual (https://support.industry.siemens.com/cs/ww/en/view/58649293)
- *"ET 200SP HA manual"* refers to the ET 200SP HA system manual (https://support.industry.siemens.com/cs/ww/en/view/109817158)
- *"ET 200AL manual"* refers to the ET 200AL system manual (https://support.industry.siemens.com/cs/ww/en/view/89254965)
- *"ET 200MP manual"* refers to the S7-1500/ET 200MP Distributed I/O system manual (https://support.industry.siemens.com/cs/ww/en/view/59191792)
- *"S7-1500R/H manual"* refers to the S7-1500R/H system manual (https://support.industry.siemens.com/cs/ww/en/view/109754833)
- *"SIMATIC Drive Controller manual"* refers to the SIMATIC Drive Controller system manual (https://support.industry.siemens.com/cs/ww/en/view/109766665)
- *"ET 200SP Modules manual"* refers to the equipment manuals for F-Modules of the ET 200SP Distributed I/O System (https://support.industry.siemens.com/cs/ww/en/ps/14059/man)
- *"ET 200SP HA Modules manual"* refers to the equipment manuals for F-Modules of the ET 200SP HA Distributed I/O System (https://support.industry.siemens.com/cs/ww/en/ps/28940/man)

- *"ET 200MP Modules manual"* refers to the equipment manuals for F-Modules of the ET 200MP Distributed I/O System (https://support.industry.siemens.com/cs/ww/en/ps/29727/man)
- *"ET 200AL Module manual"* refers to the equipment manual for the F-Module of the ET 200AL Distributed I/O System (https://support.industry.siemens.com/cs/ww/en/view/109798489)

Phase	Note the following	Reference	Check
Planning			
Requirement: "Safety requirements specification" available for the intended application	Process-dependent	—	
Specification of system architecture	Process-dependent	—	
Assignment of functions and subfunctions to system components	Process-dependent	under Product Overview (Page 23)	
Selection of sensors and actuators	Requirements for actuators	*F-SMs Manual*, section 6.5; *F-Modules Manual*, section 4.5; *ET 200eco Manual*, section 5.5 *ET 200eco PN Manual*, section 5.2; *ET 200pro Manual*, section 4.4 *ET 200iSP Manual*, section 4.5 *ET 200SP Manual*, section 6.2.2 *ET 200SP HA Manual*, sec. 9 *ET 200MP Manual*, section 6.2.2	
Specification of required safety properties for individual components	IEC 61508:2010	—	
Configuration			
Installing license	Requirement for installation	under Installing/uninstalling the STEP 7 Safety Advanced V21 license (Page 30)	
Selection of S7 components	Descriptions of configuration	under Product Overview (Page 23); *F-SMs Manual*, section 3; *F-Modules Manual*, section 3; *ET 200eco Manual*, section 3 *ET 200eco PN Manual*, section 4; *ET 200pro Manual*, section 2 *ET 200iSP Manual*, section 3 *ET 200SP Manual*, section 4 *ET 200SP HA Manual*, sec. 4 *ET 200AL Manual*, section 3 *ET 200MP Manual*, section 4 *S7-1500R/H Manual*, section 4 *HB SIMATIC Drive Controller*, section 4	
Configuration of hardware	• Description of F-systems • Verification of utilized hardware components based on Annex 1 of Report in the Certificate	under Configuring (Page 35); Annex 1 of Report on the Certificate	

Phase	Note the following	Reference	Check
Configuration of F-CPU	• Protection level, "Write protection for F-blocks" (S7-300, S7-400) • Protection level, at least "Full access" (S7-1200, S7-1500) • Password • F-capability enabled • Definition/setting of F-specific parameters • Cycle time for the F-runtime group in which the safety program is to be executed, defined in accordance with the requirements and safety regulations - same as with standard system	under Configuring an F-CPU (Page 41); *Standard-S7-300 Manual;* *Standard-S7-400 Manual;* *Standard-S7-1200 Manual;* *Standard-S7-1500 Manual;* under Monitoring and response times (Page 629)	
Configuration of F-I/O	• Settings for safety mode • Setting of passivation type • Configuring monitoring times • Defining sensor evaluation • Defining diagnostic behavior • Special F-parameters • Assigning names • Unique PROFIsafe address	under Configuring F-I/O (Page 53) orPeculiarities when configuring fail-safe GSD based DP slaves and fail-safe GSD based I/O devices (Page 39) under Monitoring and response times (Page 629); *F-SMs Manual*, sections 3, 9, 10; *F-Modules Manual*, sections 2.4, 7; *ET 200eco Manual*, sections 3, 8; *ET 200eco PN Manual*, section 6; *ET 200pro Manual,* Sections 2.4, 8; *ET 200iSP Manual,* Sections 2.4, 7, 8 *ET 200SP Modules Manual*, section 4 *ET 200SP HA Modules manual,* sec. 9 *ET 200MP Modules Manual*, section 4 *ET 200AL Module Manual*, section 4	
Programming			
Defining program design and structure	• Follow warnings and notes on programming	under Overview of Programming (Page 111), Program structure of the safety program (S7-300, S7-400) (Page 111), Program structure of the safety program (S7-1200, S7-1500) (Page 113); Programming startup protection (Page 156);	
Creating the F-runtime groups	• Assignment of F-FB or F-FC as main safety block to the calling block (S7-300, S7-400) or F-OB (S7-1200, S7-1500) • Setting maximum cycle time for the F-runtime group in accordance with requirements (dependent on process and safety regulations) • Creating DB for F-runtime group communication • (S7-300, S7-400) Call of main safety blocks directly in OBs (e.g. OB 35), FBs, or FCs • (S7-1200, S7-1500) Call of the main safety block from the F-OB	under Defining F-Runtime Groups (Page 133) under Monitoring and response times (Page 629)	

Phase	Note the following	Reference	Check
Creating/inserting the F-blocks	• Generating, editing, and saving F-FBs, F-FCs, and F-DBs in accordance with the requirements of the program structure • Description: – F-I/O access – Passivation and reintegration of F-I/O – Inserting F-blocks from global libraries – Safety-related CPU-CPU communication – Communication with the standard user program	under Creating F-blocks (Page 150) under Addressing F-I/O (Page 157) under Implementation of user acknowledgment (Page 190) under Reuse of F-blocks (Page 152) under Communication via SENDDP/RCVDP or SENDS7/RCVS7(S7-300, S7-400) instructions: (Page 202) and Communication via SENDDP/RCVDP instructions (S7-1200, S7-1500): (Page 248) under Data exchange between standard user program and safety program (Page 198)	
Compiling the safety program	—	under Compiling the safety program (Page 349)	
Implementing safety program call (S7-300, S7-400)	Check whether the main safety block is called directly in OBs (e.g., OB 35), FBs, or FCs.	under Defining F-Runtime Groups (Page 133)	
Installation			
Hardware configuration	Description of • Installation • Wiring	under Overview of Configuration (Page 35); AUTOHOTSPOT; *F-SMs Manual*, sections 5, 6; *F-Modules Manual*, sections 3, 4; *ET 200eco Manual*, sections 3, 4; *ET 200eco PN Manual*, sections 4, 5; *ET 200pro Manual*, Sections 2, 3; *ET 200iSP Manual*, sections 3 and 4; *ET 200SP Manual*, sections 5 and 6 *ET 200SP HA Manual*, sec. 5, 6 *ET 200AL Manual*, sections 4, 5 *ET 200MP Manual*, sections 5 and 6 *S7-1500R/H Manual*, sections 5, 6 *HB SIMATIC Drive Controller*, section 5, 6	
Commissioning, Testing			
Switching on	Description of commissioning – same as in standard	*Standard-S7-300 Manual;* *Standard-S7-400 Manual;* *Standard-S7-1200 Manual;* *Standard-S7-1500 Manual;* *Standard-S7-1500 Software Controller Manual;* *S7-1500R/H Module Manual* *HB SIMATIC Drive Controller Manual*	
Downloading safety program and standard user program	Description • Downloading • Program identification • Comparing safety programs	under Downloading project data (Page 350) under Comparing Safety Programs (Page 386)	
Testing the safety program	• Description of disabling of safety mode • Procedures for changing safety program data	under Downloading project data (Page 350); Testing the safety program (Page 397); Disabling safety mode (Page 393)	

Phase	Note the following	Reference	Check
Changing the safety program	Description • Disabling safety mode • Changing the safety program	under Changing the safety program in RUN mode (S7-300, S7-400) (Page 404), Safety program in RUN mode (S7-1200, S7-1500) (Page 406), Disabling safety mode (Page 393), Deleting the safety program (Page 131)	
Testing the safety-related parameters	Description of configuration	under Creating a safety summary (Page 388); *F-SMs Manual*, sections 4, 9, 10; *F-Modules Manual*, sections 2.4, 7; *ET 200eco Manual*, sections 3, 8; *ET 200eco Manual*, section 6; *ET 200pro Manual,* Sections 2.4, 8; *ET 200iSP Manual,* Sections 2.4, 7, 8 *ET 200SP Modules Manual*, section 4 *ET 200SP HA Modules manual*, sec. 5 *ET 200MP Modules Manual*, section 4 *ET 200AL Module Manual*, section 4	
System acceptance			
Acceptance	• Description and notes on acceptance • Printouts	under System acceptance (Page 413)	
Operation, maintenance			
General operation	Notes on operation	under Notes on Safety Mode of the Safety Program (Page 441)	
Access protection	—	under Access protection (Page 99)	
Diagnostics	Responses to faults and events	under Guide to Diagnostics (Page 448)	
Replacement of software and hardware components	Description • Module replacement • Update of operating systems of F-CPU – same as in standard • Update of SW components Notes • Operating system update of IMs	under Replacing Software and Hardware Components (Page 444); Addressing F-I/O (Page 157); Help on *STEP 7*	
Uninstalling the license, Disassembly	• Notes for uninstalling the license • Notes for disassembling modules	under Installing/uninstalling the STEP 7 Safety Advanced V21 license (Page 30), Replacing Software and Hardware Components (Page 444)	

Assignment of the F-BaseID during runtime (delayed transfer) (S7-1200 G2, S7-1500)

C

If the F-BaseID with the value "Delayed transfer" (0x7C6) has been configured, you must generate an active F-BaseID in runtime and assign it to the F-CPU and the F-I/O. The following instructions are available to you for this purpose.

- BASEID_GEN
- BASEID_DATA
- BASEID_CODING
- BASEID_HOST

The instructions are made available in the "Extended instructions" area under "Safety-Extensions".

These instructions are not fail-safe instructions and can only be placed in a standard block. They may only be used in the case "Delayed transfer of the F-BaseID".

In the case "Delayed transfer of the F-BaseID", the F-BaseID is generated once by these instructions on the first startup of the F-CPU, saved in the F-CPU and assigned to the F-I/Os. After the F-CPU starts up again, the F-BaseID previously saved in the F-CPU is applied again and forwarded to the F-I/O.

NOTE

The F-I/Os of PROFIsafe address type 3 remain passivated until the logic for assignment of the F-BaseID is complete.

NOTE

You can find the BaseIDAssign application example on the Internet. It presents a possible interconnection of the above-named instructions for the delayed transfer of the F-BaseID.

NOTE

The BASEID_GEN instruction is only supported for S7-1500 HF-CPUs as of firmware V4.2.

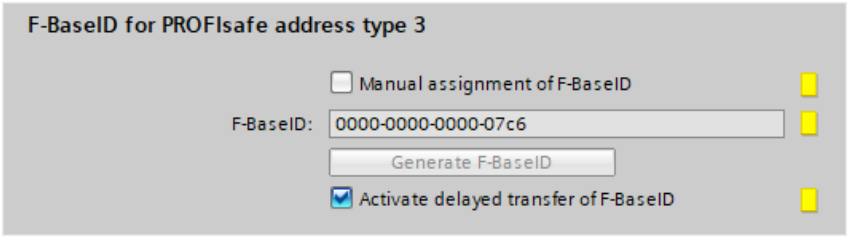
Configuring delayed transfer of the F-BaseID

For delayed transfer of the F-BaseID, configure the F-CPUs and the F-I/Os with the "Activate delayed transfer of F-BaseID" option enabled in accordance with the PROFIsafe specification.

Proceed as follows:

1. For the F-I/Os of PROFIsafe address type 3, configure (Page 70) the F-BaseID with the value 0 and with the intended F-source and F-destination addresses.
2. Download (Page 350) the project data to the F-CPU.
3. Assign (Page 73) the PROFIsafe addresses with SIMATIC Safety.
4. Create the user program.

5. In the F-CPU parameters (Page 41), enable the "Activate delayed transfer of the F-BaseID" option. The F-BaseID changes to 0x7C6.



6. In the standard program, program the logic for assigning the F-BaseID with BASEID_GEN, BASEID_DATA, BASEID_CODING and BASEID_HOST.
7. Download (Page 350) the project data to the F-CPU.
8. Switch the F-CPU to RUN. The F-BaseID is assigned.

⚠ WARNING

During assignment of the F-BaseID by means of the BASEID_CODING instruction, only the F-modules to which the F-BaseID is to be assigned should be accessible. Therefore, temporarily isolate the network in which the F-CPU and the assigned F-modules are located. *(S119)*

As an alternative to the BASEID_CODING instruction, you can use the instruction PROFIsafeAddrAssign (https://support.industry.siemens.com/cs/ww/en/view/109748466) (as of Version 3.0). With this instruction, you can additionally assign the F-destination and F-source addresses. Steps 1 to 3 are omitted in this case.

⚠ WARNING

Note that when the PROFIsafeAddrAssign instruction with F-I/Os of PROFIsafe address type 3 is being used, you must always confirm the correctness of the assignment using suitable measures. This applies even if you only changed the F-BaseID, because all parts of the PROFIsafe address are assigned at the same time in this case. In addition, you must prevent an unintended execution of the PROFIsafeAddrAssign instruction. You can find details on this in the named application example. *(S118)*

Order when programming the instructions for F-BaseID

In order to set the F-BaseID at runtime, you must interconnect the following instructions from the standard library under "Safety Extensions" in a cyclic standard user program:

- BASEID_GEN
- BASEID_DATA
- BASEID_CODING
- BASEID_HOST

NOTE

The instructions BASEID_GEN, BASEID_DATA, BASEID_CODING and BASEID_HOST are processed asynchronously. You must interconnect these instructions in such a way that they are processed sequentially in the above-named order.

C.1 Safety Extensions

C.1.1 BASEID_GEN

Description

At a positive edge at input REQ, the BASEID_GEN instruction generates the F-BaseID once and provides it at output BASEID.

If the F-CPU already has a set F-BaseID, this F-BaseID is provided at output BASEID.

Parameters

The following table shows the parameters of the BASEID_GEN instruction:

Parameter	Declaration	Data type	Description
REQ	Input	BOOL	REQ = 1: Runs BASEID_GEN.
HW_IDENTIFIER_IM	Input	HW_IO	The HW identifier of the executing F-CPU.
BASEID	Output	LINT	The read or generated F-BaseID.
CALCULATED	Output	BOOL	CALCULATED = 1: The output F-BaseID was generated.
DONE	Output	BOOL	DONE = 1: Processing of the instruction is complete.
BUSY	Output	BOOL	BUSY = 1: Generation of the F-BaseID is active.
ERROR	Output	BOOL	ERROR = 1: An error has occurred.
STATUS	Output	DWORD	Error information

NOTE

If you want to regenerate the F-BaseID but the parameter of the F-CPU no longer has the value 0x7C6, you must first reset this to 0x7C6 using the BASEID_Host instruction.

Instruction versions

The following version is available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	—	—	x[1]	x[1]	BASEID_GEN reads the F-BaseID from the parameters of the F-CPU or, if F-BaseID = 0x7C6 was read, generates a new F-BaseID.

[1] Supported as of firmware V4.1 (for S7-1500 HF-CPUs only as of V4.2) or as of V40.1 for S7-1500 F Software Controllers

STATUS output

The table below shows the meaning of the values for the STATUS parameter:

Error code	Explanation
0x0000	No error
0x8001	Error in the generation of the F-BaseID

C.1.2 BASEID_DATA

Description

The instruction BASEID_DATA calculates the F-BaseID CRC according to the PROFIsafe specification and writes the F-BaseID with the calculated F-BaseID CRC to the data record of the F-module specified at input HW_WRITE_RECORD_NR. To calculate the F-BaseID CRC, the F_Par_CRC is read from the data record of the F-module specified at the HW_READ_RECORD_NR input.

Parameters

The following table shows the parameters of the BASEID_DATA instruction:

Parameter	Declaration	Data type	Description
REQ	Input	BOOL	REQ = 1: Runs BASEID_DATA.
BASEID	Input	LINT	The F-BaseID to be written.
HW_IDENTIFIER_MODULE	Input	HW_IO	The HW identifier of the F-module.
HW_READ_RECORD_NR	Input	DINT	Data record number of the F-module for reading the F_Par_CRC (needed to calculate the F-BaseID CRC).
HW_WRITE_RECORD_NR	Input	DINT	Data record number of the F-module for writing the F-BaseID.
DONE	Output	BOOL	DONE = 1: Processing of the instruction is complete.
BUSY	Output	BOOL	BUSY = 1: The instruction is running.
ERROR	Output	BOOL	ERROR = 1: An error has occurred.
STATUS	Output	DWORD	Error information

NOTE

The inputs HW_READ_RECORD_NR (128) and HW_WRITE_RECORD_NR (130) are preassigned with the usual data record numbers for F-modules by SIEMENS.

Instruction versions

The following version is available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	—	—	x[1]	x[1]	BASEID_DATA writes the F-BaseID to the data record of an F-module

[1] Supported as of firmware V4.1 or as of V40.1 for S7-1500 F Software Controllers

STATUS output

The table below shows the meaning of the values for the STATUS parameter:

Error code	Explanation
0x0000	No error
0x8001	Error during generation of the F-BaseID CRC
0x0050	The F-BaseID is in the reserved range.

C.1.3 BASEID_CODING

Description

The instruction BASEID_CODING writes the F-BaseID to the electronic coding element of the F-module specified at input HW_IDENTIFIER_MODULE. If the values in the data record and the electronic coding element do not match, an inconsistency occurs and the F-module is passivated until the values match.

NOTE

Using the BASEID_CODING instruction, the F-BaseID in the electronic coding element can only be overwritten if it still has the value 0.

Otherwise, the firmware prevents the writing of the new F-BaseID and the existing value in the electronic coding element is retained. To be able to change an F-BaseID, a project with F-BaseID with value 0 must therefore first be loaded and an assignment made via TIA Portal.

"Delayed transfer" must then be parameterized again in the project and this project must be loaded. Now the F-BaseID can be set at the electronic coding element again with the BASEID_CODING instruction.

NOTE

As an alternative to using the BASEID_CODING instruction, the F-BaseID can also be assigned together with the PROFIsafe addresses. For this you can use the instruction PROFIsafeAddrAssign (https://support.industry.siemens.com/cs/ww/en/view/109748466) (as of Version 3.0) available on the Internet.

⚠ WARNING

Note that when the PROFIsafeAddrAssign instruction with F-I/Os of PROFIsafe address type 3 is being used, you must always confirm the correctness of the assignment using suitable measures. This applies even if you only changed the F-BaseID, because all parts of the PROFIsafe address are assigned at the same time in this case. In addition, you must prevent an unintended execution of the PROFIsafeAddrAssign instruction. You can find details on this in the named application example. *(S118)*

⚠ WARNING

During assignment of the F-BaseID by means of the BASEID_CODING instruction, only the F-modules to which the F-BaseID is to be assigned should be accessible. Therefore, temporarily isolate the network in which the F-CPU and the assigned F-modules are located. *(S119)*

Parameters

The following table shows the parameters of the BASEID_CODING instruction:

Parameter	Declaration	Data type	Description
REQ	Input	BOOL	REQ = 1: Runs BASEID_CODING.
BASEID	Input	LINT	The F-BaseID to be written.
HW_IDENTIFIER_MODULE	Input	HW_IO	The HW identifier of the F-module.
DONE	Output	BOOL	DONE = 1: Processing of the instruction is complete.
BUSY	Output	BOOL	BUSY = 1: The instruction is running.
ERROR	Output	BOOL	ERROR = 1: An error has occurred.
STATUS	Output	DWORD	Error information

Instruction versions

The following version is available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	—	—	x[1]	x[1]	BASEID_CODING writes the F-BaseID to the electronic coding element of an F-module.

[1] Supported as of firmware V4.1 or as of V40.1 for S7-1500 F Software Controllers

STATUS output

The table below shows the meaning of the values for the STATUS parameter:

Error code	Explanation
0x0000	No error
0x0050	The F-BaseID is in the reserved range.

C.1.4 BASEID_HOST

Description

The instruction BASEID_HOST writes the specified F-BaseID to the parameters of the executing F-CPU.

Parameters

The following table shows the parameters of the BASEID_HOST instruction:

Parameter	Declaration	Data type	Description
REQ	Input	BOOL	REQ = 1: Runs BASEID_HOST.
BASEID	Input	LINT	The F-BaseID to be written.
DONE	Output	BOOL	DONE = 1: Processing of the instruction is complete.
BUSY	Output	BOOL	BUSY = 1: The instruction is running.
ERROR	Output	BOOL	ERROR = 1: An error has occurred.
STATUS	Output	DWORD	Error information

Instruction versions

The following version is available for this instruction:

Version	S7-300/400	S7-1200 G1	S7-1200 G2	S7-1500	Function
1.0	—	—	x[1]	x[1]	BASEID_HOST writes the F-BaseID to the parameters of the F-CPU.

[1] Supported as of firmware V4.1 or as of V40.1 for S7-1500 F Software Controllers

STATUS output

The table below shows the meaning of the values for the STATUS parameter:

Error code	Explanation
0x0000	No error
0x0050	The F-BaseID is in the reserved range.

Glossary

Access protection

→ Fail-safe systems must be protected against dangerous, unauthorized access. Access protection for F-systems is implemented by assigning a password for the → F-CPU and a password or → project protection for the → Safety program.

Automatically generated F-blocks

→ F-blocks that are automatically generated and, if necessary, called when the → Safety program is compiled, in order to generate an executable Safety program from the Safety program programmed by the user.

Category

Category in accordance with ISO 13849-1:2015 or EN ISO 13849-1:2015

With SIMATIC Safety, use in → safety mode up to category 4 is possible.

Channel fault

Channel-specific fault, such as a wire break or short-circuit

Collective F-HW-Signature

The Collective F-HW-Signature uniquely identifies a particular state of safety-related hardware configuration. It is important for documenting the change/non-change of the safety-related hardware configuration, for example in the context of an acceptance of changes. Bear in mind that the → F-BaseID does not enter into the Collective F-HW signature.

Collective F-Signature

The Collective F-Signature uniquely identifies a particular state of safety-related project data. It is important for the program identification as well as the on-site acceptance of the Safety program, for example by → experts.

Collective F-SW-Signature

The Collective F-SW-Signature uniquely identifies a particular state of the Safety program. The Collective F-SW-Signature is important to document the change/non-change of the Safety program, for example in the context of an acceptance of changes.

CPU-wide

In the context of F-I/Os, "CPU-wide" means all F-I/Os assigned to an F-CPU: Central F-I/O of this F-CPU, F-I/O for which the F-CPU is DP master/IO controller and assigned F-I/O in a shared device. An F-I/O that is addressed using I-slave-slave communication is assigned to the F-CPU of the I-slave and not to the F-CPU of the DP master / IO controller.

In the context of safety-related CPU-CPU communication, "CPU-wide" encompasses all the safety-related communication connections that are configured in an F-CPU.

With a redundant S7-1500HF system, both F-CPUs of the redundant S7-1500HF system are to be regarded as one F-CPU.

CRC

Cyclic Redundancy Check → CRC signature

CRC signature

The validity of the process data in the → safety message frame, the correctness of the assigned address relationships, and the safety-related parameters are validated by means of a CRC signature contained in the safety message frame.

DB for F-runtime group communication

-> F-DB for safety-related communication between F-runtime groups of a Safety program.

Depassivation

→ Reintegration

Disabled safety mode

Temporary deactivation of → safety mode for test purposes, commissioning, etc.

A timer is started when the safety mode is disabled. After the timer has expired, the → F-CPU goes to STOP. The time for the timer is parameterizable.

The following actions are possible only in deactivated safety mode:

- Downloading changes of the → Safety program to the -> F-CPU during operation (in RUN mode)
- Test functions such as "Modify" or other write access to data of the → Safety program (with limitations)

Whenever safety mode is deactivated, the safety of the system must be ensured by other organizational measures, such as operation monitoring and manual safety shutdown.

Discrepancy analysis

Discrepancy analysis for equivalence or non-equivalence is used for fail-safe inputs to detect errors caused by the time characteristic of two signals with the same functionality. The discrepancy analysis is initiated when different levels are detected in two associated input signals (when testing for non-equivalence: the same level). On expiration of an assignable period (→ discrepancy time), a check is made to determine whether the difference in levels (for non-equivalence testing, the same level) has disappeared after an assignable time period, the so-called discrepancy time. If not, there is a discrepancy error. The discrepancy analysis is performed between the two input signals of the 1oo2 evaluation of the sensors (→ sensor evaluation) in the fail-safe input.

Discrepancy time

Assignable time for the → discrepancy analysis. If the discrepancy time is set too high, the fault detection time and → fault reaction time are prolonged unnecessarily. If the discrepancy time is set too low, availability is decreased unnecessarily because a discrepancy error is detected when, in reality, no error exists.

DP/DP coupler

Device for coupling two PROFIBUS DP subnets required for master-master communication between → Safety programs in different → F-CPUs in SIMATIC Safety and S7 Distributed Safety.

Expert

The acceptance of a system, i.e., the safety-related acceptance test of the system, is usually carried out by an independent expert (for example, from TÜV).

Fail-safe GSD based DP slaves

Fail-safe GSD based DP slaves are standard slaves that are operated on PROFIBUS with the DP protocol. They must comply with IEC 61784-1-3 (PROFIBUS protocol) and IEC 61784-3-3 (PROFIsafe specifications). A GSD file is used for their configuration

Fail-safe GSD based I/O devices

Fail-safe GSD based I/O devices are standard devices that are operated on PROFINET with the I/O protocol. They must comply with IEC 61784-2-3 (PROFINET profile) and the PROFIsafe bus profile in accordance with IEC 61784-3-3. A GSD file is used for their configuration.

Fail-safe I/O modules

ET 200AL modules, ET 200eco modules and ET 200eco PN modules that can be used for safety-related operation (→ Safety mode). These modules are equipped with integrated → safety functions. They operate in accordance with IEC 61784-2-3 (PROFINET profile) and the PROFIsafe requirements in accordance with IEC 61784-3-3.

Fail-safe modules

Fail-safe modules ET 200SP, ET 200SP HA, ET 200S, ET 200pro, ET 200iSP which can be used in the ET 200SP, ET 200SP HA, ET 200S, ET 200pro or ET 200iSP distributed I/O systems.

Fail-safe modules S7-1500/ET 200MP, which can be used centrally in an S7-1500 or in a distributed I/O ET 200MP system.

Fail-safe module S7-1200 which can be used centrally in an S7-1200 system.

These modules are equipped with integrated safety functions (→ Safety mode) for fail-safe operation (→ Fail-safe operation). They operate in accordance with the → PROFIsafe bus profile.

Fail-safe systems

Fail-safe systems (F-systems) are systems that remain in a safe state or immediately switch to another safe state as soon as particular failures occur.

Fault reaction function

→ User safety function

Fault reaction time

The maximum fault reaction time for an F-system specifies the time between the occurrence of any error and a safe reaction at all affected fail-safe outputs.

F-BaseID

→ PROFIsafe address

F-blocks

The following fail-safe blocks are designated as F-blocks:

- those created by the user in LAD or FBD
- those created by the user as → F-DBs
- those selected by the user from a global library
- those added automatically in the → Safety program (→ F-SBs, → automatically generated F-blocks, → F-shared DB, → F-I/O DBs; instance DBs of F-FBs)

All F-blocks are shown in yellow in the project tree.

F-CALL

"F-call blocks" for the → Safety program in *S7 Distributed Safety*.

F-Communication Address Signature

The F-communication address signature is formed from the names and the F-communication UUIDs of communication connections with Flexible F-Link that are used in the Safety program.

F-communication DBs

Fail-safe data blocks for the

- safety-related CPU-CPU communication via S7 connections
- Communication with Flexible F-Link

F-compliant PLC data type (UDT)

An F-compliant PLC data type (UDT) is a PLC data type (UDT) in which you can use all data types that can be used in Safety programs.

F-CPU

An F-CPU is a central processing unit with fail-safe capability that is approved for use in SIMATIC Safety and in which a → Safety program can run in addition to the → standard user program.

F-cycle time

The F-cycle time is the time between two calls of an F-runtime group. It is monitored by the F-CPU. As soon as it exceeds the maximum F-cycle time (property of the F-runtime group), the F-CPU is set to the STOP operating state.

(S7-1200, S7-1500): The F-runtime group also has a warning limit. If the F-cycle time exceeds this warning limit, an entry is written to the diagnostic buffer.

F-DBs

Optional fail-safe data blocks that can be read-/write-accessed from anywhere within the Safety program (exception: DBs for F-runtime group communication).

F-destination address

→ PROFIsafe address

F-FBs

Fail-safe function blocks (with instance DBs), in which the user programs the → Safety program in FBD or LAD.

F-FCs

Fail-safe FCs, in which the user programs the → Safety program in → FBD or → LAD.

F-I/O

Collective name for fail-safe inputs and outputs available in *SIMATIC S7* for integration in SIMATIC Safety, among others. The following are available:

- → ET 200eco fail-safe I/O module
- → ET 200eco PN fail-safe I/O module
- → ET 200AL fail-safe I/O module
- → S7-300 fail-safe signal modules
- → Fail-safe modules for S7-1200
- → Fail-safe modules for ET 200MP
- → Fail-safe modules for ET 200SP
- → Fail-safe modules for ET 200SP HA
- → Fail-safe modules for ET 200S
- → Fail-safe modules for ET 200pro
- → Fail-safe modules for ET 200iSP

- → Fail-safe GSD based DP slaves
- → Fail-safe GSD based I/O devices

F-I/O DB

Fail-safe data block for F-CPUs to an → F-I/O in *STEP 7 Safety*. An F-I/O DB is automatically created for each F-I/O when the F-I/O is configured in the *hardware and network editor*. The F-I/O DB contains tags that the user can or must evaluate or write in the Safety program as follows:

- For reintegration of the F-I/O after communication errors
- For reintegration of F-I/O after F-I/O or channel faults
- If the F-I/O must be passivated as a result of particular states of the Safety program (for example, group passivation)
- For reassignment of parameters for fail-safe GSD based DP slaves/GSD based I/O devices or enabling HART communication for the F-I/O with the corresponding functionality
- For evaluation of whether substitute values or process values are being output

F-I/O faults

Module-related F-I/O fault, such as a communication error or parameter assignment error

F-I/O of PROFIsafe address type 3

F-I/O for which the uniqueness of the PROFIsafe address can be ensured through an F-BaseID in addition to the combination of F-source address and F-destination address, for example, certain S7-1500/ET 200SP F-modules. The F-BaseID saves you the work of ensuring the uniqueness of the PROFIsafe address network-wide/worldwide yourself. You normally assign the PROFIsafe address (including the F-BaseID) with *STEP 7 Safety*.

The terminology "PROFIsafe address type 3" used in this documentation is equivalent to the designation "PROFIsafe address type 2 with BaseID". This applies in TIA Portal as well as to F-I/Os.

F-I/Os of PROFIsafe address type 1

F-I/Os which ensure the uniqueness of the PROFIsafe address solely with the F-destination address, for example, ET 200S F-modules.

The PROFIsafe address is usually assigned by DIP switches.

F-I/Os of PROFIsafe address type 2

F-I/O for which the uniqueness of the PROFIsafe address can be ensured through the combination of F-source address and F-destination address, for example, certain S7-1500/ET 200MP F-modules. You obtain the PROFIsafe address type from the respective equipment manual.
You normally assign the PROFIsafe address with *STEP 7 Safety*.

F-modules

→ Fail-safe modules

F-OB

The F-OB calls the main safety block of an F-runtime group in S7-1200/1500 F-CPUs.

F-RTG

→ F-runtime group

F-runtime group

The → Safety program consists of one or two F-runtime groups. An F-runtime group is a logical construct of several associated → F-blocks. It is generated internally by the F-system. An F-runtime group consists of the following F-blocks:

→ Main safety block, F-OB (S7-1200, S7-1500), if applicable → F-FBs/ → F-FCs, if applicable → F-DBs, → F-I/O DBs, F-blocks of global libraries, instance DBs, → F-SBs, and → automatically generated F-blocks.

F-runtime group information DB

The F-runtime group information DB provides key information on the corresponding → F-runtime group and on the → Safety program as a whole.

F-shared DB

(S7-300, S7-400) Fail-safe data block that contains all of the shared data of the → Safety program and more information needed by the F-system. The F-shared DB is automatically inserted and expanded when the hardware configuration is compiled. Using its name F_GLOBDB, the user can evaluate certain data of the → Safety program.

F-SMs

→ S7-300 fail-safe signal modules

F-source address

→ PROFIsafe address

F-system blocks

Fail-safe system blocks that are automatically inserted and called when the → Safety program is compiled in order to generate an executable Safety program from the user's Safety program.

F-systems

→ Fail-safe systems

Hardware configuration

The hardware configuration includes the configuration of standard CPUs and standard I/Os as well as the configuration of F-CPUs and F-I/Os.

Hypervisor

A hypervisor provides the configured hardware offered by the host for a virtual machine.

I-device

The functionality of the "I-device" (intelligent I/O-device) of a CPU allows data exchange with an I/O-controller and thus, its use as an intelligent preprocessor of sub-processes, for example. In this case, the I-device is connected as an I/O-device to a "parent" I/O-controller.

IE/PB link

Device for coupling PROFINET IO and PROFIBUS DP-systems required, among other things, for IO-controller-I-slave communication between -> Safety programs in different → F-CPUs in SIMATIC Safety.

i-parameter

Individual parameters of → fail-safe GSD based DP slaves and → fail-safe GSD based I/O devices

I-slave

The functionality of the "I-slave" (intelligent DP slave) of a CPU allows data exchange with a DP master and, thus, its use as an intelligent preprocessor of sub-processes, for example. In this case, the I-slave is connected as a DP slave to a "parent" DP master.

Main safety block

"Introductory F-block" for fail-safe programming of the → Safety program in *STEP 7 Safety*. The main safety block is an → F-FB or → F-FC that the user assigns to the calling F-OB (S7-1200, S7-1500) or block (OB, FC, FB) (S7-300, S7-400) of an → F-runtime group.

The main safety block contains the Safety program and any calls of other → F-FBs/F-FCs for program structuring.

Network-wide

A network consists of one or more subnets. "Network-wide" means beyond the boundaries of the subnet. In PROFIBUS, a network includes all the nodes accessible via PROFIBUS DP. In PROFINET IO, a network includes all the nodes accessible via RT_Class_1/2/3 (Ethernet/WLAN/Bluetooth, Layer 2) and, if applicable, RT_Class_UDP (IP, Layer 3).

OPC UA Safety

The safety-related communication via OPC UA Safety is specified by the OPC Foundation in OPC UA Specification OPC 10000-15 and is based on the mechanisms of OPC UA.

Organizational measures

For the *SIMATIC Safety* F-system, these are suitable measures to be defined by the operator of the plant:

- To ensure the safety of the plant in certain operating situations, for example:
  - Monitored operation with manual safety shutdown
  - Operating instructions
  - Regular instructions
- To protect the plant against unauthorized operation or access, for example:
  - Access control to the plant or to the F-CPU or to the PG/PC
  - Operating instructions
  - Regular instructions

Passivation

When passivation occurs in an → F-I/O with inputs, the → F-system provides the Safety program with fail-safe values (0) instead of the process data pending at the fail-safe inputs in the PII.

When passivation occurs in an F-I/O with outputs, the F-system transfers fail-safe values (0) to the fail-safe outputs instead of the output values in the PIQ provided by the Safety program.

PL

Performance Level (PL) in accordance with ISO 13849-1:2015 or EN ISO 13849-1:2015

With SIMATIC Safety, use up to Performance Level (PL) e is possible in → safety mode.

PN/PN coupler

Device for coupling two PROFINET IO systems required for IO controller-IO controller communication between → Safety programs in different → F-CPUs in SIMATIC Safety and S7 Distributed Safety.

Productive operation

Operation of a plant in the specified runtime environment for which the safety requirements have been specified and in which all safety requirements must be observed.

By contrast, safety requirements in a test or simulation environment must only be implemented to a limited extent because the risk potential is not the same (for example, there are no physical actuators that could pose a risk).

PROFIsafe

Safety-related bus profile of PROFIBUS DP and PROFINET IO for communication between the → Safety program and the → F-I/O in an → F-system. See IEC 61784-3-3 or PROFIsafe – Profile for Safety Technology on PROFIBUS DP and PROFINET IO; Order No: 3.192.

PROFIsafe address

The PROFIsafe address (see IEC 61784-3-3) is used as a safety measure for standard addressing mechanisms, such as IP addresses. The PROFIsafe address consists of the F-source address, the F-destination address and the optional F-BaseID. For this reason, each → F-I/O has the following address components: F-source address, F-destination address and, if supported, F-BaseID. (If an F-I/O does not support the F-BaseID, the F-BaseID implicitly equals 0.)

The F-source address is automatically assigned and is displayed for fail-safe GSD based DP slaves/GSD based I/O devices and F-modules ET 200SP, F-modules ET 200SP HA, F-modules ET 200SP, F-modules ET 200MP, F-modules ET 200AL, ET 200eco PN and F-modules S7-1200. The F-source address for F-modules ET 200S, ET 200eco, ET 200pro, ET 200iSP and F-SMs S7-300 is always 1. For ET 200SP F-modules, ET 200MP F-modules, ET 200AL F-modules and ET 200eco PN, the F-source address corresponds to the "Central F-source address" parameter of the assigned F-CPU.

You must configure the F-destination address in the hardware and network editor. You assign the F-destination address for the ET 200S, ET 200eco, ET 200pro, ET 200iSP and F-SMs S7-300 F-modules with a switch. For ET 200SP F-modules, ET 200SP HA F-modules, ET 200MP F-modules, ET 200AL F-modules and ET 200eco PN, you assign the PROFIsafe address in the hardware and network editor. For S7-1200 F-modules, the F-destination address is automatically assigned by the F-system.

You configure the optional F-BaseID in the hardware and network editor of the F-CPU. For certain F-modules, it is possible to assign the F-BaseID as part of the PROFIsafe address (see the respective equipment manual).

Program signature

→ Collective F-Signature

Project data

The project data includes the → hardware configuration and the → user program.

Project protection

A user administration (UMAC - User Management and Access Control) to create and manage users and roles for projects. In addition, you can protect your project and determine which users are allowed to carry out which functions. For safety-related project data, you have the function right "Edit safety-related project data".

Redundant systems

Redundant systems are characterized by the fact they contain important automation components present multiple times (redundantly). If a redundant component fails, control of the process is maintained.

Reintegration

The switchover from fail-safe values (0) to process data (reintegration of an → F-I/O) takes place automatically or following user acknowledgment in the F-I/O DB. The reintegration method depends on the following:

- The reason for → passivation of the F-I/O/channels of the F-I/O
- The parameter assignment in the → F-I/O DB or in the configuration itself (for example, ET 200MP fail-safe modules on an S7-1500 F-CPU and S7-1200 fail-safe modules on an S7-1200 F-CPU)

Following reintegration for an → F-I/O module with inputs, the process data pending at the inputs in the PII are provided again for the Safety program. For an F-I/O with outputs, the F-system again transfers the output values provided in the PIQ in the Safety program to the fail-safe outputs.

RIOforFA Safety

Remote IO for Factory Automation with PROFIsafe; profile for F-I/O

RTG

→ F-runtime group

S7-300 fail-safe signal modules

Fail-safe signal modules of the S7-300 module series that can be used for safety-related operation (→ Safety mode) as centralized modules in an S7-300 or as distributed modules in the ET 200M distributed I/O system. The fail-safe signal modules are equipped with integrated → safety functions. They operate in accordance with the → PROFIsafe bus profile.

S7-PLCSIM

The *S7-PLCSIM* application enables you to execute and test your program on a simulated automation system on your programming device or PC. Because the simulation takes place completely in your programming device or PC, you do not need any hardware (CPU, I/O).

Safe state

The basic principle of the safety concept in → fail-safe systems is the existence of a safe state for all process variables. For digital → F-I/O that conform to IEC 61508:2010, this is always the value "0".

Safety Administration Editor

The *Safety Administration Editor* provides support for the main tasks of your Safety program.

Safety function

Mechanism integrated in the → F-CPU and → F-I/O that allows them to be used in -> fail-safe systems.

According to IEC 61508:2010, a function that is implemented by a safety device in order to maintain the system in the safe state or bring the system to a safe state in the event of a specific fault. (fault reaction function -> user safety function)

Safety message frame

In → safety mode, data are transferred in a safety message frame between the → F-CPU and → F-I/O, or between the F-CPUs in safety-related CPU-CPU communication.

Safety mode

1. Operating mode of → F-I/O in which → safety-related communication can take place using → safety message frames.
2. Operating mode of the Safety program. In safety mode of the Safety program, all safety mechanisms for error detection and fault reaction are enabled. In safety mode, the Safety program cannot be modified during operation. Safety mode can be disabled by the user (→ disabled safety mode).

Safety program

Safety-related user program

Safety protocol

→ Safety message frame

Safety summary

The safety summary is the documentation of the safety-related project data, which supports you during acceptance of the system. The safety summary can be created in printed or electronic form, for example, as a PDF file.

Safety Unit

A Software Unit that contains the entire Safety program of an F-CPU.

SafetyConsumer

OPC UA Safety designates an instance that evaluates the data provided by a SafetyProvider instance as a SafetyConsumer.

SafetyProvider

OPC UA Safety refers to an instance that provides data as a SafetyProvider.

Safety-related communication

Safety-related communication is used to exchange fail-safe data.

Safety-related hardware configuration

The safety-related hardware configuration includes the safety-related parameters of the F-CPUs and F-I/Os.

Safety-related project data

The safety-related project data includes the safety-related hardware configuration as well as the → Safety program.

SAPI

The Safety Application Program Interface (SAPI) is the interface for your Safety program (or standard user program) for OPC UA Safety communication.

Sensor evaluation

There are two types of sensor evaluation:

- 1oo1 evaluation – sensor signal is read once
- 1oo2 evaluation - sensor signal is read twice by the same → F-I/O and compared internally

Shared device

The "Shared device" functionality enables modules of an IO device to be shared by different IO controllers. More information can be found in the "PROFINET with STEP 7" Function Manual.

Signature

→ Collective F-Signature

SIL

Safety Integrity Level (SIL) in accordance with IEC 61508:2010 or EN 61508:2010. The higher the Safety Integrity Level, the more rigid the measures for prevention of systematic faults and for management of systematic faults and random hardware failures.

With SIMATIC Safety, up to Safety Integrity Level SIL3 is possible in safety mode.

SPI

The Safety Parameter Interface (SPI) is the interface for the parameterization of the OPC UA Safety communication for safety-oriented data transfer.

Standard communication

Communication used to exchange non-safety-related data

Standard mode

Operating mode of → F-I/O in which → safety-related communication between the F-CPU and the F-I/O by means of → safety message frames is not possible; only → standard communication is possible in this operating mode.

Standard project data

The standard project data includes the standard hardware configuration and the → standard user program.

Standard user program

Non-safety-related user program

Startup of F-system

See section "Notes on startup of the F-system".

User program

The user program comprises the → standard user program and the → Safety program.

User safety function

The → safety function for the process can be provided through a user safety function or a fault reaction function. The user only has to program the user safety function. In the event of an error, if the → F-system can no longer execute its actual user safety function, it executes the fault reaction function: for example, the associated outputs are disabled, and the → F-CPU switches to STOP mode, if necessary.

Value status

The value status is additional binary information for a channel value. The value status is entered in the process image input and provides information on the validity of the channel value.

1: A valid process data is output for the channel value.

0: A fail-safe value is output for the channel value.

Versioned instruction

Instruction for which a version is displayed in the "Version" column of the "Instructions" task card.

Index

=

=, 476

A

ABS, 568

Acceptance, 79
  of system, 413
  of safety-related changes, 433

Acceptance of safety-related changes, 433

Access
  To tags of F-I/O DB, 172

Access permission
  Validity, 100
  Setup for F-CPU, 105
  Canceling, 106
  Validity, 106
  Setup for F-CPU, 108
  Canceling, 108
  Validity, 108

Access protection, 99
  safety-related project data, 100
  Setting up for the safety program, 100
  CPU-wide, 100
  UMAC, 102
  project-wide, 102
  for the F-CPU, 103
  through organizational measures, 108

ACK_GL, 530

ACK_NEC, 166

ACK_OP, 604

ACK_REI, 167

ACK_REQ, 170

Acknowledgment
  Channel fault, 54
  Fail-safe, 604

Add, 553

ADD, 553

Address assignment
  Rules, 68
  Rules, 69
  Rules, 71

AND, 472, 596, 596

Approvals, 5

Assignment, 461, 476

B

Behavior
  After startup, 173
  After communication errors, 175
  After F-I/O or channel faults, 177

Bit logic operation
  Insert binary input, 457
  Invert RLO, 458
  Normally open contact, 459
  Normally closed contact, 460
  Invert RLO, 460
  Assignment, 461
  Reset output, 462
  Set output, 463
  Set/reset flip-flop, 464
  Reset/set flip-flop, 466
  Scan operand for positive signal edge, 467
  Scan operand for negative signal edge, 468
  Scan RLO for positive signal edge, 470
  Scan RLO for negative signal edge, 471
  AND, 472
  OR, 473
  EXCLUSIVE OR, 474
  Assignment, 476
  Reset output, 476
  Set output, 478
  Set/reset flip-flop, 479
  Reset/set flip-flop, 480
  Scan operand for positive signal edge, 481
  Scan operand for negative signal edge, 483
  Scan RLO for positive signal edge, 484
  Scan RLO for negative signal edge, 485

Bit memory, 198

Block size of automatically generated F-blocks, 350

BO_W, 580
Breakpoints, 399

C

Category, 23
Central F-source address, 44
Change
  of the safety program in RUN mode, 404
changes not supported , 406
Changing
  Data of the safety program, 397
  of the safety program in RUN mode, 406
  Detecting, 433
  Acceptance, 433
Channel fault, 54
  Acknowledgment, 54
Checking the program version, 431
Checklist, 638
Checks
  through the F-system, 65
CMP <, 552
CMP <=, 550
CMP <>, 547
CMP ==, 546
CMP >, 551
CMP >=, 548
Code review of the safety program, 414
Collective F-Signature, 393
Communication
  Standard user program and safety program, 198
  Standard user program and safety program, 200
  Monitoring time, 632
  Monitoring time, 633
Communication error, 175
  SENDDP/RCVDP, 612
Comparator operations
  Equal, 546
  Not equal, 547
  Greater or equal, 548
  Less or equal, 550
  Greater than, 551
  Less than, 552
Comparing
  Safety programs, 386
Compiling errors
  Alarms, 349
Completeness
  Checking the safety summary, 415
Configuration control, 57
Configuring
  Overview, 35
  Fail-safe GSD based I/O devices, 39
  Fail-safe GSD based DP slaves, 39
  F-CPU, 41
  F-I/O, 53
constants
  Boolean, 123
  TRUE, 123
  FALSE, 123
Conversion operations
  Convert value, 578
  Convert BOOL to WORD, 580
  Convert WORD to BOOL, 582
  Scale values, 584
  Scale values to data type DINT, 586
Convert
  Value, 578
  Data, 580
  Data, 582
CONVERT, 578
Convert data, 580, 582
Conveyor equipment, stopped, 495
Correctness
  Hardware configuration, 418
  Safety-related CPU-CPU communication, 424
Count
  Up, 540
  Down, 542
  Up and down, 544
Count down, 542
Count up, 540
Count up and down, 544
CPU-CPU communication, 35
  Options for safety-related, 35
  Overview of safety-related, 202
  Overview of safety-related, 248
  Overview of safety-related, 283
Create twos complement, 565
CTD, 542

CTU, 540
CTUD, 544
Cycle time
F-runtime group, 136
F-runtime group, 139
Maximum, 630
Monitoring time for, 632

D

Data block, 198
Data exchange
between standard user program and safety program, 198
Data transfer
From safety program to standard user program, 198
From standard user program to safety program, 200
Data type conversion, 122
Data types
For safety program, 120
DB access, fully qualified, 124, 172
DB access, non-fully qualified, 125
Deleting
F-blocks, 131
DIAG
F-I/O DB, 171
ESTOP1: Emergency STOP up to Stop Category 1, 486
TWO_H_EN: Two-hand monitoring with enable, 492
MUTING: Muting, 495
MUT_P: Parallel muting, 504
EV1oo2DI: 1oo2 evaluation with discrepancy analysis, 515
FDBACK: Feedback monitoring, 520
SFDOOR: Safety door monitoring, 525
SENDDP/RCVDP, 612
SENDS7, 622
RCVS7, 622
Diagnostic parameters, 448
Diagnostics
Fail-safe system, 448
Diagnostic tag, 448
DISABLE, 169
Disabling
F-capability, 41
Safety mode, 392
Safety mode, 393
Discrepancy error, 495
DIV, 562
Divide, 562
Downloading
Standard user program, 350
Hardware configuration, 350
Safety program, 350
Downloading Standard user program, 350
DP/DP coupler, 212, 259

E

Empty box
Inserting a LAD element, 453
Inserting an FBD element, 456
EN, 119
Enabling
F-capability, 41
Safety mode, 396
ENO, 119
ESTOP1, 486
EV1oo2DI, 515
EXCLUSIVE OR, 474, 598
Executing a system acceptance, 413

F

F_CRC_Seed, 41
F_IO_StructureDescCRC, 39, 41
F_Passivation, 41
Fail-safe acknowledgment, 604, 608
Fail-safe GSD based DP slaves
Configuring, 39
Fail-safe GSD based I/O devices
Configuring, 39
Fail-safe system, 23
Fail-safe value, 143, 162
F-array
read, 571
read, 573
Fast Commissioning mode, 406

Fault reaction function, 11, 23
F-BaseID, 44, 44, 63, 70, 643
FBD element
  Inserting, 456
F-block
  Deleting, 131
  Copying, 151
F-change history, 412
F-channel faults, fail-safe value output, 162
F-Communication UUID, 97
F-compliant PLC data type (UDT), 126
F-components, 35
F-CPU, 31
  migrating, 31
  Configuring, 41
  Setting up access permission, 105
  Setting up access permission, 108
  Going to STOP mode, 441
F-cycle time, monitoring time, 632
F-DB, 117
  for F-runtime group communication, 143
  Creating, 150
  F-shared DB, 154
FDBACK, 520
F-destination address, 67, 68
F-destination address range, 43
F-FB, 117, 150
F-FC, 117, 150
F-I/O, 35
  Configuring, 53
  addressing, 157
  Reintegration, 163
  Reintegration, 173
  Reintegration, 175
  Reintegration, 177
  Removing and inserting during operation, 444
F-I/O access, 157
  Via the process image, 157
  Via the process image, 189
  During operation, 404
  Restrictions in RUN mode, 405
F-I/O access via I-slave-slave communication, 183
F-I/O DB, 55
  Number, 55
  Name, 55
  Structure of DIAG, 171
  Access to, 172
F-I/O DB, 117
  Access to, 164
  Number, 172
  Name, 172
F-I/O faults, fail-safe value output, 162
F-I/O or channel faults, 177
Firmware update, 444
First steps, 33
Flexible F-Link, 97
  F-monitoring time, 634
Flip-flop
  Set/reset, 464
  Reset/set, 466
  Set/reset, 479
  Reset/set, 480
F-monitoring time, 45
  F-communication, 97
F-OB, 55
  Copying, 151
Form absolute value, 568
F-parameters, 36
F-runtime group, 111
  Rules, 133
  Default setting, 135
  Defining, 136
  Maximum cycle time, 136
  Default setting, 138
  Defining, 139
  Maximum cycle time, 139
  Safety-related communication, 143
  Deleting, 149
  Changing, 149
  Changing, 149
  Maximum cycle time, 634
F-runtime group communication, 136
  Restrictions in RUN mode, 404
  Monitoring time, 634
F-runtime group information DB, 154
F-runtime groups signature, 154

F-shared DB, 154, 198, 393
F-source address, 44, 68
F-system
  Checks, 65
  Monitoring time, 629
  Response time, 629
Fully qualified DB access, 124, 172
Function test of the safety program, 391, 414

G

Getting Started, 33
Global data block
  Open, 595
Group diagnostics for fail-safe signal modules, 56
Group passivation, 181
GSD files
  Configuration, 39

H

Hardware components, 24
Hardware configuration, 37
  Downloading, 350
  Checking for correctness, 418
HW identifier, 612

I

IE/PB link, 236, 281
Image file
  Creating, 375
  Installing, 376
Implementation of user acknowledgment, 194
Insert binary input, 457
Installation
  STEP 7 Safety, 31
Installing an image file, 375
Instance DB, 125, 150
Instructions
  for the Safety program, 118
  Testing for acceptance, 416
  Get status bit OV, 611
IPAR_EN, 168
IPAR_OK, 170

J

JMP, 588
JMPN, 590
Jump
  If RLO = 1, 588
  If RLO = 0, 590
Jump label, 592

L

LABEL, 592
LAD element
  Inserting, 453
Life cycle of fail-safe automation systems, 638
Light curtain, 495
Local data, 124

M

Main safety block, 117, 150
Math functions
  Add, 553
  Subtract, 556
  Multiply, 559
  Divide, 562
  Create twos complement, 565
  Form absolute value, 568
Maximum cycle time, 630, 634
Memory reset, 397
Migration
  F-CPU, 31
  Printout, 390
Modifying, 391
  Safety program, 397
  Safety program, 397
Monitoring, 391
  Safety program, 397
  Safety program, 397
  Two-hand monitoring, 490
  Two-hand monitoring, 492

Monitoring time, 629
F-cycle time, 632
Communication between I-salve and slave, 632
Communication between F-CPU and F-I/O, 632
Communication between F-CPUs, 633
Move
Move value, 570
Read F-array, 571
Read F-array, 573
Write value indirectly to an F-DB, 575
MOVE, 570
Move operations
Read value indirectly from an F-DB, 577
MUL, 559
Multiply, 559
MUT_P, 504
MUTING, 495
Structure of DIAG, 495
Muting operation
With reflection light barriers, 495
With 4 muting sensors, 495

N

N, 468, 483
N_TRIG, 471, 485
NEG, 565
Network
Inserting, 452
Inserting, 455
Normally closed contact, 460
Normally open contact, 459
NOT, 460

O

Off delay, 537
Offline-online comparison of safety programs, 387
On delay, 535
Operand
Scan for positive signal edge, 467
Scan for negative signal edge, 468
Scan for positive signal edge, 481
Scan for negative signal edge, 483
Operand area
For safety program, 121
Operating principle
RCVDP, 612
SENDDP, 612
SENDS7, 622
RCVS7, 622
Operating system update, 444
Operational safety of the system, 11
OPN, 595
Option handling, 57
OR, 473, 597, 597
Output
Reset, 462
Set, 463
Reset, 476
Set, 478
OV, 609, 610, 611

P

P, 467, 481
P_TRIG, 470, 484
Parameters, 37
Safety-related, 37
Parameter types, 120
PASS_ON, 166
PASS_OUT, 169
Passivation
Channel-granular, 54
Output of fail-safe values, 162
F-I/O, 172
Passivation of F-I/O, 172
After startup, 173
After communication errors, 175
After F-I/O or channel faults, 177
Group passivation, 181
Password, 100
Safety program, 100
F-CPU, 105
F-CPU, 108
Performance level, 23
Plausibility check, 200
Data transfer from standard program to safety program, 431

PLC data type
F-compliant, 126
PLCSIM, 391, 397
PN/PN coupler, 205, 251
Process image, 55, 157, 198
Process safety time, 635
Productive operation, 99
PROFIBUS DP, 24
PROFINET IO, 24
PROFIsafe address
Assigning, 40
Recommendations, 63
Assigning, 73
assign, 74
assign, 77
Changing, 79
PROFIsafe address type, 35
PROFIsafe address type 1, 43
PROFIsafe address type 2, 43
PROFIsafe address type 3, 43
PROFIsafe destination address, 43
Program control operations
Jump if RLO = 1, 588
Jump if RLO = 0, 590
Jump label, 592
Return, 594
Open global data block, 595
Programming
Overview, 111
Startup protection, 156
Group passivation, 181
Plausibility checks, 200
Programming an F-Communication DB, 238
Projects
upgrading, 33
Proof test, 444
Protection level of the F-CPU, 46
Pulse
Generate, 532

Q

QBAD, 169, 172, 172
QBAD_I_xx, 169
QBAD_O_xx, 169

R

R, 462, 476
RCVDP, 209
Receiving data, 612
Timing diagrams, 612
Structure of DIAG, 612
Behavior in the event of communication errors, 612
RCVS7, 238, 393, 622
RD_ARRAY_DI, 571, 573
RD_FDB, 577
Recommendations
PROFIsafe address, 63
Reflection light barriers, 495
Reintegration of F-I/O, 163
After startup of F-system, 173
After communication errors, 175
After F-I/O or channel faults, 177
with group passivation, 181
Programming a user acknowledgment, 190
Programming a user acknowledgment, 194
Replacing
Software components, 444
Response time of F-system, 629, 635
Restart inhibit
MUTING, 495
On interruption of the light curtain, 495
On interruption of the light curtain, 504
MUT_P, 504
Restart protection, 156
RET, 594
Return, 594
RLO
Invert, 458
Invert, 460
Scan for positive signal edge, 470
Scan for negative signal edge, 471
Scan for positive signal edge, 484
Scan for negative signal edge, 485
RS, 466, 480

Rules
Address assignment, 68
Address assignment, 69
Address assignment, 71
for testing, 397
RUN, 404
RUN mode, 404

S

S, 463, 478
S7 connection
Safety-related communication, 237
S7-PLCSIM, 391
testing with, 399
Safety Administration Editor, 80
SafetyConsumerTimeout, 634
Safety function, 23
Example, 24
Calculation of response time, 635
Safety functions
ESTOP1: Emergency STOP up to Stop Category 1, 486
TWO_HAND: Two-hand monitoring, 490
TWO_H_EN: Two-hand monitoring with enable, 492
MUTING: Muting, 495
MUT_P: Parallel muting, 504
EV1oo2DI: 1oo2 evaluation with discrepancy analysis, 515
FDBACK: Feedback monitoring, 520
SFDOOR: Safety door monitoring, 525
ACK_GL: Global acknowledgment of all F-I/O in an F-runtime group, 530
Safety integrity level, 23
Safety mode
Disabling, 392
Disabling, 393
Enabling, 396
Safety program, 24
Automatic generation, 46
Password, 100
Structuring, 111
Structuring, 113
Instructions, 118
Data types, 120
Deleting, 149
Output of fail-safe values, 162
Work memory requirement, 349
Downloading, 350
Comparing, 386
Function test, 391
Monitoring, 391
Modifying, 391
Testing, 397
Modifying, 397
Monitoring, 397
Modifying, 397
Monitoring, 397
Code review, 414
Online consistency, 430
Safety-related communication between F-runtime groups, 143
Safety-related communication via S7 connections
Configuring, 237
Data transfer limits, 242
Safety-related CPU-CPU communication, 35
Options, 35
F-communication DB, 238
Restrictions in RUN mode, 404
Checking for correctness, 424
SENDDP, 612
RCVDP, 612
Safety-related IO controller-I-device communication
Configuring, 220
Programming, 223
Data transfer limits, 225
Configuring, 267
Programming, 272
Data transfer limits, 274
Configuring, 284

Safety-related IO controller-IO controller communication
Configuring, 205
Programming, 209
Data transfer limits, 212
Configuring, 251
Programming, 255
Data transfer limits, 259
Safety-related IO controller-I-slave communication, 236, 281
Safety-related I-slave-I-slave communication
Programming, 228
Data transfer limits, 230
Configuring, 231
Safety-related I-slave-slave communication
Configuring, 183
Data transfer limits, 189
Safety-related master-I-slave communication
Configuring, 226
Programming, 228
Data transfer limits, 230
Configuring, 275
Programming, 278
Data transfer limits, 280
Safety-related master-master communication
Configuring, 212
Programming, 217
Data transfer limits, 220
Configuring, 259
Programming, 263
Data transfer limits, 266
Safety-related parameters, 37
Safety requirements, 11, 23
Safety summary, 79, 414
Scale
Values, 584
Values, 586
SCALE, 584
SCALE_D, 586
SENDDP, 209
Structure of DIAG, 612
Sending data, 612
Timing diagrams, 612
Behavior in the event of communication errors, 612
Sending and receiving data via S7 connections, 622
SENDS7, 238, 393, 622
SFDOOR, 525
Shared device
Configuring, 61
cross-project, 61
Shift and rotate
Shift right, 600
Shift left, 602
Shift left, 602
Shift right, 600
SHL, 602
SHR, 600
SIL, 23
SIMATIC Safety, 4
Product overview, 23
Principles of safety functions, 23
Hardware and software components, 24
Configuring and programming software, 24
Safety program, 24
Simulation, 391
Simulation devices in the F-system, 441
Slice access, 123
Software components, 24, 444
Software requirements, 30
SR, 464, 479
Startup, 156, 173
Startup behavior
MUT_P, 504
RCVDP, 612
SENDDP, 612
SENDS7, 622
RCVS7, 622
Startup protection, 156
Status bit OV
Get, 609
Get negated, 610
Get, 611
STEP 7 Safety, 4
Additional support, 4
Service & Support, 4
Basic knowledge, required, 4
Documentation, 5
Information landscape, 5
Writing conventions, 9
STOP, 441
STP, 441
Structure of the safety program, 111, 113

SUB, 556
Subtract, 556
Supported configurations, 64

T

Tag
F-I/O DB, 165
Monitoring/modifying, 397
Testing of safety program, 397
TIMEOUT, 630, 633
Timer operations
Generate pulse, 532
Generate on-delay, 535
Generate off-delay, 537
Timing diagrams, 495
RCVDP, 612
SENDDP, 612
TOF, 537
TON, 535
TP, 532
Training center, 4
Transfer area, 250
Truth table
AND, 473
OR, 474
EXCLUSIVE OR, 475
TÜV certificate, 416
TWO_H_EN, 492
TWO_HAND, 490

U

UMAC, 102
Uninstallation
STEP 7 Safety, 31
Upgrading
Projects, 33
User acknowledgment, 190
operator control and monitoring system, 191
operator control and monitoring system, 192
Example, 194
User safety function, 4, 24

V

V2-MODE, 39
Value
Move, 570
Write indirectly to an F-DB, 575
Read indirectly from an F-DB, 577
Convert, 578
Scale, 584
Scale to data type DINT, 586
Value status, 159, 169

W

W_BO, 582
Watch table, 398
Wiring test, 397
Word logic operations
AND, 596
OR, 597
EXCLUSIVE OR, 598
Work memory requirement of safety program, 349
WR_FDB, 575

X

X, 474
XOR, 598