SIEMENS

SIMATIC

Industrial Software
S7 F/FH Systems - Configuring and Programming

Programming and Operating Manual

05/2009
A5E00085588-06

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
with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

CAUTION
without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

NOTICE
indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

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 device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Proper use of Siemens products

Note the following:

⚠WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. 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 adhered to. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. 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.

이 기기는 업무용(A급) 전자파 적합기기로서 판매자 또는 사용자는 이 점을 주의하시기 바라며 가정 외의 지역에서 사용하는 것을 목적으로 합니다.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E00085588-06
℗ 05/2009

Preface

Purpose of this documentation

The information in this manual enables you to configure and program fail-safe S7 F/FH Systems using *S7 F Systems* V6.1.

As a supplement, you need the " Safety Engineering in SIMATIC S7 (http://support.automation.siemens.com/WW/view/en/12490443) " system manual.

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
- S7-400 Automation Systems
- Distributed I/O systems on PROFIBUS DP
- *STEP 7* basic software, particularly:
  - Working with *SIMATIC Manager*
  - Hardware configuration with *HW Config*
  - Communication between CPUs
  - *CFC* optional software

Scope of this documentation

	Order number	Release number and higher
S7 F Systems optional package V6.1 including authorization license V6.1	• Full version: 6ES7833-1CC02-0YA5 • Upgrade version from V5.x/V6.0: 6ES7833-1CC01-0YE5	V6.1
S7 F Systems RT Licence (Copy Licence)	• 6ES7833-1CC00-6YX0	V5.0

The *S7 F Systems* optional package is used for configuring and programming S7 F/FH Systems. Integration of the F-I/Os listed below in S7 F/FH Systems is also addressed:

- ET 200S fail-safe modules
- ET 200eco fail-safe I/O modules
- ET 200pro fail-safe modules
- S7-300 fail-safe signal modules
- Fail-safe DP standard slaves
- Fail-safe PA field devices

What's New?

The innovations in *S7 F Systems* V6.1 are described below:

- New functionality
  - Maintenance Override
- Revised safety program comparison function
- Expanded access protection
- F-Forcing support
- New F-Blocks in the F-Library:
  - F_CH_II: F-Channel driver for inputs of data type INT of fail-safe DP standard slaves and fail-safe standard I/O devices
  - F_CH_IO: F-Channel driver for outputs of data type INT of fail-safe DP standard slaves and fail-safe standard I/O devices
  - F_CH_DII: F-Channel driver for inputs of data type DINT of fail-safe DP standard slaves and fail-safe standard I/O devices
  - F_CH_DIO: F-Channel driver for outputs of data type DINT of fail-safe DP standard slaves and fail-safe standard I/O devices
  - F_FDI_FR: Conversion from F_DINT to F_REAL
  - F_FR_FDI: Conversion from F_REAL to F_DINT
  - F_POLYG: F-Control block with non-linear characteristic
  - F_INT_P: Integration function with integration mode and track mode
  - F_PT1_P: First order delay
  - F_DEADTM: Monitoring of changes in F_REAL values at the same measuring point
  - F_SWC_P: Centralized control of operator input via the OS
  - F_SWC_BO: Processing of a parameter of data type F_BOOL for operator input via the OS
  - F_SWC_R: Processing of a parameter of data type F_REAL for operator input via the OS
- New block in the F-Library:
  - SWC_MOS: Command function for Maintenance Override
  - FORCEOFF: Deactivation of F-Force

Approvals

S7 F/FH Systems and the F-I/O are certified for use in safety mode for:

- Safety Integrity Level SIL1 to SIL3 in accordance with IEC 61508
- Category 1 to 4 in accordance with EN 954-1

Position in the information landscape

Depending on your application, you will need the following supplementary documentation when working with *S7 F/FH Systems*.

This documentation includes references to the supplementary documentation where appropriate.

Documentation for	Brief Description of Relevant Contents
Safety Engineering in SIMATIC S7	The " Safety Engineering in SIMATIC S7 (http://support.automation.siemens.com/WW/view/en/12490443) " system manual provides an informational overview of the use, installation, and mode of operation of the S7 Distributed Safety and S7 F/FH Systems fail-safe automation systems, and describes basic properties and detailed technical information about these F-Systems.
S7 F/FH Systems	• The "Automation System S7-400 Hardware and Installation (http://support.automation.siemens.com/WW/view/en/1117849)" installation manual describes the assembly and wiring of S7-400 systems. • The " Automation System S7-400H Fault-Tolerant Systems (http://support.automation.siemens.com/WW/view/en/1186523) " manual describes the CPU 41x-H central modules and the tasks required to set up and commission an S7-400H fault-tolerant system.
S7 Distributed Safety	The following elements are described in the " S7 Distributed Safety - Configuring and Programming (http://support.automation.siemens.com/WW/view/en/22099875) " operating manual and online help: • Configuration of the F-CPU and the F-I/O • Programming of the F-CPU in F-FBD or F-LAD
S7-300 Automation System	The " Automation System S7-300 Fail-Safe Signal Modules (http://support.automation.siemens.com/WW/view/en/19026151) " manual describes the hardware of the S7-300 fail-safe signal modules (including installation, wiring, and technical specifications).
ET 200S Distributed I/O System	The " Distributed I/O System Fail-Safe Engineering ET 200S Distributed I/O System - Fail-Safe Modules (http://support.automation.siemens.com/WW/view/en/12490437) " operating instructions describe the hardware of the ET 200S fail-safe modules (including installation, wiring, and technical specifications).
ET 200pro Distributed I/O System	The " ET 200pro Distributed I/O Device - Fail-Safe Modules (http://support.automation.siemens.com/WW/view/en/22098524) " operating instructions describe the hardware of the ET 200pro fail-safe modules (including installation, wiring, and technical specifications).
ET 200eco Distributed I/O System	The " ET 200eco Distributed I/O Station Fail-Safe I/O Module (http://support.automation.siemens.com/WW/view/en/22099642) " manual describes the hardware of the ET 200eco fail-safe module (including installation, wiring, and technical specifications).

Documentation for	Brief Description of Relevant Contents
STEP 7 manuals	• The Configuring Hardware and Communication Connections with STEP 7 V5.4 (http://support.automation.siemens.com/WW/view/en/18652631) manual describes the operation of the relevant standard tools of *STEP 7*. • The System Software for S7-300/400 System and Standard Functions (http://support.automation.siemens.com/WW/view/en/1214574) reference manual describes functions for distributed I/O access and diagnostics for distributed I/O /CPU. • The " Programming with STEP 7 V 5.4.4 (http://support.automation.siemens.com/WW/view/en/18652056) " manual describes the procedure for programming with *STEP 7*. • The " CFC for S7 Continuous Function Chart (http://support.automation.siemens.com/WW/view/en/21401430) " manual/online help provides a description of programming with *CFC*. • " Modifying the System during Operation via CiR (http://support.automation.siemens.com/WW/view/en/14044916) " manual
STEP 7 Online Help	• Describes how to operate the standard tools of *STEP 7*. • Contains information on configuring and assigning parameters for I/Os with *HW Config*.
PCS 7	• The " PCS 7 manuals (http://support.automation.siemens.com/WW/view/en/10806846/133300) "describe operation of the *PCS 7* process control system (necessary when the F-System is integrated into a higher-level control system).

Guide

This documentation describes how to work with the *S7 F Systems* optional package. It includes both instructional material and reference material (description of fail-library blocks).

The following topics are addressed:

- Configuration of *S7 F Systems*
- Access protection for *S7 F Systems*
- Programming of the safety program (safety-related user program)
- Safety-related communication
- Support for the system acceptance test
- Operation and maintenance of *S7 F Systems*
- F-Libraries

Conventions

In this documentation, the terms "safety engineering" and "fail-safe engineering" are used synonymously. The same applies to the terms "fail-safe" and "F-".

When *S7 F Systems* appears in italics, it refers to the optional package for the "S7 F/FH Systems" fail-safe system.

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 user program is referred to as the "standard user program".

"F-CPU" denotes a CPU with fail-safe capability. A CPU with fail-safe capability is a central processing unit that is approved for use in S7 F/FH Systems and S7 Distributed Safety.

Additional support

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

You will find information on who to contact on the Web (http://www.siemens.com/automation/partner).

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 will find the online catalog and online ordering system on the Web (http://mall.automation.siemens.com).

Training center

We offer courses to help you get started with the *SIMATIC S7* automation system. Contact your regional training center or the central training center in D -90327 Nuremberg, Federal Republic of Germany.

You will find more information on the Web (http://www.sitrain.com).

H/F Competence Center

The H/F Competence Center in Nuremberg offers special workshops on *SIMATIC S7* fail-safe and fault-tolerant automation systems. The H/F Competence Center can also provide assistance with onsite configuration, commissioning and troubleshooting.

For questions about workshops, etc., contact: hf-cc.aud@siemens.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).

You can find additional information about our Technical Support on the Web (http://www.siemens.com/automation/service).

Service & Support on the Internet

In addition to our paper documentation, our complete knowledge base is available to you on the Web (http://www.siemens.com/automation/service&support).

There, you will find the following information:

- Newsletters providing the latest information on your products
- A search engine in Service & Support for locating the documents you need
- A forum where users and experts from all over the world exchange ideas
- Your local contact partner for Industry Automation products in our Contact Partners database
- Information about on-site service, repairs, spare parts, and much more under "Repairs, spare parts, and consulting"

Important note for maintaining the operational safety of your system

Note

Systems with safety-related characteristics are subject to special operational safety requirements on the part of the operator. The supplier is also obliged to comply with special product monitoring measures. For this reason, we publish a special newsletter containing information on product developments and features that are (or could be) relevant to operation of safety-related systems. By subscribing to the relevant newsletter, you will always have the latest information and able to make changes to your system, when necessary. Just visit us on the Web (https://www.automation.siemens.com/WW/newsletter/guiThemes2Select.aspx?HTTPS=REDIR&subjectID=2).

There, you can register for the following newsletters:

- S7-300/S7-300F
- S7-400/S7-400H/S7-400F/FH
- Distributed I/O
- SIMATIC Industrial Software

To receive these newsletters, select the check box "Update".

Warnings index

Warning	Section
Section: Product Overview	
S7 F/FH Systems operation	1.2
Section: Installation	
Possible change in response time due to migration from Failsafe Blocks 1_2 to *S7 F Systems Lib* V1_3 SP1	2.3
Section: Configuration	
An F-CPU containing a safety program must have a password	3.3
Configuring a protection level	
Group diagnostics for fail-safe S7-300 signal modules	3.4
Rule for PROFIBUS subnets	
Section: Access protection	
Limiting access using the ES	4.2
Transferring the safety program to multiple F-CPUs	
Password protection	
Limiting access using the ES	4.3
Passwords must be unique	
Section: Programming	
Default setting of the maximum MAX_CYC	5.2.3
Do not change values created during compilation	5.2.4
The call interval of cyclic interrupt OB 3x is monitored for the maximum value	
Compression changes the signature	5.2.5
Optimization of the runtime sequence in the CFC	5.2.7
Entries for F-Blocks in the symbol table must not be changed	5.3.1
Illegal changes to input parameters of F-Blocks can cause a shutdown of the safety program and its outputs	5.3.2

Warning	Section
Do not change automatically inserted F-Control blocks	5.4
Saved error information is lost during an F-Startup	5.5
Outputs of F-Blocks always use the predefined initial values	5.7.2
Validity check	5.9.2
The two acknowledgement steps must not be triggered with a single operation	5.10
If your OS can access multiple F-CPUs	
Section: F-I/O access	
For F-I/O with inputs, the fail-safe value 0 provided at the F-channel driver must be further processed for (digital) channels of data type BOOL in the safety program	6.3
Section: Programming communication	
Safety-related CPU-CPU communication is not permitted via public networks.	7.1.1
The value for each address association	7.1.3
It can be ensured (from a fail-safe standpoint) that a signal level to be transferred will be detected on the sender side and transferred to the receiver only if the signal is pending for at least as long as the assigned F-Monitoring time (TIMEOUT).	
If the F-CPU with the associated F_SENDBO/F_SDS_BO/F_SENDR is in deactivated safety mode, you can no longer assume that the data received from this F-CPU were generated safely.	
The S7 program must be recompiled if the S7 connections for communication between F-CPUs have been changed.	
Section: Maintenance Override function	
The "Maintenance Override" functionality allows changes to the safety program to be made during RUN mode.	8.2.2.1
Warnings in the descriptions of F-Blocks	8.2.2.2 to 8.2.2.5
You can edit the faceplates for Maintenance Override.	8.2.3
Initiator and confirmer must not accept an invalid value	8.3.1
Technological assignment must be appropriate for the environment	
Transaction for changing an F-Parameter	
Section: Safety Data Write function	
Warnings in the descriptions of F-Blocks	9.2.2
Static values of the SAFE_ID1 and SAFE_ID2 attributes	9.2.4
Initiator and confirmer must not accept an invalid value	9.3.1
Technological assignment must be appropriate for the environment	
Transaction for changing an F-Parameter	
Section: Compiling and commissioning an S7 program	
Deactivating safety mode	10.5.1
Do not copy F-Blocks with *SIMATIC Manager*	10.6
Safety program on a memory card	10.6.1
If multiple F-CPUs can be accessed from an ES via a network (e.g., MPI)	
Shutdown of the safety program following a change to the fail-safe outputs	10.7
A simulation is no substitute for a function test.	10.7.1
Changing the collective signature for changes in CFC test mode	10.8.1
Do not change values created during compilation	
Download operation aborted	10.8.2
Moving F-Blocks or F-Runtime groups	
Modifying the safety program in RUN mode	

Warning	Section
Section: System acceptance test	
Rule for PROFIBUS subnets	11.2.1
Section: Operation and maintenance	
If you operate simulation devices or simulation programs	12.1
Switching from STOP to RUN from the ES	
STOP status initiated with SFC 46 "STP"	
Two F-CPUs not simultaneously as master system	
Forcing is only permitted when the safety of the system is ensured by other measures.	12.3
Section: F-Libraries	
Values of PAR_ID and COMPLEM must not be changed	A.1.1
Value for the relevant adress reference	A.2.2.1 to A.2.2.2
Measure and transfer signal level	
User acknowledgement is always required for communication errors	A.2.2.2
Value for the relevant address reference	A.2.2.3 to A.2.2.4
Measure and transfer signal level	
User acknowledgement is always required for communication errors	A.2.2.4
Value for the relevant address reference	A.2.2.5 to A.2.2.6
Measure and transfer signal level	
User acknowledgement is always required for communication errors	A.2.2.6
Fail-safe user times	A.2.4.1 to A.2.4.4
Validity check	A.2.5
The "Maintenance Override" functionality allows changes to the safety program to be made during RUN mode.	A.2.5.1 to A.2.5.2
Interconnection of the CS_VAL input is not permitted.	A.2.5.2
F-Startup	
The "Maintenance Override" functionality allows changes to the safety program to be made during RUN mode.	A.2.5.3
The CS_VAL, MIN, and MAX inputs must not be interconnected.	
F-Startup	
Reintegration through user acknowledgement with F_QUITES	A.2.5.8
The "Safety Data Write" functionality makes changes in the safety program during RUN mode	A.2.5.13
The CHANGED output cannot be evaluated in the safety program	
The MIN, MAX, and MAXDELTA inputs must not be interconnected	
Parameters SAFE_ID1 and SAFE_ID2	
F-Startup	
The "Safety Data Write" functionality makes changes in the safety program during RUN mode	A.2.5.14
The CHANGED output cannot be evaluated in the safety program	
Parameters SAFE_ID1 and SAFE_ID2	
F-Startup	
Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.	A.2.6.1 to A.2.6.2
Startup protection for short-term power failure of the fail-safe DP standard slaves	
Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.	A.2.6.3 to A.2.6.4
Startup protection for short-term power failure of the fail-safe PA field device	

Warning	Section
Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.	A.2.6.5 to A.2.6.7
Startup protection for short-term power failure of the F-I/O	
Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.	A.2.6.8 to A.2.6.11
Startup protection for short-term power failure of the fail-safe DP standard slaves	
Fail-safe user times	A.2.9.2 to A.2.9.4
Fail-safe user times	A.2.10.1 to A.2.10.2
F-Startup	A.2.13.1
Safety note - do not change automatically inserted F-Control blocks	A.3 to A.3.3
Default setting of the maximum MAX_CYC	A.3.3
Safety note - do not change automatically inserted F-Control blocks	A.3.4 to A.3.18

Table of contents

Product Overview

1

1.1 Overview

S7 F/FH Systems fail-safe systems

The fail-safe automation systems ("F-Systems") S7 F/FH Systems are used in systems with stringent safety requirements. The objective of S7 F/FH Systems is to control processes with an immediately achievable safe state. In other words, F-Systems control processes in which an immediate shutdown does not endanger people or the environment.

The *S7 F Systems* optional package comprises the following two components:

- *S7 F Configuration Pack* V5.5 SP6
- *S7 F Systems* V6.1

Achievable safety requirements

With S7 F/FH Systems, you achieve the following safety requirements:

- Safety Integrity Level SIL1 to SIL3 in accordance with IEC 61508
- Category 1 to 4 in accordance with EN 954-1

The principle of safety functions in S7 F/FH Systems

Functional safety is implemented principally through safety functions in the software. Safety functions are performed by S7 F/FH Systems whenever a dangerous event occurs:

- To place the system in a safe state

  or

- To keep the system in a safe state

Safety functions are contained mainly in the following components:

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

The F-I/O ensures safe processing of field information (such as temperature and level monitoring). They have all of the required hardware and software components for safe processing, in accordance with the required safety class. You only have to program the user safety function. The safety function for the process can be provided through a user safety function or a fault reaction function. In the event of a fault, if the F-system can no longer execute its actual user safety function, it executes the fault reaction function. For more information, refer to the section entitled "F-STOP (Page 84)".

Example of user safety functions and fault reaction functions

In the event of overpressure, the F-system opens a valve (user safety function). In the event of a dangerous fault in the F-CPU, all outputs are switched off (fault reaction function). The valve is opened and the other actuators also achieve a safe state. If the F-system is intact, only the valve would be opened.

Fail-safety and availability

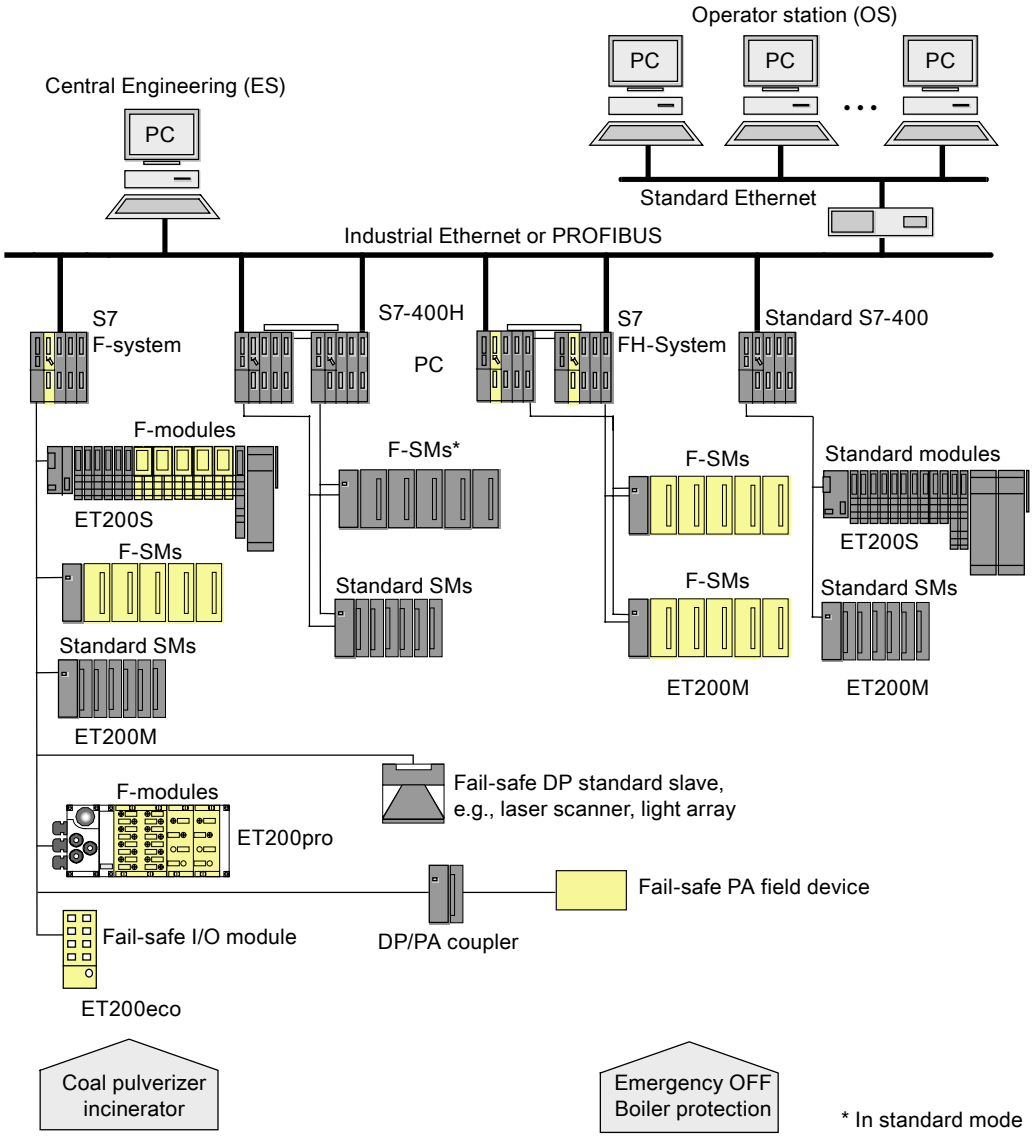
To increase availability of the automation system and, thus, to prevent process failures due to faults in the F-System, you can optionally equip fail-safe systems with a fault-tolerant feature. You achieve this increased availability through component redundancy:

- Power supply
- Central processing unit
- Communication
- F-I/O

With fail-safe, high-availability S7 F/FH Systems, you can resume production without harming people or the environment.

Use in process engineering

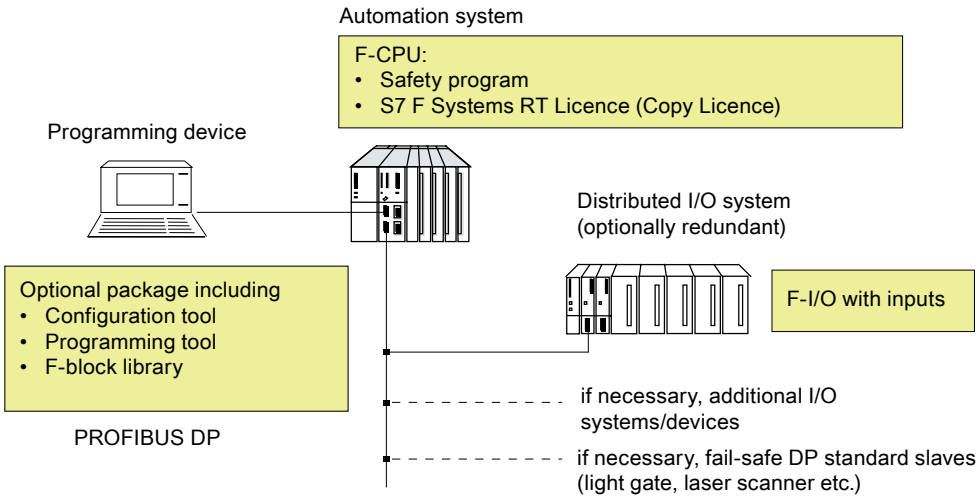
The figure below shows you the possible ways of integrating S7 F/FH Systems into you process automation system with *PCS 7*.



1.2 Hardware and software components

Hardware and software components of S7 F/FH Systems

The figure below provides an overview of the hardware and software components you need to configure and operate S7 F/FH Systems.



Hardware components

The hardware components of S7 F/FH Systems include the following:

- F-CPU (CPU 412-3H, CPU 414-4H, or CPU 417-4H)
- Fail-safe inputs and outputs (F-I/O), such as:
  - S7-300 fail-safe signal modules in ET 200M (distributed configuration)
  - Fail-safe power and electronic modules in ET 200S
  - ET 200eco fail-safe I/O module
  - ET 200pro fail-safe modules
  - Fail-safe DP standard slaves
  - Fail-safe PA field devices

You can expand the configuration with standard I/O.

Software components

⚠ WARNING
S7 F/FH Systems operation
You may only operate S7 F/FH Systems in the released system environments.
Operation on terminal server/clients or on a virtual server/system is expressly excluded.

The software components of S7 F/FH Systems include the following:

- *S7 F Systems* optional package on the ES for configuring and programming the F-system
- Safety program in the F-CPU

You also need the *STEP 7* basic software and the *CFC* optional software on the ES for configuring and programming.

S7 F Systems optional package

This documentation describes *S7 F Systems*. *S7 F Systems* is the configuration and programming software for S7 F/FH Systems. With *S7 F Systems,* you receive the following:

- Support for configuring the F-I/O in *STEP 7* with *HW Config*
- Support for creating the safety program and integrating fault detection functions into the safety program
- F-Library containing F-Blocks that you can use in your safety program
- Moreover, *S7 F Systems* offers functions for comparing safety programs and for assisting you with the system acceptance test.
- Support for operating fail-safe parameters of a *PCS 7* OS during operation (Safety Data Write).
- Support for safety-related modification of F-Parameters in the safety program of an F-CPU from a *PCS 7*-OS (Maintenance Override).
- Support during operation and maintenance with F-Forcing.

Safety program

You create a safety program with the *CFC Editor* in *STEP 7* from the F-Blocks that are provided in an F-Library with the *S7 F Systems* optional package.

Safety checks are automatically performed and additional F-Blocks for error detection and fault reaction are inserted when you compile the safety program. This ensures that failures and faults are detected, and appropriate responses are initiated. This maintains the F-System in a safe state or places it in a safe state.

In the CPU module, the S7 program consists of fail-safe components (safety program) and non-fail-safe components (standard user program).

Data can be exchanged between the safety user program and the standard user program in the F-CPU with special F-Blocks for data conversion.

Installation

2

2.1 Installing the S7 F Systems optional package V6.1

Software requirements

You must install the following software packages in order to operate *S7 F Systems* V6.1:

- On the ES
  - *STEP 7* V5.3 HF4 or later
  - *CFC* V6.1 SP2 HF10 or later
  - Optional: *PCS 7* V6.1 SP2 or later
- On the OS (for S7 F Systems HMI)
  - *PCS 7* V6.1 SP2 or later
- For off-line testing
  - *S7 PLCSIM* V5.4

Available installation units

S7 F Systems comprises the following installation units:

- *S7 F Systems* V6.1
- *S7 F Systems HMI* V6.1
- *S7 F Systems Lib* V1_3 SP1
- *S7 F Configuration Pack* V5.5 SP6
- *Automation License Manager* V4.0 SP2

The correct version of the *S7 F Configuration Pack* will be installed based on whether you have installed *PCS 7* and which version you have installed. For more information, read the installation notes in Section 3 of the "S7 F ConfigurationPack - readme" file for *S7 F Configuration Pack* V5.5 SP6.

Reading readme files

You will find important information about the supplied software in the readme files "S7 F Systems - readme", "S7 F Configuration Pack - readme," and "S7 F Systems HMI - readme". You can display these files at the end of the corresponding setup program. At a later point, you can open the readme file by selecting **Start > SIMATIC > Product Notes > English**.

Installing *S7 F Systems*

1. Start your ES/workstation. Ensure that no *STEP 7* applications are open.
2. Insert the optional package product CD.
3. Initiate the SETUP.EXE program on the CD.
4. Follow the setup program instructions.

Starting *S7 F Systems*

The *S7 F Systems* optional package does not contain any applications that you must start specifically. Support for the configuration and programming of F-Systems is integrated into:

- *SIMATIC Manager*
- *HW Config*
- *CFC Editor*
- *PCS 7* OS

Displaying integrated Help

Context-sensitive Help is provided for the dialogs of the optional package. You can access this Help at every stage of configuring and programming using the F1 key or the "Help" button. For advanced help, select **Help > Contents > Calling Help on Optional Packages > S7 F/FH Systems - Working with F Systems**.

License key (usage authorization)

A license key is required for the *S7 F Systems* optional package. This license key is installed in the same way as for *STEP 7* and the optional packages. For information on installing and working with license keys, refer to the readme file and the *STEP 7* basic Help.

S7 F Systems RT License (Copy License)

The S7 F Systems RT license (copy license) allows you to use a CPU as an F-CPU (for example, to run a safety program on it).

2.2 Removing the S7 F Systems optional package V6.1

Removing *S7 F Systems*

The *S7 F Systems* optional package comprises the following components:

- *S7 F Configuration Pack* V5.5 SP6
- *S7 F Systems* V6.1
- *S7 F Systems Lib* V1_3 SP1
- *S7 F Systems HMI* V6.1

These components can be individually removed. Use the normal procedure in Windows for removing software:

1. In Windows, double-click the "Add/Remove Programs" icon in "Control Panel" to open the dialog box for installing software.
2. Select the appropriate entry in the list of installed software. Click "Remove" to remove the software.

2.3 Migrating to S7 F Systems V6.1

Introduction

Before migrating from an existing project to *S7 F Systems* V6.1, read through the following section carefully. This section contains the following important information for you:

- Basic information on migrating to *S7 F Systems* V6.1
- Possible consequences of migrating to *S7 F Systems* V6.1
- Use cases for migrating to *S7 F Systems* V6.1

Migration to *S7 F Systems* V6.1 with *S7 F Systems Lib* V1_3 SP1 offers you the following advantages:

- Support of additional fail-safe DP standard slaves
- New F-Blocks

Note

S7 F Systems V6.1 supports more F-I/O than *PCS 7*. Refer to the *PCS 7* documentation, if necessary.

With this F-I/O, only processing with *S7 F Systems* (but not the diagnostic functionality of *PCS 7*) is generated during compilation. For this reason, the message "This module is not supported" appears in the "Module Driver" tab during compilation.

Note

You can use fail-safe PA field devices with the F-Library *Failsafe Blocks* (V1_2) SP4 or *S7 F Systems Lib* V1_3 or later. If you want to use the blocks of the *S7 F Systems Lib* V1_3 SP1, you must keep the following in mind:

- You must first install at least *PDM* V6.0 SP2 with EDD of SITRANS DSIII PROFIsafe, V01.02.01-54 or later.
- You must at least first install *STEP 7* V5.4 SP2.

If these requirements are not met, you must continue to use the F-Library *Failsafe Blocks* (V1_2).

Note

Safety Matrix up to V5.2 HF 1 cannot be operated with *S7 F Systems Lib* V1_3 SP1. If you are using this, you must continue to use the F-Library *Failsafe Blocks* (V1_2).

Migrating to *S7 F Systems* V6.1

Note

Proceed according to the following use cases when performing the migration. Do not use the "Update Block Types" function, even with multiprojects. To update a multiproject master data library, follow the procedure described in the section entitled " Updating a multiproject master data library (Page 45) ".

Before upgrading a specific project to *S7 F Systems* V6.1, you must decide on one of two variants:

Variant	Consequences	
	Advantages	Disadvantages
Without an F-Library update	• No changes to safety program • May require a new acceptance test	• No new F-Blocks for new functionality • No support for new F-I/O released in the future
Migration without an *S7 F Systems Lib* V1_3 update	• No changes to safety program • No new acceptance test required	• No new F-Blocks for new functionality • No support for new F-I/O released in the future
With an F-Library update	• All new functions can be used • F-I/O released in the future will be supported	• Migration causes the safety program to change • The program must be downloaded to the F-CPU by means of a complete download (with STOP)

Migration without an F-Library update

Migration without an update is purely a software update on your ES. The steps you must perform are determined by the version of *S7 F Systems* you have installed on your ES. Select the appropriate case from the table below:

Migration from ...	To *S7 F Systems* V6.1
S7 F Systems V5.1	Use case 1 (Page 33)
S7 F Systems V5.2 without an SP	Use case 3 (Page 38)
S7 F Systems V5.2 SP1 to V5.2 SP4	Use case 5 (Page 42)
S7 F Systems V6.0	Use case 6 (Page 43)

Migration with an F-Library update

The steps you must perform are determined by the F-Library used in your S7 program. Select the appropriate case from the table below:

Migration from ...	To *S7 F Systems Lib* V1_3 SP1
Failsafe Blocks (V1_1)	Use case 2 (Page 34)
Failsafe Blocks (V1_2)	Use case 4 (Page 39)
S7 F Systems Lib V1_3	Use case 7 (Page 44)

Note

Refer also to the sections entitled "Differences between the F-Libraries Failsafe Blocks (V1_x) and S7 F Systems Lib V1_3 (Page 387)" and "Differences between the F-Library S7 F Systems Lib V1_3 and V1_3 SP1 (Page 409)".

⚠ WARNING

Possible change in response time due to migration from Failsafe Blocks 1_2 to *S7 F Systems Lib* V1_3 SP1

Migration to *S7 F Systems Lib* V1_3 SP1 can cause a change in the maximum response time. Use the Excel file S7FTIMEB.XLS to calculate the new maximum response time of your S7 F/FH Systems. For more information, refer to the section entitled "Run times, F-Monitoring times, and response times (Page 410)".

Follow the procedure described in the relevant use case.

You will find a description of the use cases in the sections below.

2.3.1 Use case 1

Objective

Simple software update from *S7 F Systems* V5.1 to *S7 F Systems* V6.1 without program changes.

Introduction

This use case helps you to migrate from *S7 F Systems* V5.1 to *S7 F Systems* V6.1 if you want to retain compatibility with your previous Version 5.1.

Requirements

Your S7 program must be compiled, downloaded and executable for the original *Failsafe Blocks* (1_1) F-Library. Ensure that this is the case by printing out the safety program and performing an online comparison.

Consequences

- No changes to the safety program
- No changes to the collective signature

Procedure

1. Before installing *S7 F Systems* V6.1: Copy the *Failsafe Blocks* (V1_1) F-Library, as it is deleted when *S7 F Systems* V5.1 is removed:
   - Open the *Failsafe Blocks* (V1_1) F-Library
   - In *SIMATIC Manager* , select **Save As**
   - Specify a different name, such as *"Failsafe Blocks* (V1_1)x"
2. Install *S7 F Systems* V6.1.
3. Rename the copied *Failsafe Blocks* (V1_1)x F-Library as *"Failsafe Blocks* (V1_1)"
   - Open the *"Failsafe Blocks* (V1_1)x" F-Library
   - In *SIMATIC Manager* , select **Edit > Rename**
   - Specify the name *"Failsafe Blocks* (V1_1)"
4. Prior to the initial compilation, save the current state of your safety program as a reference ("Save Reference" in the "Safety Program" dialog) so that it will be available for future comparisons
5. You can now recompile your S7 program

2.3.2 Use case 2

Objective

To upgrade your S7 program with *Failsafe Blocks* (V1_1) to *S7 F Systems Lib* V1_3 SP1.

Introduction

This use case helps you when migrating your safety program by upgrading blocks of the *Failsafe Blocks* (V1_1) F-Library to blocks of the *S7 F Systems Lib* V1_3 SP1 F-Library. You can then use the new functions of the *S7 F Systems Lib* V1_3 SP1 F-Library.

When you migrate from the *Failsafe Blocks* (V1_1) F-Library to *S7 F Systems Lib* V1_3 SP1, the F-FBs in your safety program are overwritten by F-Blocks with other block signatures. This means that the collective signature will change.

In Version V5.1 of *S7 F Systems*, you had to place the F_CYC_CO F-Block manually. When migrating to *S7 F Systems Lib* V1_3 SP1, this F-Block is automatically moved to a system runtime group.

The new shutdown logic introduced in *S7 F Systems* V6.1 is automatically created during compilation. The new shutdown logic has interfaces with each F-Runtime group.

Note

Different behavior for safety-related faults

In *S7 F Systems* V6.1 with *S7 F Systems Lib* V1_3 SP1, F-Blocks do not initiate a CPU-STOP when a safety-related fault (in the safety data format, for example) is detected. Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

You can configure the shutdown logic accordingly as:

- Partial shutdown
  Only the affected F-Shutdown group is shut down.
- Full shutdown
  The entire safety program is shut down.

For more information, refer to the sections entitled " F-Shutdown groups (Page 77) " and " F-STOP (Page 84) ".

Note

Different behavior for floating point operations

With the *Failsafe Blocks* (V1_1) F-Library, a CPU-STOP was initiated when a floating point operation resulted in an overflow (± infinity) or a denormalized or invalid (NaN) floating point number, or when an invalid floating point number (NaN) was already present as an address.

Starting with *S7 F Systems Lib* V1_3, these events no longer cause a CPU-STOP. The results "Overflow (± infinity)," "Denormalized floating point number," or "Invalid floating point number (NaN)" are either:

- Either output at the output and available for further processed by the subsequent F-Blocks

  or

- Signaled at special outputs. A fail-safe value is output, if necessary.

If the floating-point operation yields an an invalid floating point number (NaN) without the existence of a previous invalid floating point number (NaN) as an address, the following diagnostic event is recorded in the diagnostic buffer of the F-CPU:

- "Safety program: invalid REAL number in DB" (Event ID 16#75D9)

You can use this diagnostic buffer entry to identify the F-Block with the invalid floating-point number (NaN).

Also refer to the documentation about F-Blocks in Appendix " F-Libraries (Page 193) ".

If you cannot rule out the occurrence of these events in your safety program, you must decide independently of your application whether you must respond to them in your safety program. With the F-Block F_LIM_R, you can check the result of a floating-point operation for overflow (± infinity) and invalid floating-point number (NaN).

Note

With *S7 F Systems* V5.1, a combination of standard blocks (AND, OR, etc.) and F-Blocks in the same F-Runtime group during compilation was not prevented in all cases. This has been improved in *S7 F Systems* V5.2 and later. *S7 F Systems* V5.2 and later always signals an error when you insert standard blocks together with F-Blocks into the same F-Runtime group.

Note

With the *Failsafe Blocks* (V1_1) F-Library, a discrepancy analysis was not performed with redundant F-I/O and "2-channel equivalent" type of sensor interconnection in the F-Module driver, even if a discrepancy time greater than or less than (<>) 0 ms (10 ms by default) was configured in the "Redundancy" tab in *HW Config*.

Starting with *S7 F Systems Lib* V1_3 and *S7 F Configuration Pack* V5.5 SP3, a discrepancy analysis is always performed for a discrepancy time greater than or less than (<>) 0 ms.

If you want to shut down the discrepancy analysis, configure a discrepancy time equal to (=) 0 ms in the "Redundancy" tab in *HW Config*.

Requirements

- CPU 414-4H V3.1 or later or CPU 417-4H V3.1 or later
- If F-Block types are used in your project, you first must recreate these with *S7 F Systems Lib* V1_3 SP1. To do so, follow the procedure outlined in the section entitled " Updating F-Block types that you have created (Page 45) ".

Consequences

- The collective signature is changed
- A complete download with CPU-STOP is required

Procedure

1. Install *S7 F Systems* V6.1 with *S7 F Systems Lib* V1_3 SP1
2. Prior to the initial compilation, save the current state of your safety program as a reference ("Save Reference" in the "Safety Program", dialog) so that it will be available for future comparisons
3. In the Safety Program dialog, select the *S7 F Systems Lib* V1_3 SP1 F-Library

   To do so, click the "Library Version" button in the "Edit Safety Program" dialog
4. In the S7 program, update the existing F-Block types For information about how to do this, refer to the section entitled "Updating F-Block types that you have created (Page 45)"
5. In the *CFC Editor* under **Options > Block Types** , click "Clean Up"
6. Update all block types in the *CFC Editor* by selecting **Options > Block Types** and clicking "New Version"
7. Recompile your hardware configuration
8. Recompile your S7 program

Additional measures for F-Module drivers

When migrating to *S7 F Systems Lib* V1_3 SP1, interconnections of the following F-Module driver outputs may require special handling:

- PROFIsafe1
- PROFIsafe2
- DIAG_1
- DIAG_2

Proceed as follows when migrating to *S7 F Systems Lib* V1_3 SP1:

1. Before performing the migration, document the interconnection of the PROFIsafe1 and DIAG_1 outputs, along with the value of the LADDR input
2. For redundant F-I/O, also document the interconnection of the PROFIsafe2 and DIAG_2 outputs before performing the migration, along with the value of the LADDR_R input
3. Perform the migration to *S7F Systems Lib* V1_3 SP1.
4. Interconnect the documented interconnections at the PROFIsafe1 and DIAG_1 outputs to the new F-Module driver F_PS_12, whose value at the LADDR input matches the documented LADDR
5. For redundant F-I/O, interconnect the documented interconnections at the PROFIsafe2 and DIAG_2 outputs to the new F-Module driver F_PS_12, whose value at the LADDR input matches the documented LADDR_R

Table 2- 1 Non-redundant F-I/O

Failsafe Blocks (V1_1)	*S7 F Systems Lib* V1_3 SP1
Interconnection at the original F-Module driver:	**Interconnection at the F-Module driver F_PS_12:**
PROFIsafe1	PROFIsafe
DIAG_1	DIAG
LADDR	LADDR

Table 2- 2 Redundant F-I/O

Failsafe Blocks (V1_1)	*S7 F Systems Lib* V1_3 SP1
Redundant interconnection at the original F-Module driver:	**Interconnection at the first F-Module driver F_PS_12:**
PROFIsafe1	PROFIsafe
DIAG_1	DIAG
LADDR	LADDR
	Interconnection at the second F-Module driver F_PS_12:
PROFIsafe2	PROFIsafe
DIAG_2	DIAG
LADDR_R	LADDR

Additional measures for redundant fail-safe digital input modules SM 326; DI 8 X NAMUR and SM 326; DI 24 X DC 24 V

With the redundant fail-safe digital input modules SM 326; DI 8 X NAMUR and SM 326; DI 24 X DC 24 V, information about detected discrepancy errors is provided at the DIAG_1 and DIAG_2 outputs of the F_M_DI8 and F_M_DI24 F-Block drivers when the *Failsafe Blocks* (V1_1) F-Library is used.

Starting with *S7 F Systems Lib* V1_3 SP1, discrepancy error information is output at the DISCF and DISCF_R outputs of the F-Channel driver F_CH_DI.

If you are using logic that evaluates this information, modify it accordingly.

2.3.3 Use case 3

Objective

Simple software update from *S7 F Systems* V5.2 (no SP) to *S7 F Systems* V6.1 without program changes.

Introduction

This use case helps you to migrate from *S7 F Systems* V5.2 without an SP to *S7 F Systems* V6.1 if you want to retain compatibility with your previous Version 5.2.

Requirements

Your S7 program must be compiled, downloaded and executable for the original *Failsafe Blocks* (V1_2) or *Failsafe Blocks* (V1_1) F-Library. Ensure that this is the case by printing out the safety program and performing an online comparison.

Consequences

- No changes to safety program
- No changes to the collective signature

Procedure

1. Before installing *S7 F Systems* V6.1: Copy the F-Library *Failsafe Blocks* (V1_2) or (V1_1):
   - Open the F-Library, *Failsafe Blocks* (V1_2), for example
   - In *SIMATIC Manager* , select **Save As**
   - Specify a different name, such as *"Failsafe Blocks* (V1_2)x"
2. Install *S7 F Systems* V6.1.
3. Rename the copied F-Library as its original name
   - Open the F-Library, *Failsafe Blocks* (V1_2)x, for example
   - In *SIMATIC Manager* , select **Edit > Rename**
   - Specify the original name, such as *"Failsafe Blocks* (V1_2)"
4. Prior to the initial compilation, save the current state of your safety program as a reference ("Save Reference" in the "Safety Program", dialog) so that it will be available for future comparisons
5. You can now recompile your S7 program

2.3.4 Use case 4

Objective

To upgrade your S7 program with *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3 SP1.

Introduction

This use case helps you when migrating your safety program by upgrading blocks of the *Failsafe Blocks* (V1_2) F-Library to blocks of the *S7 F Systems Lib* V1_3 SP1 F-Library, enabling you to use the new functions of the *S7 F Systems Lib* V1_3 SP1 F-Library.

When you migrate from the *Failsafe Blocks* (V1_2) F-Library to *S7 F Systems Lib* V1_3 SP1, the F-FBs in your safety program are overwritten by F-Blocks with other block signatures. This means that the collective signature will change.

Note

With the *Failsafe Blocks* (V1_2) F-Library, a discrepancy analysis was not performed with redundant F-I/O and "2-channel equivalent" type of sensor interconnection in the F-Module driver, even if a discrepancy time greater than or less than (<>) 0 ms (10 ms by default) was configured in the "Redundancy" tab in *HW Config*.

Starting with *S7 F Systems Lib* V1_3 SP1 and *S7 F Configuration Pack* V5.5 SP3, a discrepancy analysis is always performed for a discrepancy time greater than or less than (<>) 0 ms.

If you want to shut down the discrepancy analysis, configure a discrepancy time equal to (=) 0 ms in the "Redundancy" tab in *HW Config*.

Requirements

If F-Block types are used in your project, you first must recreate these with *S7 F Systems Lib* V1_3 SP1. To do so, follow the procedure outlined in the section entitled " Updating F-Block types that you have created (Page 45) ".

Consequences

- The collective signature is changed
- A complete download with CPU-STOP is required

Procedure

1. Install *S7 F Systems* V6.1 with *S7 F Systems Lib* V1_3 SP1
2. Prior to the initial compilation, save the current state of your safety program as a reference ("Save Reference" in the "Safety Program", dialog) so that it will be available for future comparisons.
3. In the Safety Program dialog, select the *S7 F Systems Lib* V1_3 SP1 F-Library

   To do so, click the "Library Version" button in the "Edit Safety Program" dialog
4. In the S7 program, update the existing F-Block types For information about how to do this, refer to the section entitled "Updating F-Block types that you have created (Page 45)"
5. In the *CFC Editor* under **Options > Block Types** , click "Clean Up"
6. Update all block types in the *CFC Editor* by selecting **Options > Block Types** and clicking "New Version"
7. Recompile your hardware configuration
8. Recompile your S7 program

Additional measures if your project contains the F-Blocks F_1oo2_R or F_2oo3_R

The F-Blocks F_1oo2_R and F_2oo3_R have their own DELTA input. This input has the data type F_REAL in *S7 F Systems Lib* V1_3 SP1. Until *Failsafe Blocks* (V1_2), the DELTA input had the data type REAL.

Proceed as follows when migrating to *S7 F Systems Lib* V1_3 SP1:

1. Before upgrading, document the parameter assignments and interconnections to this input
2. Perform the migration to *S7 F Systems Lib* V1_3 SP1
3. Reintroduce the documented parameter assignments and interconnections into your project, using the F_R_FR converter and a validity check, if necessary For information about the validity check, refer to the section entitled " Programming data exchange from the standard user program to the safety program (Page 93) "

Additional measures for F-Module drivers

When migrating to *S7 F Systems Lib* V1_3 SP1, interconnections of the following F-Module driver outputs may require special handling:

- PROFIsafe1
- PROFIsafe2
- DIAG_1
- DIAG_2

Proceed as follows when migrating to *S7 F Systems Lib* V1_3 SP1:

1. Before performing the migration, document the interconnection of the PROFIsafe1 and DIAG_1 outputs, along with the value of the LADDR input
2. For redundant F-I/O, also document the interconnection of the PROFIsafe2 and DIAG_2 outputs before performing the migration, along with the value of the LADDR_R input
3. Perform the migration to *S7 F Systems Lib* V1_3 SP1
4. Interconnect the documented interconnections at the PROFIsafe1 and DIAG_1 outputs to the new module driver F_PS_12, whose value at the LADDR input matches the documented LADDR
5. For redundant F-I/O, interconnect the documented interconnections at the PROFIsafe2 and DIAG_2 outputs to the new F-Module driver F_PS_12, whose value at the LADDR input matches the documented LADDR_R

Table 2- 3 Non-redundant F-I/O

Failsafe Blocks (V1_2)	*S7 F Systems Lib* V1_3 SP1
Interconnection at the original F-Module driver:	**Interconnection at the F-Module driver F_PS_12:**
PROFIsafe1	PROFIsafe
DIAG_1	DIAG
LADDR	LADDR

Table 2- 4 Redundant F-I/O

Failsafe Blocks (V1_2)	*S7 F Systems Lib* V1_3 SP1
Redundant interconnection at the original F-Module driver:	**Interconnection at the first F-Module driver F_PS_12:**
PROFIsafe1	PROFIsafe
DIAG_1	DIAG
LADDR	LADDR
	Interconnection at the second F-Module driver F_PS_12:
PROFIsafe2	PROFIsafe
DIAG_2	DIAG
LADDR_R	LADDR

Additional measures for redundant fail-safe digital input modules SM 326; DI 8 X NAMUR and SM 326; DI 24 X DC 24 V

With the redundant fail-safe digital input modules SM 326; DI 8 X NAMUR and SM 326; DI 24 X DC 24 V, information about detected discrepancy errors is provided at the DIAG_1 and DIAG_2 outputs of the F_M_DI8 and F_M_DI24 F-Block drivers when the *Failsafe Blocks* (V1_1) F-Library is used.

Starting with *S7 F Systems Lib* V1_3 SP1, discrepancy error information is output at the DISCF and DISCF_R outputs of the F-Channel driver F_CH_DI.

If you are using logic that evaluates this information, modify it accordingly.

2.3.5 Use case 5

Objective

Simple software update from *S7 F Systems* V5.2 SP1 through V5.2 SP4 to *S7 F Systems* V6.1 without program changes.

Introduction

This use case helps you to migrate from *S7 F Systems* V5.2 SP1 through V5.2 SP4 to V6.1 if you want to retain compatibility with your previous Version 5.2 SP1 through SP4.

Requirements

Your S7 program must be compiled, downloaded and executable for the original *Failsafe Blocks* (V1_2) F-Library. Ensure that this is the case by printing out the safety program and performing an online comparison.

Consequences

- No changes to the safety program
- No changes to the collective signature

Procedure

1. Install *S7 F Systems* V6.1.
2. Prior to the initial compilation, save the current state of your safety program as a reference ("Save Reference" in the "Safety Program", dialog) so that it will be available for future comparisons.
3. You can now recompile your S7 program.

Note

For certain projects that were created with *S7 F Systems* V5.2 SP1 to SP3, migration to *S7 F Systems* V6.1 can result in a signature change in spite of the procedure described here.

For more information, refer to the FAQs under:

FAQ: (http://support.automation.siemens.com/WW/view/en/23541471)

2.3.6 Use case 6

Objective

Simple software update from *S7 F Systems* V6.0 to *S7 F Systems* V6.1 without program changes.

Introduction

This use case helps you to migrate from *S7 F Systems* V6.0 if you want to retain compatibility with your previous Version V6.0.

Requirements

Your S7 program must be compiled, downloaded and executable for the original *S7 F Systems Lib* V1_3. Ensure that this is the case by printing out the safety program and performing an online comparison.

Consequences

- No changes to the safety program
- No changes to the collective signature

Procedure

1. Install *S7 F Systems* V6.1.
2. Prior to the initial compilation, save the current state of your safety program as a reference ("Save Reference" in the "Safety Program", dialog) so that it will be available for future comparisons.
3. You can now recompile your S7 program.

2.3.7 Use case 7

Objective

Update your S7 program from *S7 F Systems Lib* V1_3 to *S7 F Systems Lib* V1_3 SP1

Introduction

This use case helps you when migrating your safety program by upgrading blocks of the *S7 F Systems Lib* V1_3 to blocks of the *S7 F Systems Lib* V1_3 SP1 F-Library, enabling you to use the new functions of the *S7 F Systems Lib* V1_3 SP1 F-Library.

When you migrate from *S7 F Systems Lib* V1_3 to *S7 F Systems Lib* V1_3 SP1, the F-FBs in your safety program are overwritten by F-Blocks with other block signatures. This means that the collective signature will change.

Requirements

If F-Block types are used in your project, you first must recreate these with *S7 F Systems Lib* V1_3 SP1. To do so, follow the procedure outlined in the section entitled "Updating F-Block types that you have created (Page 45)".

Consequences

For possible consequences, refer to the section entitled "Acceptance test following system upgrade (Page 177)".

Procedure

1. Install *S7 F Systems* V6.1 with *S7 F Systems Lib* V1_3 SP1.
2. Prior to the initial compilation, save the current state of your safety program as a reference ("Save Reference" in the "Safety Program", dialog) so that it will be available for future comparisons.
3. In the Safety Program dialog, select the *S7 F Systems Lib* V1_3 SP1 F-Library.

   To do so, click the "Library Version" button in the "Edit Safety Program" dialog.
4. In the S7 program, update the existing F-Block types. For information about how to do this, refer to the section entitled "Updating F-Block types that you have created (Page 45)".
5. In the *CFC Editor* under **Options > Block Types** , click "Clean Up".
6. Update all block types in the *CFC Editor* by selecting **Options > Block Types** and clicking "New Version".
7. Recompile your hardware configuration.
8. Recompile your S7 program.

2.3.8 Updating F-Block types that you have created

If F-Block types are used in your project, you must recreate these with *S7 F Systems Lib* V1_3 SP1. To do so, you must have the project (source project) in which the F-Block type was created in the *CFC Editor* with the menu command **Chart > Compile > Chart as Block Type**.

Proceed as follows:

1. Install *S7 F Systems* V6.1 with *S7 F Systems Lib* V1.3 SP1.
2. In the Safety Program dialog, select the *S7 F Systems Lib* V1.3 SP1 library.
3. In the *CFC Editor* under **Options > Block Types** , click "Clean Up".
4. Update all block types in the CFC Editor by selecting **Options > Block Types**.
5. Open the CFC chart to be compiled and compile it in the *CFC Editor* using the menu command **Chart > Compile > Chart as Block Type**.
6. You can now copy the compiled F-Block type to your S7 programs in which you want to use it.

See also

Creating F-Block types (Page 86)

2.3.9 Updating a multiproject master data library

Introduction

The following describes how you transfer the F-Blocks from *S7 F Systems Lib* V1.3 SP1 to the master data library of your multiproject.

Requirements

The user projects are already updated.

Note

Update the user projects in your multiproject as described in the sections "Migrating to S7 F Systems V6.1 (Page 30)" to "Use case 5 (Page 42)".

If you are using F-Block types that you have created in your master data library, you must update these F-Block types as described in the section entitled "Updating F-Block types that you have created (Page 45)".

All attributes of the F-Blocks must be applied. Do not perform a comparison with the old F-Block attributes.

Procedure

Proceed as follows to continue using the master data library with fail-safe blocks as usual in the multiproject:

1. Open the block folder in the master data library of your multiproject and select the "Details" view option.
2. Delete all blocks with the author "F_SAFE11" or "F_SAFE12".

   Important: When doing so, enable the "Also delete symbolic block names" option.
3. In *SIMATIC Manager,* select **File > Open** and switch to the "Libraries" tab.
4. Select the *"S7 F Systems Lib* V1_3 SP1" library and confirm with "OK".

   Result: The library is opened.
5. Select the "F-User Blocks" library component to be copied. Select the **Edit > Copy** menu command.
6. Select the folder in the master data library (destination) in which the copied library component is to be placed.
7. Select the menu command **Edit > Paste**. The copied library component is placed into the master data library.
8. Repeat Steps 3 to 5 for the "F-Control Blocks" library component.
9. Repeat Steps 3 to 5 for the block folder containing the F-Block types that you created.
10. In *SIMATIC Manager* , select **Options > Charts > Update Block Types** for the master data library. This will update all blocks in your sample solutions and process tag types in the master data library.

Configuration

3

3.1 Configuration overview

Introduction

The following section lists the main points in which the configuration of an F-System differs from that of an S7 standard system.

F-Components that must be configured

You must configure the following hardware components for *S7 F Systems* V6.1:

1. F-CPU, such as CPU 414-4H
2. F-I/O, such as:
   - ET 200S fail-safe modules
   - S7-300 fail-safe signal modules (in ET 200M)
   - ET 200eco fail-safe I/O modules
   - ET 200pro fail-safe I/O modules
   - Fail-safe DP standard slaves
   - Fail-safe PA field devices

3.2 Particularities for configuring an F-System

Configuring same as in standard system

You configure an S7 F/FH Systems fail-safe system the same as a standard S7 system. That is, you configure and assign parameters for the hardware in *HW Config* as a centralized configuration (F-CPU) and as a distributed configuration (F-CPU, F-SMs in ET 200M, F-Modules in ET 200S, ET 200pro, and ET 200eco, and fail-safe DP standard slaves).

For a detailed description of the configuration options, refer to System Manual " Safety Engineering in SIMATIC S7 (http://support.automation.siemens.com/WW/view/en/12490443) ".

Special F-relevant tabs

There are a few special tabs for the F-Functionality included in the object properties of the F-I/O. These tabs are described in the following sections.

Assigning symbols for fail-safe inputs/outputs of F-I/O

For convenience when programming S7 F/FH Systems, it is particularly important that you assign symbols for the fail-safe inputs and outputs of the F-I/O in *HW Config*.

Saving and compiling the hardware configuration

You must save and compile the hardware configuration of S7 F/FH Systems in *HW Config*. This is required for subsequent programming of the safety program.

Changing safety-relevant parameters

Note

If you change a safety-relevant parameter for an F-I/O or an F-CPU, you must recompile the S7 program.

The same applies to changes in S7 connections for safety-related communication via S7 connections.

Rules for F-Systems

In addition to the rules that are generally applicable for the arrangement of modules in an S7-400, you must also comply with the following conditions for an F-System:

- Prior to downloading the safety program, you must download the hardware configuration to the F-CPU.
- If you have changed the configuration of an F-I/O or the F-CPU (cycle times of the cyclic-interrupt OB), you must recompile the S7 program and download it to the F-CPU.

3.3 Configuring the F-CPU

Rules for configuring an F-CPU

⚠WARNING

An F-CPU containing a safety program must have a password.

You must satisfy the following conditions:

- The "CPU contains safety program" option must be selected
- A password must *always* be assigned

You specify these settings via the object properties of the F-CPU in *HW Config*.

⚠WARNING

Configuring a protection level

Access by means of the F-CPU password must not be authorized when making changes to the standard user program, since this also enables changes to be made to the safety program. To rule out this possibility, you must configure Protection Level 1.

Procedure for configuring the protection level

Use the following procedure to configure Protection Level 1:

1. In *HW Config*, select the F-CPU, such as CPU 417-4H, and select the **Edit > Object Properties** menu command.
2. Open the "Protection" tab.
3. Set Protection Level 1: Access protection for F-CPU or keyswitch setting and Removable with password.

   Enter a password of the F-CPU in the fields provided and select the "CPU contains safety program" option.

   For information on the password for the F-CPU, refer to "Setting up access rights for the F-CPU (Page 65)". Pay special attention to the warning in the section entitled "Setting up access rights for the F-CPU (Page 65)".

Important parameters for the F-CPU in S7 FH Systems

To prevent the time monitoring from being triggered during a master-reserve switchover (for example, H-CiR), you must configure the OB 3x organization block(s) provided for safety programs with a priority greater than (>) 15 in the Cyclic Interrupts tab of the F-CPU. You should not place any standard blocks in these OBs.

The cyclic interrupt OB of the safety program must be configured as a "cyclic interrupt OB with special handling". Only in this case will this cyclic interrupt be called immediately before the start of the disable time for priority classes greater than (>) 15 minutes during updating of the reserve. In the "Cyclic Interrupt with Special Handling" field of the "H-Parameters" tab of the CPU properties, enter the number of the highest-priority cyclic-interrupt OB to which F-Blocks of the safety program section are assigned in the *CFC Editor*.

- Ensure that the correction factor is set to 0 ms in the "Diagnostics/Clock" tab of the "Clock" group.

Note

For S7 FH Systems, only settings up to 12 hours are permitted.

In S7 FH Systems, you are not permitted to modify safety-related self-tests by means of SFC 90 "H_CTRL". Otherwise, the safety program will go to F-STOP after 24 hours, at the latest. It is forbidden to switch test components on or off (submodules 0 to 5 of modes 20, 21, and 22).

For the same reason, you must not disable the update too long by means of SFC 90 "H_CTRL".

If these rules are not adhered to, an F-STOP is triggered. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: error detected" (event ID 16#75E1)

Changing the OB3x cycle time

You must recompile the S7 program after making changes to the OB 3x cycle times.

See also

Overview of access protection (Page 63)

3.4 Configuring the F-I/O

Configuring same as in standard system

The ET 200S, ET 200eco, and ET 200pro F-Modules and the S7-300 F-SMs are always configured in the same way:

Once the F-I/O have been inserted into the station window of *HW Config*, you can access the configuration dialog by selecting **Edit > Object Properties** or by double-clicking the F-I/O.

When making changes to F-I/O in *HW Config*, you will be prompted to enter the password for the F-CPU.

The values in the shaded fields are automatically assigned by *S7 F Systems* in the F-relevant tab. You can change the values in the non-shaded fields.

Additional Information

For information on which ET 200S, ET 200eco, ET 200pro F-Modules and which S7-300 F-SMs you can use, refer to System Manual " Safety Engineering in SIMATIC S7 (http://support.automation.siemens.com/WW/view/en/12490443) ".

For a description of the parameters, refer to the *context-sensitive online help* for the tab and the relevant *F-I/O manual*.

For information on what you must consider when configuring the F-Monitoring time for F-I/O, refer to System Manual " Safety Engineering in SIMATIC S7 (http://support.automation.siemens.com/WW/view/en/12490443) ".

Assigning symbols for fail-safe inputs/outputs of F-I/O

For convenience when programming S7 F/FH Systems, it is important that you assign symbols for the fail-safe inputs and outputs of the F-I/O in *HW Config*.

Note that for certain F-I/O (such as S7-300 F-SMs and ET 200S fail-safe modules), a 1oo2 sensor evaluation can be assigned. In this case, only one of the two combined channels is available.

We recommend that you identify the unavailable channel as reserved in the symbol table. To find out which of the channels combined by the 1oo2 sensor evaluation you can access in the safety program, refer to the relevant manuals for the F-I/O.

Operating mode

For S7-300 fail-safe signal modules, the operating mode parameter setting determines whether the modules are operated in standard mode (used as standard S7-300 signal modules except for SM 326; DO 8 × DC 24 V/2 A) or in safety mode.

ET 200S, ET 200pro, and ET 200eco fail-safe modules can only be used in safety mode.

Group diagnostics for fail-safe S7-300 signal modules

The Group diagnostics parameter activates and deactivates the transmission of channel-specific diagnostic messages of F-SMs (such as wire break and short circuit) to the F-CPU. For availability reasons, you should shut down the group diagnostics on unused input or output channels of the following F-SMs:

- SM 326; DI 8 x NAMUR
- SM 326; DO 10 x DC 24 V/2A
- SM 336; AI 6 x 13 bits

⚠ WARNING

"Group diagnostics" must be activated on all connected channels of fail-safe F-SMs in safety mode.

Check to verify that you have shutdown group diagnostics only for unused input and output channels.

You can optionally enable diagnostic interrupts.

For SM 326; DI 24 x DC 24 V (order no. 6ES7326-1BK01-0AB0 and later) and SM 326; DO 8 x DC 24 V/2A PM, the following applies:

By disabling a channel in *HW Config* you also disable its group diagnostics function.

PROFIsafe addresses

The PROFIsafe addresses (assigned F_source_address" and "F_destination_address" parameters) uniquely identify the source and destination.

F_destination_address

The F_destination_address uniquely identifies the PROFIsafe destination (of the F-I/O). Therefore, the F_destination_address must be unique network-wide and station-wide (see Section "Rules for address assignment").

To prevent incorrect parameter assignment, a *station-wide unique* F_destination_address is automatically assigned when the F-I/O are placed in *HW Config*.

In S7 F/FH Systems, you must ensure that the F_destination_address is *unique network-wide* when multiple stations are present in a network by manually changing the F_destination_addresses.

If you change the F_destination_address, the uniqueness of the F_destination_address within the station is checked automatically. You yourself must make sure that the F_destination_address is unique network-wide.

You must set the F_destination_address on the F-I/O via the DIP switch before installing the F-I/O.

Note

For the following S7-300 F-SMs, the F_destination_address is the same as the start address of the F-SM/8:

- SM 326; DI 24 x DC 24 V (order no. 6ES7326-1BK00-0AB**0**),
- SM 326; DI 8 x NAMUR (order no. 6ES7326-1RF00-0AB**0**)
- SM 326 DO 10 x DC 24 V/2A (order no. 6ES7326-2BF01-0AB**0**)
- SM 336; AI 6 x 13 Bit (order no. 6ES7336-1HE00-0AB**0**)

Assign low start addresses for these F-SMs if you are also using other F-I/O.

F_source_address

The F_source_address is a unique identification of the PROFIsafe source (of the associated F-CPU). The F_source_address is automatically assigned to avoid an incorrect parameter assignment.

Rules for address assignment

⚠ WARNING

Rule for PROFIBUS subnets:

The PROFIsafe destination address and, thus, the switch setting on the address switch of the F-I/O must be unique network-wide* and station-wide** (system-wide).

For S7-300 F-SMs and ET 200S, ET 200eco and ET 200pro F-modules, you can assign a maximum of 1022 different PROFIsafe destination addresses.

* A network consists of one or more subnets. "Network-wide" means across subnet boundaries.

** "Station-wide" means for one station in *HW Config* (e.g., an S7-400H station).

For information about using S7 F/FH Systems and S7 Distributed Safety on Ethernet subnets, refer to Manual " S7 Distributed Safety - Configuring and Programming (http://support.automation.siemens.com/WW/view/en/22099875) ".

3.5 Configuring fail-safe DP standard slaves

Requirements

In order to use fail-safe DP standard slaves, the standard slaves must be on the PROFIBUS DP and support the PROFIsafe bus profile.

Configuration with a GSD file

As is the case in a standard system, the fail-safe DP standard slaves are configured based on the device specification in the GSD file (generic station description).

A GSD file contains all of the properties of a DP standard slave. For fail-safe DP standard slaves, portions of the specification are protected by a cyclic redundancy check.

The GSD files are supplied by the device manufacturers. The supplied GSD file must satisfy the PROFIsafe Specification V2.0 in order for fail-safe DP standard slaves to be operated with *S7 F Systems*. Ask for confirmation of this from the device manufacturer.

Import the GSD files into your project (see *STEP 7* online help). Once the fail-safe DP standard slave is imported, it can be selected from the hardware catalog of *HW Config*.

Backing up the data structure of the device in GSD files

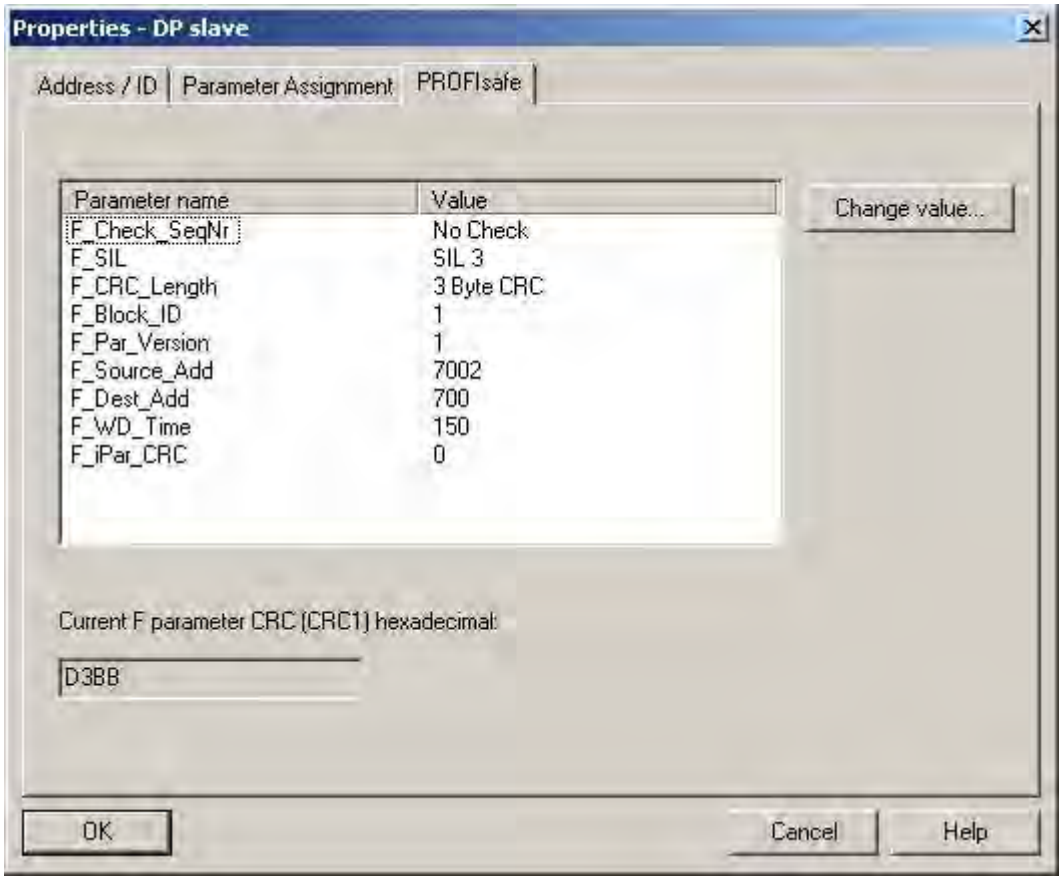
Starting with *PROFIsafe Specification V2.0*, the device data structure described in the GSD file must be backed up with a CRC stored in this file ("setpoint" for F_IO_StructureDescCRC).

Procedure for configuring with a GSD file

Import the GSD file into your project (see *STEP 7* online help).

1. Select the fail-safe DP standard slave in the hardware catalog of *HW Config* and insert it into your DP master system.
2. Select the fail-safe DP master.
3. Open the object properties dialog using the **Edit > Object Properties** menu command or by double-clicking the slot of the F-Component.

Channel-level passivation is not supported for fail-safe DP standard slaves.



"PROFIsafe" tab

The parameter texts specified in the GSD file are contained in the "PROFIsafe" tab under "Parameter name". The associated current value is included under "Value". You can modify this value using the "Change Value" button.

The parameters are explained below.

Parameter "F_Check_SeqNr"

This parameter defines whether the sequence number is to be incorporated in the consistency check (CRC calculation) of the F-User data frame.

The "F_Check_SeqNr" parameter must be set to "No check" in the PROFIsafe V1 MODE. Only fail-safe DP standard slaves that behave accordingly are supported.
"F_CHECK_SeqNr" is irrelevant in PROFIsafe V2 MODE.

"Parameter F_SIL"

This parameter defines the safety class of the fail-safe DP standard slave. The parameter is device-dependent. Possible settings for the "F_SIL" parameter are "SIL 1" to "SIL 3", depending on the GSD file.

Parameter "F_CRC_Length"

The length of the CRC signature must be 2 bytes, 3 bytes or 4 bytes, depending on the length of the F-User data (process data), the safety class, and the PROFIsafe mode. This parameter provides information to the F-CPU on the size of the CRC2 key in the safety message frame.

In PROFIsafe V1 MODE:

For a user data length less than or equal to 12 bytes, select 2-byte CRC as the setting for the "F_CRC_Length" parameter; for a user data length ranging from 13 bytes to 122 bytes, select 4-byte CRC.

S7 F Systems supports only "2-byte CRC"; the fail-safe DP standard slave must behave accordingly.

In PROFIsafe V2 MODE:

For a user data length less than or equal to 12 bytes, select 3-byte CRC as the setting for the "F_CRC_Length" parameter; for a user data length ranging from 13 bytes to 123 bytes, select 4-byte CRC.

S7 F Systems supports only "3-byte CRC"; the fail-safe DP standard slave must behave accordingly.

"F_Block_ID" parameter

The F_Block_ID parameter has the value 1 if the F_iPar_CRC parameter exists, otherwise it has the value 0.

The value 1 of the F_Block_ID parameter indicates that the data record for the value of F_iPar_CRC has been extended by 4 bytes. You cannot change the parameter.

"Parameter F_Par_Version"

This parameter identifies the PROFIsafe operating mode. You can find out the value range offered from the operating modes supported by the device.

For fail-safe DP standard slaves, you can set this parameter to the following:

- Set "F_Par_Version" to "1" (PROFIsafe V2 MODE) for a homogenous PROFIBUS DP network, if the device and the F-CPU support this. Otherwise, set it to "0" (PROFIsafe V1 MODE).

Note

The following F-CPUs support V2 MODE:

CPU 412-3H, order no. 6ES7412-3HJ14-0AB0 and later

CPU 414-4H, order no. 6ES7414-4HM14-0AB0 and later

CPU 417-4H, order no. 6ES7417-4HT14-0AB0 and later

If you set "F_Par_Version" to "1" for F-CPUs that do not support V2 MODE, this will result in a communication error for the safety-related communication with the device. One of the following diagnostic events is then entered in the diagnostic buffer of the F-CPU:

- "F-I/O passivated": Check value error (CRC)/Sequence number error ...
- "F-I/O passivated": F-Monitoring time exceeded at the safety message frame detected in the F-CPU ...

Parameters "F_Source_Add" and "F_Dest_Add"

The PROFIsafe addresses ("F_Source_Add" and "F_Dest_Add" parameters) uniquely identify the source and destination.

The "F_Source_Add" and "F_Dest_Add" parameters for fail-safe DP standard slaves correspond to the assigned "F_source_address" and "F_destination_address" parameters of other F-I/O. Therefore, the information about PROFIsafe address assignment provided in the section entitled "Configuring the F-I/O (Page 51)" is generally applicable for fail-safe DP standard slaves.

Parameter "F_WD_Time"

This parameter defines the F-monitoring time in the fail-safe DP standard slave.

The "F_WD_Time" parameter can be set in 1 ms increments. The value range of the "F_WD_Time" parameter is specified by the GSD file.

For more information about the F-Monitoring time, refer to the section entitled "Run times, F-Monitoring times, and response times (Page 410)".

Parameter "F_iPar_CRC"

CRC via individual device parameters (i-parameter).

The individual device parameters (i-parameter) of a fail-safe DP standard slave are configured with their own configuration tool provided by the device manufacturer.

Enter the CRC calculated by the configuration tool from the device manufacturer for the backup of the i-parameters. *S7 F Systems* takes the value into account when calculating the F-Parameter CRC (CRC1) .

See also

Safety engineering in SIMATIC S7
(http://support.automation.siemens.com/WW/view/en/12490443)

3.6 Configuring fail-safe PA field devices

Fail-safe PA field devices are configured in the same way as fail-safe DP standard slaves.

When configuring PA field devices, follow the procedure described in the chapter entitled "Configuring fail-safe DP standard slaves (Page 54)".

3.7 Configuring redundant F-I/O

Introduction

To increase availability of your automation system and, thus, to prevent process failures due to faults in the F-System, you can optionally equip fail-safe S7 F/FH Systems with a fault-tolerant feature (S7 FH Systems). You can achieve this increased availability by component redundancy (F-CPU, communication connections, and F-I/O).

For S7 F Systems, availability can also be increased without fault-tolerant configuration. You can use S7-300 fail-safe signal modules (F-SMs) redundantly in one ET 200M or in various ET 200Ms.

Note

With redundant F-SMs, you must observe the following:

- Both F-SMs must be of the same type
- For both F-SMs, the "Safety Mode" operating mode must be selected in the "Parameters" tab

Procedure

Proceed as follows to configure two S7-300 fail-safe signal modules redundantly, for example:

1. In *HW Config*, configure both F-SMs in the ET 200M(s).
2. Configure the first F-SM:

   In the "Parameters" tab, select "Safety Mode".
3. Configure the second F-SM:

   In the "Parameters" tab, select "Safety Mode".
4. For the second F-SM, set the "2 Modules" operating mode in the "Redundancy" tab.
5. In the "Find Redundant Module" dialog box for the F-SM, select the first F-SM.
6. Set additional parameters, if necessary. The settings will be applied automatically for the first F-SM. As soon as there are two redundant F-SMs, parameter assignment changes for one will also be applied for the other.
7. For redundant fail-safe digital input modules, the F-Channel driver F_CH_DI can perform a discrepancy analysis for increased availability. This requires you to set the "Discrepancy time" parameter. Setting the discrepancy time to "0" disables the discrepancy analysis. For more information, refer to the online help for the "Redundancy" tab.

See also

F_CH_DI: Fail-safe channel drivers for digital inputs of F-I/O (except fail-safe DP standard slaves) (Page 283)

3.8 Configuration in Run (CiR)

Introduction

Certain process control systems must not be shut down during operation. This is due to the complex nature of automation systems or the high cost of a restart, for example. At certain times, however, these systems do require changes or expansions. This is possible with Configuration in RUN mode (CiR for short). With CiR, the program sequence is stopped for up to 2500 ms. The process outputs retain their current value during this period. This has no effect on the actual process, especially in process control systems.

System change during operation by means of CiR is based on provisions in the master system of the initial configuration for a subsequent hardware expansion of your automation system. You define suitable CiR elements that you can later replace with real elements on a step-by-step basis in RUN mode. You can download a configuration modified in this way to the F-CPU during process operation.

Before performing the procedures described below, read the CiR instructions in manual " Modifying the System during Operation via CiR (http://support.automation.siemens.com/WW/view/en/14044916) ".

Calculating F-Monitoring times

When calculating the minimum F-Monitoring times, take the CiR synchronization time into account. Refer also to the section entitled " Run times, F-Monitoring times, and response times (Page 410) ".

Reducing F-Monitoring times

If the calculated values for the process are not acceptable, you can recalculate the F-Monitoring time by reducing the CiR synchronization time. You have the following options for doing so:

- Decrease the number of input and output bytes of the master system.
- Decrease the number of guaranteed slaves of the master systems that you intend to change.
- Decrease the number of master systems that you intend to change during a CiR.

Extending the maximum cycle time using CiR

If CiR is used, the maximum cycle time is extended by the *lesser* of the following two values:

- CiR synchronization time of F-CPU

  The CiR synchronization time of the F-CPU is the sum of the CiR synchronization times for all DP master systems that are to be changed simultaneously. The CiR synchronization time of a DP master system is displayed in *HW Config* in the Properties dialog for the relevant CiR object.

- Upper limit of CiR synchronization time

  The default value for this upper limit is 1 second. You can increase or decrease this value according to your requirements by calling SFC 104 "CiR".

For instructions on determining the maximum cycle time, refer to the manual for the F-CPU you are using.

Limiting CiR synchronization time:

The F-CPU compares the actually calculated CiR synchronization time with the current upper limit for the CiR synchronization time. If the calculated value is less than the current upper limit, CiR is enabled. The default value for the upper limit of the CiR synchronization time in the F-CPU is 1 second. SFC 104 enables you to change this value. You can raise or lower the upper limit within a range of 200 ms to 2500 ms. For a detailed description of SFC 104, refer to manual " System Software for S7-300/400 System and Standard Functions (http://support.automation.siemens.com/WW/view/en/1214574) ".

3.8.1 Configuring F-I/O with CiR

Introduction

With CiR, you can add new F-I/O to your system or delete existing F-I/O from your system. The procedure for doing so is presented in the following two sections.

Adding F-I/O with CiR

Follow these steps to add F-I/O to your system:

1. Configure the new F-I/O in *HW Config*. To do so, follow the procedure outlined in the manual entitled "Modifying the System during Operation via CiR (http://support.automation.siemens.com/WW/view/en/14044916)". Handle the F-I/O as standard I/O.
2. Expand your S7 program and compile it with the "Generate module drivers" option enabled.
3. Disable safety mode. For more information, refer to the section entitled "Deactivating safety mode (Page 162)".
4. Download your safety program.

   Note

   A user acknowledgement at input ACK_REI of the F-Channel driver is necessary to enable the F-I/O.

5. Enable safety mode. For more information, refer to the section entitled "Activating safety mode (Page 163)".

   Note

   Parameter reassignment is *not* supported for F-I/O. This also pertains to H-CiR. For more information, refer to the manual entitled "Automation System S7-400H Fault-tolerant Systems (http://support.automation.siemens.com/WW/view/en/1186523)".

Deleting F-I/O with CiR

Follow these steps to delete F-I/O from your system:

1. Delete the F-I/O in *HW Config* by following the procedure described in the manual entitled "Modifying the System during Operation via CiR (http://support.automation.siemens.com/WW/view/en/14044916)". Handle the F-I/O as standard I/O.
2. Change your S7 program and compile it with the "Generate module drivers" option enabled.
3. Disable safety mode. For more information, refer to the section entitled "Deactivating safety mode (Page 162)".
4. Download your safety program.
5. Download your configuration using CiR.
6. Enable safety mode. For more information, refer to the section entitled "Activating safety mode (Page 163)".

Note

You can only delete existing F-I/O using CiR if the F-I/O in the relevant master system is assigned to a CiR object.

Access Protection

4

4.1 Overview of access protection

Purpose and mode of operation

Access protection protects S7 F/FH Systems from unauthorized access, such as undesirable downloads to the F-CPU from the Engineering System (ES). In addition to the password for the F-CPU, you need an additional password for the safety program for S7 F/FH Systems.

The table below provides information about the password for the F-CPU and the password for the safety program.

Password for F-CPU	
Password assignment	In *HW Config* during configuration of the F-CPU in the "Protection" tab of the "Properties" dialog box
Password request when	• Downloading the entire S7-program from the *CFC Editor* or *SIMATIC Manager* • Downloading safety program changes from the *CFC Editor* • Performing a memory reset from the *CFC Editor* or *SIMATIC Manager* • Changing non-interconnected inputs in CFC test mode
Password validity	Access permission is valid without restriction until they are explicitly canceled using the corresponding function of *SIMATIC Managers* (with the **PLC > Access Rights > Cancel** menu command) or until you close the last *STEP 7* application. Access permission can become invalid if the hardware configuration of the CPU is changed and downloaded.

Password for Safety Program	
Password assignment	In *SIMATIC Manager*, **Options > Edit Safety Program** menu command
Password request when	• Compiling changes to the safety program • Downloading changes to the safety program • Disabling and enabling safety mode • Changing non-interconnected inputs in CFC test mode • Saving the safety program as a reference • Changing the shutdown behavior in the "Safety Program" dialog • Adding F-I/O in which safety mode has been enabled or that only support safety mode • Opening the Properties dialog for F-I/O in *HW Config* • Making changes in the PROFIsafe tab in *HW Config* • Making changes in the "F-Configuration" tab for a fail-safe intelligent DP slave In addition, starting with *PCS 7* V7.1: • Opening an F-Chart • With an open F-Chart – Editing object properties of an F-block – Assigning parameters to an input/output on an F-Block – Instantiating an F-block – Inserting an F-Block or CFC chart • With F-Runtime groups – Opening a CFC chart (with the "Read only" option) – Opening an F-Runtime group in the runtime view – Moving an F-Runtime group in the runtime view – Modifying the properties of an F-Runtime group
Password validity	The access permission lasts for one hour after correct password entry, during which time it is reset to another hour after each action requiring a password, or until access permission is explicitly canceled in *SIMATIC Manager* (**Options > Edit safety program** menu command, then click the **Password** button followed by the **Cancel access rights** button).

4.2 Setting up access rights for the F-CPU

Procedure

1. In *SIMATIC Manager*, select the F-CPU or its S7 program.
2. Select the menu command **PLC > Access Rights > Setup**. In the "Protection" tab of the dialog box that appears, enter the password that was assigned during the F-CPU parameter assignment.

Access permission is always valid until you revoke them again (**PLC > Access Rights > Cancel**) or until you close the last *STEP 7* application.

⚠WARNING

Limiting access using the ES

If you have not used access protection to limit access to the ES to only those persons who are authorized to modify the safety program, you must ensure the effectiveness of password protection in the ES with the following organizational measures:

- Only authorized individuals are granted access to the password.
- Authorized individuals must explicitly cancel the access permission for the F-CPU before exiting the ES. If you do not adhere to these measures strictly, you must also use a screen saver with a password that is accessible only to authorized individuals.

Access by means of the F-CPU password must not be authorized when making changes to the standard user program, since this also enables changes to be made to the safety program. To rule out this possibility, you must configure *Protection Level 1*.

If safety mode is enabled after the access permission has been revoked, check whether:

- The collective signature of the safety program online

  and
- The collective signature of the safety program for which an acceptance test was performed are identical

If not, you must download the correct safety program to the F-CPU.

Note

Automatic downloading of safety programs is not supported in multiprojects. Passwords must be entered when downloading to the respective F-CPU.

Transferring the safety program to multiple F-CPUs

⚠WARNING

If multiple F-CPUs can be accessed over a network (such as MPI) from one ES, you must take the following actions to ensure that the safety program is downloaded to the correct F-CPU:

Use passwords specific to each F-CPU, e.g., a uniform password for the F-CPUs having the respective MPI address as an extension (max. 8 characters): PW_8.

Note the following:

- Before downloading a safety program to an F-CPU for which access permission by means of an F-CPU password does not yet exist, you must first revoke existing access permission for any other F-CPU.

Changing the password

The password can be changed only by modifying the configuration.

In an S7 F-System, you must switch the F-CPU to STOP to do so.

In an S7 FH-System, a password change (configuration change) is possible without interrupting the process (in RUN).

⚠WARNING

Password protection

After an unbuffered restart (cold start), the current password is deleted from the RAM load memory, and the old password from the Flash EPROM memory card becomes valid again. You should use organizational measures to prevent this old password on the Flash EPROM memory card from becoming known to many people.

4.3 Setting up access permission for the safety program

Requirements

An existing safety program (F-Plan) is required to set up access permission for the safety program.

Procedure for setting up/changing access permission for the safety program

Proceed as follows to set up or change the password for the safety program:

1. In *SIMATIC Manager*, select the F-CPU or its S7 program.
2. Select the menu command **Options > Edit Safety Program**.
3. In the "Safety Program" dialog box that appears, click the "Password" button. Now perform the necessary step for your situation:
   - Either enter the password for the safety program for the first time. In this case, you can ignore the "Old Password" prompt.
   - Or change the current password for the safety program. In this case, you must enter the old password in the "Old Password" field.

     With the "Cancel Access Rights" button, you must now immediately cancel the one-hour access permission in effect since the last time the password was entered. Every user who wants to perform an action that requires a password to be entered must now re-enter the password for the safety program, regardless of whether or not an hour has elapsed since the password was last entered.

⚠WARNING

Limiting access using the ES

If you have not used access protection to limit access to the ES to only those persons who are authorized to modify the safety program, you must ensure the effectiveness of password protection in the ES with the following organizational measures:

- Only authorized individuals are granted access to the password.
- Authorized individuals must explicitly cancel the access permission for the safety program before leaving the ES. If you do not adhere to these measures strictly, you must also use a screen saver with a password that is accessible only to authorized individuals.

Note

The access authorization relates to the safety program itself and not to the individuals working on the ES. This must be taken into account in particular with regard to multi-user engineering projects.

Note

Automatic editing and compilation of safety programs is not supported.

Each action requires a valid password.

Assigning a new password for the safety program

If you have not yet entered a password for the safety program, you will be prompted to enter one when compiling the safety program.

⚠ WARNING
Passwords must be unique Use different passwords for the F-CPU and the safety program to increase the level of access protection. The passwords for different safety programs must also be unique.

Changing the password for the safety program

As in Windows, you change the password by entering the old password once and the new password twice.

Revoking access permission for the safety program

You can cancel access permission at any time with the password for the safety program. Proceed as follows:

1. In *SIMATIC Manager*, select the F-CPU or its S7 program.
2. Select the menu command **Options > Edit Safety Program**.
3. In the dialog box that appears, click the "Password" button.
4. In the "Password" dialog box, click "Cancel Access Rights".

Programming

5

5.1 Overview of programming

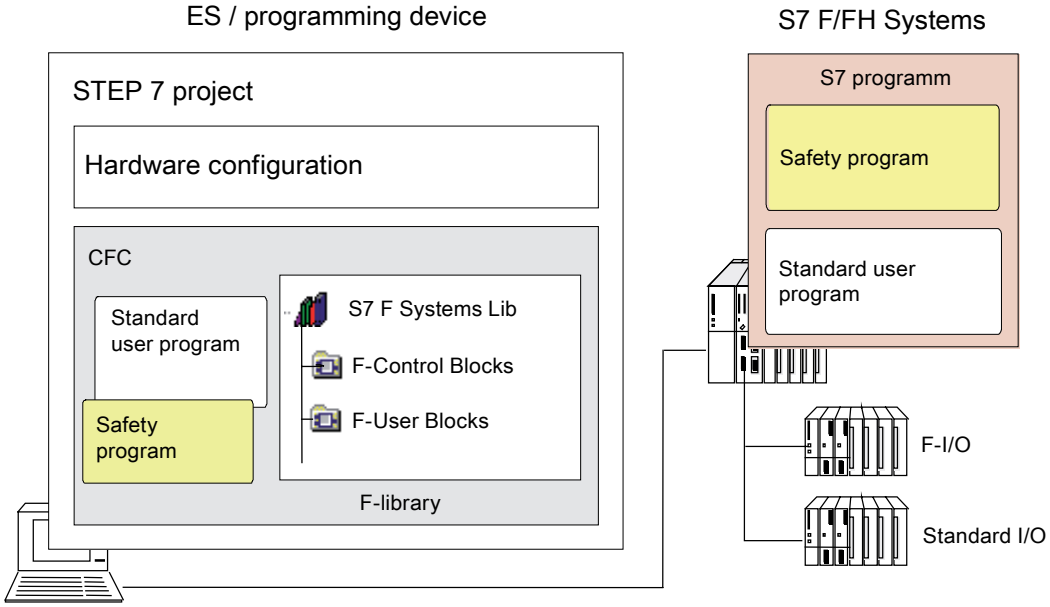
Introduction

A safety program consists of fail-safe blocks that you select from an F-Library and interconnect using the CFC programming languages and fail-safe blocks that are automatically added when the safety program is compiled.

During compilation, fault control measures are automatically added to the safety program you create, and additional safety-related tests are performed.

Schematic Structure of a Project with Standard User Program and Safety Program

In the figure below, you can see the schematic structure of an S7 program in the ES and the F-CPU:



The S7 program generally consists of a standard user program in which you program program sections that are not necessary for the safety function and a safety program for the safety function.

5.1.1 Structure of the safety program

Representation of Program Structure

The figure below shows the schematic structure of a safety program for *S7 F Systems*. A safety program consists of CFC charts with F-Blocks that are assigned to F-Runtime groups.

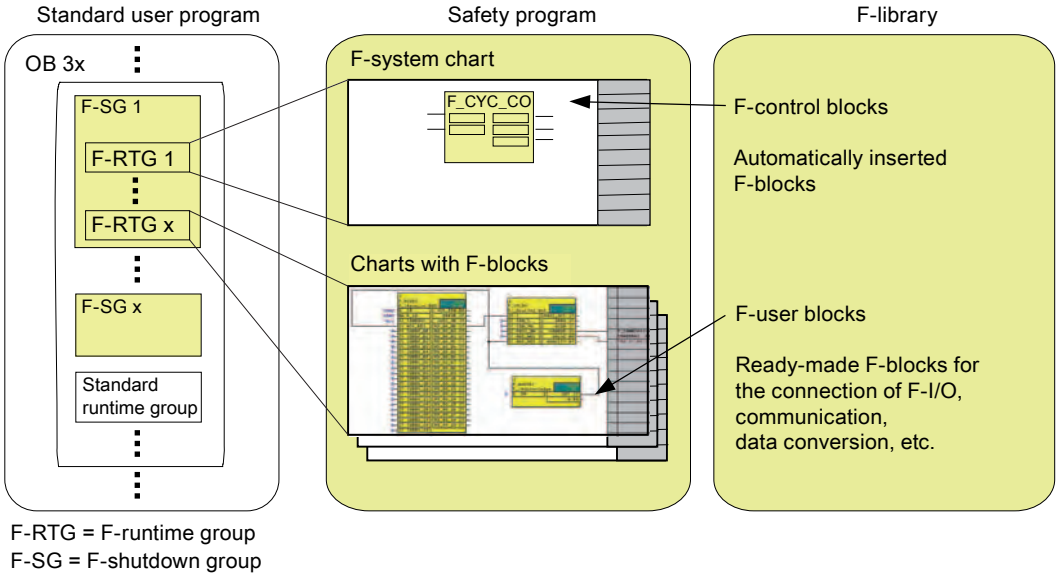


Figure 5-1 Components of the safety program in *S7 F Systems*

Description of Program Structure

The safety program contains F-Runtime groups and charts assigned to them. The charts contain F-Blocks including their parameter assignment and interconnection.

You insert the F-Runtime groups at the beginning of an OB. Use a cyclic interrupt OB (OB 30 to OB 38) to do so. F-Runtime groups are combined in F-Shutdown groups.

The cyclic interrupt OB can also contain standard runtime groups.

F-Runtime groups

When programming the safety program, you cannot insert F-Blocks directly into tasks (OBs). First create an F-Runtime group into which you then insert the F-Blocks. An F-Runtime group does not become an F-Runtime group until F-Blocks are called in it. As long as the F-Runtime group is empty, it appears as a standard runtime group. Your safety program comprises several F-Runtime groups.

F-Shutdown groups

An F-Shutdown group forms a self-contained unit of your safety program. An F-Shutdown group contains user logic that is executed or shut down simultaneously. The F-Shutdown group contains one or more F-Runtime groups that are assigned to a common task. You can choose whether an error in the execution of the safety program should cause a full shutdown of the entire safety program (full shutdown) or a partial shutdown, i.e., only for the F-Shutdown group in which the error occurred. F-Blocks can only exchange data between F-Shutdown groups via special F-Blocks. All F-Channel drivers that belong to an F-I/O must be located in the same F-Shutdown group.

See also

Creating the Safety Program (Page 72)

F-STOP (Page 84)

5.2 Creating the Safety Program

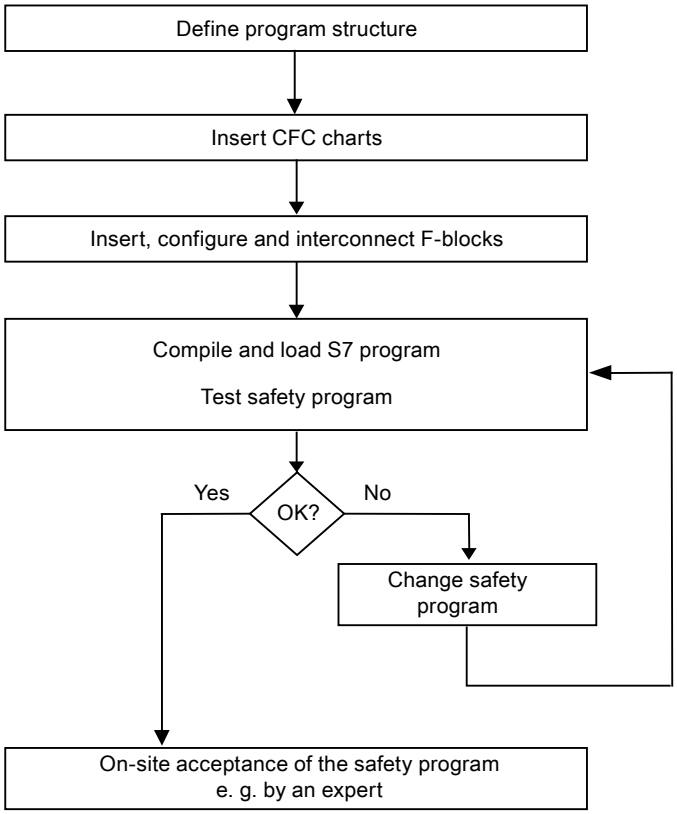
5.2.1 Basic procedure for creating the safety program

Requirements

- You must create a project structure in *SIMATIC Manager*.
- You must already have configured the hardware components of your project - in particular, the F-CPU and the F-I/O - prior to programming the safety mode.
- You must have assigned your safety program to an F-capable central processing unit, such as CPU 412-3H, CPU 414-4H, or CPU 417-4H.

Basic procedure

Proceed as follows to create a safety program:



5.2.2 Defining the program structure

Introduction

When designing an S7 program for S7 F/FH Systems, you must answer the following additional questions as compared to a standard program:

- Which components of the S7 program must be fail-safe?
- What response times do you want to achieve?

  Based on this, you must divide your S7 program into different OB 3x cyclic interrupts.

Note

You can improve performance by writing sections of the program that are not required for the safety functions in the standard user program.

When determining which elements to include in the standard user program and which to include in the safety program, 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 test.

Rules for the program structure

You must keep the following rules in mind when designing a safety program for S7 F/FH Systems:

- You can only assign F-Shutdown groups with F-Blocks to the OB 3x (OB 30 to OB 38) cyclic interrupts.
- A chart can contain both F-Blocks and standard blocks. You cannot compile these charts as F-Block types.
- The F-I/O can only be accessed in the safety program via the F-Channel drivers.

5.2.3 Assigning parameters for the maximum F-cycle monitoring

The F-CPU monitors the F-Cycle time for each cyclic interrupt OB 3x that contains F-Runtime groups. The first time you compile the S7 program, you will be prompted to enter a value for the maximum cycle time "MAX_CYC" that can elapse between two calls of this OB. For information about setting F-Monitoring times, refer to chapter "Run times, F-Monitoring times, and response times (Page 410)".

If you need to change the maximum F-Cycle time, set the F-Cycle time at the MAX_CYC parameter of block F_CYC_CO-OB3x in chart @F_CycCo-OB3x.

⚠WARNING

Default setting of the maximum MAX_CYC

The default setting for the maximum F-Cycle time is 3,000 milliseconds. Check whether this setting is appropriate for your process. Change the default setting if necessary.

Note

For changes to the F-Cycle time during RUN mode, refer to chapter " Changing the time ratios or F-Monitoring times (Page 174) ".

5.2.4 Rules for programming

⚠WARNING

Do not change values created during compilation

During compilation, you must not change automatically executed placements, interconnections, and parameter assignments of F-Blocks.

- In particular, you must not manipulate the structural components COMPLEM and PAR_ID of F-Data types.
- You must not change the F-Control blocks that are automatically inserted in the safety program (in F-System charts) (except the MAX_CYC parameter at F_CYC_CO).
- In F-Blocks, you are only permitted to interconnect or assign the parameters that are described in the online help or manual.

You must not change or delete the F-Blocks in the block container.

⚠WARNING

The cyclic interrupt OB 3x group call interval is monitored relative to the maximum value; that is, monitoring is performed to determine whether the call is executed often enough, but not whether it is executed too often.

For this reason, you must implement fail-safe times using F-Blocks such as F_TON, F_TOF, and F_TP, rather than via counters (OB calls).

5.2.5 Notes for working with CFC

⚠WARNING

Compression changes the signature

Compressing a CFC program (using the **Options > Customize > Compile/Download** menu command in the *CFC Editor*), changes the collective signature of your safety program.

You must therefore do this prior to the acceptance test.

F-Blocks appear in the CFC chart highlighted in color. They are highlighted in yellow to indicate that a safety program is involved.

CFC charts and F-Runtime groups with F-Blocks are yellow and marked with an "**F**" in order to distinguish them from the charts and runtime groups of the standard user program.

5.2.6 Inserting CFC charts

Procedure

In the charts folder, insert individual CFC charts in the same way as for standard user programs:

- In *SIMATIC Manager* by selecting the menu command **Insert > S7 Software > CFC**
- Directly in the *CFC Editor* with the menu command **Chart > New**

Note

To install the newly inserted CFC charts directly in the respective planned cyclic interrupt OB 3x, you must position the CFC installation pointer accordingly.

Hierarchical charts

Chart outputs of a lower-level chart that are not interconnected internally cannot be further interconnected in the higher-level chart.

5.2.7 Inserting F-Runtime groups

Rules for F-Runtime groups of the safety program

- We recommend the following procedure in order to achieve F-cycles of the same length:

  When mixing F-Runtime groups and standard runtime groups in a cyclic interrupt OB, execute the F-Runtime groups *before* the standard runtime groups to avoid unnecessarily prolonging the runtime of the F-Shutdown group and influencing the response time.
- An F-Runtime group must retain the default setting for the runtime properties Reduction ratio and Phase shift as follows:
  - Reduction ratio = 1
  - Phase shift = 0

  These values must not be modified.
- The automatically generated F-Runtime groups must not be shifted. Changes must not be made within this F-Runtime group, either.

⚠ WARNING
Optimization of the runtime sequence in the *CFC* can cause a change in the collective signature and an increase in the response times of the safety program. Therefore, starting with *CFC* V7.0 SP1, the runtime sequence can no longer be optimized.

Procedure

As with standard user programs, you insert F-Runtime groups in the runtime editor of the *CFC Editor*.

5.2.8 F-Shutdown groups

Rules for F-Shutdown groups of the safety program

- You are not permitted to interconnect F-Blocks belonging to different F-Shutdown groups.

  For more information, refer to chapter "Programming data exchange between F-Shutdown groups in an F-CPU (Page 90)"
- All F-Channel drivers that belong to an F-I/O must be located in the same F-Shutdown group.

Defining F-Shutdown groups

As soon as you place F-Blocks in the *CFC Editor* for the first time, all F-Runtime groups in each OB 3x form an F-Shutdown group.

Dividing/combining F-Shutdown groups through manual placement of F_PSG_M

When you add or delete one or more F_PSG_M blocks in your project, the order of your F-Shutdown groups will change. If you change the layout of your F-Shutdown groups, you must make sure that the F-module driver and all associated F-Channel drivers are integrated in the same F-Shutdown group.

You have the option of dividing one F-Shutdown group into two F-Shutdown groups. To do so, in the runtime editor of the *CFC Editor*, place the F_PSG_M block in the last F-Runtime group to be associated with the first F-Shutdown group. All subsequent F-Runtime groups then form the second F-Shutdown group. The F_PSG_M block is not an F-Block. However, you are still permitted to place it in F-Runtime groups. For more information, refer to chapter "Determining the runtime sequence (Page 80)".

The number of F-Shutdown groups in all tasks is limited to 110. The number of F-Runtime groups in one task is unlimited.

You have the option of combining two F-Shutdown groups. To do so, in the runtime editor of the CFC Editor, delete the F_PSG_M block between the F-Shutdown groups. If you combine F-Shutdown groups that exchange data by means of F-System blocks into one common F-Shutdown group, you must delete these F-System blocks and replace them with direct interconnections.

5.3 Inserting and interconnecting F-Blocks

5.3.1 Inserting F-Blocks

Procedure

Insert the F-Blocks into your chart as usual in *CFC*.

Note

All F-Blocks are highlighted in yellow in the *CFC Editor* and in *SIMATIC Manager*. Only these blocks are part of your safety program. In addition, the F-User Blocks folder in the F-Library contains standard blocks, for example, to convert F-Data types to standard data types.

Rules for F-Blocks

- The blocks in the **F-Control Blocks** folder are automatically inserted when the S7 program is compiled. You are not permitted to insert these blocks yourself.
- You are not permitted to place an F-Block instance in several F-Runtime groups. This can take place, for example, by copying and inserting an F-Runtime group into another task.

Note

F-libraries in different versions

Your ES can contain *multiple* versions of the F-Library at the same time. However, a safety program can only contain F-Blocks from *one* version.

⚠WARNING

Entries for F-Blocks in the symbol table must not be changed

You are not permitted to change or delete the names of the F-Blocks in the "Symbol" column of the symbol table in your S7 program. This also applies for changes in the symbol table that is assigned to the F-Library.

5.3.2 Parameter assignment and interconnection of F-Blocks

Procedure

Inputs and outputs of the F-Blocks are parameterized and interconnected as usual in *CFC*.

Rules for parameter assignment and interconnection of F-Blocks

- You are only permitted to assign parameters or interconnect the parameters documented in chapter "F-Libraries (Page 193)".
- You are not permitted to interconnect input EN and output ENO of the F-Blocks and F-Runtime groups. Likewise, you are not permitted to assign a value of 0 (FALSE) to EN.
- The F-Data types are implemented in the program as structures in which only the first **DATA** component is relevant for you.

  If you do not take this into account, the safety program/F-Runtime group will go into F-STOP, i.e., an F-Startup will be required.

⚠WARNING

Illegal changes to input parameters of F-Blocks can cause a shutdown of the safety program and its outputs.

Changes can be made to the input parameters of F-Blocks with F-Data types as follows:

- Offline using the *CFC Editor*

  or
- Online using the CFC test module with safety mode disabled

If you change F-Data types online with safety mode enabled without using the CFC test mode, this can cause a shutdown of the relevant outputs or an F-STOP will be initiated.

Recommendation: meaningful names for placed F-Blocks

Assign a meaningful name to each placed F-Block. You are free to choose the name.

5.3.3 Determining the runtime sequence

Correct runtime sequence of F-Blocks

The sequence of the F-Blocks within the F-Shutdown group is relevant. The number of F-Runtime groups into which the F-Shutdown group is divided is irrelevant.

In principle, the correct runtime sequence of the different F-Block types is as follows:

1. Automatically placed:
   - F-Module drivers for F-I/O with inputs or with inputs and outputs
   - F-Communication blocks and F-System blocks for receiving
   - F-Blocks for converting data
2. F-Channel drivers for inputs
3. F-Blocks for user logic
4. F-Channel drivers for outputs
5. Automatically placed:
   - F-Block F_PLK
   - F-Block F_PSG_M
   - F-Module drivers for F-I/O with outputs or with inputs and outputs
   - F-Communication blocks and F-System blocks for sending
   - F-Block F_PLK_O
   - F-Block F_DIAG

The runtime sequence of the F-Blocks listed in Items 1 and 5 is corrected automatically when the S7 program is compiled. You must, however, always ensure that the F-Channel drivers and F-Blocks for user logic are placed appropriately and adhere to the sequence described above. This ensures that all inputs are read first, the appropriate processing steps are initiated, and then all outputs are written to.

Determining the runtime sequence

You determine the runtime sequence in the *CFC Editor* in the same way as for a standard user program.

Note

A change in the runtime sequence also changes the collective signature.

5.4 Automatically inserted F-Blocks

F-Control blocks

During compilation of a CFC chart with F-Blocks, the following F-Control blocks are automatically inserted into the safety program:

- F_DIAG
- F_CYC_CO
- F_PLK
- F_PLK_O
- F_PS_12
- F_PS_MIX
- F_PSG_M *
- F_TEST
- F_TESTC
- F_TESTM

*) The F_PSG_M block is only placed one time during the migration of *Failsafe Blocks* (V1_1) or for programs with *Failsafe Blocks* (V1_2) of *S7 F Systems* V5.2 with no SP.

During compilation of a CFC chart with F-Blocks, the following blocks are automatically inserted into the standard user program:

- DB_INIT
- DB_RES
- F_SHUTDN
- RTGLOGIC
- F_VFSTP1
- F_VFSTP2
- F_MOVRWS *
- F_CHG_WS *

*) Insertion of the blocks F_MOVRWS and F_CHG_WS depends on your programmed user logic.

⚠WARNING

Do not change automatically inserted F-Control blocks

The automatically inserted F-Control blocks are visible following compilation. You must not delete these F-Blocks and must not make any changes to them, as this could cause errors during the next compilation. For exceptions, refer to the description of the F-Blocks in Appendix " F-Libraries (Page 193) ".

Note

When the S7 program is compiled, additional blocks (DB_RES) and calls that you are not permitted to change are automatically inserted at the beginning of the runtime sequence in OB 100.

5.5 F-Startup and reprogramming restart/startup protection

F-Startup

S7 F Systems does not distinguish between a CPU cold restart and a CPU warm restart. The F-Blocks F_CHG_BO, F_CHG_R, and F_MOV_R are an exception to this. For more information, refer to the sections entitled "Blocks and F-Blocks for data conversion (Page 236)" and "Multiplex blocks (Page 353)". Both a CPU cold restart and a CPU warm restart result in an F-Startup.

Following an F-Startup, the safety program starts up automatically with the initial values.

An F-Startup takes place:

- After a CPU-STOP if you carry out a restart (warm restart) or a cold restart of the F-CPU
- After an F-STOP if you carry out the following steps:·
  - Assign a 1 for the restart at the "Restart" input
  - Reset the value to the original value of 0 after the value has been applied

After a partial shutdown of the safety program, only the F-Shutdown groups that were in F-STOP carry out an F-Startup.

F-Shutdown groups that are not error-free remain in F-STOP.

⚠WARNING

Saved error information is lost during an F-Startup.

During an F-Startup following an F-CPU STOP, the F-System performs an automatic reintegration of the F-I/O.

A data handling error or an internal fault can also trigger a safety program restart with the initial values of the F-Blocks. If your process does not allow such a startup, you must program a restart/startup protection in the safety program: Process data outputs must be blocked until manually enabled. The process data output block must not be released until it is safe to do so and faults have been corrected.

After the error has been corrected, one of the following actions is required:

- User acknowledgment on the F-Channel driver
- User acknowledgment on the F_RCVBO or F_RCVR F-Block or F_RDS_BO

The receipt data are reintegrated automatically for the F-Blocks F_R_BO and F_R_R, which are used for data exchange between F-Runtime groups.

Restart/startup protection

If the process does not allow the safety program to start up automatically with the initial values, you must program a response to the F-Startup. The F_START F-Block is available to signal an F-Startup of the safety program with the initial values.

The COLDSTRT output parameter indicates to you that an F-Startup has occurred.

Examples

The following measures are available for you to respond to a safety program startup with the initial values:

- Programming of an output **interlock** following startup via the PASS_ON passivation inputs at the channel drivers for outputs. To do so, interconnect the COLDSTRT output of the F_START F-Block with input S of an SR-flip-flop (F_SR_FF) and output Q of the F_SR_FF with PASS_ON of the fail-safe channel driver for outputs. You can then manually enable the interlock:
  - By means of a button that is accessed via an F-I/O

    or
  - By entering an ES/OS by means of the F_QUITES F-Block.

    You must interconnect output Q of the F-Channel driver associated with the button or output OUT of F_QUITES with input R of F_SR_FF.
- Programming of a **wait loop** so that the internal statuses of the safety program correspond again to the process status.
- Programming using multiplexers: The output of an F_MUX2_R multiplexer is controlled by the COLDSTRT output of the F_START F-Block. This enables a program branch other than in cyclic mode to be executed after a restart.

5.6 F-STOP

Introduction

When the safety program detects a safety-relevant error, it initiates an fault reaction. If substitute values cannot be output, the executed fault reaction is referred to as an F-STOP.

Types of F-STOP

There are two types of F-STOP:

- **Full shutdown**

  All F-Shutdown groups of the F-CPU are shut down. The shutdown takes place in the following order:

  – First, the F-Shutdown group in which the fault was detected is shut down.

  – All other F-Shutdown groups are then shut down within twice the time interval that you have configured as the F-monitoring time for the slowest OB.

- **Partial shutdown**

  Only the F-Blocks of the F-Shutdown group in which the fault was detected are shut down.

A shutdown of F-Shutdown groups means:

- The outputs of the F-I/O controlled by the F-Shutdown group are passivated.
- The F-Channel drivers of the F-Shutdown group set output QBAD to "1" and output QUALITY to "0".
- The safety-related communication of the F-Shutdown group to other F-CPUs is interrupted.
- The data exchange of the F-Shutdown group with other F-Shutdown groups is interrupted.
- During data exchange from the safety program to the standard user program, the last valid values are provided to the standard user program
- The F_SHUTDN block generates a message that you can display on an OS.
- Diagnostic events are entered in the diagnostic buffer of the F-CPU.

The standard user program of the F-CPU continues to run, even in the event of an F-STOP.

To assign parameters for F-STOP, use the "Shutdown ..." button in the "Safety Program" dialog box. See also " "Shutdown Behavior" dialog box (Page 151) ".

Errors that trigger an F-STOP

- Corruption of
  – Data
  – Program sequence
  – Code
- CPU fault

Errors that always trigger a full shutdown

When an OB request error occurs (due to an OB overload, for example), a full shutdown is triggered irrespective of the F-STOP parameter assignment.

Manual initiation of an F-STOP

You can manually initiate an F-STOP by creating a positive edge at the RQ_FULL input of the "F_SHUTDN" F-Block.

Sequence of an F-STOP in S7 FH-Systems

Before a safety program in a redundant F-CPU goes to F-STOP, it performs the following steps:

- The fault occurs in the master:
  - The S7 FH-System carries out a master/reserve switchover.
  - The previous master then goes into TROUBLESHOOTING mode.

    If a fault is not found, the F-CPU reconnects itself. For more information, refer to Manual " Automation System S7-400H Fault-tolerant Systems (http://support.automation.siemens.com/WW/view/en/1186523) ".

    If a fault is found, the previous master goes into DEFECTIVE mode.

    With redundant F-CPUs, single-sided faults do not cause a shutdown of the program execution.
- The fault occurs in both F-CPUs:
  - The safety program immediately goes into F-STOP mode.

Ending an F-STOP

Carry out an F-restart as described in chapter " F-Startup and reprogramming restart/startup protection (Page 82)".

See also

Initial run and startup characteristics (Page 176)

Group passivation (Page 100)

5.7 Creating F-Block types

5.7.1 Introduction

S7 F Systems gives you the option of generating an F-Block type from the CFC chart of a safety program. You can re-use F-Block types in other safety programs.

5.7.2 Rules for F-Block types

Rules for F-Block types

When creating a new F-Block type with F-Blocks, you follow the same basic procedure as for the standard user program. The same rules apply as for creating block types in *CFC*. In addition, you must also keep the following in mind:

- The new F-Block type can only contain F-Blocks from the F-Library, except for:
  - F-Channel driver
  - F-Blocks for F-Communication
  - F-Blocks F_CHG_BO, F_CHG_R, F_MOV_R or F_SWC_x
  - All F-Control blocks
  - All F-System blocks, except for F_START
- The F-Blocks that are called in the new F-Block type and the F-Blocks of the entire safety program in which the F-Block type is used must originate from the same library version. F-Blocks from different versions of the F-Library are not permitted.
- You are not permitted to connect an output of the F-Block with two chart inputs/outputs.
- The runtime sequence within one F-Block type is not automatically corrected during compilation. The sequence determined during creation is retained.

  Note

  If the runtime sequence is different from the data flow, for example, due to feedback, compilation of the F-Block type is canceled with an error.

- The chart inputs/outputs of the new F-Block type can have both F-Data types and standard data types.

- You are not permitted to use names of F-Blocks in the F-Library as the names of F-Block types.
- For instances of F-Blocks that are called in an F-Block type, we recommend that you assign names as follows:
  - Numbers only, as specified in the *CFC Editor*

    or

    Alphanumeric names, but that must begin with **F_**
  - Upper-case letters only
  - No "_" at the end

Note

Starting with *S7 F Systems* V6.1, you can set the S7_m_c attribute to 'true' for standard outputs. If you use this option, however, your safety program will no longer be backward-compatible with *S7 F Systems* V6.0.

⚠WARNING

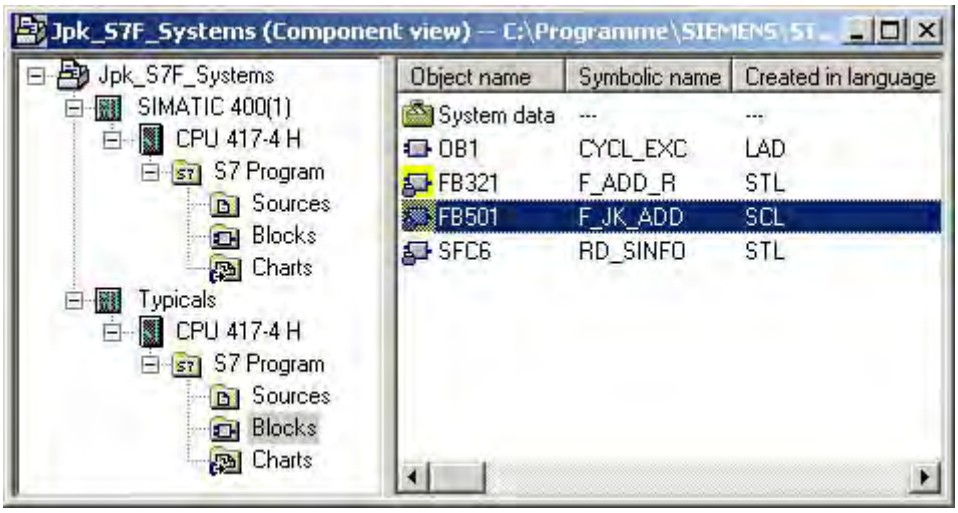
Outputs of F-Blocks always use the predefined initial values

When creating F-Block types, you are not permitted to change any initial values at F-Block outputs. *CFC* allows this and shows you the change. However, *S7 F Systems* always uses the initial values described in the F-Block description under "Default".

5.7.3 Creating F-Block types with "Compile Chart as F-Block Type"

Procedure

1. Create the CFC chart in a separate S7 program that is assigned to an F-CPU. The S7 program can be located in the same project.



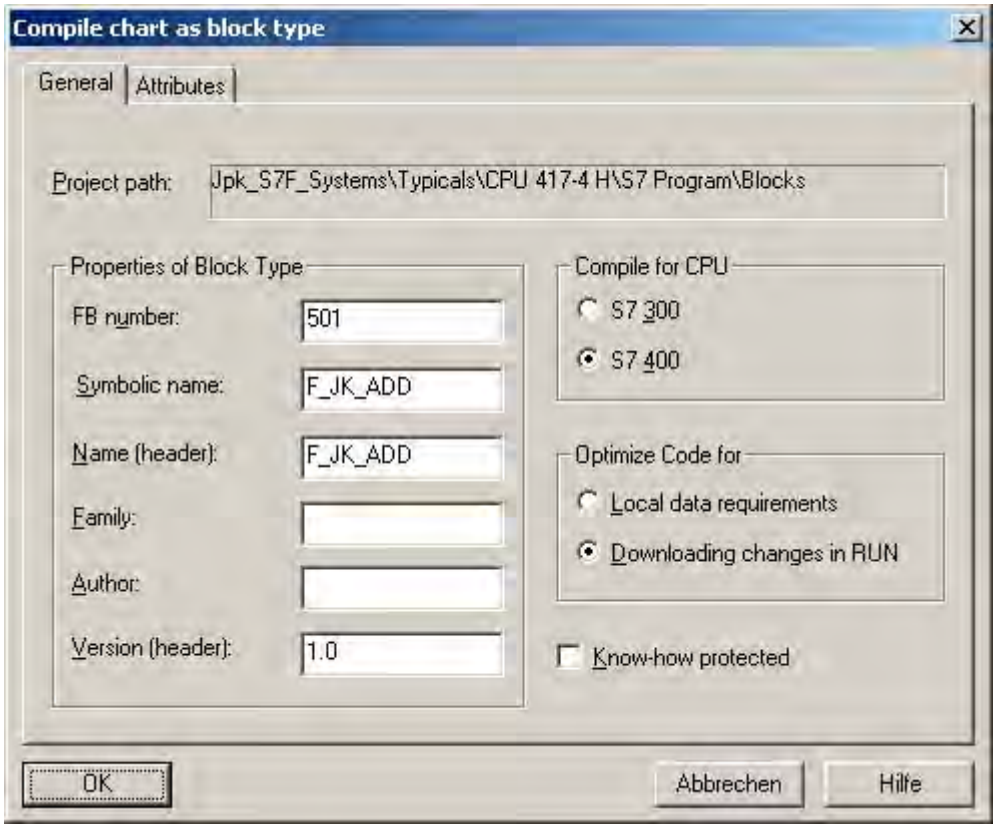
Note

Use a separate AS station to create an F-Block type.

As is common in *CFC*, always use a separate AS station containing only the safety program of the F-Block type to create an F-Block type. If you are using a CFC version prior to V6.1, do not compile these charts as a program ("**Chart" > "Compile" > "Charts as Program"** menu command). Otherwise, the new F-Block type may be defective because it incorrectly contains data from the project in which it was created. This can cause errors in your safety program and a safety program abort.

2. Open the required chart.

3. Select the menu command **Chart > Compile > Chart as Block Type**. A dialog box for entering the block properties is displayed.



4. Enter the properties of the new F-Block type. Make sure that the names under "Symbolic Name" and "Name (Header)" are identical.
5. Enable the options "Compile for CPU - S7 400" and "Optimize code - Downloading Changes in RUN" and confirm with OK.

   Know-how protection is always enabled, irrespective of the option setting.

   Result: A new block type that you can use in a safety program is generated.
6. Insert the new F-Block type along with the F-Blocks that it calls into a safety program and test is there.

Note

Attributes

Attributes whose names begin with "F_" are managed by *S7 F Systems*. For your own attributes, assign other names, as they could otherwise be deleted or overwritten during compilation.

5.7.4 Modifying F-Block types

Modifying F-Block types

You have to update modified F-Block types just as with all other block types in the *CFC-editor*. To do this, open the dialog "Block types" with the menu command **Options > Block Types** and click the button "New Version".

Modifications to already used F-Block types may result in your having to recompile and download the complete S7 program afterwards.

If you want to use a new version of the F-Library, you have to compile the F-Block types with this new version of the F-Library. You can find additional information in the chapter "Updating F-Block types that you have created (Page 45)"

See also

Downloading changes (Page 169)

System Acceptance Test (Page 179)

5.8 Programming data exchange between F-Shutdown groups in an F-CPU

Rules for the exchange of data between F-Shutdown groups

- If you want to exchange data between two F-Shutdown groups, you cannot interconnect the inputs and outputs directly. You have to use a special F-Block for this.
- You can find information on the run sequence in chapter " Determining the runtime sequence (Page 80) ".

Available F-Blocks

For the data exchange between F-Blocks in various F-Shutdown groups, you have to use the following F-System blocks:

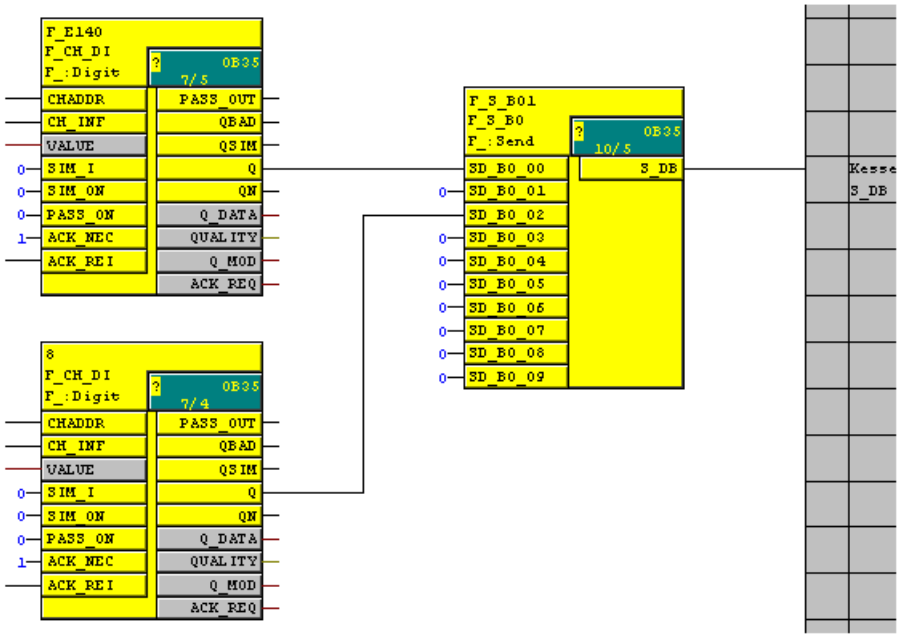
F-Block	Description
F_S_R / F_R_R	Safe transmission of 5 files of the F-Data type F_REAL.
F_S_BO / F_R_BO	Safe transmission of 10 files of the F-Data type F_BOOL.

Procedure

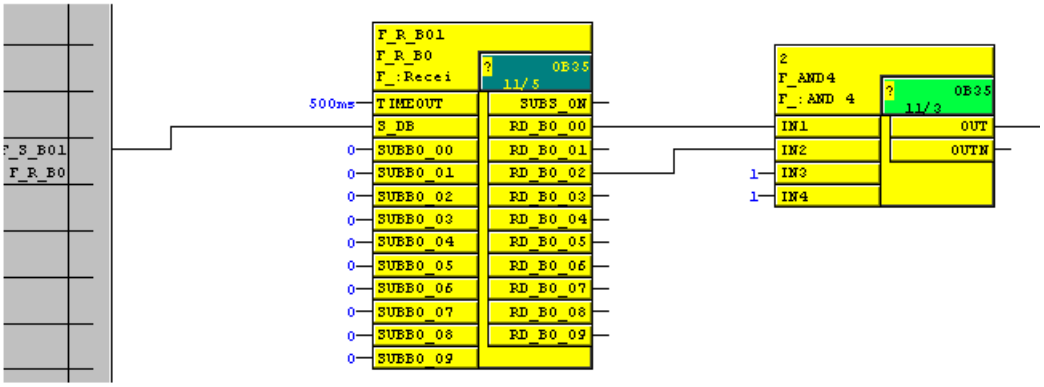
1. In the F-Shutdown group *from* which data are to be transmitted, insert an F-Block of the type F_S_R or F_S_BO.
2. In the F-Shutdown group *to* which data are to be transmitted, insert an F-Block of the type F_R_R or F_R_BO.
3. Interconnect the inputs SD_R_xx of the F_S_R or SD_BO_xx of the F_S_BO with the data to be transmitted.
4. Interconnect the outputs RD_R_xx of the F_R_R or RD_BO_xx of the F_R_BO with the inputs of the F-Blocks for the further processing of the received data.
5. Interconnect the output S_DB of the send block with with the input S_DB of the associated receive block.
6. Configure the TIMEOUT inputs of the F_R_R and F_R_BO with the required F-Monitoring time.

   For more information about calculation of the F-Monitoring time, refer to chapter " Run times, F-Monitoring times, and response times (Page 410) ".

Examples: Section from the chart of the F-Shutdown group *from which* the data are to be transmitted



Example: Section from the chart of the F-Shutdown group *to which* the data are to be transmitted



Note

If you interconnect F-Blocks in different F-Shutdown groups directly with each other (without the F-System blocks mentioned above), a compilation error will be generated at the next compilation.

If you interconnect F-Blocks within an F-Shutdown group with the F-System blocks mentioned above, an error message is generated.

5.9 Data exchange between safety program and standard user program

Overview

The standard user program and safety program use different data formats. Safety-related F-Data types are used in safety programs. Standard data types are used in the standard user program.

Therefore, you have to use special conversion blocks for data exchange.

Parameters are output as safety-related F-Data types in the safety program.

Data Transfer from the Safety Program to the Standard User Program

If data from the safety program is to be processed further in the standard user program, e.g., for monitoring, then you have to insert a data conversion block (F_F*data type_data type*) in the *CFC editor* between the two programs to convert F-Data types to standard data types. You can find these blocks in the F-Library.

Data Transfer from the Standard User Program to the Safety Program

Data from the standard user program cannot be processed in the safety program until a validity check is performed. You must perform additional process-specific validity checks in the safety program to ensure that no hazardous conditions can arise.

To process data from the standard user program in the safety program, you have to use F-Blocks to convert the data (F_*data type*_F*data type*) from the standard data types to safety-related F-Data types. If necessary, you must then subject the converted data to a programmed validity check. These F-Blocks can be found in the F-Library.

5.9.1 Programming data exchange from the safety program to the standard user program

Available conversion blocks

The following blocks are available for conversion:

Block	Description
F_FBO_BO	Converts F_BOOL to standard BOOL
F_FR_R	Converts F_REAL to standard REAL
F_FI_I	Converts F_INT to standard INT
F_FTI_TI	Converts F_TIME to standard TIME

Procedure

Proceed as follows:

1. Insert blocks of type F_FBO_BO, F_FR_R, F_FI_I, or F_FTI_TI into the charts of the standard user program. You can find these blocks in the F-Library.
2. Interconnect the inputs of type F_*data type* with similar signals from the safety program.
3. Interconnect the outputs of the standard data type with similar signals from the standard user program.

5.9.2 Programming data exchange from the standard user program to the safety program

Available F-conversion blocks

The following F-Blocks are available for conversion:

F-Block	Description
F_BO_FBO	Converts standard BOOL to F_BOOL
F_I_FI	Converts standard INT to F_INT
F_R_FR	Converts standard REAL to F_REAL
F_TI_FTI	Converts standard TIME to F_TIME

Procedure

Proceed as follows:

1. Insert F-Blocks of type F_BO_FBO, F_I_FI, F_TI_FTI, or F_R_FR into the charts of the safety program.
2. Interconnect the inputs of the standard data type with similar signals from the standard user program.
3. Interconnect the outputs of F-Data types by means of a validity check with similar signals in the safety program.

Note

The adding, changing, and deleting of interconnections from the standard user program to the F-Conversion blocks is considered a change in the safety program, even if this involves interconnections of a standard data type. This means that access permission is required for compilation (see "Access Protection (Page 63)").

WARNING

Validity check

The F-Blocks F_BO_FBO, F_I_FI, F_TI_FTI, and F_R_FR only perform a data conversion. This means you must program additional measures for validity checks in the safety program.

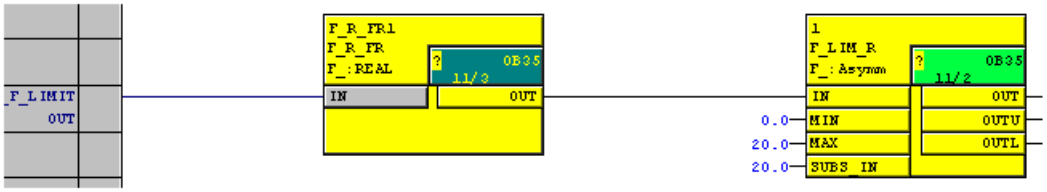
Validity check

The simplest type of validity check is a range definition with a fixed upper and lower limit, such as F_LIM_R.

Not all input parameters can be checked for plausibility in a sufficiently simple manner.

Example: Converting standard data types to F-Data types

Section from an F-Chart for converting REAL to F_REAL:



5.10 Implementation of user acknowledgment

Options for user acknowledgment

You can implement a user acknowledgment in one of the following ways:

- By means of an acknowledgment key that you connect to an F-I/O with inputs
- By means of manual input via the OS

User Acknowledgment by Means of Acknowledgment Key

Note

If you use the option of user acknowledgment by means of an acknowledgment key, and a communication error, an F-I/O fault, or a channel fault occurs at 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 remedied by a STOP-to-RUN transition of the F-CPU. Consequently, we recommend that you also provide for an acknowledgment by means of an OS for the acknowledgment for reintegration of an F-I/O to which an acknowledgment key is connected.

User acknowledgement by means of an OS

The F-Block F_QUITES is required to implement a user acknowledgment by means of an OS.

Procedure for programming user acknowledgment by means of an OS

1. Insert the F-Block F_QUITES into your safety program. The acknowledgment signal for evaluating user acknowledgments is provided at output OUT of F_QUITES.
2. On your OS, set up a field for manual entry of an "acknowledgment value" of "6" (first step in acknowledgment) and an "acknowledgment value" of "9" (second step in acknowledgment) in the input IN of F_QUITES.
3. Optional: On your OS, evaluate output Q of F_QUITES to indicate the time frame within which the second step in acknowledgment must occur or to indicate that the first step in acknowledgment has already occurred.

⚠WARNING

The two acknowledgment steps must not be triggered by one single operation, for example, by automatically storing them along with the time conditions in one program and using one operation to trigger them. By programming separate acknowledgement steps, you prevent erroneous triggering of an acknowledgement by your non-fail-safe operator station.

⚠WARNING

If your OS can access multiple F-CPUs that use F_QUITES for fail-safe acknowledgment, or if you have networked operator control and monitoring systems and F-CPUs (with F_QUITES F-Blocks), you must be sure that the correct F-CPU is in fact being addressed before executing the two acknowledgment steps:

- In each F-CPU, store a network-wide unique name for the F-CPU in a DB of your standard user program.
- In your OS, set up a field from which you can read out the F-CPU name from the DB online before executing the two acknowledgment steps.
- Optional: In your OS, set up a field to permanently store the F-CPU designation. 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.

See also

F-Blocks for F-Communication between F-CPUs (Page 201)

F-Channel drivers for F-I/O (Page 266)

F-I/O access

6

Access via F-Driver Blocks

In S7 F/FH Systems, the F-I/O are accessed via F-driver blocks rather than via the process image.

One F-Module driver per F-I/O and one F-Channel driver for each F-I/O input and output channel used are required.

F-Module driver

The F-Module driver takes over the PROFIsafe communication between the safety program and the F-I/O. The F-Module driver is automatically placed and interconnected in the safety program by the CFC driver generator.

F-Channel drivers

In your safety program, the F-Channel drivers form the interface to one channel of an F-I/O and execute signal processing. There are various F-Channel drivers, depending on the F-I/O (see "F-Channel drivers for F-I/O" (Page 266)).

You must place and interconnect F-Channel drivers in the safety program.

For redundant F-I/O, you only need one F-Channel driver for two redundant channels.

6.1 Positioning, interconnecting, and assigning parameters to F-Channel drivers

Requirement: Symbolic names

Enter a symbolic name for each channel used. You must assign this name to the VALUE or I_OUT_D inputs/outputs of the associated F-Channel driver. For greater clarity, enter the unused channels as reserved or unused in the symbol table.

Procedure

1. Position the appropriate F-Channel driver for each input channel and output channel used.
2. For each F-Channel driver, interconnect the VALUE or I_OUT_D input/output with the symbolic name of the associated channel. This step is required for all placed F-Channel drivers. For redundant F-I/O, interconnect the VALUE input/output with the symbolic name of the channel with the smaller channel address.
3. Interconnect the following inputs/outputs with your user logic:
   - I inputs of F-Channel drivers F_CH_DO and F_CH_BO
   - Q and QN outputs of F-Channel drivers F_CH_DI, F_PA_DI, and F_CH_BI
   - V outputs of F-Channel drivers F_CH_AI and F_PA_AI
4. Optional: Interconnect the simulation inputs/outputs.
5. Optional: Interconnect the PASS_ON input if you want to enable passivation of a channel, for example, as a function of particular states in your safety program.
6. Optional: Assign a value of "1" to the relevant ACK_NEC input if a user acknowledgement is required when reintegrating the channel. The default value of input ACK_NEC is "0" (see chapter "Group passivation (Page 100)").
7. Interconnect the relevant ACK_REI input with the reintegration acknowledgement signal (see chapter "Group passivation (Page 100)").
8. Optional: Interconnect the PASS_OUT or QBAD output to determine whether a fail-safe value or valid process data is output.
9. Optional: Evaluate the QUALITY output in the standard user program or the OS if you want to query or specify the process data status (Quality Code).
10. Optional: Evaluate the ACK_REQ output in the standard user program or the OS to determine whether a user acknowledgement is required.

Depending on the F-Channel driver, there are additional inputs and outputs that you can or must interconnect (see Appendix "F-Channel drivers for F-I/O (Page 266)")

6.2 Generating F-Module drivers

Generating F-Module drivers

Use the *CFC* driver generator to generate F-Module drivers.

When compiling the S7 program, select the "Generate module drivers" option in the "Compile Program" dialog box.

The driver generator will then position all automatically generated F-Module drivers in separate CFC charts called @F_(1), @F_(2), etc. The F-Module driver instances will automatically be assigned the name you have entered in *HW Config* for the associated F-I/O (F_Name_x). The F-Channel drivers are interconnected with the associated F-Module drivers.

If you are using *PCS 7*, additional blocks will be inserted by the driver generator (refer to the *PCS 7* documentation).

6.3 Process data or fail-safe values

When are fail-safe values used?

The safety function requires that fail-safe values 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:

- During an F-Startup
- 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
- If F-I/O or channel faults are detected (e.g. wire break, short-circuit, or discrepancy error)
- As long as you have enabled an F-I/O passivation on the F-Channel driver at input PASS_ON

Fail-safe output for F-I/O/channels of an F-I/O

In the case of an F-I/O with inputs, the F-System provides fail-safe values at the F-Channel driver during passivation instead of the process data pending at the fail-safe inputs.

A fail-safe value of 0 is provided for (digital) channels of data type BOOL.

For analog channels, you must assign the fail-safe values at input SUBS_V of the F-Channel driver and enable them by assigning 1 to input SUBS_ON or select the last valid value as the fail-safe value by assigning 0 (default value) to input SUBS_ON.

⚠ WARNING
For F-I/O with inputs, the fail-safe value 0 provided at the F-Channel driver must be further processed for (digital) channels of data type BOOL in the safety program.

In the case of an F-I/O with outputs, the F-System transfers fail-safe values to the fail-safe outputs during passivation instead of the output values provided by the F-Channel driver.

Reintegration

The changeover from fail-safe values to process data (reintegration of an F-I/O) is executed either automatically or after user acknowledgment on the F-Channel driver.

The reintegration method depends on the following:

- Cause of passivation of the F-I/O/channels of the F-I/O
- Parameter to be assigned by you on the F-Channel driver

Note

For F-I/O with outputs, acknowledgment after F-I/O faults or channel faults may only be possible minutes after the fault has been eliminated due to necessary test signal inputs (see F-I/O manuals).

See also

F-Channel drivers for F-I/O (Page 266)

6.4 Group passivation

Description

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 output or PASS_ON input to perform a group passivation of 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 outputs of the F-Channel drivers in the group with F_OR4 F-Blocks and interconnect the result at the OUT output of F_OR4 with all PASS_ON inputs of the F-Channel drivers in the group.

See also

F-Channel drivers for F-I/O (Page 266)

Programming communication

7

7.1 Safety-related communication between F-CPUs

7.1.1 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 connection tables in *NetPro*, in the same way as with standard programs.

Note

In S7 F/FH Systems, safety-related communication via S7 connections is possible to and from the following F-CPUs:

- CPU 412-3H
- CPU 414-4H
- CPU 417-4H

⚠WARNING

Safety-related CPU-CPU communication is not permitted via public networks.

Creating an S7 Connection in the Connection Table

For each communication connection between two F-CPUs, you must create an S7 connection in the connection table in *NetPro*.

STEP 7 assigns a local ID and a partner ID for each connection end-point. If necessary, you can change the local ID in *NetPro*. You assign the local ID to the ID parameter of the appropriate F-Blocks in the safety programs.

Note

Safety-related communication via S7 connections to unspecified partners is not possible.

Procedure for Configuring S7 Connections

You configure the S7 connections for safety-related CPU-to-CPU communication the same way as for standard systems, or even as a fault-tolerant S7 connection, if necessary.

Note

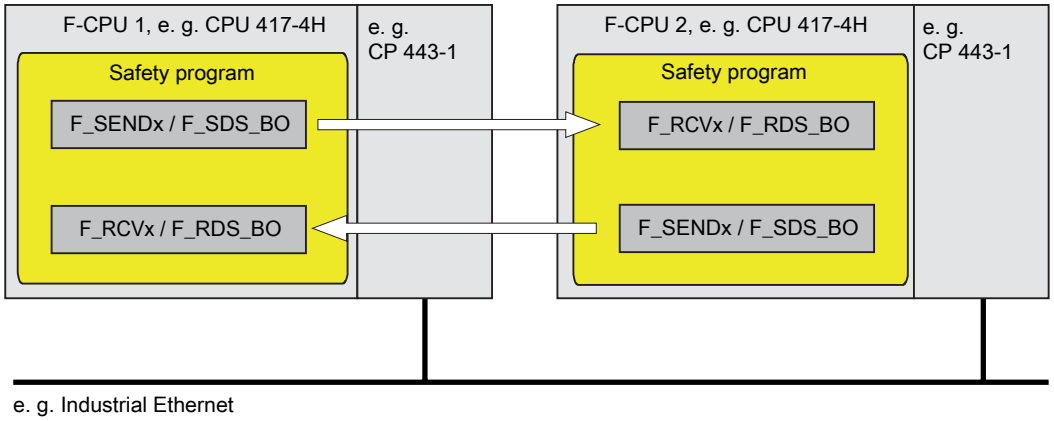
If you modify the configuration of S7 connections for safety-related communication, you must recompile the relevant S7 programs and download them to the F-CPUs.

Additional Information

You will find a description of how to configure S7 connections in the following sources:

- Manual " Configuring Hardware and Communication Connections STEP 7 V5.x (http://support.automation.siemens.com/WW/view/en/18652631) "
- Manual " Automation System S7-400H Fault-tolerant Systems (http://support.automation.siemens.com/WW/view/en/1186523) "
- *STEP 7 online help*

7.1.2 Communication via F_SENDBO/F_RCVBO, F_SENDR/F_RCVR, and F_SDS_BO/F_RDS_BO



You use the **F_SENDBO/F_RCVBO, F_SENDR/F_RCVR, and F_SDS_BO/F_RDS_BO** F-Communication blocks for sending and receiving data in a fail-safe manner via S7 connections.

This allows you to safely transfer a *fixed* number of up to 20 data elements of F-Data type F_REAL and up to 20/32 data elements of F-Data type F_BOOL.

7.1.3 Programming safety-related CPU-to-CPU communication via S7 connections

Requirements for Programming

The following requirements must be met prior to programming:

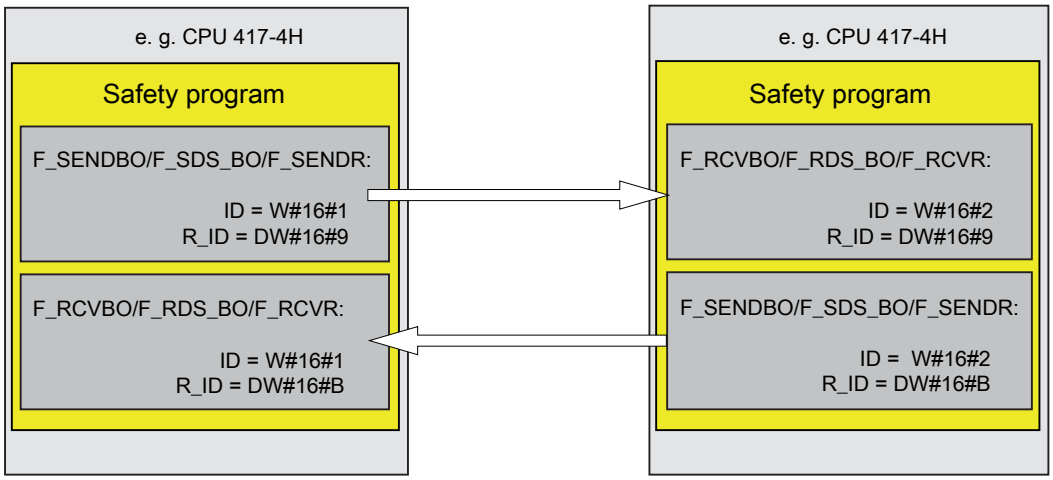
- The S7 connections between the relevant F-CPUs must be configured in *NetPro*
- Both CPUs must be configured as F-CPUs:
  - The "CPU contains safety program" option must be enabled

    and
  - The password for the F-CPU must be entered

Programming Procedure

1. In the safety program used to send data, insert the send F-Block F_SENDBO/F_SDS_BO/F_SENDR.
2. In the safety program used to receive data, insert the receive F-Block F_RCVBO/F_RDS_BO/F_RCVR.
3. Assign the local ID of the S7 connection (data type: WORD) configured in *NetPro* to the input ID of F_SENDBO/F_SDS_BO/F_SENDR.
4. Assign the local ID of the S7 connection (data type: WORD) configured in *NetPro* to the input ID of F_RCVBO/F_RDS_BO/F_RCVR.

5. Assign an odd number (data type: DWORD) to the R_ID inputs of F_SENDBO/F_SDS_BO/F_SENDR and F_RCVBO/F_RDS_BO/F_RCVR. This defines an association between F_SENDBO/F_SDS_BO/F_SENDR and F_RCVBO/F_RDS_BO/F_RCVR. The associated F-Blocks are given the same R_ID value.



WARNING

The value for each address association (input parameter R_ID; data type: DWORD) is user-defined; however, it must be unique from all other safety-related communication connections in the network. The value R_ID + 1 is internally assigned and must not be used.

6. Interconnect inputs SD_BO_xx and SD_R_xx of the F-Blocks F_SENDBO/F_SDS_BO/F_SENDR with the send signals.
7. Interconnect the outputs RD_BO_xx and RD_R_xx of the F-Blocks F_RCVBO/F_RDS_BO/F_RCVR with the F-Blocks for further processing of the received signals.
8. Assign the fail-safe values to be made available at the outputs RD_BO_xx or RD_R_xx to the inputs SUBBO_xx and SUBR_xx of the F-Blocks F_RCVBO/F_RDS_BO/F_RCVR:
   - While the connection between communication peers is being established for the first time after F-Startup of the F-Systems
   - Whenever a communication error occurs

9. Assign the required F-monitoring time to the TIMEOUT inputs of F_SENDBO/F_SDS_BO/F_SENDR and F_RCVBO/F_RDS_BO/F_RCVR.

⚠ WARNING

It can be ensured (from a fail-safe standpoint) that a signal level to be transferred will be detected on the sender side and transferred to the receiver only if the signal is pending for at least as long as the assigned F-Monitoring time (TIMEOUT).

For information about calculating F-Monitoring times, refer to the section entitled "Run times, F-Monitoring times, and response times (Page 410)".

Note

For safety reasons, parameter assignment at the TIMEOUT inputs must take place within the minimum F-Monitoring time. TIMEOUT must not be used to increase availability.

10. To reduce the bus load, you can temporarily shut down communication between the F-CPUs by assigning "0" (default = "1") to input EN_SEND of F_SENDBO/F_SDS_BO/F_SENDR. In this case, send data are no longer sent to the associated F_RCVBO/F_RDS_BO/F_RCVR, and the recipient F_RCVBO/F_RDS_BO/F_RCVR provides the assigned fail-safe values for this time period. If communication was already established between the connection partners, a communication error is detected.
11. Optional: Evaluate the ACK_REQ output of F_RCVBO/F_RDS_BO/F_RCVR in the standard user program, for example, in order to query or to indicate whether user acknowledgment is required.
12. Interconnect the ACK_REI input of F_RCVBO/F_RDS_BO/F_RCVR with the reintegration acknowledgement signal.
13. Optional: Evaluate output SUBS_ON of F_RCVBO/F_RDS_BO/F_RCVR or F_SENDBO/F_SDS_BO/F_SENDR to query whether F_RCVBO/F_RDS_BO/F_RCVR is outputting the fail-safe values you assigned at inputs SUBBO_xx/SUBR_xx.

14. Optional: Evaluate the ERROR output of F_RCVBO/F_RDS_BO/F_RCVR or F_SENDBO/F_SDS_BO/F_SENDR in the standard user program, for example, in order to query or to indicate whether a communication error has occurred.
15. Optional: Evaluate output SENDMODE of F_RCVBO/F_RDS_BO/F_RCVR to query whether the F-CPU with the associated F_SENDBO/F_SDS_BO/F_SENDR is in deactivated safety mode.

⚠ WARNING

If the F-CPU with the associated F_SENDBO/F_SDS_BO/F_SENDR is in deactivated safety mode, you can no longer assume that the data received from this F-CPU were generated safely. You must then implement organizational measures such as operation monitoring and manual safety shutdown to ensure safety in those portions of the system that are affected by the received data. Alternatively, you must output fail-safe values instead of the received data in the F-CPU with F_RCVBO/F_RDS_BO/F_RCVR by evaluating SENDMODE.

⚠ WARNING

The S7 program must be recompiled if the S7 connections for communication between F-CPUs have been changed

If the safety program contains F-Blocks for safety-related CPU-to-CPU communication, the S7 program involved in communication must be recompiled after the following actions in order to update the connection data:

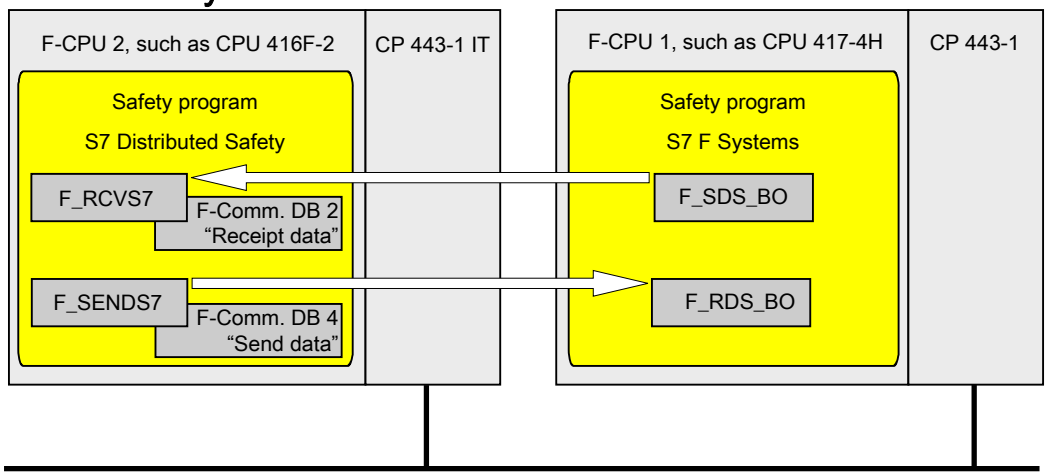
- Copying of an F-CPU
- Copying of a safety program or chart to another F-CPU
- Changing of a communication peer of an S7 connection
- Removal from/insertion into the multiproject of a project containing the communication peer of an S7 connection

See also

Safety engineering in SIMATIC S7 System Manual (http://support.automation.siemens.com/WW/view/en/12490443)

Determining the runtime sequence (Page 80)

7.2 Safety-related communication between S7 F Systems and S7 Distributed Safety



Procedure on the *S7 F Systems* side

On the *S7 F Systems* side, proceed as described in chapter " Safety-related communication between F-CPUs (Page 101) ".

Particularity:

Communication between *S7 F Systems* and *S7 Distributed Safety* is only possible on the *S7 F Systems* side with the F-Blocks F_SDS_BO/F_RDS_BO.

Procedure on the *S7 Distributed Safety* side

On the *S7 Distributed Safety* side, proceed as described in chapter "Safety-related communication via S7 communications" in Manual "S7 Distributed Safety - Configuring and Programming (http://support.automation.siemens.com/WW/view/en/22099875) ".

Particularity:

For communication between *S7 F Systems* and *S7 Distributed Safety*, you must create the F-DB with exactly 32 data elements of data type BOOL on the *S7 Distributed Safety* side.

Maintenance Override function

8

8.1 Maintenance Override concept

What is Maintenance Override?

Maintenance Override offers you the possibility of setting bypasses in the safety program of the OS. Starting with *S7 F Systems* V6.1, you can create a bypass for F_BOOL or F_REAL on up to three process signals. These bypasses can be made mutually exclusive as needed. You can also use Maintenance Override to modify the fail-safe values for the process signals and assign a parameter for the reset time so that the bypasses you have set will be reset automatically after this time.

Maintenance Override is based on Secure Write Command++ (SWC++). With SWC++, actions to modify parameters in the F-CPU from the WinCC OS are separated into:

Part in the F-CPU	• Protocol execution • Parameter receiving
Part in the WinCC OS	• Object that calculates the checksum • Control interface to confirm the transaction

Each of this actions is performed either in individual F-Blocks in the F-CPU or in individual objects in the WinCC OS. The SWC++ protocol is an extension of the SWC protocol of *S7 Safety Matrix* V6.1.

For Maintenance Override, *S7 F Systems* V6.1 offers:

- F_SWC_BO: Maintenance Override for data type F_BOOL
- F_SWC_R: Maintenance Override for data type F_REAL
- F_SWC_P: Centralized control of operator input via the OS
- SWC_MOS: Establishes the connection to the WinCC faceplate.
- SWC_TR chart-in-chart for time-controlled Maintenance Override
- The associated faceplates that you must integrate in your OS

For more information on these blocks and F-Blocks, refer to the section entitled "Blocks and F-Blocks for data conversion (Page 236)".

Note

When used with *PCS 7*, one PO license is used for each instance of the SWC_MOS block in the safety program.

Operator types for Maintenance Override

A transaction with Maintenance Override is performed in the OS by means of a faceplate. The transaction consists of a sequence of operations that can be performed by one or two operators.

8.2 Programming Maintenance Override

8.2.1 Basic procedure

Basic procedure

Proceed as follows to perform Maintenance Override by means of an OS:

On the engineering station (ES)

1. Position the SWC_MOS block and the F-Blocks F_SWC_BO/F_SWC_R and F_SWC_P in your *CFC chart* and interconnect them.

   For more information, refer to the section entitled "Positioning, interconnecting, and assigning parameters to F-Blocks in the CFC chart (Page 111)".

2. Configure the faceplate for MOS.

   For more information, refer to the section entitled "Configuring a faceplate for Maintenance Override (Page 120)".

On the operator station (OS)

- Set up a bypass with Maintenance Override for maintenance on the F-Channel drivers and change the fail-safe value if necessary.

For more information, refer to the section entitled "Operating Maintenance Override (Page 124)".

8.2.2 Positioning, interconnecting, and assigning parameters to F-Blocks in the CFC chart

8.2.2.1 Introduction

Introduction

The following sections will show you typical applications for Maintenance Override. They contain information on how to position, interconnect, and assign parameters to blocks and F-Blocks in CFC charts for Maintenance Override.

In the following sections, you will find applications for:

- Application: Simulating an F-Channel driver (Page 112)
- Application: Grouped Maintenance Override with mutually exclusive interlock (Page 114)
- Application: Time-controlled Maintenance Override (Page 116)
- Application: Maintenance Override with logic blocks (Page 118)

Note

The creation of F-Block types with the Maintenance Override function is not supported.

Using a key-operated switch

To ensure that only authorized persons carry out operations, you can connect the F_SWC_P F-Block to a key-operated switch at the EN_SWC input.

During an operation, the EN_SWC input must be set to one (EN_SWC = 1). If the input is reset to zero (EN_SWC = 0) after an operation, all existing bypasses are disabled. However, all fail-safe value settings are retained.

⚠ WARNING

The "Maintenance Override" functionality allows changes to the safety program to be made during RUN mode.

As a result, the following safety measures are required:

- Make sure that operations that could compromise plant safety cannot be carried out. You can use the EN_SWC input on the F_SWC_P F-Block for this purpose, for example, by controlling it with a key-operated switch or on a process-specific basis via the safety program.
- Make sure that only authorized persons can carry out operations. Examples:
  - Control the EN_SWC input on the F_SWC_P F-Block with a key-operated switch.
  - Set up access protection at operator stations where the "Maintenance Override" function can be performed.

8.2.2.2 Application: Simulating an F-Channel driver

Application

This application shows you how to simulate an F-Channel driver with Maintenance Override.

Procedure

⚠ WARNING
Warnings in the descriptions of F-Blocks Make sure that you comply with the warnings in the descriptions of the F-Blocks F_SWC_BO and F_SWC_R.

1. Position the SWC_MOS block in your CFC chart. Make sure that you follow the name assignment instructions in the section entitled "SWC_MOS: Command function for Maintenance Override (Page 264)".
2. If necessary, position the F_SWC_P F-Block.
3. Position one F_SWC_BO F-Block to start or stop the simulation.
4. Position one F_SWC_BO or F_SWC_R F-Block to modify the simulation value, if you wish.
5. Connect the EN_SWC input on the F_SWC_P F-Block to a key-operated switch.
6. At the MAX_TIME input on the F_SWC_P F-Block, assign the maximum duration of an operation (default setting is one minute).
7. On the F_SWC_BO F-Block used to start the simulation, connect the:
   - OUT output to the SIM_ON input on the F-Channel driver
   - AKT_VAL output to the AKT_B1 input on the SWC_MOS block
8. On the F_SWC_BO or F_SWC_R F-Block used to modify the simulation value, connect the:
   - OUT output to the SIM_I or SIM_V input on the F-Channel driver
   - AKT_VAL output to the AKT_V_B or AKT_V_R input on the SWC_MOS block
9. Optional:

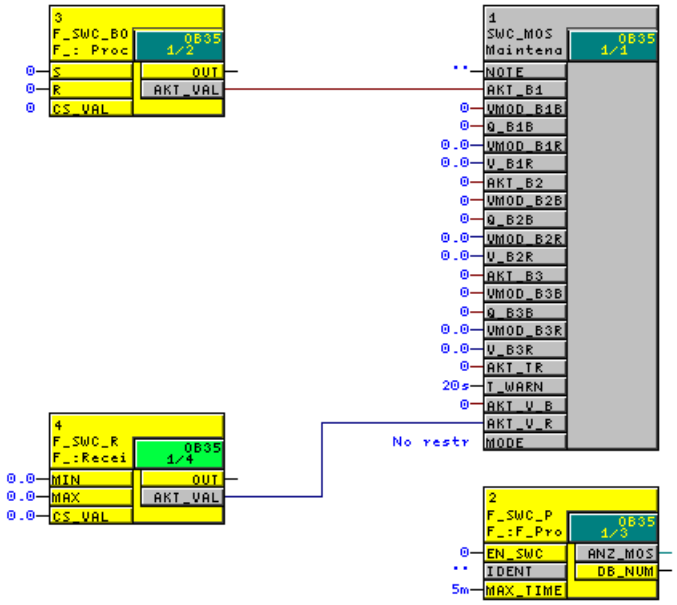
   At the MIN and MAX inputs on the F_SWC_R F-Block, assign a lower limit and upper limit, respectively, for the fail-safe value (default settings: 0.0 and 100.0). If necessary, assign parameters for the CS_VAL input on the F_SWC_R F-Block.
10. Optional:

    If you want to display the current value of an F-I/O in the faceplate when a bypass is enabled, connect the Q_MOD and V_MOD outputs on the F-Channel driver to the V_MOD_B1B and V_MOD_B1R inputs, respectively, on the SWC_MOS block.
11. Optional:

    If you want to display the process value and its QUALITY on the F-Channel driver in the faceplate, connect the Q_DATA and V_DATA outputs on the F-Channel driver to the Q_B1B and V_B1R inputs, respectively, on the SWC_MOS block.

12. Before compiling, make sure the SWC_MOS block assignment is correct. This block must be assigned to a standard runtime group.
13. Compile your CFC chart.

    During the compilation process, additional connections are made between the SWC_MOS block, the F_SWC_BO/F_SWC_R and F_SWC_P F-Blocks, and the F-Channel drivers.



14. Continue as described in the section entitled "Configuring a faceplate for Maintenance Override (Page 120)".

8.2.2.3 Application: Grouped Maintenance Override with mutually exclusive interlock

Application

This application shows you how to create a grouped Maintenance Override.

Procedure

⚠ WARNING
Warnings in the descriptions of F-Blocks Make sure that you comply with the warnings in the descriptions of the F-Blocks F_SWC_BO and F_SWC_R.

1. Position the SWC_MOS block in your CFC chart. Make sure that you follow the name assignment instructions in the section entitled "SWC_MOS: Command function for Maintenance Override (Page 264)".
2. If necessary, position the F_SWC_P F-Block.
3. Position two or three F_SWC_BO F-Blocks to start or stop the simulation.
4. If necessary, position one F_SWC_BO or F_SWC_R F-Block to modify the simulation value.
5. Connect the EN_SWC input on the F_SWC_P F-Block to a key-operated switch.
6. At the MAX_TIME input on the F_SWC_P F-Block, assign the maximum duration of an operation (default setting is one minute).
7. On the F_SWC_BO F-Blocks used to start the simulation, connect the:
   - OUT outputs to the SIM_ON inputs on the associated F-Channel drivers
   - AKT_VAL outputs to the AKT_Bx inputs on the SWC_MOS block
8. On the F_SWC_BO or F_SWC_R F-Block used to modify the simulation value, connect the:
   - OUT outputs to the SIM_I or SIM_V inputs on the F-Channel drivers
   - AKT_VAL output to the AKT_V_B or AKT_V_R input on the SWC_MOS block
9. On the SWC_MOS block, assign the input MODE = 'MutualExclBypass' to enable the mutually exclusive interlock.
10. Optional:

    At the MIN and MAX inputs on the F_SWC_R F-Block, assign a lower limit and upper limit, respectively, for the fail-safe value (default settings: 0.0 and 100.0). If necessary, assign the CS_VAL input on the F_SWC_R F-Block.
11. Optional:

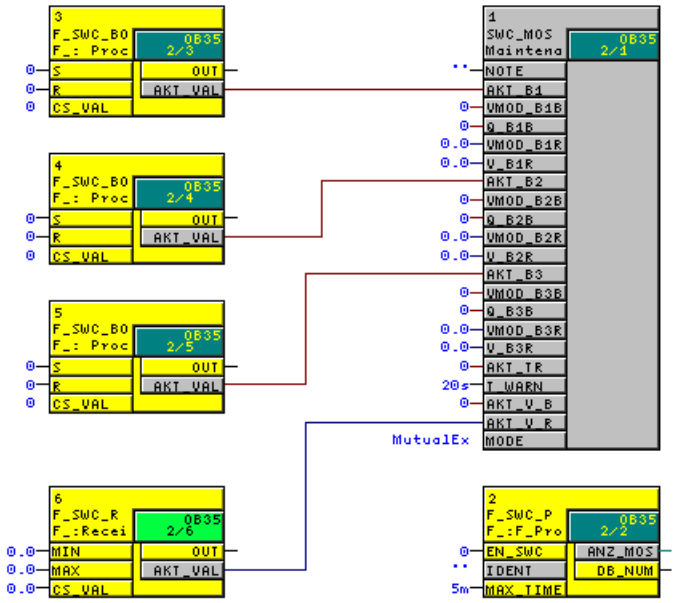
    If you want to display the current value of an F-I/O in the faceplate when a bypass is enabled, connect the Q_MOD and V_MOD outputs on the F-Channel driver to the V_MOD_BxB and V_MOD_BxR inputs, respectively, on the SWC_MOS block.

12. Optional:

    If you want to display the process value and its QUALITY on the F-Channel driver in the faceplate, connect the Q_DATA and V_DATA outputs on the F-Channel driver to the Q_BxB and V_BxR inputs on the SWC_MOS block.

13. Compile your CFC chart.

    During the compilation process, additional connections are made between the SWC_MOS block, the F_SWC_BO/F_SWC_R and F_SWC_P F-Blocks, and the F-Channel drivers.



14. Continue as described in the section entitled "Configuring a faceplate for Maintenance Override (Page 120)".

8.2.2.4 Application: Time-controlled Maintenance Override

Application

This application shows you how to create a time-controlled Maintenance Override.

Procedure

⚠WARNING

Warnings in the descriptions of F-Blocks

Make sure that you comply with the warnings in the descriptions of the F-Blocks F_SWC_BO and F_SWC_R.

1. Position the SWC_MOS block in your CFC chart. Make sure that you follow the name assignment instructions in the section entitled "SWC_MOS: Command function for Maintenance Override (Page 264)".
2. If necessary, position the F_SWC_P F-Block.
3. Position one or more F_SWC_BO F-Blocks to start or stop the simulation.
4. Position one F_SWC_BO or F_SWC_R F-Block to modify the simulation value.
5. Position the SWC_TR chart-in-chart.
6. Connect the EN_SWC input on the F_SWC_P F-Block to a key-operated switch.
7. At the MAX_TIME input on the F_SWC_P F-Block, assign the maximum duration of an operation (default setting is one minute).
8. On the F_SWC_BO F-Blocks used to start the simulation, connect the:
   - OUT outputs to the SIM_ON inputs on the associated F-Channel drivers
   - AKT_VAL outputs to the AKT_Bx inputs on the SWC_MOS block
9. On the F_SWC_BO or F_SWC_R F-Block used to modify the simulation value, connect the:
   - OUT outputs to the SIM_I or SIM_V inputs on the F-Channel drivers
   - AKT_VAL output to the AKT_V_B or AKT_V_R input on the SWC_MOS block
10. Connect the AKT_TR output of the SWC_TR chart-in-chart to the AKT_TR input on the SWC_MOS block.
11. Optional:

    At the MIN and MAX inputs on the F_SWC_R F-Block, assign a lower limit and upper limit, respectively, for the fail-safe value (default settings: 0.0 and 100.0). If necessary, assign the CS_VAL input on the F_SWC_R F-Block.
12. Optional:

    On the SWC_MOS block, set the input MODE = 'MutualExclBypass' to enable the mutually exclusive interlock.

13. Optional:

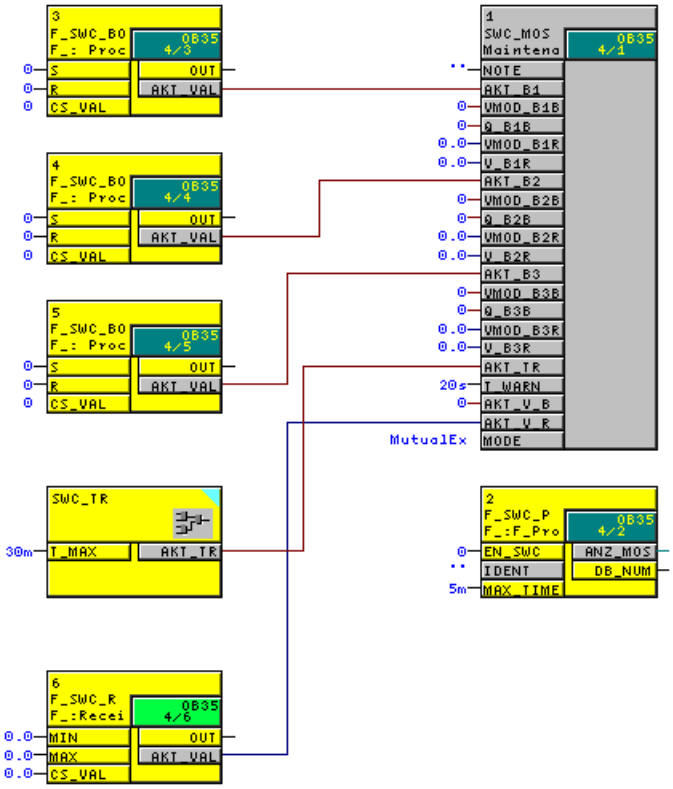
    If you want to display the current value of an F-I/O in the faceplate when a bypass is enabled, connect the Q_MOD and V_MOD outputs on the F-Channel driver to the V_MOD_BxB and V_MOD_BxR inputs, respectively, on the SWC_MOS block.

14. Optional:

    If you want to display the process value and its QUALITY on the F-Channel driver in the faceplate, connect the Q_DATA and V_DATA outputs on the F-Channel driver to the Q_BxB and V_BxR inputs on the SWC_MOS block.

15. Compile your CFC chart.

    During the compilation process, additional connections are made between the SWC_MOS block, the F_SWC_BO/F_SWC_R and F_SWC_P F-Blocks, and the F-Channel drivers.



16. Continue as described in the section entitled "Configuring a faceplate for Maintenance Override (Page 120)".

8.2.2.5 Application: Maintenance Override with logic blocks

Application

This application shows you how to control a signal in your system using Maintenance Override in conjunction with a control signal from your system.

Procedure

⚠WARNING

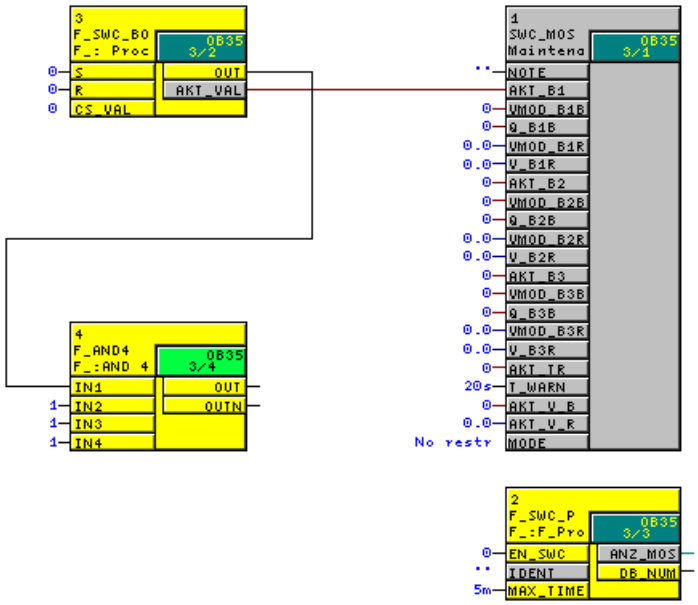
Warnings in the descriptions of F-Blocks

Make sure that you comply with the warnings in the descriptions of the F_SWC_BO F-Block.

1. Position the SWC_MOS block in your CFC chart. Make sure that you follow the name assignment instructions in the section entitled "SWC_MOS: Command function for Maintenance Override (Page 264)".
2. If necessary, position the F_SWC_P F-Block.
3. Position one F_SWC_P F-Block and one F_AND4 F-Block.
4. Connect the EN_SWC input on the F_SWC_P F-Block to a key-operated switch.
5. At the MAX_TIME input on the F_SWC_P F-Block, assign the maximum duration of an operation (default setting is one minute).
6. On the F_SWC_BO F-Block, interconnect the:
   - OUT to the INx input on the F_AND4 F-Block
   - AKT_VAL output to the AKT_B1 input on the SWC_MOS block
7. Interconnect the INy input of the F_AND4 F-Block to the *controlling* signal from your system.
8. Interconnect the OUT output of the F_AND4 F-Block to the signal *to be controlled* from your system.

9. Compile your CFC chart.

   During the compilation process, additional connections are made between the SWC_MOS block, and the F_SWC_BO and F_SWC_P F-Blocks.



10. Continue as described in the section entitled "Configuring a faceplate for Maintenance Override (Page 120)".

8.2.3 Configuring a faceplate for Maintenance Override

A faceplate must be created in the OS for each instance of an SWC_MOS block in the safety program. The operator steps for Maintenance Override are performed on the faceplate in the required order by one or two operators. The associated block icon in the OS is used to call the corresponding faceplate.

⚠ WARNING

You can edit the faceplates for Maintenance Override.

If you encounter constraints, you can restore a backup copy of the respective file/function from the "Extras\FSYSTEMSHMI" directory on the product CD.

Requirements

- All required F_SWC_R and F_SWC_BO F-Blocks are placed, assigned parameters, and interconnected in the CFC charts of the safety program.

  For more information, refer to the section entitled "Positioning, interconnecting, and assigning parameters to F-Blocks in the CFC chart (Page 111)".
- The CFC charts with the SWC_MOS F-Blocks are located in the plant hierarchy.

Configuring faceplates on the ES

Configure the faceplates for Maintenance Override on the ES with the following steps:

1. Create block icons.
2. Initialize properties of the block icons.
3. Set up authorizations for operators.
4. Transfer the configuration to the OS.

The individual steps are described below.

Creating block icons

1. Open the *PCS 7* project in *SIMATIC Manager*.
2. Create a new picture object in the level of the plant hierarchy containing the CFC charts with the SWC_MOS F-Blocks.
3. For use with *PCS 7*:
   - Select the picture object and open the object properties.
   - In the "Block Icons" tab, select the "Derive block icons from the plant hierarchy" option.
4. Click "OK" or "Apply" to confirm the revised properties.
5. Select the OS object and select "Compile" on the shortcut menu to compile the OS.

   If necessary, enable the "Compile OS" wizard when selecting the data to be compiled. If you are using *PCS 7* V7.0 or lower, select the "Generate/update block icons" option when selecting the scope of the compilation.

   Click the "Compile" button in the last dialog box.

Result: When the OS is compiled, the block icons are automatically inserted into the new picture.

Initializing properties of the block icons

1. Double-click the picture file in the Plant view of the *PCS 7* project.

   Result: WinCC Explorer is started and the picture file is displayed in the Graphics Designer. The name is displayed in the header of each block icon. The name of the block icon is formed from the name of the CFC chart and the name of the associated F-Block instance.

2. Select a block icon and open the object properties.
3. In the "Properties" tab, select "User configuration".
4. Assign the required authorizations to the "LevelInitiate", "LevelConfirm", "LevelBypass", and "LevelBypassValue" attributes. Alternatively, accept the default operator authorizations. See also the section entitled "Setting up user authorizations for operators".

   Default authorizations (correspond to the user hierarchies from *PCS 7*):

   – For the user who initiates a bypass or fail-safe value change (Initiator): No. 5, Operator-process communications
   – For the user who only initiates a bypass using Maintenance Override (Bypass): No. 5, Operator-process communications
   – For the user who initiates a fail-safe value change using Maintenance Override (BypassValue): No. 5, Operator-process communications
   – For the user who confirms the bypass and a fail-safe value change using Maintenance Override (Confirmer): No. 6, Higher-order operator-process communications

5. Repeat Steps 2 and 4 for all available block icons.
6. Save the picture file.

Setting up user authorizations for operators

Maintenance Override is performed by two operators. Create two users:

- The initiator initiates the bypass and/or the setting of bypass values.
- The confirmer confirms the bypass and/or the setting of bypass values.

Alternatively, both steps can be performed by only one user. For this, create a user with the Initiator *and* Confirmer authorizations.

Create the users in WinCC Explorer using the "User Administrator" editor according to the following table.

User	Action	Required authorizations			
		Initiator	Confirmer	Bypass	BypassValue
Initiator	Sets bypasses	X	—	X	—
	Sets bypass values	X	—	—	X
	Sets bypasses and bypass values	X	—	X	X
Confirmer	Confirms bypasses	—	X	X	—
	Confirms bypass values	—	X	—	X
	Confirms bypasses and bypass values	—	X	X	X
Initiator & confirmer	Sets and confirms bypasses	X	X	X	—
	Sets and confirms bypass values	X	X	—	X
	Sets and confirms bypasses and bypass values	X	X	X	X

Activating the OS

Activate the WinCC Runtime system of the OS, e.g., by selecting **File > Activate** in WinCC Explorer.

Result

Once the WinCC Runtime system is activated, the hierarchy levels appear as buttons in the runtime system of the OS. Click the button to display the block icons for this level.

Example

In the following figure, you can see two block icons in the runtime system of the OS.

Clicking a block icon opens the faceplate. For maintenance, you can use Maintenance Override to set up a bypass on the F-Channel drivers.

You can identify an enabled bypass by the B symbol in the block icon.

Detailed information

For detailed information on the described steps, refer to:

- Configuration Manual "PCS 7 Operator Station (http://support.automation.siemens.com/WW/view/en/27002758)"
- Online Help for the WinCC editors (e.g., Graphics Designer and User Administrator)

8.2.4 Integrating Maintenance Override into and existing project

Introduction

You can also integrate the Maintenance Override function into and existing project.

Requirements

To integrate Maintenance Override into an existing project, you must update your project.

Updating an existing project

1. Launch *WinCC* Explorer for the OS contained in the project.
2. Open the OS Project Editor.
3. Make sure the "@PCS7Typicals_S7F_SDW.PDL" picture – if already present in the project – is selected in the "Accept faceplates from libraries" area of the "Basic Data" tab.

   User-specific changes in this picture will be lost.
4. Make sure all other settings in the OS Project Editor correspond to your specifications.
5. Now click "OK".

The project is reconfigured and, as a result, the new block icon is received along with the new pictures.

Integrating Maintenance Override

In order to introduce the new block icon into existing plant pictures, you must recompile the relevant project.

1. Start SIMATIC Manager.
2. If you are working with *PCS 7* V7.0 or earlier:

   Make sure that the "Derive block icons from the plant hierarchy" option is selected in the "Block icons" tab of the object properties for the relevant picture object.

   Note

   If user settings for the block icon of a Maintenance Override are to be retained during the subsequent OS compilation of an existing picture, you must clear the "Derive block icons from the plant hierarchy" option for this *WinCC* picture.

3. Select the OS object and select "Compile" on the shortcut menu to compile the OS.
4. Click the "Compile" button in the last dialog of the "Compile OS" wizard.

Result

Once you have performed these steps, your project contains the new Maintenance Override block icon and the required pictures.

8.3 Operating Maintenance Override

8.3.1 Requirements and general instructions

A bypass with Maintenance Override is created in the OS by means of a faceplate. The bypass is switched on and off through a sequence of operations that must be performed by one or two operators.

Requirements

- The S7 program is compiled and downloaded to the F-CPU.
- The user(s) with the relevant authorizations are set up.
- If you are using *PCS 7* V7.0 or later, the pictures/faceplates of the Maintenance Override function must be converted.
- The configuration of the faceplates is compiled and downloaded to the OS.
- When using OS clients, make sure that no default server is set for tags (in WinCC Explorer select "Server Data," in the shortcut menu select "Default Server" and in the "Configure Default Server" dialog for the "Tags" component select "No Default Server").

General information

⚠WARNING

Initiator and confirmer must not accept an invalid value

Before starting the transaction, you must verify the technological assignment of the faceplate. This consists of the technological name in the header and the CPU identification.

As the initiator or confirmer, you must not accept an invalid value. If any inconsistencies arise, you must cancel the operation for setting the bypass on the F-Channel drivers.

As an operator, you must not rely on individual display fields of the faceplate; rather, you must check the values and compare them among each other.

⚠WARNING

Technological assignment must be appropriate for the environment

When opening the faceplate, make sure that the technological assignment in the top line is appropriate for the environment in which the block icon was placed.

⚠WARNING

Transaction for changing an F-Parameter

You can only perform one transaction for changing an F-Parameter at a time. You must use organizational measures to ensure that multiple transactions are not performed simultaneously for the same F-Parameter. Otherwise, the transaction cannot be performed correctly, resulting in unexpected results, such as:

- Display of incorrect values in the faceplate fields
  or
- Unexpected cancellation of the transaction

If an operation is already active

If an operation for another faceplate is already in progress, the message "Other command function active" appears when opening the faceplate in WinCC Runtime.

8.3.2 Bypass on the F-Channel driver with two operators

Operator Authorizations

Two operators having different authorizations are required to create a bypass.

- The initiator initiates the bypass on the F-Channel driver. This user must have the authorization for initiating the Bypass, LevelBypass, and LevelBypassValue but not for confirming them. The authorization corresponds to the "InitiatorAuthorization" attribute in the properties for the block icon. The default setting is No. 5, Operator-process communications.
- The confirmer verifies and confirms the change. This operator must have the necessary authorization for confirming the change, LevelBypass, and LevelBypassValue but not for initiating them. The authorization corresponds to the "ConfirmerAuthorization" attribute in the properties for the block icon. The default setting is No. 6, Higher-level operator-process communications.

Reset time

If you have configured a retrigger function in the CFC charts, the simulation is only enabled for the time configured at the T_MAX input of the SWC_TR chart-in-chart. As the initiator, if you click the "Retrigger" button while the configured reset time is elapsing, the reset time restarts with the configured time after the change is confirmed by the confirmer.

Quality of the process value on the F-Channel driver

The quality of the process value on the F-Channel driver is indicated in the faceplate by the following symbols:

Symbol	State	Quality code
No symbol	Valid value	16#80
	Simulation	16#60
	SUBSTITUTION VALUE	16#48
	Last valid value	16#44
	Invalid value (F-STOP)	16#00

See also the section entitled "F-Channel drivers for F-I/O (Page 266)".

Value on the F-Channel driver

If the V_MOD_Bx inputs are interconnected on the SWC_MOS block, the values on the F-Channel drivers are displayed under V_MOD.

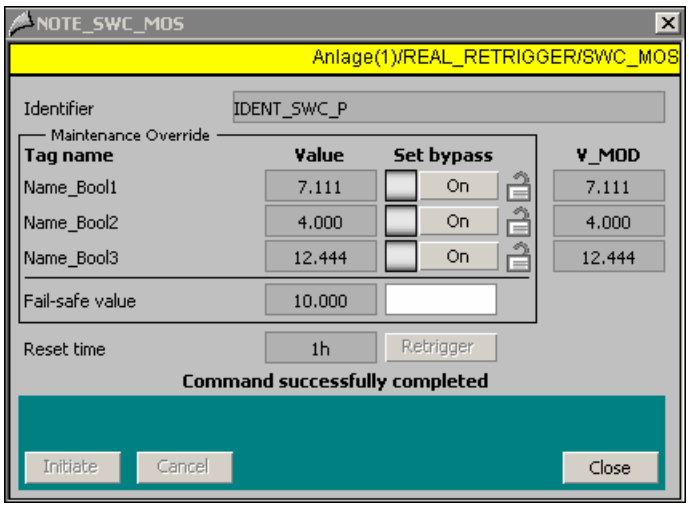
Note

The sections below describe the necessary transaction steps for the two operators. The figures illustrate the example of an F_REAL parameter with the login:

- level5 – Initiator
- level6 – Confirmer

Initiator: Initiating a bypass

1. Log on to the OS as a user with "Initiator" authorization.
2. Click the desired block icon to open the faceplate.



Under "Value" on the Maintenance Override faceplate, you can see the current process value on the F-Channel drivers and the current fail-safe value setting. The values on the F-Channel drivers are displayed in the V_MOD column.

The symbols under "Set Bypass" show you the current status of the bypass (SIM_ON) on the F-channel drivers:

Symbol	Meaning
	Bypass not active
B	Bypass active
	A bypass can be created for this F-Channel driver.
	For this F-Channel driver, either a bypass cannot be created (mutually exclusive interlock) or the user authorization is insufficient.

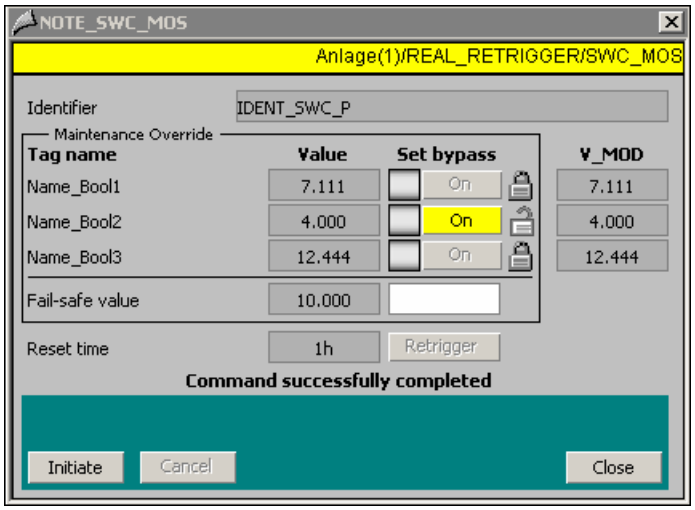
1. To enable a bypass for one or more F-Channel drivers, click the corresponding button under "Set Bypass".

   If the input setting MODE = 'MutualExclBypass' has been assigned on the SWC_MOS block, the remaining F-Channel drivers are interlocked when a bypass is enabled. The interlocked F-Channel drivers are indicated by a lock symbol ( ).

2. If you want to change the current fail-safe value on F-Channel drivers for F_BOOL, click the button under "Set Bypass".

   If you are using F-Channel drivers for F_REAL and want to change the fail-safe value, enter the new fail-safe value in the text box and confirm your entry by pressing Enter. The configured Min/Max values are evaluated in the process.

3. If you want to reset the reset time to the configured initial value, click the "Retrigger" button.



4. Click the "Initiate" button.

The confirmer must then continue the transaction.

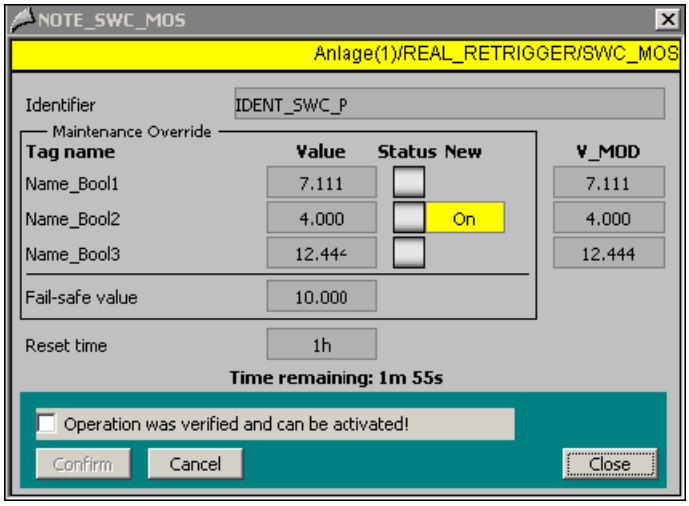
If you cancel the transaction after clicking "Initiate", check whether the previously valid value is displayed in the "Value" field.

Confirmer: Confirming a bypass

1. Log on to the OS as a user with "Confirmer" authorization.

   You can log on to a second OS or on the same OS as the initiator.

2. Click the desired block icon to open the faceplate.



3. Verify that:

- The correct F-CPU has been selected (for ID, refer to section entitled "F_SWC_P: Centralized control of operator input via the OS (Page 237)").
- The correct parameter is going to be changed (tag name).
- The change (modified value) is displayed correctly.
- New values of the modified parameters are highlighted in yellow under "New".
- No other fields for new values are highlighted in yellow.

1. Confirm the change with "Operation has been checked and should be activated" or cancel the operation by clicking "Cancel".

   Note

   Do not close faceplate

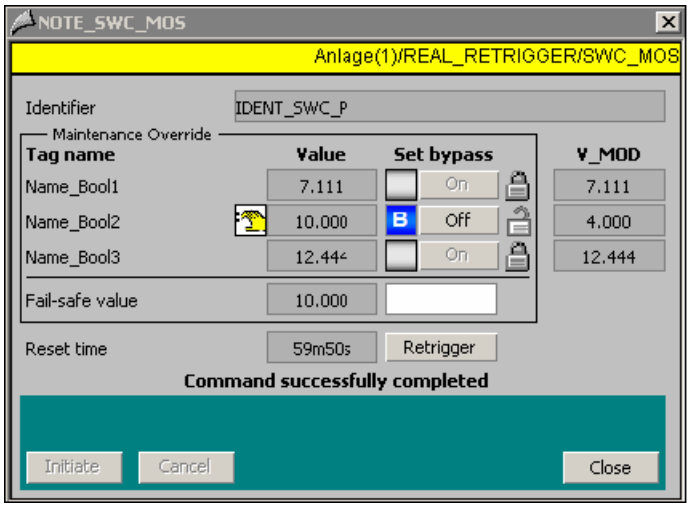
   You must not close this faceplate until you have performed step 5. If you do close the faceplate, the transaction cannot be continued.

2. Click "Confirm" to enable the bypass. Click "Cancel" to cancel the operation.

Result

The successful change on the F-Channel drivers is signaled. The F-Channel driver on which the bypass has been enabled is indicated by this symbol: B. Depending on the interconnection on SWC_MOS, additional status displays become visible (see section entitled "SWC_MOS: Command function for Maintenance Override (Page 264)").

If you have configured a reset time, the countdown for this time begins. Bypasses are automatically canceled when the reset time has elapsed.



8.3.3 Bypass on the F-Channel driver with one operator

Operator Authorization

If only one operator is to perform the bypass on the F-Channel driver, this operator must be authorized to both initiate and confirm the bypass.

You must therefore create a user who has "LevelInitiate", "LevelConfirm", and "LevelBypass" assigned in the properties for the block icon. For more information, refer to the section entitled "Configuring a faceplate for Maintenance Override (Page 120)".

Creating a bypass with only one operator

The procedure is the same as for operation with two operators, except that one operator is able to perform all of the steps (see also the section entitled "Bypass on the F-Channel driver with two operators (Page 126)").

The difference is that it is no longer necessary to wait for the confirmer; instead, the operator can immediately check and confirm the operation after clicking the "Initiate" button.

All other steps remain the same.

Safety Data Write function

9

9.1 Safety Data Write concept

What is Safety Data Write?

The "Safety Data Write" functionality enables safety-related changes to be made to F-Parameters in the safety program of an F-CPU from an operator station (OS).

A special safety protocol is used for changing F-Parameters during safety mode operation. This ensures compliance with the safety requirements of Safety Integrity Level SIL1 to SIL3 in accordance with IEC 61508. The modified F-Parameter values can be retained even after a warm restart of S7 F/FH Systems.

The *S7 F Systems* optional software offers the following for Safety Data Write:

- Two F-Blocks that you must integrate in the CFC charts of your safety program
  - F_CHG_R: Safety Data Write for F-Parameters of data type F_REAL
  - F_CHG_BO: Safety Data Write for F-Parameters of data type F_BOOL
- The associated faceplates that you must integrate in your OS

Transaction for Safety Data Write

Safety Data Write allows an F-Parameter in the safety program of an F-CPU to be changed, provided a certain operating sequence is carried out in the OS within a certain time. The entire change operation is referred to as a "transaction".

Operator Types for Safety Data Write

A transaction can be performed by an individual operator who initiates, verifies, and confirms the change. However, a transaction can also be performed by two operators. One operator (the initiator) initiates the change, and the second (the confirmer) re-enters, verifies, and confirms this value.

9.2 Programming Safety Data Write

9.2.1 Basic procedure

Basic procedure

To implement Safety Data Write by means of an OS, you must perform the following steps:

On the ES

1. Insert the F-Blocks F_CHG_R and F_CHG_BO into the *CFC chart* and interconnect them.
2. Configure the faceplate for Safety Data Write.

On the operator station (OS)

- Change the F-Parameters with Safety Data Write.

The individual steps are described in detail in the sections below.

9.2.2 Positioning, interconnecting, and assigning parameters to F-Blocks in the CFC chart

Application

You can make changes to F-Parameters of the safety program by means of Safety Data Write using the F-Blocks F_CHG_R and F_CHG_BO.

Procedure

⚠WARNING

Warnings in the descriptions of F-Blocks

Make sure that you comply with the warnings in the descriptions of the F_CHG_R and F_CHG_BO F-Blocks.

1. Insert one F_CHG_R or F_CHG_BO F-Block, respectively, for each input of data type F_REAL or F_BOOL that is to be changed using Safety Data Write (see Example 1: F_CHG_R (Page 135) and Example 2: F_CHG_BO (Page 135)).
2. Interconnect the OUT output to the input whose value you want to change using Safety Data Write.
3. Assign a pair of numbers to the SAFE_ID1 and SAFE_ID2 inputs. This ensures the association between the instance of F_CHG_R/F_CHG_BO and the corresponding faceplate. SAFE_ID1 must be unique from all others in the program. The pair of numbers for SAFE_ID1 and SAFE_ID2 must be unique from all others in the system. You must configure the same pair of numbers on the block icon of the associated faceplate.
4. Interconnect the EN_CHG input to the enable signal for Safety Data Write.
5. Assign the maximum permissible time for the duration of the transaction to the TIMEOUT input. The transaction starts as soon as the initiator has accepted his entry.

   All steps for verifying the transaction must be taken into account when configuring this time. For example, if two operators are required to enable the change, an appropriate amount of time must be allotted for both operators to log on and perform the necessary steps.
6. For F_CHG_R only: Assign limit values to the MIN and MAX inputs to specify the time during which the F-Parameters (output OUT) can be changed.
7. For F_CHG_R only: Assign the value of the maximum permissible increment of the change to the MAXDELTA input to specify the amount by which the F-Parameter (output OUT) can change relative to the current existing value.
8. Assign the initial value to input CS_VAL that is to be applied to output OUT in the event of a cold restart.
   For F_CHG_R only: When a cold restart occurs, CS_VAL is applied at output OUT irrespective of the values for MIN and MAX. The configured value at input CS_VAL must be between the MIN and MAX values.
9. Optional: Assign 0 to input WS_MODE if the value at input CS_VAL is also to be applied to output OUT during a warm restart. The default value of input WS_MODE is 1.

10. Optional: Evaluate the CS_USED output in the safety program if you need to respond differently after an F-Startup in your safety program depending on whether the CS_VAL value or the last valid value at the OUT output has been made available.
11. For F_CHG_R only: Set the unit of measurement for the F-Parameter to be changed.

    To do so, open the properties for the F-Block and select output CURR_R in the "Outputs" tab. In the "Unit" field, select the desired unit of measurement (e.g., kg/min) from the drop-down list.

    The unit is displayed on the faceplate in the OS.

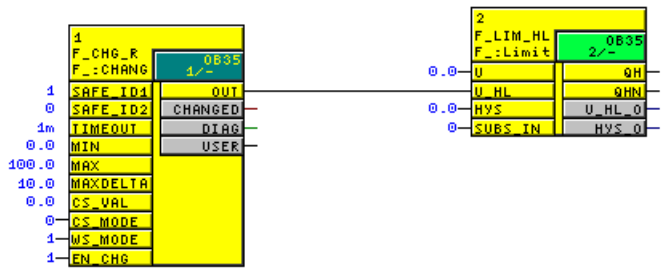
See also

CFC for S7 Continuous Function Chart
(http://support.automation.siemens.com/WW/view/en/21401430)

9.2.3 Examples: Safety Data Write

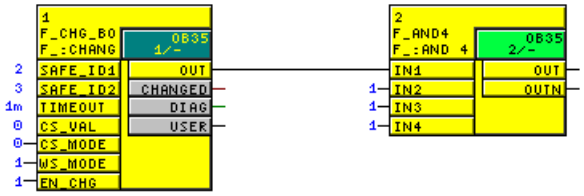
9.2.3.1 Example 1: F_CHG_R

The following figure shows an instance of F_CHG_R. The OUT output is interconnected to the "U_HL" input of F_LIM_HL whose value is to be changed in a fail-safe manner using Safety Data Write.



9.2.3.2 Example 2: F_CHG_BO

The following figure shows an instance of F_CHG_BO. The OUT output is interconnected to the "IN1" input of F_AND4 whose value is to be changed in a fail-safe manner using Safety Data Write.



9.2.4 Configuring the Faceplate for Safety Data Write.

A faceplate must be created in the OS for each instance of an F-CHG_R or F_CHG_BO F-Block in the safety program. The operator steps for the Safety Data Write transaction are performed on the faceplate in the required order by one or two operators. The associated block icon in the OS is used to call the corresponding faceplate.

Requirements

- All required F_CHG_R and F_CHG_BO F-Blocks are placed, assigned parameters, and interconnected in the CFC charts of the safety program.
- The CFC charts with the F_CHG_R and F_CHG_BO F-Blocks are located in the plant hierarchy.
- The safety program is compiled.

Configuring faceplates on the ES

Configure the faceplates for Safety Data Write on the ES with the following steps:

1. Create block icons
2. Initialize properties of the block icons.
3. Set up authorizations for operators.
4. Transfer the configuration to the OS.

The individual steps are described below.

Creating block icons

1. Open the *PCS 7* project in *SIMATIC Manager*.
2. Create a new picture object in the level of the plant hierarchy containing the CFC charts with the F_CHG_R and F_CHG_BO F-Blocks.
3. Select the picture object and open the object properties.
4. In the "Block Icons" tab, select the "Derive block icons from the plant hierarchy" option.
5. Click "OK" or "Apply" to confirm the revised properties.
6. Select the OS object and select "Compile" on the shortcut menu to compile the OS.
7. If necessary, select the "Generate/update block icons" option in the "Compile OS" wizard when selecting the data to be compiled and the scope of the compilation. Click the "Compile" button in the last dialog box.

Result: When the OS is compiled, the block icons are automatically inserted into the new picture.

Note

In order to prevent SAFE_ID1 and SAFE_ID2 from being overwritten, clear the "Derive block icons from the plant hierarchy" option in the object properties of the WinCC picture before recompiling the OS.

Initializing properties of the block icons

1. Double-click the picture file in the Plant view of the *PCS 7* project.

   Result: WinCC Explorer is started and the picture file is displayed in the Graphics Designer. The name is displayed in the header of each block icon. The name of the block icon is formed from the name of the CFC chart and the name of the associated F-Block instance.

2. Select a block icon and open the object properties.
3. In the "Properties" tab, select "User configuration".
4. Assign the exact static values to the SAFE_ID1 and SAFE_ID2 attributes that were configured at the SAFE_ID1 and SAFE_ID2 inputs of the associated F-Block instance.

   ⚠ WARNING

   Static values of the SAFE_ID1 and SAFE_ID2 attributes

   The static values of the SAFE_ID1 and SAFE_ID2 attributes must be identical to the F-Parameters that are configured at the SAFE_ID1 and SAFE_ID2 inputs of the associated F-Block instance.

   Note that these values at the F-Blocks in the *CFC Editor* and at the block icons in *WinCC* are independent and must be entered separately.

5. Assign the required authorizations to the "InitiatorAuthorizaton" and "ConfirmerAuthorization" attributes. Alternatively, accept the default operator authorizations. See also "Setting up authorizations for operators".

   Default authorizations (correspond to the user hierarchies from *PCS 7*):

   – For the operator who initiates a change to an F-Parameter using Safety Data Write (the initiator): No. 5, Operator-process communications

   – For the operator who confirms a change to an F-Parameter using Safety Data Write (the confirmer): No. 6, Higher-order operator-process communications

6. Repeat Steps 2 and 5 for all available block icons.
7. Save the picture file.

Examples

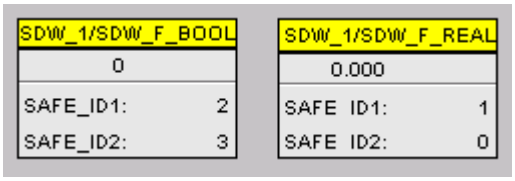


Figure 9-1 Example: Picture File with Block Icons

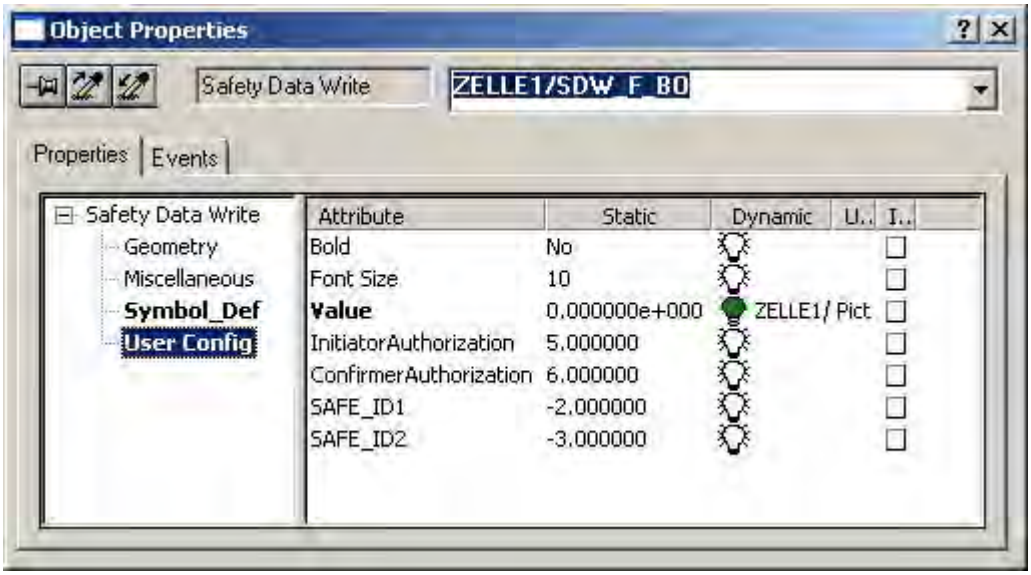


Figure 9-2 Example: Properties of a Block Icon

Setting Up User Authorizations for Operators

Create the following users based on whether the transaction is to be performed by two operators or by one operator only:

- If the transaction for an F-Parameter is to be performed by two operators, create two users:
  - The initiator initiates a change to an F-Parameter using Safety Data Write. This user must have the authorization assigned to the "InitiatorAuthorization" attribute in the properties for the block icon. However, the initiator does not have authorization to confirm the change.
  - The confirmer verifies and confirms the change. This user must have the authorization assigned to the "ConfirmerAuthorization" attribute in the properties for the block icon. However, the confirmer does not have authorization to initiate the change.
- If only one operator is to perform all of the transaction steps, create a user who has both authorizations assigned to the "InitiatorAuthorization" and "ConfirmerAuthorization" attributes in the properties for the block icon.

Create the users in WinCC Explorer using the "User Administrator" editor.

Activating the OS

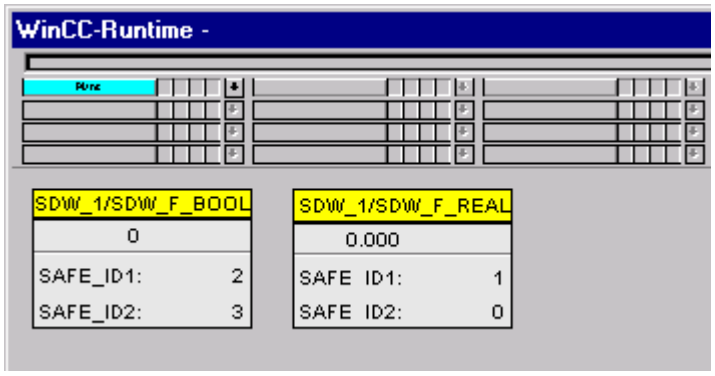
Activate the WinCC Runtime system of the OS, e.g., by selecting **File > Activate** in WinCC Explorer.

Result

Once the WinCC Runtime system is activated and login is complete, the hierarchy levels appear as buttons in the runtime system of the OS. Click the button to display the block icons for this level.

Example

The following figure shows two block icons in the runtime system of the OS.



Clicking a block icon opens the faceplate that you can use to change an F-Parameter by means of Safety Data Write.

Detailed information

For detailed information on the described steps, refer to:

- Configuration Manual "PCS 7 Operator Station (http://support.automation.siemens.com/WW/view/en/27002758)"
- Online Help for the WinCC editors (e.g., Graphics Designer and User Administrator)

9.3 Changing F-Parameters with Safety Data Write

9.3.1 Requirements and General Instructions

You perform a transaction for changing an F-Parameter using Safety Data Write by means of a faceplate in the OS. The transaction consists of a sequence of operations that can be performed by one or two operators.

Requirements

- The S7 program is compiled and downloaded to the F-CPU.
- The user(s) with the relevant authorizations are set up.
- The configuration of the faceplates is downloaded to the OS.
- The AS/OS connection is okay. The operator can test the AS/OS connection using the "OS Test" button (see the section entitled "Testing AS/OS connection" below).
- The EN_CHG input of the F-Block instance of F_CHG_R or F_CHG_BO for enabling Safety Data Write is set to TRUE.
- When using OS clients, make sure that no default server is set for tags (in WinCC Explorer select "Server Data," in the shortcut menu select "Default Server" and in the "Configure Default Server" dialog for the "Tags" component select "No Default Server").

Specifications for Changing an F-Parameter using Safety Data Write

The operator(s) need the following information to change an F-Parameter using Safety Data Write:

- Name of the block icon
- New value for the F-Parameter

General Information

The transaction must be completed within a specified time interval (Timeout). If the transaction is not finished before the Timeout interval elapses, the transaction is automatically canceled once the Timeout interval expires.

⚠WARNING

Initiator and confirmer must not accept an invalid value

As the initiator or confirmer, you must not accept an invalid value. If there are inconsistencies, you must cancel the transaction.

As an operator, you must not rely on individual display fields of the faceplate; rather, you must check the values and compare them among each other.

Before starting the transaction, you must verify the plant name in the header of the faceplate.

⚠WARNING

Technological assignment must be appropriate for the environment

When opening the faceplate, make sure that the technological assignment in the top line is appropriate for the environment in which the faceplate was placed.

⚠WARNING

Transaction for changing an F-Parameter

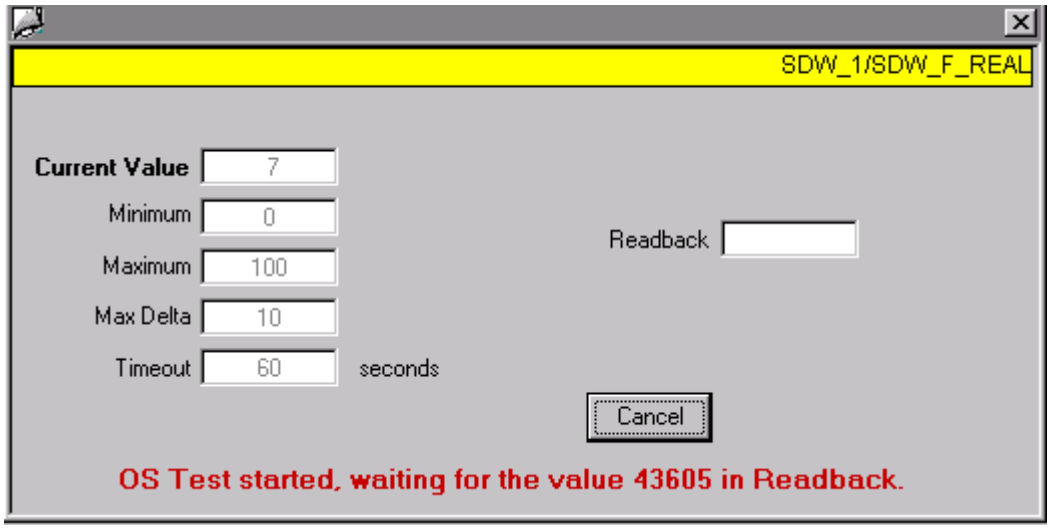
You can only perform one transaction for changing an F-Parameter at a time. You must use organizational measures to ensure that multiple transactions are not performed simultaneously for the same F-Parameter. Otherwise, the transaction cannot be performed correctly, resulting in unexpected results, such as:

- Display of incorrect values in the faceplate fields

  or
- Unexpected cancellation of the transaction

Testing AS/OS Connection

Before starting the transaction, you can test the AS/OS connection by clicking the "OS Test" button.



If the AS/OS connection is okay, a message to that effect is output and the expected value is displayed in the "Read Back" field.

If the AS/OS connection is not okay, the following error message is displayed: "OS test failed".

If the block is assigned

If a transaction for a faceplate has been started already, the following message appears when opening the faceplate in WinCC Runtime:

"Block is assigned. Please wait..."

To start a new transaction, click "Cancel" and reopen the faceplate.

9.3.2 Changing an F-Parameter with Two Operators

Operator Authorizations

The transaction requires two operators having different authorizations.

- The initiator initiates a change to an F-Parameter using Safety Data Write. This user must have the authorization for initiating the change but not for confirming it. The authorization corresponds to the "InitiatorAuthorization" attribute in the properties for the block icon. Default is No. 5, Operator-process communications.
- The confirmer enters the modified value again, verifies it, and confirms the change. This user must have the necessary authorization for confirming the change but not for initiating it. The authorization corresponds to the "ConfirmerAuthorization" attribute in the properties for the block icon. The default setting is No. 6, Higher-level operator-process communications.

Note

The sections below describe the necessary transaction steps for the two operators. The figures illustrate the example of an F_REAL parameter with the login:

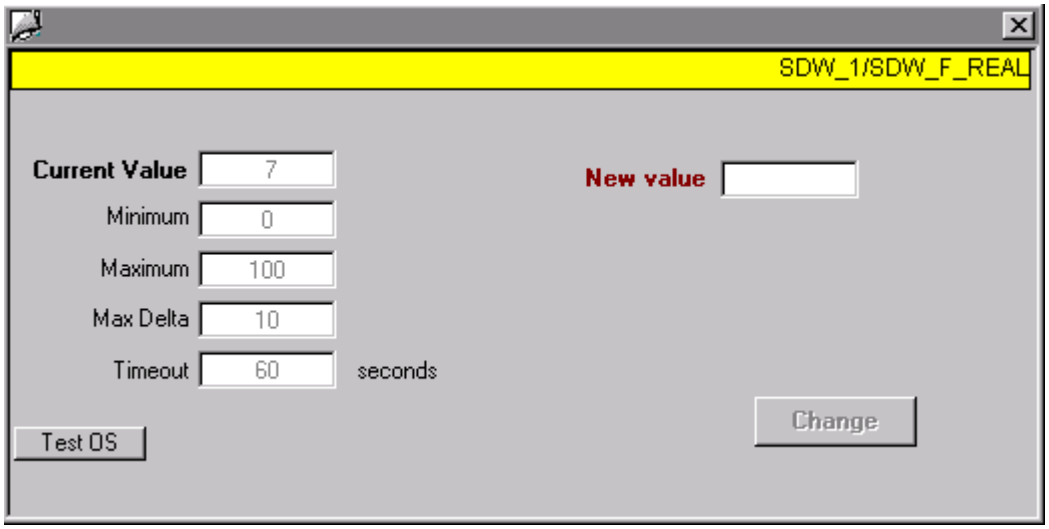
- level5 – Initiator
- level6 – Confirmer

Note

When changing F_BOOL parameters using Safety Data Write, you must enter the value "true" or "false" and not "1" or "0". This entry is not case-sensitive.

Initiator: Initiating a change

1. Log on to the OS as a user with initiator authorization.
2. Click the desired block icon to open the faceplate.



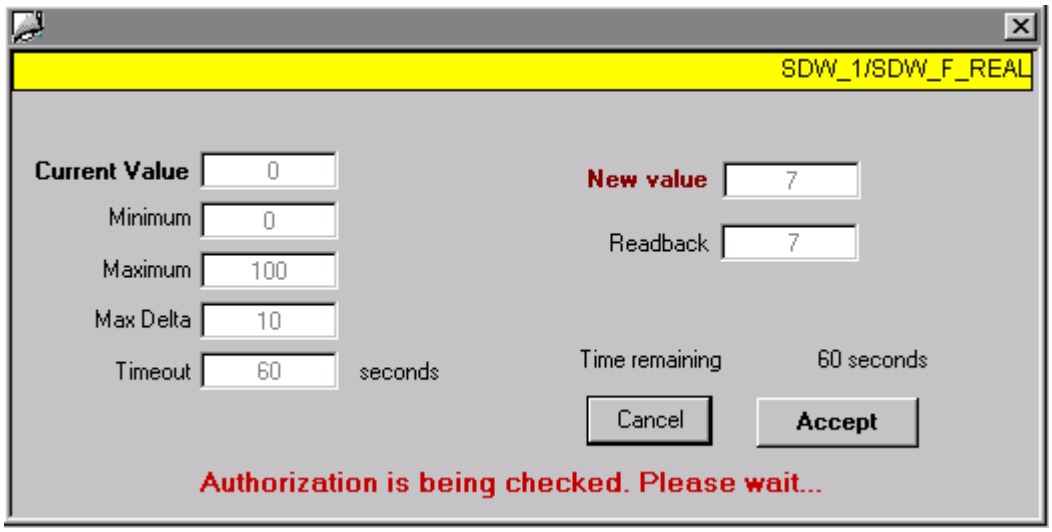
The Safety Data Write dialog indicates the current value, the Timeout value in seconds, and, in the case of F_CHG_R, the values for the change limits (Minimum, Maximum, and MaxDelta) as well as the unit of measurement, where applicable.

3. Enter the new value in the "New value" field (using a maximum of 10 characters including decimal separators and plus or minus signs).

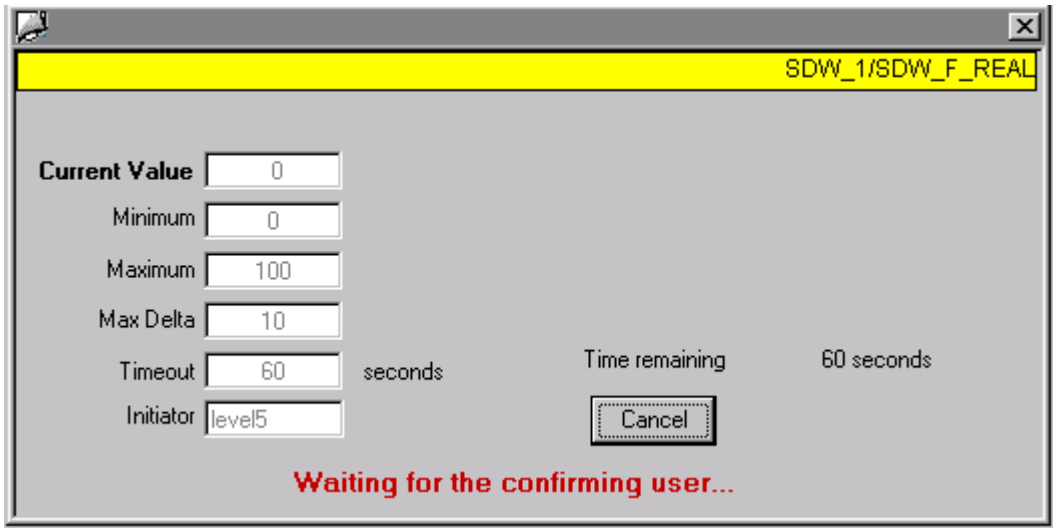
   In the case of an F_REAL value, verify that the change limits (Minimum, Maximum, and MaxDelta) are not violated. If the new value violates one of the limit values, an error message is displayed and the "Change" button cannot be activated.

4. Click "Change". The modified value is also displayed in the "Readback" field.
5. Compare the values in the "New value" and "Readback" fields. If they are identical, click the "Accept" button.

   Note: If the block input EN_CHG changes to FALSE before you click the "Accept" button, this is indicated by a message, and the "Accept" button is disabled (see also the description of the F-Blocks "F_CHG_R: Safety Data Write for F_REAL (Page 250)" and "F_CHG_BO: Safety Data Write for F_BOOL (Page 256)").



Result: The timeout monitoring is started, and you are informed that the change must be confirmed by a second operator.



The confirmer must then continue the transaction.

If you cancel the transaction after clicking "Accept," check whether the previously valid value is displayed in the "Current value" field.

Confirmer: Confirming the change

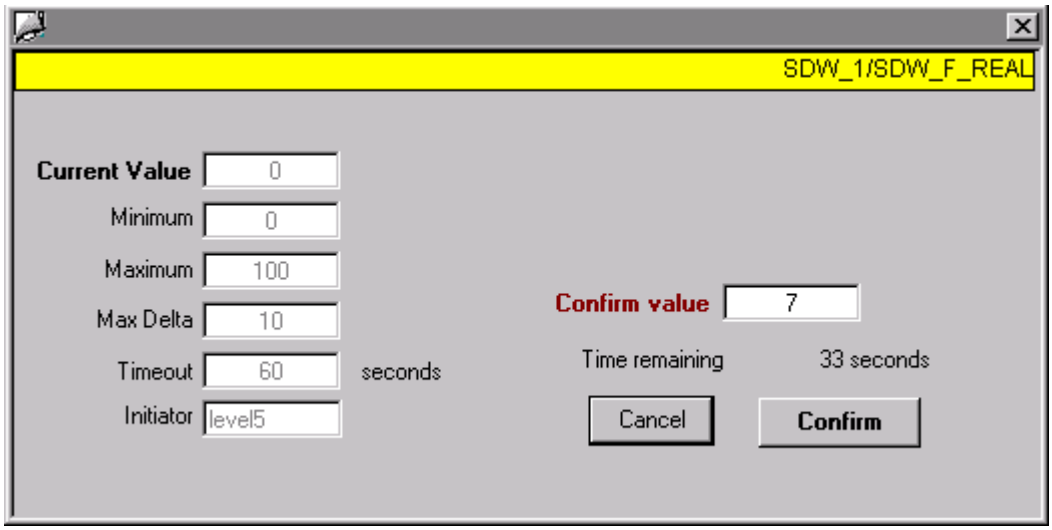
Note

The confirmation must take place before the remaining time expires.

1. Log on to the OS as a user with "confirmer authorization".

   You can log on to a second OS or on the same OS as the initiator.

2. Click the desired block icon to open the faceplate.



3. Enter the new value in the "Confirm value" field. If the confirm value differs from the new value that was entered by the initiator, an error message is displayed and the "Confirm" button cannot be activated.

   Note

   You must confirm the change by entering the new value separately. The value is deliberately not displayed since an unbiased confirmation by the second operator is required.

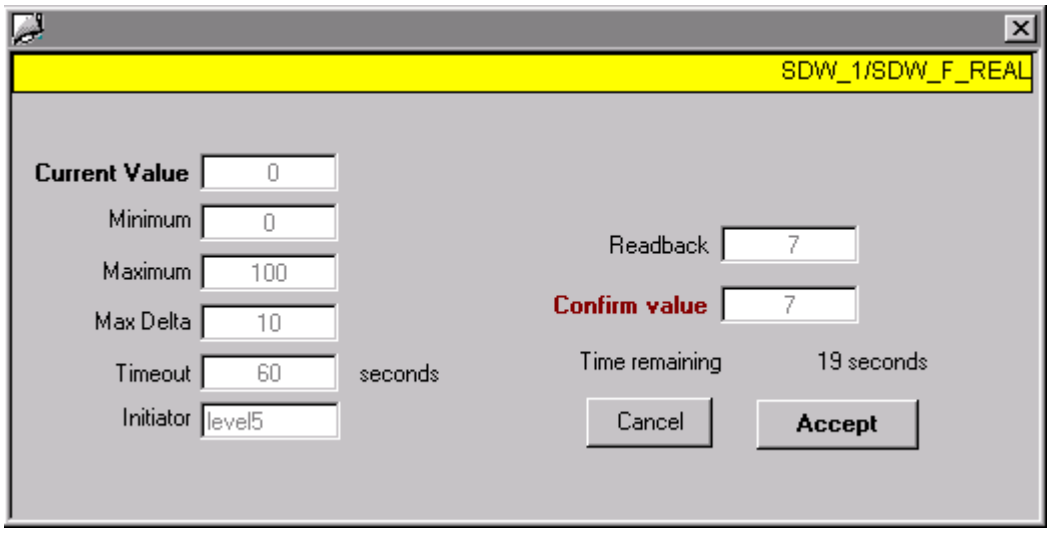
4. Click "Confirm".

   The value entered by the initiator is displayed in the "Readback" field.

   Note: If the block input EN_CHG is changed to FALSE, this is indicated by a message, and the input is canceled. Values can be reentered once EN_CHG changes back to TRUE (see the description of the F-Blocks "F_CHG_R: Safety Data Write for F_REAL (Page 250)" and "F_CHG_BO: Safety Data Write for F_BOOL (Page 256)").

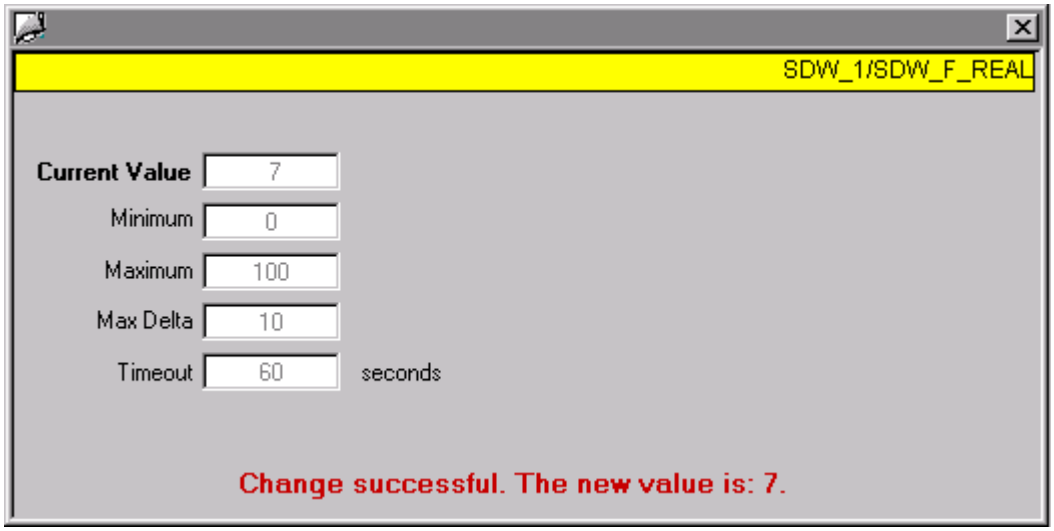
5. Compare the values in the "Confirm value" and "Readback" fields. If they are identical, click the "Accept" button to permanently save the change. If the values do not match, you must click "Cancel".

   Note: If the block input EN_CHG changes to FALSE before you click the "Accept" button, this is indicated by a message, and the "Accept" button is disabled (see also the description of the F-Blocks "F_CHG_R: Safety Data Write for F_REAL (Page 250)" and "F_CHG_BO: Safety Data Write for F_BOOL (Page 256)").



Result

If the transaction is finished within the remaining time, a successful F-Parameter change is signaled.



9.3.3 Changing an F-Parameter with One Operator

Operator Authorization

If only one operator is to perform the transaction, this operator must be authorized to both initiate and confirm changes using Safety Data Write. The authorization must include the values of both the "InitiatorAuthorization" and "ConfirmerAuthorization" attributes. Default is No. 5, Operator-process communications and No. 6, Higher-level operator-process communications.

Transaction Sequence with Only One Operator

The procedure is the same as for operation with two operators, except that one operator is able to perform all of the steps (see also the section entitled "Changing an F-Parameter with Two Operators (Page 143)").

The difference is there is no waiting period for the confirmer. Rather, the operator is prompted immediately to enter the confirm value.

All other steps remain the same.

Compiling and commissioning an S7 program 10

10.1 Compiling an S7 program

Introduction

You compile a safety program in the usual way in the *CFC Editor*, that is, by compiling the entire S7 program.

Procedure

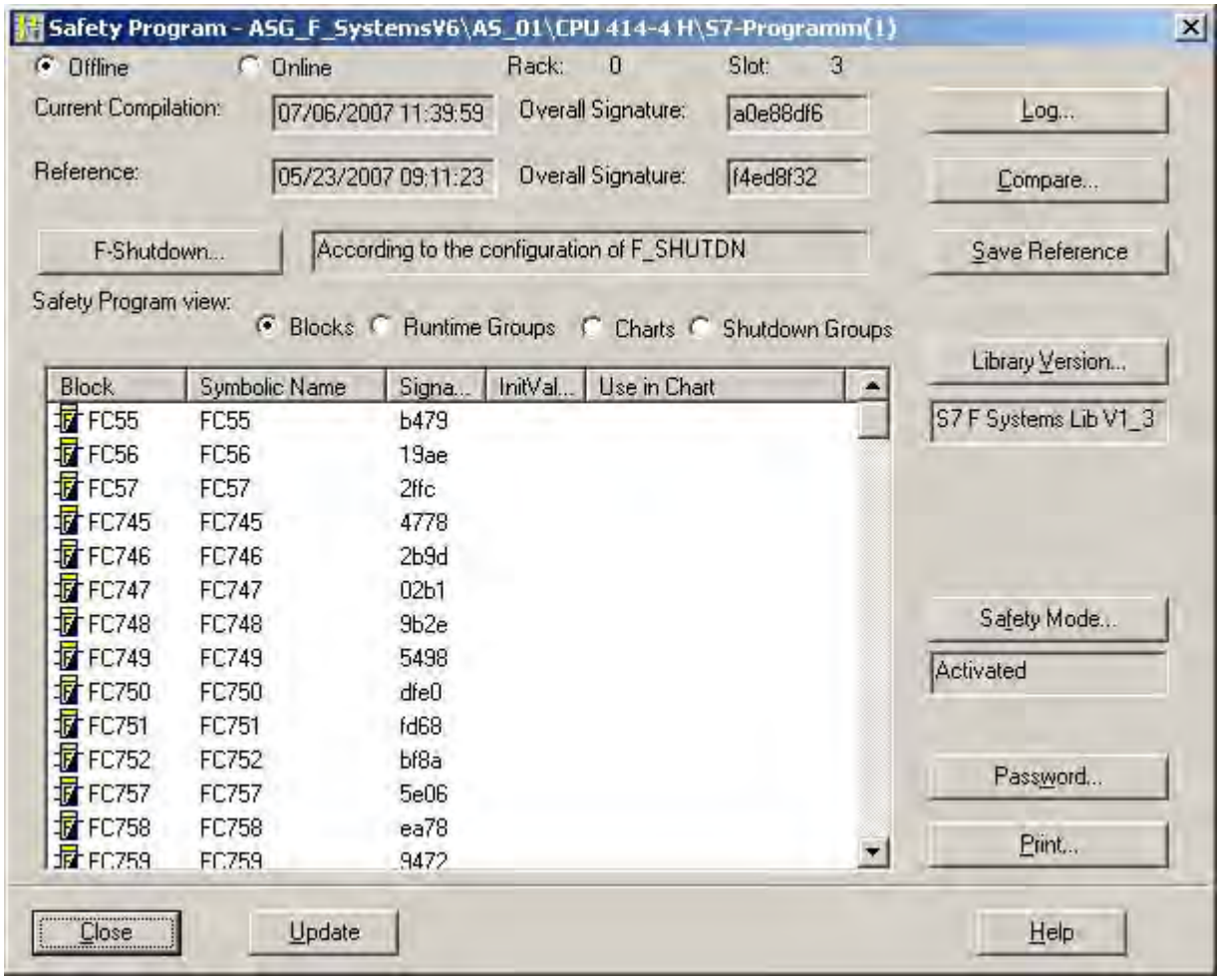
If an S7 program contains a safety program, then the safety program is automatically compiled when the CFC charts are compiled. Fault control measures are automatically added, and additional safety-related checks are performed.

Read and comply with the documentation for *CFC*: " CFC for S7 Continuous Function Chart (http://support.automation.siemens.com/WW/view/en/21401430)".

If you have changed the safety program since the last compilation, you will be prompted for the password to your safety program during compilation. You must enter the password to your safety program to continue the compilation process.

10.2 "Safety Program" dialog

In *SIMATIC Manager*, open the "Safety Program" dialog box by selecting "**Options > Edit Safety Program**".



The following information about the safety program located online on the F-CPU or offline in the ES will be displayed in the "Safety Program" dialog box:

- A list of all included F-Blocks with signatures and initial value signatures
- Current compilation: Date and collective signature

Note

After a CFC online change has been made, the collective signature is updated, but not the time stamp. For more information, refer to the section entitled "Testing a safety program (Page 166)".

- Reference: Date and collective signature
- If the *Failsafe Blocks* (V1_1) library version is displayed, you can use the "32-bit signature" check box to display the collective signature as a 16-bit signature or a 32-bit signature. If the "32-bit signature" check box is selected, the collective signature is displayed as a 32-bit signature.

Buttons in the "Safety Program" dialog box

The dialogs you can access and the actions you can perform in the "Safety Program" dialog box are described in the sections below.

10.2.1 "Shutdown Behavior" dialog box

Description

In the "Shutdown Behavior" dialog box, you can choose how the safety program should behave when an error is detected, i.e., during an F-STOP:

- "Full shutdown": All F-Shutdown groups of a safety program are shut down the first time an error is detected in an F-Shutdown group.
- "According to the configuration of F_SHUTDN":
  - The faulty F-Shutdown group or groups of a safety program are shut down the first time an error is detected in an F-Shutdown group (partial shutdown).

    or
  - All F-Shutdown groups of a safety program are shut down the first time an error is detected in an F-Shutdown group.

You must recompile the S7 program after changing the shutdown behavior.

You must also enter the password for the safety program when you change the shutdown behavior.

See also

F-STOP (Page 84)

10.2.2 "Logs..." button

Click the "Logs..." button to open the "Logs" dialog of the *CFC Editor*. The "Compile" and "Download" logs are relevant for the safety program acceptance test. For information about the acceptance test, refer to the section entitled " System Acceptance Test (Page 179) ".

10.2.3 "Save Reference" button

You can save all data of a safety program (charts, parameters, etc.) as a reference to be used as necessary for comparisons.

10.2.4 "Library Version" button

Description

The "Library Version..." button enables you to upgrade the F-Library version used in the project to the current version of the F-Library.

The window below the button displays the F-Library version *currently used in the project*.

See also

Migrating to S7 F Systems V6.1 (Page 30)

10.2.5 "Password for Safety Program Creation" dialog

Description

You must create a password for each safety program. You must enter this password by means of the "Password..." button in the "Safety Program" dialog box before you can perform the operations presented in the section entitled "Overview of access protection (Page 63)":

The user obtains access permission by entering the password for the safety program when performing one of these operations. Access permission is valid for one hour. After this, the user is again prompted for and must enter the safety program password the next time he wants to perform one of the operations above.

For each safety-related action, the access permission is reset to one hour.

Access permission can also be revoked in the "Create Password for Safety Program" dialog box.

10.2.6 "Update" button

Description

Use this button to refresh all displayed information. This may be necessary if any changes have been made in other applications, such as the *CFC Editor*, since the dialog box was opened.

Pressing this button will also display information from the "Use in Chart" field. For performance reasons, this area is still empty when the dialog box is opened.

10.3 Comparing safety programs

Introduction

The "Compare Programs" dialog box enables you to compare safety programs and display and print out differences.

You can compare the following safety programs:

- Online safety program in the F-CPU
- Current offline safety program
- Last compilation of the current S7 program
- Saved reference program

The result of the comparison shows you whether the following are the same or different:

- Collective signature
- Individual signatures
- Parameter values
- Differences in the safety program and control structures
- Modified or deleted F-Blocks and interconnections, etc.

With the "Compare Programs" dialog, you can also tell if a safety program was *not* modified. by comparing the safety program to the reference program.

In *S7 F/FH Systems* V6.1 and later, system-related changes are shown in a combined display, making it easy for you to identify changes that are relevant for checks. This facilitates the acceptance test for changes.

System-related changes are primarily found:

- In system charts beginning with @F_x
- In runtime groups beginning with @F_x
- On driver blocks

Program/reference

Select one of these option boxes to specify whether you want to compare the current program or the reference program.

Compare with:

Use this drop-down list box to specify the second safety program to which you want to compare the safety program you just selected.

Program	Compare with ...	
	Reference	Last saved reference for this safety program
	Last compilation	The last compilation of this S7 program during which safety-related changes were detected.
	Online	Currently downloaded safety program in the F-CPU
	Other project	Any offline program. Use the "Browse" button to select the offline program.
Reference	**Compare with ...**	
	Current safety program	Current offline program
	Last compilation	The last compilation of this S7 program during which safety-related changes were detected.
	Online	Currently downloaded safety program in the F-CPU
	Other project	Any offline program. Use the "Browse" button to select the offline program.

"Browse" button

Use this button and the "Open" dialog to select the offline program of any project to be compared.

"Start" button

Click this button to start the comparison.

View options

If you want to compare two *offline* programs, you can switch back and forth between the following options by clicking the relevant option buttons:

- **Block view**:

  Shows you a list with the differing blocks (different block signatures).

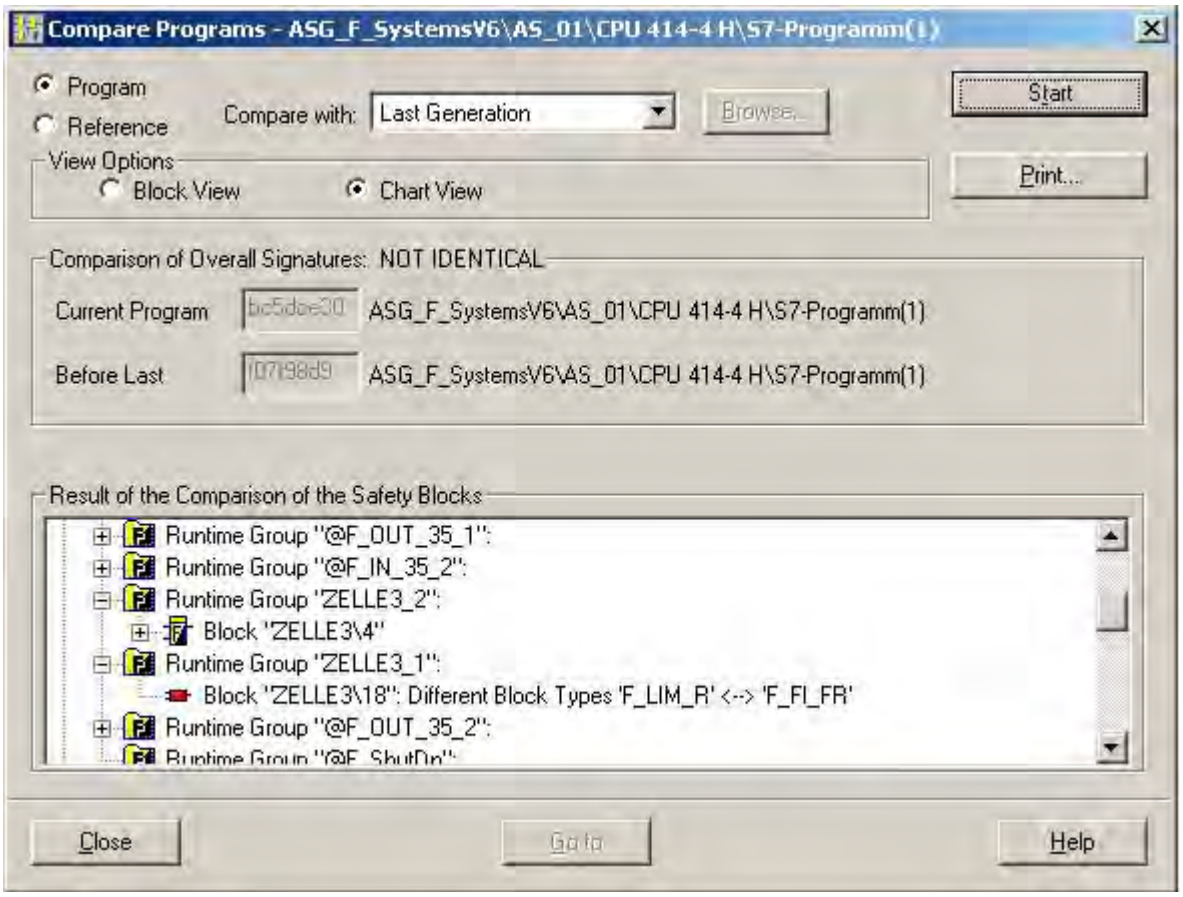
- **Chart view**:

  Shows you a hierarchy of all differences in the:

  – Task
  – F-Runtime group
  – F-Block
  – Parameters

  In this view, the "Go to" button is available.

Result of the comparison (both safety programs offline)



A note is displayed indicating whether or not the collective signatures of all F-Blocks are identical.

Display of differences in the block view

In the block view, all F-Blocks whose signatures have changed are displayed with the relevant signature, but the F-Runtime group and task are not displayed.

Display of differences in the chart view

The differences between charts are displayed in a hierarchical format similar to Explorer. In this view, all F-Blocks are shown under the relevant task and F-Runtime group. Information about the possible changes are shown individually for each F-Block. This information relates to the task, the F-Runtime group, and the sequence within the F-Runtime group, as well as the parameter assignment and interconnections of the F-Blocks.

Only tasks, F-Runtime groups, F-Blocks, and parameters in which changes were found are displayed.

Changes are described as follows:

Text	Meaning
Deleted	F-Block only present in source
Added	F-Block only present in comparison program
Runtime position changed	F-Block is located in a different runtime position in the F-Runtime group
Interface changed	• Additional parameters • Removed parameters • Modified data type (e.g., F-Bool <- Bool)
Signature changed	Signature of F-Block type (FB) changed
Value: '"new" <- "old"	The parameter assignment of an input or output or the interconnection source of an input has been changed from "old" to "new". "Not-interconnected" can also be specified as the interconnection source if an interconnection has been deleted or newly created.

Note

If "Different versions of F-Reference data" appears in the chart view when comparing the safety program to a reference, this means that you created the reference with an older version of *S7 F Systems* and did not overwrite it with the current version during migration. See " Migrating to S7 F Systems V6.1 (Page 30) ".

Instead, use the old project version that you archived prior to migration.

Displayed changes

Note the following when changing names:

The *F Systems* comparator references the elements according to name. If an element name is changed, the element can no longer be assigned.

- Chart names
- Name of a runtime group
- Block name (instance in a chart)
- Parameter name (for F-Block types)

Although chart names are not relevant for runtime, changes still affect the "Chart view":

- Each time a chart name is changed, the chart is displayed with the old name as "Deleted" and with the new name as "Added".
- In *CFC*, an F-Runtime group with the same name is renamed at the same time. Therefore, this F-Runtime group is also displayed with the old name as "Deleted" and with the new name as "Added".
- All interconnects of F-Blocks outside of this chart to F-Blocks within this chart as displayed as changed. The reason for this is that the chart name is also used as the name component of an interconnection peer to identify the interconnection.
- The block view correctly returns no difference in this case. Likewise, the collective signature of the safety program does not change. In order to prevent such unnecessary entries in the chart view, we recommend that you do not rename any F-Charts or shift between F-Charts after performing the acceptance test.

Note the following:

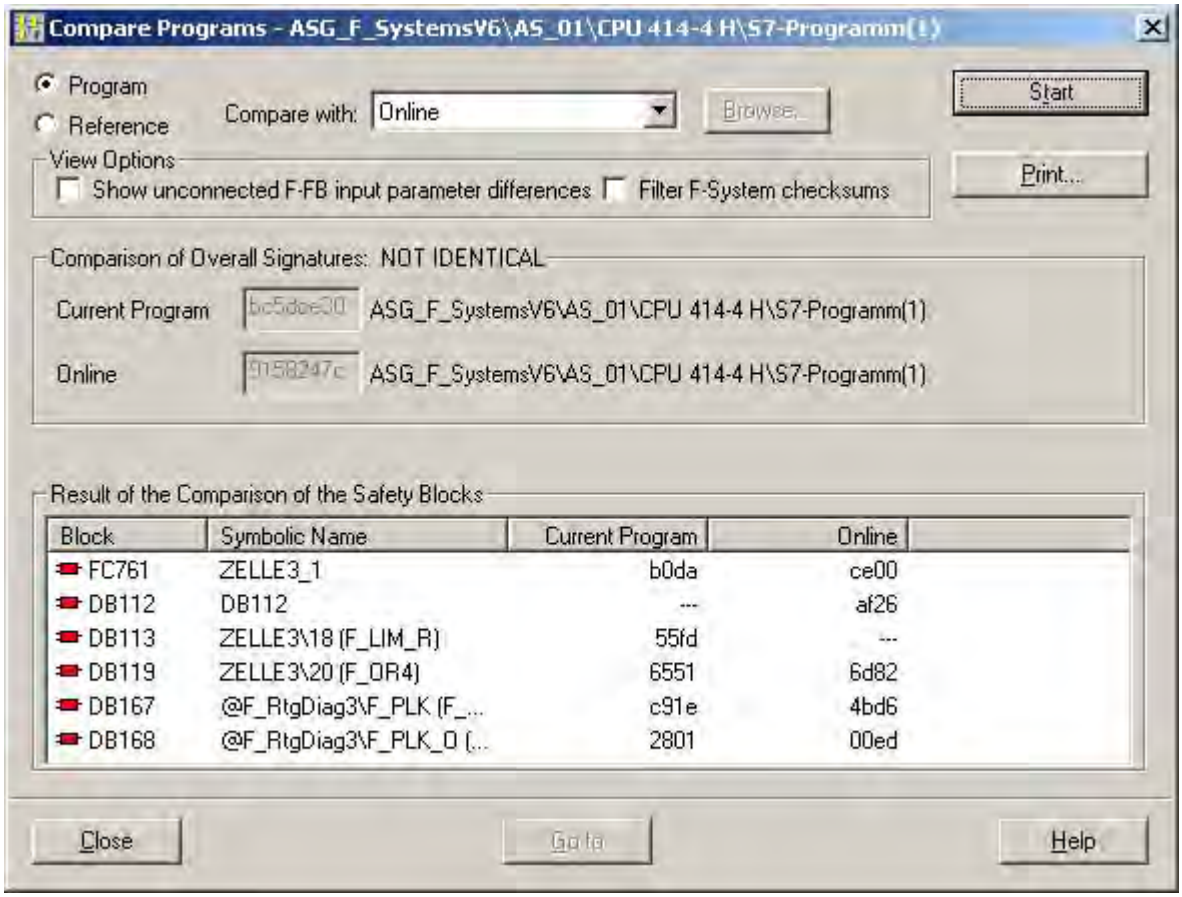
- In the chart view of the comparison, only differences pertaining to the safety program are generally displayed. In particular, changes in interconnections between the safety program and the standard program or global addresses are not displayed.
- If an interconnection of an output is changed at the same time as the initial value of this output, the modified interconnection will be displayed, but not the modified initial value.

Result of the comparison (online safety programs with offline)

When a comparison to the online program is made, an indication is given as to whether the source, load memory, and work memory match (this allows you to detect non-permissible data manipulations on non-interconnected, fail-safe input parameters in the work memory). See also Section "Checking the collective signature" in the section entitled "Commissioning a safety program (Page 179)".

If you have selected the online program in the "Compare with" drop-down list box, only the block view is available. In this case, the following two view options are available:

- Display differences in non-interconnected F-FB input parameters
- Filter F-System signatures



Just as in the offline block view, the window shows you all F-Blocks whose signatures differ.

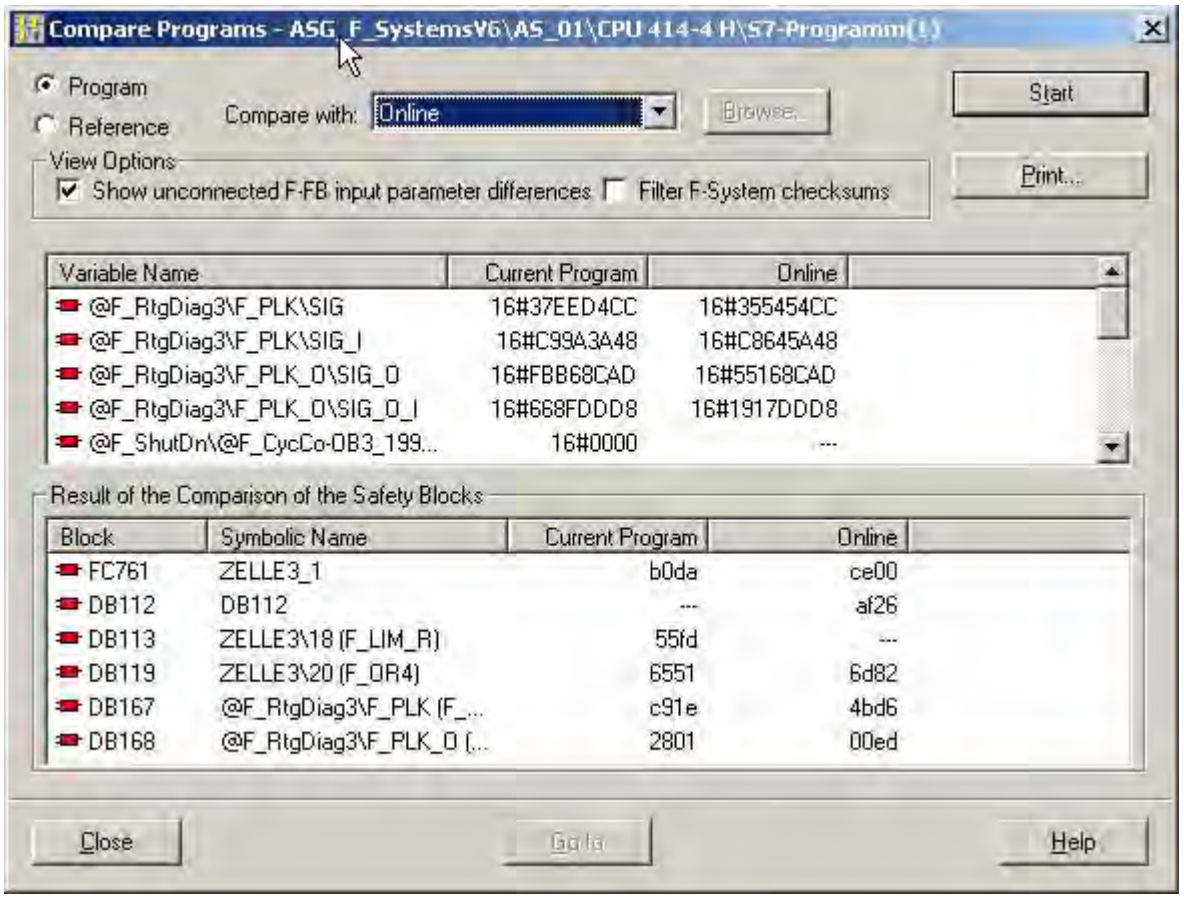
"Show unconnected F-FB input parameter differences" view option

This option compares the assigned parameter values of all non-interconnected inputs. It compares the online program to the offline program.

The differences are displayed in the list at the top of the dialog box.

This view option is normally selected only if the collective signatures already match. This indicates that the offline program has not been changed since the last time it was downloaded to the F-CPU.

This option enables you to perform a thorough search for parameters that have been changed online, but not through compilation or download.



"Filter F-System checksums" view option:

This option suppresses expected differences that can occur when the F-CPU writes to specific F-Blocks (for example, input signature values of F_PLK and F_PLK_O). You can only use this view option in connection with the "Show unconnected F-FB input parameter differences..." option.

"Print" button

Click this button to print out the result of the comparison.

"Go to" button

In the chart view, you can select any F-Block or parameter in the differences display and then click this button to access the relevant block in the *CFC Editor*.

10.4 Printing project data of the safety program

Procedure:

You receive a printout of all important project data as follows

1. Select the program folder (e.g., "S7 Program").
2. Select the menu command **Options > Edit Safety Program**.
   The "Safety Program" dialog will appear.
3. Click "Print". In the "Print" dialog, you can select the parts of the project you want to print:

- **Chart (standard and safety chart):**

  Prints all charts of a safety program in a graphical representation.
- **Safety program: Block list and signatures**
  Offline/online status log
  Name of the safety program
  Date of the last compile operation and the collective signature of the safety program
  Date of the last compile operation and collective signature of the reference program
  F-Blocks in the safety program
  Safety-related parameters if the corresponding option is selected
  The footer on each page of the printout shows you the version of *S7 F Systems* used to generate the printout along with the collective signature.
- **HW configuration:**

  Printout of the complete hardware configuration or portions thereof. The "Print" dialog will appear so that you can specify what information is to be printed for the F-I/O.

The printout of the safety program also contains the collective signature and the date of the last compilation, which are relevant to the onsite acceptance test of the safety program (e.g., by experts). The collective signature of the compiled S7 program appears twice in the printout:

1. In the program information section as a value of the block container
2. In the footer as a value from the chart container

(See also the section "Checking the collective signature" in chapter " Downloading the S7 program to the F-CPU (Page 183) ").

10.5 Safety mode

Introduction

Safety mode of the safety program in the F-CPU can be deactivated and reactivated at times. This allows you to make changes in the safety program in RUN mode.

Description

All the safety mechanisms for fault detection and fault reaction are activated in safety mode. The safety program cannot be modified during operation (in RUN mode) in safety mode.

You can activate or deactivate safety mode in the F-CPU in RUN mode using the "Safety Mode..." button in the "Safety Program" dialog. Downloading safety program changes in RUN mode is only made possible by temporarily switching the safety mode to "deactivated" using this button.

The window below this button indicates whether safety mode is "activated" or "deactivated". It will indicate "Unknown" if the safety program does not correspond to the safety program in the F-CPU or if no communication is taking place with the F-CPU.

You can also determine whether or not safety mode is enabled from the SAFE_M output of the F_SHUTDN block (located in the @F_ShutDn chart).

See also

Downloading the safety program (Page 164)

10.5.1 Deactivating safety mode

Introduction

The safety program will continue to run in deactivated safety mode. Safety mechanisms for fault detection and fault reaction are deactivated in deactivated safety mode.

⚠ WARNING

Deactivating safety mode

Because changes to the safety program can be made in RUN mode when safety mode is deactivated, you must take the following into account:

- Deactivation of safety mode is intended for test purposes, commissioning, etc. 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.
- It must be possible to verify that safety mode has been deactivated. A log function is required and can be ensured, for example, by using an OS. The automatically placed F_SHUTDN block generates corresponding messages for this purpose. Otherwise, you must use organizational measures to log the deactivation of safety mode.
- We also recommend that deactivation of safety mode be displayed on the OS, for example. The automatically placed F_SHUTDN F-Block sets the SAFE_M output to "0" on deactivation of safety mode (or the F_TESTM F-Block sets the TEST output to "1").
- Safety mode is deactivated across the F-CPU only. You must take the following into account for safety-related CPU-CPU communication:

  If the F-CPU with the F_SENDBO, F_SENDR, or F_SDS_BO is in deactivated safety mode, you can no longer assume that the data sent by this F-CPU are generated safely. You must then implement organizational measures such as operation monitoring and manual safety shutdown to ensure safety in those portions of the system that are affected by the sent data. Alternatively, you must output fail-safe values instead of the received data in the F-CPU with F_RCVBO, F_RCVR or F_RDS_BO by evaluating SENDMODE.

Requirements

The F-CPU is in RUN mode (the mode selector is set to the RUN or RUN-P position) and safety mode is activated.

Procedure

1. In *SIMATIC Manager*, select the F-CPU or its S7 program.
2. Select the menu command **Options > Edit Safety Program**.
3. Select the "Safety Mode" button.

You can now download safety program changes to the F-CPU during operation (in RUN mode).

See also

Testing with S7-PLCSIM (Page 167)

10.5.2 Activating safety mode

Introduction

After downloading safety program changes, you must activate safety mode again in order to ensure secure execution of the safety program.

Requirements

The F-CPU is in RUN mode (the mode selector is set to the RUN or RUN-P position) and safety mode is deactivated.

Procedure

1. In *SIMATIC Manager*, select the F-CPU or its S7 program.
2. Select the menu command **Options > Edit Safety Program**.
3. Select the "Safety Mode" button.

Note

If the safety program detects a safety-related error while in deactivated safety mode, you cannot activate safety mode. You receive a message to that effect along with remedial measures.

See also

Downloading changes (Page 169)

10.6 Downloading the safety program

Introduction

After compiling, you can download the CFC program to the target system. You can download the entire safety program or just the safety program changes, depending on whether safety mode is activated or deactivated, as follows:

Downloading	F-CPU in STOP mode	F-CPU in RUN mode, safety mode activated	F-CPU in RUN mode, safety mode deactivated
the entire S7 program	Possible	F-CPU is automatically placed in STOP mode by the *CFC Editor*	F-CPU is automatically placed in STOP mode by the *CFC Editor*
Changes in the standard user program	Possible	Possible	Possible
Changes in the complete S7 program	Possible	Not possible	Possible

Requirements

- The hardware configuration data of the station are downloaded to the F-CPU.
- The S7 program has been compiled without errors.
- You have access permission to the target system.
- An online connection exists between the F-CPU and your ES.

Rules for downloading

- You can only download the safety program from the *CFC Editor* or from *SIMATIC Manager* by means of the chart folder.
- When downloading an approved safety program, you must check the collective signature after downloading same as for the acceptance test.

  See also the section "Checking the collective signature" in the section entitled " Downloading the S7 program to the F-CPU (Page 183) ".

⚠ WARNING

Do not copy F-Blocks with *SIMATIC Manager*

As is customary in *PCS 7*, you must not copy individual blocks between the block containers online and offline. For this purpose, you must download in the *CFC Editor* or download the chart folder.

For detailed information, refer to Manual " CFC for S7 Continuous Function Chart (http://support.automation.siemens.com/WW/view/en/21401430) ", Section 3, "Downloading the User Program to the Target System" and "Reading Back Charts".

10.6.1 Downloading the S7 program

Procedure

To download the safety program to the target system, call the **CPU > Download > Scope: Entire program** menu command in the *CFC Editor*. The F-CPU goes to STOP mode.

Note

You are prompted to enter the password for the F-CPU prior to downloading the safety program if changes in the safety program are detected.

Working with safety programs on a memory card

⚠WARNING

Safety program on a memory card

If you are using a safety program on a memory card, you must observe the following:

- Before switching the S7 F-System to RUN mode, compare the collective signature of the safety program on the Flash EPROM memory card to the collective signature of the reference data. If necessary, label the memory card with the collective signature.
- In the case of a fault-tolerant S7 FH System, make sure that the memory cards of the redundant F-CPUs are of the same type (RAM or Flash EPROM) and that the same safety program is located on redundant Flash EPROM memory cards.
- Ensure access protection against removal and insertion of memory cards.

⚠WARNING

If multiple F-CPUs can be accessed over a network (such as MPI) from one ES, you must take the following actions to ensure that the safety program is downloaded to the correct F-CPU:

Use passwords specific to each F-CPU, e.g., a uniform password for the F-CPUs having the respective MPI address as an extension (max. 8 characters): PW_8.

Note the following:

- Before downloading a safety program to an F-CPU for which access permission by means of an F-CPU password does not yet exist, you must first revoke existing access permission for any other F-CPU.

10.7 Testing a safety program

Introduction

Testing occurs as usual in *CFC* by switching to test mode.

Switching to test mode

After compiling and downloading, you have the option of testing the safety program. You test safety programs by switching to test mode using the **Debug > Test Mode** menu command in the *CFC Editor*. In test mode, you have an online connection to the automation system (F-CPU).

Rules for testing

⚠ WARNING

Shutdown of the safety program following changes to fail-safe outputs

In test mode of the *CFC Editor*, you can monitor safety programs and modify inputs of F-Blocks that are not interconnected. It is not permitted to change fail-safe outputs and automatically initialized inputs/outputs online; this causes the safety program to shut down.

10.7.1 Testing with S7-PLCSIM

Procedure

The *S7-PLCSIM* software package allows you to simulate a safety program on your ES.

You use the same procedure to simulate your safety program with *S7-PLCSIM* as in the standard case.

If you download the safety program in *S7-PLCSIM*, the "Set Up Access Rights" dialog appears. You are prompted for the F-CPU password.

(If you are using *S7 PLCSIM* V5.4 or earlier, enter plcsim (in lower-case letters). Here it makes no difference what password you set up for the F-CPU in *HW Config*.)

You can download changes in the safety program only as part of the complete safety program.

Note

If an F-STOP is triggered for the safety program, you must follow these steps:

- Reset the memory of the virtual F-CPU (*S7-PLCSIM*).
- Download the configuration data and the S7 program again.

⚠WARNING

A simulation is no substitute for a function test.

If the simulation takes place on an ES having an online connection to the F-CPU, you must not deactivate safety mode. Likewise, you must not be granted access authorization through the F-CPU password.

10.8 Modifying a safety program

Introduction

Changes in the safety program can be made offline as well as online. Online changes are made by means of the CFC test mode and take effect immediately. You must then download offline changes to the F-CPU.

Note

Safety program changes made otherwise, for example, by means of the "Monitor/Modify Variables" function, can lead to an F-STOP.

10.8.1 Online changes in CFC test mode

Introduction

In test mode of the *CFC Editor*, you can modify the values of non-interconnected inputs of F-Blocks during operation.

Rules

- For inputs in the safety data format, you may only modify the DATA component and not COMPLEM or PARID.
- You must not modify any outputs or any inputs not documented in the block description.

Requirements

Before you enable the test mode of the *CFC Editor*, make sure that the following requirements are met:

- The F-CPU must be in RUN mode.
- Safety mode of the safety program must be deactivated. Otherwise, when you attempt to change the first parameter you will be prompted to deactivate safety mode.

⚠ WARNING

Changing the collective signature for changes in CFC test mode

Making changes in the safety program in CFC test mode causes the collective signature to change. This means that the safety program must undergo acceptance testing again, if necessary.

Procedure

To modify the fail-safe block interfaces, you use the same standard procedure as in the *CFC Editor*.

The collective signature at output F_SIG_OUT of the the F_SHUTDN F-Block is set to "0" on the first change in CFC test mode and is updated after CFC test mode is exited.

⚠ WARNING

Do not change values created during compilation

When safety mode is activated, you are not permitted to directly operate safety programs! You may enter safety parameters for non-interconnected inputs:

- From the standard user program by means of F-conversion blocks with an additional validity check

  or
- In test mode of the *CFC Editor* and when safety mode is deactivated

  or
- With the Safety Data Write or Maintenance Override function

If this warning is not adhered to, an F-STOP is triggered. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: error detected" (event ID 16#75E1)

10.8.2 Downloading changes

Requirements

- Safety mode must be deactivated.
- S7 FH Systems must be in the redundant system state.

Procedure

1. Use the same procedure to download safety program changes as is standard when downloading changes in *CFC*. For more information, refer to the manual entitled "CFC for S7 Continuous Function Chart (http://support.automation.siemens.com/WW/view/en/21401430)".
2. Reactivate safety mode by answering the prompt that follows.
3. If necessary, repeat steps 1 and 2, e.g., to download step-wise changes.
4. In *SIMATIC Manager,* select the **Options > Edit Safety Program** menu command.
5. To do so, follow the procedure outlined in the section entitled "Acceptance test of safety program changes (Page 184)".

⚠WARNING

Download operation aborted

If the download operation aborts, you must download the changes again and recheck the collective signatures online and offline. You thereby ensure that the data in the load memory and work memory are consistent.

Note

Undoing a change

If you undo a change but nevertheless download it, it is possible that a different collective signature will be generated than prior to the change.

⚠WARNING

Moving F-Blocks or F-Runtime groups

Note that:

- F-Blocks that have been moved to a different F-Runtime group

  or

- F-Runtime groups that have been moved to a different task

may not be able to be processed at all when changes are downloaded over multiple processing cycles, or they may be processed multiple times.

⚠WARNING

Modifying the safety program in RUN mode

- Changes in the safety program in RUN mode when safety mode is deactivated can cause changeover effects to occur. You should implement organizational measures to ensure that such changes do not affect the safety of the system.
- To the extent possible, the standard user program and the safety program should be modified separately, and changes should be downloaded; otherwise, an error could be downloaded simultaneously to the standard user program, thus preventing a necessary protective feature in the safety program from taking effect or thus causing changeover effects to occur in both the safety program and the standard program.

Note

- Prior to downloading changes, please review the relevant FAQs (http://support.automation.siemens.com/WW/view/en/13711209/133000) on the Internet.
- Changes in automatically generated charts and F-Runtime groups are always forbidden and can cause an F-STOP. Exceptions to this are:
  - The MAX_CYC parameter of the F_CYC_CO blocks where you assign the F-monitoring time for a time interrupt OB
  - Parameter assignments in the F_SHUTDN block for the shutdown behavior

Note

Dividing/combining F-Runtime groups in running safety programs represents a significant change in the runtime sequence. Use the "Compare Safety Programs" dialog to check for shifted fail-safe module drivers prior to downloading changes.

These can lead to the following unintended behavior patterns while downloading changes in RUN mode:

- Passivation of output channels
- Processing of non-up-to-date input data in the input channels

Changing the runtime sequence causes the associated fail-safe module drivers to shift to other F-Runtime groups.

See also

Activating safety mode (Page 163)

10.8.2.1 Changes that can be transferred by downloading changes

You can transfer the following changes to the F-CPU by downloading changes.

If you do not observe the information in Chapter "Downloading changes (Page 169)" and the boundary conditions listed below, an F-STOP can be triggered for the safety program.

- Inserting new F-Runtime groups with new instances of F-Blocks/F-Block types.
- Inserting, modifying, and deleting interconnections of F-Blocks.
- Deleting and reinserting F-Blocks or moving F-Blocks in the runtime sequence within the F-Runtime group.
- Changing values of inputs and outputs of F-Blocks.

  Exception: Changes in safety-related communication between F-CPUs (see " Change in the safety-related communication between F-CPUs (Page 175) ")
- Moving of instances of F-Blocks/F-Block types between F-Runtime groups within an F-Shutdown group.
- Moving of instances of F-Blocks between F-Runtime groups of different F-Shutdown groups.

  Boundary condition: Note that all fail-safe channel drivers of an F-I/O must be contained in a common F-Shutdown group.
- Inserting/deleting F-Shutdown groups by means of F_PSG_M

  Boundary condition:
  - The must be no instances of F-Block types prior to the position in the F-Shutdown group where you insert or delete the F_PSG_M.
  - Note that all fail-safe channel drivers of an F-I/O must be contained in a common F-Shutdown group.
- Moving the F-Runtime groups that do not contain instances of F-Block types to another task.

  Boundary conditions:
  - Note that all fail-safe channel drivers of an F-I/O must be contained in a common F-Shutdown group.
- Adding F-I/O by means of CiR

  Boundary condition: Note the information about CIR in Chapter " Configuration in Run (CiR) (Page 59) ".

10.8.2.2 Changes requiring an F-Startup

The following changes require an F-Startup of the safety program. You cannot download these changes to the F-CPU without triggering an F-STOP; see the section entitled " F-STOP (Page 84) ". These changes may only be downloaded by means of a complete download.

- Dividing/combining F-Shutdown groups by means of F_PSG_M
  - There are instances of F-Block types prior to the position in the F-Shutdown group where you insert or delete the F_PSG_M.
- Moving of instances of F-Block types between different F-Shutdown groups.
- Moving of F-Runtime groups that do not contain instances of F-Block types to another task.

10.8.2.3 Changes that require a cold restart or warm restart (restart) of the F-CPU

The following changes take effect only after a cold restart or warm restart (restart) of the F-CPU:

- Change of values of the ID or R_ID parameters of the F_SENDR/BO, F_RCVR/BO, F_SDS_BO or F_RDS_BO F-Blocks. (See also Chapter " Change in the safety-related communication between F-CPUs (Page 175) ").

10.8.2.4 Changes that require an F-CPU STOP in a single CPU

You can make exactly the same changes to the hardware configuration in an S7 FH System as in an S7 H System; see Manual " Automation System S7-400H Fault-tolerant Systems (http://support.automation.siemens.com/WW/view/en/1186523) ".

If you are operating a non-redundant F-CPU, an F-CPU STOP is required to download these changes.

Special features for S7 FH Systems:

- The F-I/O can receive modified parameters in an S7 FH System only after removal and insertion. The F-I/O detect a communication error after the first change is downloaded.

10.8.2.5 Changing the time ratios or F-Monitoring times

Make sure that the time monitoring functions are not triggered when the time ratios or F-Monitoring times are changed.

- Changing the OB cycle time

Procedure for changing the OB cycle time

1. Using the newly specified value for the OB cycle time, calculate the minimum F-Monitoring times for:
   - F-Cycle time monitoring at input MAX_CYC at the F_CYC_CP F-control block
   - TIMEOUT inputs of the F-Blocks for safety-related communication between F-CPUs
   - TIMEOUT inputs of the F-Blocks for data exchange between F-Shutdown groups
   - F-I/O

   For more information about the F-monitoring time, refer to Chapter " Run times, F-Monitoring times, and response times (Page 410) ".
2. If the values assigned up to now are less than the newly calculated values, you must increase the F-Monitoring times prior to changing the OB cycle time. Compile the S7 program and download the changes.
3. Change the OB cycle time.

Note

Changing the OB cycle time involves a change in the hardware configuration. Refer to chapter " Changes that require an F-CPU STOP in a single CPU (Page 173) ".

- Moving of F-Runtime groups to a different task

  Corresponds to a change of the OB cycle times of the relevant tasks (see above).
- Changing of F-Monitoring times for F-Blocks for safety-related communication between F-CPUs and for data exchange between F-Shutdown groups.
- Changing the F-Monitoring times of an F-I/O.

Note

Changing the F-Monitoring times of an F-I/O involves a change in the hardware configuration. Refer to chapter " Changes that require an F-CPU STOP in a single CPU (Page 173) ".

When changing these F-Monitoring times, ensure that the values do not fall below the calculated minimum F-Monitoring times. For more information about the F-monitoring time, refer to Chapter " Run times, F-Monitoring times, and response times (Page 410) ".

10.8.2.6 Change in the safety-related communication between F-CPUs

Introduction

If the safety-related communication between F-CPUs is to continue to run in all phases, you must proceed in multiple steps.

Rule

You must never simultaneously change the interconnection for a send data element at F_SENDBO/F_SDS_BO/F_SENDR and for the associated receive data element at F_RCVBO/F_RDS_BO/F_RCVR. The simultaneous activation of the new interconnections is otherwise not ensured.

Procedure for changing interconnections

For changing an interconnection to a send data element of the F_SENDBO/F_SDS_BO/F_SENDR F-Blocks or from a receive data element of the F_RCVBO/F_RDS_BO/F_RCVR F-Blocks, the following sequence must be adhered to:

1. Interconnect the new data element to be sent with a previously unused input SD_BO_xx/SD_R_xx of the F_SENDBO/F_SDS_BO/F_SENDR. Compile the S7 program and download the change.

   Result: The new data element is now available at the corresponding RD_BO_xx/RD_R_xx output of F_RCVBO/F_RDS_BO/F_RCVR.

2. Now, interconnect the blocks again to the new RD_BO_xx/RD_R_xx output for further processing of the received signals. Compile the S7 program and download the change.

   Result: Through this method, you ensure a consistent switchover to the new data path.

3. Delete the superfluous interconnection at F_SENDBO/F_SDS_BO/F_SENDR.
4. Compile the S7 program and download the change.

Procedure for replacing the communication partner

When a communication partner is replaced, the following sequence must be adhered to:

1. Configure the new S7 connection in *NetPro*. Download the connection data in RUN mode.
2. Place a new instance of F_SENDBO/F_SDS_BO/F_SENDR on the sender side. Assign the data for the new S7 connection to the ID and R_ID inputs. Interconnect the new data element to be sent with the SD_BO_xx/SD_R_xx inputs of the F_SENDBO/F_SDS_BO/F_SENDR. Compile the S7 program and download the change.
3. Place a new instance of F_RCVBO/F_RDS_BO/F_RCVR on the receiver side. Assign the data for the new S7 connection to the ID and R_ID inputs.

   Compile the S7 program and download the change.

   Result: The data of the old and new communication partner are now available to you on the receiver side.
4. Now, interconnect the blocks again to the RD_BO_xx/RD_R_xx outputs of the new R_RCVBO/F_RDS_BO/F_RCVR for further processing of the received signals.

   Delete the superfluous F_RCVBO/F_RDS_BO/F_RCVR. Compile the S7 program and download the change.

   Result: Through this method, you ensure a consistent switchover to the new communication partner.
5. Delete the superfluous F_SENDBO/F_SDS_BO/F_SENDR. Compile the S7 program and download the change.
6. If applicable, delete the superfluous connection from *NetPro*. Download the connection data in RUN mode.

10.8.2.7 Initial run and startup characteristics

Newly inserted F-Blocks execute an initial run after online changes. In this regard, note the startup characteristics described in the block descriptions. In cases where the initial run is not specifically mentioned, the behavior described after an F-Startup also applies to the initial run.

10.9 Deleting the safety program

Procedure

To delete a safety program from an F-CPU, follow these steps:

1. Delete all F-Charts from the chart folder. The symbols of these charts are highlighted with a yellow background in *SIMATIC Manager*.
2. Delete all charts whose name begins with "@F_".
3. Compile the S7 program with the "Generate module drivers" option selected.
4. In *HW Config*, open the properties dialog for the F-CPU from which you want to delete the safety program. Clear the "CPU contains safety program" option under "Protection".
5. Compile the hardware configuration.
6. Compile the S7 program.

10.10 Acceptance test following system upgrade

Acceptance test following system upgrade

The table below shows you whether migration to *S7 F Systems* V6.1 results in a signature change or requires an F-CPU STOP or a new acceptance test.

Migration from	Signature change	F-CPU STOP required	New acceptance test required
S7 F Systems V5.2 without an F-Library update	No	No	No
S7 F Systems V5.2 SPx without an F-Library update	No	No	No
S7 F Systems V6.0 without an F-Library update	No	No	No
Failsafe Blocks (V1_2) to *S7 F Systems Lib* V1_3	Yes	Yes	Changes[2]
Failsafe Blocks (V1_2) SPx to *S7 F Systems Lib* V1_3 SP1	Yes	Yes	Changes[2]
S7 F Systems Lib V1_3 to *S7 F Systems Lib* V1_3 SP1			
When using the new F-Blocks	Yes	No	Changes[2]
When using the modified F_CH_DO	Yes	Yes[1]	Changes[2]
When using the modified F_CH_BI	Yes	No	Changes[2]
When using the modified F_QUITES	Yes	No	Changes[2]
When using the modified F_CH_AI	Yes	No	Changes[2]
When using the modified F_PA_AI	Yes	No	Changes[2]
When using the modified F_SQRT	Yes	No	Changes[2]
When using the modified F_CHG_BO	No	No	Changes[2]
When using the modified F_CHG_R	No	No	Changes[2]

1: The change is not safety-related and does not affect the usability of the existing project.
2: With *S7 F Systems* V6.1, the acceptance test for changes is minimized.

See also

Acceptance test of safety program changes (Page 184)

System Acceptance Test

11

11.1 Overview of system acceptance test

Introduction

During the system acceptance test, all relevant application-specific standards must be adhered to as well as the following procedures. This also applies to systems that are not subject to acceptance testing. For acceptance testing, you must note the systems requiring approval in the Certification Report.

As a general rule, the acceptance test of an F-System is performed by independent experts.

Special functions in *SIMATIC Manager* assist you for the acceptance test of an F-System. You can use these functions to:

- Compare safety programs
- Log safety programs
- Print safety programs

All data relevant to the acceptance test of the S7 F System can be archived in *SIMATIC Manager* (**File > Archive**) and printed as needed.

For more information, refer to Chapters "Comparing safety programs (Page 153)", ""Logs..." button (Page 151)" and "Printing project data of the safety program (Page 160)".

11.2 Commissioning a safety program

General procedure for the initial acceptance test of a safety program

1. Preliminary test of the configuration of the F-CPU and F-I/O (optional)
2. Backup of the *STEP 7* project
3. Inspection of the printout
4. Downloading the S7 program to the F-CPU
5. Implementation of a complete function test

11.2.1 Preliminary test of the configuration of the F-CPU and F-I/O (optional)

Introduction

After you finish configuring the hardware and assigning parameters for the F-CPU and F-I/O, you can perform an initial acceptance test for the F-I/O configuration.

In order to do this, the hardware configuration data must be printed out, checked, and saved together with the overall *STEP 7* project.

Printing hardware configuration data

1. Select the correct F-CPU or S7 program assigned to it.
2. In *SIMATIC Manager,* select the **Options > Edit Safety Program** menu command.

   The "Safety Program" dialog will appear.
3. Click the "Print" button and select the "HW Configuration" option in the next dialog:
4. Select "All" for the print range, and select the "Module description" and "Address list" options there. In addition, select the "Including parameter description" option to include your parameter descriptions in the printout.

Checking hardware configuration data

1. Check the parameters of the F-CPU in the printout.

   In safety mode, access by means of the F-CPU password must not be authorized when making changes to the standard user program, since changes to the safety program can also be made. To rule out this possibility, you must configure **Protection Level 1**. In addition, you must select the "CPU contains safety program" option. The corresponding protection level and "CPU contains safety program" is included in the printout.
2. Check the safety-related parameters of the F-I/O in the printout.

   These safety-related parameters can be found in the printout for the respective F-I/O. The data are structured differently according to the F-I/O as follows:

   SM 326; DI 24 x DC 24 V (Order No. 6ES7326-1BK00-0AB0), SM 326; DI 8 x Namur, SM 326 DO 10 x DC 24V/2A and SM 336; AI 6 x13 Bit

   – The PROFIsafe source address does not appear in the printout.

   – You determine the PROFIsafe destination address from the address value under "Addresses – Inputs – Start". Divide this address value by "8".

   – The safety-related parameters are found under "Parameters – Basic Settings" or "Parameters – Input/Output x".

   ET200S, ET 200pro, ET 200eco fail-safe modules, SM 326; DI 24 x DC 24 V (as of Order No. 6ES7326-1BK01-0AB0) and SM 326; DO 8 x DC 24V/2A PM

   – The PROFIsafe source address is found under "Parameters – F-Parameters – F_Source_Address".

   – The PROFIsafe destination address is found under "Parameters – F-Parameters – F_destination_address".

   – The safety-related parameters are found under "Parameters – F-Parameters" and "Parameters – Module parameters".

Fail-safe DP standard slaves

- The PROFIsafe source address is found under "PROFIsafe – F_Source_Add".
- The PROFIsafe destination address is found under "PROFIsafe – F_Dest_Add".
- The safety-related parameters are found under "PROFIsafe".

  For information on handling of any process- and safety-related parameters, refer to the documentation for the respective DP standard slave.

3. Once the safety-related parameters of an F-I/O module are checked, the parameter CRCs in the printout are sufficient as reference for further acceptance testing. These parameter CRCs have the following appearance (address/F-address = PROFIsafe address):

   S7-300 fail-safe signal modules (SM 326; DI 24 x DC 24 V, with Order No. 6ES7326-1BK00-0AB0; SM 326; DI 8 x NAMUR; SM 326; DO 10 x DC 24V/2A; SM 336; AI 6 x 13Bit)

   - Parameter CRC (including address): 12345
   - Parameter CRC (excluding address): 54321

   ET200S, ET 200pro, ET 200eco fail-safe modules and S7-300 fail-safe signal modules (SM 326; DI 24 x DC 24 V, as of Order No. 6ES7326-1BK01-0AB0; SM 326; DO 8 x DC 24V/2A PM)

   - Parameter CRC: 12345
   - Parameter CRC (excluding F-addresses): 54321

   Fail-safe DP standard slaves

   - F_Par_CRC: 12345
   - F_Par_CRC (excluding F-addresses): 54321

   F-I/O that are to be assigned the same safety-related parameters can be copied during configuration. All safety-related parameters for these no longer have to checked individually: It is sufficient to compare every other CRC (for example, "Parameter CRC (excluding address)") of the copied F-I/O to the corresponding CRC of the previously checked F-I/O and to check the PROFIsafe source and destination addresses.

4. Check that the PROFIsafe addresses are unique from one another.

   To determine the PROFIsafe addresses of individual F-I/O, refer to step 1.

⚠WARNING

Rule for PROFIBUS subnets:

The PROFIsafe destination address and, thus, the switch setting on the address switch of the F-I/O must be unique network-wide* and station-wide** (system-wide). You can assign up to 1022 different PROFIsafe destination addresses.

* A network consists of one or more subnets. "Network-wide" means across subnet boundaries.

** "Station-wide" means for one station in *HW Config* (e.g., an S7-400H station).

11.2.2 Backup of the STEP 7 project

Requirements

Prior to the acceptance test, compile the safety program to be tested.

Backing up and archiving

The safety program that is to undergo approval testing must be backed up and archived with the complete *STEP 7* project. You must print out all of the project data *unfiltered* and archive them together with the *STEP 7* project:

- Chart (standard chart and F-Chart)
- Safety program: Block lists and signatures
- Safety-related parameters
- HW configuration
- Compilation log
- Download log

The procedure for backing up and archiving *STEP 7* projects is described in the basic help of *STEP 7*.

11.2.3 Inspection of the printout

Introduction

Print the entire project as described in the section entitled "Printing project data of the safety program (Page 160)".

Printout

The printout contains the collective signature as a reference. The collective signature appears twice in the printout: Once in the program section as a value of the block container and in the footer as a value from the source. The values must match.

The version number of the utilized *S7 F Systems* optional package appears in the footer of the printout and must be checked by you.

If a collective signature is not printed in the footer, this means that the safety program or the configuration (*HW Config* or *NetPro*) has changed. In this case, you must recompile the safety program.

Check of safety-related parameters

Check the values of all safety-related parameters in the corresponding section of the printout for the safety program.

The following will be printed out:

- Values of all non-interconnected, invisible input parameters
- Values of all special input parameters to be checked, such as F-Monitoring times

The following will be printed out and marked with an asterisk (*):

- Values of all output parameters for which the runtime sequence does not correspond to the data flow

  This is the case if the F-Block is first called after the output parameter was already transferred to another F-Block, for example, in a feedback loop.
- Inputs or outputs on an F-Block that have been identified by the system as parameters to be taken into account in the printout

Checking the signatures and initial value signatures of the F-Blocks

The signatures and initial value signatures of all F-Blocks must match those in Annex 1 of the Certificate Report.

Checking the signatures and initial value signatures of the F-Block types

The signatures and initial value signatures of all F-Block types must match those in the acceptance test documents of the F-Block types (see section entitled "Acceptance test of F-Block types (Page 184)").

The acceptance test documents of the F-Block types also list the signatures and initial value signatures of all called F-Blocks. These signatures must also match those in the safety program.

11.2.4 Downloading the S7 program to the F-CPU

Introduction

You download the S7 program to the F-CPU as described in Chapter " Downloading the safety program (Page 164) ". Afterwards, you check the signatures.

Checking the collective signature

After downloading the S7 program to the F-CPU, you must compare the collective signature of the safety program in the F-CPU to the collective signature in the accepted printout. S7 FH Systems must be in the "Redundant" system state, and safety mode must be activated.

You obtain the collective signature of the safety program and the signatures of the F-Blocks in the F-CPU using the **Options > Edit Safety Program** menu command.

11.3 Acceptance test of safety program changes

Procedure

To perform an acceptance test on your safety program changes, follow these steps:

1. Back up your safety program.
2. Compare your new safety program with your accepted safety program. For more information, refer to Chapter " Comparing safety programs (Page 153) ".
3. Inspect the changes in the printout. You must locate the changes that you made to your safety program on the printout again. Check the signature in the printout (and in the footer). To do so, follow the same procedure as for the initial acceptance test.
4. Download your modified safety program to the F-CPU.
5. Perform a function test of your changes.

11.4 Acceptance test of F-Block types

Initial acceptance test

The procedure for the initial acceptance test of a newly created F-Block type is the same as for the initial acceptance test of a safety program. The function test of the F-Block type must be performed in a different safety program as the test environment.

The signature and initial value signature of the F-Block generated from the F-Block type is relevant for acceptance testing of F-Block types. You can obtain these signatures from the safety program printout. In addition, you must also check the signatures and initial value signatures of the called F-Blocks.

The collective signatures in the footers of the printouts of the safety program and the CFC chart of the F-Block type must match; otherwise, you must recompile the F-Block type.

All F-Blocks called in an F-Block type must be compared.

Note

You must check the signatures of the F-Block type and the signatures of all called F-Blocks for the test of a safety program in which an F-Block type is used.

Acceptance test of changes

The procedure for the acceptance test of F-Block type changes is the same as for a safety program.

For the acceptance test of the F-Block types, you use a printout to document the signature and initial value signature of the new F-Block type as well as the signatures and initial value signatures of all F-Blocks called in the F-Block type.

In addition, you must perform a function test to check all locations in the test safety program where the new F-Block type is called. Modified signatures of F-Blocks are indicated when safety programs are compared in the chart view.

Modified calculation of signatures of F-Block types with the *Failsafe Blocks* F-Library (V1_2)

In V5.2 SP4 and higher, the initial value signature of the F-Block types is calculated independent of the content of the block container of the F-Block type. In versions up to V5.2 SP3, different initial value signatures were calculated, depending on whether or not the F-Blocks called from the F-Block type are contained in the S7 program. Provided you have calculated the initial value signature for F-Block types you created yourself in a tested (executable, complete) S7 program, they remain unchanged. This pertains to user-created F-Block types and F-Blocks F_1oo2_R and F_2oo3_R of the *Failsafe Blocks* F-Library(V1_2).

- User-created F-Block types:

  If necessary, correct the signatures of the user-created F-Block types in your documentation.

- F-Blocks F_1oo2_R and F_2oo3_R:

  The signatures specified in Annex 1 of the Certificate Report have been added, accordingly. The F-Blocks themselves are not changed.

Note

Change of initial value signature, although the F-Block type has not changed.

In *S7 F Systems* V5.2 SP4 and higher, the calculation of the initial value signature of F-Block types has changed. This results in output of a modified initial value signature, although the F-Block type has not changed.

Another acceptance test is *not* required if you adhere to the following steps. To calculate the corrected initial value signature of an F-Block type, follow these steps:

1. Open the "Edit Safety Program" dialog with the safety program that you want to use to perform the acceptance test of the F-Block type. For this purpose, use your previous version of *S7 F Systems* (version prior to V5.2 SP4).
2. Generate a safety printout again and consult the accepted safety printout to make sure that the signature of the F-Block type and the charts are identical to your printout.
3. Install the new version of *S7 F Systems* (V5.2 SP4 or higher). You do not have to compile again since you already ensured the identity of the safety program with the accepted version.
4. Open the "Edit Safety Program" dialog.
5. Generate a printout of the safety program.
6. Document the signatures in the printout along with the version of *S7 F Systems* to which each signature applies.

Operation and Maintenance 12

12.1 Notes on safety mode of the safety program

Introduction

The rules and safety information for operation of S7 F/FH Systems is presented below.

Using simulation devices / simulation programs

⚠ WARNING

If you operate simulation devices or simulation programs that generate safety message frames, e.g., in accordance with PROFIsafe, and make them available to the S7 F/FH System via the bus system (e.g., PROFIBUS DP), you must ensure the safety of the F-system using organizational measures, e.g., such as operational monitoring and manual safety shutdown.

If you use the *S7-PLCSIM* function of *STEP 7* to simulate safety programs, these measures are not necessary because *S7-PLCSIM* cannot establish an online connection to a real S7 component.

Note, for example, that a protocol analyzer may not perform functions that reproduce recorded message frame sequences with correct time behavior.

STOP by means of ES operation, mode selector, or communication function

⚠ WARNING

Switching from STOP to RUN mode by means of an ES operation, mode selector, or communication function is not interlocked. For example, only one keystroke on the ES is necessary to switch from STOP to RUN mode. For this reason, a STOP that you have set by means of an ES operation, mode selector, or communication function must not be regarded as a safety condition.

Therefore, always switch off the F-CPU directly at the device when performing maintenance work.

Placing F-CPU in STOP with SFC 46 "STP"

⚠WARNING

A STOP state initiated with SFC 46 "STP" can be canceled very easily (and unintentionally) by means of an ES operation. For this reason, an F-CPU STOP initiated by SFC 46 is not a fail-safe STOP.

Fiber-optic cable between the synchronization modules in S7 F/FH Systems

⚠WARNING

Two F-CPUs not simultaneously as master system

In S7 F/FH Systems, you must ensure that the two F-CPUs are not master systems simultaneously. Otherwise, this could lead to dangerous errors.

This situation (both F-CPUs as master simultaneously) can occur if the two fiber-optic cables used to connect the F-CPUs in S7 F/FH Systems in the redundant system state are unplugged or interrupted simultaneously. You must prevent this by routing the fiber-optic cables separately.

This situation (both F-CPUs as master simultaneously) can also occur after an F-CPU is repaired if the F-CPUs have not yet been connected using *both* fiber-optic cables prior to switching on the power supply.

You must implement organizational measures to ensure following replacement of an F-CPU that both connections are established using the fiber optic cables *prior* to switching on the power supply.

Additional Information

Information about replacing components in fault-tolerant systems can be found in Manual " Automation System S7-400H Fault-tolerant Systems (http://support.automation.siemens.com/WW/view/en/1186523) ".

12.2 Replacing software and hardware components

Replacement of software components

When replacing software components on your ES, e.g., with a new version of *PCS 7* or *STEP 7*, you must observe the notes regarding upward and downward compatibility in the documentation and readme files for these products.

Installing new versions of software packages

After installation of a new version of *STEP 7* or the *CFC*, *SCL*, etc., optional packages, follow these steps:

1. Compile the S7 program in the new environment.
2. Compare the collective signature of the newly compiled S7 program to the collective signature of the accepted safety program (see also Section "Checking the Collective Signature" in Chapter "Commissioning a safety program (Page 179)").
3. If the collective signatures are identical, the safety programs match.
4. If the collective signatures are not identical, the safety program has been changed. In this case, follow the same procedure as for a safety program change.

Replacement of hardware components

Hardware components for S7 F/FH Systems (modules, batteries, etc.) are replaced in the same way as in standard mode.

Removing and inserting F-I/O during operation

It is possible to remove and insert F-I/O during operation the same as with standard 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 at input ACK_REI of the fail-safe channel driver. Otherwise, the F-I/O will remain passivated.

CPU operating system update

Check of the CPU operating for F-validity: When using a new CPU operating system (operating system 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 Certification Report. This information and any notes on the new CPU operating system must be taken into account.

Operating system update for interface modules

When using a new operating system for an interface module, e.g., IM 151-1 HIGH FEATURE of ET 200S (operating system update, see online help for *STEP 7*), you must observe the following:

If the "Activate firmware after download" check box is selected for the operating system update, the IM will be automatically reset following a successful loading operation and will then run on the new operating system. The entire F-I/O is passivated after startup of the IM.

The F-I/O is reintegrated in the same way as when a communication error occurs, that is, an acknowledgment at input ACK_REI of the fail-safe channel driver is required.

Duration of repair for S7 F/FH Systems

For S7 F/FH Systems, the repair of redundant components should be organized in such a way that when a failure occurs, the repair should not take more than 24 hours, if possible. A repair duration of 72 hours on weekends is also possible for unmanned systems. Basically, availability increases as the repair duration decreases.

Fiber-optic cables in S7 F/FH Systems

After repair of an F-CPU, you must not unplug the fiber-optic cables from the F-CPU simultaneously.

Preventive maintenance (proof test)

The probability values for the certified F-System components guarantee a *proof-test interval of 10 years* for ordinary configurations. For detailed information, refer to the F-I/O manuals. Proof test for complex electronic components generally means replacement with unused items. If for particular reasons you require a proof-test interval in excess of 10 years, contact your SIEMENS representative.

As a rule, shorter proof-test intervals are required for sensors and actuators.

Removing *S7 F Systems*

For information about removing the software, refer to Chapter "Installing the S7 F Systems optional package V6.1 (Page 27)".

You disassemble and dispose of the hardware of an F-System the same as for standard automation systems. For more information, refer to the *Hardware Manuals*.

12.3 F-Forcing

Introduction

Depending on the *CFC* version you are using, *S7 F Systems* V6.1 and later with *S7 F Systems Lib* V1_3 SP1 supports F-Parameter forcing in deactivated safety mode.

F-Forcing allows you to modify F-Parameters at user interconnections. The modification of F-Parameters at system interconnections is not supported.

Consult the documentation for *CFC* or *PCS 7* to find out which *CFC* versions support forcing of F-Parameters, in particular.

⚠ WARNING
Forcing is only permitted when the safety of the system is ensured by other measures.

Procedure

1. Configure forcing for F-Parameters in *CFC* using the same procedure as for forcing with standard parameters.
2. If you haven't already done so, you will be prompted to deactivate safety mode.
   - Modify and check the force values for F-Parameters.
   - Enable F-Forcing for F-Parameters.
3. In your *CFC* program, make changes to F-Parameters of user interconnections by means of F-Forcing.
4. Activate safety mode again when forcing is no longer taking place in the F-Parameters.

Note

F-Forcing is deactivated automatically any time the F-Program starts up. The F-Program starts up:

- Each time the CPU restarts (cold/warm restart), e.g., following a brief power outage
- Each time the CPU restarts after a full shutdown

Note

Safety mode cannot be activated if F-Forcing is activated for an F-Parameter.

Note

F-Forcing is a typical commissioning function. The final F-Program should not include F-Forcing of F-Parameters.

Use the Maintenance Override function for the maintenance functions. For more information about the Maintenance Override function, refer to the section entitled "Maintenance Override function (Page 109)".

F-Libraries

A

A.1 Overview of F-Library S7 F Systems Lib V1_3 SP1

Overview

In the *S7 F Systems Lib* F-Library V1_3 SP1, you will find:

- In the "F-User Blocks\Blocks" block container: F-Blocks
- In the "F-Control Blocks\Blocks" block container: F-Control blocks

Note

Refer also to the sections entitled "Differences between the F-Libraries Failsafe Blocks (V1_x) and S7 F Systems Lib V1_3 (Page 387)" and "Differences between the F-Library S7 F Systems Lib V1_3 and V1_3 SP1 (Page 409)".

Note

You must not change the F-Library name.

Note

FB numbers of F-Blocks

You must not change the F-Block numbers.

The following new F-Block that have been added to *S7 F Systems Lib* V1_3 SP1 use FBs that are also used in *S7 Distributed Safety*:

S7 F Systems Lib V1_3 SP1	FB number	*Distributed Safety (V1)* F-Library
F_CH_DII	FB 465	F_IGNTR
F_CH_DIO	FB 466	F_TIGHTN
F_POLYG	FB 467	F_GAS_BU
F_INT_P	FB 468	F_OIL_BU
F_PT1_P	FB 469	F_AIRD

A.1.1 F-Data types

Function

Special F-Data types in a safety data format are used for fail-safe block interfaces. The safety data format is used to expose data and address errors.

Example

F_BOOL:	
	STRUCT
DATA	BOOL
PAR_ID	WORD
COMPLEM	WORD
	END_STRUCT

If you want to change the value (default) of a block interface with an F-Data type, you can only change the DATA component.

⚠WARNING

Values of PAR_ID and COMPLEM must not be changed

You must not change the PAR_ID and COMPLEM components after the S7 program has been compiled since this might result in serious errors remaining undetected. If errors in the safety data format are detected while the safety program is running, an F-STOP is triggered. You must recompile the S7 program and download it to the F-CPUs, if necessary.

A.1.2 Block interfaces

Note the following special features regarding the block interfaces of F-Blocks:

- The EN and ENO inputs/outputs are neither evaluated nor assigned by the program code of the F-Block and you must not interconnect them.
- All F-Blocks have additional inputs/outputs in addition to the inputs/outputs documented in the following block descriptions. These are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

A.1.3 Behavior of F-Blocks with floating-point operations in the event of a number range overflow

The "Overflow (± infinity)", "Denormalized floating-point number", or "Invalid floating-point number (NaN)" events are:

- Either output at the output and available for further processed by the subsequent F-Blocks

  or

- Signaled at special outputs. If necessary, a fail-safe value is output.

If the floating-point operation yields an invalid floating-point number (NaN) and an invalid floating-point number (NaN) does not already exist as an address, the following diagnostic event is entered in the diagnostic buffer of the F-CPU:

- "Safety program: invalid REAL number in DB" (Event ID 16#75D9)

You can use this diagnostic buffer entry to identify the F-Block with the invalid floating-point number (NaN).

Refer also to the documentation of the F-Blocks.

If you cannot rule out the occurrence of these events in your safety program, you must decide independently of your application whether you have to react to these events in your safety program. With F-Block F_LIM_R, you can check the result of a floating-point operation for overflow (± infinity) and invalid floating-point number (NaN).

A.1.4 Behavior of F-Blocks in the event of safety-related faults

If F-Blocks or F-Control blocks detect a safety-related fault, they trigger a fault reaction. Error information is entered in the diagnostic buffer of the F-CPU. The online help for the diagnostic events provides detailed information and suggests corrective actions.

The respective fault reactions and other diagnostic options can be found in the documentation for the F-Blocks and F-Control blocks.

A.2 F-Blocks in S7 F Systems Lib V1_3 SP1

A.2.1 Logic blocks with the BOOL data type

Overview

Block name	Block number	Description
F_AND4	FB 301	AND logic operation on four inputs
F_OR4	FB 302	OR logic operation on four inputs
F_XOR2	FB 303	XOR logic operation on two inputs
F_NOT	FB 304	NOT logic operation
F_2OUT3	FB 305	2oo3 evaluation of inputs of data type BOOL
F_XOUTY	FB 306	XooY evaluation of inputs of data type BOOL

A.2.1.1 F_AND4: AND logic operation on four inputs

Function

This block links the INx inputs by means of AND. The OUT output is "1" when all INx inputs are "1". Otherwise the OUT output is "0". The OUTN output corresponds to the negated OUT output.

Truth table

IN1	IN2	IN3	IN4	OUT	OUTN
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	1	0

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_BOOL	Input 1	1
	IN2	F_BOOL	Input 2	1
	IN3	F_BOOL	Input 3	1
	IN4	F_BOOL	Input 4	1
Outputs:	OUT	F_BOOL	Output	1
	OUTN	F_BOOL	Negated output	0

Error handling

None

A.2.1.2 F_OR4: OR logic operation on four inputs

Function

This F-Block combines the INx inputs with a logical OR. The OUT output is "1" when at least one INx input is "1". If all INx inputs are "0", the OUT output is "0". The OUTN output corresponds to the negated OUT output.

Truth table

IN1	IN2	IN3	IN4	OUT	OUTN
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	1	0

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_BOOL	Input 1	0
	IN2	F_BOOL	Input 2	0
	IN3	F_BOOL	Input 3	0
	IN4	F_BOOL	Input 4	0
Outputs:	OUT	F_BOOL	Output	0
	OUTN	F_BOOL	Negated output	1

Error handling

None

A.2.1.3 F_XOR2: XOR logic operation on two inputs

Function

This F-Block combines the INx inputs with an exclusive OR. The OUT output is "1" if exactly one INx input is "1". The OUTN output corresponds to the negated OUT output.

Truth table

IN1	IN2	OUT	OUTN
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_BOOL	Input 1	0
	IN2	F_BOOL	Input 2	0
Outputs:	OUT	F_BOOL	Output	0
	OUTN	F_BOOL	Negated output	1

Error handling

None

A.2.1.4 F_NOT: NOT logic operation

Function

This F-Block inverts the input.

Truth table

IN	OUT
0	1
1	0

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	F_BOOL	Input	0
Output:	OUT	F_BOOL	Output	1

Error handling

None

A.2.1.5 F_2OUT3: 2oo3 evaluation of inputs of data type BOOL

Function

This F-Block monitors three binary inputs for signal state "1". The OUT output is "1" when at least two INx inputs are "1". Otherwise the OUT output is "0". The OUTN output corresponds to the negated OUT output.

Truth table

IN1	IN2	IN3	OUT	OUTN
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_BOOL	Input 1	0
	IN2	F_BOOL	Input 2	0
	IN3	F_BOOL	Input 3	0
Outputs:	OUT	F_BOOL	Output	0
	OUTN	F_BOOL	Negated output	1

Error handling

None

A.2.1.6 F_XOUTY: XooY evaluation of inputs of data type BOOL

Function

This F-Block monitors up to 16 binary inputs IN1 to IN16 for signal state "1". The input signals are monitored for signal state "1" starting with the IN1 input up to and including the INY input. The number of binary inputs to be monitored is set with the Y parameter. The OUT output is 1 when at least x inputs IN1 to IN16 are "1". Otherwise the OUT output is "0". The OUTN output corresponds to the negated OUT output.

The binary inputs must be assigned consecutively starting with IN1. If X > Y, X ≤ 0, X > 16, Y ≤ 0, the OUT output is "0". If Y > 16, the OUT output behaves the same as if Y = 16.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_BOOL	Input 1	0
	IN2	F_BOOL	Input 2	0
	IN3	F_BOOL	Input 3	0
	…		…	
	IN16	F_BOOL	Input 16	0
	X	F_INT	Minimum number of inputs with "1": 0 < X ≤ 16	0
	Y	F_INT	Number of inputs to be monitored: 0 < Y ≤ 16	0
Outputs:	OUT	F_BOOL	Output	0
	OUTN	F_BOOL	Negated output	1

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the data buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.2 F-Blocks for F-Communication between F-CPUs

Overview

F-Block name	Block number	Description
F_SENDBO	FB 370	Sending of 20 data elements of data type F_BOOL in a fail-safe manner to another F-CPU
F_RCVBO	FB 371	Receiving of 20 data elements of data type F_BOOL in a fail-safe manner from another F-CPU
F_SENDR	FB 372	Sending of 20 data elements of data type F_REAL in a fail-safe manner to another F-CPU
F_RCVR	FB 373	Receiving of 20 data elements of data type F_REAL in a fail-safe manner from another F-CPU
F_SDS_BO	FB 352	Sending of 32 data elements of data type F_BOOL in a fail-safe manner to another F-CPU
F_RDS_BO	FB 353	Receiving of 32 data elements of data type F_BOOL in a fail-safe manner from another F-CPU

See also

Run times, F-Monitoring times, and response times (Page 410)

A.2.2.1 F_SENDBO: Sending of 20 data elements of data type F_BOOL in a fail-safe manner to another F-CPU

Function

The F_SENDBO F-Block sends the data of data type F_BOOL pending at the SD_BO_xx inputs in a fail-safe manner to another F-CPU. The data must be received there using the F_RCVBO F-Block.

To reduce the bus load, you can temporarily shut down communication between the F-CPUs. To do so, supply input EN_SEND with "0" (default = "1"). Send data are then no longer sent to the associated F_RCVBO and the assigned fail-safe values are made available to F_RCVBO during this time period. If communication was already established between the connection partners, a communication error is detected.

You must specify the local ID of the S7 connection from the perspective of the F-CPU (from the connection table in *NetPro*) at input ID.

Communication between F-CPUs takes place hidden in the background by means of a special safety protocol. You must define a communication association between an F_SENDBO in one F-CPU and an F_RCVBO in the other F-CPU by assigning an odd number at the R_ID input of the F_SENDBO and F_RCVBO. Associated F_SENDBOs and F_RCVBOs receive the same value for R_ID.

⚠ WARNING

Value for the relevant address reference

The value for each address association (input parameter R_ID; data type: DWORD) is user-defined; however, it must be unique from all other safety-related communication connections in the network. 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 F-Block.

You must assign the desired F-monitoring time at input TIMEOUT. The TIMEOUT input cannot be interconnected.

⚠ WARNING

Measure and transfer signal level

It can be ensured (from a fail-safe standpoint) that a signal level to be transferred will be detected on the sender side and transferred to the receiver only if the signal is pending for at least as long as the assigned F-Monitoring time (TIMEOUT).

For information about calculating F-Monitoring times, refer to chapter " Run times, F-Monitoring times, and response times (Page 410) ".

Note

If the data are received with the F_RCVBO F-Block of the *Failsafe Blocks* F-Library (V1_2) or (V1_1), you must assign input EN_SMODE with "0" (default = "1"). Otherwise, a CRC error will be detected by F_RCVBO.

In all other cases, you must accept the default setting for input EN_SMODE so that the operating mode of the F-CPU with the F_SENDBO can be evaluated at the SENDMODE output of F_RCVBO.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	EN_SEND	BOOL	1 = ENABLE SEND	1
	ID	WORD	ADDRESS PARAMETER ID	W#16#0
	R_ID	DWORD	ADDRESS PARAMETER R_ID	DW#16#0
	SD_BO_00	F_BOOL	SEND DATA 00	0
	…		…	
	SD_BO_19	F_BOOL	SEND DATA 19	0
	CRC_IMP	DWORD	ADDRESS RELATION CRC	DW#16#0 Automatically initialized *
	TIMEOUT	F_TIME	F MONITORING TIME	T#0ms
	EN_SMODE	F_BOOL	1 = ENABLE SENDMODE	1
Outputs:	ERROR	F_BOOL	1 = COMMUNICATION ERROR	0
	SUBS_ON	F_BOOL	1 = SUBSTITUTE VALUES USED FROM RECEIVER	0
	RETVAL	WORD	ERROR CODE	W#16#0

*) Input CRC_IMP is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input CRC_IMP is indicated as changed if changes have been made to the connection configuration in *NetPro*.

Fail-safe value

Fail-safe values are output from the receiver F_RCVBO in the following cases:

- A communication error (e.g., CRC error, timeout) has been detected.
- Communication has been canceled with EN_SEND = 0.
- An F-Startup is pending.

The SUBS_ON output is set to 1.

If the output of the fail-safe value is caused by a communication error, output ERROR = 1 is set additionally.

A "Timeout" communication error is not detected unless communication between the F_SENDBO and F_RCVBO connection partners has already be established once. If communication cannot be established after startup of the sending and receiving F-Systems, check the configuration of the safety-related CPU-CPU communication, F_SENDBO and F_RCVBO parameter assignment, and the bus connection. You can also find possible causes of error by evaluating the RETVAL outputs of the F_SENDBO and F_RCVBO. In general, you should always evaluate RETVAL of the F_SENDBO and F_RCVBO because it is possible that only one of the two outputs contains error information.

Reintegration

After a communication error, the data from the receiver pending at the SD_BO_xx inputs are only output again if the communication error is no longer detected and the error has been acknowledged with a positive edge at the ACK_REI input of F_RCVBO.

Startup characteristics

After the sending and receiving F-Systems are started up, communication must be established initially between the F_SENDBO and F_RCVBO connection partners. The receiver F_RCVBO makes fail-safe values available during this time period. The SUBS_ON output is set to 1.

Output RETVAL

The RETVAL output provides non-fail-safe information on the type of communication errors that occurred for service purposes. You can read out this information on your ES/OS or, if necessary, the information can be evaluated in your standard user program. The DIAG bits are saved until acknowledgment at input ACK_REI of the associated F_RCVBO.

Structure of RETVAL

Bit No.	Assignment	Possible error causes	Remedies
Bit 0	Reserve	—	—
Bit 1	SUBSTITUTE VALUES USED FROM RECEIVER	See Bits 2 to 7	Check Bits 2 to 7
Bit 2	ERROR bit of USEND set	Basic communication problems detected by internally called SFB 8 "USEND"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 3	ERROR bit of USEND set	Basic communication problems detected by internally called SFB 8 "USEND"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 4	ERROR bit of URCV set	Basic communication problems detected by internally called SFB 9 "URCV"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 5	CRC error detected	See description for Bit 7	See description for Bit 7
Bit 6	Sequence number error detected	See description for Bit 7	See description for Bit 7

Bit No.	Assignment	Possible error causes	Remedies
Bit 7	Timeout detected	Connection configuration is incorrect	Check and reload connection configuration
		Interference in bus connection to partner F-CPU	Check bus connection and ensure that no external interference sources are present
		F-Monitoring time setting for F-CPU and partner F-CPU is too low.	Check assigned F-Monitoring time TIMEOUT at F_SENDBO and F_RCVBO of both F-CPUs. If necessary, set a higher value. Recompile the S7 programs and download them to the F-CPUs.
		STOP or internal CP fault	Switch CPs to RUN mode Check diagnostic buffer of CPs Replace CPs, if necessary
		STOP, partial or full shutdown, or internal fault in F-CPU or partner F-CPU	Switch F-CPUs to RUN mode Perform an F-Startup Check diagnostic buffer of F-CPUs Replace F-CPUs, if necessary
		Communication was canceled with EN_SEND = 0	Enable communication again at associated F_SENDBO with EN_SEND = 1
		S7 connection has changed, the IP address of the CP has changed, for example	Recompile the S7 programs and download them to the F-CPUs
Bits 8 to 15	= "STATUS" error information of internally called SFB 8 "USEND" or SFB 9 "URCV"	See description of the "STATUS" error information in the online help for SFB 8/SFB 9 or in Manual " System Software for S7-300/400 System and Standard Functions (http://support.automation.siemens.com/WW/view/en/1214574) "	—

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.2.2 F_RCVBO: Receiving of 20 data elements of data type F_BOOL in a fail-safe manner from another F-CPU

Function

The F_RCVBO F-Block receives 20 data elements of data type F_BOOL from another F-CPU and makes them available to the RD_BO_xx outputs. The data must be sent from the other F-CPU with the F_SENDBO F-Block.

You must specify the local ID of the S7 connection from the perspective of the F-CPU (from the connection table in *NetPro*) at input ID.

Communication between F-CPUs takes place hidden in the background by means of a special safety protocol. You must define a communication association between an F_RCVBO in one F-CPU and an F_SENDBO in the other F-CPU by assigning an odd number at the R_ID input of the F_SENDBO and F_RCVBO. Associated F_SENDBOs and F_RCVBOs receive the same value for R_ID.

⚠WARNING

Value for the relevant address reference

The value for each address association (input parameter R_ID; data type: DWORD) is user-defined; however, it must be unique from all other safety-related communication connections in the network. 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 F-Block.

You must assign the desired F-monitoring time at input TIMEOUT. The TIMEOUT input cannot be interconnected.

⚠WARNING

Measure and transfer signal level

It can be ensured (from a fail-safe standpoint) that a signal level to be transferred will be detected on the sender side and transferred to the receiver only if the signal is pending for at least as long as the assigned F-Monitoring time (TIMEOUT).

For information about calculating F-Monitoring times, refer to the section entitled " Run times, F-Monitoring times, and response times (Page 410) ".

The operating mode of the F-CPU with the F_SENDBO is provided at output SENDMODE. If the F-CPU with the F_SENDBO is in deactivated safety mode, output SENDMODE = 1.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ID	WORD	ADDRESS PARAMETER ID	W#16#0
	R_ID	DWORD	ADDRESS PARAMETER R_ID	DW#16#0
	CRC_IMP	DWORD	ADDRESS RELATION CRC	DW#16#0 Automatically initialized *
	TIMEOUT	F_TIME	F MONITORING TIME	T#0ms
	ACK_REI	F_BOOL	ACKNOWLEDGMENT REINTEGRATION	0
	SUBBO_00	F_BOOL	SUBSTITUTE VALUE FOR RECEIVE DATA 00	0
	…		…	
	SUBBO_19	F_BOOL	SUBSTITUTE VALUE FOR RECEIVE DATA 19	0
Outputs:	ACK_REQ	BOOL	ACKNOWLEDGEMENT REQUEST	0
	ERROR	F_BOOL	COMMUNICATION ERROR	0
	SUBS_ON	F_BOOL	SUBSTITUTE VALUES USED	0
	RD_BO_00	F_BOOL	RECEIVE DATA 00	0
	…		…	
	RD_BO_19	F_BOOL	RECEIVE DATA 19	0
	SENDMODE	F_BOOL	1 = SAFETY MODE OF F-CPU WITH F_SENDBO DEACTIVATED	0
	RETVAL	WORD	ERROR CODE	W#16#0

*) Input CRC_IMP is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input CRC_IMP is indicated as changed if changes have been made to the connection configuration in *NetPro*.

Fail-safe values

The fail-safe values pending at the SUBBO_xx inputs are output in the following cases:

- A communication error (e.g., CRC error, timeout) has been detected.
- Communication has been canceled at the associated F_SENDBO with EN_SEND = 0.
- An F-Startup is pending.

The SUBS_ON output is set to 1.

If the output of the fail-safe value is caused by a communication error, output ERROR = 1 is set additionally.

A "Timeout" communication error is not detected unless communication between the F_SENDBO and F_RCVBO connection partners has already be established once. If communication cannot be established after startup of the sending and receiving F-Systems, check the configuration of the safety-related CPU-CPU communication, F_SENDBO and F_RCVBO parameter assignment, and the bus connection. You can also find possible causes of error by evaluating the RETVAL outputs of the F_SENDBO and F_RCVBO. In general, you should always evaluate RETVAL of the F_SENDBO and F_RCVBO because it is possible that only one of the two outputs contains error information.

Reintegration

After a communication error, the data pending at the SD_BO_xx inputs of the associated F_SENDBO are only output again to the RD_BO_xx outputs if the communication error is no longer detected and the error has been acknowledged with a positive edge at the ACK_REI input.

Output ACK_REQ = 1 signals that a user acknowledgment is necessary at input ACK_REI to acknowledge the communication error.

⚠ WARNING
A user acknowledgement is always required for communication errors. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

Startup characteristics

After the sending and receiving F-Systems are started up, communication must be established initially between the F_SENDBO and F_RCVBO connection partners. The fail-safe values pending at the SUBBO_xx inputs are output during this time period. The SUBS_ON output is set to 1.

The SENDMODE output has default setting "0" and is not updated as long as output SUBS_ON = 1.

Output RETVAL

Bit No.	Assignment	Possible error causes	Remedies
Bit 0	Reserve	—	—
Bit 1	SUBSTITUTE VALUES USED FROM RECEIVER	See Bits 2 to 7	Check Bits 2 to 7
Bit 2	ERROR bit of USEND set	Basic communication problems detected by internally called SFB 8 "USEND"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 3	ERROR bit of USEND set	Basic communication problems detected by internally called SFB 8 "USEND"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 4	ERROR bit of URCV set	Basic communication problems detected by internally called SFB 9 "URCV"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 5	CRC error detected	See description for Bit 7	See description for Bit 7
Bit 6	Sequence number error detected	See description for Bit 7	See description for Bit 7
Bit 7	Timeout detected	Connection configuration is incorrect	Check and reload connection configuration
		Interference in bus connection to partner F-CPU	Check bus connection and ensure that no external interference sources are present
		F-Monitoring time setting for F-CPU and partner F-CPU is too low.	Check assigned F-Monitoring time TIMEOUT at F_SENDBO and F_RCVBO of both F-CPUs. If necessary, set a higher value. Recompile the S7 programs and download them to the F-CPUs.
		STOP or internal CP fault	Switch CPs to RUN mode Check diagnostic buffer of CPs Replace CPs, if necessary
		STOP, partial or full shutdown, or internal fault in F-CPU or partner F-CPU	Switch F-CPUs to RUN mode Perform an F-Startup Check diagnostic buffer of F-CPUs Replace F-CPUs, if necessary
		Communication was canceled with EN_SEND = 0	Enable communication again at associated F_SENDBO with EN_SEND = 1
		S7 connection has changed, the IP address of the CP has changed, for example	Recompile the S7 programs and download them to the F-CPUs
Bits 8 to 15	= "STATUS" error information of internally called SFB 8 "USEND" or SFB 9 "URCV"	See description of the "STATUS" error information in the online help for SFB 8/SFB 9 or in Manual " System Software for S7-300/400 System and Standard Functions (http://support.automation.siemens.com/WW/view/en/1214574) "	—

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.2.3 F_SENDR: Sending of 20 data elements of data type F_REAL in a fail-safe manner to another F-CPU

Function

The F_SENDR F-Block sends the data of data type F_REAL pending at the SD_R_xx inputs in a fail-safe manner to another F-CPU. The data must be received there using the F_RCVR F-Block.

To reduce the bus load, you can temporarily shut down communication between the F-CPUs. To do so, supply input EN_SEND with "0" (default = "1"). Send data are then no longer sent to the associated F_RCVR and the assigned fail-safe values are made available to F_SENDR during this time period. If communication was already established between the connection partners, a communication error is detected.

You must specify the local ID of the S7 connection from the perspective of the F-CPU (from the connection table in *NetPro*) at input ID.

Communication between F-CPUs takes place hidden in the background by means of a special safety protocol. You must define a communication association between an F_SENDR in one F-CPU and an F_RCVR in the other F-CPU by assigning an odd number at the R_ID input of the F_SENDR and F_RCVR. Associated F_SENDR and F_RCVR receive the same value for R_ID.

⚠ WARNING

Value for the relevant address reference

The value for each address association (input parameter R_ID; data type: DWORD) is user-defined; however, it must be unique from all other safety-related communication connections in the network. 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 F-Block.

You must assign the desired F-monitoring time at input TIMEOUT. The TIMEOUT input cannot be interconnected.

⚠ WARNING

Measure and transfer signal level

It can be ensured (from a fail-safe standpoint) that a signal level to be transferred will be detected on the sender side and transferred to the receiver only if the signal is pending for at least as long as the assigned F-Monitoring time (TIMEOUT).

For information about calculating F-Monitoring times, refer to chapter " Run times, F-Monitoring times, and response times (Page 410) ".

Note

If the data are received with the F_RCVR F-Block of the *Failsafe Blocks* F-Library (V1_2) or (V1_1), you must assign input EN_SMODE with "0" (default = "1"). Otherwise, a CRC error will be detected by F_RCVR.

In all other cases, you must accept the default setting for input EN_SMODE so that the operating mode of the F-CPU with the F_SENDR can be evaluated at the SENDMODE output of F_RCVR.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	EN_SEND	BOOL	1 = ENABLE SEND	1
	ID	WORD	ADDRESS PARAMETER ID	W#16#0
	R_ID	DWORD	ADDRESS PARAMETER R_ID	DW#16#0
	SD_R_00	F_REAL	SEND DATA 00	0
	…		…	
	SD_R_19	F_REAL	SEND DATA 19	0
	CRC_IMP	DWORD	ADDRESS RELATION CRC	DW#16#0 Automatically initialized *
	TIMEOUT	F_TIME	F MONITORING TIME	T#0ms
	EN_SMODE	F_BOOL	1 = ENABLE SENDMODE	1
Outputs:	ERROR	F_BOOL	1 = COMMUNICATION ERROR	0
	SUBS_ON	F_BOOL	1 = SUBSTITUTE VALUES USED FROM RECEIVER	0
	RETVAL	WORD	ERROR CODE	W#16#0

*) Input CRC_IMP is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input CRC_IMP is indicated as changed if changes have been made to the connection configuration in *NetPro*.

Fail-safe value

Fail-safe values are output from the receiver F_RCVR in the following cases:

- A communication error (e.g., CRC error, timeout) has been detected.
- Communication has been canceled with EN_SEND = 0.
- An F-Startup is pending.

The SUBS_ON output is set to 1.

If the output of the fail-safe value is caused by a communication error, output ERROR = 1 is set additionally.

A "Timeout" communication error is not detected unless communication between the F_SENDR and F_RCVR connection partners has already be established once. If communication cannot be established after startup of the sending and receiving F-Systems, check the configuration of the safety-related CPU-CPU communication, F_SENDR and F_RCVR parameter assignment, and the bus connection. You can also find possible causes of error by evaluating the RETVAL outputs of the F_SENDR and F_RCVR. In general, you should always evaluate RETVAL of the F_SENDR and F_RCVR because it is possible that only one of the two outputs contains error information.

Reintegration

After a communication error, the data from the receiver pending at the SD_R_xx inputs are only output again if the communication error is no longer detected and the error has been acknowledged with a positive edge at the ACK_REI input of F_RCVR.

Startup characteristics

After the sending and receiving F-Systems are started up, communication must be established initially between the F_SENDR and F_RCVR connection partners. The receiver F_RCVR makes fail-safe values available during this time period. The SUBS_ON output is set to 1.

Output RETVAL

The RETVAL output provides non-fail-safe information on the type of communication errors that occurred for service purposes. You can read out this information on your ES/OS or, if necessary, the information can be evaluated in your standard user program. The DIAG bits are saved until acknowledgment at input ACK_REI of the associated F_RCVR.

Structure of RETVAL

Bit No.	Assignment	Possible error causes	Remedies
Bit 0	Reserve	—	—
Bit 1	SUBSTITUTE VALUES USED FROM RECEIVER	See Bits 2 to 7	Check Bits 2 to 7
Bit 2	ERROR bit of USEND set	Basic communication problems detected by internally called SFB 8 "USEND"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 3	ERROR bit of USEND set	Basic communication problems detected by internally called SFB 8 "USEND"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 4	ERROR bit of URCV set	Basic communication problems detected by internally called SFB 9 "URCV"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 5	CRC error detected	See description for Bit 7	See description for Bit 7
Bit 6	Sequence number error detected	See description for Bit 7	See description for Bit 7
Bit 7	Timeout detected	Connection configuration is incorrect	Check and reload connection configuration
		Interference in bus connection to partner F-CPU	Check bus connection and ensure that no external interference sources are present
		F-Monitoring time setting for F-CPU and partner F-CPU is too low.	Check assigned F-Monitoring time TIMEOUT at F_SENDR and F_RCVR of both F-CPUs. If necessary, set a higher value. Recompile the S7 programs and download them to the F-CPUs
		STOP or internal CP fault	Switch CPs to RUN mode Check diagnostic buffer of CPs Replace CPs, if necessary
		STOP, partial or full shutdown, or internal fault in F-CPU or partner F-CPU	Switch F-CPUs to RUN mode Perform an F-Startup Check diagnostic buffer of F-CPUs Replace F-CPUs, if necessary
		Communication was canceled with EN_SEND = 0	Enable communication again at the associated F_SENDR with EN_SEND = 1
		S7 connection has changed, the IP address of the CP has changed, for example	Recompile the S7 programs and download them to the F-CPUs
Bits 8 to 15	= "STATUS" error information of internally called SFB 8 "USEND" or SFB 9 "URCV"	See description of the "STATUS" error information in the online help for SFB 8/SFB 9 or in Manual " System Software for S7-300/400 System and Standard Functions (http://support.automation.siemens.com/WW/view/en/1214574) "	—

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.2.4 F_RCVR: Receiving of 20 data elements of data type F_REAL in a fail-safe manner from another F-CPU

Function

The F_RCVR F-Block receives 20 data elements of data type F_REAL from another F-CPU and makes them available to the RD_R_xx outputs. The data must be sent from the other F-CPU with the F_SENDR F-Block.

You must specify the local ID of the S7 connection from the perspective of the F-CPU (from the connection table in *NetPro*) at input ID.

Communication between F-CPUs takes place hidden in the background by means of a special safety protocol. You must define a communication association between an F_SENDR in one F-CPU and an F_RCVR in the other F-CPU by assigning an odd number at the R_ID input of the F_SENDR and F_RCVR. Associated F_SENDR and F_RCVR receive the same value for R_ID.

⚠ WARNING

Value for the relevant address reference

The value for each address association (input parameter R_ID; data type: DWORD) is user-defined; however, it must be unique from all other safety-related communication connections in the network. 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 F-Block.

You must assign the desired F-monitoring time at input TIMEOUT. The TIMEOUT input cannot be interconnected.

⚠ WARNING

Measure and transfer signal level

It can be ensured (from a fail-safe standpoint) that a signal level to be transferred will be detected on the sender side and transferred to the receiver only if the signal is pending for at least as long as the assigned F-Monitoring time (TIMEOUT).

For information about calculating F-Monitoring times, refer to chapter " Run times, F-Monitoring times, and response times (Page 410) ".

The operating mode of the F-CPU with the F_SENDR is provided at output SENDMODE. If the F-CPU with the F_SENDR is in deactivated safety mode, output ENABLE SENDMODE = 1.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ID	WORD	ADDRESS PARAMETER ID	W#16#0
	R_ID	DWORD	ADDRESS PARAMETER R_ID	W#16#0
	CRC_IMP	DWORD	ADDRESS RELATION CRC	W#16#0 Automatically initialized *
	TIMEOUT	F_TIME	F MONITORING TIME	T#0ms
	ACK_REI	F_BOOL	ACKNOWLEDGMENT REINTEGRATION	0
	SUBR_00	F_REAL	SUBSTITUTE VALUE FOR RECEIVE DATA 00	0
	…		…	
	SUBR_19	F_REAL	SUBSTITUTE VALUE FOR RECEIVE DATA 19	0
Outputs:	ACK_REQ	BOOL	ACKNOWLEDGEMENT REQUEST	0
	ERROR	F_BOOL	1 = COMMUNICATION ERROR	0
	SUBS_ON	F_BOOL	1 = SUBSTITUTE VALUES USED	0
	RD_R_00	F_REAL	RECEIVE DATA 00	0
	…		…	
	RD_R_19	F_REAL	RECEIVE DATA 19	0
	SENDMODE	F_BOOL	1 = F-CPU with F_SENDR in deactivated safety mode	
	RETVAL	WORD	ERROR CODE	W#16#0

*) Input CRC_IMP is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input CRC_IMP is indicated as changed if changes have been made to the connection configuration in *NetPro*.

Fail-safe values

The fail-safe values pending at the SUBR_xx inputs are output in the following cases:

- A communication error (e.g., CRC error, timeout) has been detected.
- Communication has been turned off at the associated F_SENDR with EN_SEND = 0.
- An F-Startup is pending.

The SUBS_ON output is set to 1.

Output SENDMODE is not updated while output SUBS_ON = 1.

If the output of the fail-safe value is caused by a communication error, output ERROR = 1 is set additionally.

A "Timeout" communication error is not detected unless communication between the F_SENDR and F_RCVR connection partners has already be established once. If communication cannot be established after startup of the sending and receiving F-Systems, check the configuration of the safety-related CPU-CPU communication, F_SENDR and F_RCVR parameter assignment, and the bus connection. You can also find possible causes of error by evaluating the RETVAL outputs of the F_SENDR and F_RCVR. In general, you should always evaluate RETVAL of the F_SENDR and F_RCVR because it is possible that only one of the two outputs contains error information.

Reintegration

After a communication error, the data pending at the SD_R_xx inputs of the associated F_SENDR are only output again to the RD_R_xx outputs if the communication error is no longer detected and the error has been acknowledged with a positive edge at the ACK_REI input.

Output ACK_REQ = 1 signals that a user acknowledgment is necessary at input ACK_REI to acknowledge the communication error.

⚠ WARNING
A user acknowledgement is always required for communication errors. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

Startup characteristics

After the sending and receiving F-Systems are started up, communication must be established initially between the F_SENDR and F_RCVR connection partners. The fail-safe values pending at the SUBR_xx inputs are output during this time period. The SUBS_ON output is set to 1.

The SENDMODE output has default setting "0" and is not updated as long as output SUBS_ON = 1.

Output RETVAL

The RETVAL output provides non-fail-safe information on the type of communication errors that occurred for service purposes. You can read out this information on your ES/OS or, if necessary, the information can be evaluated in your standard user program. DIAG bits are saved until acknowledgment at input ACK_REI.

Structure of RETVAL

Bit No.	Assignment	Possible error causes	Remedies
Bit 0	Reserve	—	—
Bit 1	SUBSTITUTE VALUES USED FROM RECEIVER	See Bits 2 to 7	See Bits 2 to 7
Bit 2	ERROR bit of USEND set	Basic communication problems detected by internally called SFB 8 "USEND"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 3	ERROR bit of USEND set	Basic communication problems detected by internally called SFB 8 "USEND"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 4	ERROR bit of URCV set	Basic communication problems detected by internally called SFB 9 "URCV"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"

Bit No.	Assignment	Possible error causes	Remedies
		See also description for Bit 7	See also description for Bit 7
Bit 5	CRC error detected	See description for Bit 7	See description for Bit 7
Bit 6	Sequence number error detected	See description for Bit 7	See description for Bit 7
Bit 7	Timeout detected	Connection configuration is incorrect	Check and reload connection configuration
		Interference in bus connection to partner F-CPU	Check bus connection and ensure that no external interference sources are present
		F-Monitoring time setting for F-CPU and partner F-CPU is too low.	Check assigned F-Monitoring time TIMEOUT at F_SENDR and F_RCVR of both F-CPUs. If necessary, set a higher value. Recompile the S7 programs and download them to the F-CPUs.
		STOP or internal CP fault	Switch CPs to RUN mode. Check diagnostic buffer of CPs. Replace CPs, if necessary.
		STOP, partial or full shutdown, or internal fault in F-CPU or partner F-CPU	Switch F-CPUs to RUN mode. Perform an F-Startup. Check diagnostic buffer of F-CPUs. Replace F-CPUs, if necessary.
		Communication was canceled with EN_SEND = 0.	Enable communication again at the associated F_SENDR with EN_SEND = 1.
		S7 connection has changed, the IP address of the CP has changed, for example	Recompile the S7 programs and download them to the F-CPUs.
Bits 8 to 15	= "STATUS" error information of internally called SFB 8 "USEND" or SFB 9 "URCV"	See description of the "STATUS" error information in the online help for SFB 8/SFB 9 or in Manual " System Software for S7-300/400 System and Standard Functions (http://support.automation.siemens.com/WW/view/en/1214574) "	—

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.2.5 F_SDS_BO: Sending of 32 data elements of data type F_BOOL in a fail-safe manner to another F-CPU

Function

The F_SDS_BO F-Block sends the data of data type F_BOOL pending at the SD_BO_*xx* inputs in a fail-safe manner to another F-CPU. The data must be received there using the F_RDS_BO F-Block.

Note

The F_SDS_BO F-Block can also send the data of data type F_BOOL pending at the SD_BO_*xx* inputs in a fail-safe manner to another F-CPU with *S7 Distributed Safety*. The data must be received there with the F_RCVS7 F-Block and an F-Communication DB with exactly 32 data elements of data type F_BOOL.

To reduce the bus load, you can temporarily shut down communication between the F-CPUs. To do so, supply input EN_SEND with "0" (default = "1"). Send data are then no longer sent to the associated F_RDS_BO and the assigned fail-safe values are made available to F_RDS_BO during this time period. If communication was already established between the connection partners, a communication error is detected.

You must specify the local ID of the S7 connection from the perspective of the F-CPU (from the connection table in *NetPro*) at input ID.

Communication between F-CPUs takes place hidden in the background by means of a special safety protocol. You must define a communication association between an F_SDS_BO in one F-CPU and an F_RDS_BO in the other F-CPU by assigning an odd number at the R_ID input of the F_SDS_BO and F_RDS_BO. Associated F_SDS_BO and F_RDS_BO receive the same value for R_ID.

⚠ WARNING

Value for the relevant address reference

The value for each address association (input parameter R_ID; data type: DWORD) is user-defined; however, it must be unique from all other safety-related communication connections in the network. 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 F-Block.

You must assign the desired F-monitoring time at input TIMEOUT. The TIMEOUT input cannot be interconnected.

⚠ WARNING

Measure and transfer signal level

It can be ensured (from a fail-safe standpoint) that a signal level to be transferred will be detected on the sender side and transferred to the receiver only if the signal is pending for at least as long as the assigned F-Monitoring time (TIMEOUT).

For information about calculating F-Monitoring times, refer to chapter " Run times, F-Monitoring times, and response times (Page 410) ".

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	EN_SEND	BOOL	1 = ENABLE SEND	1
	ID	WORD	ADDRESS PARAMETER ID	W#16#0
	R_ID	DWORD	ADDRESS PARAMETER R_ID	DW#16#0
	SD_BO_00	F_BOOL	SEND DATA 00	0
	…		…	
	SD_BO_31	F_BOOL	SEND DATA 31	0
	CRC_IMP	DWORD	ADDRESS RELATION CRC	DW#16#0 Automatically initialized *
	TIMEOUT	F_TIME	F MONITORING TIME in ms	T#0ms
Outputs:	ERROR	F_BOOL	1 = COMMUNICATION ERROR	0
	SUBS_ON	F_BOOL	1 = SUBSTITUTE VALUES USED FROM RECEIVER	0
	RETVAL	WORD	ERROR CODE	W#16#0

*) Input CRC_IMP is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input CRC_IMP is indicated as changed if changes have been made to the connection configuration in *NetPro*.

Fail-safe values

Fail-safe values are output from the receiver F_RDS_BO in the following cases:

- A communication error (e.g., CRC error, timeout) has been detected.
- Communication has been canceled with EN_SEND = 0.
- An F-Startup is pending.

The SUBS_ON output is set to 1.

If the output of the fail-safe value is caused by a communication error, output ERROR = 1 is set additionally.

A "Timeout" communication error is not detected unless communication between the F_SDS_BO and F_RDS_BO connection partners has already be established once. If communication cannot be established after startup of the sending and receiving F-Systems, check the configuration of the safety-related CPU-CPU communication, F_SDS_BO and F_RDS_BO parameter assignment, and the bus connection. You can also find possible causes of error by evaluating the RETVAL outputs of the F_SDS_BO and F_RDS_BO. In general, you should always evaluate RETVAL of the F_SDS_BO and F_RDS_BO because it is possible that only one of the two outputs contains error information.

Reintegration

After a communication error, the data from the receiver pending at the SD_BO_*xx* inputs are only output again if the communication error is no longer detected and the error has been acknowledged with a positive edge at the ACK_REI input of F_RDS_BO.

Startup characteristics

After the sending and receiving F-Systems are started up, communication must be established initially between the F_SDS_BO and F_RDS_BO connection partners. The receiver F_RDS_BO makes fail-safe values available during this time period. The SUBS_ON output is set to 1.

Output RETVAL

The RETVAL output provides non-fail-safe information on the type of communication errors that occurred for service purposes. You can read out this information on your ES/OS or, if necessary, the information can be evaluated in your standard user program. The DIAG bits are saved until acknowledgment at input ACK_REI of the associated F_RDS_BO.

Structure of RETVAL

Bit No.	Assignment	Possible error causes	Remedies
Bit 0	Reserve	—	—
Bit 1	SUBSTITUTE VALUES USED FROM RECEIVER	See Bits 2 to 7	Check Bits 2 to 7
Bit 2	ERROR bit of USEND set	Basic communication problems detected by internally called SFB 8 "USEND"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 3	ERROR bit of USEND set	Basic communication problems detected by internally called SFB 8 "USEND"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 4	ERROR bit of URCV set	Basic communication problems detected by internally called SFB 9 "URCV"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 5	CRC error detected	See description for Bit 7	See description for Bit 7
Bit 6	Sequence number error detected	See description for Bit 7	See description for Bit 7

Bit No.	Assignment	Possible error causes	Remedies
Bit 7	Timeout detected	Connection configuration is incorrect	Check and reload connection configuration
		Interference in bus connection to partner F-CPU	Check bus connection and ensure that no external interference sources are present
		F-monitoring time setting for F-CPU and partner F-CPU is too low.	Check assigned F-monitoring time TIMEOUT at F_SDS_BO and F_RDS_BO of both F-CPUs. If necessary, set a higher value. Recompile the S7 programs and load them to the F-CPUs.
		STOP or internal CP fault	Switch CPs to RUN mode. Check diagnostic buffer of CPs. Replace CPs, if necessary.
		STOP, partial or full shutdown, or internal fault in F-CPU or partner F-CPU	Switch F-CPUs to RUN mode. Perform an F-Startup. Check diagnostic buffer of F-CPUs. Replace F-CPUs, if necessary.
		Communication was canceled with EN_SEND = 0.	Enable communication again at the associated F_SDS_BO with EN_SEND = 1.
		S7 connection has changed, the IP address of the CP has changed, for example.	Recompile the S7 programs and download them to the F-CPUs.
Bits 8 to 15	= "STATUS" error information of internally called SFB 8 "USEND" or SFB 9 "URCV"	See description of the "STATUS" error information in the online help for SFB 8/SFB 9 or in Manual " System Software for S7-300/400 System and Standard Functions (http://support.automation.siemens.com/WW/view/en/1214574) "	—

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.2.6 F_RDS_BO: Receiving of 32 data elements of data type F_BOOL in a fail-safe manner from another F-CPU

Function

The F_RDS_BO F-Block receives 32 data elements of data type F_BOOL from another F-CPU and makes them available to the RD_BO_*xx* outputs. The data must be sent from the other F-CPU with the F_SDS_BO F-Block.

Note

The F_RDS_BO F-Block can also receive the 32 data elements of data type F_BOOL in a fail-safe manner from an F-CPU with *S7 Distributed Safety*. The data must be sent there with the F_SENDS7 F-Block and an F-Communication DB with exactly 32 data elements of data type F_BOOL.

You must specify the local ID of the S7 connection from the perspective of the F-CPU (from the connection table in *NetPro*) at input ID.

Communication between F-CPUs takes place hidden in the background by means of a special safety protocol. You must define a communication association between an F_SDS_BO in one F-CPU and an F_RDS_BO in the other F-CPU by assigning an odd number at the R_ID input of the F_SDS_BO and F_RDS_BO. Associated F_SDS_BO and F_RDS_BO receive the same value for R_ID.

⚠ WARNING

Value for the relevant address reference

The value for each address association (input parameter R_ID; data type: DWORD) is user-defined; however, it must be unique from all other safety-related communication connections in the network. 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 F-Block.

You must assign the desired F-monitoring time at input TIMEOUT. The TIMEOUT input cannot be interconnected.

⚠ WARNING

Measure and transfer signal level

It can be ensured (from a fail-safe standpoint) that a signal level to be transferred will be detected on the sender side and transferred to the receiver only if the signal is pending for at least as long as the assigned F-Monitoring time (TIMEOUT).

For more information about calculation of the F-monitoring time, refer to chapter " Run times, F-Monitoring times, and response times (Page 410) ".

The operating mode of the F-CPU with the F_SDS_BO is provided at output SENDMODE. If the F-CPU with the F_SDS_BO is in deactivated safety mode, output ENABLE SENDMODE = 1.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ID	WORD	ADDRESS PARAMETER ID	W#16#0
	R_ID	DWORD	ADDRESS PARAMETER R_ID	DW#16#0
	CRC_IMP	DWORD	ADDRESS RELATION CRC	DW#16#0 Automatically initialized *
	TIMEOUT	F_TIME	F MONITORING TIME in ms	T#0ms
	ACK_REI	F_BOOL	ACKNOWLEDGEMENT REINTEGRATION	0
	SUBBO_00	F_BOOL	SUBSTITUTE VALUE FOR RECEIVE DATA 00	0
	...		...	
	SUBBO_31	F_BOOL	SUBSTITUTE VALUE FOR RECEIVE DATA 31	0
Outputs:	ACK_REQ	BOOL	ACKNOWLEDGEMENT REQUEST	0
	ERROR	F_BOOL	1 = COMMUNICATION ERROR	0
	SUBS_ON	F_BOOL	1 = SUBSTITUTE VALUES USED	0
	RD_BO_00	F_BOOL	RECEIVE DATA 00	0
	...		...	
	RD_BO _31	F_BOOL	RECEIVE DATA 31	0
	SENDMODE	F_BOOL	1 = SAFETY MODE OF F-CPU WITH F_SDS_BO DEACTIVATED	0
	RETVAL	WORD	ERROR CODE	W#16#0

*) Input CRC_IMP is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input CRC_IMP is indicated as changed if changes have been made to the connection configuration in *NetPro*.

Fail-safe values

The fail-safe values pending at the SUBBO_*xx* inputs are output in the following cases:

- A communication error (e.g., CRC error, timeout) has been detected.
- Communication has been canceled at the associated F_SDS_BO with EN_SEND = 0.
- An F-Startup is pending.

The SUBS_ON output is set to 1.

Output SENDMODE is not updated while output SUBS_ON = 1.

If the output of the fail-safe value is caused by a communication error, output ERROR = 1 is set additionally.

A "Timeout" communication error is not detected unless communication between the F_SDS_BO and F_RDS_BO connection partners has already be established once. If communication cannot be established after startup of the sending and receiving F-Systems, check the configuration of the safety-related CPU-CPU communication, F_SDS_BO and F_RDS_BO parameter assignment, and the bus connection. You can also find possible causes of error by evaluating the RETVAL outputs of the F_SDS_BO and F_RDS_BO. In general, you should always evaluate RETVAL of the F_SDS_BO and F_RDS_BO because it is possible that only one of the two outputs contains error information.

Reintegration

After a communication error, the data pending at the SD_BO_*xx* inputs of the associated F_SDS_BO are only output again to the RD_BO_*xx* outputs if the communication error is no longer detected and the error has been acknowledged with a positive edge at the ACK_REI input.

Output ACK_REQ = 1 signals that a user acknowledgment is necessary at input ACK_REI to acknowledge the communication error.

⚠WARNING

A user acknowledgement is always required for communication errors.

For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

Startup characteristics

After the sending and receiving F-Systems are started up, communication must be established initially between the F_SDS_BO and F_RDS_BO connection partners. The fail-safe values pending at the SUBBO_*xx* inputs are output during this time period. The SUBS_ON output is set to 1.

The SENDMODE output has default setting "0" and is not updated as long as output SUBS_ON = 1.

Output RETVAL

The RETVAL output provides non-fail-safe information on the type of communication errors that occurred for service purposes. You can read out this information on your ES/OS or, if necessary, the information can be evaluated in your standard user program. DIAG bits are saved until acknowledgment at input ACK_REI.

Structure of RETVAL

Bit No.	Assignment	Possible error causes	Remedies
Bit 0	Reserve	—	—
Bit 1	SUBSTITUTE VALUES USED FROM RECEIVER	See Bits 2 to 7	Check Bits 2 to 7
Bit 2	ERROR bit of USEND set	Basic communication problems detected by internally called SFB 8 "USEND"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 3	ERROR bit of USEND set	Basic communication problems detected by internally called SFB 8 "USEND"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"
		See also description for Bit 7	See also description for Bit 7
Bit 4	ERROR bit of URCV set	Basic communication problems detected by internally called SFB 9 "URCV"	Bits 8 to 15 = Evaluate "STATUS" of SFB 8 "USEND" or SFB 9 "URCV"

Bit No.	Assignment	Possible error causes	Remedies
		See also description for Bit 7	See also description for Bit 7
Bit 5	CRC error detected	See description for Bit 7	See description for Bit 7
Bit 6	Sequence number error detected	See description for Bit 7	See description for Bit 7
Bit 7	Timeout detected	Connection configuration is incorrect	Check and reload connection configuration
		Interference in bus connection to partner F-CPU	Check bus connection and ensure that no external interference sources are present.
		F-Monitoring time setting for F-CPU and partner F-CPU is too low.	Check assigned F-Monitoring time TIMEOUT at F_SDS_BO and F_RDS_BO of both F-CPUs. If necessary, set a higher value. Recompile the S7 programs and load them to the F-CPUs.
		STOP or internal CP fault	Switch CPs to RUN mode. Check diagnostic buffer of CPs. Replace CPs, if necessary.
		STOP, partial or full shutdown, or internal fault in F-CPU or partner F-CPU	Switch F-CPUs to RUN mode. Perform an F-Startup. Check diagnostic buffer of F-CPUs. Replace F-CPUs, if necessary.
		Communication was canceled with EN_SEND = 0.	Enable communication again at the associated F_SDS_BO with EN_SEND = 1.
		S7 connection has changed, the IP address of the CP has changed, for example.	Recompile the S7 programs and download them to the F-CPUs
Bits 8 to 15	= "STATUS" error information of internally called SFB 8 "USEND" or SFB 9 "URCV"	See description of the "STATUS" error information in the online help for SFB 8/SFB 9 or in Manual " System Software for S7-300/400 System and Standard Functions (http://support.automation.siemens.com/WW/view/en/1214574) "	—

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.3 F-Blocks for comparing two input values of the same type

Overview

Block name	Block number	Description
F_CMP_R	FB 313	Comparator for two REAL values
F_LIM_HL	FB 314	Monitoring of upper limit violation of a REAL value
F_LIM_LL	FB 315	Monitoring of lower limit violation of a REAL value

A.2.3.1 F_CMP_R Comparator for two REAL values

Function

This F-Block compares two inputs of data type F_REAL and sets outputs GT, GE, EQ, LT or LE to "1", whatever the comparator result:

- GT = 1 if IN1 > IN2
- GE = 1 if IN1 ≥ IN2
- EQ = 1 if IN1 = IN2
- LT = 1 if IN1 < IN2
- LE = 1 if IN1 ≤ IN2

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_REAL	Input 1	0
	IN2	F_REAL	Input 2	0
Outputs:	GT	F_BOOL	IN1 > IN2	0
	GE	F_BOOL	IN1 ≥ IN2	0
	EQ	F_BOOL	IN1 = IN2	0
	LT	F_BOOL	IN1 < IN2	0
	LE	F_BOOL	IN1 ≤ IN2	0

Error handling

- If one of the inputs IN1 or IN2 is an invalid floating point number (NaN), outputs GT and LT are set to 1.
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.3.2 F_LIM_HL: Monitoring of upper limit violation of a REAL value

Function

This F-Block monitors the input variable U for limit violation (U_HL). A hysteresis can also be specified at the HYS input to avoid fluttering of the QH output in the event of fluctuations in the input value.

- U ≥ U_HL: If the upper limit is exceeded, output QH = 1.
- (U_HL – HYS) ≤ U < U_HL: QH remains unchanged in this range.
- U < (U_HL – HYS): If the limit value hysteresis is fallen below, output QH = 0.

The QHN output corresponds to the negated QH output.

The limit value and hysteresis are also available as non-fail-safe data at the U_HL_O and HYS_O outputs for further processing in the standard user program.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	U	F_REAL	INPUT	0.0
	U_HL	F_REAL	UPPER LIMIT	100.0
	HYS	F_REAL	HYSTERESIS	0.0
	SUBS_IN	F_BOOL	SUBSTITUTE VALUE	0
Outputs:	QH	F_BOOL	1 = UPPER LIMIT VIOLATION	0
	QHN	F_BOOL	NEGATING OUTPUT QH	1
	U_HL_O	REAL	UPPER LIMIT	100.0
	HYS_O	REAL	HYSTERESIS	0.0

Error handling

- If one of the inputs U, U_HL or HYS is an invalid floating point number (NaN) or if invalid floating-point numbers (NaN) arise due to calculations in the F-Block, the fail-safe value at the input SUBS_IN is output at output QH.

  If invalid floating-point numbers (NaNs) arise due to calculations in the F-Block, the following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

  – "Safety program: invalid REAL number in DB" (Event ID 16#75D9)
- An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

  – "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.3.3 F_LIM_LL: Monitoring of lower limit violation of a REAL value

Function

This F-Block monitors the input variable U for lower limit violation

(U_LL). A hysteresis can also be specified at the HYS input to avoid fluttering of the QL output in the event of fluctuations in the input value.

- U ≤ U_LL: If the lower limit is violated, output QL = 1.
- U_LL < U ≤ (U_LL + HYS): QL remains unchanged in this range.
- U > (U_LL + HYS): If the upper limit is exceeded violated + hysteresis, output QL = 0.

Output QLN corresponds to the negated QL output.

The limit value and hysteresis are also available as non-fail-safe data at the U_LL_O and HYS_O outputs for evaluation in the standard user program.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	U	F_REAL	INPUT	0.0
	U_LL	F_REAL	LOWER LIMIT	100.0
	HYS	F_REAL	HYSTERESIS	0.0
	SUBS_IN	F_BOOL	FAIL-SAFE VALUE	0
Outputs:	QL	F_BOOL	1 = LOWER LIMIT VIOLATION	0
	QLN	F_BOOL	NEGATING OUTPUT QL	1
	U_LL_O	REAL	LOWER LIMIT	100.0
	HYS_O	REAL	HYSTERESIS	0.0

Error handling

- If one of the inputs U, U_LL or HYS is an invalid floating point number (NaN) or if invalid floating-point numbers (NaNs) arise due to calculations in the F-Block, the fail-safe value at the input SUBS_IN is output at output QL.

  If invalid floating-point numbers (NaNs) arise due to calculations in the F-Block, the following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

  – "Safety program: invalid REAL number in DB" (Event ID 16#75D9)
- An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

  – "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.4 Voter blocks for inputs of data type REAL and BOOL

Overview

Block name	Block number	Description
F_2oo3DI	FB 316	2oo3 evaluation of inputs of data type BOOL with discrepancy analysis
F_2oo3AI	FB 317	2oo3 evaluation of inputs of data type REAL with discrepancy analysis
F_1oo2AI	FB 318	1oo2 evaluation of inputs of data type REAL with discrepancy analysis

A.2.4.1 F_2oo3DI: 2oo3 evaluation of inputs of data type BOOL with discrepancy analysis

Function

This block monitors three binary inputs for signal state 1. The OUT output is 1 when at least two inputs INx are 1. Otherwise the output is OUT 0. The OUTN output corresponds to the negated OUT output.

If the input DIS_ON = 1 is set, a discrepancy analysis is carried out. If the discrepancy between an input INx and the two other inputs INy is longer than the assigned discrepancy time DIS_TIME, a discrepancy error is detected and stored at the outputs DIS and DIS_D with 1.

If the discrepancy 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 acknowledge the discrepancy error with a rising edge at input ACK.

The ACK_REQ = 1 output signals that a user acknowledgment is necessary at input ACK to acknowledge the discrepancy error.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_BOOL	Input 1	0
	IN2	F_BOOL	Input 2	0
	IN3	F_BOOL	Input 3	0
	DIS_ON	F_BOOL	1 = Discrepancy analysis	0
	DIS_TIME	F_TIME	Discrepancy time in ms	1000
	ACK_NEC	F_BOOL	1 = Acknowledgment necessary	0
	ACK	F_BOOL	Acknowledgment	0
Outputs:	OUT	F_BOOL	Output	0
	OUTN	F_BOOL	NEGATING OUTPUT	1
	DIS	F_BOOL	DISCREPANCY ERROR	0
	DIS_D	BOOL	DISCREPANCY ERROR DATA	0
	ACK_REQ	BOOL	ACKNOWLEDGMENT REQUEST	0

Fail-safe user times

⚠ WARNING

When using an F-Block with time processing, take the following timing imprecision sources into account when determining your response times:

- Known timing imprecision (based on standard systems) resulting from cyclic processing
- Tolerance of internal time monitoring in the F-CPU
  - For time values between 10 ms and 50 s: 5 ms
  - For time values from > n × 50 s to (n+1) × 50 s: ± (n+1) × 5 ms

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.4.2 F_2oo3AI: 2oo3 evaluation of inputs of the REAL data type with discrepancy analysis

Function

This F-Block carries out a 2oo3 evaluation of REAL values using discrepancy analysis. If a REAL value is invalid, a 1oo2 evaluation is carried out. It calculates the mean value and the median or maximum and the minimum of inputs INx depending on inputs QBADx:

- If all INx inputs are valid (QBAD1, QBAD2 and QBAD3 = 0) and if no discrepancy error was stored (DIS1CH = 0, DISALL = 0), the mean [(IN1+IN2+IN3)/3] is made available at output OUT_AVG and the median of IN1, IN2 and IN3 is made available at outputs MED_MAX and MED_MIN.
- If all INx inputs are valid (QBAD1, QBAD2 and QBAD3 = 0) and if no discrepancy error was stored (DIS1CH = 1, DISALL = 0), the mean value of the valid, discrepancy-free INx inputs is made available at output OUT_AVG and the median of IN1, IN2 and IN3 is made available at outputs MED_MAX and MED_MIN.
- If only two INx inputs are valid (QBADx = 0 and QBADy = 1), the valid inputs INx mean is set at output OUT_AVG, the maximum at output MED_MAX and the minimum at output MED_MIN and QBAD_1CH = 1 set.
- If only one INx input is valid (QBADx = 0 and QBADy = 1), INx is made available at outputs OUT_AVG, MED_MAX and MED_MIN and QBAD_2CH = 1 is set.
- If no input INx is valid (QBAD1, QBAD2 and QBAD3 = 1), the SUBS_V the fail-safe value is made available at outputs OUT_AVG, MED_MAX and MED_MIN and QBAD_ALL = 1 is set.

A discrepancy analysis is carried out as follows:

- All inputs INx are valid (QBAD1, QBAD2 and QBAD3 = 0):
  - If the discrepancy between an input INx and the two other inputs INy is greater than the assigned DELTA tolerance and longer than the assigned discrepancy time DIS_TIME, a discrepancy error is detected and stored at the outputs DIS1CH and DIS1CH_D with 1.
  - If the INx inputs discrepancy is greater than the assigned DELTA tolerance and longer than the assigned discrepancy time DIS_TIME, a discrepancy error is detected and stored at the outputs DIS and DIS_D with 1.
- Two INx inputs are valid (QBADx = 0 and QBADy = 1):
  - If the discrepancy between two valid INx inputs is greater than the assigned DELTA tolerance and longer than the assigned discrepancy time DIS_TIME, a discrepancy error is detected and stored at the outputs DISALL and DISALL_D with 1.
- Only one INx input is valid (QBADx = 0 and QBADy = 1) or all INx inputs are invalid (QBAD1, QBAD2 and QBAD3 = 1):
  - No discrepancy analysis is carried out.

Instead, the absolute value is always used for the DELTA and DIS_TIME inputs.

When the assigned tolerance is adhered to again, the discrepancy error is acknowledged according to the ACK_NEC parameter assignment:

- If ACK_NEC = 0, the acknowledgment is automatic.
- If ACK_NEC = 1 you must acknowledge the discrepancy error with a rising edge at input ACK.

The ACK_REQ = 1 output signals that a user acknowledgment is necessary at input ACK to acknowledge the discrepancy error.

Note

If you want to implement safety function triggering when an upper limit is exceeded (e.g. with F-Block F_LIM_HL), you must use output MED_MAX for limit violation monitoring. If you want to implement safety function triggering when a lower limit is violated (e.g. with F-Block F_LIM_LL), you must use output MED_MIN for limit violation monitoring.

Output OUT_AVG may only be used in evaluations in which - depending on the process situation - the maximum and the minimum each represent the safe direction once. In this case, output DISALL = 1 should also trigger the safety function.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	DELTA	F_REAL	TOLERANCE BETWEEN INx	0.0
	DIS_TIME	F_TIME	DISCREPANCY TIME IN MS	1000
	IN1	F_REAL	Input 1	0.0
	IN2	F_REAL	Input 2	0.0
	IN3	F_REAL	Input 3	0.0
	QBAD1	F_BOOL	1 = INPUT IN1 INVALID	0
	QBAD2	F_BOOL	1 = INPUT IN2 INVALID	0
	QBAD3	F_BOOL	1 = INPUT IN3 INVALID	0
	SUBS_V	F_REAL	SUBSTITUTE VALUE	0.0
	ACK_NEC	F_BOOL	1 = ACKNOWLEDGMENT NECESSARY	0
	ACK	F_BOOL	ACKNOWLEDGMENT	0
Outputs:	OUT_AVG	F_REAL	AVERAGE VALUE OF INx	0.0
	MED_MAX	F_REAL	MEDIAN/MAXIMUM VALUE OF INx	0.0
	MED_MIN	F_REAL	MEDIAN/MAXIMUM VALUE OF INx	0.0
	QBAD_1CH	F_BOOL	ONE INPUT INx INVALID	0
	QBAD_2CH	F_BOOL	TWO INPUTS INx INVALID	0
	QBAD_ALL	F_BOOL	ALL INPUTS INx INVALID	0
	DIS1CH	F_BOOL	ONE INPUT INx DISCREPANCY ERROR	0
	DISALL	F_BOOL	ALL INPUT INx DISCREPANCY ERROR	0
	DIS1CH_D	BOOL	ONE INPUT INx DISCREPANCY ERROR DATA	0
	DISALL_D	BOOL	ALL INPUT INx DISCREPANCY ERROR DATA	0
	ACK_REQ	BOOL	ACKNOWLEDGMENT REQUEST	0

Used together with F-Channel driver F_CH_AI

If you interconnect input INx of the F_2oo3AI with output V of an F_CH_AI, you must observe the following:

- Interconnect input QBADx of the F_2oo3AI with the QBAD output of the F_CH_AI and its output V with input INx of the F_2oo3AI.

Fail-safe user times

⚠WARNING

When using an F-Block with time processing, take the following timing imprecision sources into account when determining your response times:

- Known timing imprecision (based on standard systems) resulting from cyclic processing
- Tolerance of internal time monitoring in the F-CPU
  - For time values between 10 ms and 50 s: 5 ms
  - For time values from > n × 50 s to (n+1) × 50 s: ± (n+1) × 5 ms

Error handling

- If an input INx is an invalid floating point number (NaN), it is treated as an invalid input INx with QBAD = 1.
- If the DELTA input is an invalid floating point number (NaN), DIS1CH, DISALL, DIS1CH_D and DISALL_D are set to 1.
- If invalid floating-point numbers (NaN) arise due to calculations in the F-Block, the fail-safe value SUBS_V is made available at the outputs OUT_AVG, MED_MAX and MED_MIN, QBAD_1CH, QBAD_2CH and QBAD_ALL = 1 are set and the following diagnostic event is entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Invalid REAL number in DB" (Event ID 16#75D9).
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

A.2.4.3 F_1oo2AI: 1oo2 evaluation of inputs of data type REAL with discrepancy analysis

Function

This F-Block carries out a 1oo2 evaluation of REAL values using discrepancy analysis. It calculates the mean value, the maximum and the minimum of inputs IN1 and IN2 depending on inputs QBADx:

- If both INx inputs are valid (QBAD1 and QBAD2 = 0), the IN1 and IN2 mean [(IN1+IN2)/2] is made available at output OUT_AVG, the maximum at output OUT_MAX and the minimum at output OUT_MIN.
- If only input INx is valid (QBADx = 0 and QBADy = 1), INx is made available at outputs OUT_AVG, OUT_MAX and OUT_MIN and QBAD_1CH = 1 is set.
- If no input INx is valid (QBAD1 and QBAD2 = 1), the SUBS_V the fail-safe value is made available at outputs OUT_AVG, OUT_MAX and OUT_MIN and QBAD_ALL = 1 is set.

If both inputs INx are valid (QBAD1 and QBAD2 = 0), a discrepancy analysis is carried out:

If the INx inputs discrepancy is greater than the assigned DELTA tolerance and longer than the assigned discrepancy time DIS_TIME, a discrepancy error is detected and stored at the outputs DIS and DIS_D with 1. The absolute value is always used for the DELTA and DIS_TIME inputs.

When the assigned tolerance is adhered to again, the discrepancy error is acknowledged according to the ACK_NEC parameter assignment:

- If ACK_NEC = 0, the acknowledgment is automatic.
- If ACK_NEC = 1 you must acknowledge the discrepancy error with a rising edge at input ACK.

The ACK_REQ = 1 output signals that a user acknowledgment is necessary at input ACK to acknowledge the discrepancy error.

Note

If you want to implement safety function triggering when an upper limit is exceeded (e.g. with F-Block F_LIM_HL), you must use output OUT_MAX for limit violation monitoring. If you want to implement safety function triggering when a lower limit is violated (e.g. with F-Block F_LIM_LL), you must use output OUT_MIN for limit violation monitoring.

Output OUT_AVG may only be used in evaluations in which - depending on the process situation - the maximum and the minimum each represent the safe direction once. In this case, output DIS = 1 should also trigger the safety function.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	DELTA	F_REAL	Tolerance between INx	0.0
	DIS_TIME	F_TIME	Discrepancy time in ms	0
	IN1	F_REAL	Input 1	0.0
	IN2	F_REAL	Input 2	0.0
	QBAD1	F_BOOL	1 = Input IN1 invalid	0
	QBAD2	F_BOOL	1 = Input IN2 invalid	0
	SUBS_V	F_REAL	SUBSTITUTE VALUE	0.0
	ACK_NEC	F_BOOL	1 = ACKNOWLEDGMENT NECESSARY	0
	ACK	F_BOOL	ACKNOWLEDGMENT	0
Outputs:	OUT_AVG	F_REAL	AVERAGE VALUE OF INx	0.0
	OUT_MAX	F_REAL	MAXIMUM VALUE OF INx	0.0
	OUT_MIN	F_REAL	MINIMUM VALUE OF INx	0.0
	QBAD_1CH	F_BOOL	ONE INPUT INx INVALID	0
	QBAD_ALL	F_BOOL	ALL INPUTS INx INVALID	0
	DIS	F_BOOL	DISCREPANCY ERROR	0
	DIS_D	BOOL	DISCREPANCY ERROR DATA	0
	ACK_REQ	BOOL	ACKNOWLEDGMENT REQUEST	0

Used together with F-Channel driver F_CH_AI

If you interconnect input INx of the F_1oo2AI with output V of an F_CH_AI, you must observe the following:

- Interconnect input QBADx of the F_1oo2AI with the QBAD output of the F_CH_AI and its output V with input INx of the F_1oo2AI.

Fail-safe user times

⚠WARNING

When using an F-Block with time processing, take the following timing imprecision sources into account when determining your response times:

- Known timing imprecision (based on standard systems) resulting from cyclic processing
- Tolerance of internal time monitoring in the F-CPU
  - For time values between 10 ms and 50 s: 5 ms
  - For time values from > n × 50 s to (n+1) × 50 s: ± (n+1) × 5 ms

Error handling

- If an input INx is an invalid floating point number (NaN), it is treated as an invalid input INx with QBADx = 1.
- If the DELTA input is an invalid floating point number (NaN), DIS and DIS_D are set to 1.
- If invalid floating-point numbers (NaNs) arise due to calculations in the F-Block, the fail-safe value SUBS_V is made available at the outputs OUT_AVG, OUT_MAX and OUT_MIN, QBAD_1CH and QBAD_ALL = 1 are set and the following diagnostic event is entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Invalid REAL number in DB" (Event ID 16#75D9).
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

A.2.5 Blocks and F-Blocks for data conversion

Overview

F-Blocks

Block name	Block number	Description
F_SWC_P	FB 335	Centralized control of operator input via the OS (Maintenance Override)
F_SWC_BO	FB 336	Processing of a parameter of data type F_BOOL for operator input via the OS (Maintenance Override)
F_SWC_R	FB 337	Processing of a parameter of data type F_REAL for operator input via the OS (Maintenance Override)
F_FR_FDI	FB 339	Conversion from F_REAL to F_DINT
F_FDI_FR	FB 340	Conversion from F_DINT to F_REAL
F_BO_FBO	FB 361	Conversion from BOOL to F_BOOL
F_R_FR	FB 362	Conversion from REAL to F_REAL
F_QUITES	FB 367	Fail-safe acknowledgement via the ES/OS
F_TI_FTI	FB 368	Conversion from TIME to F_TIME
F_I_FI	FB 369	Conversion from INT to F_INT
F_FI_FR	FB 460	Conversion from F_INT to F_REAL
F_FR_FI	FB 461	Conversion from F_REAL to F_INT
F_CHG_R	FB 478	Safety Data Write for F_REAL
F_CHG_BO	FB 479	Safety Data Write for F_BOOL

Blocks

Block name	Block number	Description
F_FBO_BO	FC 303	Conversion from F_BOOL to BOOL
F_FR_R	FC 304	Conversion from F_REAL to REAL
F_FI_I	FC 305	Conversion from F_INT to INT
F_FTI_TI	FC 306	Conversion from F_TIME to TIME
SWC_MOS	FB 338	Command function for Maintenance Override

Validity check

⚠WARNING

Validity check

The F-Blocks F_BO_FBO, F_I_FI, F_TI_FTI, and F_R_FR only perform a data conversion. This means you must program additional measures for validity checks in the safety program.

The simplest type of validity check is a range definition with a fixed upper and lower limit, such as F_LIM_R.

However, not all input parameters can be checked for validity in a sufficiently simple way.

A.2.5.1 F_SWC_P: Centralized control of operator input via the OS

Function

This F-Block executes a protocol using the OS to control F_BOOL and F_REAL parameters. For this purpose, it implements a special safety protocol and monitors the required operator sequence. It has no relation to the function behind the operation. An F_SWC_P capable of controlling multiple command functions (SWC_MOS) must be positioned for each F-Runtime group.

For the Maintenance Override function, you must assign a unique system-wide ID to the F-CPU. There are two ways you can do this:

- Assign the ID to the IDENT input on the F_SWC_P F-Block.
- Assign the ID to the higher-level designation (HID) of the F-CPU.

Priority is given to the ID at the IDENT input. If you assign the ID to the HID of the F-CPU rather than using the IDENT input, the IDENT input remains empty during compilation.

Using a key-operated switch

To ensure that only authorized persons carry out operations via the OS, you can connect the F_SWC_P F-Block to a key-operated switch at the EN_SWC input.

During an operation, the EN_SWC input must be set to one (EN_SWC = 'true'). If the input is reset to zero (EN_SWC = 'false') after an operation, all existing bypasses are disabled. However, all fail-safe value settings are retained.

⚠WARNING

The "Maintenance Override" functionality allows changes to the safety program to be made during RUN mode.

As a result, the following safety measures are required:

- The identification of the F-CPU must be unique throughout the entire system. S7 F Systems use the IDENT parameter on F_SWC_P or the HID of the F-CPU for this purpose..
- Make sure that changes that could compromise plant safety cannot be made. You can use the EN_SWC input on the F_SWC_P F-Block for this purpose, for example, by controlling it with a key-operated switch or on a process-specific basis via the safety program.
- Make sure that only authorized persons can make changes.

  Examples:
  - Control the EN_SWC input on the F_SWC_P F-Block with a key-operated switch.
  - Set up access protection at operator stations where the "Maintenance Override" function can be performed.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	EN_SWC	F_BOOL	Key-operated switch: 0=no operation permitted 1=operation permitted	0
	IDENT	STRING [32]	CPU ID	''
	MAX_TIME	F_TIME	Maximum duration of an operation; timeout time	1 min

A.2.5.2 F_SWC_BO: Processing of a parameter of data type F_BOOL for operator input via the OS

Function

The F_SWC_BO F-Block enables changes to be made to F-Parameters of data type F_BOOL in the safety program of the F-CPU from an operator station (Maintenance Override).

The OUT output is interconnected in the safety program to the input/output whose value is to be changed.

OUT and AKT_VAL can be set or reset independently of an operation via the S and R inputs. OUT and AKT_VAL are set on a positive edge at S. Priority is given to resets and therefore, resetting is performed as long as R equals 1 (R = 1). Because the key-operated switch is only relevant for bypasses resulting from an operation (soft bypass), OUT and AKT_VAL can also be set when the key-operated switch is disabled.

S and R can be used as a hard bypass for connecting a sensor. The hard bypass always has priority over a soft bypass from the OS. This is why an operation in progress is canceled when a hard bypass is enabled.

If a change has been made on the faceplate according to the specified operator sequence within the time assigned at MAX_TIME on F_SWC_P, the value entered on the faceplate is made available at output OUT.

⚠WARNING

The "Maintenance Override" functionality allows changes to the safety program to be made during RUN mode.

As a result, the following safety measures are required:

- Make sure that changes that could compromise plant safety cannot be made. You can use the EN_SWC input on the F_SWC_P F-Block for this purpose, for example, by controlling it with a key-operated switch or on a process-specific basis via the safety program.
- Make sure that only authorized persons can make changes. Examples:
  - Control the EN_SWC input on the F_SWC_P F-Block with a key-operated switch.
  - Set up access protection at operator stations where the "Maintenance Override" function can be performed.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	S	F_BOOL	Set input	0
	R	F_BOOL	Reset input	0
	CS_VAL	F_BOOL	Cold restart	0
Outputs:	OUT	F_BOOL	Current value of the operated parameter	0
	AKT_VAL	BOOL	Current value of the operated parameter for the OS	0

Note

The interconnection of the AKT_VAL output establishes the connection to the OS.

⚠ WARNING

Interconnection of the CS_VAL input is not permitted.

Startup characteristics

During startup, OUT and AKT_VAL are initialized with the value of CS_VAL during a cold start.

⚠ WARNING

F-Startup

After an F-Startup, make sure that the safety of the system is not compromised by the presence of the CS_VAL value at outputs OUT and AKT_VAL.

Error handling

- An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

See also

SWC_MOS: Command function for Maintenance Override (Page 264)

A.2.5.3 F_SWC_R: Processing of a parameter of data type F_REAL for operator input via the OS

Function

The F_SWC_R F-Block enables changes to be made to F-Parameters of data type F_REAL in the safety program of the F-CPU from an operator station (Maintenance Override).

The OUT output is interconnected in the safety program to the input/output whose value is to be changed.

The limits for the change are specified using inputs MIN and MAX.

If a change has been made on the faceplate according to the specified operator sequence within the time assigned at MAX_TIME on F_SWC_P, the value entered on the faceplate is made available at output OUT provided it satisfies the following conditions:

- The value is within the limits assigned at inputs MIN and MAX.

⚠ WARNING

The "Maintenance Override" functionality allows changes to the safety program to be made during RUN mode.

As a result, the following safety measures are required:

- Make sure that changes that could compromise plant safety cannot be made. You can use the EN_SWC input on the F_SWC_P F-Block for this purpose, for example, by controlling it with a key-operated switch or on a process-specific basis via the safety program.
- Make sure that only authorized persons can make changes.
  Examples:
  - Control the EN_SWC input on the F_SWC_P F-Block with a key-operated switch.
  - Set up access protection at operator stations where the "Maintenance Override" function can be performed.

As an alternative to the measures above, select the inputs MIN, and MAX so that no values that could compromise plant safety can be specified via the Maintenance Override function.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	MIN	F_REAL	Minimum value for IN	0.0
	MAX	F_REAL	Maximum value for IN	100.0
	CS_VAL	F_REAL	Cold start value	0.0
Outputs:	OUT	F_REAL	Current value of the operated parameter	0.0
	AKT_VAL	REAL	Current value of the operated parameter for the OS	0.0

⚠ WARNING

The CS_VAL, MIN, and MAX inputs must not be interconnected.

Note

The interconnection of the AKT_VAL output establishes the connection to the OS.

Startup characteristics

During startup, OUT and AKT_VAL are initialized with the cold start value CS_VAL provided it falls within the MIN and MAX limits. If CS_VAL is less than (<) MIN, OUT and AKT_VAL are initialized with the value of MIN. If CS_VAL is greater than (>) MAX, OUT and AKT_VAL are initialized with the value of MAX.

⚠ WARNING

F-Startup

After an F-Startup, make sure that the safety of the system is not compromised by the presence of the CS_VAL value at outputs OUT and AKT_VAL.

Error handling

- An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

See also

SWC_MOS: Command function for Maintenance Override (Page 264)

A.2.5.4 F_FR_FDI: Conversion from F_REAL to F_DINT

Function

This F-Block converts the F_REAL F-Data type at the IN input to the F_DINT F-Data type at the OUT output.

Following the conversion of F_REAL to F_DINT, if the value at the IN input exceeds the upper limit that can be portrayed by the F_INT data type, 2,147,483,647 is output at the OUT output and output OUTU is set to 1. At F_DINT values greater than (>) 2,147,483,583, the range is already exceeded.

If the range is undershot (IN is less than (<) the F_DINT value that can be portrayed), the smallest F_DINT value of -2,147,483,648 is output at output OUT, and output OUTL is set to 1.

Inaccuracies/rounding

If the value at input IN is located outside the range -16777216,0 to 16777215,0, it is possible for the output value to be rounded in F_DINT format, as values in the F_REAL format require 8 bits of the 32-bit real value to represent the exponent.

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	F_REAL	Input	0.0
Outputs:	OUT	F_DINT	Output	0
	OUTU	F_BOOL	Upper number range violation	0
	OUTL	F_BOOL	Lower number range violation	0

Error handling

- If the IN input is an invalid floating point number (NaN), 0 is output at the OUT output and OUTU and OUTL are set to 1.
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.5.5 F_FDI_FR: Conversion from F_DINT to F_REAL

Function

This F-Block converts the F_DINT F-Data type at the IN input to the F_REAL F-Data type at the OUT output.

Inaccuracies/rounding

If the value at input IN is greater than (>) 16,777,215 or less than (<) -16,777,216, this can result in an inaccuracy in the output value of 127, maximum, compared to the input value. That is, the value in F_DINT format is rounded up or rounded off for representation in F_REAL format, as 8 bits of the 32-bit real value are required to represent the exponent. If the value is rounded off, RND_OFF = 1 is set. If the value is rounded up, RND_UP = 1 is set.

If values at input IN are greater than or equal to (>=) 2,147,483,584, the output value of data type F_REAL is always rounded up. In this case, RND_UP =1 is always set.

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	F_DINT	Input	0
Outputs:	OUT	F_REAL	Output	0.0
	RND_UP	F_BOOL	Output value is a rounded-up value	0
	RND_OFF	F_BOOL	Output value is a rounded-off value	0

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

A.2.5.6 F_BO_FBO: Conversion from BOOL to F_BOOL

Function

This F-Block converts the BOOL data type at the IN input to the corresponding F_BOOL F-Data type at the OUT output.

This enables signals formed in the standard user program to be evaluated in the safety program following a validity check.

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	BOOL	Input	0
Output:	OUT	F_BOOL	Output	0

Error handling

None

A.2.5.7 F_R_FR: Conversion from REAL to F_REAL

Function

This F-Block converts the REAL data type at the IN input to the corresponding F_REAL F-Data type at the OUT output.

This enables signals formed in the standard user program to be evaluated in the safety program following a validity check (using F-Block F_LIM_R, for example).

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	REAL	Input	0.0
Output:	OUT	F_REAL	Output	0.0

Error handling

None

A.2.5.8 F_QUITES: Fail-safe acknowledgement via the ES/OS

Function

This F-Block enables fail-safe acknowledgment from a non-fail-safe ES/OS. This allows reintegration of F-I/O to be controlled via the ES/OS, for example. Acknowledgment takes place in two steps:

1. Changing the IN input to the value "6"
2. Changing the IN input from the value "6" to the value "9" within one minute

The F-Block evaluates whether or not the value at input IN changes to "9" within **1 second at the earliest** or **1 minute at the latest** after the value changes to "6". The signal "1" is then output at the OUT output (output for acknowledgement) for the duration of one cycle.

If an invalid value is entered or if the change in the value to "9" occurs before 1 second or after 1 minute has elapsed, the IN input is reset to 0 and the two steps indicated above have to be repeated.

During the time in which the change from "6" to "9" must occur, the non-fail-safe Q output is set to 1. Otherwise, Q has a value of 0.

⚠WARNING

Reintegration through User Acknowledgment with F_QUITES

The two acknowledgment steps must not be triggered by a single operation, for example, by automatically storing them along with the time conditions in one program requiring a single operation to trigger them. By programming separate acknowledgment steps, you prevent erroneous triggering of an acknowledgment by your non-fail-safe operator station.

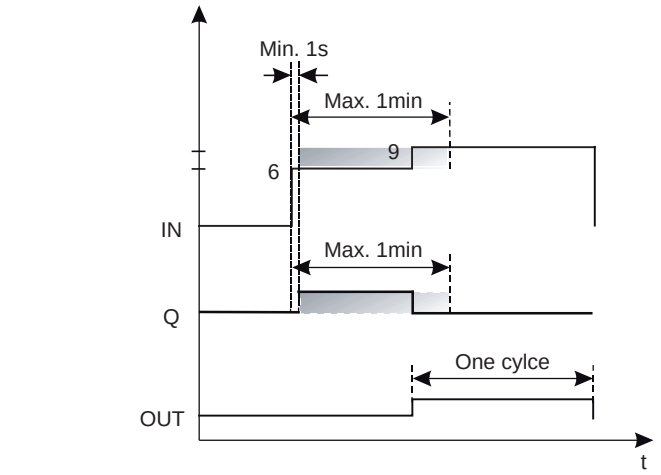
Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	INT	Input	0
Outputs:	OUT	F_BOOL	ACKNOWLEDGMENT OUTPUT	0
	Q	BOOL	Status of the time evaluation	0

Changing the collective signature of the offline safety program

If the above two acknowledgement steps are entered directly via the ES in CFC test mode rather than via the OS, the collective signature of the offline safety program changes as a result of the acknowledgment. To avoid this, you must ensure that a zero is entered after a 9 or an invalid value.

Timing Diagram



Operator control and monitoring

Parameters IN and Q have the system attribute S7_m_c. They can therefore be directly operated and monitored from an operator interface system (OS).

Error handling

An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

A.2.5.9 F_TI_FTI: Conversion from TIME to F_TIME

Function

This F-Block converts the TIME data type at the IN input to the corresponding F_TIME F-Data type at the OUT output.

This enables signals formed in the standard user program to be evaluated in the safety program following a validity check (using F-Block F_LIM_TI, for example).

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	TIME	Input	T# 0ms
Output:	OUT	F_TIME	Output	T# 0ms

Error handling

None

A.2.5.10 F_I_FI: Conversion from INT to F_INT

Function

This F-Block converts the INT data type at the IN input to the corresponding F_INT F-Data type at the OUT output.

This enables signals formed in the standard user program to be evaluated in the safety program following a validity check (using F-Block F_LIM_I, for example).

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	INT	Input	0
Output:	OUT	F_INT	Output	0

Error handling

None

A.2.5.11 F_FI_FR: Conversion from F_INT to F_REAL

Function

This F-Block converts the F_INT F-Data type at the IN input to the F_REAL F-Data type at the OUT output.

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	F_INT	Input	0
Output:	OUT	F_REAL	Output	0.0

Error handling

None

A.2.5.12 F_FR_FI: Conversion from F_REAL to F_INT

Function

This F-Block converts the F_REAL F-Data type at the IN input to the F_INT F-Data type at the OUT output.

If the value at the IN input exceeds the upper limit which can be portrayed by the INT data type (range: -32768 to +32767), +32767 is output at the OUT output and output OUTU is set to 1. If the value lower range is violated, -32768 is output and the OUTL output is set to 1.

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	F_REAL	Input	0.0
Output:	OUT	F_INT	Output	0
	OUTU	F_BOOL	Upper number range violation	0
	OUTL	F_BOOL	Lower number range violation	0

Error handling

- If the IN input is an invalid floating point number (NaN), 0 is output at the OUT output and OUTU and OUTL are set to 1.
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.5.13 F_CHG_R: Safety Data Write for F_REAL

Function

The F_CHG_R F-Block enables changes to be made to F-Parameters in the safety program of the F-CPU from an operator station (Safety Data Write). For this purpose, the block implements a special safety protocol and monitors the required operator sequence.

The F-Block can only be used in conjunction with the associated faceplate in the OS (see section "Connection to the faceplate" below).

The OUT output is interconnected in the safety program to the input/output whose value is to be changed.

The limits for the change are specified using inputs MIN and MAX.

The maximum increment of the change is specified at input MAXDELTA.

The time during which the change must be completed is specified at the TIMEOUT input.

If a change has been made on the faceplate according to the specified operator sequence within the monitoring time assigned at the TIMEOUT input, the value entered on the faceplate is made available at output OUT, provided the following conditions are met:

- The value is within the limits assigned at inputs MIN and MAX.
- The maximum change increment assigned at input MAXDELTA is not exceeded.

The "Safety Data Write" functionality must be enabled using input EN_CHG = 1.

Note

If EN_CHG changes to 0 during a transaction that has already started, a final confirmer value from the confirmer isn't made available at output OUT until input EN_CHG changes back to 1 (within the F-monitoring time).

⚠WARNING

The "Safety Data Write" functionality makes changes in the safety program during RUN mode.

As a result, the following safety measures are required:

- Make sure that changes that could compromise plant safety cannot be made. You can use input EN_CHG for this purpose, for example, by controlling it with a keyswitch or on a process-specific basis via the safety program.
- Make sure that only authorized persons can make changes. Examples:
  - Control input EN_CHG with a keyswitch.
  - Set up access protection at operator stations where the "Safety Data Write" function can be performed.

As an alternative to the measures above, select the inputs MIN, MAX and MAXDELTA so that no values that could compromise plant safety can be specified via Safety Data Write.

If input CS_MODE = 1, the value made available at output OUT is applied to input CS_VAL, and output CHANGED = 1 is set.

⚠WARNING

The CHANGED output cannot be evaluated in the safety program.

CHANGED = 1 merely indicates that a change at output OUT has been transferred to input CS_VAL.

If necessary, the value at input CS_VAL must be corrected manually in the offline program and the load memory so that the updated value at CS_VAL is actually in effect at the next cold restart.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	SAFE_ID1	F_DINT	ADDRESS RELATION ID1 for interconnecting the block instance and the faceplate	0
	SAFE_ID2	F_DINT	ADDRESS RELATION ID2 for interconnecting the block instance and the faceplate	0
	TIMEOUT	F_TIME	TIMEOUT FOR F-PARAMETER CHANGE	T#60000 ms
	MIN	F_REAL	MINIMUM LIMIT FOR F-PARAMETER CHANGE	0.0
	MAX	F_REAL	MAXIMUM LIMIT FOR F-PARAMETER CHANGE	100.0
	MAXDELTA	F_REAL	MAXIMUM DELTA FOR F-PARAMETER CHANGE	10.0
	CS_VAL	F_REAL	INITIAL VALUE OF OUT AT COLDSTART	0.0
	CS_MODE	F_BOOL	1 = TRANSFER CHANGED OUT TO CS_VAL 0 = CS_VAL REMAINS UNCHANGED	0
	WS_MODE	F_BOOL	1 = USE LAST OUT AT WARMSTART 0 = USE CS_VAL AT WARMSTART	1
	EN_CHG	F_BOOL	ENABLE F-PARAMETER CHANGE. 1 = ENABLE 0 = DISABLE	0

	Name	Data type	Description	Default
Outputs:	OUT	F_REAL	Current, fail-safe REAL value that is used by the safety program	0.0
	CHANGED	BOOL	1 = CS_VAL HAS BEEN CHANGED	0
	CS_USED	F_BOOL	shows which value was made available for OUT after fail-safe start 1 = CS_VAL IS USED 0 = LAST VALID VALUE	0
	DIAG	WORD	DIAGNOSTIC INFORMATION Bit 0 = 1: Error in the safety data format Bit 1 = 1: MIN error Bit 2 = 1: MAX error Bit 3 = 1: DELTA error Bit 4 = 1: TIMEOUT error Bit 5 = 1: ID1 error Bit 6 = 1: ID2 error Bit 7 = 1: ID1_C error Bit 8 = 1: ID2_C error Bit 9 = 1: Test_ID1 error Bit 10 = 1: Test_ID2 error Bit 11 = 1: Error in the safety data format IN Bit 12 = 1: TIMEOUT error during OS test Bit 13 = 1: Error: negative number at TIMEOUT input Bit 14-15: Reserve	W#16#0
	USER	STRING [24]	USERNAME FROM OS	''
	CURR_R	REAL	Copy of OUT.DATA Here, the unit of measurement to be displayed in the faceplate can be assigned using the "Unit" input/output property.	0.0

⚠ WARNING

The MIN, MAX, and MAXDELTA inputs must not be interconnected.

Connection to the faceplate

Communication between a block instance and its assigned faceplate takes place in the background by means of a special safety protocol. To configure the association between a block instance and its assigned faceplate, select a pair of numbers that are unique from all others in the system (Part 1 and Part 2). Assign the pair of numbers to the SAFE_ID1 and SAFE_ID2 parameters as follows:

- To the SAFE_ID1 and SAFE_ID2 inputs of the F_CHG_R in *CFC* of your safety program
- To the SAFE_ID1 and SAFE_ID2 parameters of the associated block icon in the *WinCC Graphics Designer*

> **⚠ WARNING**
>
> **Parameters SAFE_ID1 and SAFE_ID2**
>
> The pair of numbers for SAFE_ID1 and SAFE_ID2 of an F-Block instance must be unique from all others in the system.
>
> An instance of the F-Block and the block icon of the associated faceplate must be given the same pair of numbers for the SAFE_ID1 and SAFE_ID2 parameters.
>
> The SAFE_ID1 parameter must be a non-zero value that is unique from all others in the program.

For information about the associated faceplate, refer to " Configuring the Faceplate for Safety Data Write. (Page 135) S7 F/FH Systems Programming and Operating Manual, chapter: Configuring faceplate for Safety Data Write ".

Startup characteristics

Following an F-Startup, the F-Block behaves as follows:

- Following a CPU-STOP with subsequent coldstart of the F-CPU or during initial run:

  During the first cycle after a coldstart or after an initial run, the value assigned at input CS_VAL is made available at output OUT. The CS_USED output is set to 1. CS_USED is reset to 0 once the "Safety Data Write" is performed successfully for the first time.

  Note

  The configured value at input CS_VAL must be between the MIN and MAX values.

- Following a CPU STOP with subsequent restart (warm restart) of the F-CPU, or following an F-STOP with subsequent rising edge at the RESTART input of the F-Block F_SHUTDN:

  In the first cycle after a restart (warm restart), or following a rising edge at the RESTART input of the F-Block F_SHUTDN, the last valid OUT value is made available at the OUT output if input WS_MODE = 1. The CS_USED output retains its default value (0). If input WS_MODE = 0, the F-Block behaves the same as after a cold restart.

  Note

  Prior to initial processing of the F-Block following an F-Startup, the default value is applied at output OUT and CS_USED.

⚠ WARNING

F-Startup

Following an F-Startup, plant safety must not be compromised due to either the presence of the CS_VAL value or the presence of the last valid value at the OUT output.

If necessary, evaluate the CS_USED output to determine whether the CS_VAL value or the last valid value at the OUT output has been made available after an F-Startup. In addition, the default value "0" of CS_USED must not be changed.

If a restart (warm restart) is performed after a cold restart, CS_USED is reset to the default value (0), even if the CS_VAL value is currently present at output OUT.

Error handling

- An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Safety Data Format Failure in DB the DB" (Event ID 16#75DA).
- The DIAG output of the F-Block signals when an error is detected. This output must be checked if a transaction (Safety Data Write) fails. The individual errors remain active until the failed action has been repeated successfully. The meaning of the individual bits is described below:

All bits = 0	No problem; error-free operation
Bit 0 = 1	Safety data format error at an input of the F-Block
Bit 1 = 1	MIN error: Transaction failed because the modified value is less than the MIN limit value.
Bit 2 = 1	MAX error: Transaction failed because the modified value is greater than the MAX limit value.
Bit 3 = 1	DELTA error: Transaction failed because the change increment exceeds the permissible MAXDELTA value; the modified value must be between OUT ± MAXDELTA.
Bit 4 = 1	TIMEOUT error: A transaction was initiated but not completed within the specified time.
Bit 5 = 1	ID1 error: Transaction failed because SAFE_ID1 does not match at the F-Block instance and the faceplate in the OS.
Bit 6 = 1	ID2 error: Transaction failed because SAFE_ID2 does not match at the F-Block instance and the faceplate in the OS.
Bit 7 = 1	ID1_C error: Transaction failed because SAFE_ID1 does not match at the F-Block instance and the faceplate in the OS.
Bit 8 = 1	ID2_C error: Transaction failed because SAFE_ID2 does not match at the F-Block instance and the faceplate in the OS.
Bit 9=1	Test_ID1 error: OS test failed because SAFE_ID1 does not match at the F-Block instance and the faceplate in the OS.
Bit 10 = 1	Test_ID2 error: OS test failed because SAFE_ID2 does not match at the F-Block instance and the faceplate in the OS.
Bit 11 = 1	Error in the safety data format IN: Transaction failed because of safety data format error at the new value of the faceplate
Bit 12 = 1	TIMEOUT error: During OS test
Bit 13 = 1	TIMEOUT error: Negative number at input TIMEOUT of F-Block

A.2.5.14 F_CHG_BO: Safety Data Write for F_BOOL

Function

The F_CHG_BO F-Block enables changes to be made to F-Parameters in the safety program of the F-CPU from an operator station (Safety Data Write). For this purpose, the F-Block implements a special safety protocol and monitors the required operator sequence.

The F-Block can only be used in conjunction with the associated faceplate in the OS (see "Connection to the faceplate" below).

The OUT output is interconnected in the safety program to the input/output whose value is to be changed.

The time during which the change must be completed is specified at the TIMEOUT input.

If a change has been made on the faceplate according to the specified operator sequence within the monitoring time assigned at the TIMEOUT input, the value entered on the faceplate is made available at output OUT.

The "Safety Data Write" functionality must be enabled using input EN_CHG = 1.

Note

If EN_CHG changes to 0 during a transaction that has already started, a final confirmer value from the confirmer isn't made available at output OUT until input EN_CHG changes back to 1 (within the F-monitoring time).

⚠ WARNING

The "Safety Data Write" functionality makes changes in the safety program during RUN mode.

As a result, the following safety measures are required:

- Make sure that changes that could compromise plant safety cannot be made. You can use input EN_CHG for this purpose, for example, by controlling it with a keyswitch or on a process-specific basis via the safety program.
- Make sure that only authorized persons can make changes. Examples:
  - Control input EN_CHG with a keyswitch.
  - Set up access protection at operator stations where the "Safety Data Write" function can be performed.

If input CS_MODE = 1, the value made available at output OUT is applied to input CS_VAL, and output CHANGED = 1 is set.

⚠ WARNING

The CHANGED output cannot be evaluated in the safety program.

CHANGED = 1 merely indicates that a change at output OUT has been transferred to input CS_VAL.

If necessary, the value at input CS_VAL must be corrected manually in the offline program and the load memory so that the value updated at the CS_VAL input using the Safety Data Write function is actually in effect at the next cold restart.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	SAFE_ID1	F_DINT	ADDRESS RELATION ID1 for interconnecting the block instance and the faceplate	0
	SAFE_ID2	F_DINT	ADDRESS RELATION ID2 for interconnecting the block instance and the faceplate	0
	TIMEOUT	F_TIME	TIMEOUT FOR F-PARAMETER CHANGE	T#60000 ms
	CS_VAL	F_BOOL	INITIAL VALUE OF OUT AT COLDSTART	0
	CS_MODE	F_BOOL	1 = TRANSFER CHANGED OUT TO CS_VAL 0 = CS_VAL remains unchanged.	0
	WS_MODE	F_BOOL	1 = USE LAST OUT AT WARMSTART 0 = USE CS_VAL AT WARMSTART	1
	EN_CHG	F_BOOL	ENABLE F-PARAMETER CHANGE 1 = Enable 0 = Disable	0

	Name	Data type	Description	Default
Outputs:	OUT	F_BOOL	Current, fail-safe BOOL value that is used by the safety program	0
	CHANGED	BOOL	1 = CS_VAL HAS BEEN CHANGED	0
	CS_USED	F_BOOL	Shows which value was made available for OUT following startup 1 = CS_VAL IS USED 0 = LAST VALID VALUE	0
	DIAG	WORD	Diagnostic information Bit 0 = 1: Error in the safety data format Bit 1 = 1: Reserve Bit 2 = 1: Reserve Bit 3 = 1: Reserve Bit 4 = 1: TIMEOUT error Bit 5 = 1: ID1 error Bit 6 = 1: ID2 error Bit 7 = 1: ID1_C error Bit 8 = 1: ID2_C error Bit 9 = 1: Test_ID1 error Bit 10 = 1: Test_ID2 error Bit 11 = 1: Error in the safety data format IN Bit 12 = 1: TIMEOUT error during OS test Bit 13 = 1: Error: negative number at TIMEOUT input Bit 14-15: Reserve	W#16#0
	USER	STRING [24]	Login of current operator on the OS.	''

Connection to the faceplate

Communication between a block instance and its assigned faceplate takes place in the background by means of a special safety protocol. To configure the association between a block instance and its assigned faceplate, select a pair of numbers that are unique from all others in the system (Part 1 and Part 2). Assign the pair of numbers to the SAFE_ID1 and SAFE_ID2 parameters as follows:

- To the SAFE_ID1 and SAFE_ID2 inputs of the F_CHG_BO in *CFC* of your safety program
- To the SAFE_ID1 and SAFE_ID2 parameters of the associated block icon in the *WinCC Graphics Designer*

> **⚠ WARNING**
>
> **Parameters SAFE_ID1 and SAFE_ID2**
>
> The pair of numbers for SAFE_ID1 and SAFE_ID2 of an F-Block instance must be unique from all others in the system.
>
> An instance of the F-Block and the block icon of the associated faceplate must be given the same pair of numbers for the SAFE_ID1 and SAFE_ID2 parameters.
>
> The SAFE_ID1 parameter must be a non-zero value that is unique from all others in the program.

For information about the associated faceplate, refer to " Configuring the Faceplate for Safety Data Write. (Page 135) S7 F/FH Systems Programming and Operating Manual, chapter: Configuring faceplate for Safety Data Write ".

Startup characteristics

Following an F-Startup, the F-Block behaves as follows:

- Following a CPU-STOP with subsequent coldstart of the F-CPU or during initial run:

  During the first cycle after a coldstart or after an initial run, the value assigned at input CS_VAL is made available at output OUT. The CS_USED output is set to 1. CS_USED is reset to 0 once the "Safety Data Write" is performed successfully for the first time.

- Following a CPU STOP with subsequent restart (warm restart) of the F-CPU, or following an F-STOP with subsequent rising edge at the RESTART input of the F-Block F_SHUTDN:

  In the first cycle after a restart (warm restart), or following a rising edge at the RESTART input of the F-Block F_SHUTDN, the last valid OUT value is made available at the OUT output if input WS_MODE = 1. The CS_USED output retains its default value (0). If input WS_MODE = 0, the F-Block behaves the same as after a cold restart.

Note

Prior to initial processing of the F-Block following an F-Startup, the default value is applied at output OUT and CS_USED.

⚠WARNING

F-Startup

Following an F-Startup, plant safety must not be compromised due to either the presence of the CS_VAL value or the presence of the last valid value at the OUT output.

If necessary, evaluate the CS_USED output to determine whether the CS_VAL value or the last valid value at the OUT output has been made available after an F-Startup. In addition, the default value "0" of CS_USED must not be changed.

If a restart (warm restart) is performed after a cold restart, CS_USED is reset to the default value (0), even if the CS_VAL value is currently present at output OUT.

Error handling

- An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Safety Data Format Failure in DB the DB" (Event ID 16#75DA)
- The DIAG output of the F-Block signals when an error is detected. This output must be checked if a transaction (Safety Data Write) fails. The individual errors remain active until the failed action has been repeated successfully. The meaning of the individual bits is described below:

All bits = 0	No problem; error-free operation
Bit 0 = 1	Safety data format error at an input of the F-Block
Bit 1 = 1	Reserve
Bit 2 = 1	Reserve
Bit 3 = 1	Reserve
Bit 4 = 1	TIMEOUT error: A transaction was initiated but not completed within the specified time.
Bit 5 = 1	ID1 error: Transaction failed because SAFE_ID1 does not match at the F-Block instance and the faceplate in the OS.
Bit 6 = 1	ID2 error: Transaction failed because SAFE_ID2 does not match at the F-Block instance and the faceplate in the OS.
Bit 7 = 1	ID1_C error: Transaction failed because SAFE_ID1 does not match at the F-Block instance and the faceplate in the OS.
Bit 8 = 1	ID2_C error: Transaction failed because SAFE_ID2 does not match at the F-Block instance and the faceplate in the OS.
Bit 9 = 1	Test_ID1 error: OS test failed because SAFE_ID1 does not match at the F-Block instance and the faceplate in the OS.
Bit 10 = 1	Test_ID2 error: OS test failed because SAFE_ID2 does not match at the F-Block instance and the faceplate in the OS.
Bit 11 = 1	Error in the safety data format IN: Transaction failed because of safety data format error at the new value of the faceplate
Bit 12 = 1	TIMEOUT error: During OS test
Bit 13 = 1	TIMEOUT error: Negative number at input TIMEOUT of F-Block

A.2.5.15 F_FBO_BO: Conversion from F_BOOL to BOOL

Function

This block converts F-Data type F_BOOL at input IN to the elementary data type BOOL at output OUT.

This enables you to evaluate signals that were generated in the safety program in the standard user program, as well.

This block must be placed in the standard user program.

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	F_BOOL	Input	—
Output:	OUT	BOOL	Output	—

Error handling

None

A.2.5.16 F_FR_R: Conversion from F_REAL to REAL

Function

This block converts F-Data type F_REAL at input IN to the elementary data type REAL at output OUT.

This enables you to evaluate signals that were generated in the safety program in the standard user program, as well.

This block must be placed in the standard user program.

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	F_REAL	Input	—
Output:	OUT	REAL	Output	—

Error handling

None

A.2.5.17 F_FI_I: Conversion from F_INT to INT

Function

This block converts F-Data type F_INT at input IN to the elementary data type INT at output OUT.

This enables you to evaluate signals that were generated in the safety program in the standard user program, as well.

This block must be placed in the standard user program.

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	F_INT	Input	—
Output:	OUT	INT	Output	—

Error handling

None

A.2.5.18 F_FTI_TI: Conversion from F_TIME to TIME

Function

This block converts F-Data type F_TIME at input IN to the elementary data type TIME at output OUT.

This enables you to evaluate signals that were generated in the safety program in the standard user program, as well.

This block must be placed in the standard user program.

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	F_TIME	Input	—
Output:	OUT	TIME	Output	—

Error handling

None

A.2.5.19 SWC_MOS: Command function for Maintenance Override

SWC_MOS

This is a standard block that establishes a connection to the faceplate. It also provides the block icon and faceplate on the OS with all values for display and protocol execution and generates messages for *PCS 7* via Alarm_8P.

For each command function, an SWC_MOS must be positioned and inserted into the plant hierarchy.

With the SWC_MOS block, only operator input of a fail-safe value is possible.

Note

When used with *PCS 7*, one PO license is used for each instance of the SWC_MOS block in the safety program.

This block generates the following ALARM_8 messages for the message system:

- Advance warning message indicating expiration of the bypass time
- End-of-operation status
- Bypass active/inactive

NOTICE
When assigning a name for the block, keep in mind that the following illegal characters are automatically replaced by a $ sign during the transfer to the OS: blank ? * ' : Avoid these characters, otherwise an operator input will not be possible.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	NOTE	STRING [32]	Faceplate title	—
	AKT_B1	BOOL	Actual value of first parameter for OS	0
	VMOD_B1B	BOOL	Status of channel (BOOL)	0
	Q_B1B	BOOL	Process value	0
	VMOD_B1R	REAL	Status of channel (REAL)	0.0
	V_B1R	REAL	Process value	0.0
	AKT_B2	BOOL	Actual value of second parameter for OS	0
	VMOD_B2B	BOOL	Status of channel (BOOL)	0
	Q_B2B	BOOL	Process value	0
	VMOD_B2R	REAL	Status of channel (REAL)	0.0
	V_B2R	REAL	Process value	0.0
	AKT_B3	BOOL	Actual value of third parameter for OS	0
	VMOD_B3B	BOOL	Status of channel (BOOL)	0
	Q_B3B	BOOL	Status of channel (BOOL)	0
	VMOD_B3R	REAL	Status of channel (REAL)	0.0
	V_B3R	REAL	Process value	0.0
	AKT_TR	BOOL	Actual value of retrigger signal for OS	0
	T_WARN	TIME	Advance warning time for active bypass	0s
	AKT_V_B	BOOL	Actual value of BOOL substitute value for OS	0
	AKT_V_R	REAL	Actual value of REAL substitute value for OS	0.0
	MODE	WORD	Mutually exclusive interlock	W#16#0

A.2.6 F-Channel drivers for F-I/O

Overview

Block name	Block number	Description
F_CH_BI	FB 354	F-Channel driver for inputs of data type BOOL of fail-safe DP standard slaves and fail-safe standard I/O devices
F_CH_BO	FB 355	F-Channel driver for outputs of data type BOOL of fail-safe DP standard slaves and fail-safe standard I/O devices
F_PA_AI	FB 356	Fail-safe channel driver for fail-safe "Transmitter" PA field device
F_PA_DI	FB 357	Fail-safe channel driver for fail-safe "Discrete Input" PA field device
F_CH_DI	FB 377	Fail-safe channel drivers for digital inputs of F-I/O (except fail-safe DP standard slaves)
F_CH_DO	FB 378	Fail-safe channel drivers for digital outputs of F-I/O (except fail-safe DP standard slaves)
F_CH_AI	FB 379	Fail-safe channel drivers for analog inputs of F-I/O (except fail-safe DP standard slaves)
F_CH_II	FB 454	F-Channel driver for inputs of data type INT of fail-safe DP standard slaves and fail-safe standard I/O devices
F_CH_IO	FB 455	F-Channel driver for outputs of data type INT of fail-safe DP standard slaves and fail-safe standard I/O devices
F_CH_DII	FB 465	F-Channel driver for inputs of data type DINT of fail-safe DP standard slaves and fail-safe standard I/O devices
F_CH_DIO	FB 466	F-Channel driver for outputs of data type DINT of fail-safe DP standard slaves and fail-safe standard I/O devices

A.2.6.1 F_CH_BI: F-Channel driver for inputs of data type BOOL of fail-safe DP standard slaves and fail-safe standard I/O devices

Function

This F-Block is used for signal processing of an input value of data type BOOL of fail-safe DP standard slaves and fail-safe standard I/O devices.

The F-Block cyclically reads the input value of data type BOOL of a fail-safe DP standard slave addressed at input VALUE from the associated fail-safe-module driver F_PS_12 that communicates with the fail-safe DP standard slave by means of a safety message frame in accordance with the PROFIsafe bus profile. The fail-safe module driver is automatically placed and interconnected with the *CFC* function "Generate module drivers".

If the digital input value is valid, it is made available at output Q.

A quality code (QUALITY output) is generated for the result value at output Q. The quality code can assume the following states:

State	Quality code (QUALITY output)
Valid value	16#80
Simulation	16#60
Fail-safe value	16#48
Invalid value (F-STOP)	16#00

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ADR_CODE	DWORD	CODE FOR VALUE INTERCONNECTION	Automatically initialized *
	VALUE	BOOL	INPUT VALUE	0
	SIM_I	F_BOOL	SIMULATION VALUE	0
	SIM_ON	F_BOOL	1 = ACTIVATE SIMULATION	0
	PASS_ON	F_BOOL	1 = ACTIVATE PASSIVATION	0
	ACK_NEC	F_BOOL	1 = ACKNOWLEDGEMENT NECESSARY	0
	ACK_REI	F_BOOL	ACKNOWLEDGMENT REINTEGRATION	0
	IPAR_EN	F_BOOL	1 = ENABLE I-PARAMETER ASSIGNMENT	0
Outputs:	PASS_OUT	F_BOOL	1 = PASSIVATION BECAUSE OF ERROR	0
	QBAD	F_BOOL	1 = PROCESS VALUE INVALID	0
	QSIM	F_BOOL	1 = SIMULATION ACTIVE	0
	Q	F_BOOL	PROCESS VALUE	0
	QN	F_BOOL	NEGATED PROCESS VALUE	1
	Q_DATA	BOOL	PROCESS VALUE DATA (for monitoring)	0
	QUALITY	BYTE	QUALITY CODE OF PROCESS VALUE	0
	Q_MOD	BOOL	VALUE FROM MODULE	0
	ACK_REQ	BOOL	ACKNOWLEDGEMENT REQUEST	0
	IPAR_OK	F_BOOL	1 = NEW I-PARAMETER VALUES ASSIGNED	0

*) Input ADR_CODE is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input ADR_CODE is indicated as changed if changes have been made to the address or the symbolic name of the signal at input VALUE.

Addressing

You must interconnect the symbol for the input value of data type BOOL generated with *HW Config* in the symbol table with the VALUE input.

Note

Inversion of the VALUE input in the *CFC Editor* has no effect. Use the QN output instead.

Normal value

If the input value received from the fail-safe DP standard slave is valid, the value is output at output Q with quality code (QUALITY) 16#80.

Simulation

A simulation value can be output at output Q instead of the normal value that is received from the fail-safe DP standard slave.

The value of input SIM_I with quality code (QUALITY) 16#60 is output when input SIM_ON = 1. Simulation has the highest priority. QBAD = 0 is always set. QSIM = 1 is set if the F-Block is in simulation state.

If simulation is enabled, the input value received from the fail-safe DP standard slave is output at output Q_MOD. If communication with the fail-safe DP standard slave is not possible or if a user acknowledgement has not yet occurred following an error, "0" is output.

Q_DATA is output if simulation is disabled.

Fail-safe value

In the following cases, the fail-safe value "0" is output at output Q:

- The digital input value is invalid due to a communication error (PROFIsafe).
- The digital input value is invalid due to a module error or a fail-safe value is received by the module.
- A passivation exists with PASS_ON = 1.
- An F-Startup is pending.
- FV_ACTIVATED is signaled by the module.

The quality code (QUALITY) is set to 16#48, and QBAD = 1 is set.

If the output of the fail-safe value is not caused by a passivation, PASS_OUT = 1 is set additionally in order to passivate other channels.

Reassignment of parameters of a fail-safe DP standard slave

Input IPAR_EN and output IPAR_OK are available for reassignment of parameters of a fail-safe DP standard slave.

Input IPAR_EN corresponds to variable iPar_EN_C and output IPAR_OK corresponds to variable iPar_OK_S in the PROFIsafe bus profile (PROFIsafe Specification V1.30 and higher). If you must set or reset input IPAR_EN for reassignment of parameters of a fail-safe DP standard slave or to find out how to evaluate the IPAR_OK output, refer to PROFIsafe Specification V1.30 and higher or the documentation for the fail-safe DP standard slave.

If more than one fail-safe channel driver is placed for a fail-safe DP standard slave, iPar_EN_C is formed from an OR logic operation of all IPAR_EN inputs of the F-Channel drivers belonging to the fail-safe DP standard slave.

If passivation should occur when IPAR_EN = 1, you must also set variable PASS_ON = 1.

Reintegration after error elimination

After an error has been eliminated, the input value received from the DP standard slave can be reintegrated automatically or not until after a user acknowledgement.

If ACK_NEC = 1 is assigned, a user acknowledgement at input ACK_REI is required after an error has been eliminated. If ACK_NEC = 0 is assigned, automatic reintegration is carried out.

Output ACK_REQ = 1 signals that the error has been eliminated and that a user acknowledgment at input ACK_REI is necessary for reintegration.

A user acknowledgement is not required for reintegration after PASS_ON = 1. A user acknowledgement is not required for reintegration after F-Startup following a CPU-STOP if the fail-safe DP standard slave starts up by means of the "System start" slave state (20) in accordance with PROFIsafe Specification V1.30 and higher. Otherwise, a communication error (PROFIsafe) is detected.

⚠WARNING

Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.

Communication errors (PROFIsafe) always have to be acknowledged at input ACK_REI irrespective of ACK_NEC. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

⚠WARNING

Startup protection for short-term power failure of the fail-safe DP standard slave

Following a power failure of the fail-safe DP standard slave lasting less than the F-Monitoring time for the fail-safe DP standard slaves specified in *HW Config* (see section entitled "Run times, F-Monitoring times, and response times (Page 410)"), automatic reintegration can occur regardless of your setting for input ACK_NEC, as described for the case when ACK_NEC = 0.

If for this case, automatic reintegration is not permissible for the relevant process, you must program startup protection by evaluating the QBAD or PASS_OUT outputs.

In the event of a power failure of the fail-safe DP standard slave lasting longer than the F-Monitoring time for the fail-safe DP standard slave specified in *HW Config*, the F-System detects a communication error.

Startup characteristics

After an F-Startup, communication first has to be established between the F-Module driver and the fail-safe DP standard slave. During this time, the fail-safe value "0" is output with quality code (QUALITY output) 16#48, and outputs QBAD = 1 and PASS_OUT = 1 are set.

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

Behavior during F-STOP

In the event of an F-STOP, quality code 16#00 is output at the QUALITY output and QBAD.DATA = 1 is set. All other variables are frozen.

A.2.6.2 F_CH_BO: F-Channel driver for outputs of data type BOOL of fail-safe DP standard slaves and fail-safe standard I/O devices

Function

This F-Block is used for signal processing of an output value of data type BOOL of fail-safe DP standard slaves and fail-safe standard I/O devices.

The F-Block cyclically writes the output value of data type BOOL for the output of a fail-safe DP standard slave addressed at output VALUE to the associated fail-safe module driver F_PS_12 that communicates with the fail-safe DP standard slave by means of a safety message frame in accordance with the PROFIsafe bus profile. The fail-safe module driver is automatically placed and interconnected with the *CFC* function "Generate module drivers".

A quality code is generated for the output value that is written to the fail-safe DP standard slave. The quality code can assume the following states:

State	Quality code
Valid value	16#80
Simulation	16#60
Fail-safe value	16#48
Invalid value (F-STOP)	16#00

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ADR_CODE	DWORD	CODE FOR VALUE INTERCONNECTION	Automatically initialized *
	I	F_BOOL	PROCESS VALUE	0
	SIM_I	F_BOOL	SIMULATION VALUE	0
	SIM_MOD	F_BOOL	1 = SIMULATION HAS PRIORITY	0
	SIM_ON	F_BOOL	1 = ACTIVATE SIMULATION	0
	PASS_ON	F_BOOL	1 = ACTIVATE PASSIVATION	0
	ACK_NEC	F_BOOL	1 = ACKNOWLEDGEMENT NECESSARY	0
	ACK_REI	F_BOOL	ACKNOWLEDGMENT REINTEGRATION	0
	IPAR_EN	F_BOOL	1 = ENABLE I-PARAMETER ASSIGNMENT	0
Outputs:	PASS_OUT	F_BOOL	1 = PASSIVATION BECAUSE OF ERROR	0
	QBAD	F_BOOL	1 = PROCESS VALUE INVALID	0
	QSIM	F_BOOL	1 = SIMULATION ACTIVE	0
	VALUE	BOOL	OUTPUT VALUE	0
	QUALITY	BYTE	QUALITY CODE OF PROCESS VALUE	0
	ACK_REQ	BOOL	ACKNOWLEDGEMENT REQUEST	0
	IPAR_OK	F_BOOL	1 = NEW I-PARAMETER VALUES ASSIGNED	0

*) Input ADR_CODE is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input ADR_CODE is indicated as changed if changes have been made to the address or the symbolic name of the signal at output VALUE.

Addressing

You must interconnect the symbol for the output value of data type BOOL generated with *HW Config* in the symbol table with the VALUE output.

Normal value

The process data pending at input I are written to the fail-safe DP standard slave. The quality code (QUALITY) is set to 16#80.

Simulation

Instead of the process data pending at input I, a simulation value can also be written to the fail-safe DP standard slave.

If input SIM_ON = 1 and SIM_MOD = 0, the value of input SIM_I is written to the fail-safe DP standard slave and output at output VALUE, provided there is no communication error (PROFIsafe), no module or channel faults (such as a wire break), and no F-Startup. The quality code (QUALITY) is set to 16#60.

If input SIM_ON = 1 and SIM_MOD = 1, the value of input SIM_I is also output at output VALUE in the event of a communication error (PROFIsafe), a module or channel fault (such as a wire break), or an F-Startup in order to be able to simulate "error-free" operation even in the absence of an actual fail-safe DP standard slave.

In both cases, the quality code (QUALITY) is set to 16#60, and QSIM = 1 is set.

Note

If you have placed more than one fail-safe channel driver for outputs for a fail-safe DP standard slave, a simulation value is not written if input PASS_ON of another F-Channel driver for outputs of the fail-safe DP standard slave is "1" and input SIM_ON is "0".

Fail-safe value

The fail-safe value "0" is written to the fail-safe DP standard slave in the following cases:

- If a communication error occurs (PROFIsafe)
- If a module or channel fault occurs (such as a wire break)
- During an F-Startup
- If a passivation with PASS_ON = 1 occurs

The quality code (QUALITY) is set to 16#48, and QBAD = 1 is set.

If the output of the fail-safe value is not caused by a passivation, PASS_OUT = 1 is set additionally in order to passivate other channels.

Note

Channel-specific passivation via PASS_ON is not possible for outputs of fail-safe DP standard slaves. If you have placed more than one fail-safe channel driver for outputs for a fail-safe DP standard slave, the fail-safe value "0" is written for all outputs of the fail-safe DP standard slave when passivation occurs with PASS_ON = 1 at one of the fail-safe channel drivers. If you want to evaluate outputs QBAD and QUALITY of the other F-Channel drivers when PASS_ON = 1 is set at one of the F-Channel drivers, you must activate the PASS_ON inputs of all F-Channel drivers synchronously.

Reassignment of parameters of a fail-safe DP standard slave

Input IPAR_EN and output IPAR_OK are available for reassignment of parameters of a fail-safe DP standard slave.

Input IPAR_EN corresponds to variable iPar_EN_C and output IPAR_OK corresponds to variable iPar_OK_S in the PROFIsafe bus profile (PROFIsafe Specification V1.30 and higher). If you must set or reset input IPAR_EN for reassignment of parameters of a fail-safe DP standard slave or to find out how to evaluate the IPAR_OK output, refer to PROFIsafe Specification V1.30 and higher or the documentation for the fail-safe DP standard slave.

If more than one fail-safe channel driver is placed for a fail-safe DP standard slave, iPar_EN_C is formed from an OR logic operation of all IPAR_EN inputs of the F-Channel drivers belonging to the fail-safe DP standard slave.

If passivation should occur when IPAR_EN = 1, you must also set variable PASS_ON = 1.

Reintegration after error elimination

After an error has been eliminated, the fail-safe DP standard slave can be reintegrated automatically or not until after a user acknowledgement. If ACK_NEC = 1 is assigned, a user acknowledgement at input ACK_REI is required after an error has been eliminated. If ACK_NEC = 0 is assigned, automatic reintegration is carried out.

Output ACK_REQ = 1 signals that the error has been eliminated and that a user acknowledgment at input ACK_REI is necessary for reintegration.

A user acknowledgement is not required for reintegration after PASS_ON = 1. A user acknowledgement is not required for reintegration after F-Startup following a CPU-STOP if the fail-safe DP standard slave starts up by means of the "System start" slave state (20) in accordance with PROFIsafe Specification V1.30 and higher. Otherwise, a communication error (PROFIsafe) is detected.

Note

Channel-specific reintegration is not possible for outputs of fail-safe DP standard slaves. If you place more than one F-Channel driver for outputs for a fail-safe DP standard slave, you must activate the ACK_REI inputs of all F-channel drivers for outputs of the fail-safe DP standard slave synchronously.

⚠ WARNING

Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.

Communication errors (PROFIsafe) always have to be acknowledged at input ACK_REI irrespective of ACK_NEC. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

⚠WARNING

Startup protection for short-term power failure of the fail-safe DP standard slave

Following a power failure of the fail-safe DP standard slave lasting less than the F-Monitoring time for the fail-safe DP standard slaves specified in *HW Config* (see section entitled "Run times, F-Monitoring times, and response times (Page 410)"), automatic reintegration can occur regardless of your setting for input ACK_NEC, as described for the case when ACK_NEC = 0.

If for this case, automatic reintegration is not permissible for the relevant process, you must program startup protection by evaluating the QBAD or PASS_OUT outputs.

In the event of a power failure of the fail-safe DP standard slave lasting longer than the F-Monitoring time for the fail-safe DP standard slave specified in *HW Config*, the F-System detects a communication error.

Startup characteristics

After an F-Startup, communication first has to be established between the F-Module driver and the fail-safe DP standard slave. During this time the fail-safe value "0" is written to the fail-safe DP standard slave. The quality code (QUALITY) is set to 16#48 and outputs QBAD = 1 and PASS_OUT = 1 are set.

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

Behavior during F-STOP

In the event of an F-STOP, quality code 16#00 is output at the QUALITY output and QBAD.DATA = 1 is set. All other variables are frozen.

A.2.6.3 F_PA_AI: Fail-safe channel driver for fail-safe "Transmitter" PA field device

Function

This block is used for signal processing of an analog input value from a fail-safe slot (F-slot) of a "Transmitter" fail-safe PA field device.

The F-Block cyclically reads the process data with status byte (quality code) of the fail-safe PA field device addressed at input VALUE from the associated F-Module driver that communicates with the F-slot of a fail-safe PA field device by means of a safety message frame in accordance with the PROFIsafe bus profile. The fail-safe module driver is automatically placed and interconnected with the *CFC* function "Generate module drivers".

If the process data representing the physical quantity is valid, this value is made available at output V. The status byte (quality code) is made available at the STATUS output and contains information about the state of the fail-safe PA field device.

A quality code (QUALITY output) is generated for the result value of output V. The quantity code can assume the following states:

State	Quality code (QUALITY output)
Valid value	16#80
Simulation	16#60
Fail-safe value	16#48
LAST VALID VALUE	16#44
Unsure, device-specific	16#68
Unsure, process-specific	16#78
Unsure, device-specific, range violation	16#54
Maintenance request	16#A4
Invalid value (F-STOP)	16#00

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ADR_CODE	DWORD	CODE FOR VALUE INTERCONNECTION	Automatically initialized *
	VALUE	REAL	INPUT VALUE	0
	SIM_V	F_REAL	SIMULATION VALUE	0.0
	SIM_ON	F_BOOL	1 = ACTIVATE SIMULATION	0
	SUBS_V	F_REAL	SUBSTITUTION VALUE	0.0
	SUBS_ON	F_BOOL	1 = ENABLE FAILURE SUBSTITUTION	0
	PASS_ON	F_BOOL	1 = ACTIVATE PASSIVATION	0
	ACK_NEC	F_BOOL	1 = ACKNOWLEDGEMENT NECESSARY	0
	ACK_REI	F_BOOL	ACKNOWLEDGMENT REINTEGRATION	0
	IPAR_EN	F_BOOL	1 = ENABLE I-PARAMETER ASSIGNMENT	0
Outputs:	PASS_OUT	F_BOOL	1 = PASSIVATION BECAUSE OF ERROR	0
	QBAD	F_BOOL	1 = PROCESS VALUE INVALID	0
	QSIM	F_BOOL	1 = SIMULATION ACTIVE	0
	QSUBS	F_BOOL	1 = FAILURE SUBSTITUTION ACTIVE	0
	V	F_REAL	PROCESS VALUE	0.0
	V_DATA	REAL	PROCESS VALUE DATA (for monitoring)	0.0
	QUALITY	BYTE	QUALITY CODE OF PROCESS VALUE	0
	STATUS	BYTE	PROCESS VALUE STATUS	0
	V_MOD	REAL	VALUE FROM MODULE	0.0
	ACK_REQ	BOOL	ACKNOWLEDGEMENT REQUEST	0
	IPAR_OK	F_BOOL	1 = NEW I-PARAMETER VALUES ASSIGNED	0

*) Input ADR_CODE is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input ADR_CODE is indicated as changed if changes have been made to the address or the symbolic name of the signal at input VALUE.

Addressing

You must interconnect the symbol for the analog input channel generated with *HW Config* in the symbol table with the VALUE input.

Normal value

If the analog input value received from the fail-safe PA field device is valid, this value is output at output V. The quality code (QUALITY) is set to 16#80, 16#54, 16#60, 16#68, 16#78 or 16#A4, depending on the quality code received from the fail-safe PA field device.

Simulation

A simulation value can be output at output V instead of the normal value that is received from the fail-safe PA field device.

The value of input SIM_V with quality code (QUALITY) 16#60 is output when input SIM_ON = 1. Simulation has the highest priority. QBAD and QSUBS are always set = 0 . QSIM = 1 is set if the F-Block is in simulation state due to SIM_ON = 1.

Note

Quality code (QUALITY) 16#60 is also output if a simulation was started on the fail-safe PA field device and if there is no event for the output of a fail-safe value or last valid value.

If simulation is enabled, the analog input value received from the fail-safe PA field device is output at output V_MOD. If communication with the fail-safe PA field device is not possible or if a user acknowledgement has not yet occurred following an error, "0.0" is output.

V_DATA is output if simulation is disabled.

Fail-safe value

If input SUBS_ON = 1, the fail-safe value SUBS_V is output at output V in the following cases:

- The analog input value is invalid due to a communication error (PROFIsafe).
- The analog input value is invalid due to a module error or a fail-safe value is received by the module.
- A passivation exists with PASS_ON = 1.
- An F-Startup is pending.
- FV_ACTIVATED is signaled by the module.

The quality code (QUALITY) is set to 16#48 and QSUBS = 1 and QBAD = 1 are set.

If the output of the fail-safe value is not caused by a passivation, PASS_OUT = 1 is set additionally in order to passivate other channels.

Keep last value

If input SUBS_ON = 0, the last valid value of V is output at output V in the following cases:

- The analog input value is invalid due to a communication error (PROFIsafe).
- The analog input value is invalid due to a module error or a fail-safe value is received by the module.
- A passivation exists with PASS_ON = 1.
- FV_ACTIVATED is signaled by the module.

The quality code (QUALITY) is set to 16#44 and QSUBS = 0 and QBAD = 1 are set.

If the output of the last valid value is not caused by a passivation, output PASS_OUT = 1 is set additionally in order to passivate other channels.

Reassignment of parameters of a fail-safe PA field device

Input IPAR_EN and output IPAR_OK are available for reassignment of parameters of a fail-safe PA field device.

Input IPAR_EN corresponds to variable IPar_EN_C and output IPAR_OK corresponds to variable IPar_OK_S in the PROFIsafe bus profile (PROFIsafe Specification V1.30 and higher). If you have to set or reset input IPAR_EN for reassignment of parameters of a fail-safe PA field device or to find out how to evaluate the IPAR_OK output, refer to PROFIsafe Specification V1.30 and higher or the documentation for the fail-safe PA field device.

If more than one F-Channel driver is placed for an F-slot of a fail-safe PA field device, IPar_EN_C is formed from an OR logic operation of all the IPAR_EN inputs of the fail-safe channel drivers associated with the F-slot of the fail-safe PA field device.

If passivation should occur when IPAR_EN = 1, you must also set variable PASS_ON = 1.

Reintegration after error elimination

After an error has been eliminated, the analog input value received from the fail-safe PA field device can be reintegrated automatically or not until after a user acknowledgement.
If ACK_NEC = 1 is assigned, a user acknowledgement at input ACK_REI is required after an error has been eliminated. If ACK_NEC = 0 is assigned, automatic reintegration is carried out.

Output ACK_REQ = 1 signals that the error has been eliminated and that a user acknowledgment at input ACK_REI is necessary for reintegration.

A user acknowledgement is not required for reintegration after PASS_ON = 1. A user acknowledgement is not required for reintegration after F-Startup following a CPU-STOP if the fail-safe PA field device starts up by means of the "System start" slave state (20) in accordance with PROFIsafe Specification V1.30 and higher. Otherwise, a communication error (PROFIsafe) is detected.

⚠ WARNING

Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.

Communication errors (PROFIsafe) always have to be acknowledged at input ACK_REI irrespective of ACK_NEC. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

⚠ WARNING

Startup protection for short-term power failure of the fail-safe PA field device

Following a power failure of the fail-safe PA field device lasting less than the F-Monitoring time for the fail-safe PA field device specified in *HW Config* (see section entitled "Run times, F-Monitoring times, and response times (Page 410)"), automatic reintegration can occur regardless of your setting for input ACK_NEC, as described for the case when ACK_NEC = 0.

If for this case, automatic reintegration is not permissible for the relevant process, you must program startup protection by evaluating the QBAD or PASS_OUT outputs.

In the event of a power failure of the fail-safe PA field device lasting longer than the F-Monitoring time for the fail-safe PA field device specified in *HW Config*, the F-System detects a communication error.

Startup characteristics

After an F-Startup, communication first has to be established between the F-Module driver and the fail-safe PA field device. During this time the fail-safe value SUBS_V with quality code (QUALITY code 16#48) is output irrespective of the parameter assignment at input SUBS_ON and outputs QSUBS = 1, QBAD = 1 and PASS_OUT = 1 are additionally set.

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

Behavior during F-STOP

In the event of an F-STOP, quality code 16#00 is output at the QUALITY and STATUS outputs and QBAD.DATA = 1 is set. All other variables are frozen.

A.2.6.4 F_PA_DI: Fail-safe channel driver for fail-safe "Discrete Input" PA field device

Function

This F-Block is used for signal processing of a digital input value from a fail-safe slot (F-slot) of a "Discrete Input" fail-safe PA field device.

The F-Block cyclically reads the process data with status byte (quality code) of the fail-safe PA field device addressed at input I_OUT_D from the associated F-Module driver that communicates with the F-slot of a fail-safe PA field device by means of a safety message frame in accordance with the PROFIsafe bus profile. The fail-safe module driver is automatically placed and interconnected with the *CFC* function "Generate module drivers".

If the process data are valid, the bit (0 to 7) assigned at input BIT_NR is made available by the process data (byte) at output Q. The status byte (quality code) is made available at the STATUS output and contains information about the state of the fail-safe PA field device.

A quality code (QUALITY output) is generated for the result value at output Q. The quality code can assume the following states:

State	Quality code (QUALITY output)
Valid value	16#80
Simulation	16#60
Fail-safe value	16#48
Unsure, device-specific	16#68
Unsure, process-specific	16#78
Unsure, device-specific, range violation	16#54
Maintenance request	16#A4
Invalid value (F-STOP)	16#00

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ADR_CODE	DWORD	CODE FOR I_OUT_D INTERCONNECTION	Automatically initialized *
	BIT_NR	F_INT	REQUIRED BIT NUMBER 0 to 7	0
	I_OUT_D	BYTE	INPUT VALUE	0
	SIM_I	F_BOOL	SIMULATION VALUE	0
	SIM_ON	F_BOOL	1 = ACTIVATE SIMULATION	0
	PASS_ON	F_BOOL	1 = ACTIVATE PASSIVATION	0
	ACK_NEC	F_BOOL	1 = ACKNOWLEDGEMENT NECESSARY	0
	ACK_REI	F_BOOL	ACKNOWLEDGMENT REINTEGRATION	0
	IPAR_EN	F_BOOL	1 = ENABLE I-PARAMETER ASSIGNMENT	0
Outputs:	PASS_OUT	F_BOOL	1 = PASSIVATION BECAUSE OF ERROR	0
	QBAD	F_BOOL	1 = PROCESS VALUE INVALID	0
	QSIM	F_BOOL	1 = SIMULATION ACTIVE	0
	Q	F_BOOL	PROCESS VALUE	0
	QN	F_BOOL	NEGATED PROCESS VALUE	1
	Q_DATA	BOOL	PROCESS VALUE DATA (for monitoring)	0
	QUALITY	BYTE	QUALITY CODE OF PROCESS VALUE	0
	STATUS	BYTE	PROCESS VALUE STATUS	0
	Q_MOD	BOOL	VALUE FROM MODULE	0
	Q0	BOOL	PROCESS VALUE BIT 0	0
	...		...	
	Q7	BOOL	PROCESS VALUE BIT 7	0
	ACK_REQ	BOOL	ACKNOWLEDGEMENT REQUEST	0
	IPAR_OK	F_BOOL	1 = NEW I-PARAMETER VALUES ASSIGNED	0

*) Input ADR_CODE is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input ADR_CODE is indicated as changed if changes have been made to the address or the symbolic name of the signal at input I_OUT_D.

Addressing

You must interconnect the symbol for the process data generated with *HW Config* in the symbol table with the I_OUT_D input.

Note

If the symbol for the process data generated with *HW Config* in the symbol table is of data type "BOOL" and not "BYTE", you must add a symbol with data type BYTE in the symbol table.

Normal value

If the digital input value received from the fail-safe PA field device is valid, this value is output at output Q. The quality code (QUALITY) is set to 16#80, 16#54, 16#60, 16#68, 16#78 or 16#A4, depending on the quality code received from the fail-safe PA field device.

Simulation

A simulation value can be output at output Q instead of the normal value that is received from the fail-safe PA field device.

The value of input SIM_I with quality code (QUALITY) 16#60 is output when input SIM_ON = 1. Simulation has the highest priority. QBAD = 0 is always set. QSIM = 1 is set if the F-Block is in simulation state due to SIM_ON = 1.

Note

Quality code (QUALITY) 16#60 is also output if a simulation was started on the fail-safe PA field device and if there is no event for the output of a fail-safe value.

If simulation is enabled, the digital input value received from the fail-safe PA field device is output at output Q_MOD. If communication with the fail-safe PA field device is not possible or if a user acknowledgement has not yet occurred following an error, "0" is output.

Q_DATA is output if simulation is disabled.

Fail-safe value

In the following cases, the fail-safe value "0" is output at output Q:

- The digital input value is invalid due to a communication error (PROFIsafe).
- The digital input value is invalid due to a module error or a fail-safe value is received by the module.
- A passivation exists with PASS_ON = 1.
- An F-Startup is pending.
- FV_ACTIVATED is signaled by the module.

The quality code (QUALITY) is set to 16#48, and QBAD = 1 is set.

If the output of the fail-safe value is not caused by a passivation, PASS_OUT = 1 is set additionally in order to passivate other channels.

Reassignment of parameters of a fail-safe PA field device

Input IPAR_EN and output IPAR_OK are available for reassignment of parameters of a fail-safe PA field device.

Input IPAR_EN corresponds to variable iPar_EN_C and output IPAR_OK corresponds to variable iPar_OK_S in the PROFIsafe bus profile (PROFIsafe Specification V1.30 and higher). If you must set or reset input IPAR_EN for reassignment of parameters of a fail-safe PA field device or to find out how to evaluate the IPAR_OK output, refer to PROFIsafe Specification V1.30 and higher or the documentation for the fail-safe PA field device.

If more than one F-Channel driver is placed for an F-slot of a fail-safe PA field device, iPar_EN_C is formed from an OR logic operation of all the IPAR_EN inputs of the fail-safe channel drivers associated with the F-slot of the fail-safe PA field device.

If passivation should occur when IPAR_EN = 1, you must also set variable PASS_ON = 1.

Reintegration after error elimination

After an error has been eliminated, the digital input value received from the fail-safe PA field device can be reintegrated automatically or not until after a user acknowledgement.
If ACK_NEC = 1 is assigned, a user acknowledgement at input ACK_REI is required after an error has been eliminated. If ACK_NEC = 0 is assigned, automatic reintegration is carried out.

Output ACK_REQ = 1 signals that the error has been eliminated and that a user acknowledgment at input ACK_REI is necessary for reintegration.

A user acknowledgement is not required for reintegration after PASS_ON = 1. A user acknowledgement is not required for reintegration after F-Startup following a CPU-STOP if the F-I/O start up by means of the "System start" slave state (20) in accordance with PROFIsafe Specification V1.30 and higher. Otherwise, a communication error (PROFIsafe) is detected.

⚠ WARNING

Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.

Communication errors (PROFIsafe) always have to be acknowledged at input ACK_REI irrespective of ACK_NEC. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

⚠ WARNING

Startup protection for short-term power failure of the fail-safe PA field device

Following a power failure of the fail-safe PA field device lasting less than the F-Monitoring time for the fail-safe PA field device specified in *HW Config* (see section entitled "Run times, F-Monitoring times, and response times (Page 410)"), automatic reintegration can occur regardless of your setting for input ACK_NEC, as described for the case when ACK_NEC = 0.

If for this case, automatic reintegration is not permissible for the relevant process, you must program startup protection by evaluating the QBAD or PASS_OUT outputs.

In the event of a power failure of the fail-safe PA field device lasting longer than the F-Monitoring time for the fail-safe PA field device specified in *HW Config*, the F-System detects a communication error.

Startup characteristics

After an F-Startup, communication first has to be established between the F-Module driver and the fail-safe PA field device. During this time, the fail-safe value "0" is output with quality code (QUALITY output) 16#48, and outputs QBAD = 1 and PASS_OUT = 1 are set.

Error handling

- If input BIT_NR is assigned a value <> 0 to 7, the fail-safe value "0" is output at output Q.
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

Behavior during F-STOP

In the event of an F-STOP, quality code 16#00 is output at the QUALITY and STATUS outputs and QBAD.DATA = 1 is set. All other variables are frozen.

See also

Configuring fail-safe PA field devices (Page 58)

A.2.6.5 F_CH_DI: Fail-safe channel drivers for digital inputs of F-I/O (except fail-safe DP standard slaves)

Function

This F-Block is used for signal processing of a digital input value of an F-I/O (except fail-safe DP standard slaves). It supports channel-selective passivation and redundantly configured F-I/O.

The F-Block cyclically reads the digital input value of an F-I/O addressed at input VALUE from the associated fail-safe-module driver F_PS_12 that communicates with the F-I/O by means of a safety message frame in accordance with the PROFIsafe bus profile. The fail-safe module driver is automatically placed and interconnected with the *CFC* function "Generate module drivers".

If the digital input value is valid, it is made available at output Q.

In the case of redundantly configured F-I/O, the digital input value of the relevant channel of the redundantly configured F-I/O is also read.

A quality code (QUALITY output) is generated for the result value at output Q. The quality code can assume the following states:

State	Quality code (QUALITY output)
Valid value	16#80
Simulation	16#60
Fail-safe value	16#48
Invalid value (F-STOP)	16#00

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ADR_CODE	DWORD	CODE FOR VALUE INTERCONNECTION	Automatically initialized *
	VALUE	BOOL	INPUT VALUE	0
	SIM_I	F_BOOL	SIMULATION VALUE	0
	SIM_ON	F_BOOL	1 = ACTIVATE SIMULATION	0
	PASS_ON	F_BOOL	1 = ACTIVATE PASSIVATION	0
	ACK_NEC	F_BOOL	1 = ACKNOWLEDGEMENT NECESSARY	0
	ACK_REI	F_BOOL	ACKNOWLEDGMENT REINTEGRATION	0
Outputs:	PASS_OUT	F_BOOL	1 = PASSIVATION BECAUSE OF ERROR	0
	QBAD	F_BOOL	1 = PROCESS VALUE INVALID	0
	QSIM	F_BOOL	1 = SIMULATION ACTIVE	0
	Q	F_BOOL	PROCESS VALUE	0
	QN	F_BOOL	NEGATED PROCESS VALUE	1
	Q_DATA	BOOL	PROCESS VALUE DATA (for monitoring)	0
	QUALITY	BYTE	QUALITY CODE OF PROCESS VALUE	0
	Q_MOD	BOOL	VALUE FROM MODULE	0
	ACK_REQ	BOOL	ACKNOWLEDGEMENT REQUEST	0
	DISCF	BOOL	DISCREPANCY ERROR MODULE	0
	DISCF_R	BOOL	DISCREPANCY ERROR REDUNDANCY	0
	QMODF	BOOL	1 = MODULE REMOVED/FAULTY	0
	QMODF_R	BOOL	1 = REDUNDANT MODULE REMOVED/FAULTY	0

*) Input ADR_CODE is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input ADR_CODE is indicated as changed if changes have been made to the address or the symbolic name of the signal at input VALUE.

Addressing

You must interconnect the symbol for the digital input channel generated with *HW Config* in the symbol table with the VALUE input.

Note

Inversion of the VALUE input in the *CFC Editor* has no effect. Use the QN output instead.

Normal value

If the digital input value received from the F-I/O is valid, the value is output at output Q with quality code (QUALITY) 16#80.

Normal value for redundantly configured F-I/O

If both of the digital input values received from redundantly configured F-I/O are valid, the values are ORed and the result is output at output Q with quality code (QUALITY) 16#80. If only one of the digital input values received from the F-I/O is valid, this value is output at output Q with quality code (QUALITY) 16#80.

Simulation

A simulation value can be output at output Q instead of the normal value received from the F-I/O.

The value of input SIM_I with quality code (QUALITY) 16#60 is output when input SIM_ON = 1. Simulation has the highest priority. QBAD = 0 is always set. QSIM = 1 is set if the F-Block is in simulation state.

If simulation is enabled, the digital input value received from the F-I/O is output at output Q_MOD. If communication with the F-I/O is not possible or if a user acknowledgement has not yet occurred following an error, "0" is output. Q_DATA is output if simulation is disabled.

Fail-safe value

In the following cases, the fail-safe value "0" is output at output Q:

- The digital input value is invalid due to a communication error (PROFIsafe).
- The digital input value is invalid due to a module or channel fault (such as a wire break) or a fail-safe value is received by the module.
- For redundantly configured F-I/O: Both digital input values are invalid because of a communication error (PROFIsafe) or a module or channel fault (such as a wire break).
- A passivation exists with PASS_ON = 1.
- An F-Startup is pending.

The quality code (QUALITY) is set to 16#48, and QBAD = 1 is set.

If the output of the fail-safe value is not caused by a passivation, PASS_OUT = 1 is set additionally in order to passivate other channels.

Reintegration after error elimination

After an error has been eliminated, the digital input value received from the F-I/O can be reintegrated automatically or not until after a user acknowledgement.

If ACK_NEC = 1 is assigned, a user acknowledgement at input ACK_REI is required after an error has been eliminated. If ACK_NEC = 0 is assigned, automatic reintegration is carried out.

In the case of redundantly configured F-I/O, a user acknowledgement is also required if the indicated errors occurred only on one F-I/O and, thus, did not trigger a fail-safe value at output Q.

Output ACK_REQ = 1 signals that the error has been eliminated and that a user acknowledgment at input ACK_REI is necessary for reintegration.

No user acknowledgement is required for a reintegration after PASS_ON = 1 or after F-Startup following a CPU-STOP.

⚠ WARNING

Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.

Communication errors (PROFIsafe) always have to be acknowledged at input ACK_REI irrespective of ACK_NEC. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

⚠ WARNING

Startup protection for short-term power failure of the F-I/O

Following a power failure of the F-I/O lasting less than the F-Monitoring time for the F-I/O specified in *HW Config* (see section entitled "Run times, F-Monitoring times, and response times (Page 410)"), automatic reintegration can occur regardless of your setting for input ACK_NEC, as described for the case when ACK_NEC = 0.

If for this case, automatic reintegration is not permissible for the relevant process, you must program startup protection by evaluating the QBAD or PASS_OUT outputs.

In the event of a power failure of the F-I/O lasting longer than the F-Monitoring time for the F-I/O specified in *HW Config*, the F-System detects a communication error.

Discrepancy analysis for redundantly configured F-I/O

In the case of redundantly configured F-I/O, the F-Block performs a discrepancy analysis if a discrepancy time <> 0 was configured in *HW Config* during redundancy configuration.

If a discrepancy between the digital input channel addressed at input VALUE and its redundant channel is present that persists for more than the discrepancy time, a discrepancy error is detected. The F-Block sets output DISCF if the digital input channel that was addressed at input VALUE supplies the "0" signal. If the redundant channel supplies the "0" signal, the F-Block sets output DISCF_R. DISCF/DISCF_R is reset as soon as the discrepancy disappears.

For example, the discrepancy analysis enables detection of defective sensors because it is assumed that fail-safe sensors supply a "0" signal when an error occurs. This can increase the availability of the system. Discrepancy errors have no effect on the Q, QBAD or PASS_OUT outputs. The non-fail-safe DISCF/DISCF_R outputs can be read out by means of an OS for purposes of service activities or evaluated in the standard user program.

Startup characteristics

After an F-Startup, communication first has to be established between the F-Module driver and the F-I/O. During this time, the fail-safe value "0" is output with quality code (QUALITY output) 16#48, and outputs QBAD = 1 and PASS_OUT = 1 are set. In the case of redundantly configured F-I/O, the substitute value "0" is output until communication is established with one of the redundant F-I/O.

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

Behavior during F-STOP

In the event of an F-STOP, quality code 16#00 is output at the QUALITY output and QBAD.DATA = 1 is set. All other variables are frozen.

A.2.6.6 F_CH_DO: Fail-safe channel drivers for digital outputs of F-I/O (except fail-safe DP standard slaves)

Function

This F-Block is used for signal processing of a digital output value of an F-I/O (except fail-safe DP standard slaves). It supports channel-selective passivation and redundantly configured F-I/O.

The F-Block cyclically writes the digital output value for the output of an F-I/O addressed at output VALUE to the associated fail-safe-module driver F_PS_12 that communicates with the F-I/O by means of a safety message frame in accordance with the PROFIsafe bus profile. The fail-safe module driver is automatically placed and interconnected with the *CFC* function "Generate module drivers".

In the case of redundantly configured F-I/O, the digital output value is also written to the fail-safe module driver of the redundantly configured F-I/O.

A quality code is generated for the digital output value that is written to the F-I/O. The quality code can assume the following states:

State	Quality code (QUALITY output)
Valid value	16#80
Simulation	16#60
Fail-safe value	16#48
Invalid value (F-STOP)	16#00

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ADR_CODE	DWORD	CODE FOR VALUE INTERCONNECTION	Automatically initialized *
	I	F_BOOL	PROCESS VALUE	0
	SIM_I	F_BOOL	SIMULATION VALUE	0
	SIM_MOD	F_BOOL	1 = SIMULATION HAS PRIORITY	0
	SIM_ON	F_BOOL	1 = ACTIVATE SIMULATION	0
	PASS_ON	F_BOOL	1 = ACTIVATE PASSIVATION	0
	ACK_NEC	F_BOOL	1 = ACKNOWLEDGEMENT NECESSARY	0
	ACK_REI	F_BOOL	ACKNOWLEDGMENT REINTEGRATION	0
Outputs:	PASS_OUT	F_BOOL	1 = PASSIVATION BECAUSE OF ERROR	0
	QBAD	F_BOOL	1 = PROCESS VALUE INVALID	0
	QSIM	F_BOOL	1 = SIMULATION ACTIVE	0
	VALUE	BOOL	OUTPUT VALUE.	0
	QUALITY	BYTE	QUALITY CODE OF PROCESS VALUE	0
	ACK_REQ	BOOL	ACKNOWLEDGEMENT REQUEST	0
	QMODF	BOOL	1 = MODULE REMOVED/FAULTY	0
	QMODF_R	BOOL	1 = REDUNDANT MODULE REMOVED/FAULTY	0

*) Input ADR_CODE is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input ADR_CODE is indicated as changed if changes have been made to the address or the symbolic name of the signal at output VALUE.

Addressing

You must interconnect the symbol for the digital output channel generated with *HW Config* in the symbol table with the VALUE output.

Normal value

The process data pending at input I are written to the F-I/O. The quality code (QUALITY) is set to 16#80.

Normal value for redundantly configured F-I/O

For redundantly configured F-I/O, the process data pending at input I are written to both F-I/O provided there is no communication error (PROFIsafe), no module or channel faults (such as a wire break), and no F-Startup for the two F-I/O. If a communication error is pending for an F-I/O (PROFIsafe) or if there is a module or channel fault (such as a wire break) or an F-Startup, the fail-safe value "0" is written to this F-I/O. Quality code (QUALITY) 16#80 is output.

Simulation

Instead of the process data pending at input I, a simulation value can also be written to the F-I/O.

If input SIM_ON = 1 and SIM_MOD = 0, the value of input SIM_I is written to the F-I/O and output at output VALUE, provided there is no communication error (PROFIsafe), no module or channel faults (such as a wire break), and no F-Startup.

If input SIM_ON = 1 and SIM_MOD = 1, the value of input SIM_I is also output at output VALUE in the event of a communication error (PROFIsafe), a module or channel fault (such as a wire break), or an F-Startup in order to be able to simulate "error-free" operation even in the absence of an actual F-I/O.

In both cases, the quality code (QUALITY) is set to 16#60, and QSIM = 1 is set.

Fail-safe value

The fail-safe value "0" is written to the F-I/O in the following cases:

- If a communication error occurs (PROFIsafe)
- If a module or channel fault occurs (such as a wire break)
- During an F-Startup
- For redundantly configured F-I/O: If a communication error (PROFIsafe), a module or channel fault (such as a wire break) or an F-Startup occurs on both F-I/O
- If a passivation with PASS_ON = 1 occurs

The quality code (QUALITY) is set to 16#48, and QBAD = 1 is set.

If the output of the fail-safe value is not caused by a passivation, PASS_OUT = 1 is set additionally in order to passivate other channels.

Reintegration after error elimination

After an error has been eliminated, the F-I/O can be reintegrated automatically or not until after a user acknowledgement.

If ACK_NEC = 1 is assigned, a user acknowledgement at input ACK_REI is required after an error has been eliminated. If ACK_NEC = 0 is assigned, automatic reintegration is carried out.

In the case of redundantly configured F-I/O, a user acknowledgement is also required if the indicated errors occurred only on one F-I/O and, thus, did not trigger a fail-safe value output to the process.

Output ACK_REQ = 1 signals that the error has been eliminated and that a user acknowledgment at input ACK_REI is necessary for reintegration.

No user acknowledgement is required for a reintegration after PASS_ON = 1 or after F-Startup following a CPU-STOP.

⚠ WARNING

Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.

Communication errors (PROFIsafe) always have to be acknowledged at input ACK_REI irrespective of ACK_NEC. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

⚠ WARNING

Startup protection for short-term power failure of the F-I/O

Following a power failure of the F-I/O lasting less than the F-Monitoring time for the F-I/O specified in *HW Config* (see section entitled "Run times, F-Monitoring times, and response times (Page 410)"), automatic reintegration can occur regardless of your setting for input ACK_NEC, as described for the case when ACK_NEC = 0.

If for this case, automatic reintegration is not permissible for the relevant process, you must program startup protection by evaluating the QBAD or PASS_OUT outputs.

In the event of a power failure of the F-I/O lasting longer than the F-Monitoring time for the F-I/O specified in *HW Config*, the F-System detects a communication error.

Startup characteristics

After an F-Startup, communication first has to be established between the F-Module driver and the F-I/O. The fail-safe value "0" is written to the F-I/O during this time. The quality code (QUALITY) is set to 16#48 and outputs QBAD = 1 and PASS_OUT = 1 are set. In the case of redundantly configured F-I/O, the quality code (QUALITY) is set to 16#80 and outputs QBAD = 0 and PASS_OUT = 0 are set as soon as communication is established with an F-I/O.

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

Behavior during F-STOP

In the event of an F-STOP, quality code 16#00 is output at the QUALITY output and QBAD.DATA = 1 is set. All other variables are frozen.

A.2.6.7 F_CH_AI: Fail-safe channel drivers for analog inputs of F-I/O (except fail-safe DP standard slaves)

Function

This F-Block is used for signal processing of an analog input value of an F-I/O (except fail-safe DP standard slaves). It supports channel-selective passivation and redundantly configured I/O.

The F-Block cyclically reads the analog input value (raw value) of an F-I/O addressed at input VALUE from the associated fail-safe-module driver F_PS_12 that communicates with the F-I/O by means of a safety message frame in accordance with the PROFIsafe bus profile. The fail-safe module driver is automatically placed and interconnected with the *CFC* function "Generate module drivers".

If the analog input value is valid, it is adjusted to its physical quantity and made available at output V.

In the case of redundantly configured F-I/O, the analog input value of the relevant channel of the redundantly configured F-I/O is also read.

A quality code (QUALITY output) is generated for the result value at output V. The quality code can assume the following states:

State	Quality code (QUALITY output)
Valid value	16#80
Simulation	16#60
Fail-safe value	16#48
LAST VALID VALUE	16#44
Invalid value (F-STOP)	16#00

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ADR_CODE	DWORD	CODE FOR VALUE INTERCONNECTION	Automatically initialized *
	MODE	F_WORD	MEASURING RANGE CODING	Automatically initialized *
	VALUE	WORD	INPUT VALUE	0
	VHRANGE	F_REAL	HIGH RANGE OF PROCESS VALUE	0.0
	VLRANGE	F_REAL	LOW RANGE OF PROCESS VALUE	0.0
	CH_F_ON	F_BOOL	1 = ENABLE LIMIT VALUE MONITORING	0
	CH_F_HL	F_REAL	OVERRANGE LIMIT OF INPUT VALUE (mA)	0.0
	CH_F_LL	F_REAL	UNDERRANGE LIMIT OF INPUT VALUE (mA)	0.0
	SIM_V	F_REAL	SIMULATION VALUE	0.0
	SIM_ON	F_BOOL	1 = ACTIVATE SIMULATION	0
	SUBS_V	F_REAL	SUBSTITUTION VALUE	0.0
	SUBS_ON	F_BOOL	1 = ENABLE FAILURE SUBSTITUTION	0
	PASS_ON	F_BOOL	1 = ACTIVATE PASSIVATION	0
	ACK_NEC	F_BOOL	1 = ACKNOWLEDGEMENT NECESSARY	0
	ACK_REI	F_BOOL	ACKNOWLEDGMENT REINTEGRATION	0
	IPAR_EN **	F_BOOL	1 = ENABLE I-PARAMETER ASSIGNMENT	0
	IPAR_ENR **	F_BOOL	1 = ENABLE I-PARAMETER ASSIGNMENT (redundant module)	0
Outputs:	PASS_OUT	F_BOOL	1 = PASSIVATION BECAUSE OF ERROR	0
	QCHF_HL	F_BOOL	1 = INPUT VALUE HIGH LIMIT FAILURE	0
	QCHF_LL	F_BOOL	1 = INPUT VALUE LOW LIMIT FAILURE	0
	QBAD	F_BOOL	1 = PROCESS VALUE INVALID	0
	QSIM	F_BOOL	1 = SIMULATION ACTIVE	0
	QSUBS	F_BOOL	1 = FAILURE SUBSTITUTION ACTIVE	0
	OVHRANGE	F_REAL	HIGH RANGE OF PROCESS VALUE (COPY)	0.0
	OVLRANGE	F_REAL	LOW RANGE OF PROCESS VALUE (COPY)	0.0
	V	F_REAL	PROCESS VALUE	0.0
	V_DATA	REAL	PROCESS VALUE DATA (for monitoring)	0.0
	QUALITY	BYTE	QUALITY CODE OF PROCESS VALUE	0

	Name	Data type	Description	Default
	V_MOD	REAL	VALUE FROM MODULE	0.0
	ACK_REQ	BOOL	ACKNOWLEDGEMENT REQUEST	0
	IPAR_OK **	F_BOOL	1 = NEW I-PARAMETER VALUES ASSIGNED	
	IPAR_OKR **	F_BOOL	1 = NEW I-PARAMETER VALUES ASSIGNED (redundant module)	
	QMODF	BOOL	1 = MODULE REMOVED/FAULTY	0
	QMODF_R	BOOL	1 = REDUNDANT MODULE REMOVED/FAULTY	0
	AL_STATE	STRUCT	ALARM STATUS	
	RAW_VALUE	WORD	RAW VALUE	0
	OVHRANGE	REAL	COPY OF OVHRANGE	0.0
	OVLRANGE	REAL	COPY OF OVLRANGE	0.0
	PASS_ON	BOOL	COPY OF PASS_ON	0
	PASS_OUT	BOOL	COPY OF PASS_OUT	0
	QCHF_HL	BOOL	COPY OF QCHF_HL	0
	QCHF_LL	BOOL	COPY OF QCHF_LL	0
	QBAD	BOOL	COPY OF QBAD	0
	QSIM	BOOL	COPY OF QSIM	0
	QSUBS	BOOL	COPY OF QSUBS	0
	ACK_REQ	BOOL	COPY OF ACK_REQ	0
	V_DATA	REAL	COPY OF V_DATA	0.0
	QUALITY	BYTE	COPY OF QUALITY	0
	V_MOD	REAL	COPY OF V_MOD	0.0

*) The ADR_CODE and MODE inputs are automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input ADR_CODE is indicated as changed if changes have been made to the address or the symbolic name of the signal at input VALUE. The MODE input is indicated as changed if changes are made during configuration of the F-I/O.

**) These inputs/outputs are not visible. If you are using this F-Channel driver with an SM 336; F-AI 6 x 0/4 to 20 mA HART, you are permitted to make these inputs/outputs visible and use them.

Addressing

You must interconnect the symbol for the analog input channel generated with *HW Config* in the symbol table with the VALUE input.

Raw value checking

Depending on the measurement type and measurement range, there is a nominal range of the F-I/O, in which the analog signal is converted to a digitized raw value. To this end, there is an overrange and an underrange in which the analog signal can still be converted. Overflow and underflow apply beyond these limits. The fail-safe channel driver indicates whether the raw value lies within the nominal range of the F-I/O with analog inputs.

- If the value falls below the nominal range, output parameter QCHF_LL = 1 is set.
- If the value exceeds the nominal range, output parameter QCHF_HL = 1 is set.

In the case of overflow or underflow, output QBAD = 1 is also set, and, depending on the parameter assignment at input SUBS_ON, the fail-safe value SUBS_V or the last valid value is output.

In the event of channel faults (such as a wire break), 16#7FFF (overflow) or 16#8000 (underflow) is output as a raw value by the F-I/O with analog inputs. Accordingly, the fail-safe channel driver detects an overflow or underflow and sets outputs QCHF_HL or QCHF_LL = 1 and QBAD = 1.

(NAMUR) limit value check in the 4 to 20 mA measuring range

In the NAMUR guidelines for analog signal processing, limit values are defined for life zero (4 to 20 mA) analog signals where there is a channel fault:

3.6 mA < analog signal < 21 mA.

By default, the above NAMUR limits are permanently set for limit value checking. If other channel fault limits are to be set, input CH_F_ON = 1 must be set and inputs CH_F_HL and CH_F_LL must be set in mA with corresponding new limit values.

CH_F_LL < analog signal < CH_F_HL

In the case of overflow or underflow of the active channel fault limits, output QBAD = 1 is also set and, depending on the parameter assignment at input SUBS_ON, the fail-safe value SUBS_V or the last valid value is output.

Note

The selectable limit values must be below the upper limit of the overrange and above the lower limit of the underrange of the F-I/O with analog inputs. Values outside the NAMUR range are thus also possible, unless the F-I/O with analog inputs automatically limits the measured values.

Normal value

If the raw value received from the F-I/O is valid, it is adjusted to its physical quantity on the basis of the VLRANGE and VHRANGE inputs and the measuring range coding and output with quality code (QUALITY) = 16#80 at output V.

To enable the settings for VLRANGE and VHRANGE to be interconnected to other block parameters, these are written to the OVLRANGE and OVHRANGE outputs.

The conversion algorithm assumes a linear input signal.

When VLRANGE = 0.0 and VHRANGE = 100.0, a percentage value is output.

If VHRANGE = VLRANGE is set, the input signal of the F-I/O with analog inputs is output according to the measuring range coding (such as a mA value).

Assignment of VHRANGE < VLRANGE is not allowed and causes invalid outputs.

Measuring range coding of the F-I/O with analog inputs

The measuring range is coded in *HW Config* by assigning the "Measuring range" and, if necessary, "Measurement type" parameters. It is automatically transferred to the MODE parameter of the fail-safe channel driver on compilation. The F-Channel driver supports the following measuring range codings:

Measurement type	Measuring range	MODE (decimal/hex.)
4-wire measuring transducer	0 to 20 mA	514 / 16#0202
or measurement type irrelevant	4 to 20 mA	515 / 16#0203
2-wire measuring transducer	4 to 20 mA	771 / 16#0303

Normal value for redundantly configured F-I/O

In the case of redundantly configured F-I/O, the raw value of the F-I/O that first supplies a valid value after an F-Startup or initial run is output at output V after adjustment to its physical quantity with quality code (QUALITY) = 16#80. The analog input value of the redundantly configured F-I/O is switched to when the currently output analog input value becomes invalid.

Simulation

A simulation value can be output at output V instead of the normal value received from the F-I/O.

The value of input SIM_V with quality code (QUALITY) 16#60 is output when input SIM_ON = 1. Simulation has the highest priority. QSIM = 1 is set if the F-Block is in simulation state.

When VLRANGE = 0.0 and VHRANGE = 100.0, the value at input SIM_V must be a percentage value.

In order to also be able to simulate the states of outputs QCHF_LL and QCHF_HL, the simulation value is converted to a raw value on the basis of the VHRANGE and VLRANGE inputs and the measuring range coding and is checked like a raw value received from the F-I/O.

If there is an overflow/underflow or violation of the active channel fault limits (for measuring range of 4 to 20 mA), the fail-safe value SUBS_V or the last valid value, depending on the parameter assignment at input SUBS_ON, and not simulation value SIM_V is output at output V with quality code (QUALITY) 16#60. QBAD = 1 is set.

If simulation is enabled, the analog input value received from the F-I/O is output as process data at output V_MOD. If communication with the F-I/O is not possible or if a user acknowledgement has not yet occurred following an error, "0.0" is output.

V_DATA is output if simulation is disabled.

Fail-safe value

If input SUBS_ON = 1, the fail-safe value SUBS_V is output at output V in the following cases:

- The analog input value is invalid due to a communication error (PROFIsafe).
- The analog input value is invalid due to a module or channel fault (such as a wire break) or a fail-safe value is received by the module.
- The analog input value is invalid due to an overflow or underflow.
- The analog input value is invalid due to a limit violation of the active channel fault limits (for measuring range 4-20 mA).
- For redundantly configured F-I/O: Both analog input values are invalid due to a communication error (PROFIsafe), a module or channel fault (such as a wire break) or an overflow/underflow or limit violation of the active channel limits (for a measurement range of 4 to 20 mA).
- A passivation exists with PASS_ON = 1.
- An F-Startup is pending.

The quality code (QUALITY) is set to 16#48 and QSUBS = 1 and QBAD = 1 are set.

If the output of the fail-safe value is not caused by a passivation, PASS_OUT = 1 is set additionally in order to passivate other channels.

Keep last value

If input SUBS_ON = 0, the last valid value of V is output at output V in the following cases:

- The analog input value is invalid due to a communication error (PROFIsafe).
- The analog input value is invalid due to a module or channel fault (such as a wire break) or a fail-safe value is received by the module.
- The analog input value is invalid due to an overflow or underflow.
- The analog input value is invalid due to a limit violation of the active channel fault limits (for measuring range 4-20 mA).
- For redundantly configured F-I/O: Both analog input values are invalid due to a communication error (PROFIsafe), a module or channel fault (such as a wire break) or an overflow/underflow or limit violation of the active channel limits (for a measurement range of 4 to 20 mA).
- A passivation exists with PASS_ON = 1.

The quality code (QUALITY) is set to 16#44 and QSUBS = 0 and QBAD = 1 are set.

If the output of the last valid value is not caused by a passivation, the output PASS_OUT = 1 is set additionally in order to passivate other channels.

Reintegration after error elimination

After an error has been eliminated, the analog input value received from the F-I/O can be reintegrated automatically or not until after a user acknowledgement.

If ACK_NEC = 1 is assigned, a user acknowledgement at input ACK_REI is required after an error has been eliminated. If ACK_NEC = 0 is assigned, automatic reintegration is carried out.

In the case of redundantly configured F-I/O, a user acknowledgement is also required if the indicated errors occurred only on one F-I/O and, thus, did not trigger a fail-safe value at output V.

Output ACK_REQ = 1 signals that the error has been eliminated and that a user acknowledgment at input ACK_REI is necessary for reintegration.

No user acknowledgement is required for a reintegration after PASS_ON = 1 or after F-Startup following a CPU-STOP.

⚠ WARNING

Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.

Communication errors (PROFIsafe) always have to be acknowledged at input ACK_REI irrespective of ACK_NEC. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

⚠WARNING

Startup protection for short-term power failure of the F-I/O

Following a power failure of the F-I/O lasting less than the F-Monitoring time for the F-I/O specified in *HW Config* (see section entitled "Run times, F-Monitoring times, and response times (Page 410)"), automatic reintegration can occur regardless of your setting for input ACK_NEC, as described for the case when ACK_NEC = 0.

If for this case, automatic reintegration is not permissible for the relevant process, you must program startup protection by evaluating the QBAD or PASS_OUT outputs.

In the event of a power failure of the F-I/O lasting longer than the F-Monitoring time for the F-I/O specified in *HW Config*, the F-System detects a communication error.

Configurable alarm limits

At output AL_STATE, the raw value and inputs/outputs of the F-Channel driver are also bundled in a structured manner and made available as non-fail-safe information. This enables you to evaluate configurable alarm limits in the standard user program. By mapping onto a structure, information can be exchanged between a fail-safe channel driver and a standard block by means of a single interconnection.

Reassignment of parameters of an F-I/O

Input IPAR_EN and output IPAR_OK are available for reassignment of parameters of an F-I/O. Input IPAR_EN and output IPAR_OK are not visible. When using the SM 336; F-AI 6 x 0/4 to 20 mA HART, make the input and output visible.

The IPAR_EN input corresponds to the iPar_EN_C variable and the IPAR_OK output corresponds to the iPar_OK_S variable in the PROFIsafe bus profile, PROFIsafe Specification V1.30 or later. To find out when to set or reset input IPAR_EN for reassignment of parameters of an F-I/O or how to evaluate the IPAR_OK output, refer to PROFIsafe Specification V1.30 or later or the documentation for the F-I/O.

If more than one F-Channel driver is positioned for an F-I/O, iPar_EN_C is formed from an OR logic operation of all IPAR_EN inputs of the F-Channel drivers belonging to the F-I/O.

If passivation should occur when IPAR_EN = 1, you must also set variable PASS_ON = 1.

The signals IPAR_ENR and IPAR_OKR are used for redundant F-I/O.

The inputs IPAR_EN and IPAR_ENR and the outputs IPAR_OK and IPAR_OKR are not visible. When using the SM 336; F-AI 6 x 0/4 to 20 mA HART, make the input and output visible.

The IPAR_EN input and IPAR_OK output are used with the SM 336; F-AI 6 x 0/4 to 20 mA HART to deactivate the HART protocol. For a detailed description of how to process signals in the safety program, refer to the manual for the SM 336; F-AI 6 x 0/4 to 20 mA HART on the Web (http://support.automation.siemens.com/WW/view/en/19026151).

Startup characteristics

After an F-Startup, communication first has to be established between the F-Module driver and the F-I/O. During this time the fail-safe value SUBS_V with quality code (QUALITY code 16#48) is output irrespective of the parameter assignment at input SUBS_ON and outputs QSUBS = 1, QBAD = 1 and PASS_OUT = 1 are additionally set. In the case of redundantly configured F-I/O, the fail-safe value SUBS_V is output until communication is established with one of the redundant F-I/O.

Error handling

- If there is a non-supported measuring range encoding at input MODE, it is assumed that the raw value is invalid.
- If any of the VHRANGE, VLRANGE, CH_F_HL, CH_F_LL, SIM_V or SUBS_V inputs are invalid floating-point numbers (NaN) or if invalid floating-point numbers (NaN) are produced by the calculation in the F-Block, the fail-safe value SUBS_V or the last valid value, depending on the parameter assignment at input SUBS_ON, is output at output V. Outputs QBAD, QCHF_LL and QCHF_HL are set to 1. The appropriate quality code (QUALITY) and QSUBS are generated.

  If the invalid floating-point numbers (NaN) are the result of the calculation in the F-Block, the following diagnostic event is entered in the diagnostic buffer of the CPU.

  – "Safety program: invalid REAL number in DB" (event ID 16#75D9)
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

  – "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

Behavior during F-STOP

In the event of an F-STOP, quality code 16#00 is output at the QUALITY output and QBAD.DATA = 1 is set. All other variables are frozen.

A.2.6.8 F_CH_II: F-Channel driver for inputs of data type INT of fail-safe DP standard slaves and fail-safe standard I/O devices

Function

This F-Block is used for signal processing of an input value of data type INT of fail-safe DP standard slaves and fail-safe standard I/O devices.

The F-Block cyclically reads the input value of data type INT of a fail-safe DP standard slave addressed at input VALUE from the associated fail-safe module driver F_PS_12 that communicates with the fail-safe DP standard slave by means of a safety message frame in accordance with the PROFIsafe bus profile. The fail-safe module driver is automatically positioned and interconnected with the CFC function "Generate module drivers".

If the input value is valid, it is made available as F_REAL at output V and as an integer at output V_INT.

A quality code (QUALITY output) is generated for the result value at output V. The quality code can assume the following states:

State	Quality code (QUALITY output)
Valid value	16#80
Simulation	16#60
Last valid value	16#44
Fail-safe value	16#48
Invalid value (F-STOP)	16#00

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	ADR_CODE	DWORD	Code for VALUE interconnection	Automatically initialized*
	VALUE	INT	Input channel address	0
	SIM_V	F_INT	Simulation value	0
	SIM_ON	F_BOOL	1= activate simulation value	0
	SUBS_V	F_INT	SUBSTITUTION VALUE	0
	SUBS_ON	F_BOOL	1=enable SUBSTITUTION VALUE	0
	PASS_ON	F_BOOL	1=enable passivation	0
	ACK_NEC	F_BOOL	1=user acknowledgement necessary for reintegration following error	0
	ACK_REI	F_BOOL	Acknowledgement for reintegration	0
	IPAR_EN	F_BOOL	1=enable assignment of I-parameters	0
Outputs:	PASS_OUT	F_BOOL	1=passivation due to error	0
	QBAD	F_BOOL	1=process value invalid	0
	QSIM	F_BOOL	1=simulation active	0
	QSUBS	F_BOOL	1=fail-safe value active	0
	V	F_REAL	Process value	0.0
	V_DATA	REAL	DATA component of process value (for monitoring)	0.0
	QUALITY	BYTE	Quality code of process value	B#16#0
	V_INT	F_INT	Process value INT	0
	V_MOD	REAL	Value from F-I/O	0.0
	ACK_REQ	BOOL	Acknowledgment required for reintegration	0
	IPAR_OK	F_BOOL	1=new I-parameter values have been assigned	0

*) Input ADR_CODE is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input ADR_CODE is indicated as changed if changes have been made to the address or the symbolic name of the signal at input VALUE.

Addressing

You must interconnect the symbol for the input value of data type INT generated with *HW Config* in the symbol table with the VALUE input.

Normal value

If the input value received from the fail-safe DP standard slave is valid, the value is output at output V and V_INT with quality code (QUALITY) 16#80.

Simulation

A simulation value can be output at output V and V_INT instead of the normal value that is received from the fail-safe DP standard slave.

The value of input SIM_V with quality code (QUALITY) 16#60 is output when input SIM_ON = 1. Simulation has the highest priority. QBAD = 0 is always set. QSIM = 1 is set if the block is in simulation state.

If simulation is enabled, the input value received from the fail-safe DP standard slave is output at output V_MOD. If communication with the fail-safe DP standard slave is not possible or if a user acknowledgement has not yet occurred following an error, "0" is output.

V_DATA is output if simulation is disabled.

Fail-safe value

If input SUBS_ON = 1, the fail-safe value SUBS_V is output at output V and V_INT in the following cases:

- The input value is invalid due to a communication error (PROFIsafe).
- The input value is invalid due to a module error or a fail-safe value is received by the module.
- A passivation exists with PASS_ON = 1.
- An F-Startup is pending.
- FV_ACTIVATED is signaled by the module.

The quality code (QUALITY) is set to 16#48 and QSUBS = 1 and QBAD = 1 are set.

If the output of the fail-safe value is not caused by a passivation, PASS_OUT = 1 is set additionally in order to passivate other channels.

Keep last value

If input SUBS_ON = 0, the last valid value of V is output at output V and V_INT in the following cases:

- The input value is invalid due to a communication error (PROFIsafe) or a fail-safe value is received by the module.
- The input value is invalid due to a module error or a fail-safe value is received by the module.
- A passivation exists with PASS_ON = 1.
- FV_ACTIVATED is signaled by the module.

The quality code (QUALITY) is set to 16#44 and QSUBS = 0 and QBAD = 1 are set.

If the output of the last valid value is not caused by a passivation, output PASS_OUT = 1 is set additionally in order to passivate other channels.

Reintegration

After an error has been eliminated, the input value received from the DP standard slave can be reintegrated either automatically or not until after a user acknowledgement.

If ACK_NEC = 1 is assigned, a user acknowledgement at input ACK_REI is required after an error has been eliminated. If ACK_NEC = 0 is assigned, automatic reintegration is carried out.

Output ACK_REQ = 1 signals that the error has been eliminated and that a user acknowledgment at input ACK_REI is necessary for reintegration.

A user acknowledgement is not required for reintegration after PASS_ON = 1. A user acknowledgement is not required for reintegration after F-Startup following a CPU-STOP if the F-I/O start up by means of the "System start" slave state (20) in accordance with PROFIsafe Specification V1.30 or later. Otherwise, a communication error (PROFIsafe) is detected.

⚠WARNING

Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.

Communication errors (PROFIsafe) always have to be acknowledged at input ACK_REI irrespective of ACK_NEC. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

⚠WARNING

Startup protection for short-term power failure of the fail-safe DP standard slave

Following a power failure of the fail-safe DP standard slave lasting less than the F-Monitoring time for the fail-safe DP standard slave specified in *HW Config* (see section entitled "Run times, F-Monitoring times, and response times (Page 410)"), automatic reintegration can occur regardless of your setting for input ACK_NEC, as described for the case when ACK_NEC = 0.

If for this case, automatic reintegration is not permissible for the relevant process, you must program startup protection by evaluating the QBAD or PASS_OUT outputs.

In the event of a power failure of the F-I/O lasting longer than the F-Monitoring time for the F-I/O specified in *HW Config*, the F-System detects a communication error.

Reassignment of parameters of a fail-safe DP standard slave

Input IPAR_EN and output IPAR_OK are available for reassignment of parameters of a fail-safe DP standard slave.

The IPAR_EN input corresponds to the iPar_EN_C variable and the IPAR_OK output corresponds to the iPar_OK_S variable in the PROFIsafe bus profile, PROFIsafe Specification V1.30 or later. To find out when to set or reset input IPAR_EN for reassignment of parameters of a fail-safe DP standard slave or how to evaluate the IPAR_OK output, refer to PROFIsafe Specification V1.30 or later or the documentation for the fail-safe DP standard slave.

If more than one F-Channel driver is positioned for a fail-safe DP standard slave, iPar_EN_C is formed from an OR logic operation of all IPAR_EN inputs of the F-Channel drivers belonging to the fail-safe DP standard slave.

If passivation should occur when IPAR_EN = 1, you must also set variable PASS_ON = 1.

Startup characteristics

After an F-Startup, communication first has to be established between the fail-safe module driver and the fail-safe DP standard slave. During this time the fail-safe value SUBS_V with quality code (QUALITY output) 16#48 is output irrespective of the parameter assignment at input SUBS_ON and outputs QSUBS = 1, QBAD = 1 and PASS_OUT = 1 are additionally set.

Error handling

- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

Behavior during F-STOP

In the event of an F-STOP, quality code 16#00 is output at the QUALITY output and QBAD.DATA = 1 is set. All other variables are frozen.

A.2.6.9 F_CH_IO: F-Channel driver for outputs of data type INT of fail-safe DP standard slaves and fail-safe standard I/O devices

Function

This F-Block is used for signal processing of an output value of data type INT of fail-safe DP standard slaves and fail-safe standard I/O devices.

The F-Block cyclically writes the output value of data type INT for the output of a fail-safe DP standard slave addressed at output VALUE to the associated fail-safe module driver F_PS_12 that communicates with the fail-safe DP standard slave by means of a safety message frame in accordance with the PROFIsafe bus profile. The fail-safe module driver is automatically positioned and interconnected with the CFC function "Generate module drivers".

A quality code is generated for the output value that is written to the fail-safe DP standard slave. The quality code can assume the following states:

State	Quality code (QUALITY output)
Valid value	16#80
Simulation	16#60
Fail-safe value	16#48
Invalid value (F-STOP)	16#00

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ADR_CODE	DWORD	Code for VALUE interconnection	Automatically initialized*
	I	F_INT	Process value	0
	SIM_I	F_INT	Simulation value	0
	SIM_MOD	F_BOOL	1=simulation value has priority	0
	SIM_ON	F_BOOL	1=activate simulation value	0
	PASS_ON	F_BOOL	1=enable passivation	0
	ACK_NEC	F_BOOL	1=user acknowledgement necessary for reintegration following error	0
	ACK_REI	F_BOOL	Acknowledgement for reintegration	0
	IPAR_EN	F_BOOL	1=enable assignment of I-parameters	0
Outputs:	PASS_OUT	F_BOOL	1=passivation due to error	0
	QBAD	F_BOOL	1=process value invalid	0
	QSIM	F_BOOL	1=simulation active	0
	VALUE	INT	Output channel address	0
	QUALITY	BYTE	Quality code of process value	B#16#0
	ACK_REQ	BOOL	Acknowledgment required for reintegration	0
	IPAR_OK	F_BOOL	1=new I-parameter values have been assigned	0

*) Input ADR_CODE is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input ADR_CODE is indicated as changed if changes have been made to the address or the symbolic name of the signal at input VALUE.

Addressing

You must interconnect the symbol for the input value of data type INT generated with *HW Config* in the symbol table with the VALUE input.

Normal value

The process value pending at input I is written to the fail-safe DP standard slave. The quality code (QUALITY) is set to 16#80.

Fail-safe value

The fail-safe value "0" is written to the fail-safe DP standard slave in the following cases:

- If a communication error occurs (PROFIsafe)
- If a module or channel fault occurs (such as a wire break)
- During an F-Startup
- If a passivation with PASS_ON = 1 occurs

The quality code (QUALITY) is set to 16#48, and QBAD = 1 is set.

If the output of the fail-safe value is not caused by a passivation, PASS_OUT = 1 is set additionally in order to passivate other channels.

Note

Channel-specific passivation via PASS_ON is not possible for outputs of fail-safe DP standard slaves. If you have positioned more than one fail-safe channel driver for outputs for a fail-safe DP standard slave, the fail-safe value "0" is written for all outputs of the fail-safe DP standard slave when passivation occurs with PASS_ON = 1 at one of the fail-safe channel drivers. If you want to evaluate outputs QBAD and QUALITY of the other F-Channel drivers when PASS_ON = 1 is set at one of the F-Channel drivers, you must activate the PASS_ON inputs of all F-Channel drivers synchronously.

Simulation

Instead of the process value pending at input I, a simulation value can also be written to the fail-safe DP standard slave.

If input SIM_ON = 1 and SIM_MOD = 0, the value of input SIM_I is written to the fail-safe DP standard slave and output at output VALUE, provided there is no communication error (PROFIsafe), no module or channel faults (such as a wire break), and no F-Startup.

If input SIM_ON = 1 and SIM_MOD = 1, the value of input SIM_I is also output at output VALUE in the event of a communication error (PROFIsafe), a module or channel fault (such as a wire break), or an F-Startup in order to simulate "error-free" operation even in the absence of an actual fail-safe DP standard slave.

In both cases, the quality code (QUALITY) is set to 16#60, and QSIM = 1 is set.

Note

If you have placed more than one fail-safe channel driver for outputs for a fail-safe DP standard slave, a simulation value is not written if input PASS_ON of another F-Channel driver for outputs of the fail-safe DP standard slave is "1" and input SIM_ON is "0".

Reintegration

After an error has been eliminated, the fail-safe DP standard slave can be reintegrated either automatically or not until after a user acknowledgement.

If ACK_NEC = 1 is assigned, a user acknowledgement at input ACK_REI is required after an error has been eliminated. If ACK_NEC = 0 is assigned, automatic reintegration is carried out.

Output ACK_REQ = 1 signals that the error has been eliminated and that a user acknowledgment at input ACK_REI is necessary for reintegration.

A user acknowledgement is not required for reintegration after PASS_ON = 1. A user acknowledgement is not required for reintegration after F-Startup following a CPU-STOP if the fail-safe DP standard slave starts up by means of the "System start" slave state (20) in accordance with PROFIsafe Specification V1.30 or later. Otherwise, a communication error (PROFIsafe) is detected.

Note

Channel-specific reintegration is not possible for outputs of fail-safe DP standard slaves. If you position more than one F-Channel driver for outputs for a fail-safe DP standard slave, you must activate the ACK_REI inputs of all F-channel drivers for outputs of the fail-safe DP standard slave synchronously.

⚠ WARNING

Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.

Communication errors (PROFIsafe) always have to be acknowledged at input ACK_REI irrespective of ACK_NEC. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

⚠ WARNING

Startup protection for short-term power failure of the fail-safe DP standard slave

Following a power failure of the fail-safe DP standard slave lasting less than the F-Monitoring time for the fail-safe DP standard slave specified in *HW Config* (see section entitled "Run times, F-Monitoring times, and response times (Page 410)"), automatic reintegration can occur regardless of your setting for input ACK_NEC, as described for the case when ACK_NEC = 0.

If for this case, automatic reintegration is not permissible for the relevant process, you must program startup protection by evaluating the QBAD or PASS_OUT outputs.

In the event of a power failure of the F-I/O lasting longer than the F-Monitoring time for the F-I/O specified in *HW Config*, the F-System detects a communication error.

Reassignment of parameters of a fail-safe DP standard slave

Input IPAR_EN and output IPAR_OK are available for reassignment of parameters of a fail-safe DP standard slave.

The IPAR_EN input corresponds to the iPar_EN_C variable and the IPAR_OK output corresponds to the iPar_OK_S variable in the PROFIsafe bus profile, PROFIsafe Specification V1.30 or later. To find out when to set or reset input IPAR_EN for reassignment of parameters of a fail-safe DP standard slave or how to evaluate the IPAR_OK output, refer to PROFIsafe Specification V1.30 or later or the documentation for the fail-safe DP standard slave.

If more than one F-Channel driver is positioned for a fail-safe DP standard slave, iPar_EN_C is formed from an OR logic operation of all IPAR_EN inputs of the F-Channel drivers belonging to the fail-safe DP standard slave.

If passivation should occur when IPAR_EN = 1, you must also set variable PASS_ON = 1.

Startup characteristics

After an F-Startup, communication first has to be established between the fail-safe module driver and the fail-safe DP standard slave. During this time the fail-safe value "0" is written to the fail-safe DP standard slave. The quality code (QUALITY) is set to 16#48 and outputs QBAD = 1 and PASS_OUT = 1 are set.

Error handling

- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

Behavior during F-STOP

In the event of an F-STOP, quality code 16#00 is output at the QUALITY output and QBAD.DATA = 1 is set. All other variables are frozen.

A.2.6.10 F_CH_DII: F-Channel driver for inputs of data type DINT of fail-safe DP standard slaves and fail-safe standard I/O devices

Function

This F-Block is used for signal processing of an input value of data type DINT of fail-safe DP standard slaves and fail-safe standard I/O devices.

The F-Block cyclically reads the input value of data type DINT of a fail-safe DP standard slave addressed at input VALUE from the associated fail-safe module driver F_PS_12 that communicates with the fail-safe DP standard slave by means of a safety message frame in accordance with the PROFIsafe bus profile. The fail-safe module driver is automatically positioned and interconnected with the CFC function "Generate module drivers".

If the input value is valid, it is made available as F_REAL at output V and as data type F_DINT at output V_DINT.

Note

When values are converted from F_DINT to F_REAL, an inaccuracy of 127, maximum, occurs with values greater than (>) +16,777,215 or less than (<) -16,777,216. That is, the value in F_DINT format is rounded up or rounded off for representation in F_REAL format, as 8 bits of the 32-bit real value are required to represent the exponent.

A quality code (QUALITY output) is generated for the result value at output V. The quality code can assume the following states:

State	Quality code (QUALITY output)
Valid value	16#80
Simulation	16#60
Last valid value	16#44
Fail-safe value	16#48
Invalid value (F-STOP)	16#00

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ADR_CODE	DWORD	Code for VALUE interconnection	Automatically initialized*
	VALUE	DINT	Input channel address	0
	SIM_V	F_DINT	Simulation value	0
	SIM_ON	F_BOOL	1=activate simulation value	0
	SUBS_V	F_DINT	SUBSTITUTION VALUE	0
	SUBS_ON	F_BOOL	1=enable SUBSTITUTION VALUE	0
	PASS_ON	F_BOOL	1=enable passivation	0
	ACK_NEC	F_BOOL	1=user acknowledgement necessary for reintegration following error	0
	ACK_REI	F_BOOL	Acknowledgement for reintegration	0
	IPAR_EN	F_BOOL	1=enable assignment of I-parameters	0
Outputs:	PASS_OUT	F_BOOL	1=passivation due to error	0
	QBAD	F_BOOL	1=process value invalid	0
	QSIM	F_BOOL	1=simulation active	0
	QSUBS	F_BOOL	1=fail-safe value active	0
	V	F_REAL	Process value	0.0
	V_DATA	REAL	DATA component of process value (for monitoring)	0.0
	QUALITY	BYTE	Quality code of process value	B#16#0
	V_DINT	F_DINT	Process value DINT	0
	V_MOD	REAL	Value from F-I/O	0.0
	ACK_REQ	BOOL	Acknowledgment required for reintegration	0
	IPAR_OK	F_BOOL	1=new I-parameter values have been assigned	0

*) Input ADR_CODE is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input ADR_CODE is indicated as changed if changes have been made to the address or the symbolic name of the signal at input VALUE.

Addressing

You must interconnect the symbol for the input value of data type DINT generated with *HW Config* in the symbol table with the VALUE input.

Normal value

If the input value received from the fail-safe DP standard slave is valid, the value is output at output V and V_DINT with quality code (QUALITY) 16#80.

Simulation

A simulation value can be output at output V and V_DINT instead of the normal value that is received from the fail-safe DP standard slave.

The value of input SIM_V with quality code (QUALITY) 16#60 is output when input SIM_ON = 1. Simulation has the highest priority. QBAD = 0 is always set. QSIM = 1 is set if the block is in simulation state.

If simulation is enabled, the input value received from the fail-safe DP standard slave is output at output V_MOD. If communication with the fail-safe DP standard slave is not possible or if a user acknowledgement has not yet occurred following an error, "0" is output.

V_DATA is output if simulation is disabled.

Fail-safe value

If input SUBS_ON = 1, the fail-safe value SUBS_V is output at output V and V_DINT in the following cases:

- The input value is invalid due to a communication error (PROFIsafe).
- The input value is invalid due to a module error or a fail-safe value is received by the module.
- A passivation exists with PASS_ON = 1.
- An F-Startup is pending.
- FV_ACTIVATED is signaled by the module.

The quality code (QUALITY) is set to 16#48 and QSUBS = 1 and QBAD = 1 are set.

If the output of the fail-safe value is not caused by a passivation, PASS_OUT = 1 is set additionally in order to passivate other channels.

Keep last value

If input SUBS_ON = 0, the last valid value of V is output at output V and V_DINT in the following cases:

- The input value is invalid due to a communication error (PROFIsafe).
- The input value is invalid due to a module error or a fail-safe value is received by the module.
- A passivation exists with PASS_ON = 1.
- FV_ACTIVATED is signaled by the module.

The quality code (QUALITY) is set to 16#44 and QSUBS = 0 and QBAD = 1 are set.

If the output of the last valid value is not caused by a passivation, output PASS_OUT = 1 is set additionally in order to passivate other channels.

Reintegration

After an error has been eliminated, the input value received from the DP standard slave can be reintegrated either automatically or not until after a user acknowledgement.

If ACK_NEC = 1 is assigned, a user acknowledgement at input ACK_REI is required after an error has been eliminated. If ACK_NEC = 0 is assigned, automatic reintegration is carried out.

Output ACK_REQ = 1 signals that the error has been eliminated and that a user acknowledgment at input ACK_REI is necessary for reintegration.

A user acknowledgement is not required for reintegration after PASS_ON = 1. A user acknowledgement is not required for reintegration after F-Startup following a CPU-STOP if the F-I/O start up by means of the "System start" slave state (20) in accordance with PROFIsafe Specification V1.30 or later. Otherwise, a communication error (PROFIsafe) is detected.

⚠WARNING

Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.

Communication errors (PROFIsafe) always have to be acknowledged at input ACK_REI irrespective of ACK_NEC. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

⚠WARNING

Startup protection for short-term power failure of the fail-safe DP standard slave

Following a power failure of the fail-safe DP standard slave lasting less than the F-Monitoring time for the fail-safe DP standard slave specified in *HW Config* (see section entitled "Run times, F-Monitoring times, and response times (Page 410)"), automatic reintegration can occur regardless of your setting for input ACK_NEC, as described for the case when ACK_NEC = 0.

If for this case, automatic reintegration is not permissible for the relevant process, you must program startup protection by evaluating the QBAD or PASS_OUT outputs.

In the event of a power failure of the F-I/O lasting longer than the F-Monitoring time for the F-I/O specified in *HW Config*, the F-System detects a communication error.

Reassignment of parameters of a fail-safe DP standard slave

Input IPAR_EN and output IPAR_OK are available for reassignment of parameters of a fail-safe DP standard slave.

The IPAR_EN input corresponds to the iPar_EN_C variable and the IPAR_OK output corresponds to the iPar_OK_S variable in the PROFIsafe bus profile, PROFIsafe Specification V1.30 or later. To find out when to set or reset input IPAR_EN for reassignment of parameters of a fail-safe DP standard slave or how to evaluate the IPAR_OK output, refer to PROFIsafe Specification V1.30 or later or the documentation for the fail-safe DP standard slave.

If more than one F-Channel driver is positioned for a fail-safe DP standard slave, iPar_EN_C is formed from an OR logic operation of all IPAR_EN inputs of the F-Channel drivers belonging to the fail-safe DP standard slave.

If passivation should occur when IPAR_EN = 1, you must also set variable PASS_ON = 1.

Startup characteristics

After an F-Startup, communication first has to be established between the fail-safe module driver and the fail-safe DP standard slave. During this time the fail-safe value SUBS_V with quality code (QUALITY output) 16#48 is output irrespective of the parameter assignment at input SUBS_ON and outputs QSUBS = 1, QBAD = 1 and PASS_OUT = 1 are additionally set.

Error handling

- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

Behavior during F-STOP

In the event of an F-STOP, quality code 16#00 is output at the QUALITY output and QBAD.DATA = 1 is set. All other variables are frozen.

A.2.6.11 F_CH_DIO: F-Channel driver for outputs of data type DINT of fail-safe DP standard slaves and fail-safe standard I/O devices

Function

This F-Block is used for signal processing of an output value of data type DINT of fail-safe DP standard slaves and fail-safe standard I/O devices.

The F-Block cyclically writes the output value of data type DINT for the output of a fail-safe DP standard slave addressed at output VALUE to the associated fail-safe module driver F_PS_12 that communicates with the fail-safe DP standard slave by means of a safety message frame in accordance with the PROFIsafe bus profile. The fail-safe module driver is automatically positioned and interconnected with the CFC function "Generate module drivers".

A quality code is generated for the output value that is written to the fail-safe DP standard slave. The quality code can assume the following states:

State	Quality code (QUALITY output)
Valid value	16#80
Simulation	16#60
Fail-safe value	16#48
Invalid value (F-STOP)	16#00

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ADR_CODE	DWORD	Code for VALUE interconnection	Automatically initialized*
	I	F_DINT	Process value	0
	SIM_I	F_DINT	Simulation value	0
	SIM_MOD	F_BOOL	1=simulation value has priority	0
	SIM_ON	F_BOOL	1=activate simulation value	0
	PASS_ON	F_BOOL	1=enable passivation	0
	ACK_NEC	F_BOOL	1=user acknowledgement necessary for reintegration following error	0
	ACK_REI	F_BOOL	Acknowledgement for reintegration	0
	IPAR_EN	F_BOOL	1=enable assignment of I-parameters	0
Outputs:	PASS_OUT	F_BOOL	1=passivation due to error	0
	QBAD	F_BOOL	1=process value invalid	0
	QSIM	F_BOOL	1=simulation active	0
	VALUE	DINT	Output channel address	0
	QUALITY	BYTE	Quality code of process value	B#16#0
	ACK_REQ	BOOL	Acknowledgment required for reintegration	0
	IPAR_OK	F_BOOL	1=new I-parameter values have been assigned	0

*) Input ADR_CODE is automatically initialized when the S7 program is compiled and must not be changed. When safety programs are compared, input ADR_CODE is indicated as changed if changes have been made to the address or the symbolic name of the signal at input VALUE.

Addressing

You must interconnect the symbol for the output value of data type DINT generated with *HW Config* in the symbol table with the VALUE output.

Normal value

The process value pending at input I is written to the fail-safe DP standard slave. The quality code (QUALITY) is set to 16#80.

Simulation

Instead of the process value pending at input I, a simulation value can also be written to the fail-safe DP standard slave.

If input SIM_ON = 1 and SIM_MOD = 0, the value of input SIM_I is written to the fail-safe DP standard slave and output at output VALUE, provided there is no communication error (PROFIsafe), no module or channel faults (such as a wire break), and no F-Startup.

If input SIM_ON = 1 and SIM_MOD = 1, the value of input SIM_I is also output at output VALUE in the event of a communication error (PROFIsafe), a module or channel fault (such as a wire break), or an F-Startup in order to simulate "error-free" operation even in the absence of an actual fail-safe DP standard slave.

In both cases, the quality code (QUALITY) is set to 16#60, and QSIM = 1 is set.

Note

If you have placed more than one fail-safe channel driver for outputs for a fail-safe DP standard slave, a simulation value is not written if input PASS_ON of another F-Channel driver for outputs of the fail-safe DP standard slave is "1" and input SIM_ON is "0".

Fail-safe value

The fail-safe value "0" is written to the fail-safe DP standard slave in the following cases:

- If a communication error occurs (PROFIsafe)
- If a module or channel fault occurs (such as a wire break)
- During an F-Startup
- If a passivation with PASS_ON = 1 occurs

The quality code (QUALITY) is set to 16#48, and QBAD = 1 is set.

If the output of the fail-safe value is not caused by a passivation, PASS_OUT = 1 is set additionally in order to passivate other channels.

Note

Channel-specific passivation via PASS_ON is not possible for outputs of fail-safe DP standard slaves. If you have positioned more than one fail-safe channel driver for outputs for a fail-safe DP standard slave, the fail-safe value "0" is written for all outputs of the fail-safe DP standard slave when passivation occurs with PASS_ON = 1 at one of the fail-safe channel drivers. If you want to evaluate outputs QBAD and QUALITY of the other F-Channel drivers when PASS_ON = 1 is set at one of the F-Channel drivers, you must activate the PASS_ON inputs of all F-Channel drivers synchronously.

Reintegration

After an error has been eliminated, the fail-safe DP standard slave can be reintegrated either automatically or not until after a user acknowledgement.

If ACK_NEC = 1 is assigned, a user acknowledgement at input ACK_REI is required after an error has been eliminated. If ACK_NEC = 0 is assigned, automatic reintegration is carried out.

Output ACK_REQ = 1 signals that the error has been eliminated and that a user acknowledgment at input ACK_REI is necessary for reintegration.

A user acknowledgement is not required for reintegration after PASS_ON = 1. A user acknowledgement is not required for reintegration after F-Startup following a CPU-STOP if the fail-safe DP standard slave starts up by means of the "System start" slave state (20) in accordance with PROFIsafe Specification V1.30 or later. Otherwise, a communication error (PROFIsafe) is detected.

Note

Channel-specific reintegration is not possible for outputs of fail-safe DP standard slaves. If you position more than one F-Channel driver for outputs for a fail-safe DP standard slave, you must activate the ACK_REI inputs of all F-channel drivers for outputs of the fail-safe DP standard slave synchronously.

⚠ WARNING

Assignment of input ACK_NEC = 0 is only allowed if an automatic reintegration is permissible under safety aspects for the process.

Communication errors (PROFIsafe) always have to be acknowledged at input ACK_REI irrespective of ACK_NEC. For this purpose, you must interconnect input ACK_REI with a signal generated by an operator input. An interconnection with an automatically generated signal is not permitted.

⚠ WARNING

Startup protection for short-term power failure of the fail-safe DP standard slave

Following a power failure of the fail-safe DP standard slave lasting less than the F-Monitoring time for the fail-safe DP standard slave specified in *HW Config* (see section entitled "Run times, F-Monitoring times, and response times (Page 410)"), automatic reintegration can occur regardless of your setting for input ACK_NEC, as described for the case when ACK_NEC = 0.

If for this case, automatic reintegration is not permissible for the relevant process, you must program startup protection by evaluating the QBAD or PASS_OUT outputs.

In the event of a power failure of the F-I/O lasting longer than the F-Monitoring time for the F-I/O specified in *HW Config*, the F-System detects a communication error.

Reassignment of parameters of a fail-safe DP standard slave

Input IPAR_EN and output IPAR_OK are available for reassignment of parameters of a fail-safe DP standard slave.

The IPAR_EN input corresponds to the iPar_EN_C variable and the IPAR_OK output corresponds to the iPar_OK_S variable in the PROFIsafe bus profile, PROFIsafe Specification V1.30 or later. To find out when to set or reset input IPAR_EN for reassignment of parameters of a fail-safe DP standard slave or how to evaluate the IPAR_OK output, refer to PROFIsafe Specification V1.30 or later or the documentation for the fail-safe DP standard slave.

If more than one F-Channel driver is positioned for a fail-safe DP standard slave, iPar_EN_C is formed from an OR logic operation of all IPAR_EN inputs of the F-Channel drivers belonging to the fail-safe DP standard slave.

If passivation should occur when IPAR_EN = 1, you must also set variable PASS_ON = 1.

Startup characteristics

After an F-Startup, communication first has to be established between the fail-safe module driver and the fail-safe DP standard slave. During this time the fail-safe value "0" is written to the fail-safe DP standard slave. The quality code (QUALITY) is set to 16#48 and outputs QBAD = 1 and PASS_OUT = 1 are set.

Error handling

- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

Behavior during F-STOP

In the event of an F-STOP, quality code 16#00 is output at the QUALITY output and QBAD.DATA = 1 is set. All other variables are frozen.

A.2.7 F-System blocks

Overview

Block name	Block number	Description
F_S_BO	FB 390	Sending of 10 data elements of data type F_BOOL in a fail-safe manner to another F-Shutdown group
F_R_BO	FB 391	Receiving of 10 data elements of data type F_BOOL in a fail-safe manner from another F-Shutdown group
F_S_R	FB 392	Sending of 5 data elements of data type F_REAL in a fail-safe manner to another F-Shutdown group
F_R_R	FB 393	Receiving of 5 data elements of data type F_REAL in a fail-safe manner from another F-Shutdown group
F_START	FB 394	F-Start detection
F_PSG_M	FB 471	Marker block for F-Shutdown groups

Integration in F Block types

With the exception of F_START, the F-System blocks must not be integrated in F-Block types.

A.2.7.1 F_S_BO: Sending of 10 data elements of data type F_BOOL in a fail-safe manner to another F-Shutdown group.

Function

The F-Block transfers the data of data type F_BOOL fail-safe adjacent to input SD_BO_*xx* to another F-Shutdown group. The data must be received there with the F_R_BO F-Block.

You must interconnect output S_DB with the input of the same name of the corresponding F_R_BO.

Note

Initialization

You are not allowed to initialize output S_DB with values <> 0.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	SD_BO_00	F_BOOL	SEND DATA BOOL 00	0
	…		…	
	SD_BO_09	F_BOOL	SEND DATA BOOL 09	0
Output:	S_DB	F_WORD	Connection to F_R_BO	0

Error handling

None

A.2.7.2 F_R_BO: Receiving of 10 data elements of data type F_BOOL in a fail-safe manner from another F-Shutdown group

Function

This F-Block receives 10 data elements of data type F_BOOL fail-safe from another F-Shutdown group and makes them available on outputs RD_BO_*xx*. The data must be transferred from the other F-Shutdown group with the F_S_BO F-Block. Interconnect the data at outputs RD_BO_*xx* for further processing with other F-Blocks.

You must interconnect input S_DB with the output of the same name of the corresponding F_S_BO.

You must assign the desired F-Monitoring time at input TIMEOUT. For information about calculating F-Monitoring times, refer to chapter " Run times, F-Monitoring times, and response times (Page 410) ".

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	TIMEOUT	F_TIME	F-MONITORING TIME	T# 0ms
	S_DB	F_WORD	Connection to F_S_BO	0
	SUBBO_00	F_BOOL	SUBSTITUTE FOR BOOL 00	0
	…		…	
	SUBBO_09	F_BOOL	SUBSTITUTE FOR BOOL 09	0
Outputs:	SUBS_ON	F_BOOL	1 = SUBSTITUTE OUTPUT ON	0
	RD_BO_00	F_BOOL	RECEIVED DATA BOOL 00	0
	…		…	
	RD_BO_09	F_BOOL	RECEIVED DATA BOOL 09	0

Substitute values

In the following cases the configured substitute values at inputs SUBBO_*xx* are output at outputs RD_BO_*xx*:

- No updated data is received from the corresponding F_S_BO within the configured F-Monitoring time at input TIMEOUT, because for example partial shutdown is pending for the F-Shutdown group with the corresponding F_S_BO.
- An F-Startup is pending.

The SUBS_ON output is set to 1.

Startup characteristics

After an F-Startup data exchange has to first be established with the corresponding F_S_BO. In this case the configured substitute values at inputs SUBBO_*xx* are output at outputs RD_BO_*xx* and output SUBS_ON is set to 1.

Error handling

An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.7.3 F_S_R: Sending of 5 data elements of data type F_REAL in a fail-safe manner to another F-Shutdown group

Function

This F-Block transfers the data of data type F_REAL fail-safe from the input SD_R_*xx* to another F-Shutdown group. The data must be received there with the F_R_R F-Block.

You must interconnect output S_DB with the input of the same name of the corresponding F_R_R.

Note

Initialization

You are not allowed to initialize output S_DB with values <> 0.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	SD_R_00	F_REAL	SEND DATA REAL 00	0.0
	…		…	
	SD_R_04	F_REAL	SEND DATA REAL 04	0.0
Output:	S_DB	F_WORD	Connection to F_R_R	0

Error handling

None

A.2.7.4 F_R_R: Receiving of 5 data elements of data type F_REAL in a fail-safe manner from another F-Shutdown group.

Function

This F-Block receives 5 data elements of data type F_REAL fail-safe from another F-Shutdown group and makes them available on outputs RD_BO_*xx*. The data must be transferred from the other F-Shutdown group with the F_S_R F-Block.

You must interconnect input S_DB with the output of the same name of the corresponding F_S_R.

You must assign the desired F-Monitoring time at input TIMEOUT. For information about calculating F-Monitoring times, refer to chapter " Run times, F-Monitoring times, and response times (Page 410) ".

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	TIMEOUT	F_TIME	F-MONITORING TIME	T# 0ms
	S_DB	F_WORD	Connection to F_S_R	0
	SUBR_00	F_REAL	SUBSTITUTE FOR REAL 00	0.0
	…		…	
	SUBR_04	F_REAL	SUBSTITUTE FOR REAL 04	0.0
Outputs:	SUBS_ON	F_BOOL	1 = Fail-safe values are output	0
	RD_R_00	F_REAL	RECEIVED REAL 00	0.0
	…		…	
	RD_R_04	F_REAL	RECEIVED REAL 04	0.0

Substitute values

In the following cases the configured substitute values at inputs SUBR_*xx* are output at outputs RD_R_*xx*:

- No updated data is received from the corresponding F_S_R within the configured F-Monitoring time at input TIMEOUT, because for example partial shutdown is pending for the F-Shutdown group with the corresponding F_S_R.
- An F-Startup is pending.

The SUBS_ON output is set to 1.

Startup characteristics

The data exchange has to first be established with the corresponding F_S_R following an F-Startup. At this time the configured substitute values at inputs SUBR_*xx* are output at outputs RD_R_*xx* and the output SUBS_ON is set to 1.

Error handling

An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.7.5 F_START: F-Startup identifier

Function

In the first cycle after an F-Startup or a initial run, the F-Block signals with 1 at output COLDSTRT that an F-Startup was executed. COLDSTRT remains present until the next call of F_START.

The F_START must be called before the evaluating F-Blocks.

Inputs/outputs

	Name	Data type	Description	Default
Output:	COLDSTRT	F_BOOL	F-Startup identifier	1

Error handling

None

A.2.7.6 F_PSG_M: Marker block for F-Shutdown groups

Function

With the F_PSG_M block you have the possibility to split an F-Shutdown group into two F-Shutdown groups.

In the sequence editor of the CFC editor, place the block F_PSG_M in the last F-Runtime group, which should belong to the first F-Shutdown group. Any following F-Runtime groups then form the second F-Shutdown group. The F_PSG_M block is not an F-Block. However, you are still permitted to place it in F-Runtime groups.

Inputs/outputs:

None

Error handling:

None

A.2.8 Flip-flop blocks

Overview

Block name	Block number	Description
F_RS_FF	FB 307	RS Flip-Flop, resetting dominant
F_SR_FF	FB 308	SR Flip-Flop, setting dominant

A.2.8.1 F_RS_FF: RS Flip-Flop, resetting dominant

Function

This F-Block executes the function of an RS Flip-Flops (resetting dominant). The output Q is reset when input R = 0 and input S = 1. The output Q is reset when input R = 1 and input S = 0. Output Q is reset if 1 is at both inputs. The QN output corresponds to the negated Q output.

Truth table

R	S	Qn	QNn
0	0	Qn-1	QNn-1
0	1	1	0
1	0	0	1
1	1	0	1

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	R	F_BOOL	Reset	0
	S	F_BOOL	Set	0
Outputs:	Q	F_BOOL	Output	0
	QN	F_BOOL	Negated output	1

Error handling

An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.8.2 F_SR_FF: SR Flip-Flop, setting dominant

Function

The F-Block executes the function of an *SR Flip-Flop* (setting dominant). The output Q is reset when input R = 0 and input S = 1. The output Q is reset when input R = 1 and input S = 0. Output Q is set if 1 is at both inputs. The QN output corresponds to the negated Q output.

Truth table

R	S	Qn	QNn
0	0	Qn-1	QNn-1
0	1	1	0
1	0	0	1
1	1	1	0

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	R	F_BOOL	Reset	0
	S	F_BOOL	Set	0
Outputs:	Q	F_BOOL	Output	0
	QN	F_BOOL	Negated output	1

Error handling

An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID16#75DA).

A.2.9 IEC pulse and counter blocks

Overview

Block name	Block number	Description
F_CTUD	FB 341	Up and down counter
F_TP	FB 342	Timer pulse
F_TON	FB 343	Timer switch-on delay
F_TOF	FB 344	Timer switch-off delay

A.2.9.1 F_CTUD: Up and down counter

Function

This F-Block is an edge-controlled up/down counter.

The CV count value responds to rising edges of the CU and CD inputs as well as to the level of the LOAD and R inputs:

- Rising edge at CU: CV is increased by 1.

  When the counter value reaches the upper limit (32.767), it no longer counts up.
- Rising edge at CD: CV is decreased by 1.

  When the counter value reaches the lower limit (-32.768), it no longer counts down.
- LOAD = 1: CV is preset with the value of the PV input.

  The values at inputs CU and CD are ignored.
- R = 1: CV is reset to 0.

  The values at inputs CU, CD, and LOAD are ignored.

If a rising edge is available at both the CU input and the CD input during a cycle, the counter keeps its current value.

The QU output is set if the count value is greater than or equal to the preset value PV. The QD output is set if the count value is less than or equal to zero.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	CU	F_BOOL	COUNT UP	0
	CD	F_BOOL	COUNT DOWN	0
	R	F_BOOL	RESET	0
	LOAD	F_BOOL	LOAD PV	0
	PV	F_INT	PRESET VALUE	0
Outputs:	QU	F_BOOL	COUNTER UP QU has the value • 1: If CV ≥ PV • 0: If CV < PV	0
	QD	F_BOOL	COUNTER DOWN QD has the value • 1: If CV ≤ 0 • 0: If CV > 0	0
	CV	F_INT	COUNTER VALUE	0

Error handling

An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.9.2 F_TP: Timer pulse

Function

The F-Block generates a pulse with duration PT at output Q.

The pulse is initiated on a rising edge at input IN. Output Q remains set for duration PT, irrespective of any further variation of the input signal (that is, even if input IN switches from 0 back to 1 before time PT has elapsed).

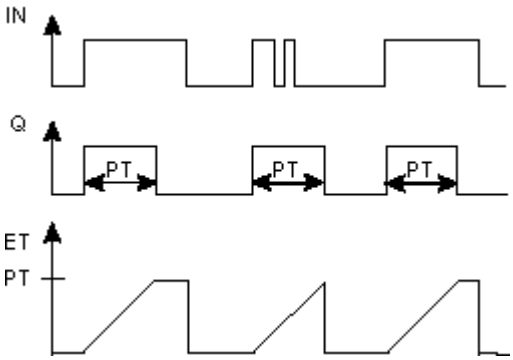
Output ET displays how long output Q has already been set. It can have a maximum value equal to the value of input PT. It is reset when the input IN changes to 0, but only after time PT expires.

If PT < 0, outputs Q and ET are reset.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN	F_BOOL	START INPUT	0
	PT	F_TIME	TIMESET	T# 0ms
Outputs:	Q	F_BOOL	OUTPUT	0
	ET	F_TIME	ELAPSED TIME	T# 0ms

Timing diagram



Fail-safe user times

⚠ WARNING

When using an F-Block with time processing, take the following timing imprecision sources into account when determining your response times:

- Known timing imprecision (based on standard systems) resulting from cyclic processing
- Tolerance of internal time monitoring in the F-CPU
  - For time values of 10 ms to 50 s: 5 ms
  - For time values of > n × 50 s to (n+1) × 50 s: ± (n+1) × 5 ms

Error handling

An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.9.3 F_TON: Timer switch-on delay

Function

This F-Block delays a rising edge by the time PT.

A rising edge at input IN results in a rising edge at output Q once time PT has elapsed. Q remains set until input IN changes to 0.

If input IN changes back to 0 before time PT has elapsed, then output Q remains at 0.

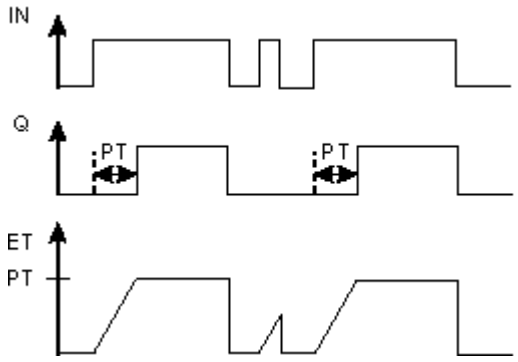
Output ET supplies the time that has passed since the last rising edge at input IN, not to exceed the value at input PT. ET is reset if input IN changes to 0.

If PT < 0, outputs Q and ET are reset.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN	F_BOOL	START INPUT	0
	PT	F_TIME	TIMESET	T# 0ms
Outputs:	Q	F_BOOL	OUTPUT	0
	ET	F_TIME	ELAPSED TIME	T# 0ms

Timing diagram



Fail-safe user times

⚠ WARNING

When using an F-Block with time processing, take the following timing imprecision sources into account when determining your response times:

- Known timing imprecision (based on standard systems) resulting from cyclic processing
- Tolerance of internal time monitoring in the F-CPU
  - For time values of 10 ms to 50 s: 5 ms
  - For time values of > n × 50 s to (n+1) × 50 s: ± (n+1) × 5 ms

Error handling

An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.9.4 F_TOF: Timer switch-off delay

Function

This F-Block delays a falling edge by the time PT.

A rising edge at input IN causes a rising edge at output Q. A falling edge at input IN results in a falling edge at output Q once time PT has elapsed.

If input IN changes back to 1 before time PT has elapsed, then output Q remains at 1.

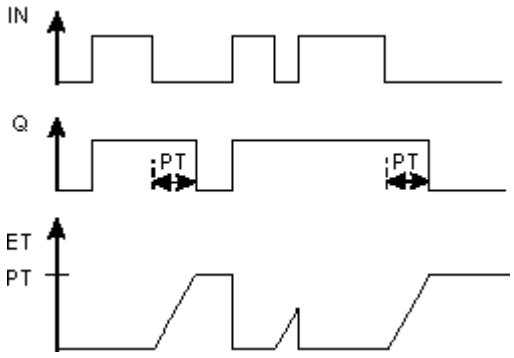
Output ET supplies the time that has passed since the last falling edge at input IN, not to exceed the value at input PT. ET is reset if input IN changes to 1.

If PT < 0, output ET is reset and output Q corresponds to input IN.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN	F_BOOL	START INPUT	0
	PT	F_TIME	TIMESET	T# 0ms
Outputs:	Q	F_BOOL	OUTPUT	0
	ET	F_TIME	ELAPSED TIME	T# 0ms

Timing diagram



Fail-safe user times

⚠ WARNING

When using an F-Block with time processing, take the following timing imprecision sources into account when determining your response times:

- Known timing imprecision (based on standard systems) resulting from cyclic processing
- Tolerance of internal time monitoring in the F-CPU
  - For time values of 10 ms to 50 s: 5 ms
  - For time values of > n × 50 s to (n+1) × 50 s: ± (n+1) × 5 ms

Error handling

An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.10 Pulse blocks

Overview

Block name	Block number	Description
F_REPCYC	FB 309	Clock
F_ROT	FB 310	Timer with on delay and hold function
F_LIM_TI	FB 345	Asymmetrical limiter of a TIME value
F_R_TRIG	FB 346	Detection of a rising edge
F_F_TRIG	FB 347	Detection of a falling edge

A.2.10.1 F_REPCYC: Clock

Function

This F-Block implements a clock with an adjustable period, pulse duration, and interpulse period.

A rising edge at input IN starts the clock. The clock starts at output Q with "0" or "1" depending on the setting at input START.

- When input START = 0, the clock first outputs "0" at output Q for the interpulse period, and then "1" for the pulse duration.
- When input START = 1, the clock first outputs "1" at output Q for the pulse duration, and then "0" for the interpulse period.

The clock is repeatedly changed to 0 until IN. Then, Q = 0 is set.

Output ET always supplies the time that has elapsed since the start of a new period. Output RT always supplies the time remaining until the end of the period. ET is reset when a period ends or when IN = 0. RT is set to the period when a period ends or when IN = 0.

Period, pulse duration, and interpulse period are dependent on the settings at the OFFTIME, ONTIME, and PCTON inputs (where 0 ≤ PCTON ≤ 100). OFFTIME, ONTIME, and PCTON must be specified in such a way that the period does not exceed the maximum value of data type TIME.

- For OFFTIME > 0 ms, the following applies:

  Interpulse period = OFFTIME

  Pulse duration = PCTON x ONTIME

  Period = OFFTIME + (PCTON × ONTIME)
- For OFFTIME = 0 ms, the following applies:

  Interpulse period = ONTIME - (PCTON x ONTIME)

  Pulse duration = PCTON x ONTIME

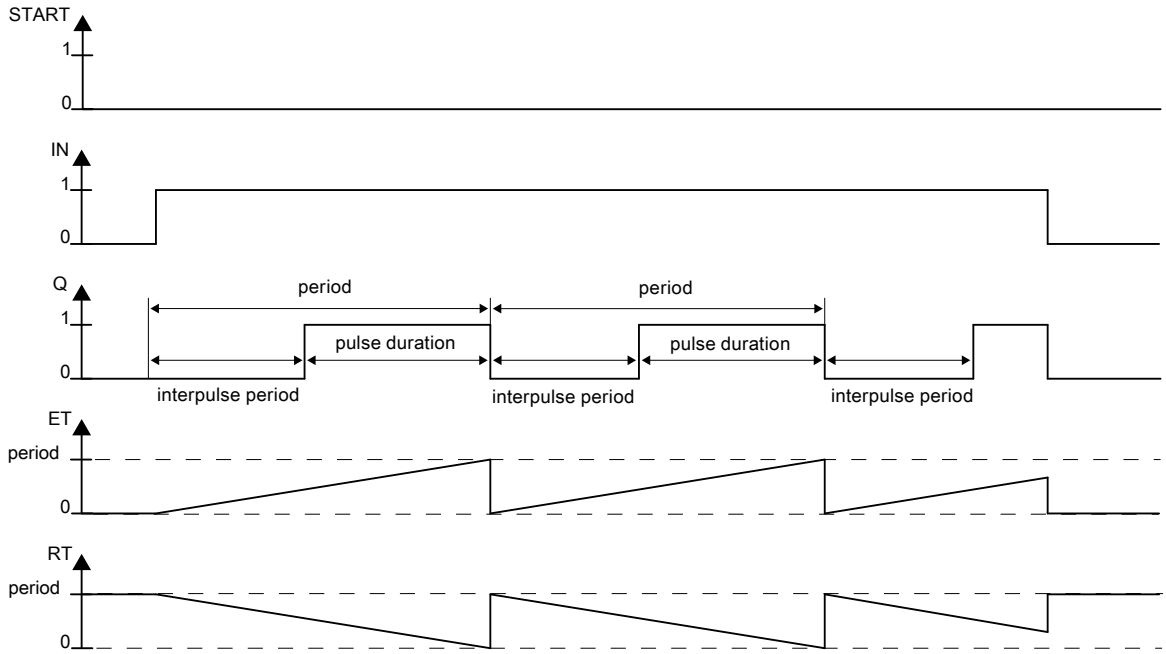
  Period = ONTIME

While input IN = 1, the time values at inputs ONTIME and OFFTIME must not be changed.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN	F_BOOL	Start input	0
	PCTON	F_REAL	Percentage value for pulse duration	0
	START	F_BOOL	0 = Start of period with Q=0 1 = Start of period with Q=1	1
	OFFTIME	F_TIME	Parameter for interpulse period	0 ms
	ONTIME	F_TIME	Parameter for pulse duration	0 ms
Outputs:	Q	F_BOOL	Output	0
	ET	F_TIME	Elapsed time	0 ms
	RT	F_TIME	Remaining time	0 ms

Timing diagram



Fail-safe user times

⚠ WARNING

When using an F-Block with time processing, take the following timing imprecision sources into account when determining your response times:

- Known timing imprecision (based on standard systems) resulting from cyclic processing
- Tolerance of internal time monitoring in the F-CPU
  - For time values of 10 ms to 50 s: 5 ms
  - For time values of > n × 50 s to (n+1) × 50 s: ± (n+1) × 5 ms

Error handling

- If input PCTON is an invalid floating-point number (NaN) or a negative time is present at inputs ONTIME or OFFTIME, the clock shuts down (behavior same as when IN = 0). If an invalid floating-point number (NaN) or a negative time is no longer pending and IN = 1, the clock is restarted (behavior same as for a rising edge at input IN).
- When PCTON < 0.0, ET and RT are generated same as when PCTON = 0, and Q is set to 0. When PCTON > 100.0, ET and RT are generated same as when PCTON = 100, and Q is set to 1.
- If the period exceeds the maximum value of data type TIME, the behavior of the F-Block is undefined.
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.10.2 F_ROT: Timer with on delay and hold function

Function

This F-Block implements a timer with on delay and hold function.

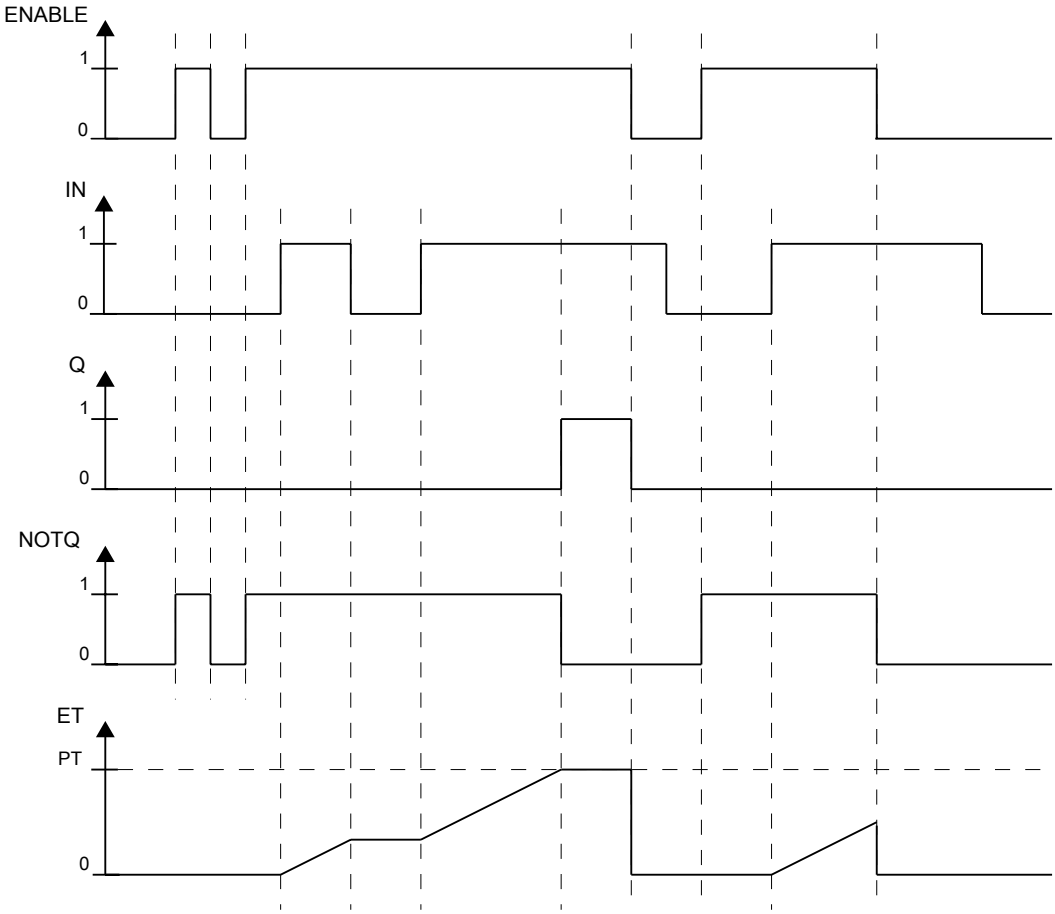
- The timer is enabled with input ENABLE = 1. If input IN = 1, the time at output ET is incremented, but only as high as the value of input PT. If IN changes to "0", the time is halted.

  Q is set to "1" as soon as ET = PT. NOTQ corresponds to the inverted Q.
- The timer is reset with input ENABLE = 0. Output ET is set to 0 ms and Q and NOTQ are set to 0.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ENABLE	F_BOOL	1=Enable timer	0
	IN	F_BOOL	Start input	0
	PT	F_TIME	Time duration	0 ms
Outputs:	Q	F_BOOL	Output	0
	NOTQ	F_BOOL	NEGATING OUTPUT (if ENABLE=1)	0
	ET	F_TIME	Elapsed time	0 ms

Timing diagram



Fail-safe user times

⚠ WARNING

When using an F-Block with time processing, take the following timing imprecision sources into account when determining your response times:

- Known timing imprecision (based on standard systems) resulting from cyclic processing
- Tolerance of internal time monitoring in the F-CPU
  - For time values of 10 ms to 50 s: 5 ms
  - For time values of > n × 50 s to (n+1) × 50 s: ± (n+1) × 5 ms

Error handling

- If a negative time is pending at input PT, the timer is halted (behavior same as when IN = 0). If a negative time is no longer pending, and IN = 1, the timer resumes.
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.10.3 F_LIM_TI: Asymmetrical limiter of a TIME value

Function

This F-Block checks whether input IN is within or outside the interval between MIN and MAX. If input IN lies within the interval, it is passed through to output OUT. If it lies outside of the interval it is limited to MIN or MAX.

- Is IN > MAX, then an upper limit violation exists. MAX is output to output OUT. OUTU is set to 1 and OUTL to 0.
- If IN < MIN, then a lower limit violation exists. MIN is output to output OUT. OUT is set to 0 and OUTL to 1.
- If input IN lies between MIN and MAX, IN is passed through to output OUT. OUTU and OUTL are always set to 0.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN	F_TIME	INPUT	T# 0ms
	MIN	F_TIME	MINIMUM	T# 0ms
	MAX	F_TIME	MAXIMUM	T# 24d 20h 31m 23s 647ms
Outputs:	OUT	F_TIME	Output	T# 0ms
	OUTU	F_BOOL	UPPER LIMIT	0
	OUTL	F_BOOL	LOWER LIMIT	0

Error handling

- Is MIN ≥ MAX, MAX is output at output OUT. OUTU and OUTL are always set to 1.
- An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.10.4 F_R_TRIG: Detection of a rising edge

Function

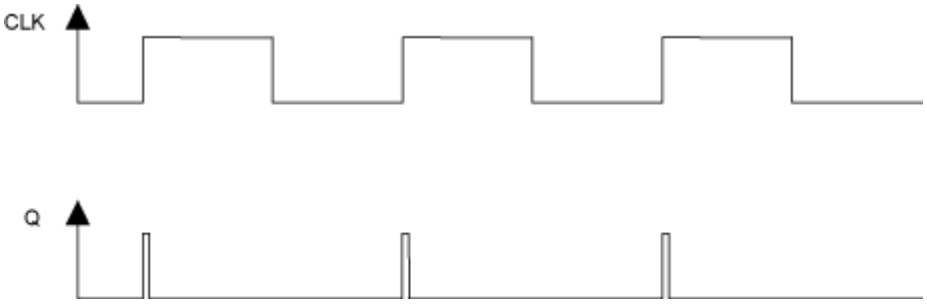
The F-Block checks input CLK for the occurrence of a rising edge.

At a rising edge of input CLK, output Q is set to 1 until the next call of the block.

Inputs/outputs

	Name	Data type	Description	Default
Input:	CLK	F_BOOL	Input	0
Output:	Q	F_BOOL	Output	0

Timing diagram



Startup characteristics

If input CLK has a value of 1 during the first cycle after a F-Startup or an initial run 1, no edge is detected and output Q is set to 0 until the next rising edge on output CLK.

Error handling

None

A.2.10.5 F_F_TRIG: Detection of a falling edge

Function

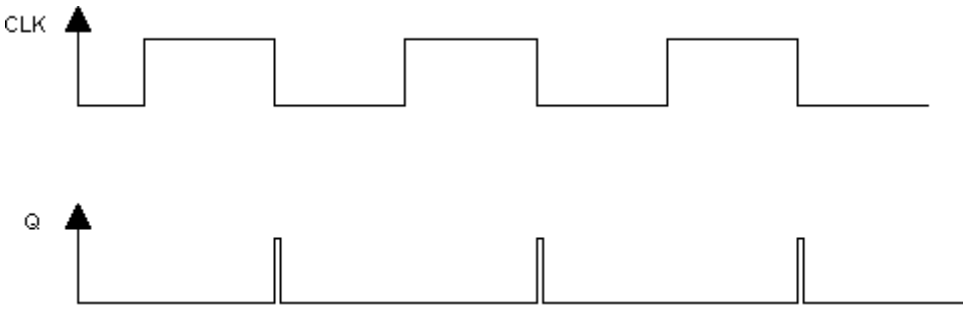
This F-Block checks input CLK for the occurrence of a falling edge.

At a falling edge of input CLK, output Q is set to 1 until the next call of the block.

Inputs/outputs

	Name	Data type	Description	Default
Input:	CLK	F_BOOL	Input	0
Output:	Q	F_BOOL	Output	0

Timing diagram



Startup characteristics

During the first cycle after a F-Start or initial run, no edge is detected.

Error handling

None

A.2.11 Arithmetic blocks with the REAL data type

Overview

Block name	Block number	Description
F_ADD_R	FB 321	Addition of two REAL values
F_SUB_R	FB 322	Subtraction of two REAL values
F_MUL_R	FB 323	Multiplication of two REAL values
F_DIV_R	FB 324	Division of two REAL values
F_ABS_R	FB 325	Absolute value of a REAL value
F_MAX3_R	FB 326	Maximum of three REAL values
F_MID3_R	FB 327	Mean value of three REAL values
F_MIN3_R	FB 328	Minimum of three REAL values
F_LIM_R	FB 329	Asymmetrical limiter of a REAL value
F_SQRT	FB 330	Square root of a REAL value
F_AVEX_R	FB 331	Mean value of a maximum of nine REAL values
F_SMP_AV	FB 333	Sliding mean value of maximum 33 REAL values
F_2oo3_R	FB 456	Median value of three REAL values with 2oo3 evaluation
F_1oo2_R	FB 457	1oo2 evaluation of inputs of data type REAL

A.2.11.1 F_ADD_R: Addition of two REAL values

Function

This F-Block adds the inputs IN1 and IN2 and outputs the sum at output OUT.

OUT = IN1 + IN2

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_REAL	Input 1	0.0
	IN2	F_REAL	Input 2	0.0
Output:	OUT	F_REAL	Output	0.0

Error handling

If an invalid floating-point number (NaN) has been created due to the calculation at output OUT, the following diagnostic event is entered in the diagnostic buffer of the F-CPU:

- "Safety program: invalid REAL number in DB" (Event ID 16#75D9)

A.2.11.2 F_SUB_R: Subtraction of two REAL values

Function

This F-Block subtracts the IN2 input from the IN1 input and outputs the difference at the output OUT.

OUT = IN1 - IN2

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_REAL	Input 1	0.0
	IN2	F_REAL	Input 2	0.0
Output:	OUT	F_REAL	Output	0.0

Error handling

If an invalid floating-point number (NaN) has been created due to the calculation at output OUT, the following diagnostic event is entered in the diagnostic buffer of the F-CPU:

- "Safety program: invalid REAL number in DB" (Event ID 16#75D9)

A.2.11.3 F_MUL_R: Multiplication of two REAL values

Function

This F-Block multiplies the inputs IN1 and IN2 and outputs the product at output OUT.

OUT = IN1 × IN2

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_REAL	Input 1	0.0
	IN2	F_REAL	Input 2	0.0
Output:	OUT	F_REAL	Output	0.0

Error handling

If an invalid floating-point number (NaN) has been created due to the calculation at output OUT, the following diagnostic event is entered in the diagnostic buffer of the F-CPU:

- "Safety program: invalid REAL number in DB" (Event ID 16#75D9)

A.2.11.4 F_DIV_R: Division of two REAL values

Function

This F-Block divides the IN1 input by the IN2 input and outputs the quotient at output OUT.

OUT = IN1 / IN2

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_REAL	Input 1	0.0
	IN2	F_REAL	Input 2	1.0
Output:	OUT	F_REAL	Output	0.0

Error handling

If an invalid floating-point number (NaN) has been created due to the calculation at output OUT, the following diagnostic event is entered in the diagnostic buffer of the F-CPU:

- "Safety program: invalid REAL number in DB" (Event ID 16#75D9)

A.2.11.5 F_ABS_R: Absolute value of a REAL value

Function

This F-Block outputs the absolute value (amount) of input IN at the output OUT.

OUT = | IN |

Inputs/Outputs

	Name	Data type	Description	Default
Input:	IN	F_REAL	Input	0.0
Output:	OUT	F_REAL	Output	0.0

Error handling

None

A.2.11.6 F_MAX3_R: Maximum of three REAL values

Function

This F-Block compares the inputs IN1, IN2 and IN3 and outputs its maximum at output OUT. All the inputs are preset with a value of -3,402823e+38 (largest negative REAL number), so that even a maximum value can be formed from only two inputs.

OUT = MAX {IN1, IN2 , IN3}

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_REAL	Input 1	-3.402823e+38
	IN2	F_REAL	Input 2	-3.402823e+38
	IN3	F_REAL	Input 3	-3.402823e+38
Output:	OUT	F_REAL	Output	-3.402823e+38

Error handling

- If one of the inputs IN1, IN2 and IN3 is an invalid floating-point number (NaN), an invalid floating-point number (NaN) is output at output OUT.
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.11.7 F_MID3_R: Mean value of three REAL values

Function

This F-Block compares the inputs IN1, IN2 and IN3 and outputs its mean value at output OUT.

OUT = mean value {IN1, IN2, IN3}

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_REAL	Input 1	0.0
	IN2	F_REAL	Input 2	0.0
	IN3	F_REAL	Input 3	0.0
Output:	OUT	F_REAL	Output	0.0

Error handling

- If one of the inputs IN1, IN2 and IN3 is an invalid floating-point number (NaN), an invalid floating-point number (NaN) is output at output OUT.
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.11.8 F_MIN3_R: Minimum of three REAL values

Function

This F-Block compares the inputs IN1, IN2 and IN3 and outputs its minimum at output OUT. All the inputs are preset with a value of 3,402823e+38 (largest positive REAL number), so that even a minimum value can be formed from only two inputs.

OUT = MIN {IN1, IN2, IN3}

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_REAL	Input 1	3.402823e+38
	IN2	F_REAL	Input 2	3.402823e+38
	IN3	F_REAL	Input 3	3.402823e+38
Output:	OUT	F_REAL	Output	3.402823e+38

Error handling

- If one of the inputs IN1, IN2 and IN3 is an invalid floating-point number (NaN), an invalid floating-point number (NaN) is output at output OUT.
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.11.9 F_LIM_R: Asymmetrical limiter of a REAL value

Function

This F-Block checks whether input IN is within or outside the interval between MIN and MAX. If input IN lies within the interval, it is passed through to output OUT. If it lies outside of the interval it is limited to MIN or MAX.

With the F-Block you can also check the result of a floating-point operation for overflow (± infinity) and invalid floating-point number (NaN).

- Is IN > MAX or "+ infinity", then an upper limit violation exists. MAX is output at output OUT. OUTU is set to 1 and OUTL to 0.
- Is IN < MIN or "- infinity", then a lower limit violation exists. MIN is output to output OUT. OUT is set to 0 and OUTL on 1.
- If IN lies between MIN and MAX, input IN is passed through to output OUT. OUTU and OUTL are always set to 0.
- If IN is an invalid floating-point number (NaN), the fail-safe value SUBS_INis is output at output OUT. OUTU and OUTL are always set to 1.

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	IN	F_REAL	INPUT	0.0
	MIN	F_REAL	LOWER LIMIT	-100.0
	MAX	F_REAL	UPPER LIMIT	100.0
	SUBS_IN	F_REAL	SUBSTITUTE VALUE	0.0
Outputs:	OUT	F_REAL	OUTPUT	0.0
	OUTU	F_BOOL	UPPER LIMIT VIOLATION	0
	OUTL	F_BOOL	LOWER LIMIT VIOLATION	0

Error handling

- Is MIN ≥ MAX, MAX is output at output OUT. OUTU and OUTL are always set to 1.
- If one of the inputs IN, MIN, MAX or SUBS_IN is an invalid floating-point number (NaN) the fail-safe value SUBS_IN is output at output OUT. OUTU and OUTL are always set to 1.
- An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.11.10 F_SQRT: Square root of a REAL value

Function

This F-Block calculates the square root of the input IN and then outputs it at the output OUT.

OUT = √(IN)

The IN input must be positive.

Inputs/outputs

	Name	Data type	Description	Default
Input:	IN	F_REAL	Input	0.0
Output:	OUT	F_REAL	Output	0.0

Error handling

- If the calculation at output OUT yields an invalid floating-point number (NaN) or a negative value is pending at IN, NaN is output to OUT and the following diagnostic event is entered in the diagnostic buffer of the F-CPU:
  - "Safety program: invalid REAL number in DB" (Event ID 16#75D9)
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.11.11 F_AVEX_R: Mean value of a maximum of nine REAL values

Function

This F-Block calculates the mean value from the inputs INx and outputs the result at output OUT.

OUT = (IN1 + IN2 + ... + IN8 + IN9) / 9

Inputs without a set validity bit VALIDINx are not included in the mean value calculation. If at least MIN inputs are valid, output VALIDOUT = 1 is set. If less than MIN inputs are valid, output VALIDOUT = 0 is set.

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_REAL	INPUT 1	0.0
	...		...	
	IN9	F_REAL	INPUT 9	0.0
	VALIDIN1	F_BOOL	INPUT 1 VALID	1
	...		...	
	VALIDIN9	F_BOOL	INPUT 9 VALID	1
	MIN	F_INT	MINIMUM NUMBER OF VALID INPUTS	9
Outputs:	OUT	F_REAL	OUTPUT	0.0
	VALIDOUT	F_BOOL	OUTPUT VALID	1

Error handling

- If an invalid floating-point number (NaN) has been created due to the calculation at output OUT, the following diagnostic event is entered in the diagnostic buffer of the F-CPU:
  - "Safety program: invalid REAL number in DB" (Event ID 16#75D9)
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.11.12 F_SMP_AV: Sliding mean value of maximum 33 REAL values

Function

This F-Block outputs the mean value of the last N input values IN at output OUT.

$OUT = (IN_k + IN_{k-1} + \ldots + IN_{k-N+1}) / N$

IN_k is the current input value.

The number N of input values must fulfill the condition 0 < N < 33.

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	IN	F_REAL	Input	0.0
	N	F_INT	NUMBER OF INPUTS MONITORED	1
Output:	OUT	F_REAL	OUTPUT	0.0

Startup characteristics

As long as N input values have not been read in after an F-Start or after an initial run, only the available input values (< N) are taken into account for averaging. Input values saved before the start are not taken into account.

Error handling

- If the condition 0 < N < 33 is not fulfilled, the current existing value at input IN is output at output OUT.
- If an invalid floating-point number (NaN) has been created due to the calculation at output OUT, the following diagnostic event is entered in the diagnostic buffer of the CPU:
  - "Safety program: invalid REAL number in DB" (Event ID 16#75D9)
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.11.13 F_2oo3_R: Middle value of three REAL values with 2oo3 evaluation

Function

This F-Block compares the three inputs IN1, IN2 and IN 3 independently of the QBADx inputs and outputs the median value at the OUT output:

- OUT = mean value {IN1, IN2, IN3}

If two or more INx inputs are invalid (two or more QBADx = 1), the OUT output is also invalid and the QBAD output is set to 1.

If the discrepancy between an input INx and the mean of the three inputs IN1, IN2 and IN3 is greater than the assigned DELTA tolerance, a discrepancy error is detected and the output DISx is set.

If in the case of only one invalid INx input its value were to be output as the mean value at the OUT output, it would cause a discrepancy error to be falsely detected for the invalid INx input; in order to avoid this, the fail-safe value for an invalid INx input must differ by more than the DELTA tolerance window of the values which typically occur at the INx inputs during operation.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_REAL	Input 1	0.0
	IN2	F_REAL	Input 2	0.0
	IN3	F_REAL	Input 3	0.0
	QBAD1	F_BOOL	1 = Input IN1 invalid	0
	QBAD2	F_BOOL	1 = Input IN2 invalid	0
	QBAD3	F_BOOL	1 = Input IN3 invalid	0
	DELTA	F_REAL	Tolerance between INx	0.0
Outputs:	OUT	F_REAL	OUTPUT output	0.0
	QBAD	BOOL	1 = OUT output is invalid	0
	DIS1	BOOL	Discrepancy input IN1	0
	DIS2	BOOL	Discrepancy input IN2	0
	DIS3	BOOL	Discrepancy input IN3	0

Used together with F-Channel driver F_CH_AI

If you interconnect input INx of the F_2oo3_R with output V of an F_CH_AI, you must observe the following:

1. Interconnect the QBADx input of the F_2oo3_R with the QBAD output of the F_CH_AI and its output V with input INx of the F_2oo3_R.
2. Configure the SUBS_V input of the F_CH_AI with a value which differs by more than the DELTA tolerance window from the values which typically occur at the INx inputs during operation.
3. Configure the SUBS_ON input of the F_CH_AI with 1.

Error handling

- If one of the IN1, IN2, IN3 inputs is an invalid floating point number (NaN), an invalid floating point number (NaN) is output at the OUT output. DIS1, DIS2 and DIS3 are set to 1.
- If the DELTA input is an invalid floating point number (NaN) or if invalid floating-point numbers (NaNs) arise due to calculations in the F-Block, DIS1, DIS2 and DIS3 are set to 1.

  If invalid floating-point numbers (NaNs) arise due to calculations in the F-Block, the following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

  – "Safety program: Invalid REAL number in DB" (Event ID 16#75D9).
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

  – "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

A.2.11.14 F_1oo2_R: 1oo2 evaluation of inputs of data type REAL

Function

This F-Block outputs either the IN1 or IN2 input at the OUT output, depending on the QBAD1 input:

- QBAD1 = 0: OUT = IN1
- QBAD1 = 1: OUT = IN2

If both the IN1 and IN2 inputs are invalid (QBAD1 and QBAD2 = 1), the OUT output is also invalid and the QBAD output is set to 1.

If inputs IN1 and IN2 differ by more than the assigned DELTA tolerance, a discrepancy error is detected and output

- DIS1 = 1 is set if IN2 is output at the OUT output.
- DIS2 = 1 is set if IN1 is output at the OUT output.

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	IN1	F_REAL	Input 1	0.0
	IN2	F_REAL	Input 2	0.0
	QBAD1	F_BOOL	1 = Input IN1 invalid	0
	QBAD2	F_BOOL	1 = Input IN2 invalid	0
	DELTA	F_REAL	Tolerance between INx	0.0
Outputs:	OUT	F_REAL	Output	0.0
	QBAD	F_BOOL	1 = Output OUT invalid	0
	DIS1	F_BOOL	Discrepancy input IN1	0
	DIS2	F_BOOL	Discrepancy input IN2	0

Used together with F-Channel driver F_CH_AI

If you interconnect input INx of the F_1oo2_R with output V of an F_CH_AI, you must observe the following:

- Interconnect the QBADx input of the F_1oo2_R with the QBAD output of the F_CH_AI and its output V with input INx of the F_1oo2_R.
- Configure the SUBS_V input of the F_CH_AI with a value which differs by more than the DELTA tolerance window from the values which typically occur at the INx inputs during operation.
- Configure the SUBS_ON input of the F_CH_AI with 1.

Error handling

- If one of the IN1, IN2 or DELTA inputs is an invalid floating point number (NaN) or if invalid floating-point numbers (NaNs) arise due to calculations in the F-Block, DIS1 and DIS2 are set to 1.

  If invalid floating-point numbers (NaNs) arise due to calculations in the F-Block, the following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

  – "Safety program: Invalid REAL number in DB" (Event ID 16#75D9).
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

  – "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

A.2.12 Arithmetic blocks with the INT data type

Overview

Block name	Block number	Description
F_LIM_I	FB 350	Asymmetrical limiter of an INT value

A.2.12.1 F_LIM_I: Asymmetrical limiter of an INT value

Function

This block checks whether input IN is within or outside the interval between MIN and MAX. If input IN lies within the interval, it is passed through to output OUT. If it lies outside of the interval it is limited to MIN or MAX.

- Is IN > MAX, then an upper limit violation exists. MAX is output at output OUT. OUTU is set to 1 and OUTL to 0.
- If IN < MIN, then a lower limit violation exists. MIN is output to output OUT. OUT is set to 0 and OUTL on 1.
- If IN lies between MIN and MAX, input IN is passed through to output OUT. OUTU and OUTL are always set to 0.

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	IN	F_INT	Input	0
	MIN	F_INT	MINIMUM	-32768
	MAX	F_INT	MAXIMUM	32767
Outputs:	OUT	F_INT	OUTPUT	0
	OUTU	F_BOOL	UPPER LIMIT	0
	OUTL	F_BOOL	LOWER LIMIT	0

Error handling

- Is MIN ≥ MAX, MAX is output at output OUT. OUTU and OUTL are always set to 1.
- An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.13 Multiplex blocks

Overview

Block name	Block number	Description
F_MOV_R	FB 311	Copy 15 values of data type REAL
F_MUX2_R	FB 332	Multiplexer for 2 REAL values with BOOL selection
F_MUX16R	FB 334	Multiplexer for 16 REAL values with INT selection

A.2.13.1 F_MOV_R: Copy 15 values of data type REAL

Function

This F-Block copies the INx inputs to the OUTx outputs when input ENABLE = 1. When ENABLE = 0, the last valid values are retained at the OUTx outputs.

Output OENABLE corresponds to input ENABLE.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	ENABLE	F_BOOL	1 = Enable copying	0
	IN1	F_REAL	Input 1	0.0
	...		...	
	IN15	F_REAL	Input 15	0.0
Outputs:	OENABLE	F_BOOL	1 = Copying is enabled	0
	OUT1	F_REAL	Output 1	0.0
	...		...	
	OUT15	F_REAL	Output 15	0.0
	CS_USED	F_BOOL	1 = Default values used	0

Startup characteristics

Following an F-Startup, the F-Block behaves as follows:

- Following a CPU-STOP with subsequent cold restart of the F-CPU or during initial run:

  When ENABLE = 0, the (assigned) default values are made available at the OUTx outputs. The CS_USED output is set to "1". CS_USED is reset to "0" as soon as ENABLE changes to "1".

  When ENABLE = 1, the INx inputs are copied to the OUTx outputs. The CS_USED output is set to "0".

- Following a CPU-STOP with subsequent restart (warm restart) of the F-CPU, or following an F-STOP with subsequent positive edge at the RESTART input of the F_SHUTDN block:

  When ENABLE = 0, the last valid values are made available at the OUTx outputs. The CS_USED output retains its default value (0).

  When ENABLE = 1, the INx inputs are copied to the OUTx outputs. The CS_USED output is set to "0".

Note

Prior to initial processing of the F-Block following an F-Startup, the default values are present at outputs OUTx and CS_USED.

⚠ WARNING

F-Startup

Following an F-Startup, plant safety must not be compromised due to either the presence of the (assigned) default values at the OUTx outputs or the presence of the last valid values at the OUTx outputs.

If necessary, evaluate the CS_USED output to determine whether the (assigned) default values or the last valid values were made available at the OUTx outputs after an F-Startup. In addition, the default value "0" of CS_USED must not be changed.

If a restart (warm restart) is performed after a cold restart, CS_USED is reset to the default value (0), even if the default values are still present at the OUTx outputs.

Error handling

An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.13.2 F_MUX2_R: Multiplexer for 2 REAL values with BOOL selection

Function

This F-Block outputs one of the IN0 or IN1 inputs, depending on selection input K, at output OUT:

- K = 0: OUT = IN0
- K = 1: OUT = IN1

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	K	F_BOOL	Selection input	0
	IN0	F_REAL	Input 0	0.0
	IN1	F_REAL	Input 1	0.0
Output:	OUT	F_REAL	Output	0.0

Error handling

An F_STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.13.3 F_MUX16R: Multiplexer for 16 REAL values with INT selection

Function

This block outputs one of the inputs INx, depending on selection input K, at output OUT:

- 0 ≤ K ≤ 15 OUT = IN[K]

Inputs/Outputs

	Name	Data type	Description	Default
Inputs:	K	F_INT	Selection input	0
	IN0	F_REAL	Input 0	0.0
	...		...	
	IN15	F_REAL	Input 15	0.0
Output:	OUT	F_REAL	Output	0.0

Error handling

- If K < 0 or K > 15 0.0 is output at output OUT.
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)

A.2.14 F-Control blocks

Overview

Block name	Block number	Description
F_POLYG	FB 467	Polyline or non-linear characteristic with 24 data points, maximum
F_INT_P	FB 468	Integration function with integration and track mode
F_PT1_P	FB 469	First order delay

A.2.14.1 F_POLYG: F-Control block with non-linear characteristic

Function/mode of operation

The polygon function is used to approach any analog function by means of a specific number of intervals. These are defined by their X/Y coordinates. Within the limits of the approach, up to 24 X/Y coordinate pairs can be defined. The number of X/Y coordinate pairs must be assigned via input N.

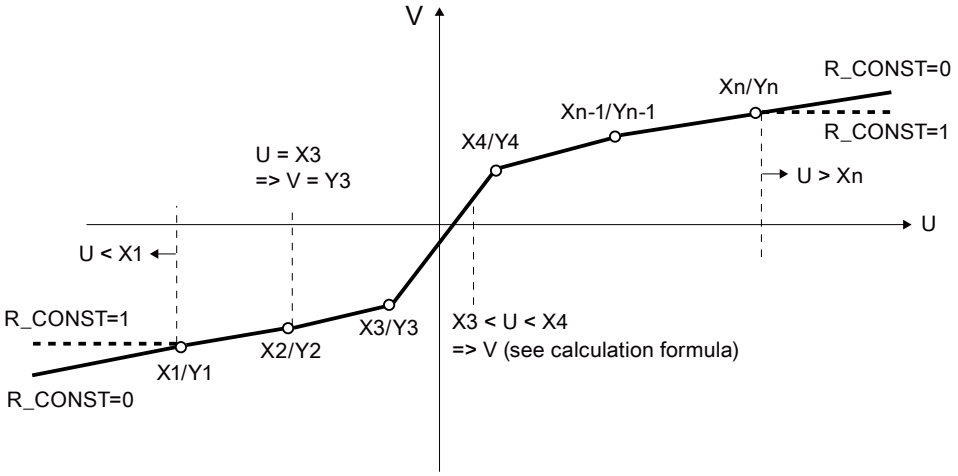
The F-Block converts input U to output V following the non-linear characteristic defined by means of the X/Y coordinate pairs, where X is the value of the analog input and Y the value of the analog output. Linear interpolation is carried out between the Xn/Yn data points.

When R_CONST = "0", extrapolation occurs outside of the end data points based on the first two and last two data points.

If R_CONST = "1" and U is less than (<) X_1, Y_1 is written to output V; similarly, if U is greater than (>) X_N, Y_N is written to output V.

In the event of an invalid parameter assignment of N (2 > N > 24). V = U is output; the same applies for an invalid sequence of X/Y coordinate pairs (Xn ≥ Xn+1 for n = 1, 2, ... N-1).

The figure below provides a graphical illustration of the functionality of this F-Block.



If input value U lies between two X/Y points (Xn < U < Xn+1), V is calculated based on the following formula:

$$V = Y_n + (U - X_n) * \left(\frac{Y_{n+1} - Y_n}{X_{n+1} - X_n} \right)$$

V	Output value
U	Input value
Yn/Xn	Data point n
Yn+1/Xn+1	Data point n+1

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	U	F_REAL	Input value	0.0
	IERR	F_BOOL	1=input value invalid	0
	N	F_INT	Number of data points	0
	R_CONST	F_BOOL	0=extrapolation 1=lowest/highest Y value	0
	X1	F_REAL	X coordinate 1	0.0
	Y1	F_REAL	Y coordinate 1	0.0
	:			
	X24	F_REAL	X coordinate 24	0.0
	Y24	F_REAL	Y coordinate 24	0.0
Outputs:	V	F_REAL	Output value	0.0
	QERR	F_BOOL	Output value invalid	0

Error handling

The validity of input signal U is read in via input IERR. This input parameter can be connected to QBAD of the corresponding input channel driver or of a voter block.

Output QERR is set when one of the following conditions is met:

- U = NaN or one Xn/Yn = NaN

  NaN is assigned to output V.
- The calculation yields NaN.

  NaN is assigned to output V.
- Parameter assignment error Xn >= Xn+1

  U is assigned to output V.
- Input IERR = 1

Diagnostic buffer entry

- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA).
- If an invalid REAL number is determined for V during the calculation, an entry is made in the diagnostic buffer (event ID 16#75D9).

A.2.14.2 F_INT_P: Integration function with integration and track mode

The F_INT_P F-Block works in two different modes:

- Integration mode
- TRACK mode

These two modes are described separately below.

Integration mode

Function/mode of operation

In integration mode, output V rises with a positive input signal U and falls with a negative input signal U.

This F-Block operates in integration mode by forming totals according to the trapezoidal rule for each sampling interval (Ts). The $V_{internal}$ result achieved is located in the range V_HL+hyst to V_LL-hyst, as shown in the figure. After being additionally limited to the range V_LL to V_HL, this value is then written to output V.

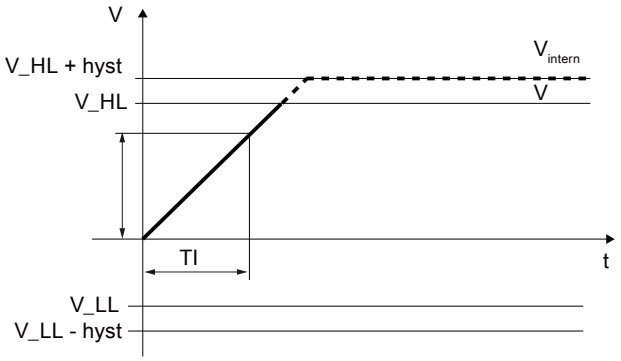


Figure A-1 Step response of F_INT_P

hyst = HYS / 100 * (V_HL – V_LL)

Output value V is calculated according to the following formula:

$$V_X = V_{X-1} + U_X * \frac{Ts}{TI}$$

V_X	Current internal output value
V_{X-1}	Last internal output value ($V_{internal}$)
Ts	Sampling time (time elapsed between two F-Block processing cycles) in seconds
TI	Integration time in seconds
U_X	Current input value

The following additional parameter assignments have an effect on output value V and its calculation:

- HOLD: When HOLD = 1, the last output value for V is held.
- RESET: When there is a positive edge at RESET, output value V is reset (V = 0.0).
- EN_INC and EN_DEC: Processing of the integration function also depends on the input parameters EN_INC and EN_DEC.
  - EN_INC and EN_DEC = 1

    The step response at output V is rising or falling depending on U.
  - EN_INC = 0 and EN_DEC = 1:

    Output value V does not rise. This means that with a positive input value at U, the last output value for V is held.
  - EN_INC = 1 and EN_DEC = 0:

    Output value V does not fall. This means that with a negative input value at U, the last output value for V is held.
  - EN_INC and EN_DEC = 0:

    The last output value for V is always held irrespective of input value U.

In addition to this functionality, limit value monitoring takes place:

- V_HL defines the upper limit for V.

  If $V_{internal}$ exceeds V_HL, V is limited to V_HL; in addition QVHL = 1.
- V_LL defines the lower limit for V.

  If $V_{internal}$ falls below V_LL, V is limited to V_LL; in addition QVLL = 1.

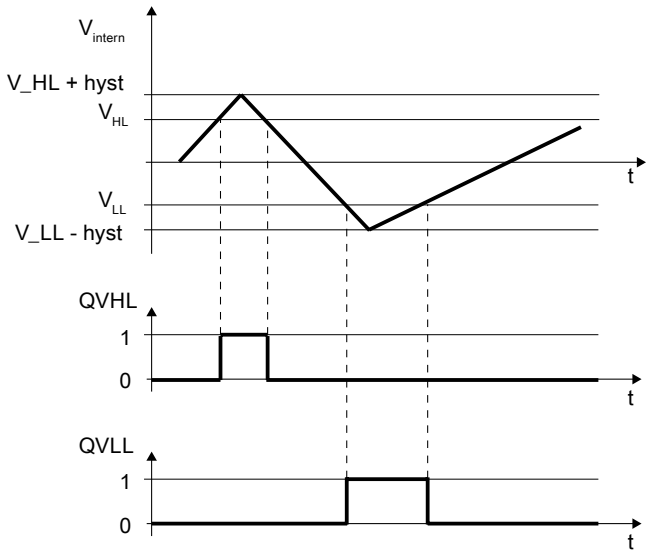


Figure A-2 Limit value monitoring of F_INT_P

Special cases:

- Hysteresis HYS < 0:

  HYS is set internally to 1%. HYS = 0.0 is permitted. In this case, $V_{internal}$ = V if V_HL is exceeded or V_LL is undershot.

- V_LL > V_HL:

  V_HL is set internally to V_LL. In this case, V always corresponds to V_LL.

- TI <= 0:

  TI is set internally to Ts. Thus the times ratio assumes a value of 1 in the equation.

The validity of input signal U is read in via input IERR. This input parameter can be connected to QBAD of the corresponding input channel driver or of a voter block.

If U, V_HL or V_LL is equal to (=) NaN, the value at output V is retained. If HYS = NaN, this only affects $V_{internal}$ and has no effect on V. In this case, $V_{internal}$ = V. Output QERR is set to 1 if NaN occurs at one of the input parameters.

Note

Denormalized values at U are processed and an error message is not output at V.

TRACK mode

In TRACK mode, input signal VTRACK is applied at output V. Thus, TRACK mode can be used to preset the integration function.

This mode is enabled via digital input TRACK = 1.

If input signal VTRACK = NaN, NaN is output at output V. The QERR output is then set to 1.

Limit value monitoring also takes place in TRACK mode:

- V_HL defines the upper limit for V.

  If VTRACK exceeds V_HL, V is limited to V_HL; in addition QVHL = 1.

- V_LL defines the lower limit for V.

  If VTRACK falls below V_LL, V is limited to V_LL; in addition QVLL = 1.

Special cases:

- Hysteresis HYS < 0:

  HYS is set internally to 1%. HYS = 0.0 is permitted. In this case, $V_{internal}$ = V if V_HL is exceeded or V_LL is undershot. HYS has no effect on the formation of V in track mode.

- V_LL > V_HL:

  V_HL is set internally to V_LL. In this case, V always corresponds to V_LL.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	TI	F_TIME	Integration time	1 s
	V_HL	F_REAL	High limit	100.0
	V_LL	F_REAL	Lower limit	0.0
	U	F_REAL	Input value	0.0
	HYS	F_REAL	Hysteresis in %	1.0
	VTRACK	F_REAL	Input value for track mode	0.0
	TRACK	F_BOOL	Mode: 1=track mode	0
	HOLD	F_BOOL	1=hold integration value	0
	RESET	F_BOOL	1=reset V	0
	EN_INC	F_BOOL	1= rising output value permitted	1
	EN_DEC	F_BOOL	1=falling output value permitted	1
	IERR	F_BOOL	1=input value invalid	0
Outputs:	V	F_REAL	Output value	0.0
	QERR	F_BOOL	1=output value invalid	0
	QVHL	F_BOOL	1=high limit violation enabled	0
	QVLL	F_BOOL	1=lower limit violation enabled	0

Error handling

The validity of input signal U is read in via input IERR. This input parameter can be connected to QBAD of the corresponding input channel driver or of a voter block.

Output QERR is set in *integration mode* when one of the following conditions is met:

- Input signal U = NaN
- Input IERR = 1

Output QERR is set in *TRACK mode* when the following condition is met:

- VTRACK = NaN

And irrespective of the mode:

- The calculation yields NaN: Output V retains the last value.
- If NaN is present at one of the input parameters V_LL, V_HL, or HYS.

Diagnostic buffer entry

- If an invalid REAL number is determined during a calculation, an entry is made in the diagnostic buffer (event ID 16#75D9).
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

A.2.14.3 F_PT1_P: First order delay

Function/mode of operation

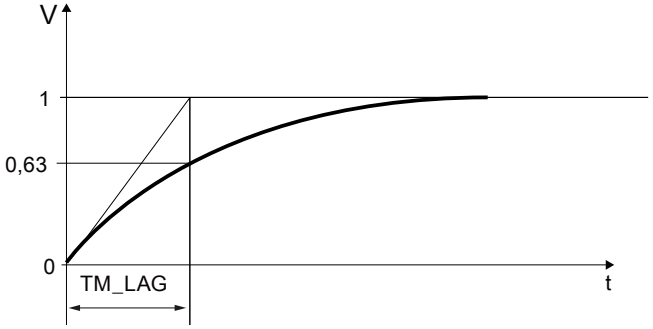
Output value V is calculated according to the following formula:

$$V_X = V_{X-1} + (U_X - V_{X-1}) * \left(\frac{Ts}{\frac{Ts}{2} + TM\_LAG} \right)$$

V_X	Current output value V
V_{X-1}	Last output value V
Ts	Sampling time (time elapsed between two block processing cycles (Diff)) in seconds
TM_LAG	Delay time in seconds
U_X	Current input value U

Input value U is output to output V with a delay corresponding to time constant TM_LAG.

The step response of an amplitude with the value U = 1.0 is reproduced in the figure below:



STOP_RES: When STOP_RES = 1, the arithmetic procedure is stopped. The last output value for V is held. During the changeover from STOP_RES 1 to 0, output V is reset to input value U.

D_OFF: When D_OFF = 1, the delay time is switched off. This means that input value U is applied at output V.

The following boundary conditions are applicable:

- TM_LAG < Ts/2:

  TM_LAG is set to Ts/2. Thus the times ratio assumes a value of 1 in the equation. This means that output value V corresponds to input value U in this case.

The validity of input signal U is read in via input IERR. This input parameter can be connected to QBAD of the corresponding input channel driver or of a voter block.

Note

Denormalized values at U are processed and do not generate an error message.

If an approach to 0 occurs (U = 0.0), V = 0.0 is output when a denormalized value is reached at V (-1.18E-38 or +1.18E-38).

If U is equal to (=) NaN, the value at output V is retained. Output QERR is set to 1.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	TM_LAG	F_TIME	Delay time	0 s
	U	F_REAL	Input value	0.0
	STOP_RES	F_BOOL	Stop/reset	0
	D_OFF	F_BOOL	1=delay switched off	0
	IERR	F_BOOL	1=input value invalid	0
Outputs:	V	F_REAL	Output value	0.0
	QERR	F_BOOL	1=output value invalid	0

Startup characteristics

During startup input value U is applied at output V. V does not behave in accordance with PT1 behavior until a change to input value U has been made subsequently.

Error handling

Output QERR is set when one of the following conditions is met:

- Input signal U is NaN.
- The calculation yields NaN: Output V retains the last value.
- Input IERR = 1

Diagnostic buffer entry

- If an invalid REAL number is determined during the calculation, an entry is made in the diagnostic buffer (event ID 16#75D9).
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

A.2.15 Additional F-Blocks

Overview

Block name	Block number	Description
F_DEADTM	FB 320	Monitoring of changes in F_REAL values at the same measuring point

A.2.15.1 F_DEADTM: Monitoring of changes in F_REAL values at the same measuring point

Function and mode of operation

This block outputs the IN value with a dead time delay at output OUT. The dead time can be configured at input DEADTM. In addition, the delta between the current IN value and the delayed IN value output at OUT is formed. This delta is output at output V_DELTA.

If the calculated delta (V_DELTA) exceeds the delta configured for input parameter DELTA by a time configured in DELAYTM, the output parameter HL (IN > OUT) or LL (IN < OUT) is activated based on the values of IN and OUT.

If 0 is configured for the DELAYTM time, output HL or LL is immediately activated as soon as the delta is exceeded.

The following boundary condition is applicable:

- If DELTA is a negative value:

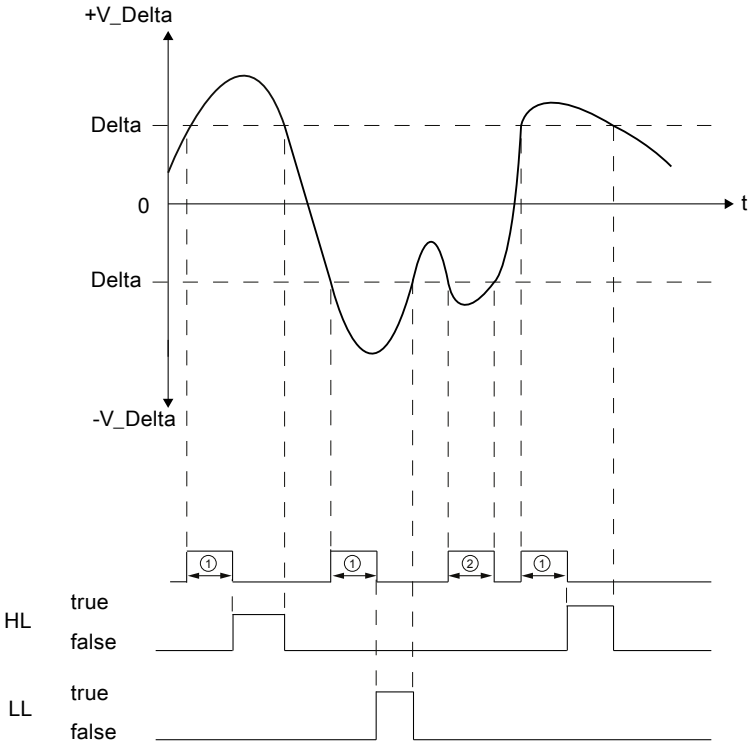
  The modulus is observed from DELTA.

- If DEADTM is a negative value:

  DEADTM is set internally to 0.0.

- If DEADTM > 2E+8 (corresponds to approx. 6 years):

  DEADTM is limited internally to 2E+8.



① Time configured in DELAYTM elapses

② Time configured in DELAYTM does not elapse

Figure A-3 Delta processing

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	IN	F_REAL	Input value	0.0
	DELTA	F_REAL	Delta between IN and OUT	0.0
	DEADTM	F_REAL	Dead time	0.0
	DELAYTM	F_TIME	Delay time for HL and LL	0 s
	RESTART	F_BOOL	1=reset all values (restart)	0
Outputs:	OUT	F_REAL	Output value	0.0
	V_DELTA	F_REAL	Current delta between IN and OUT	0.0
	HL	F_BOOL	1=delta exceeded (IN>OUT)	0
	LL	F_BOOL	1=delta exceeded (IN<OUT)	0

Startup behavior, reset

During startup or on a positive edge at input parameter RESTART, all stored values of IN are reset to the current value of IN. This IN value is output at output parameter OUT until the dead time has elapsed for the first time. Thus during the first cycle, following the events indicated above, V_DELTA is always 0 and in the following cycles until the dead time first runs out completely, V_DELTA is calculated for the time that has elapsed up to that point.

Changes to DEADTM

If changes are made to the dead time, the IN values are not output with the corresponding delay until after this time first runs out completely. During the transition time until the new dead time first runs out completely, the output values exist in relation to the old *and* new time.

Dead time tolerances

For determination of the value to be output at OUT, up to 100 different IN values can be stored within the dead time.

Values created under IN are saved and OUT and delta are processed in accordance with the OB cyclic interrupt time.

This results in the following tolerances for the dead time:

Dead time	Max. tolerance for dead time
DEADTM >= 100 × OB cyclic interrupt time	DEADTM + OB cyclic interrupt time
DEADTM < 100 × OB cyclic interrupt time	DEADTM + (DEADTM / 100)
DEADTM < MAX_CYC (at F_CYC_CO)	MAX_CYC (at F_CYC_CO)
DEADTM < OB cyclic interrupt time	

Error handling

The following error handling takes place for errors at input parameters DEADTM, DELTA, and IN:

- DEADTM:

  When input value DEADTM = NaN, the output values of OUT and V_DELTA also become NaN, and LL and HL = 1.

- DELTA/ V_DELTA:

  When input value DELTA = NaN, OUT and V_DELTA continue to be output, and LL and HL are set to 1, as comparison with DELTA is not possible.

  If an invalid REAL number (NaN) is determined during the calculation of V_DELTA, the response is the same as for NaN at DELTA.

  If a denormalized or infinite value is determined for V_DELTA, if is considered a valid value. In this case, error handling does not take place.

- IN:

  NaN at input IN is initially considered a normal IN value. If the dead time has elapsed and the stored NaN IN value is output to output OUT, the output values of OUT and V_DELTA become NaN, and LL and HL = 1.

Diagnostic buffer entry

- If an invalid REAL number is determined during the calculation, an entry is made in the diagnostic buffer (event ID 16#75D9).
- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA).

See also

F_CYC_CO: F-Control block "F-Cycle time monitoring" (Page 371)

A.3 F-Control blocks in S7 F Systems Lib V1_3 SP1

Overview

F-Control blocks are automatically inserted and interconnected in automatically generated (F-)System charts and in automatically generated (F-)Runtime groups with ID "@F_" or "@SDW_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Block name	Block number	Description
F_MOVRWS	FB 312	F-Control block
F_DIAG	FB 360	F-Control block
F_CYC_CO	FB 395	F-Control block "F-Cycle time monitoring"
F_PLK	FB 396	F-Control block
F_PLK_O	FB 397	F-Control block
F_TEST	FB 398	F-Control block
F_TESTC	FB 399	F-Control block
F_TESTM	FB 400	F-Control block "Deactivate Safety Mode"
F_SHUTDN	FB 458	F-Control block "Shutdown and F-Startup of F-Shutdown modules"
RTGLOGIC	FB 459	F-Control block
F_PS_12	FB 464	F-Control block "F-Module driver"
F_CHG_WS	FB 477	F-Control block
DB_INIT	FC 180	F-Control block
DB_RES	FC 301	F-Control block
F_PS_MIX	FC 302	F-Control block
F_VFSTP1	FC 307	F-Control block
F_VFSTP2	FC 308	F-Control block
FORCEOFF	FC 310	F-Control block "Deactivate F-Force"

A.3.1 F_MOVRWS: F-Control block

Function

This F-Control block is automatically inserted and interconnected in an automatically generated system chart and in an automatically generated runtime group with ID "@SDW_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

A.3.2 F_DIAG: F-Control block

Function

This F-Control block is automatically inserted and interconnected in an automatically generated F-System chart and in an automatically generated F-Runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

A.3.3 F_CYC_CO: F-Control block "F-Cycle time monitoring"

Function

This F-Control block is automatically inserted and interconnected in an automatically generated F-System chart "@F_CycCo-OB3x" and in an automatically generated F-Runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

The F-CPU monitors the F-Cycle time for each cyclic interrupt OB 3x that contains F-Runtime groups. The first time you compile the S7 program, a dialog will appear and prompt you to enter a value for the maximum cycle time "MAX_CYC" that can elapse between two calls of this OB.

If you need to change the maximum F-Cycle time after the initial compilation of the S7 program, you must set the F-Cycle time at the MAX_CYC input of the F_CYC_CO-OB3x block in F-System chart @F_CycCo-OB3x.

For information about setting F-Monitoring times, refer to chapter "Run times, F-Monitoring times, and response times (Page 410)".

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

	Name	Data type	Description	Default
Input:	MAX_CYC	F_TIME	Maximum F-Cycle time	Automatically initialized with 3000 ms if no change is made in the dialog on the initial compilation

⚠ WARNING

Default setting of the maximum MAX_CYC

The default setting for the maximum F-Cycle time is 3000 milliseconds. Check whether this setting is appropriate for your process. Change the default setting if necessary.

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

Error handling

- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)
- If a safety-related error is detected, an F-STOP is triggered. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error detected in F_CYC_CO" (event ID 16#75E1)

A.3.4 F_PLK: F-Control block

Function

This F-Control block is automatically inserted and interconnected in an automatically generated F-System chart and in an automatically generated F-Runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠ WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

Error handling

- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)
- If a safety-related error is detected, an F-STOP is triggered. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error detected in F_PLK" (event ID 16#75E1)

A.3.5 F_PLK_O: F-Control block

Function

This F-Control block is automatically inserted and interconnected in an automatically generated F-System chart and in an automatically generated F-Runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠ WARNING
Safety note - do not change automatically inserted F-Control blocks
Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).
Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

Error handling

- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)
- If a safety-related error is detected, an F-STOP is triggered. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error detected in F_PLK_O" (event ID 16#75E1)

A.3.6 F_TEST: F-Control block

Function

This F-Control block is automatically inserted and interconnected in an automatically generated F-System chart and in an automatically generated F-Runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

Error handling

- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)
- If a safety-related error is detected, an F-STOP is triggered. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error detected in F_TEST" (event ID 16#75E1)

A.3.7 F_TESTC: F-Control block

Function

This F-Control block is automatically inserted and interconnected in an automatically generated F-System chart and in an automatically generated F-Runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

Error handling

- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)
- If a safety-related error is detected, an F-STOP is triggered. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error detected in F_TESTC" (event ID 16#75E1)

A.3.8 F_TESTM: F-Control block "Deactivate Safety Mode"

Function

This F-Control block is automatically inserted and interconnected in an automatically generated F-System chart "@F_TestMode" and in an automatically generated F-Runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

At output TEST, you can evaluate whether or not safety mode is deactivated. The TEST output has the system attribute S7_m_c. It can therefore be monitored directly from an OS. This enables you to arrange to see on your display whether safety mode is deactivated.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

	Name	Data type	Description	Default
Output:	TEST	BOOL	1 = Safety mode deactivated	0

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

Error handling

- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)
- If a safety-related error is detected, an F-STOP is triggered. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error detected" (event ID 16#75E1)

A.3.9 F_SHUTDN: F-Control block "Control of shutdown and F-Startup of the safety program"

Function

This F-Control block is automatically inserted and interconnected in an automatically generated System chart "@F_ShutDn" and in an automatically generated F-Runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

This F-Control block allows you to assign the shutdown behavior and to control the shutdown and the F-Startup of the safety program.

If you have set "According to the configuration of F_SHUTDN" for the F-STOP behavior in the "Safety Program" dialog > "Shutdown Behavior" dialog, you can assign parameters for the SHUTDOWN input to specify how the safety program is to behave on an F-STOP:

- SHUTDOWN = Full: Full shutdown
- SHUTDOWN = Partial: Partial shutdown

Note

The parameter assignment for the SHUTDOWN input must not be changed during an active shutdown.

You can set input RQ_FULL = 1 to trigger a full shutdown of the safety program.

You can use a positive edge at the RESTART input to implement an F-Startup following a shutdown of the safety program (F-STOP) and elimination of the causes for the shutdown if you do not want to perform a restart (warm restart) or cold restart of the F-CPU.

Following an F-Startup, the safety program starts up automatically with the initial values. After a partial shutdown of the safety program, only the F-Shutdown groups that were in the F-STOP carry out an F-Startup. During the F-Startup, a few seconds can elapse before the initialization with the initial values is complete. During initialization, output EN_INIT = 1.

Note

After implementing an F-Startup with a positive edge at input RESTART, a user acknowledgement at input ACK_REI of the fail-safe channel drivers is required for reintegration of the F-I/O affected by the shutdown.

Output FULL_SD displays whether there is a full shutdown of the safety program. At output SD_TYP, you can read out the shutdown behavior set in the "Safety Program" dialog > "Shutdown behavior" dialog.

The SAFE_M output indicates whether the safety program is in safety mode (SAFE_M = 1) or safety mode is deactivated (SAFE_M = 0).

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

	Name	Data type	Description	Default
Inputs:	RESTART	BOOL	1 = F-Startup after shutdown	0
	SHUTDOWN	BOOL	Shutdown behavior	Full
	RQ_FULL	BOOL	1 = Trigger full shutdown	0
	ALARM_EN	BOOL	1 = Activate messages	1
Outputs:	FULL_SD	BOOL	1 = Full shutdown of safety program	0
	SD_TYP	BOOL	Shutdown behavior from dialog: 1 = Full shutdown	0
	EN_INIT	BOOL	1 = Initializing safety program	0
	SAFE_M	BOOL	1 = Safety program in safety mode	0
	F_SIG_OUT	DWORD	Collective signature of the safety program	0
	MSG_DONE	BOOL	= Output DONE of the SFB34 "ALARM_8"	0
	MSG_ERR	BOOL	= Output ERROR of the SFB34 "ALARM_8"	0
	MSG_STAT	WORD	= Output STATUS of the SFB34 "ALARM_8"	0
	MSG_ACK	WORD	= Output ACK_STATE of the SFB34 "ALARM_8"	0
	NFY_DONE	BOOL	= Output DONE of the SFB31 "NOTIFY_8P"	0
	NFY_ERR	BOOL	= Output ERROR of the SFB31 "NOTIFY_8P"	0
	NFY_STAT	WORD	= Output STATUS of the SFB31 "NOTIFY_8P"	0
Input-output:	MSG_TIME	TIME	Time for message repetition	8h

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

Message behavior

- When the safety program is shut down (an F-STOP has been triggered), the F-Control block F_SHUTDN issues the following messages to the OS using SFB 34 "ALARM_8" as "AS I&C message - fault with individual acknowledgement":
  - "Safety program: Partial shutdown", if a partial shutdown of one or more F-Runtime groups occurs
  - "Safety program: full shutdown", if a full shutdown of the safety program occurs
- When an F-Startup occurs after a positive edge at input RESTART, the following message is issued to the OS using SFB 31 "NOTIFY_8P" as "Operating message - no acknowledgement":
  - "F-Startup of safety program on F_SHUTDN"
- When safety mode is deactivated, the following message is issued to the OS using SFB 31 "NOTIFY_8P" both as "Operating message - no acknowledgement" and as "AS I&C message - fault with individual acknowledgement". The "AS I&C message" is repeated whenever time MSG_TIME expires if safety mode is still deactivated. When MSG_TIME = 0, the message is not repeated.
  - "Safety mode deactivated"

By assigning parameter 0 for input ALARM_EN, you can disable input ALARM_EN if a suitable message system is not available.

Outputs MSG_xxx and NFY_xxx

Non-fail-safe information about message behavior errors is made available for service purposes at the MSG_xxx and NFY_xxx outputs. You can read out this information on your ES/OS or, if necessary, the information can be evaluated in your standard user program. The outputs correspond to the inputs of SFB 34 "ALARM_8" or SFB 31 "NOTIFY_8P". For a description, refer to the online help for SFB 34/SFB 31 or in Manual " System Software for S7-300/400 System and Standard Functions (http://support.automation.siemens.com/WW/view/en/1214574) ".

Error handling/diagnostic buffer entry

- If a safety-related error is detected and a full shutdown is performed (an F-STOP has been triggered), the F-Control block F_SHUTDN enters the following event in the diagnostic buffer of the F-CPU:
  - "Complete shutdown of the F-Program active" or "Complete shutdown of the F-Program deactivated" (event ID 16#7xDE)
- When an F-Startup occurs after a positive edge at input RESTART, the following event is entered in the diagnostic buffer of the F-CPU:
  - "Initialization of safety program start" or "Initialization of safety program end" (event ID 16#7xDF)
- When safety mode is deactivated or activated, the following event is entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Safety mode deactivated" or "Safety program: Safety mode active" (event ID 16#7xDB)

See also

F-STOP (Page 84)

F-Startup and reprogramming restart/startup protection (Page 82)

A.3.10 RTGLOGIC: F-Control block

Function

This F-Control block is automatically inserted and interconnected in an automatically generated system chart and in an automatically generated runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

Error handling

If you have assigned "Partial shutdown" for the shutdown behavior and a safety-related error is detected for one F-Shutdown group, the relevant F-Shutdown group is shut down (an F-STOP is triggered). The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:

- "Safety program: Shutdown of a fail-safe shutdown group" (event ID 16#7xDD)

A.3.11 F_PS_12: F-Control block "F_Module_Driver"

Function

This F-Control block is automatically inserted and interconnected in an automatically generated F-System chart "@F_(x)" and in an automatically generated F-Runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

	Name	Data type	Description	Default
Outputs:	DIAG	DWORD	Error information	DW#16#0
	PROFISAFE	F_BOOL	1 = Communication error PROFISAFE	0

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

Output DIAG

Non-fail-safe information about safety-related communication errors between the F-CPU and F-I/O using the PROFIsafe safety protocol are made available for service purposes at output DIAG. You can read out this information on your ES/OS or, if necessary, the information can be evaluated in your standard user program.

Structure of DIAG

Bit no.	Assignment	Possible error causes	Remedies
Bit 0	Timeout detected by F-I/O	The PROFIBUS connection between the F-CPU and F-I/O is faulty. The F-Monitoring time of the F-I/O in *HW Config* is set too low. The F-I/O is receiving invalid parameter assignment data. or	Check the PROFIBUS connection and ensure that there are no external sources of interference. Check the parameter assignment of the F-I/O in *HW Config*. If necessary, set a higher value for the F-Monitoring time. Recompile the hardware configuration, and download it to the F-CPU. Compile the S7 program again. Check the diagnostic 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 error detected by F-I/O	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	Reserve	—	—
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	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	Reserve	—	—
Bits 8 to 31	Reserve	—	—

Error handling

- An F-STOP is triggered when an error occurs in the safety data format in the corresponding instance DB. The following diagnostic event is then entered in the diagnostic buffer of the F-CPU:
  - "Safety program: Error in safety data format in DB" (Event ID 16#75DA)
- The safety function requires that fail-safe values 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:
  - During an F-Startup
  - 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
  - If F-I/O or channel faults are detected (e.g. wire break, short-circuit, or discrepancy error)
  - As long as you have enabled an F-I/O passivation on the F-Channel driver at input PASS_ON

  One of the following diagnostic events is then entered in the diagnostic buffer of the F-CPU (except during F-Startup):
  - "F-I/O input channel passivated / F-I/O input channel depassivated" (event ID 16#7xE3)
  - "F-I/O output channel passivated / F-I/O output channel depassivated" (event ID 16#7xE4)
  - "F-I/O passivated / F-I/O depassivated" (event ID 16#7xE5)

A.3.12 F_CHG_WS: F-Control block

Function

This F-Control block is automatically inserted and interconnected in an automatically generated system chart and in an automatically generated runtime group with ID "@SDW_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠ WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

A.3.13 DB_INIT: F-Control block

Function

This F-Control block is automatically inserted and interconnected in an automatically generated system chart and in an automatically generated runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

A.3.14 DB_RES: F-Control block

Function

This F-Control block is automatically inserted and interconnected in an automatically generated system chart and in an automatically generated runtime group at the start of the runtime sequence in OB 100 with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

A.3.15 F_PS_MIX: F-Control block

Function

This F-Control block is automatically inserted and interconnected in an automatically generated F-System chart and in an automatically generated F-Runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

A.3.16 F_VFSTP1: F-Control block

Function

This F-Control block is automatically inserted in the S7 program when it is compiled in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

A.3.17 F_VFSTP2: F-Control block

Function

This F-Control block is automatically inserted in the S7 program when it is compiled in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

A.3.18 FORCEOFF: Deactivation of F-Force

Function

This F-Control block is automatically inserted and interconnected in an automatically generated system chart and in an automatically generated F-Runtime group with ID "@F_" during compilation of the S7 program in order to create an executable safety program from the user's safety program.

⚠WARNING

Safety note - do not change automatically inserted F-Control blocks

Automatically inserted F-Control blocks and automatically inserted (F-)System charts and (F-)Runtime groups with ID "@F_" or "@SDW_" are visible after compilation. You must not delete them or modify them in any way (except as explicitly described).

Failure to adhere to this can result in errors on the next compilation.

Inputs/outputs

Non-documented inputs/outputs are initialized or interconnected automatically when the S7 program is compiled and you must not change them. Online changes affecting non-documented inputs/outputs can lead to an F-STOP. You can overcome manipulations to these inputs/outputs by recompiling the S7 program.

A.4 F-Library Failsafe Blocks (V1_2)

The *Failsafe Blocks* F-Library (V1_2) is the predecessor version for the *S7 F Systems Lib* F-Library V1_3.

The F-Blocks of the *Failsafe Blocks* F-Library (V1_2) are described in the online help for this F-Library.

A.5 Differences between the F-Libraries Failsafe Blocks (V1_x) and S7 F Systems Lib V1_3

The following subsections describe differences between the *Failsafe Blocks* F-Library (V1_1) and *Failsafe Blocks* F-Library (V1_2) and between the *Failsafe Blocks* F-Library (V1_2) and *S7 F Systems Lib* V1_3. Only those F-Block changes that are relevant to the user and that affect the function, including the startup behavior and error handling, and the inputs/outputs of the F-Block are described.

Even if no changes (i.e., "none") are indicated, it is possible that the signatures/initial value signatures of an F-Block have changed compared to a previous version of the F-Library, for example, due to code optimizations, changes in diagnostic buffer entries, or changes in the internal interaction of the F-Blocks.

For information about the runtimes of the F-Blocks, refer to the section entitled "Run times, F-Monitoring times, and response times (Page 410)". If required, you can find out the new memory requirements from *SIMATIC Manager*.

When you upgrade to a new version of the F-Library, take note of the F-Block changes and check whether these changes may affect the behavior of your safety program. Refer also to the section entitled "Acceptance test of safety program changes (Page 184)".

Refer to Annex 1 of the Certification Report to obtain the signatures/starting value signatures for the F-Blocks of F-Library *S7 F Systems Lib* V1_3 SP1.

A.5.1 Logic blocks with the BOOL data type

F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_AND4	89B0	6837	←	←	←	←	None
F_OR4	5DCA	6B42	←	←	←	←	None
F_XOR2	6D4D	069A	←	←	←	←	None
F_NOT	9CD8	DD06	←	←	←	←	None
F_2OUT3	34DE	D79F	←	←	←	←	None
F_XOUTY	5F86	C51D	6A1C	C51D	←	←	F-STOP instead of CPU-STOP (1)

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_AND4	89B0	6837	None
F_OR4	5DCA	6B42	None
F_XOR2	6D4D	069A	None
F_NOT	9CD8	DD06	None
F_2OUT3	34DE	D79F	None
F_XOUTY	68A0	68BE	Default output OUTN = 1

A.5.2 F-Blocks for F-Communication between F-CPUs

F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_SENDBO	B204	F3D1	E223	F3D1	←	←	F-STOP instead of CPU-STOP (1)
F_RCVBO	6FFB	DCF4	A2B9	DCF4	←	←	F-STOP instead of CPU-STOP (1)
F_SENDR	3BA4	5B9D	7B16	5B9D	←	←	F-STOP instead of CPU-STOP (1)
F_RCVR	F6F3	14C1	B854	14C1	←	←	F-STOP instead of CPU-STOP (1)
F_SDS_BO	—	—	—	—	—	—	—
F_RDS_BO	—	—	—	—	—	—	—

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_SENDBO	8D63	5812	New input EN_SMODE, For function, see block description
F_RCVBO	DD4B	8360	New output SENDMODE, For function, see block description
F_SENDR	2FE2	678B	New input EN_SMODE, For function, see block description
F_RCVR	3209	B103	New output SENDMODE, For function, see block description
F_SDS_BO	C804	662A	New F-Block in *S7 F Systems Lib* V1_3
F_RDS_BO	4389	EDD9	New F-Block in *S7 F Systems Lib* V1_3

A.5.3 F-Blocks for comparing two input values of the same type

F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_CMP_R	—	—	—	—	—	—	—
F_LIM_HL	435E	CB3F	5116	7656	←	←	F-STOP instead of CPU-STOP (1) Behavior with floating-point operations (2) New input SUBS_IN, For function, see block description
F_LIM_LL	FB73	CB3F	AF69	7656	←	←	F-STOP instead of CPU-STOP (1) Behavior with floating-point operations (2) New input SUBS_IN, For function, see block description

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

2) If a floating-point operation produces an overflow (± infinity) or a denormalized or invalid floating-point number (NaN), or if an invalid floating-point number (NaN) is already present as an address, this event no longer results in a CPU-STOP. The "Overflow (± infinity)", "Denormalized floating-point number", or "Invalid floating-point number (NaN)" events are:

- Either output at the output and available for further processed by the subsequent F-Blocks

  or

- Signaled at special outputs. If necessary, a fail-safe value is output.

If the floating-point operation yields an invalid floating-point number (NaN) and an invalid floating-point number (NaN) does not already exist as an address, the following diagnostic event is entered in the diagnostic buffer of the F-CPU:

- "Safety program: invalid REAL number in DB" (Event ID 16#75D9)

You can use this diagnostic buffer entry to identify the F-Block with the invalid floating-point number (NaN).

Refer also to the documentation of the F-Block.

If you cannot rule out the occurrence of these events in your safety program, you must decide independently of your application whether you have to react to these events in your safety program. With F-Block F_LIM_R, you can check the result of a floating-point operation for overflow (± infinity) and invalid floating-point number.

F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_CMP_R	689A	602E	New F-Block in *S7 F Systems Lib* V1_3
F_LIM_HL	A43A	1E14	If the calculations in the F-Block yield invalid floating-point numbers (NaN), the fail-safe value at input SUBS_IN is output at output QH instead of "1". Default output QHN = 1
F_LIM_LL	1451	1E14	If the calculations in the F-Block yield invalid floating-point numbers (NaN), the fail-safe value at input SUBS_IN is output at output QL instead of "1". Default output QLN = 1

A.5.4 Voter blocks for inputs of data type REAL and BOOL

F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_2oo3DI	—	—	—	—	—	—	—
F_1oo2AI	—	—	—	—	—	—	—
F_2oo3AI	—	—	—	—	—	—	—

F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_2oo3DI	5323	04A0	New F-Block in *S7 F Systems Lib* V1_3
F_1oo2AI	013D	0CE3	New F-Block in *S7 F Systems Lib* V1_3
F_2oo3AI	4580	CE7E	New F-Block in *S7 F Systems Lib* V1_3

A.5.5 Blocks and F-Blocks for data conversion

Blocks / F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_BO_FBO	27AB	87DA	←	←	←	←	None
F_R_FR	6ED3	6BCE	4278	6BCE	←	←	Behavior with floating-point operations (2)
F_QUITES	89EC	B027	B433	B027	←	←	F-STOP instead of CPU-STOP (1)
F_TI_FTI	A06D	6BCE	←	←	←	←	None
F_I_FI	4871	87DA	←	←	←	←	None
F_FI_FR	—	—	—	—	672A	9FDE	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 SP1 and higher
F_FR_FI	—	—	*	*	*	*	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 and higher *) F-Block is not certified
F_CHG_R	—	—	—	—	E4CD	5DB5	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 SP2 and higher
F_CHG_BO	—	—	—	—	D042	E5F2	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 SP2 and higher
F_FBO_BO	Without	Without	Without	Without	Without	Without	None
F_FR_R	Without	Without	Without	Without	Without	Without	None
F_FI_I	Without	Without	Without	Without	Without	Without	None
F_FTI_TI	Without	Without	Without	Without	Without	Without	None

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

2) If a floating-point operation produces an overflow (± infinity) or a denormalized or invalid floating-point number (NaN), or if an invalid floating-point number (NaN) is already present as an address, this event no longer results in a CPU-STOP. The "Overflow (± infinity)", "Denormalized floating-point number", or "Invalid floating-point number (NaN)" events are:

- Either output at the output and available for further processing by subsequent F-Blocks

  or

- Signaled at special outputs. If necessary, a fail-safe value is output.

If the floating-point operation yields an invalid floating-point number (NaN) and an invalid floating-point number (NaN) does not already exist as an address, the following diagnostic event is entered in the diagnostic buffer of the F-CPU:

- "Safety program: invalid REAL number in DB" (Event ID 16#75D9)

You can use this diagnostic buffer entry to identify the F-Block with the invalid floating-point number (NaN).

Refer also to the documentation of the F-Block.

If you cannot rule out the occurrence of these events in your safety program, you must decide based on your application whether you have to react to these events in your safety program. With F-Block F_LIM_R, you can check the result of a floating-point operation for overflow (± infinity) and invalid floating-point number.

Blocks / F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_BO_FBO	27AB	87DA	None
F_R_FR	4278	6BCE	None
F_QUITES	797A	B027	None
F_TI_FTI	A06D	6BCE	None
F_I_FI	4871	87DA	None
F_FI_FR	672A	9FDE	None
F_FR_FI	2B3C	B269	F-Block is certified New outputs OUTU and OUTL; for function, see block description
F_CHG_R	E4CD	5DB5	None
F_CHG_BO	D042	E5F2	None
F_FBO_BO	Without	Without	None
F_FR_R	Without	Without	None
F_FI_I	Without	Without	None
F_FTI_TI	Without	Without	None

A.5.6 F-Channel drivers for F-I/O

F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_CH_BI	—	—	—	—	—	—	—
F_CH_BO	—	—	—	—	—	—	—
F_PA_AI	—	—	—	—	9046	14F5	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 SP4 and higher
F_PA_DI	—	—	—	—	BCD4	9564	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 SP4 and higher
F_CH_DI	E41B	F504	2346	F504	A47F	EC21	F-STOP instead of CPU-STOP (1) *S7 F Systems* V5.2 SP1 and higher, new output for internal interaction; when upgrading to this version, you must perform a full download with CPU-STOP.
F_CH_DO	6E6A	18CF	E0B9	D7F0	92C1	DA68	F-STOP instead of CPU-STOP (1) New input SIM_MOD, For function, see block description *S7 F Systems* V5.2 SP1 and higher, new output for internal interaction; when upgrading to this version, you must perform a full download with CPU-STOP.
F_CH_AI	296D AA4F	C540	8F67	D784	741E	8D4B	F-STOP instead of CPU-STOP (1) Behavior with floating-point operations (2) *S7 F Systems* V5.2 SP1 and higher, new output for internal interaction; when upgrading to this version, you must perform a full download with CPU-STOP.

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

2) If a floating-point operation produces an overflow (± infinity) or a denormalized or invalid floating-point number (NaN), or if an invalid floating-point number (NaN) is already present

as an address, this event no longer results in a CPU-STOP. The "Overflow (± infinity)", "Denormalized floating-point number", or "Invalid floating-point number (NaN)" events are:

- Either output at the output and available for further processing by subsequent F-Blocks

  or

- Signaled at special outputs. If necessary, a fail-safe value is output.

If the floating-point operation yields an invalid floating-point number (NaN) and an invalid floating-point number (NaN) does not already exist as an address, the following diagnostic event is entered in the diagnostic buffer of the F-CPU:

- "Safety program: invalid REAL number in DB" (Event ID 16#75D9)

You can use this diagnostic buffer entry to identify the F-Block with the invalid floating-point number (NaN).

Refer also to the documentation of the F-Block.

If you cannot rule out the occurrence of these events in your safety program, you must decide based on your application whether you have to react to these events in your safety program. With F-Block F_LIM_R, you can check the result of a floating-point operation for overflow (± infinity) and invalid floating-point number.

F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_CH_BI	E888	5FA7	New F-Block in *S7 F Systems Lib* V1_3
F_CH_BO	A8C7	A5E4	New F-Block in *S7 F Systems Lib* V1_3
F_PA_AI	84D9	B5A7	Order of inputs SIM_ON and SIM_V and inputs SUBS_ON and SUBS_V reversed Output IPAR_OK F_BOOL instead of BOOL New output V_MOD, For function, see block description For behavior during F-STOP, see block description Interconnection with F_PS_12 instead of F_MPA_I
F_PA_DI	2FC7	E4F2	Order of inputs SIM_ON and SIM_I reversed Inputs SUBS_ON and SUBS_I omitted Output IPAR_OK F_BOOL instead of BOOL New outputs QN, Q0 ... Q7 and Q_MOD, For function, see block description For behavior during F-STOP, see block description Interconnection with F_PS_12 instead of F_MPA_I

F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_CH_DI	3119	EA57	New inputs for internal interaction New outputs DISCF and DISCF_R receive discrepancy error information from output DIAG_1 and DIAG_2 of F_M_DI8 and F_M_DI24, New output Q_MOD, New outputs QMODF and QMODF_R For function, see block description In the case of redundantly configured F-I/O, a user acknowledgement is also required if the indicated errors occurred only on one F-I/O and, thus, did not trigger a fail-safe value output to the process. For behavior during F-STOP, see block description Interconnection with F_PS_12 instead of F_M_DI24 or F_M_DI8
F_CH_DO	F967	4F58	New inputs for internal interaction New outputs QMODF and QMODF_R For function, see block description In the case of redundantly configured F-I/O, a user acknowledgement is also required if the indicated errors occurred only on one F-I/O and, thus, did not trigger a fail-safe value output to the process. For behavior during F-STOP, see block description Interconnection with F_PS_12 instead of F_M_DO8 or F_M_DO10
F_CH_AI	D846	3A31	New inputs for internal interaction New input MODE taken over from F_M_AI6, New output V_MOD, New outputs QMODF and QMODF_R, New output AL_STATE For function, see block description New measuring range coding is supported, see block description In the case of redundantly configured F-I/O, a user acknowledgement is also required if the indicated errors occurred only on one F-I/O and, thus, did not trigger a fail-safe value output to the process. For behavior with floating-point operations, see block description For behavior during F-STOP, see block description Interconnection with F_PS_12 instead of F_M_AI6

A.5.7 F-System blocks

Blocks / F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_S_BO	CC75	1110	F353	1110	←	←	None
F_R_BO	3E82 D775	B9A5	6CE1	B9A5	←	←	F-STOP instead of CPU-STOP (1) If no updated data are received within the F-Monitoring time, a CPU-STOP does not occur; instead, the assigned fail-safe values are output.
F_S_R	D897	1FC2	372C	1FC2	←	←	None
F_R_R	6C69 6F8F	543A	64A1	543A	←	←	F-STOP instead of CPU-STOP (1) If no updated data are received within the F-Monitoring time, a CPU-STOP does not occur; instead, the assigned fail-safe values are output.
F_START	5791	2151	←	←	←	←	None
F_PSG_M	—	—	—	—	Without	Without	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 SP1 and higher

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

Blocks / F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_S_BO	59D5	1110	None
F_R_BO	CC9E	E882	None
F_S_R	7394	1FC2	None
F_R_R	AC9C	237E	None
F_START	5791	2151	None
F_PSG_M	Without	Without	None

A.5.8 Flip-flop blocks

F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_RS_FF	5A81	069A	3A1A	069A	←	←	F-STOP instead of CPU-STOP (1)
F_SR_FF	7F12	069A	61BC	069A	←	←	F-STOP instead of CPU-STOP (1)

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_RS_FF	6257	B56D	None
F_SR_FF	9EBE	B56D	None

A.5.9 IEC pulse and counter blocks

F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_CTUD	9928	F7D1	EF97	F7D1	←	←	F-STOP instead of CPU-STOP (1)
F_TP	D608	7CFC	64DD	7CFC	←	←	F-STOP instead of CPU-STOP (1)
F_TON	DD31	7CFC	F8E5	7CFC	←	←	F-STOP instead of CPU-STOP (1)
F_TOF	F899	7CFC	31A9	7CFC	←	←	F-STOP instead of CPU-STOP (1)

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_CTUD	609B	188C	None
F_TP	E671	22F6	None
F_TON	38DA	22F6	None
F_TOF	E45B	22F6	None

A.5.10 Pulse blocks

F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_REPCYC	—	—	—	—	—	—	—
F_ROT	—	—	—	—	—	—	—
F_LIM_TI	13A0	7CAB	3ABB	7CAB	←	←	F-STOP instead of CPU-STOP (1)
F_R_TRIG	3E5E	8F11	BFC8	8F11	←	←	If input CLK has a value of "1" during the first cycle after an F-Startup or an initial run, no edge is detected and output Q remains set to "0" until the next rising edge on output CLK
F_F_TRIG	75E7	8F11	←	←	←	←	None

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_REPCYC	8F66	61F4	New F-Block in *S7 F Systems Lib* V1_3
F_ROT	7ECA	73FD	New F-Block in *S7 F Systems Lib* V1_3
F_LIM_TI	6E64	68DC	None
F_R_TRIG	BFC8	8F11	None
F_F_TRIG	75E7	8F11	None

A.5.11 Arithmetic blocks with the REAL data type

F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_ADD_R	643F	206C	B495	B1DF	←	←	Behavior with floating-point operations (2)
F_SUB_R	46B5	206C	5C35	B1DF	←	←	Behavior with floating-point operations (2)
F_MUL_R	B7AC	206C	36DC	B1DF	←	←	Behavior with floating-point operations (2)
F_DIV_R	9CF2	4A67	D7A8	C0B8	←	←	Behavior with floating-point operations (2)
F_ABS_R	7E9D	4885	←	←	←	←	None
F_MAX3_R	AEA9	9A67	78DB	5833	←	←	Behavior with floating-point operations (2)
F_MID3_R	5422	6A94	D596	6ACF	←	←	Behavior with floating-point operations (2)
F_MIN3_R	A524	31E1	551B	2950	←	←	Behavior with floating-point operations (2)
F_LIM_R	C92F	0A10	4017	B4BE	←	←	F-STOP instead of CPU-STOP (1) Behavior with floating-point operations (2) New input SUBS_IN, For function, see block description
F_SQRT	C412	895D	593F	CDDB	←	←	F-STOP instead of CPU-STOP (1) Behavior with floating-point operations (2)
F_AVEX_R	9926	8CE8	BE40	1CB3	←	←	F-STOP instead of CPU-STOP (1) Behavior with floating-point operations (2)
F_SMP_AV	FB42	5B98	9D24	9CDF	←	←	F-STOP instead of CPU-STOP (1) Behavior with floating-point operations (2)

F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_2oo3_R	—	—	FC09	3D43* 36CB	←	←	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 and higher *) This initial value signature is presented up to *S7 F Systems* V5.2 SP3 if the block container does not contain an F-Block that was called in the F-Block type.
F_1oo2_R	—	—	D100	6717* 2ED6	←	←	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 and higher *) This initial value signature is presented up to *S7 F Systems* V5.2 SP3 if the block container does not contain an F-Block that was called in the F-Block type.

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

2) If a floating-point operation produces an overflow (± infinity) or a denormalized or invalid floating-point number (NaN), or if an invalid floating-point number (NaN) is already present as an address, this event no longer results in a CPU-STOP. The "Overflow (± infinity)", "Denormalized floating-point number", or "Invalid floating-point number (NaN)" events are:

- Either output at the output and available for further processing by subsequent F-Blocks

  or

- Signaled at special outputs. If necessary, a fail-safe value is output.

If the floating-point operation yields an invalid floating-point number (NaN) and an invalid floating-point number (NaN) does not already exist as an address, the following diagnostic event is entered in the diagnostic buffer of the F-CPU:

- "Safety program: invalid REAL number in DB" (Event ID 16#75D9)

You can use this diagnostic buffer entry to identify the F-Block with the invalid floating-point number (NaN).

Refer also to the documentation of the F-Block.

If you cannot rule out the occurrence of these events in your safety program, you must decide based on your application whether you have to react to these events in your safety program. With F-Block F_LIM_R, you can check the result of a floating-point operation for overflow (± infinity) and invalid floating-point number.

F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_ADD_R	DFBF	B1DF	None
F_SUB_R	E217	B1DF	None
F_MUL_R	AA0F	B1DF	None
F_DIV_R	43F6	C0B8	None
F_ABS_R	7E9D	4885	None
F_MAX3_R	C14F	F93F	F-STOP when an error occurs in the safety data format in the instance DB
F_MID3_R	EC2C	EA98	F-STOP when an error occurs in the safety data format in the instance DB
F_MIN3_R	D0D7	E12A	F-STOP when an error occurs in the safety data format in the instance DB
F_LIM_R	B3D0	3957	None
F_SQRT	E621	6B0F	None
F_AVEX_R	E57D	947D	None
F_SMP_AV	5659	EEDA	None
F_2oo3_R	AB9F	112C	In *S7 F Systems Lib* V1_3 and higher, the F-Block is not an F-Block type Data type output DELTA is F_REAL
F_1oo2_R	DA53	AA5A	In *S7 F Systems Lib* V1_3 and higher, the F-Block is not an F-Block type Data type output DELTA is F_REAL

A.5.12 Arithmetic blocks with the INT data type

F-Block	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_LIM_I	5219	F4F9	0B0C	F4F9	←	←	F-STOP instead of CPU-STOP (1)

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

F-Block	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_LIM_I	4845	4D9B	None

A.5.13 Multiplex blocks

F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_MOV_R	—	—	—	—	—	—	—
F_MUX2_R	5911	5B43	7DE0	5B43	←	←	F-STOP instead of CPU-STOP (1)
F_MUX16R	—	—	—	—	—	—	—

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_MOV_R	652F	C02B	New F-Block in *S7 F Systems Lib* V1_3
F_MUX2_R	BFE3	9CB1	None
F_MUX16R	AF74	EEFE	New F-Block in *S7 F Systems Lib* V1_3

A.5.14 F-Control blocks

Blocks / F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_MOVRWS	—	—	—	—	—	—	—
F_MPA_I	—	—	—	—	F0D1	381B	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 SP4 and higher
F_DIAG	—	—	—	—	—	—	—
F_M_DI8	4996	640D	8FA4	9D22	5078	94DC	F-STOP instead of CPU-STOP (1) *S7 F Systems* V5.2 and higher, new outputs PROFISAFE1 and PROFISAFE2, For function, see block description *S7 F Systems* V5.2 SP1 and higher, change so that *S7-PLCSIM* can be used even without F-Simulation blocks
F_M_DI24	7DA1	0D91	EB16	1FE2	F887	2EAC	F-STOP instead of CPU-STOP (1) *S7 F Systems* V5.2 and higher, new outputs PROFISAFE1 and PROFISAFE2, For function, see block description *S7 F Systems* V5.2 SP1 and higher, change so that *S7-PLCSIM* can be used even without F-Simulation blocks
F_M_DO10	A89E	EE4E	22E8	EB44	6CA7	4A6E	F-STOP instead of CPU-STOP (1) *S7 F Systems* V5.2 and higher, new outputs PROFISAFE1 and PROFISAFE2, For function, see block description *S7 F Systems* V5.2 SP1 and higher, change so that *S7-PLCSIM* can be used even without F-Simulation blocks

Blocks / F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
F_M_AI6	3CC4	75CE	AF64	EC0D	1E41	D818	F-STOP instead of CPU-STOP (1) *S7 F Systems* V5.2 and higher, new outputs PROFISAFE1 and PROFISAFE2, For function, see block description *S7 F Systems* V5.2 SP1 and higher, change so that *S7-PLCSIM* can be used even without F-Simulation blocks
F_M_DO8	—	—	7337	3B1F	86EF	BD24	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 and higher *S7 F Systems* V5.2 SP1 and higher, change so that *S7-PLCSIM* can be used even without F-Simulation blocks
F_CYC_CO	3263	CB5D	E895	6769	←	←	F-STOP instead of CPU-STOP (1)
F_PLK	E5B4	D2F9	A234	5FA0	←	←	F-STOP instead of CPU-STOP (1)
F_PLK_O	53BE	3E43	D690	834C	←	←	F-STOP instead of CPU-STOP (1)
F_TEST	D774	A04B	5B6D	38AF	←	←	F-STOP instead of CPU-STOP (1)
F_TESTC	E7E8	711C	5A93	D8AA	←	←	F-STOP instead of CPU-STOP (1)
F_TESTM	2983	BED2	←	←	←	←	None
F_SHUTDN	—	—	Without	Without	Without	Without	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 and higher
RTGLOGIC	—	—	Without	Without	Without	Without	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 and higher
F_PS_12	—	—	—	—	—	—	—
F_CHG_WS	—	—	—	—	Without	Without	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 and higher
DB_INIT	—	—	Without	Without	Without	Without	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 SP2 or later

Blocks / F-Blocks	*Failsafe Blocks* (V1_1)		*Failsafe Blocks* (V1_2)				
			Shipped with *S7 F Systems* V5.2		Shipped starting with *S7 F Systems* V5.2 SP1 to SP4		
	Signature	Initial value signature	Signature	Initial value signature	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_1) to *Failsafe Blocks* (V1_2)
FAIL_MSG	—	—	Without	Without	Without	Without	New F-Block in *Failsafe Blocks* (V1_2) *S7 F Systems* V5.2 and higher
DB_RES	Without	Without	Without	Without	Without	Without	None
F_PS_MIX	—	—	—	—	—	—	—
F_VFSTP1	—	—	—	—	—	—	—
F_VFSTP2	—	—	—	—	—	—	—

1) A CPU-STOP is not triggered if a safety-related error is detected (e.g., in the safety data format). Instead, the shutdown logic shuts down either the F-Shutdown group affected by the error or the entire safety program (F-STOP).

Blocks / F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_MOVRWS	Without	Without	New block in *S7 F Systems Lib* V1_3
F_MPA_I	—	—	F-Block omitted as of *S7 F Systems Lib* V1_3 and is replaced with F_PS_12
F_DIAG	40FC	DDF4	New F-Block in *S7 F Systems Lib* V1_3
F_M_DI8	—	—	F-Block omitted as of *S7 F Systems Lib* V1_3 and is replaced with F_PS_12. In redundantly configured F-I/O, this F-Block is replaced with two instances of F_PS_12. Inputs DISC_ON, DISCTIME and RED in *S7 F Systems Lib* V1_3 and higher on F_CH_DI Discrepancy error information from output DIAG_1 and DIAG_2 in *S7 F Systems Lib* V1_3 and higher at F_CH_DI output DISCF or DISCF_R Output DIAG_1/2 and PROFISAFE1/2 are on F_PS_12 output DIAG and PROFISAFE, respectively.
F_M_DI24	—	—	F-Block omitted as of *S7 F Systems Lib* V1_3 and is replaced with F_PS_12. In redundantly configured F-I/O, this F-Block is replaced with two instances of F_PS_12. Inputs DISC_ON, DISCTIME and RED in *S7 F Systems Lib* V1_3 and higher on F_CH_DI Discrepancy error information from output DIAG_1 and DIAG_2 in *S7 F Systems Lib* V1_3 and higher at F_CH_DI output DISCF or DISCF_R Output DIAG_1/2 and PROFISAFE1/2 are on F_PS_12 output DIAG and PROFISAFE, respectively.

Blocks / F-Blocks	*S7 F Systems Lib* V1_3		
	Signature	Initial value signature	Change from *Failsafe Blocks* (V1_2) to *S7 F Systems Lib* V1_3
F_M_DO10	—	—	F-Block omitted as of *S7 F Systems Lib* V1_3 and is replaced with F_PS_12. In redundantly configured F-I/O, this F-Block is replaced with two instances of F_PS_12. Input RED in *S7 F Systems Lib* V1_3 and higher on F_CH_DO
F_M_AI6	—	—	F-Block omitted as of *S7 F Systems Lib* V1_3 and is replaced with F_PS_12. In redundantly configured F-I/O, this F-Block is replaced with two instances of F_PS_12. Inputs MODE_xx in *S7 F Systems Lib* V1_3 and higher on F_CH_AI as input MODE Input RED in *S7 F Systems Lib* V1_3 and higher on F_CH_AI
F_M_DO8	—	—	F-Block omitted as of *S7 F Systems Lib* V1_3 and is replaced with F_PS_12. In redundantly configured F-I/O, this F-Block is replaced with two instances of F_PS_12. Input RED in *S7 F Systems Lib* V1_3 and higher on F_CH_DO
F_CYC_CO	701D	424E	None
F_PLK	CD05	A65D	None
F_PLK_O	45F2	7B78	None
F_TEST	EC5F	EB03	None
F_TESTC	680A	38BA	None
F_TESTM	8B5A	9A74	Message behavior is taken over from F_SHUTDN in *S7 F Systems Lib* V1_3 and higher
F_SHUTDN	Without	Without	New output SD_TYP, New input/output MSG_TIME, For function, see block description Parameter assignment at input SHUTDOWN is only relevant if "Based on F_SHUTDN parameter assignment" is specified for the F-STOP behavior in the "Safety Program" dialog > "Shutdown Behavior" dialog. See block description
RTGLOGIC	Without	Without	Name changed from RTG_LOGIC to RTGLOGIC
F_PS_12	A56A	B87A	New F-Block in *S7 F Systems Lib* V1_3
F_CHG_WS	Without	Without	None
DB_INIT	Without	Without	None
FAIL_MSG	—	—	Block omitted as of *S7 F Systems Lib* V1_3
DB_RES	Without	Without	None
F_PS_MIX	AD87	Without	New F-Block in *S7 F Systems Lib* V1_3
F_VFSTP1	Without	Without	New block in *S7 F Systems Lib* V1_3
F_VFSTP2	Without	Without	New block in *S7 F Systems Lib* V1_3

A.6 Differences between the F-Library S7 F Systems Lib V1_3 and V1_3 SP1

The following subsections describe the differences between the F-Library *S7 F Systems Lib* V1_3 and V1_3 SP1. Only those F-Block changes that are relevant to the user and that affect the function, including the startup behavior and error handling, and the inputs/outputs of the F-Block are described.

Even if no changes (i.e., "none") are indicated, it is possible that the signatures/initial value signatures of an F-Block have changed compared to a previous version of the F-Library, for example, due to code optimizations, changes in diagnostic buffer entries, or changes in the internal interaction of the F-Blocks.

For information about the runtimes of the F-Blocks, refer to the section entitled "Run times, F-Monitoring times, and response times (Page 410)". If required, you can find out the new memory requirements from *SIMATIC Manager*.

When you upgrade to a new version of the F-Library, take note of the F-Block changes and check whether these changes may affect the behavior of your safety program. Refer also to the section entitled "Acceptance test of safety program changes (Page 184)".

Refer to Annex 1 of the Certification Report to obtain the signatures/starting value signatures for the F-Blocks of F-Library *S7 F Systems Lib* V1_3 SP1.

F-Blocks	*S7 F Systems Lib* V1_3 SP1		Delta download-capable	Change from *S7 F Systems Lib* V1_3 to V1_3 SP1
	Signature	Initial value signature		
F_FR_FDI	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1
F_FDI_FR	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1
F_QUITES	Annex 1	Annex 1	Yes	None
F_CHG_BO	D042 *	E5F2 *	Yes	None
F_CHG_R	E4CD *	5DB5 *	Yes	None
F_CH_BI	Annex 1	Annex 1	Yes	IPAR_EN and IPAR_OK visible
F_CH_BO	A8C7 *	A5E4 *	Yes	IPAR_EN and IPAR_OK visible
F_PA_AI	Annex 1	Annex 1	Yes	IPAR_EN and IPAR_OK visible and update of V_MOD
F_PA_DI	2FC7 *	E4F2 *	Yes	IPAR_EN and IPAR_OK visible
F_CH_DO	Annex 1	Annex 1	With this change, the F-Channel driver F_CH_DO in *S7 F Systems* V6.1 or earlier can no longer be compiled.	The output of ACK_REQ has been delayed.
F_CH_AI	Annex 1	Annex 1	Yes	IPAR_EN and IPAR_OK visible and update of V_MOD and AL_STATE
F_CH_II	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1

F-Blocks	*S7 F Systems Lib* V1_3 SP1		Delta download-capable	Change from *S7 F Systems Lib* V1_3 to V1_3 SP1
	Signature	Initial value signature		
F_CH_IO	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1
F_CH_DII	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1
F_CH_DIO	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1
F_SQRT	Annex 1	Annex 1	Yes	None
F_POLYG	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1
F_INT_P	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1
F_PT1_P	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1
F_SWC_P	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1
F_SWC_BO	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1
F_SWC_R	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1
SWC_MOS	Without	Without	—	New block in S*7 F Systems* Lib V1_3 SP1
F_DEADTM	Annex 1	Annex 1	—	New block in S*7 F Systems* Lib V1_3 SP1
FORCEOFF	Annex 1	Annex 1	—	New F-Block in S*7 F Systems* Lib V1_3 SP1
* The change made was not relevant to the signature; therefore, the signatures have not changed.				

A.7 Run times, F-Monitoring times, and response times

Excel table S7FTIMEB.XLS contains information regarding:

- Execution times of F-Blocks in the various F-CPUs and aids for their calculation
- Maximum runtime of an F-Shutdown group
- Minimum F-Monitoring times
- Maximum response time of your F-System

This file is available for download on the Web (http://support.automation.siemens.com/WW/view/en/22557362).

See also

Safety engineering in SIMATIC S7 System Manual (http://support.automation.siemens.com/WW/view/en/12490443)

Checklist

B

Introduction

The table below contains a checklist summarizing all activities in the life cycle of a fail-safe S7 F/FH System, including requirements and rules that must be observed in the various phases.

Checklist

Key:

- Stand-alone section references refer to this documentation.
- "*SM*" refers to System Manual " Safety Engineering in SIMATIC S7 (http://support.automation.siemens.com/WW/view/en/12490443) ".
- "*F-SMs Manual*" refers to Manual " Automation System S7-300 Fail-Safe Signal Modules (http://support.automation.siemens.com/WW/view/en/19026151) ".
- "*ET 200S Manual*" refers to Manual " Distributed I/O System ET 200S, Fail-Safe Modules (http://support.automation.siemens.com/WW/view/en/12490437) ".
- "*ET 200pro Manual*" refers to Manual " ET 200pro Distributed I/O Device - Fail-Safe Modules (http://support.automation.siemens.com/WW/view/en/22098524) ".
- "*ET 200eco Manual*" refers to Manual " ET 200eco Distributed I/O Station Fail-safe I/O Module (http://support.automation.siemens.com/WW/view/en/19033850) ".

Phase	Requirement/Rule	Reference	Check
Planning			
Requirement: A specification with the safety requirements must be available for the planned application.	Process-dependent	—	
Specification of the system architecture	Process-dependent	—	
Assignment of functions and subfunctions to the system components	Process-dependent	Section 1 *SM*, Section 1.5 *SM*, Section 2.4	
Selection of sensors and actuators	Requirements for actuators	*SM*, Section 4.8 *F-SMs Manual*, Section 6.5 *ET 200S Manual*, Section 4.5 *ET 200pro Manual*, Section 4.4 *ET 200eco Manual*, Section 5.5	
Definition of required safety properties of the components	DIN V 19250 IEC 61508	*SM*, Sections 4.7, 4.8	

Phase	Requirement/Rule	Reference	Check
Configuration			
Installation of optional package	Requirements for installation	Section 2.1	
Selection of S7 components	Rules for configuration	Section 1.2 *SM*, Section 2.4 *F-SMs Manual*, Section 3 *ET 200S Manual*, Section 3 *ET 200pro Manual*, Section 2 *ET 200eco Manual*, Section 3	
Configuration of hardware	Rules for S7 F/FH Systems Verification of utilized hardware components based on Annex 1 of Certification Report	Section 3 Annex 1 of Certification Report	
Configuring the F-CPU	Protection level, "CPU contains safety program" Password	Sections 3.3, 4 Manual for Standard S7-400(H)	
Configuring the F-I/O	Settings for safety mode Configuring the monitoring times Module redundancy (optional) Defining the type of sensor interconnection/evaluation	Sections 3.2, 3.4-3.8 *SM*, Appendix A *F-SMs Manual*, Sections 3, 9, 10 *ET 200S Manual*, Sections 2.4, 7 *ET 200pro Manual*, Sections 2.4, 8 *ET 200eco Manual*, Sections 3, 8	
Programming			
Defining program design and structure	Warnings and notes on programming Verifying the utilized software components based on Annex 1 of Certification Report	Sections 5.1, 5.2, 5.6 Annex 1 of Certification Report	
Inserting the CFC charts	Rules for the CFC charts of the safety program	Sections 5.2.4ff, 5.3, 5.7	
Inserting F-Runtime groups	Rules for F-Runtime groups of the safety program	Sections 5.2.7, 5.3	
Defining F-Shutdown groups	Rules for F-Shutdown groups of the safety program	Section 5.2.8	

Phase	Requirement/Rule	Reference	Check
Inserting and interconnecting the F-Blocks	Rules for F-Blocks	Section 5, Appendix A	
	Rules for F-Channel drivers and module drivers	Section 6	
	Rules for interconnecting the F-Block F_CYC_CO	Section 5.2.3 *SM*, Appendix A	
	Rules for safety-related communication between F-CPUs	Section 7	
	Configuring the F-Monitoring times	Section 5.2.3, Appendix A.6 *SM*, Appendix A	
	Startup characteristics	Section 5.5	
	Creating F-Block types	Section 5.7	
	Passivation and reintegration	Sections 6.3, 6.4	
	Data exchange between F-Shutdown groups	Section 5.8	
	Data exchange with standard user program	Section 5.9	
	Changing F-Parameters from one OS	Section 8	
	User acknowledgement	Section 5.10	
Compiling the safety program	Rules for compiling	Section 10.1	
Installation			
Hardware configuration	Rules for mounting Rules for wiring	Section 12.2 *F-SMs Manual*, Sections 5, 6 *ET 200S Manual*, Sections 3, 4 *ET 200pro Manual*, Sections 2, 3 *ET 200eco Manual*, Sections 3, 4	
Commissioning, Testing			
Powering up	Rules for commissioning (in standard case)	Manual for Standard S7-400(H)	
Downloading the safety program	Rules for downloading	Sections 10.6, 10.8	
Testing the safety program	Rules for deactivating safety mode Rules for testing the safety program	Sections 10.5.1, 10.7	
Changing the safety program	Rules for deactivating safety mode	Section 10.5.1	
	Rules for changing the safety program	Sections 10.3, 10.8	
Check of safety-related parameters	Rules for configuration	Sections 10.4, 11 *F-SMs Manual*, Sections 4, 9, 10 *ET 200S Manual*, Sections 2.4, 7 *ET 200pro Manual*, Sections 2.4, 8 *ET 200eco Manual*, Sections 3, 8	
Acceptance test	Rules and notes on the acceptance test Generating printouts	Section 11	

Phase	Requirement/Rule	Reference	Check
Operation, Maintenance			
General operation	Notes on operation	Section 12	
Access protection		Section 4	
Diagnostics	Responses to faults and events	Appendix A	
Replacement of software and hardware components	Rules for module replacement Rules for updating the operating system of the F-CPU - same as for standard system Rules for updating software components Notes on IM operating system update Notes on preventive maintenance	Sections 2.3, 12.2, Manual for Standard S7-400(H)	
Removing, disassembly	Notes for removing software components Notes for disassembling modules	Sections 2.2, 12.2	

Glossary

1oo1 evaluation

Type of -> sensor evaluation: In the case of 1oo1 evaluation, a non-redundant sensor is connected via one channel to the -> F-I/O.

1oo2 evaluation

Type of -> sensor evaluation: In the case of 1oo2 evaluation, two input channels are occupied either by one two-channel sensor or by two single-channel sensors. The input signals are compared internally for equality (equivalence) or inequality (nonequivalence).

Access protection

-> Fail-safe systems must be protected against dangerous, unauthorized access. Access protection for F-Systems is implemented by assigning two passwords (for the -> F-CPU and for the -> safety program).

Bypass

Bypass function that is normally used for maintenance purposes (e.g., for checking effect logic, replacing a sensor).

Category

Category as defined by EN 954-01

S7 F Systems can be used in -> safety mode up to Category 4.

Channel fault

Channel-specific fault, such as a wire break or a short circuit.

Collective signatures

Collective signatures uniquely identify a particular state of the -> safety program. They are important for the preliminary acceptance test of the safety program, e.g., by experts.

CRC

Cyclic Redundancy Check -> CRC signature

CRC signature

The validity of the process data in the -> safety message frame, the accuracy of the assigned address references, and the safety-related parameters are ensured by means of a CRC signature contained in the -> safety message frame.

Deactivated safety mode

Deactivated safety mode is the temporary deactivation of -> safety mode for test purposes, commissioning, etc.

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.

Depassivation

-> Reintegration

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 extended 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.

ES

Engineering **S**ystem (ES): Configuration system that enables convenient, visual adaptation of the process control system to the task at hand.

Fail-safe DP standard slaves

Fail-safe DP standard slaves are standard slaves that are operated on PROFIBUS with the DP protocol. They must behave in accordance with IEC 61784-1:2002 Ed1 CP 3/1 and the PROFIsafe bus profile. A GSD file is used for your configuration.

Fail-safe I/O modules

ET 200eco modules that can be used for safety-related operation (in -> safety mode). These modules are equipped with integrated -> safety functions. They behave according to IEC 61784-1:2002 Ed1 CP 3/1 and the PROFIsafe bus profile.

Fail-safe modules

ET 200S modules that can be used for safety-related operation (-> safety mode) in the ET 200S or ET 200pro distributed I/O system. These modules are equipped with integrated -> safety functions. They behave according to IEC 61784-1:2002 Ed1 CP 3/1 and CP 3/3 and the PROFIsafe bus profile.

Fail-safe PA field devices

Fail-safe PA field devices are field devices that are operated on PROFIBUS with the PA protocol. They must behave in accordance with IEC 61784-1:2002 Ed1 CP 3/2 and the PROFIsafe bus profile. A GSD file is used for your configuration.

Fail-safe systems

Fail-safe systems (F-Systems) are systems that remain in a -> safe state or immediately switch to another safe state when particular failures occur.

Fault reaction function

-> User safety function

F-Block type

F-Block types are ready-made program sections that can be used in a CFC chart (e.g., fail-safe multiplexer F_MUX2_R, etc.). Block instances are generated on insertion. Any number of block instances can be created by one F-Block type.

The F-Block type specifies the characteristics (algorithm) for all applications of this type. The name of the F-Block type is specified in the symbol table.

F-Blocks

The following fail-safe blocks are designated as F-Blocks:

- Blocks selected by the user from an F-Library
- Blocks that are automatically added in the -> safety program

F-CPU

An F-CPU is a central processing unit with fail-safe capability that is permitted for use in *S7 F Systems*. For *S7 F Systems*, the F-Runtime license allows the user to operate the central processing unit as an F-CPU. That is, a -> safety program can be run on it. A -> standard user program can also be run in the F-CPU.

F-Cycle time

Cyclic interrupt time for OBs with -> F-Runtime groups

F-Data type

The standard user program and -> safety program use different data formats. Safety-related F-Data types are used in the safety program.

F-I/O

Group designation for fail-safe inputs and outputs available in *SIMATIC S7* for integration in *S7 F Systems*, among others. The following are available for *S7 F Systems*:

- ET 200eco fail-safe I/O modules
- S7-300 fail-safe signal modules (-> F-SMs)
- ET 200pro fail-safe modules
- -> Fail-safe modules for ET 200S
- -> Fail-safe DP standard slaves
- -> Fail-safe PA field devices

F-Runtime group

When the -> safety program is created, the -> F-Blocks cannot be inserted directly into tasks/OBs; rather, they must be inserted into F-Runtime groups. The -> safety program consists of multiple F-Runtime groups.

F-Shutdown groups

F-Shutdown groups contain one or more -> F-Runtime groups. F-Runtime group communication blocks between the -> F-Blocks in various F-Runtime groups, all of which are assigned to one F-Shutdown group, are not required. If an error is detected in an F-Shutdown group, this F-Shutdown group is shut down. Additional F-Shutdown groups are shut down according to the configuration of F_SHUTDN.

F-SMs

S7-300 fail-safe signal modules that can be used for safety-related operation (in -> safety mode) as centralized modules in an S7-300 or as distributed modules in the ET 200M distributed I/O system. F-SMs are equipped with integrated -> safety functions.

F-Startup

An F-Startup is a restart following an F-STOP or an F-CPU STOP. *S7 F Systems* do not distinguish between a cold restart and warm restart of the F-CPU.

F-Systems

Fail-safe systems

Full shutdown

All F-Blocks of the entire F-CPU are shut down. Initially, the F-Shutdown group in which the error was detected is shut down. All other F-Shutdown groups are then shut down within a period of time equal to twice the F-Monitoring time you assigned for the slowest OB.

Master-reserve switchover

In S7 FH Systems, a master-reserve switchover is triggered in the event of an F-STOP of the master. That is, there is a switchover from the master CPU to the reserve CPU.

Module redundancy

The module and a second identical module are operated in redundant mode in order to enhance availability.

OS

Operator **S**tation (OS): A configurable operator station used to operate and monitor machines and systems.

Partial shutdown

Only the F-shutdown group in which the error was detected is shut down.

Passivation

Passivation of digital output channels means that the outputs are de-energized.

Digital input channels are passivated when the inputs transmit a value of "0" to the F-CPU (by means of the fail-safe drivers), irrespective of the current process signal.

Analog input channels are passivated when the inputs transmit a fail-safe value or the last valid value to the F-CPU (by means of the fail-safe drivers), irrespective of the current process signal.

Process safety time

The process safety time of a process is the time interval during which the process can be left on its own without risk to life and limb of the operating personnel or damage to the environment.

Within the process safety time, any type of F-System process control is tolerated. That is, during this time, the -> F-System can control its process incorrectly or it can even exercise no control at all. The process safety time depends on the process type and must be determined on a case-by-case basis.

PROFIsafe

Safety-related bus profile of PROFIBUS DP/PA and PROFINET IO for communication between the -> Safety program and the -> F-I/O in an > F-System.

Proof-test interval

Period after which a component must be forced to fail-safe state, that is, it is either replaced with an unused component, or is proven faultless.

Redundancy, availability-enhancing

Multiple instances of components with the goal of maintaining component function even in the event of hardware faults.

Redundancy, safety-enhancing

Multiple availability of components with the focus set on exposing hardware faults based on comparison; for example, → 1oo2 evaluation in fail-safe signal modules.

Reintegration

Switchover from fail-safe values (0) to process data (reintegration of an F-I/O module) occurs automatically or, alternatively, only after user acknowledgment at the F-Channel driver.

The reintegration method depends on the following:

- Cause of passivation of the F-I/O or channels of the F-I/O
- Parameter assignment for the F-Channel driver

For an F-I/O with inputs, the process values pending at the fail-safe inputs are provided again at the output of the F-Channel driver after reintegration. For an F-I/O with outputs, the F-System again transfers the output values pending at the input of the F-Channel driver to the fail-safe outputs.

S7 F Systems RT License (Copy License)

Formal authorization for use of the CPU as an F-CPU for S7 F/FH Systems.

S7-PLCSIM

The *S7-PLCSIM* application enables you to execute and test your S7 program on a simulated automation system on your ES/OS. Because the simulation takes place entirely in STEP 7, you do not require any hardware (CPU, F-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, the safe state is always the value "0".

Safety class

Safety Integrity Level (SIL) is the safety level defined in IEC 61508 and prEN 50129. The higher the Safety Integrity Level is, the more stringent the actions are for avoiding and controlling system faults and random hardware failures.

S7 F Systems can be used in safety mode up to safety class SIL3.

Safety function

Mechanism built into the -> F-CPU and -> F-I/O that allows them to be used in -> fail-safe systems.

In accordance with IEC 61508: Function implemented by a safety device in order to maintain the system in a -> safe state or to place it into a safe state in the event of a particular fault (-> user safety function).

Safety message frame

In -> safety mode, data are transferred between the -> F-CPU and -> F-I/O or between the F-CPUs in safety-related CPU-CPU communication in a safety message frame.

Safety mode

1. Safety mode is the operating mode of the -> F-I/O that allows -> safety-related communication by means of -> safety message frames.
2. Operating mode of the safety program. In safety mode of the safety program, all safety mechanisms for fault detection and fault reaction are activated. In safety mode, the safety program cannot be modified during operation. Safety mode can be deactivated by the user (-> deactivated safety mode).

Safety program

Safety-related user program

Safety protocol

-> Safety message frame

Safety-related communication

Communication used to exchange fail-safe data.

Sensor evaluation

There are two types of sensor evaluation:

- 1oo1 evaluation – sensor signal is read in once
- 1oo2 evaluation - sensor signal is read in twice by the same ->F-I/O and compared internally

Signature

-> Collective signatures

Standard communication

Communication used to exchange non-safety-related data.

Standard mode

Operating mode of -> F-I/O in which -> safety-related communication by means of -> safety message frames is not possible, but rather only -> standard communication.

Standard user program

Non-safety-related user 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 a fault in which the -> F-System can no longer execute its actual user safety function, it will execute the fault reaction function: For example, the associated outputs are deactivated and the -> F-CPU switches to STOP mode if necessary.

Index

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

SIEMENS

Siemens AG

I IA AS SM ID
Postfach 1963
D-92209 Amberg

Telefax: +49(9621)80-3103
mailto:doku.automation@siemens.com

Your Address:

Name:
Company:
Position:
Street:
Postal code / Place:
Email:
Phone:
Fax:

Your Feedback as regards the S7 F/FH Systems

Dear SIMATIC user,

Our goal is to provide you information with a high degree of quality and usability, and to continuously improve the SIMATIC documentation for you. To achieve this goal, we require your feedback and suggestions. Please take a few minutes to fill out this questionnaire and return it to me by Fax, e-mail or by post.

We are giving out three presents every month in a raffle among the senders. Which present would you like to have?

☐ **SIMATIC Manual Collection** ☐ **Automation Value Card** ☐ **Laser pointer**

Dr. Thomas Rubach,
Head of Information & Documentation

General Questions

1. Are you familiar with the SIMATIC Manual Collection?

☐ yes ☐ no

2. Have you ever downloaded manuals from the internet?

☐ yes ☐ no

3. Do you use Getting Starteds?

☐ yes ☐ no

if yes, which:

4. How much experience do you have with the Fail-Safe Systems?

☐ Expert
☐ Experienced user
☐ Advanced user
☐ Beginner

Please specify the documents, for which you want to answer the questions below:

- ☐ **A: Programming and Operating Manual S7 F/FH Systems Configuring and Programming**
- ☐ **B: Manual S7-300, Fail-Safe Signal Modules**
- ☐ **C: Installation and Operating Manual ET 200S, Distributed I/O System Fail-Safe Modules**
- ☐ **D: Manual ET 200eco, Distributed I/O Fail-Safe I/O Module**
- ☐ **E: System Manual Safety Engineering in SIMATIC S7**
- ☐ **F: Operating Instructions ET 200pro Distributed I/O Device - Fail-Safe Modules**

1. **In which project phase do you use this document frequently?**

☐ Information ☐ Assembly
☐ Planning ☐ Commissioning
☐ Configuration ☐ Maintenance & Service
☐ Programming ☐ others:

2. **Finding the required information in the document:**

- How quickly can you find the desired information in the document?

☐ immediately ☐ not at all
☐ after a brief search ☐ after a long search

- Which search method do you prefer?

☐ Table of contents ☐ Index
☐ Full-text search ☐ others:

- Which supplements/improvements would you like in order to help you find the required information quickly?

3. **Your judgement of the document as regards content.**

- How satisfied are you with this document?

☐ Totally satisfied ☐ not very satisfied
☐ Very satisfied ☐ not satisfied
☐ Satisfied

- Were you able to find the required information?

☐ yes ☐ no

which was not:

4. **What is the scope of the information?**

☐ Just right
☐ Not enough - which topic:
☐ Too detailed – which topic:

5. **Is the information easy to understand (texts, figures, tables)?**

☐ yes ☐ no

if no, which was not:

6. **Are examples important to you?**

☐ no, of less importance
☐ yes, important –were the examples enough?

☐ yes ☐ no

if no, on which topic:

7. **What are your suggestions as regards the contents of the document?**

Thank you for your cooperation