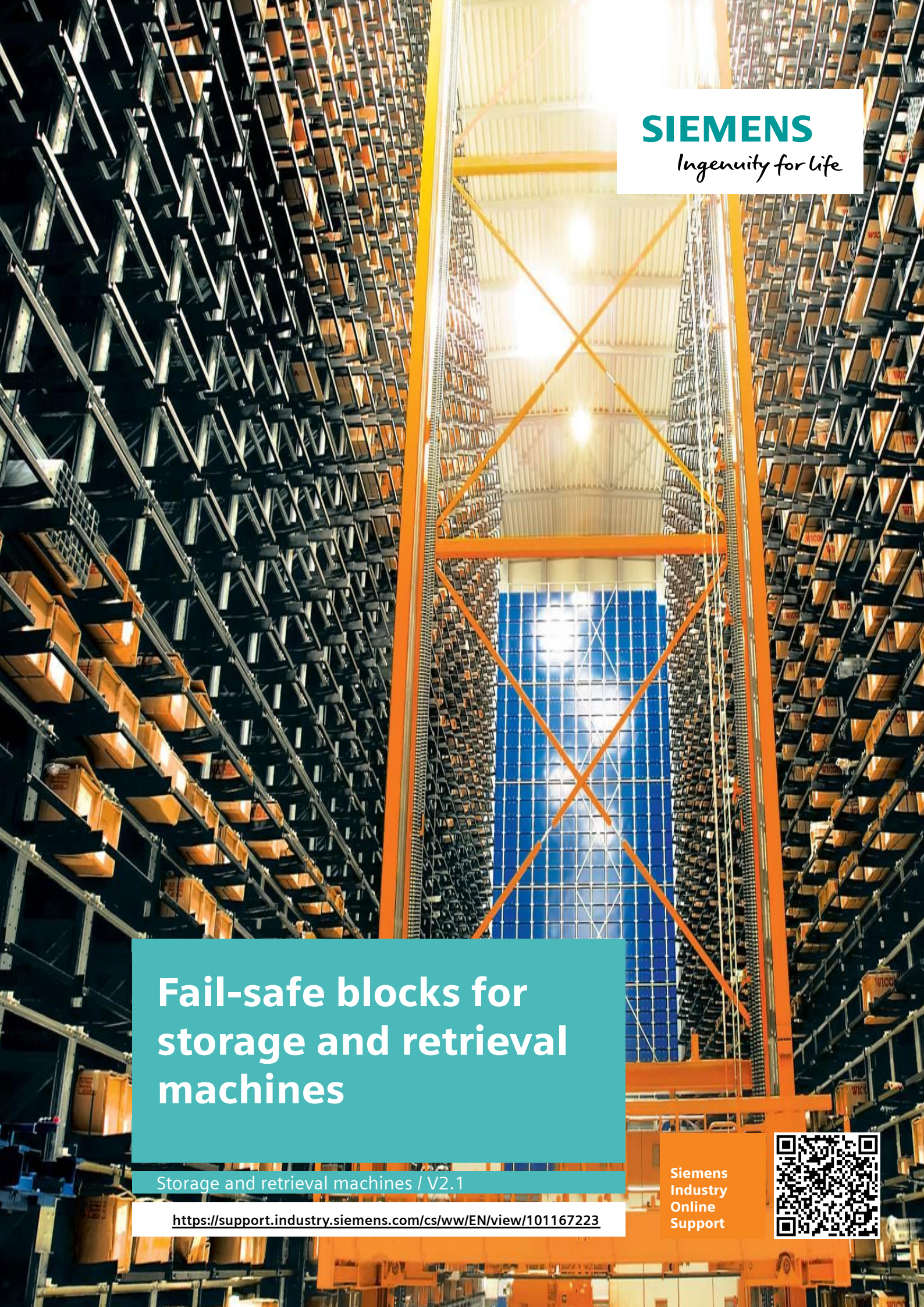


SIEMENS
Ingenuity for life

Fail-safe blocks for storage and retrieval machines

Storage and retrieval machines / V2.1

https://support.industry.siemens.com/cs/ww/EN/view/101167223

Siemens Industry Online Support

Warranty and liability

Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These Application Examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice.
If there are any deviations between the recommendations provided in these Application Examples and other Siemens publications − e.g. Catalogs − the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.
Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act ("Produkthaftungsgesetz"), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract ("wesentliche Vertragspflichten"). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.
Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of the Siemens AG.

Security information

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.
Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.
Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit http://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under **Fehler! Hyperlink-Referenz ungültig.**.

Table of contents

Warranty and liability 2

1 Storage and retrieval machines and safety functions 5

1.1 General design and structure for safe position sensing 5
1.2 Encoder combinations and configuration versions that are supported 6
1.2.1 Safety-related motor encoder with directly connected, positive locking mechanical system 6
1.2.2 Two-encoder system with connection via SINAMICS S120 6
1.2.3 Two-encoder system with connection via distributed I/O 7
1.2.4 Three-encoder system 8
1.2.5 Summary of the encoder versions 9
1.2.6 Safety-related parameters of the encoder versions 10

2 System and software requirements 15

2.1 General 15
2.2 Safety requirements 15
2.3 Software 15
2.4 Safety aspects when creating blocks 16
2.5 Standards complied with 16
2.5.1 Demarcation of EN 528:2008 with respect to the RBG block library 16

3 Fail-safe function blocks for storage and retrieval machines 19

3.1 Overview 19
3.1.1 Safety note 19
3.1.2 Fail-safe blocks 20
3.1.3 Block connections 21
3.1.4 Block numbers and signatures 21
3.1.5 Integration in cyclic interrupts - F-OB 21
3.1.6 Using instance data blocks/multi-instances 22
3.1.7 Response times 22
3.1.8 Runtimes 22
3.2 F_SAFE_POS 23
3.2.1 Introduction 23
3.2.2 Connections 24
3.2.3 Interrelationship between the assignment of the block inputs and the drive configuration 28
3.2.4 Principle of operation 33
3.3 F_SLP_MONITOR 41
3.3.1 Introduction 41
3.3.2 Connections 41
3.3.3 Principle of operation 44
3.4 F_ENDZONE 47
3.4.1 Introduction 47
3.4.2 Connections 47
3.4.3 Principle of operation 51
3.5 F_SBR_MONITOR 56
3.5.1 Introduction 56
3.5.2 Connections 56
3.5.3 Principle of operation 59
3.6 F_BRAKE_TEST 61
3.6.1 Introduction 61
3.6.2 Connections 62
3.6.3 Principle of operation 67
3.6.4 Application example for safely controlling external brakes 72

3.7 F_LOAD_MONITOR ... 75
3.7.1 Introduction ... 75
3.7.2 Connections ... 76
3.7.3 Scaling the input variables ... 79
3.7.4 Principle of operation ... 79
3.8 F_MIN_MAX ... 83
3.8.1 Introduction ... 83
3.8.2 Connections ... 83
3.8.3 Principle of operation ... 83
3.9 F_INTERPOLATION ... 85
3.9.1 Introduction ... 85
3.9.2 Connections ... 86
3.9.3 Principle of operation ... 86
3.10 LFAddDInt/LFSubDInt/LFMulDInt/LFDivDInt ... 88
3.10.1 Introduction ... 88
3.10.2 Connections ... 88
3.10.3 Principle of operation ... 89
LFAddDInt ... 89
LFSubDInt ... 89
LFMulDInt ... 89
LFDivDInt ... 89

4 Interaction between the blocks ... 90

4.1 Overview ... 90
4.2 Signal flow between components ... 90
4.2.1 Automation task ... 90
4.3 Response in the case of an error ... 91
4.4 Block interconnection ... 92
4.4.1 1-Encoder variant ... 92
4.4.2 2- and 3- encoder variant ... 93
4.4.3 Additionally required blocks ... 94
4.4.4 Additional information ... 94

5 Abbreviations ... 95

6 Support ... 96

7 Appendix ... 97

8 Related literature ... 98

9 History ... 98

1 Storage and retrieval machines and safety functions

This chapter provides a schematic overview of the application conditions of fail-safe function blocks for storage and retrieval machines and the supported hardware platforms.

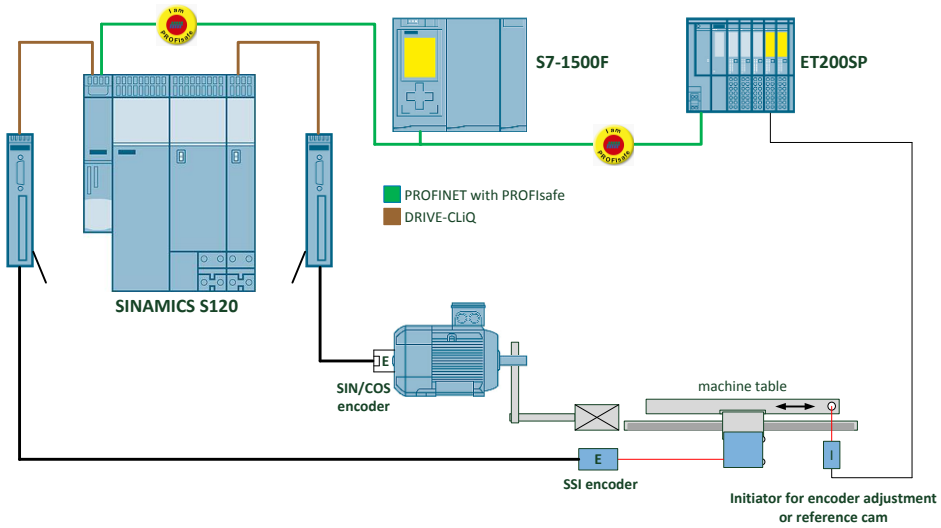
1.1 General design and structure for safe position sensing

The following components are essentially required for the use of fail-safe blocks (known as RBG blocks in the following) for storage and retrieval machines, depending on the expansion stage and the functions that are used.

- Fail-safe SIMATIC S7 control – STEP7 Safety Advanced
- SINAMICS S120 converters with CU320-2 (from firmware release 4.6), known as SINAMICS S120 in the following, with encoder, e.g. connected to
  - SMC20/SMC30
  - or connected via DRIVE-CLiQ.
- PROFIBUS/PROFINET data transfer between SINAMICS and F-CPU
- F-DQ module to control the brakes
- External mechanical brake and/or motor holding brake
- Signal source for load measurement for overload/slack cable detection, e.g. via F_AI with qualified encoder or two encoders, which mutually check signal plausibility (e.g. weighing cell and motor torque)

A typical hardware configuration looks like this:

Figure 1-1: Hardware configuration



The package of blocks covers several versions of encoder combinations, also see Table 1-1. As a consequence, the following scenarios are obtained regarding the required components. These can differ depending on the specific application, however, they must be comparable from a safety-related perspective.

1.2 Encoder combinations and configuration versions that are supported

An overview of the encoder combinations, supported by the storage and retrieval machine blocks, is provided in the following.

1.2.1 Safety-related motor encoder with directly connected, positive locking mechanical system

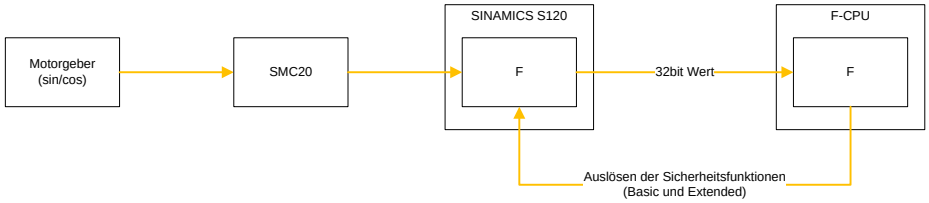
Sensing:

- Safety-related SIN/COS motor encoder, mounted in a safety-related way and connected via PROFIsafe telegram 902 from SINAMICS S120.
- The absolute position is transferred to the F-CPU, e.g. via a standard telegram from the SINAMICS S120.

SI determines the absolutes position actual value of the motor encoder, which is then transferred to the F-CPU via a safety-related telegram. The motor encoder must be a safety encoder (safety-related motor encoder that is mounted in a safety-related fashion).

As a consequence, the signal flow of the safety function looks like this:

Figure 1-2: Version 1: Safety-related motor encoder with directly connected, positive locking mechanical system



Safety note

The safety motor encoder must be safely referenced in the Safety Integrated functions of the SINAMICS S120. The safety-related absolute position actual value, the validity of the encoder and the status of the safety-relevant reference must be directly used from PROFIsafe telegram 902. These signals must be interconnected according to the interconnection example in Chapter 4.4.1.

1.2.2 Two-encoder system with connection via SINAMICS S120

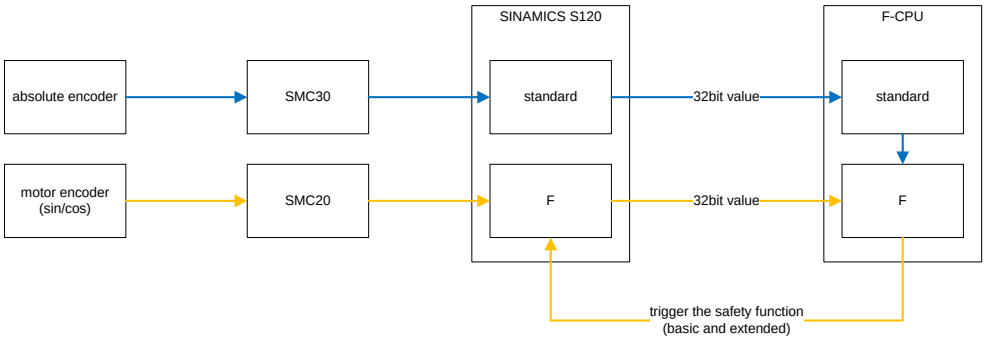
Sensing:

- The SIN/COS motor encoder is connected to the SINAMICS S120 via an SMC20 or a DRIVE-CLiQ interface (SMI), direct measuring system (SSI) e.g. via a SMC30. The closed-loop control is realized via the direct measuring system.

In order to achieve two-channel data transfer in the F-CPU, the position actual value of the direct encoder is transferred using the standard telegram. SI determines the position actual value of the motor encoder, which is then transferred to the F-CPU via a safety-related telegram. The motor encoder must satisfy the require-

ments for SINAMICS Safety Integrated. Safety-related mounting is not required, as possible faults/errors are monitored by the crosswise comparison of two encoders.

Figure 1-3: Version 2: Two-encoder system with connection via SINAMICS S120

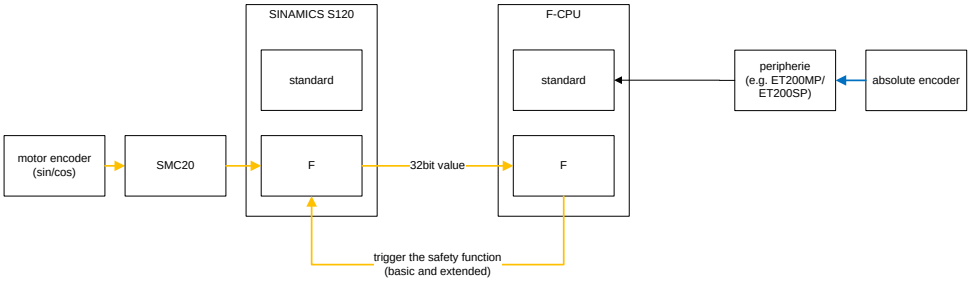


1.2.3 Two-encoder system with connection via distributed I/O

Sensing:

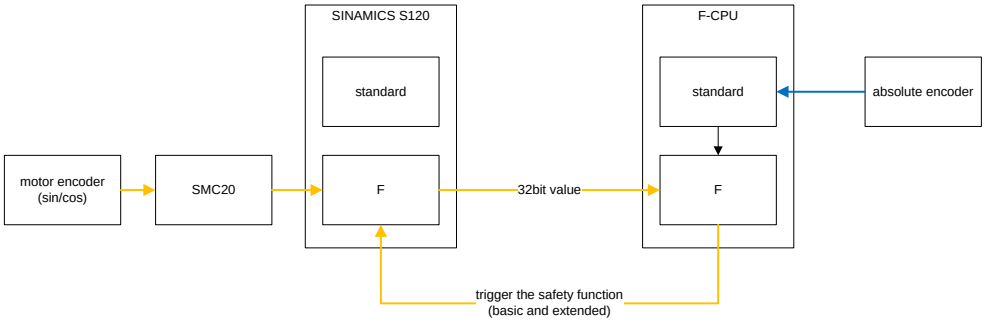
a) SIN/COS motor encoder (safety-related mounting is not required) corresponding to the requirements of SINAMICS Safety Integrated (e.g. via SMC20 or DQI/SMI) via PROFIsafe telegram from SINAMICS S120, direct measuring system via standard telegram from the SSI module (e.g. TM PosInput 2) at the F-CPU.

Figure 1-4: Version 3a): Two-encoder system, connection via distributed I/O



b) SIN/COS motor encoder (safety-related mounting is not required) corresponding to the requirements of SINAMICS Safety Integrated (e.g. via SMC20 or DQI/SMI) via PROFIsafe telegram from SINAMICS S120, direct encoder via standard telegram PROFIBUS/PROFINET-capable encoder.

Figure 1-5: Version 3 b): Two-encoder system, connection via direct encoder



1.2.4 Three-encoder system

Sensing:

a) SIN/COS motor encoder (safety-related mounting is not required) corresponding to the requirements of SINAMICS Safety Integrated (e.g. via SMC20 or DQI/SMI) via PROFIsafe telegram from SINAMICS S120.

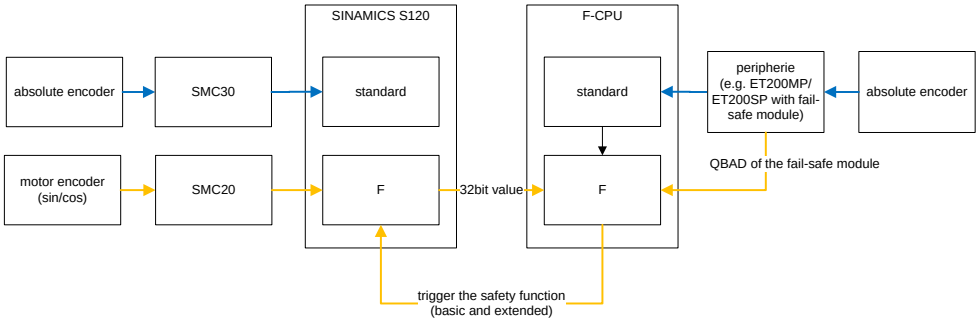
Two direct measuring systems via standard telegram (Figure 1-6):

- Position 1 via SINAMICS S120
- Position 2 from distributed I/O with safety-related communication via F module

Alternatively:

- Position 1 via SINAMICS S120 with safety-related communication via PROFIsafe telegram of the converter
- Position 2 from distributed I/O

Figure 1-6: Version 4 a): Three-encoder system, position encoder via SINAMCS 120 and distributed PLC I/O, secure communication via F module

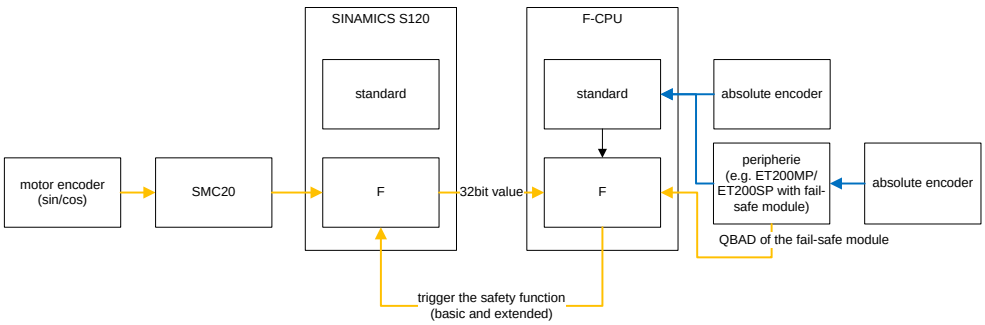


b) SIN/COS motor encoder corresponding to the requirements of SINAMICS Safety Integrated via PROFIsafe telegram from SINAMICS S120.

Two direct measuring systems via standard telegram:

- Positions 1 and 2 via distributed I/O. Single channel with safety-related communication via F module.

Figure 1-7: Version 4 b): Three-encoder system, position encoder via distributed PLC I/O, secure communication via F module



Three-encoder systems can be used if a significant level of slip is to be expected – or a higher availability is specified. Instead, the plausibility of the position is checked by making a cross comparison between the two direct measuring systems.

Safety note

In order to identify a bus driver that has frozen, i.e. communication between the measuring system and CPU is no longer operational, a fail-safe module is inserted in at least one channel in the station, via which the direct measuring system is read in. If the communication runs too slowly or completely fails, then the associated F module signals a communication error. This is then evaluated in the safety program and must be used to initiate a stop response.

Safety note

Both direct measuring systems must be installed opposing one another in order to achieve the specified diagnostics coverage.

1.2.5 Summary of the encoder versions

The following table summarizes the encoder combinations that are possible in principle - and their ability to be implemented. POS1, POS2 as well as POS_SI are referenced to the interconnection at the "F_SAFE_POS" block described in more detail in Chapter 3.2. See below for the legend.

A motor encoder (MSSI or MNSI) is always necessary to sense the safety-related position and velocity; this is sensed via the SI part of the drive.

The following encoder combinations should be used depending on the application scenario:

Table 1-1: Overview of possible encoder combinations

	POS1	POS2	POSSI
Version 1: Safety-related motor encoder with directly connected, positive locking mechanical system	---	---	MSSI(A)
Version 2: Two-encoder system: Connected via SINAMICS S120.	LD-SMx-NS	---	MSSI(R)/ MNSI(R)
Versions 3 a) and b): Two-encoder system: Connected via distributed I/O.	LD-DP-NS	---	MSSI(R)/ MNSI(R)
Version 4 a): Three-encoder system. Position encoder via SINAMCS 120 and distributed PLC I/O. Secure communication via an F module.	LD-SMx-NS[1]	LD-DP-KS[1]	MSSI(R)/ MNSI(R)
Version 4 b): Three-encoder system. Position encoder via distributed PLC I/O. Secure communication via an F module.	LD-DP-NS[1/2]	LD-DP-KS[1/2]	MSSI(R)/ MNSI(R)

[1] Overwriting the process image must be detected (diagnostics coverage), which is why the position actual values must oppose one another.

[2] The evaluation unit design must manifest diversity, e.g. POS1 via ET200SP with TM PosInput 2, POS2 directly via PROFINET/PROFIBUS.

Legend:

MSSI(A): Motor encoder, safety-related mounting, via SI F telegram 32-bit absolute safety-relevant position actual value

MSSI(R): Motor encoder, safety-related mounting, via SI F telegram 32-bit relative safety-related position actual value

MNSI(R): Motor encoder, no safety-related mounting, via SI F telegram 32-bit relative safety-related position actual value

LD-SMx-NS: Position actual value Epos from direct measuring system via SMC/SMI via standard telegram 32 bit, not safety-related

LD-DP-NS: Position actual value from direct measuring system via distributed I/O, no safety-related communication (e.g. PROFINET encoder, TM PosInput 2).

LD-DP-KS: Position actual value from direct measuring system via distributed I/O, safety-related communication using F module on the backplane bus.

Safety note
When using the relative position actual value of the SINAMICS S120 – sent in a safety-related way (MSSI(R)) – in the control, then the following FAQ should be carefully taken into consideration:
https://support.industry.siemens.com/cs/ww/en/view/109746390

The reference to the absolute position at the control level is established via the F_SAFE_POS block described in Chapter 3.2.
In version 1, it is not permissible to use the relative position actual value transferred in a safety-related way (MSSI(R)). In this case, the absolute position actual value of the SINAMICS S120, sent in a safety-related way (MSSI(A)), must be used in the control.

To be able to use the safety functions integrated in the drive, for the subsequently described software architecture, it is assumed that a SIN/COS encoder is always used as a motor encoder, and this is a read in using a fail-safe telegram from the safety program. A second encoder is used to check the plausibility, with the design versions described above.

1.2.6 Safety-related parameters of the encoder versions

Chapter 1.2.5 describes the various encoder versions. The particular version used depends on the application; for the safety-related use of the subsequently described software solution, it is always mandatory to use one of the versions described

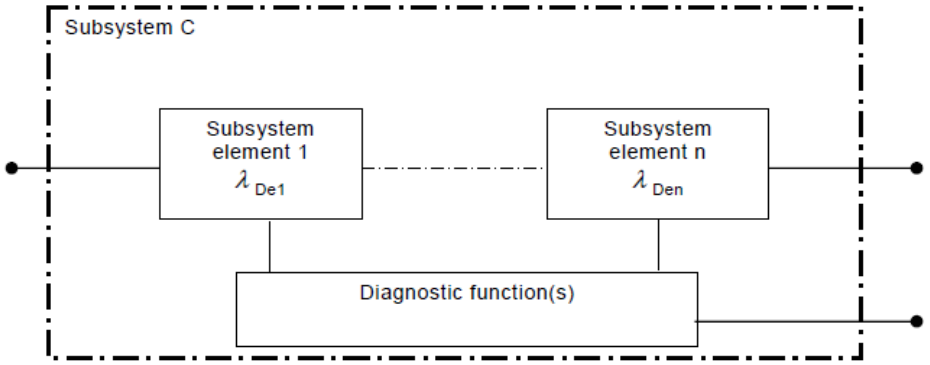
As a result of the different encoder versions - and therefore the resulting wide range of hardware versions that can be used - users must determine the safety integrity level achieved of the safety function. To comply with EN 528:2008, as a minimum this must correspond to SIL2/PL_r d over the entire safety function (sense -> evaluate -> respond).

To be able to verify this, in this section, parameters are described which are directly influenced by the software solution for calculating the safety integrity level based on EN 62061:2015. Only the "Sense" block is discussed here. The "Evaluate" block corresponds to a SIMATIC F-CPU with STEP7 Safety Advanced, certified to SIL3/PL e, the "Respond" block corresponds to a SINAMICS S120, certified to SIL2/PL d. The precise parameters of the "Evaluate" and "Respond" blocks should be taken from the corresponding data sheets.

While versions 2, 3a), 3b), 4a) and 4b) satisfy the requirements according to subsystem D, version 1 satisfies the requirements according to subsystem C, as shown in the following:

Subsystem C according to DINEN EN 62061:2015 (Chapter 6.7.8.2.4):

Figure 1-8

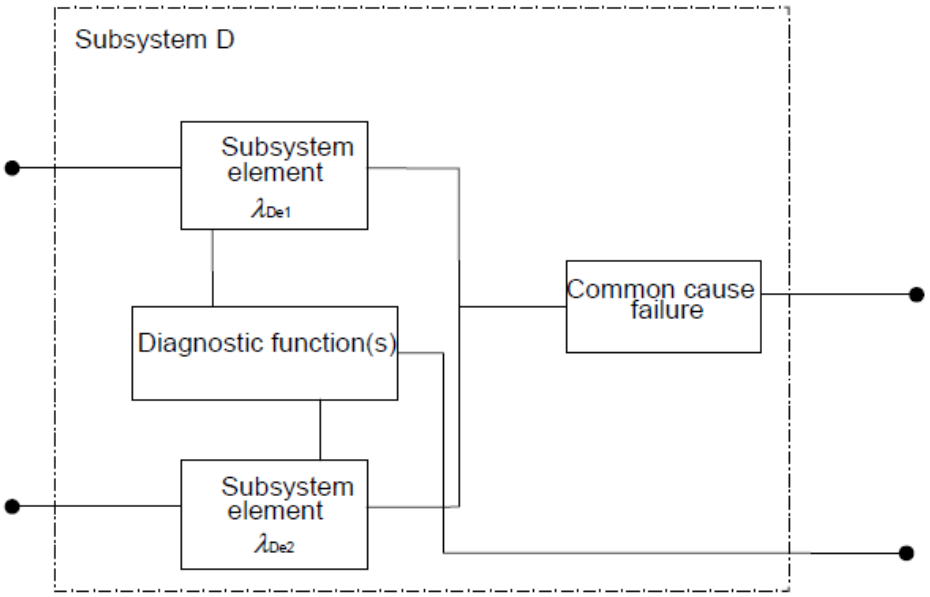


$$\lambda_{DssC} = \lambda_{De1} + (1 - DC_1) + \ldots + \lambda_{Den}(1 - DC_n)$$

$$PFH_{DssC} = \lambda_{DssC} \times 1\,h$$

Subsystem D according to EN 62061:2015 (Chapter 6.7.8.2.5):

Figure 1-9



$$\lambda_{DssD} = (1-\beta)^2 \left\{ \left[\lambda_{De1} \times \lambda_{De2} \times (DC_1 + DC_2) \right] \times T_2/2 + \left[\lambda_{De1} \times \lambda_{De2} \times (2 - DC_1 - DC_2) \right] \times T_1/2 \right\} + \beta \times (\lambda_{De1} \times \lambda_{De2})/2$$

$$PFH_{DssD} = \lambda_{DssD} \times 1h$$

Table 1-2: Parameters according to DIN EN 62061:2015

	Subsystem (SFF/HFT)	SIL CL limit	λ_{Ds1}	λ_{Ds2}	DC_1	DC_2	β	T1	T2
Version 1: Safety-related motor encoder with directly connected, positive locking mechanical system	C ((≥0.99[1]/ 0)	Internal safety function of the SINAMICS S120, certified according to SIL2/PLd							
Version 2: Two-encoder system: Connected via SINAMICS S120.	D (≥0.99[1]/ 1[2])	3[3]	Dependent on the hardware	Dependent on the hardware	99% of the diagnostics implemented in the RBG block library	Corresponding to DC1	0.02 acc. to Table 1-3: Evaluating common cause faults according to EN 62061:2015 Annex F.1.1	Dependent on the hardware	Corresponding to the call interval of the safety program
Version 3: Two-encoder system: Connected via distributed I/O.									
Version 4 a): Three-encoder system. Position encoder via SINAMCS 120 and distributed PLC I/O. Secure communication via an F module.									
Version 4 b): Three-encoder system. Position encoder via distributed PLC I/O. Secure communication via an F module.									

Comments regarding Table 1-2: Parameters according to DIN EN 62061:2015:

1) As the diagnostics implemented in the RBG block library identify all hazardous faults, it follows: $\lambda_{DU} \rightarrow 0$. When calculating SFF ($SFF = \frac{\sum \lambda_S + \sum \lambda_{DD}}{\sum \lambda_S + \sum \lambda_{DD} + \sum \lambda_{DU}}$) immediately SFF ≥ 0.99 is obtained.
2) The failure of a subsystem element does not result in the loss of the SRCF, as these faults are detected through a comparison value, as well as through a plausibility check (based on the redundant configuration). This means that HFT = 1 is directly obtained.
3) According to EN 62061:2015 Table 5, it follows that for HFT = 1 and SFF ≥ 0,99, a SIL CL of 3 is obtained.

Evaluating common cause faults according to EN 62061 Annex F.1 is shown in the following table. In some instances, measures against common cause faults are provided through the solution implemented in the RBG block library - in some instances, users must apply these measures themselves. Measures that users must always apply themselves are appropriately marked in the subsequent table. If additional measures are to be taken, which are shown in gray in the table, then these can improve the CCF factor or β value, on the other hand, measures that are not taken reduce the CCF factor or β value.

Table 1-3: Evaluating common cause faults according to EN 62061:2015 Annex F.1

Feature	Reference	Points	Reason
Disconnection/isolation			
Are SRECS signal cables for the individual channels separately routed away from other channels at all locations - or adequately protected?	1a	5	
Is the detection of signal transfer errors adequate when using information coding/decoding?	1b	**10**	Provided by the solution implemented in the RBG block library
Are SRECS signal cables and electric power cables routed separately - or adequately protected?	2	5	
Are subsystem elements accommodated in physically separate housings/enclosures if they can contribute to a CCF?	3	**5**	Requirements placed on users when installing sensors
Diversity/redundancy			
Are various electrical/electronic technologies used in the subsystems, for example, in some instances electronics or programmable electronics - and in some instances electromechanical relays?	4	8	
Are elements used in the subsystems that employee different physical principles (for example, sensing elements at a protective door that use mechanical and magnetic sensing techniques).	5	10	
Are elements used in the subsystems with different time responses with reference to functional operation and/or failure types?	6	**10**	Provided by the solution implemented in the RBG block library
Do the subsystem elements have a diagnostics test interval of ≤ 1 min?	7	**10**	Provided by the solution implemented in the RBG block library **Note:** The safety program must be called/run in less than 1 min intervals!
Complexity/design/application			
Are cross connections between channels of the subsystem prevented - with the exception of cross connections that are used for diagnostics?	8	**2**	Provided by the solution implemented in the RBG block library
Assessment/analysis			
Have the results of the failure types and effect analysis been evaluated in order to identify sources of common cause failures - and have these types of sources that have been previously determined been eliminated through an appropriate design?	9	**9**	Provided by the solution implemented in the RBG block library
Are field failures analyzed and incorporated in the design process?	10	9	

1.2 Encoder combinations and configuration versions that are supported

Feature	Reference	Points	Reason
Competence/training			
Do the engineers developing the subsystems know the reasons for and the effects of failures originating from a common cause?	11	**4**	Requirements placed on users
Monitoring ambient conditions			
Is it probable that the subsystem elements always operate - without externally monitoring the ambient conditions - within the temperature, humidity, corrosion, dust and vibration range in which they have been tested?	12	**9**	Users must appropriately select the sensors for the application
Is the subsystem immune with respect to the negative influence of electromagnetic fields - up to and including the limits defined in Annex E?	13	**9**	Users must appropriately select the sensors for the application
Result		**68**	**acc. to EN 62061:2015 Table F.2: β = 0.02**

Table 1-4: EN 62061 :2015 Annex F.2

Total number of points	Factor of the failures resulting from a common cause (β)
< 35	10 % (0.1)
35 to 65	5 % (0.05)
65 to 85	2 % (0.02)
85 to 100	1 % (0.01)

2 System and software requirements

2.1 General

The fail-safe function blocks for storage and retrieval machines described in the following chapters can be used in conjunction with the fail-safe Siemens S7-1500(T)F automation system – the following are recommended e.g.:

- CPU 1516F-3 PN/DP
- CPU 1517F-3 PN/DP
- CPU 1518F-4 PN/DP

Initially, the safety aspects when creating fail-safe function blocks are discussed before the properties and attributes are explained in detail.

Fail-safe function blocks for storage and retrieval machines have been developed based on individual subfunctions to ensure that these blocks can be used in a modular way.

2.2 Safety requirements

S7-1500F automation systems can satisfy the following safety requirements:

Safety Integrity Levels SIL1 to SIL3 according to IEC 61508 2nd Edition

2.3 Software

The following *Siemens SIMATIC* software must be installed on the PC/PG for using fail-safe function blocks for storage and retrieval machines:

- SIMATIC STEP 7 Professional V14 SP1 or higher
- SIMATIC STEP 7 Safety Advanced V14 SP1 or higher

as well as for the drive parameterization

- SINAMICS MICROMASTER STARTER V4.4 or higher

The current version - as well as all predecessor versions of the SINAMICS MICROMASTER STARTER - can be downloaded at the following link:

https://support.industry.siemens.com/cs/ww/en/view/26233208

2.4 Safety aspects when creating blocks

The blocks for the safety-related control of storage and retrieval machines were created using certified, fail-safe function blocks in F-FBD and F-LAD. The compiler of the development tool generates coded, fail-safe blocks. These can then be transferred into libraries and called in any arbitrary F-FBs and F-FCs.

Regarding the internal implementation and the software development process employed, the fail-safe function blocks for storage and retrieval machines comply with the requirements according to PLd/SIL2. However, it must be additionally proven that the function blocks from the storage and retrieval machine block library used in the user software are in compliance with the relevant standards regarding their behavior/response and their principle of operation. Generally, this proof can be provided in the form of a function test.

Also due to restrictions regarding the hardware components that can be used, specifically converters/inverters, the safety integrity level that can be achieved using the fail-safe function blocks for storage and retrieval machines is restricted to PLd/SIL2.

The safety-related parameters, required to verify the hardware components, can be taken from Chapter 1.2.6.

2.5 Standards complied with

The RBG block package was developed according to the following standards:

- EN 528:2008 (see Chapter 2.5.1)
- EN ISO 13849-1:2015
- EN ISO 13849-2:2012
- EN 62061:2015

2.5.1 Demarcation of EN 528:2008 with respect to the RBG block library

Table 2-1

Requirements according to EN528:2008 Table C.2			Covered by the RBG **block library**
No.	Safety function	Section	
1	Function to monitor access through doors	5.3.3 5.3.4 5.10.3.3 5.10.3.6	No User interconnection required (e.g. Safety Advanced Library)
2	Stop function	5.3.7	No User interconnection required (e.g. Safety Advanced Library)
3	Emergency Stop function	5.3.8 5.3.8.1 5.3.8.3	No User interconnection required (e.g. Safety Advanced Library)
4	Function to stop hoisting motion at the end stops and when the power fails	5.4.1.1 5.4.2 a), b),	Yes

Requirements according to EN528:2008 Table C.2			Covered by the RBG **block library**
No.	Safety function	Section	
		c)	
5	Function to stop travel motion at the end of the travel section (e.g. the end of the aisle), when the power fails, when collisions occur, if more than one machine is traveling on the same rail system	5.5.1.1 5.5.1.2 5.5.3	Yes
6	Additional brake and velocity reduction function if curved sections must be traveled along with reduced velocity	5.5.1.2 a) 5.5.2	Yes
7	Function of the additional brake and velocity reduction function if the end stops have not been designed so that the machine can approach them with at least 70% of its rated velocity	5.5.1.2 b) 5.5.2	Yes
8	Function to prevent the load and the load handling device from colliding with the rack itself	5.4.6.6 5.6.2 5.6.3 5.6.5 5.6.7 5.6.8.2 5.10.7.1	Yes
9	Load handling device - interlocks	5.6.5 a), b)	Yes
10	Load handling device - interlocks	5.6.5 c)	Yes
11	Load handling device - storage location assigned	5.6.5 d)	No User interconnection required (e.g. Safety Advanced Library)
12	Load handling device - load position monitoring	5.6.7	No User interconnection required (e.g. Safety Advanced Library)
13	Function of control devices for dangerous motion (manual control)	5.7.6	No User interconnection required (e.g. Safety Advanced Library)
14	Interlocking function with transfer equipment	5.8.2 5.8.3 5.8.4	No User interconnection required (e.g. Safety Advanced Library)
15	Function, which only permits slow velocity, if a person is located on the emergency control station	5.3.7 5.9.4	No User interconnection required (e.g. SLS in the converter)
16	Function to stop the unit if access is required through emergency exits and covers	5.10.3.2 c) 5.10.3.2 e) 5.10.3.4 5.10.6.4	No User interconnection required (e.g. Safety Advanced Library)
17	Area around equipment - to secure against inadvertent load movement Storage location (rack compart-	5.10.7.1 a)	No User interconnection required (e.g. Safety Advanced Library)

Requirements according to EN528:2008 Table C.2			Covered by the RBG **block library**
No.	Safety function	Section	
	ment) sensor		
18	Area around equipment - to secure against inadvertent load movement Backstop to prevent goods sliding through	5.10.7.1 b)	No User interconnection required (e.g. Safety Advanced Library)
19	Area around equipment - to secure against inadvertent load movement Backstop to prevent goods sliding through	5.10.7.1 c)	No User interconnection required (e.g. Safety Advanced Library)

Other sections of EN528:2008 covered by the RBG block library:

- 5.4.3.1 – overload protection
- 5.4.3.2 - slack cable protection

3 Fail-safe function blocks for storage and retrieval machines

3.1 Overview

3.1.1 Safety note

As shown in Chapter 2.5 the fail-safe function blocks for storage and retrieval machines comply with the requirements of EN528:2008.

The safety integrity of the particular safety function is only given with the correct interconnection. This is the reason that the correct interconnection of every fail-safe function block in this library - and the complete functionality of the safety function with the particular application-specific hardware and software - must be validated using the appropriate positive and negative tests.

Initially, these tests should be made in an area of the system that has sufficient clearance to fixed endstops. When tests are being conducted it must be ensured that the system can be safely stopped if an emergency arises.

The tests should be documented, for example, using traces so that limit value violations, shutdown conditions and stopping distances can be clearly identified, and in turn a statement can be made about the correct function of each individual safety function.

Safety note

Appropriate measures must be taken to prevent unauthorized change and manipulation. A CPU password must be parameterized in the device settings for at least the writing online access (fail-safe and standard). In the safety administration, the safety program must be secured against offline accesses using a password.

Safety note

Using non-fail-safe values from the standard part of the CPU in the fail-safe program requires seamless consistency of these values. It must be ensured that these values are neither directly nor indirectly influenced by the non-fail-safe programming or its changes. The integrity of these values must be checked after each program change – for example, whether writing cross-access operations occur.

We strongly recommend that the complete system, including fail-safe and non-fail-safe part are protected against unauthorized access in accordance with state-of-the-art IT security directives.

3.1.2 Fail-safe blocks

Library "ASRM_Failsafe_TS1500_V21.zal14" includes the following blocks:

F_SAFE_POS	Function block to generate a safety-related position and velocity actual value
F_SLP_MONITOR	Function block for safe position monitoring
F_ENDZONE	Function block to monitor the velocity at the end of the traversing range
F_BRAKE_TEST	Function block to execute a safe brake test in conjunction with drive function *SBT*
F_LOAD_MONITOR	Function block for overload and slack cable detection - with the possibility of testing the measuring equipment
F_SBR_MONITOR	Function block for brake ramp monitoring
F_MIN_MAX	Function for minimum/maximum value selection
F_INTERPOLATION	function to calculate the envelope curve for function block F_ENDZONE
LFAddDInt	Envelope block to intercept an overflow in the double integer counting range for an addition
LFSubDInt	Envelope block to intercept an overflow in the double integer counting range for a subtraction
LFMulDInt	Envelope block to intercept an overflow in the double integer counting range for a multiplication
LFDivDInt	Envelope block to intercept an overflow in the double integer counting range for a division

The following fail-safe blocks of the STEP7 Safety Advanced library are additionally required:

F_TP (V1.4)	Generates a pulse with a specific duration
F_W_BO (V2.0)	Converts a value into the WORD format in 16 data, bool data type
F_BO_W (V2.0)	Converts 16 data, BOOL data type into WORD data type

These blocks are contained under Statements -> Simple statements.

NOTE The library blocks listed under SIMATIC STEP 7 Safety Advanced must be updated to the current versions before being integrated into the RBG library. Further, the elements of the system library used in the safety administration must be updated to Version 2.0 or 2.1. Otherwise, error messages can be generated when compiling the safety program.

3.1.3 Block connections

With fail-safe blocks, a few special characteristics must be taken into account regarding the block connections.

NOTE Although the EN and ENO connections appear in the FBD/LAD editor, they are neither evaluated nor supplied from the program code of the F-block - and it is not permissible that you interconnect or parameterize them.

3.1.4 Block numbers and signatures

Table 3-1: Block Signatures

Block number	***Block name***	**SIMATIC STEP 7 Safety Advanced**
		Block Signature
FB200	F_SAFE_POS	0x2BE066C0
FB201	F_SLP_MONITOR	0xB6974076
FB202	F_ENDZONE	0x18742561
FB203	F_BRAKE_TEST	0xAEF3B5CB
FB204	F_LOAD_MONITOR	0xFCBB27C6
FB207	F_SBR_MONITOR	0x2B05EE7D
FC206	F_MIN_MAX	0xA225BBC3
FC200	F_INTERPOLATION	0x2C338B83
FC211	LFAddDInt	0x24EE93DA
FC212	LFSubDInt	0xF7AB9793
FC213	LFMulDInt	0x720E2721
FC214	LFDivDInt	0xC02D0512

In the following chapter, FB/FC numbers are assigned for the blocks to be implemented. When required, these can be adapted to the requirements of the specific machine - i.e. the blocks can be freely renumbered, however it is not permissible that they are renamed.

NOTE In the following chapter, FB/FC numbers are assigned for the blocks provided in this library. When required, these can be adapted to the requirements of the specific machine - i.e. the blocks can be freely renumbered. However, it is **not** permissible that the blocks are renamed, as otherwise it is possible that the safety program signature changes.

3.1.5 Integration in cyclic interrupts - F-OB

Safety note

The specific configuration of the cycle time of the safety program is orientated to the requirements arising from the risk assessment for the machine where the blocks are to be used. The user is responsible for correctly carrying out the risk assessment and appropriately configuring the times.

3.1.6 Using instance data blocks/multi-instances

NOTE The RBG blocks can be called as multi-instance without any restrictions.

3.1.7 Response times

The response times required can be taken from the risk assessment. This involves a block package that can be universally used. As a consequence, specific response times for the individual safety functions cannot be specified.

Safety note

Depending on the required response time, parameter T_SAMPLE (and therefore the interval in which the safety program is called) as well as POS_SI_T_SAMPLE should be parameterized in the RBG blocks so that they are always less than the maximum permitted response time. It should be taken into account that the hardware components used influence the response time. Using the s7safety_rttplus table it is possible to calculate the response time that is achieved from the sensor to the actuator.

3.1.8 Runtimes

The runtime (execution) values of the fail-safe RBG blocks on the supported F-CPUs, required to calculate the response times, can be taken from Table in Attachment I).

Safety note

It is the user's responsibility to interconnect and parameterize the RBG blocks in compliance with the applicable standards. This especially applies to the brake test rates and the overload/slack cable detection - as well as all load and velocity limits.

Safety note

All position limits must be selected so that when violated, the axis can come to a standstill before the end of the traversing range. The value to be parameterized is also dependent on the maximum velocity expected for a specific application - as well as the maximum possible and permitted deceleration.

3.2 F_SAFE_POS

3.2.1 Introduction

The fail-safe function block F_SAFE_POS generates a safety-related position actual value by comparing the discrepancy from two encoders. A velocity is calculated from the motor encoder value, and verified by comparing the position discrepancy using an absolute encoder. Within a time that can be parameterized, the positions between the motor encoder and the second encoder, used to check the plausibility, must not deviate from one another by more than the slip tolerance so that the velocity value can be considered to be safety-related.

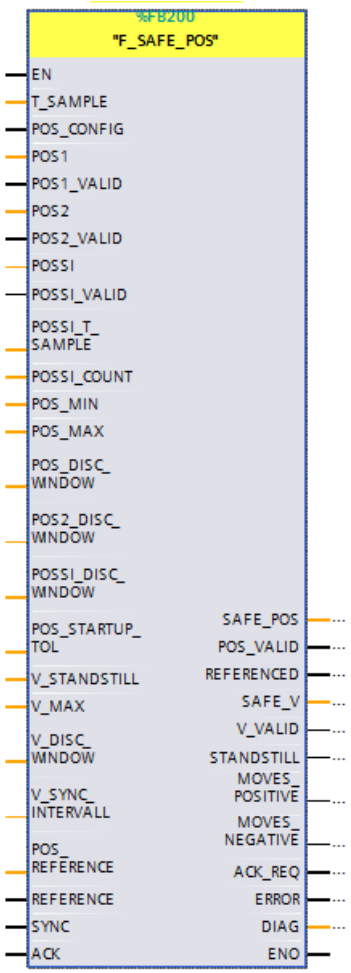
The safety-related position and velocity form the basis for other blocks described in this document.

Redundant position sensing is always required if the position cannot be uniquely identified using the Safety Integrated (SI) motor encoder in the drive. This can be the case if the encoder cannot be mounted in a safety-related fashion - or the mechanical system has slip or is subject to elongation (e.g. travel gear with a wheel-rail system or hoisting gear with a cable winch). As a consequence, the position-referred safety functions cannot be used in the SI of the drive. A direct measuring system to monitor the position must then be used. This is realized in the F-CPU via this block. The motor measuring system can then only be used to check the plausibility of the direct position actual value.

For applications where a more significant level of slip is to be expected – or a higher degree of availability is specified – the block provides the option of deriving the safety-relevant position based on a discrepancy comparison from two direct measuring systems.

NOTE When using this block, then block **F_BO_W (FC 176)** must be available in the block folder. It is not permissible to renumber these! Blocks **LFAddDInt (FC 211)**, **LFSubDInt (FC 212)**, **LFMulDInt (FC 213)** and **LFDivDInt (FC 214)** are also required from this library. These may be renumbered, but not renamed.

Figure 3-1: F_SAFE_POS



3.2.2 Connections

All bool type variables listed in the following table are preassigned FALSE, all integer variables with 0 and all word variables with W#16#0.

Inputs

Table 3-2

Name	Data type	Description
T_SAMPLE	DInt	**Block sampling time [ms]** Interval in which the safety program is called
POS_CONFIG	Bool	**Configuration word for encoder interconnection** 1: two direct measuring systems + motor encoder 0: one direct measuring systems + motor encoder
POS1	DInt	**1st direct measuring system - measured value [mm]**
POS1_VALID	Bool	Value from the process image **1st direct measuring system - encoder signal status** 1: Encoder signal valid

Name	Data type	Description
POS2	DInt	**2nd direct measuring system - measured value [mm]**
POS2_VALID	Bool	**2nd direct measuring system - encoder signal status** 1: Encoder signal valid 0: Encoder fault
POSSI	DInt	**Motor encoder Safety Integrated - measured value [µm]**
POSSI_VALID	Bool	**Motor encoder Safety Integrated - encoder signal status** 1: Encoder signal valid 0: Encoder fault
POSSI_T_SAMPLE	DInt	**SINAMICS Safety Integrated - sampling time [ms]** Sampling time of SI configured in the drive
POSSI_COUNT	DInt	**SINAMICS Safety Integrated - cycle counter [ms]** Cyclic counter value of telegram 902
POS_MIN	DInt	**Max. permitted position [mm]**
POS_MAX	DInt	**Min permitted position [mm]**
POS_DISC_WINDOW	DInt	**Max. permissible encoder deviation in operation [mm]** In operation, the value of the second encoder, used to check the plausibility (POS_SI or POS2, dependent on POS_CONFIG), may not deviate by more than this value from encoder 1 (POS1)
POS2_DISC_WINDOW	DInt	**max. permissible encoder deviation between POS1 and POS2 in operation [mm]** For a 3-encoder system, the deviation between POS1 and POS2 must not exceed the value parameterized here.
POSSI_DISC_WINDOW	DInt	**max. permissible encoder deviation between POS1 and POSSI in operation [mm]** The deviation between POS1 and POSSI must not exceed the value parameterized here.
POS_STARTUP_TOL	DInt	**max. permissible deviation when powering up [mm]** After an encoder fault of POSSI, for a 2-encoder system, the actual POS1 value must not deviate from the previously saved value of POS1 for a fault-free system more than the value parameterized here.
V_STANDSTILL	DInt	**Tracking interval velocity monitoring [ms]**
V_MAX	DInt	**Max. permissible velocity for checking the plausibility [mm/min]**
V_DISC_WINDOW	DInt	**Tolerance window velocity monitoring [mm]**
V_SYNC_INTERVALL	DInt	**Tracking interval velocity monitoring [ms]**
POS_REFERENCE	DInt	**Reference position [mm]** With a positive edge at SET, POS1 and the redundant encoder (POS_SI or POS2, depending on POS_CONFIG) are synchronized at this position
REFERENCE	Bool	**Referencing** 0 -> 1: Determine the encoder offset regarding POS_REFERENCE
SYNC	Bool	**Synchronizing** 0 -> 1: The encoder used to check the plausibility (POS_SI or POS2, depending on POS_CONFIG) is synchronized with encoder 1 (POS1)

Name	Data type	Description
ACK	Bool	**Acknowledging** If a fault occurs in normal operation, then this must be reset using ACK before the system can be restarted. Acknowledgment is realized with a positive edge at ACK, and in normal operation has no effect.

Outputs

Table 3-3

Name	Data type	Description
SAFE_POS	DInt	**Safe position actual value [mm]** Safe position (for all additional blocks in this block package)
POS_VALID	Bool	**Status, position actual value** 1: SAFE_POS was generated in a safety-related fashion
REFERENCED	Bool	**Referencing status** 1: Both encoders are referenced, and the discrepancy between both encoders is within the tolerance window
SAFE_V	DInt	**Safe velocity actual value [mm/min]** Safe velocity (for all additional blocks in this block package)
V_VALID	Bool	**Status, velocity actual value** 1: SAFE_V was generated in a safety-related fashion
STANDSTILL	Bool	**Standstill detection** 1: Actual velocity less than V_STANDSTILL
MOVES_POSITIVE	Bool	**Motion in the positive direction**
MOVES_NEGATIVE	Bool	**Motion in the negative direction**
ACK_REQ	Bool	**Acknowledgment request** 1: Faults that are no longer active can be acknowledged 0: No acknowledgment requested
ERROR	Bool	**Error** 1: At least one error has been identified 0: No error active
DIAG	Word	**Diagnostics word** Information about the function status and errors of the block are output here.

Structure of DIAG

Table 3-4

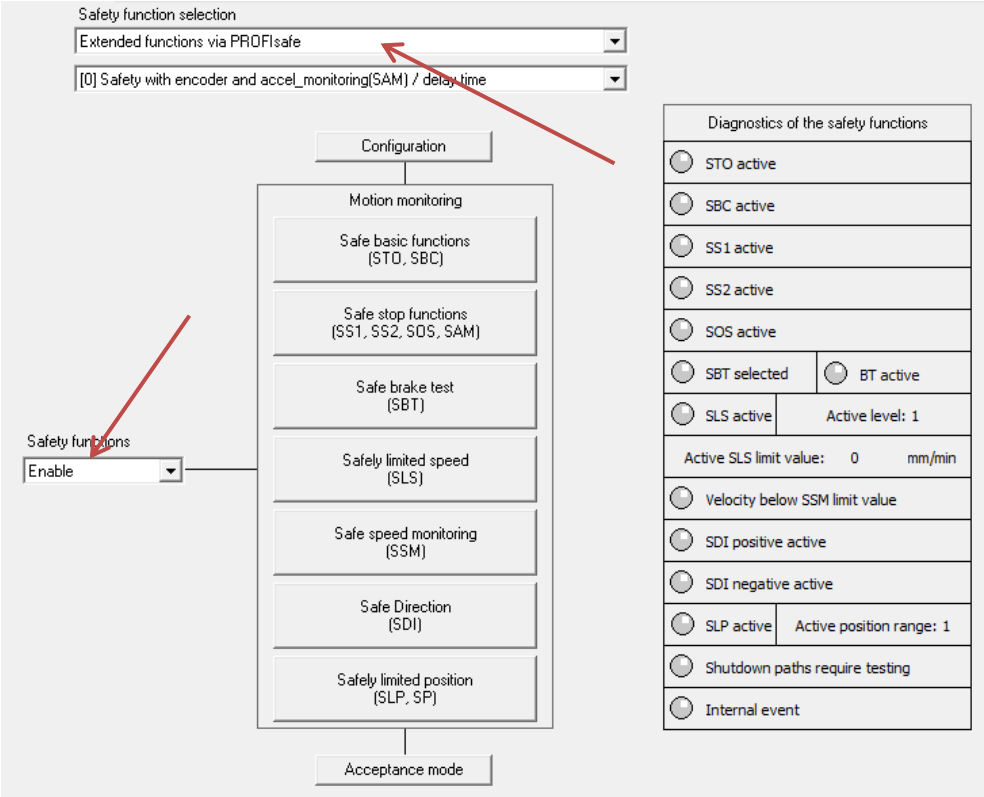
Bit	Description	Reset condition
0	Value range violation of input variables	1 <= POSSI_T_SAMPLE <= 1023
		V_SYNC_INTERVALL > 0
		0 < T_SAMPLE <= 2 * POSSI_T_SAMPLE
		V_DISC_WINDOW >= 0
		POS_DISC_WINDOW >= 0
		POS_STARTUP_TOL >= 0
1	Ratio between the input variables cannot be represented as integer number	V_SYNC_INTERVALL / T_SAMPLE is an integer number
2	Incorrect reference of the input variables to one another	V_MAX >= V_STANDSTILL
		MAX_POS >= MIN_POS

Bit	Description	Reset condition
3	Actual position > MAX_POS	Actual position <= MAX_POS and positive edge at ACK
4	Actual position < MIN_POS	Actual position>= MIN_POS and positive edge at ACK
5	Actual velocity > V_MAX	Actual velocity <= V_MAX and positive edge at ACK
6	Max. permissible velocity discrepancy exceeded	Velocity discrepancy <= V_DISC_WINDOW and positive edge at ACK
7	Initial referencing missing	Reference point approach and pos. edge at REFERENCE
8	POS1 – POSSI > POSSI_DISC_WINDOW (2-encoder system)	pos. Edge at SYNC
9	POS1_VALID/POSSI_VALID/POS2_VALID == 0	Dependent on DIAG bit 10 - 12
10	After an encoder fault: "Position actual value POS1“ – "Last valid position actual value POS1“ > POS_STARTUP_TOL (2-encoder system)	"Position actual value POS1“ – "Last valid position actual value POS1“ <= POS_STARTUP_TOL and pos. Edge at ACK or pos. Edge at SYNC
11	After an encoder fault: POS1 – POSSI > POSSI_DISC_WINDOW (2-encoder system)	POS1 – POSSI <= POS_DISC_WINDOW and pos. Edge at ACK or pos. Edge at SYNC
12	After an encoder fault: POS1 – POS2 > POS2_DISC_WINDOW (3-encoder system)	POS1 – POS2 <= POS_DISC_WINDOW and pos. Edge at ACK or pos. Edge at SYNC
13	Error for internal calculation	Pos. edge at ACK, if the fault is no longer active
14	Reserve	---
15	Reserve	---

3.2.3 Interrelationship between the assignment of the block inputs and the drive configuration

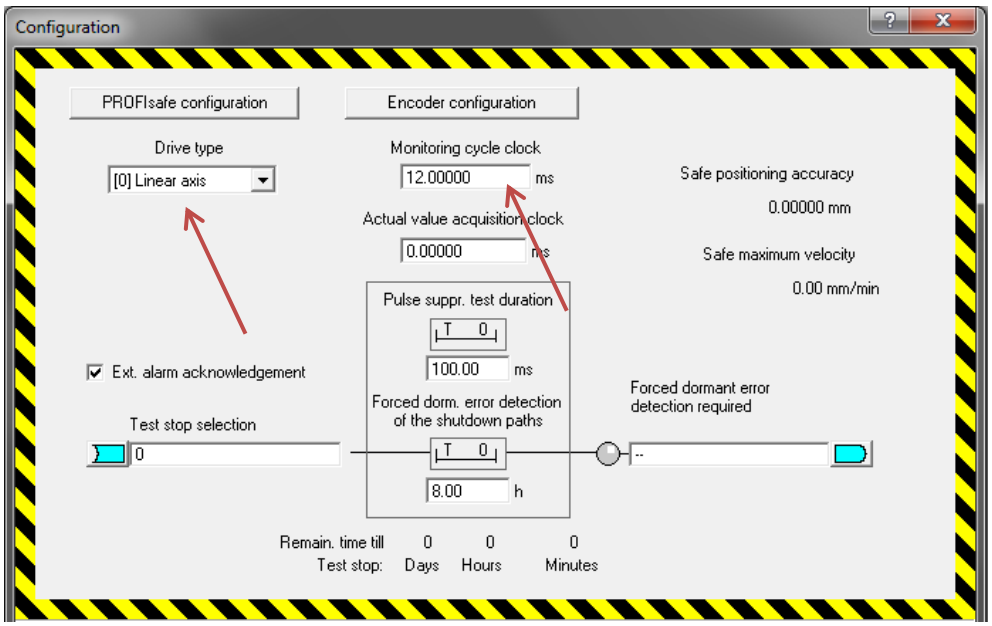
The safe absolute position actual value from the drive is transferred via PROFIsafe telegram 902 as a 32-bit value with unit µm. To do this, in the converter, the "Extended functions via PROFIsafe" should be set and the safety functions enabled.

Figure 3-2: Setting Safety Integrated in the converter



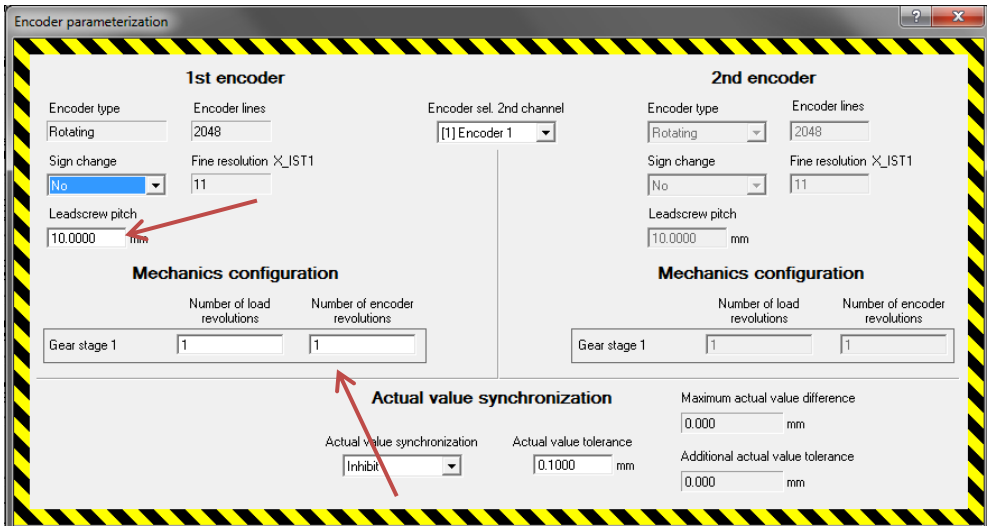
The drive type must then be set to linear axis; the monitoring clock cycle is subsequently important when parameterizing the block.

Figure 3-3: Configuring Safety Integrated



The encoder parameterization opens by selecting "Encoder configuration". Here, the leadscrew pitch as well as the gearbox stage should be set so that they match the physical mechanical arrangement.

Figure 3-4: Encoder parameterization

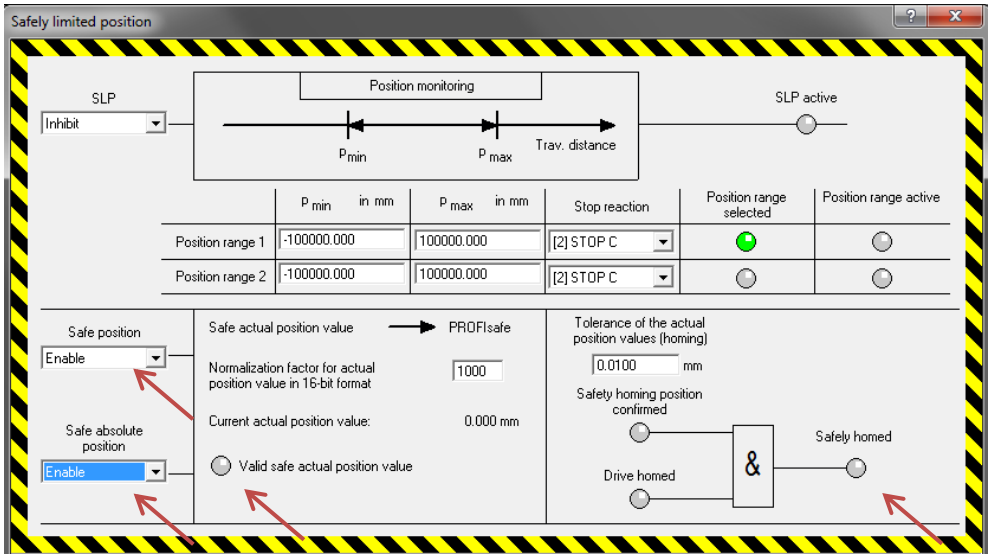


Depending on the encoder version, the relevant safety position must be enabled in the safety function.

1-encoder version

The safety-related position as well as the safety-related absolute position must be enabled for transfer here. The safety-related position value must be valid and the axis must of been safely homed (reference). The SLP function can be activated in the drive and used. Alternatively, this monitoring can be implemented using block F_SLP_MONITOR of this library.

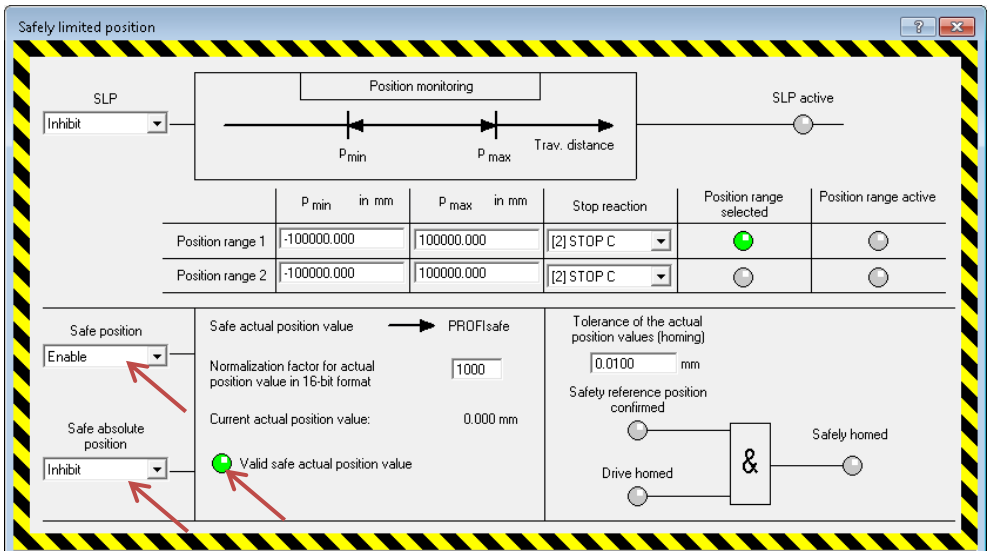
Fig. 3-5: Safety-related position, 1-encoder version



2 & 3-encoder version

For these versions, it is only permissible that "Safe position" transfer is enabled The "Safe absolute position" must remain inhibited. A safe relative position actual value sufficient, as block F_SAFE_POS establishes the absolute position actual value reference, as described in the function description.

Fig. 3-6: Safe position transfer 2 & 3-encoder version



Encoder configuration at the block, 1-encoder version

The safe absolute position actual value from the drive is transferred via PROFIsafe telegram as a 32-bit value in units [µm]. The encoder value must be interconnected to POS1 – scaled once by dividing by 1000 – and in addition, directly interconnected to POSSI, without scaling. Parameters T_SAMPLE, POSSI_T_SAMPLE, POSSI_COUNT, V_STANDSTILL, V_MAX and V_T_SAMPLE are relevant when calculating the safe velocity, and must be interconnected as described.

Note When using block F_SAFE_POS, the safety note from 1.2.1 must be carefully taken into consideration. Outputs SAFE_POS, POS_VALID and REFERENCED are not permissible, and it is only permissible that safe velocity SAVE_V is used.

Note The following parameters must be parameterized according to what is specified (Chapter 1.2.1), even if they are not relevant for a pure safe velocity calculation:

- POS_MIN
- POS_MAX
- POS_DISC_WINDOW
- POS_STARTUP_TOL
- POS2_DISC_WINDOW
- POSSI_DISC_WINDOW

Encoder configuration at the block, 2 & 3-encoder version

The safe relative position actual value from the drive is transferred via PROFIsafe telegram as a 32-bit value in units [µm]. The value of the first direct encoder is interconnected in [mm] to POS1 – and if a second direct encoder is being used, then its value is interconnected to POS2 in [mm], and POS_CONFIG is parameterized to 1.

The plausibility of the calculated velocity from POSSI is always checked using the encoder interconnected to POS1. The plausibility of the position from POS1 is, for POS_CONFIG = 0, checked using the encoder interconnected to POSSI; for POS_CONFIG = 1, the plausibility is checked using POS1 with respect to POS2.

Note POS1 and POS2 expect opposing values!

Note The encoder values at POS1 and POS2 must always be positive.

Safety note

The signals interconnected at POS1 and POS2 must come from two independent sources. If one signal source is jumpered to both inputs, then non-plausible values of this channel cannot be detected. The safety integrity of the block is then no longer guaranteed.

Safety note

Encoder value POSSI moves in the range -737280mm to +737280mm; for block F_SAFE_POS, only the value range from 0 mm - +737280mm is permissible. If POSSI assumes values higher than +737280mm – or values less than -737280mm, then an overflow occurs. As a result of the large value change and the resulting discrepancy, enable signals are reset and the system initiates the shutdown response interconnected by the user. As a consequence, POSSI should be monitored against low/high values.

3.2.4 Principle of operation

Parameterization

Safety note
To ensure error-free exchange of the actual position value for POS1/POS2 between the DB of the technology object and the safety program, the conversion from data type real into DInt must be realized in OB1 (or another OB, that has a lower priority than F-OB) – directly or in a lower-level block. Only then can the integrity of the safety program be guaranteed. The converted position actual value should be saved to a transfer DB, which is exclusively used to transfer data between a standard and safety program.

Position actual value from the technology object:

Laser1_SSI_ET200SP

		Name	Datentyp	Startwert
1		Base	TO_Axis	
2		Input		
3		Output		
4		InOut		
5		Static		
6		ActualPosition	LReal	0.0
7		ActualVelocity	LReal	0.0
8		ActualAcceleration	LReal	0.0
9		Sensor	TO_Struct_ExternalEncoder_Sensor	
10		Extrapolation	TO_Struct_Extrapolation	
11		Mechanics	TO_Struct_Mechanics	
12		LoadGear	TO_Struct_LoadGear	
13		Properties	TO_Struct_Properties	
14		Modulo	TO_Struct_Modulo	
15		Homing	TO_Struct_ExternalEncoder_Homing	
16		Units	TO_Struct_ExternalEncoder_Units	
17		StatusSensor	TO_Struct_StatusSensor	
18		StatusExtrapolation	TO_Struct_StatusExtrapolation	
19		StatusWord	DWord	16#0
20		ErrorWord	DWord	16#0
21		ErrorDetail	TO_Struct_ErrorDetail	
22		WarningWord	DWord	16#0
23		InternalToTrace	Array[1..4] of TO_Struct_Internal	

Typical code to be copied over

```
27
28  "Global_DB".Lageistwert_Fahrachse_Laser1 := LREAL_TO_DINT("Laser1_SSI_ET200SP".ActualPosition);
29
```

1. The position-defining encoder is interconnected at POS1 in [mm].
2. At input POS1_VALID, users have the opportunity of interconnecting possibly available additional validity queries relating to the position value (e.g error bit from the module). The input must be permanently set to TRUE if this type of information is not available.
3. The sampling time of the block, e.g. the configured call interval of the F-OB, which calls the safety program, is parameterized at input T_SAMPLE.

4. The sampling rate of SI in the drive is parameterized at POSSI_T_SAMPLE – and POSSI_COUNT should be interconnected with the counter value from telegram 902.
5. T_SAMPLE is relevant for calculations performed in the block.
6. The safety relative position actual value from the motor encoder from telegram 902 in [µm] is interconnected at input POSSI.
7. For applications with a high slip and/or to achieve better availability, users have the option of connecting a second direct encoder at POS2 in [mm].
8. Using POS_CONFIG, users also have the option of switching between the mode for one and two direct encoders. If the input is set, then the block is in the mode for two direct encoders; otherwise, it is in the mode for one direct encoder.
9. The significance of inputs POS2_VALID or POSSI_VALID is equivalent to the corresponding inputs for POS1.

Safety note

The block must be parameterized with fixed values; it is not permissible to parameterize the block using variables during the CPU runtime.

When parameterizing, it must be ensured that the following relationships can be represented as integer numbers:

V_SYNC_INTERVALL / T_SAMPLE

Further, the following relationships between the input variables must apply:

V_MAX >= V_STANDSTILL

MAX_POS >= MIN_POS

Sampling rates

To calculate the velocity, as time basis, the block clock cycle of the F_SAFE_POS block on the CPU (T_SAMPLE, generally the call interval of the safety program) is not used – but instead the SI clock cycle in the drive (POSSI_T_SAMPLE) is used. In order to avoid inadmissibly high subsampling, the relationship T_SAMPLE <= 2x POSSI_T_SAMPLE must be guaranteed.

The permissible value ranges of the individual inputs can be taken from the table describing the inputs.

If not all of the specified preconditions are fulfilled, then the block detects this – and signals a parameterizing error with the appropriately set DIAG bits.

Note

The block only checks the parameterization at the 1st call. This increases the performance for ongoing block operation.

As a consequence, reparameterization is not possible in operation. The safety program must be regenerated and loaded each time that the operating parameters of the block are changed.

Starting behavior

Note

Differences between position actual values POS1 and POSSI or POS1 and POS2 always refer to the position actual values with offset, referred to the reference point.

Aligning POSSI to POS1 or POS2 to POS1 always takes place referred to the position actual value with offset (referenced to the reference point) of POS1.

10. After a CPU restart, outputs V_VALID, POS_VALID and REFERENCED initially output a 0 signal
11. In order to be able to travel to the reference point, POS_VALID must be set to 1, by calibrating both encoders via the positive edge at SYNC. The position value is now valid – and can be used for making relative position statements; however, it is still not permissible to make a safe absolute evaluation of the position as long as REFERENCED outputs a 0 signal.

WARNING

As long as REFERENCED outputs a 0 signal, the position can only be used to establish a relative reference; an absolute evaluation is only permissible for REFERENCED = 1.

In a specific application, when REFERENCED = 0, it is only permissible that the axis traverses with a safely reduced velocity.

12. If the axis is located at the reference point defined using input POS_REFERENCE, then the block is referenced using the positive edge at REFERENCE – and REFERENCED changes to 1. The position value output at SAFE_POS can now also be used to make absolute statements.

Safe position actual value – 2 encoder system (POS_CONFIG == 0)

13. To generate the safe position, the position actual values must be valid, which is signaled to the block using a 1 signal at inputs POS1_VALID and POSSI_VALID.

Note

An encoder fault should always result in the withdrawal of input signal POS1_VALID or POSSI_VALID/POS2_VALID. As a consequence, the safe actual value is immediately declared as invalid, and the position tolerance monitoring hidden. As a consequence, an encoder fault does not mean an immediate loss of the safe reference, and the safe position can be reproduced after the encoder signal returns.

14. The safe position actual value is output via SAFE_POS. A 1 signal at output POS_VALID signals that the safe position actual value is valid. A 1 signal at output RERFERENCED signals that it is permissible to use the safe position actual value as safe absolute position actual value. For a 0 signal at output

REFERENCED, the safe position actual value may only be used as safe relative position actual value.

15. If, using a 0 signal at input POS1_VALID or POSSI_VALID, an invalid position actual value is signaled, or a step is detected in the position actual value from POS1 (position change > 2x maximum velocity V_MAX), then outputs POS_VALID, REFERENCED and V_VALID are reset to 0, output ERROR is set to 1 – and bit 9 in output word DIAG is set to 1. In addition, the last valid position actual value from POS1 is saved internally.
16. If, using a 1 signal at inputs POS1_VALID and POSSI_VALID, the position actual values are signaled as being again valid, then the block (internally) checks the ongoing validity of the safe reference. For this purpose, a safe reference must have been carried out beforehand.
17. If only the position-defining encoder POS1 was invalid, then after its return, a check is made as to whether the difference between the position actual values from POS1 and POSSI (with the associated position offset (see point 12)) lies within the parameterized tolerance window POS_DISC_WINDOW. If this is not the case, then bit 10 in output word DIAG is set to 1.
18. If the safety-related relative position actual value POSSI was invalid, then a check is made as to whether the difference between the actual position value from POS1 and the last saved valid position actual value from POS1 lies within the parameterized tolerance window POS_STARTUP_TOL. If this is not the case, then bit 11 in output word DIAG is set to 1.
19. If the safe reference can still be declared to be valid, output ACK_REQ is set to 1 and the fault status can be acknowledged with a positive edge at input ACK. Outputs POS_VALID, REFERENCED and V_VALID are again set to 1, outputs ERROR and ACK_REQ are reset to 0, and bits 9, 10 and 11 in output word DIAG are reset to 0.
20. If the safe reference cannot be declared to still be valid, then a new safe reference must be performed in the form of a reference point approach (homing). To do this, the encoder can be calibrated using a positive edge at input SYNC as described under Point 11 (output ERROR is reset to 0, and in output word DIAG bits 9, 10 and 11 are reset to 0) and the safe reference performed after the reference point approach as described under Point 12.
21. If the difference between position actual values of POS1 and POSSI (with the associated position offset (see Point 16)) – with position actual values signaled as valid – exceeds the tolerance window POSSI_DISC_WINDOW, then outputs POS_VALID, REFERENCED and V_VALID are reset to 0, output ERROR is set to 1 – and bit 8 in output word DIAG is set to 1.
22. In this fault state, no statements can be made regarding the validity of the safe reference, and a new safe reference must be performed using a reference point approach. To do this, the encoder can be calibrated using a positive edge at input SYNC as described under Point 11 (output ERROR is reset to 0) and the safe reference performed after the reference point approach as described under Point 16. If, when the fault state occurs, the system is already at the reference point, then the safe reference can be performed directly using a positive edge at input REFERENCE. In output word DIAG, bit 8 is again reset to 0.
23. If SAFE_POS exceeds the value of MAX_POS, output ERROR is set to 1 – and bit 3 in output word DIAG is set to 1. Output POS_VALID is reset to 0.
24. If SAFE_POS again falls below the value of MAX_POS, output ACK_REQ is set to 1 and the fault status can be acknowledged with a positive edge at input ACK. Outputs ERROR and ACK_REQ are reset to 0, bit 3 in output word DIAG is set to 0. Output POS_VALID is again set to 1.

25. If SAFE_POS falls below the value of MIN_POS, output ERROR is set to 1 – and bit 4 in output word DIAG is set to 1. Output POS_VALID is reset to 0.
26. If SAFE_POS again exceeds the value of MIN_POS, output ACK_REQ is set to 1 and the fault status can be acknowledged with a positive edge at input ACK. Outputs ERROR and ACK_REQ are reset to 0, bit 4 in output word DIAG is set to 0. Output POS_VALID is again set to 1.

Safe position actual value – 3 encoder system (POS_CONFIG == 1)

27. To generate the safe position, the position actual values must be valid, which is signaled to the block using a 1 signal at inputs POS1_VALID, POSSI_VALID and POS2_VALID.

Note An encoder fault should always result in the withdrawal of input signal POS1_VALID or POSSI_VALID/POS2_VALID. As a consequence, the safe actual value is immediately declared as invalid, and the position tolerance monitoring hidden. As a consequence, an encoder fault does not mean an immediate loss of the safe reference, and the safe position can be reproduced after the encoder signal returns.

28. The safe position actual value is output via SAFE_POS. A 1 signal at output POS_VALID signals that the safe position actual value is valid. A 1 signal at output RERFERENCED signals that it is permissible to use the safe position actual value as safe absolute position actual value. For a 0 signal at output REFERENCED, the safe position actual value may only be used as safe relative position actual value.
29. If, using a 0 signal at inputs POS1_VALID, POSSI_VALID or POS2_VALID, an invalid position actual value signal is signaled, a step is identified in the position actual value of POS1 (position change> 2x maximum velocity V_MAX), if the difference between POS1 and POSSI (with the associated position offset (see Point 29)) exceeds the tolerance window POSSI_DISC_WINDOW – or if the difference between POS1 and POS2 (with the associated position offset (see Point 29)) exceeds the tolerance window POS2_DISC_WINDOW, then outputs POS_VALID, REFERENCED and V_VALID are reset to 0, output ERROR is set to 1, and bits 9/10 and 11 in output word DIAG are set to 1.
30. If, using a 1 signal at inputs POS1_VALID, POSSI_VALID and POS2_VALID, the position actual values are signaled as being again valid, then the block (internally) checks the ongoing validity of the safe reference. For this purpose, a safe reference must have been carried out beforehand.
31. A check is made as to whether the difference between the position actual value from POS1 and POS2 are within the parameterized tolerance window POS_DISC_WINDOW. If this is not the case, then bit 12 in output word DIAG is set to 1.
32. If the safe reference can still be declared to be valid – using the technique described under Point 33 – output ACK_REQ is set to 1 and the fault status can be acknowledged with a positive edge at input ACK. Outputs POS_VALID, REFERENCED and V_VALID are again set to 1, outputs ERROR and ACK_REQ are reset to 0, and bits 9 and 12 in output word DIAG are reset to 0. Position actual value POSSI is aligned with respect to position actual value POS1 as described under Point 11. The position actual value from POS2 is not aligned with respect to position actual value POS1, as the safe reference still exists.

33. If the safe reference cannot be declared to still be valid – using the technique described under Point 33 – then a new safe reference must be performed in the form of a reference point approach (homing). To do this, the encoder can be calibrated using a positive edge at input SYNC as described under Point 11 (output ERROR is reset to 0, and in output word DIAG bits 9 and 12 are reset to 0). As the safe reference no longer exists, in addition to position actual value POSSI, position actual value POS2 is also aligned with respect to position actual value POS1. The safe reference can be performed after the reference point approach as described under Point 16.
34. If SAFE_POS exceeds the value of MAX_POS, output ERROR is set to 1 – and bit 3 in output word DIAG is set to 1. Output POS_VALID is reset to 0.
35. If SAFE_POS again falls below the value of MAX_POS, output ACK_REQ is set to 1 and the fault status can be acknowledged with a positive edge at input ACK. Outputs ERROR and ACK_REQ are reset to 0, bit 3 in output word DIAG is set to 0. Output POS_VALID is again set to 1.
36. If SAFE_POS falls below the value of MIN_POS, output ERROR is set to 1 – and bit 4 in output word DIAG is set to 1. Output POS_VALID is reset to 0.
37. If SAFE_POS again exceeds the value of MIN_POS, output ACK_REQ is set to 1 and the fault status can be acknowledged with a positive edge at input ACK. Outputs ERROR and ACK_REQ are reset to 0, bit 4 in output word DIAG is set to 0. Output POS_VALID is again set to 1.

WARNING

As long as REFERENCED outputs a 0 signal, the position can only be used to establish a relative reference; an absolute evaluation is only permissible for REFERENCED = 1.

In a specific application, when REFERENCED = 0, it is only permissible that the axis traverses with a safely reduced velocity.

WARNING

The position actual value is not generated as safe position actual value as long as POS_VALID supplies a 0 signal. For a falling edge, a stop response (specific to the application) should be initiated.

The safe velocity actual value

38. The safe velocity calculated by the block from the absolute position actual value of the motor encoder interconnected at POSSI is output at SAFE_V.
39. If SAFE_V falls below the value parameterized at V_STANDSTILL, then at output STANDSTILL this standstill is signaled using a 1 signal.
40. If SAFE_V is greater than/equal to V_STANDSTILL, then a 1 signal is output at MOVES_POSITIVE if SAFE_POS, from a time perspective, assumes increasingly higher values – or a 1 signal is output at MOVE_NEGATIVE if the values of SAFE_POS, from a time perspective, assumes increasingly lower values.
41. If SAFE_V exceeds the value parameterized at V_MAX, then ERROR is set to 1, and DIAG bit 8 is set. V_VALID is reset to 0.

42. At input V_DISC_WINDOW it is parameterized how far (specified in mm) the values of POS1 and POSSI can drift apart within V_SYNC_INTERVALL without resulting in a velocity error.
43. After the time parameterized at V_SYNC_INTERVALL, the discrepancy between the relative position of POS1 and POSSI for the velocity monitoring, which has accumulated in the block, is eliminated in order to facilitate slip tolerance.

    If the drift between POS1 and POSSI exceeds the value parameterized at V_DISC_WINDOW, then ERROR and DIAG bit 9 are set. V_VALID is reset to 0.
44. If the drift between POS1 and POSSI exceeds the value parameterized at input V_DISC_WINDOW, then output ERROR is set to 1, and bit 6 in output word DIAG is set to 1. Output V_VALID is reset to 0.
45. If the drift between POS1 and POSSI again falls below the value parameterized at input V_DISC_WINDOW, then output ACK_REQ is set to 1, and the fault state can be acknowledged using a positive edge at input ACK. Outputs ERROR and ACK_REQ are reset to 0, bit 6 in output word DIAG is reset to 0. Output V_VALID is again set to 1.

WARNING

The velocity actual value is not generated as safe velocity actual value as long as V_VALID supplies a 0 signal.

Referencing

46. With a positive edge at REFERENCE, a safe adjustment is performed in the block where, for both raw position values, a separate position offset – relative to the value specified at input POS_REFERENCE – is determined and saved. Output REFERENCED is set if referencing was successfully performed.
47. For successful referencing, both encoder actual values must be valid (POS1_VALID & POSSI_VALID (for POS_CONFIG = 0) or POS1_VALID & POS2_VALID & POSSI_VALID = 1 (for POS_CONFIG = 1)
48. REFERENCED is then set with a rising edge at input REFERENCE – and the offset is internally saved.
49. REFERENCED is reset as soon as one of the tolerances, described for the particular encoder version, is violated.

After an encoder fault, the block is able to reproduce SAFE_POS without requiring a reference point approach. If POS1_VALID and POSSI_VALID or POS2_VALID (depending on POS_CONFIG = 0) have a rising edge, then in the block, the position is declared as valid as described for the particular encoder version.

Note

An encoder fault should always result in the withdrawal of input signal POS1_VALID or POSSI_VALID/POS2_VALID. As a consequence, the safe actual value is immediately declared as invalid, and the position tolerance monitoring hidden. As a consequence, an encoder fault does not mean an immediate loss of the safe reference, and the safe position can be reproduced after the encoder signal returns.

Encoder synchronization

50. Both encoders are synchronized with a positive edge at SYNC, i.e. the discrepancy that has accumulated between the two encoders is brought to 0. To do this, both encoder actual values must be valid.

WARNING	A cyclic synchronization leads to bypass the two channels in the actual position value processing and is therefore not permissible. When synchronization is permitted is application-specific and the corresponding correct logic interconnection is the responsibility of the user.

Acknowledging errors

51. Using a positive edge at ACK, DIAG and ERROR are reset to 0, assuming that an error is no longer active. As soon as the block can be acknowledged, it indicates this using a 1 signal at output ACK_REQ. After a positive edge at ACK, ACK_REQ is reset to 0.

Safety note
Only appropriately qualified and trained technical personnel may acknowledge faults/errors, as only they are in a position to evaluate the reason for the failure and the subsequent safety integrity.

A separate acknowledgment option must be provided to acknowledge these faults/errors.

Additional diagnostic features

To optimize system performance, the currently effective position and velocity discrepancy can be read out of the instance DB of the block for diagnostic purposes.

For position discrepancy, static variable "pos1_pos2_delta" or "pos1_possi_delta", for velocity discrepancy, static variable "v_discrepancy".

3.3 F_SLP_MONITOR

3.3.1 Introduction

The fail-safe function block F_SLP_MONITOR is used to safely monitor the end stops of a travel range. The block signals if the defined traversing range is exited. Depending on the user interconnection, a stop response can be initiated.

The block has retraction logic, so that the storage and retrieval machine can be returned to the permitted travel range. With this function, the machine can be moved away from the end stop with a safe velocity parameterized at the block. The block provides two signals to control the SDI function in the drive, so that traversing back towards the end stop is inhibited.

52.

NOTE When using this block, then block **F_BO_W (FC 176)** must be available in the block folder. It is not permissible to renumber this!

3.3.2 Connections

All bool type variables listed in the following table are preassigned FALSE, all integer variables with 0 and all word variables with W#16#0.

Inputs

Table 3-5

Name	Data type	Description
SAFE_POS	DInt	**Safe position actual value [mm]** is supplied from block F_SAFE_POS.
POS_VALID	Bool	**Actual position valid** is supplied from block F_SAFE_POS. 1: Position is plausible 0: Position is not plausible, i.e. the discrepancy between the two encoders is outside the tolerance window. If a 0 signal is available here, then DIAG bit No. 5 is set.
SAFE_V	DInt	**Safe velocity actual value [mm/min]** is supplied from block F_SAFE_POS.
V_VALID	Bool	**Actual velocity valid** is supplied from block F_SAFE_POS. 1: Velocity is plausible 0: Velocity is not plausible, i.e. the increase of the deviation between the two encoders over time is outside the tolerance window. If a 0 signal is available here, and the block is in the retraction mode, then DIAG bit No. 6 is set.
X_NEGATIVE	DInt	**Minimum permitted position [mm]** If the value at input SAFE_POS falls below this limit value, then output X_NEGATIVE_OK is reset
X_POSITIVE	DInt	**Maximum permitted position [mm]** If the value at input SAFE_POS exceeds this limit value, then output X_POSITIVE_OK is reset
VMAX_RELEASE	DInt	**Retraction velocity [mm/min]**

Name	Data type	Description
		If the block is in the retraction mode, then this value is output at SLS_THRESHOLD. VMAX_RELEASE must be parameterized in the range 1-2147483647. Otherwise, DIAG bit No. 4 is set
RELEASE	Bool	**Retracting** If the permissible position range was exited, then the machine can be traversed back to the permissible position range using this input with the velocity parameterized at VMAX_RELEASE. Retraction motion is immediately stopped as soon as this input has a 0 signal.
ACK	Bool	**Acknowledging** If a fault occurs in normal operation, then this must be reset using ACK before the system can be restarted. Acknowledgment is only realized with a positive edge at ACK - and in fault-free (normal) operation it has no effect.

Outputs

Table 3-6

Name	Data type	Description
SLS_THRESHOLD	DInt	**SLS limit [mm/min]** The presently maximum permissible traversing velocity is output here. This is 2147483647 in normal operation; if the user is retracting the machine, then VMAX_RELEASE is output here. If VMAX_RELEASE has been parameterized <= 0, then an equivalent value of 1 is output here.
SLS_OK	Bool	**SLS limit status** 1: SAFE_V is less than/equal to SLS_THRESHOLD 0: SAFE_V has exceeded the value of SLS_THRESHOLD. If this output changes to 0, then a stop response should be initiated.
X_NEGATIVE_OK	Bool	**Status minimum position** 1: SAFE_POS is greater than/equal to X_NEGATIVE 0: SAFE_POS has fallen below the value of X_NEGATIVE. If this output changes to 0, then a stop response should be initiated.
X_POSITIVE_OK	Bool	**Status maximum position** 1: SAFE_POS is less than/equal to X_POSITIVE 0: SAFE_POS has exceeded the value of X_POSITIVE. If this output changes to 0, then a stop response should be initiated.
MOVE_NEGATIVE_OK	Bool	**Negative motion permitted** If a 0 signal is available at this output, then it is not permissible that the machine continues to move in the negative direction. The output is then set to 0 as soon as SAFE_POS assumes values lower than X_NEGATIVE. If SAFE_POS again lies above X_NEGATIVE, then after acknowledgment, the output is set again.
MOVE_POSITIVE_OK	Bool	**Positive motion permitted**

Name	Data type	Description
		If a 0 signal is available at this output, then it is not permissible that the machine continues to move in the positive direction. The output is then set to 0 as soon as SAFE_POS assumes values higher than X_POSITIVE. If SAFE_POS again lies below X_POSITIVE, then after acknowledgment, the output is set again.
ACK_REQ	Bool	**Acknowledgment request** If a fault has occurred, however it is no longer active, and can therefore be acknowledged, then the block indicates this using a 1 signal at ACK_REQ.
ERROR	Bool	**Error** This output is set if the block has been incorrectly parameterized, or if the block in operation detects a potentially dangerous combination of input signals. The output remains set until an error is no longer active and has been acknowledged.
DIAG	Word	**Diagnostics word** Information about the function status and errors of the block are output here.

Structure of DIAG

Table 3-7

Bit No.	Description	Reset condition
0	Lower end stop was fallen below	While retracting, SAFE_POS >= X_NEGATIVE and positive edge at ACK
1	Upper end stop was exceeded	While retracting, SAFE_POS <= X_POSITIVE and positive edge at ACK
2	Actual velocity higher than the retraction velocity	SAFE_V <= SLS_THRESHOLD and positive edge at ACK
3	Reserved	---
4	Parameterizing error, retraction velocity	0 < VMAX_RELEASE <= 2147483647 parameterized
5	Actual position invalid	Actual position valid again
6	Actual velocity invalid	Actual velocity valid again
7	Reserved	---
8	Reserved	---
9	Reserved	---
10	Reserved	---
11	Reserved	---
12	Reserved	---
13	Reserved	---
14	Reserved	---
15	Reserved	---

3.3.3 Principle of operation

Parameterization

1. At input SAFE_POS, the user must interconnect the safe position actual value of the system to be monitored - and at input POS_VALID its validity AND'ed with the valid reference (REFERENCED). Block "F_SAFE_POS" (Chapter 3.2) provides the three signals as output.
2. The same applies to inputs SAFE_V and V_VALID, which refer to the safe actual velocity.
3. The permitted travel range is parameterized by specifying the upper and lower limits at inputs X_POSITIVE and X_NEGATIVE.
4. VMAX_RELEASE must lie in the range 1 – 2147483647. The block identifies if values less than 1 or values higher than 2147483647 are parameterized and DIAG bit 4 is set. ERROR changes to 1.

If not all of the specified preconditions are satisfied, then the block identifies this, and signals a parameterizing error with the appropriately set DIAG bits.

NOTE

The block only checks the parameterization when it is called for the 1st time. This increases the performance for further block operation.

As a consequence, it is not possible to reparameterize the system while it is in operation. The safety program must be regenerated and loaded each time that the block operating parameters are changed.

Position monitoring

5. As long as the position actual value is valid, and is in the permitted range, the block does not signal an error; this means that outputs ERROR and DIAG supply a 0 signal.
6. If, although the position actual value is in the permitted range, but via POS_VALID = 0 is declared invalid, then an error code is also output to DIAG. Until acknowledged, ERROR remains in the actual state - assuming that no additional faults occur as a result of another active monitoring function. All other outputs maintain their actual state until acknowledgment or RELEASE is deselected. In this case, DIAG bit No. 5 is set
7. As soon as POS_VALID changes back to 1, DIAG bit No. 5 again has a 0 signal.
8. As soon as SAFE_POS is outside the parameterized travel range, depending on the direction in which this was exited, either X_POSITIVE_OK or X_NEGATIVE_OK is set to 0. A stop response should then be initiated in the drive by the user interconnection.
9. In addition, DIAG bit No. 0 for falling below the lower end stop - or DIAG bit No. 1 for exceeding the upper end stop is set. ERROR is set to 1.

Safety note

Block F_SAFE_POS provides a 0 signal at POS_VALID via output ERROR = 1. However, when POS_VALID goes to zero, a stop response must be initiated in the drive via a user interconnection. All other blocks indicate the status using an error code; however, in order to avoid being confronted by a flood of messages/signals, in this case ERROR is not again set to a 1 signal. This means that the end stops are no longer monitored. Active faults for the end position monitoring can be immediately acknowledged X_NEGATIVE_OK, X_POSITIVE_OK and SLS_OK are set again.

If a 1 signal is again available at POS_VALID, then the associated DIAG bit 5 is reset, and end position monitoring is resumed.

Retracting

10. To traverse from the end stop back into the permitted travel range, the retraction function of the block can be activated using a positive edge at RELEASE. The velocity parameterized at VMAX_RELEASE is then output at SLS_THRESHOLD, and depending on the direction of travel when the end zone was violated, either MOVE_POSITIVE_OK or MOVE_NEGATIVE_OK is set to 0 in order to prevent further motion towards this end zone. MOVE_POSITIVE_OK = 0 inhibits motion in the positive direction, MOVE_NEGATIVE_OK = 0 inhibits motion in the negative direction.

NOTE The signal for RELEASE must be generated in a safety-related fashion, e.g. by using a key-operated switch or similar device.

11. In order to facilitate retraction, when RELEASE is selected, X_POSITIVE_OK or X_NEGATIVE_OK is reset to 1; the stop response of the drive should be deselected using a suitable user interconnection.
12. If, during retraction, the value of SAFE_V exceeds the value of VMAX_RELEASE, then SLS_OK changes to 0 and DIAG bit 2 is set.
13. A velocity fault can always be acknowledged if the actual velocity SAFE_V is again below SLS_THRESHOLD.
14. As soon as SAFE_POS is back in the parameterized permissible range, after acknowledgment, the machine can again be traversed with the full velocity, i.e. SLS_THRESHOLD is reset to maximum velocity (maximum possible DINT value = 2147483647).
15. If, when retracting, the removal and storage machine should traverse to the opposite end stop, then the block behaves in just the same way as the corresponding end range violation in normal operation. This means that it again changes X_POSITIVE_OK or X_NEGATIVE_OK to 0, which means motion is only possible in the direction away from the end stop.
16. If V_VALID = 0 while SAFE_POS lies outside the parameterized travel range, then the retraction velocity can no longer be monitored in a safely-related way. Therefore, selection using RELEASE = 1 has no effect, and retraction motion is stopped.
17. To exit this state, V_VALID must first be again set to a 1 signal using F_SAFE_POS by acknowledging.
18. Retraction can then be continued. Alternatively, the initial state can be restored by deselecting RELEASE and subsequent acknowledgment. If SAFE_POS still

lies outside the parameterized traversing range, then the system responds corresponding to Point 9.

Safety note
Block F_SAFE_POS signals a 0 signal at V_VALID via output ERROR = 1. However, when V_VALID goes to zero, a stop response must be initiated in the drive via a user interconnection. All other blocks indicate the status using an error code; however, in order to avoid being confronted by a flood of messages/signals, in this case ERROR is not again set to a 1 signal. This means that the retraction velocity is no longer monitored. Active faults of the retraction monitoring can be immediately acknowledged, SLS_OK is again set. Retraction via RELEASE can be exited normally; MOVE_POSITIVE_OK and MOVE_NEGATIVE_OK are again set. If, at this instant in time, the axis is not in a valid position range, then X_NEGATIVE_OK or X_POSITIVE_OK is withdrawn and ERROR is set.

If a 1 signal is again available at V_VALID, then the associated DIAG bit 6 is reset, and if retraction motion is pending, then this is again monitored.

Safety note

The parameterization of input V_MAX_RELEASE must be adapted according to the permissible safely reduced velocity, derived from the application-specific risk assessment.

Safety note

The interconnection of output MOVE_POSITIVE_OK must match the selection of drive function SDI for a positive direction. For MOVE_POSITIVE_OK = 0, motion in the positive direction must no longer be possible.

The same is true when interconnecting output MOVE_NEGATIVE_OK and inhibiting the negative direction of motion.

It is absolutely crucial that the block outputs are interconnected with the correct signals to control the drive.

Otherwise, inadmissible motion towards the end stops is possible, which cannot be identified internally in the block.

<u>Acknowledging</u> errors

19. DIAG and ERROR are reset to 0 using a positive edge at ACK, assuming that there are no active errors. As soon as the block can be acknowledged, it indicates this using a 1 signal at output ACK_REQ. After a positive edge at ACK, ACK_REQ is reset to 0.

3.4 F_ENDZONE

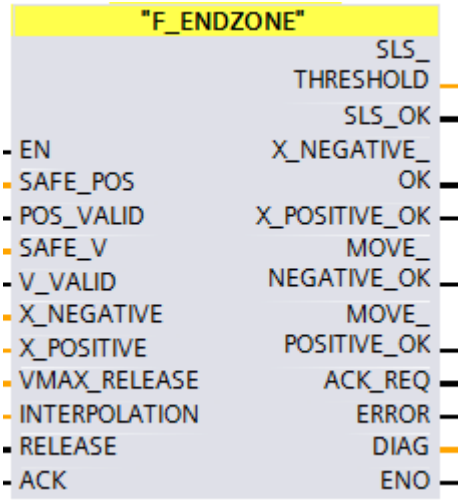
3.4.1 Introduction

The fail-safe function block F_ENDZONE is used to safely monitor the end stops of a traversing range - or the collision monitoring of two systems. If the system that is being monitored approaches the parameterizable positive or negative end stop, then its maximum permissible velocity, dependent on the actual position, is limited according to a parameterizable curve until standstill is reached. This curve is parameterized using 17 interpolation points; the system linearly interpolates the curve between these points.

If the storage and retrieval machine traverses beyond the end stop, or the maximum permissible velocity is exceeded, then the block signals this situation. Depending on the user interconnection, a stop response can be initiated.

The block has retraction logic, so that the storage and retrieval machine can be returned to the permitted traversing range if it passes and end stop. With this function, the machine can be moved away from the end stop with a low safe velocity parameterized at the block. The block provides two signals to control the SDI function in the drive, so that traversing back towards the end stop is inhibited.

Figure 3-7



NOTE

When using this block, then block **F_BO_W (FC 176)** must be available in the block folder. It is not permissible to renumber this! In addition, blocks **LFAddDInt (FC 211)**, **LFSubDInt (FC 212)** and **F_INTERPOLATION (FC 200)** are required from this library. These may be renumbered, but not renamed.

In addition, F-UDT **INTERP** is required from this library.

3.4.2 Connections

All bool type variables listed in the following table are preassigned FALSE, all integer variables with 0 and all word variables with W#16#0.

Inputs

Table 3-8

Name	Data type	Description
SAFE_POS	DInt	**Safe position actual value [mm]** is supplied from block F_SAFE_POS.
POS_VALID	Bool	**Actual position valid** is supplied from block F_SAFE_POS. 1: Position is plausible 0: Position is not plausible, i.e. the discrepancy between the two encoders is outside the tolerance window.
SAFE_V	DInt	**Safe velocity actual value [mm/min]** is supplied from block F_SAFE_POS.
V_VALID	Bool	**Actual velocity valid** is supplied from block F_SAFE_POS. 1: Velocity is plausible 0: Velocity is not plausible, i.e. the increase of the deviation between the two encoders over time is outside the tolerance window.
X_NEGATIVE	DInt	**Minimum permitted position [mm]** If the value at input SAFE_POS falls below this limit value, then output X_NEGATIVE_OK is reset
X_POSITIVE	DInt	**Maximum permitted position [mm]** If the value at input SAFE_POS exceeds this limit value, then output X_POSITIVE_OK is reset
VMAX_RELEASE	DInt	**Retraction velocity [mm/min]** If the block is in the retraction mode, then this value is output at SLS_THRESHOLD. VMAX_RELEASE must be parameterized in the range 1 <= VMAX_RELEASE <= INTERPOLATION.V16.
INTERPOLATION	F-UDT "INTERP"	**F-UDT with interpolation points to define the end zone according to Chapter 3.4.2**
RELEASE	Bool	**Retracting** If the permissible position range was exited, then the machine can be traversed back to the permissible position range by selecting this input with the velocity parameterized at VMAX_RELEASE. Retraction motion is immediately stopped as soon as this input has a 0 signal.
ACK	Bool	**Acknowledging** If a fault occurs in normal operation, then this must be reset using ACK before the system can be restarted. Acknowledgment is only realized with a positive edge at ACK - and in fault-free (normal) operation it has no effect.

Outputs

Table 3-9

Name	Data type	Description
SLS_THRESHOLD	DInt	**SLS limit [mm/min]** The presently maximum permissible traversing velocity is output here. This is cyclically calculated in the block itself.
SLS_OK	Bool	**SLS limit status** 1: SAFE_V is less than/equal to SLS_THRESHOLD 0: SAFE_V has exceeded the value of SLS_THRESHOLD. If this output changes to 0, then a stop response

Name	Data type	Description
		should be initiated.
X_NEGATIVE_OK	Bool	**Status minimum position** 1: SAFE_POS is greater than/ equal X_NEGATIVE 0: SAFE_POS has fallen below the value of X_NEGATIVE. If this output changes to 0, then a stop response should be initiated.
X_POSITIVE_OK	Bool	**Status maximum position** 1: SAFE_POS is less than/equal to X_POSITIVE 0: SAFE_POS has exceeded the value of X_POSITIVE. If this output changes to 0, then a stop response should be initiated.
MOVE_NEGATIVE_OK	Bool	**Negative motion permitted** If a 0 signal is available at this output, then it is not permissible that the machine continues to move in the negative direction. The output is then set to 0 as soon as SAFE_POS assumes values lower than X_NEGATIVE. If SAFE_POS again lies above X_NEGATIVE, then after acknowledgment, the output is set again.
MOVE_POSITIVE_OK	Bool	**Positive motion permitted** If a 0 signal is available at this output, then it is not permissible that the machine continues to move in the positive direction. The output is then set to 0 as soon as SAFE_POS assumes values higher than X_POSITIVE. If SAFE_POS again lies below X_POSITIVE, then after acknowledgment, the output is set again.
ACK_REQ	Bool	**Acknowledgment request** If a fault has occurred, however it is no longer active, and can therefore be acknowledged, then the block indicates this using a 1 signal at ACK_REQ.
ERROR	Bool	**Error** This output is set if the block has been incorrectly parameterized, or if the block in operation detects a potentially dangerous combination of input signals. The output remains set until an error is no longer active and has been acknowledged.
DIAG	Word	**Diagnostics word** Information about the function status and errors of the block are output here.

Structure of DIAG

Table 3-10

Bit No.	Description	Reset condition
0	Lower end stop was fallen below	While retracting, SAFE_POS >= X_NEGATIVE and positive edge at ACK
1	Upper end stop was exceeded	While retracting, SAFE_POS <= X_POSITIVE and positive edge at ACK
2	Retraction velocity exceeded	SAFE_V <= SLS_THRESHOLD and positive edge at ACK
3	Parameterizing error envelope curve, for	Envelope curve according to 3.9

Bit No.	Description	Reset condition
	remaining distance 0, the velocity is not 0	
4	Parameterizing error, retraction velocity	0 < VMAX_RELEASE <= V_MAX parameterized
5	Actual position invalid	Actual position valid again
6	Actual velocity invalid	Actual velocity valid again
7	Reserved	---
8	Reserved	---
9	Actual velocity too high regarding actual position and direction	SAFE_V <= SLS_THRESHOLD and positive edge at ACK
10	Reserved	---
11	Error for internal calculation	Pos. edge at ACK, if the fault is no longer active
12	Reserved	---
13	Reserved	---
14	Reserved	---
15	Reserved	---

F-UDT „INTERP“

Table 3-11

Name	Data type	Description
X_0	DInt	Remaining distance point 0 [mm]
X_1	DInt	Remaining distance point 1 [mm]
X_2	DInt	Remaining distance point 2 [mm]
X_3	DInt	Remaining distance point 3 [mm]
X_4	DInt	Remaining distance point 4 [mm]
X_5	DInt	Remaining distance point 5 [mm]
X_6	DInt	Remaining distance point 6 [mm]
X_7	DInt	Remaining distance point 7 [mm]
X_8	DInt	Remaining distance point 8 [mm]
X_9	DInt	Remaining distance point 9 [mm]
X_10	DInt	Remaining distance point 10 [mm]
X_11	DInt	Remaining distance point 11 [mm]
X_12	DInt	Remaining distance point 12 [mm]
X_13	DInt	Remaining distance point 13 [mm]
X_14	DInt	Remaining distance point 14 [mm]
X_15	DInt	Remaining distance point 15 [mm]
X_16	DInt	Remaining distance point 16 [mm]
V_0	DInt	Velocity at remaining distance point 0 [mm/min]
V_1	DInt	Velocity at remaining distance point 1 [mm/min]
V_2	DInt	Velocity at remaining distance point 2 [mm/min]
V_3	DInt	Velocity at remaining distance point 3 [mm/min]
V_4	DInt	Velocity at remaining distance point 4 [mm/min]
V_5	DInt	Velocity at remaining distance point 5 [mm/min]
V_6	DInt	Velocity at remaining distance point 6 [mm/min]
V_7	DInt	Velocity at remaining distance point 7 [mm/min]
V_8	DInt	Velocity at remaining distance point 8 [mm/min]
V_9	DInt	Velocity at remaining distance point 9 [mm/min]
V_10	DInt	Velocity at remaining distance point 10 [mm/min]
V_11	DInt	Velocity at remaining distance point 11 [mm/min]
V_12	DInt	Velocity at remaining distance point 12 [mm/min]
V_13	DInt	Velocity at remaining distance point 13 [mm/min]
V_14	DInt	Velocity at remaining distance point 14 [mm/min]

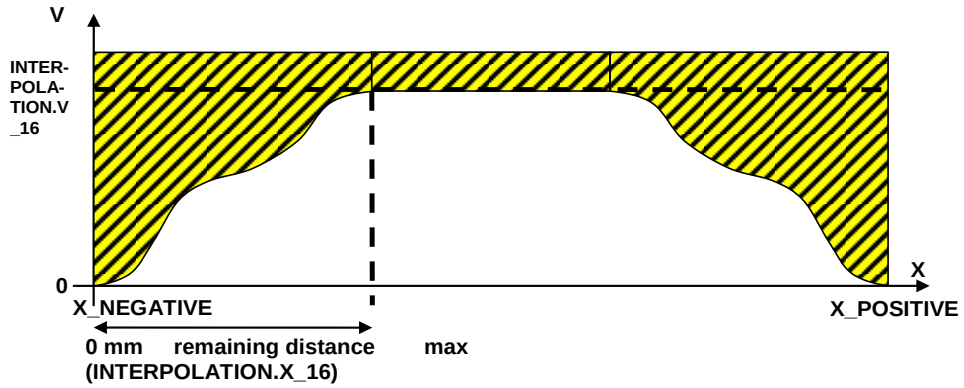
3.4.3 Principle of operation

Parameterization

1. At input SAFE_POS, the user must interconnect the safe position actual value of the system to be monitored - and at input POS_VALID its validity AND'ed with the valid reference (REFERENCED). Block "F_SAFE_POS" (Chapter 3.2) provides the three signals as output.
2. The same applies to inputs SAFE_V and V_VALID, which refer to the safe actual velocity.
3. The lower end position is parameterized via input X_NEGATIVE, and the upper end position is parameterized via X_POSITIVE.
4. The velocity envelope curve of the end zone to be monitored is parameterized using 17 interpolation points (velocity with respect to distance) using the F-UDT "INTERP", referred to the remaining distance to an end stop. To do this, a fail-safe global data block should be created, in which a variable, type "INTERP" is created. The end zone curve is defined using the start values of the positions and the associated velocities. The intermediate values between the interpolation points is calculated using linear interpolation. The velocity envelope curves are symmetrical for the positive and negative end zones.

NOTE The deceleration referred to the remaining distance to standstill is defined using a root function. The remaining distance to an end stop is used as basis to determine the velocity limit. Using linear interpolation over 17 interpolation points, this root function must be emulated corresponding to the braking response of the specific application

5. Interpolation point INTERPOLATION.X_0/ INTERPOLATION.V_0 must be parameterized with INTERPOLATION.X_0 := 0 mm and INTERPOLATION.V_0 := 0 mm/min. Velocity values parameterized higher than the interpolation value INTERPOLATION.V_16, are limited to this value.
6. VMAX_RELEASE must lie in the range 1 – INTERPOLATION.V_16. The block identifies if values lower than 1 or higher than INTERPOLATION.V_16 are parameterized, and signals this with DIAG bit 4. ERROR changes to 1.



If not all of the specified preconditions are satisfied, then the block identifies this, and signals a parameterizing error with the appropriately set DIAG bits.

Safety note

The user **must** validate the parameterized envelope curve and the correct functionality by recording traces and carrying out the appropriate tests (see Chapter 3.1.1).

NOTE The requirements relating to monotony and gradient of the envelope curve depend on the specific application and the risk assessment.

NOTE The block only checks the parameterization when it is called for the 1st time. This increases the performance for further block operation.

As a consequence, reparameterization is not possible while the system is in operation, with the exception of X_ NEGATIVE and X_ POSITIVE. The safety program must be regenerated and loaded each time that the block operating parameters are changed.

Position and velocity monitoring

7. As long as the position actual value is valid, and SAFE_V lies below the parameterized velocity envelope curve, the block does not signal an error; this means that outputs ERROR and DIAG supply a 0 signal.
8. At output SLS_THRESHOLD, depending on SAFE_POS, the associated maximum permissible velocity for this position is output.
9. If the value at input SAFE_V lies above this limit, and if the system is moving towards the end stop, then output SLS_OK is set to 0, ERROR changes to 1 and DIAG bit No. 9 is set.
10. Depending on the user interconnection, a stop response must be initiated in the drive.
11. As soon as SAFE_V is again in the permissible range, i.e. lower than SLS_THRESHOLD, then the fault can be acknowledged and ACK_REQ outputs a 1 signal.
12. The fault can be reset with a positive edge at ACK. ERROR and the appropriate DIAG bits then change back to 0, and SLS_OK outputs a 1 signal.
13. If the value at input SAFE_V lies above the permitted velocity, but the system is moving away from the end stop, then the system may traverse with 100% velocity; INTERPOLATION.V16 is output at SLS_THRESHOLD. As a consequence, a fault is not signaled, and ERROR and DIAG remain at 0.
14. If a 0 signal is available at POS_VALID, then DIAG bit 5 is set; until acknowledged, ERROR remains in the actual state - assuming that no additional monitoring functions signals a fault condition. All other outputs maintain their actual state until acknowledgment or RELEASE is deselected.

Safety note

Block F_SAFE_POS provides a 0 signal at POS_VALID via output ERROR = 1. However, when POS_VALID goes to zero, a stop response must be initiated in the drive via a user interconnection. All other blocks indicate the status using an error code; however, in order to avoid being confronted by a flood of messages/signals, in this case ERROR is not again set to a 1 signal. This means that the end stops and the envelope curve are no longer monitored. Maximum velocity INTERPOLATION.V16 - as well as the validity of the velocity actual value SAFE_V, are still monitored. Active faults for the end position and envelope curve monitoring can be immediately acknowledged X_NEGATIVE_OK, X_POSITIVE_OK and SLS_OK are set again.

If a 1 signal is again available at POS_VALID, then DIAG bit 5 is reset, and the end position and the envelope curve monitoring are resumed.

15. If a 1 signal is again available at POS_VALID, then the associated DIAG bit 5 is reset.
16. If a 0 signal is available at V_VALID, then DIAG bit 6 is set; until acknowledged, ERROR remains in the actual state - assuming that no additional active monitoring functions signals a fault condition. All other outputs maintain their actual state until acknowledgment or RELEASE is deselected.

Safety note

Block F_SAFE_POS signals a 0 signal at V_VALID via output ERROR = 1. However, when V_VALID goes to zero, a stop response must be initiated in the drive via a user interconnection. All other blocks indicate the status using an error code; however, in order to avoid being confronted by a flood of messages/signals, in this case ERROR is not again set to a 1 signal. This means that the maximum velocity and the envelope curve are no longer monitored. Active faults of the maximum velocity and envelope curve monitoring can be immediately acknowledged, SLS_OK is again set. Retraction via RELEASE can be exited normally; MOVE_POSITIVE_OK and MOVE_NEGATIVE_OK are again set. If, at this instant in time, the axis is not in a valid position range, then X_NEGATIVE_OK or X_POSITIVE_OK is withdrawn and ERROR is set.

If a 1 signal is again available at V_VALID, then the associated DIAG bit 6 is reset, and the maximum velocity and the envelope curve monitoring are resumed.

17. If a 1 signal is again available at V_VALID, then the associated DIAG bit 6 is reset.
18. If the permissible traversing range is exited, i.e. SAFE_POS assumes values greater than X_POSITIVE or less than X_NEGATIVE, then the block behaves in a comparable fashion to block "F_SLP_MONITOR" (Chapter 3.3).
19. Depending on the direction in which the traversing range was exited, either X_POSITIVE_OK or X_NEGATIVE_OK is set to 0. A stop response should then be initiated in the drive by the user interconnection.

Retracting

20. The retraction function of the block can be activated by selecting RELEASE. If the system is within the permissible traversing range, then

MOVE_POSITVE_OK and MOVE_NEGATIVE_OK are reset to 1, the velocity parameterized at VMAX_RELEASE is output at SLS_THRESHOLD and internally this value is monitored. The velocity envelope curve is still monitored. If this supplies a more restrictive value for the permitted velocity than VMAX_RELEASE, then the permissible velocity is limited to the more restrictive (lower) value. If the permissible traversing range is exited, then the block responds as described under Point 18.

21. To traverse from the end stop back into the permitted traversing range, the retraction function of the block can be activated by selecting RELEASE. The velocity parameterized at VMAX_RELEASE is then output at SLS_THRESHOLD, and depending on the direction of travel when the end zone was violated, either MOVE_POSITIVE_OK or MOVE_NEGATIVE_OK is set to 0 in order to prevent further motion towards this end zone. MOVE_POSITIVE_OK = 0 inhibits motion in the positive direction, MOVE_NEGATIVE_OK = 0 inhibits motion in the negative direction.

NOTE The signal for RELEASE must be generated in a safety-related fashion, e.g. by using a key-operated switch or similar device.

22. In order to facilitate retraction, when RELEASE is selected, X_POSITIVE_OK or X_NEGATIVE_OK is reset to 1; the stop response of the drive should be deselected using a suitable user interconnection.
23. If, during retraction, the value of SAFE_V exceeds the value of VMAX_RELEASE - or the permissible velocity of the opposite end zone, if this is below VMAX_RELEASE, then SLS_OK changes to 0 and DIAG bit 2 is set.
24. A velocity fault can always be acknowledged if the actual velocity SAFE_V is again below SLS_THRESHOLD.
25. As soon as SAFE_POS is back in the parameterized permissible range, after acknowledgment, the machine can again be traversed with full velocity, i.e. the permissible velocity of the envelope curve monitoring is output at SLS_THRESHOLD and monitored. ERROR and DIAG change back to 0.
26. If V_VALID = 0 while SAFE_POS lies outside the parameterized traversing range, then the retraction velocity can no longer be monitored in a safety-related way. Therefore, selection using RELEASE = 1 has no effect.
27. To resume retraction, after acknowledgment, at block F_SAFE_POS, V_VALID must be again set to a 1 signal.
28. Retraction can then be continued. Alternatively, the initial state can be restored by deselecting RELEASE and subsequent acknowledgment. If SAFE_POS still lies outside the parameterized traversing range, then the system responds corresponding to Point 17.
29.

The parameterization of input VMAX_RELEASE must be adapted according to the permissible safely reduced velocity, derived from the application-specific risk assessment.

WARNING

The interconnection of output MOVE_POSITIVE_OK must match the selection of drive function SDI for a positive direction. For MOVE_POSITIVE_OK = 0, motion in the positive direction must no longer be possible.
The same is true when interconnecting output MOVE_NEGATIVE_OK and inhibiting the negative direction of motion.
It is absolutely crucial that the block outputs are interconnected with the correct signals to control the drive.

Otherwise, inadmissible motion towards the end stops is possible, which cannot be identified internally in the block.

Acknowledging errors

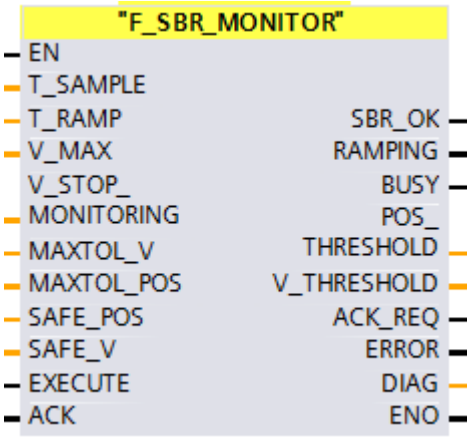
DIAG and ERROR are reset to 0 using a positive edge at ACK, assuming that there are no active errors. As soon as the block can be acknowledged, it indicates this using a 1 signal at output ACK_REQ. After a positive edge at ACK, ACK_REQ is reset to 0.

3.5 F_SBR_MONITOR

3.5.1 Introduction

The fail-safe function block F_SBR_MONITOR monitors that the braking ramp is maintained. If, for example after SS1 has been initiated, the velocity is not reduced in the drive along the parameterized down ramp, the block outputs a signal to initiate STO or to close the brake.

Figure 3-8



NOTE When using this block, then block **F_BO_W (FC 176)** must be available in the block folder. It is not permissible to renumber this! Blocks **LFAddDInt (FC 211)**, **LFSubDInt (FC 212)**, **LFMulDInt (FC 213)** and **LFDivDInt (FC 214)** are also required from this library. These may be renumbered, but not renamed.

3.5.2 Connections

All bool type variables listed in the following table are preassigned FALSE, all integer variables with 0 and all word variables with W#16#0.

Inputs

Table 3-12

Name	Data type	Description
T_SAMPLE	DInt	**Sampling time [ms]** Here, the block sampling time, i.e. the interval in which the safety program is called (cyclic interrupt OB interval for the F-OB) is parameterized in ms.
T_RAMP	DInt	**Ramp-down time [ms]** Here, the corresponding value in ms for the ramp-down time from maximum velocity down to standstill is parameterized as in the drive. In conjunction with V_MAX, the gradient of the down ramp is calculated from this value. It must be ensured that T_RAMP/T_SAMPLE is an integer multiple.
V_MAX	DInt	**Max. permissible velocity [mm/min]** Here, the corresponding value for the maximum operating velocity is parameterized, just the same as in the drive, whereby the conversion from rpm to mm/min must be observed. From this value, in conjunction with T_RAMP, the down ramp gradient is calculated. It

Name	Data type	Description
		must be ensured that V_MAX/(T_RAMP/T_SAMPLE) is an integer multiple.
V_STOP_MONITORING	DInt	**Shutdown threshold for monitoring [mm/min]** As soon as the actual velocity falls below this threshold, after the brake ramp monitoring has been initiated, the block can be acknowledged.
MAXTOL_V	DInt	**Velocity tolerance [mm/min]** Max. permissible value that SAFE_V can exceed the configured braking ramp
MAXTOL_POS	DInt	**Position tolerance [mm]** Max. value that SAFE_POS can exceed the position limit according to the configured braking ramp
SAFE_POS	DInt	**Safe position actual value [mm]** is supplied from block F_SAFE_POS. The block derives the velocity from the rate that this value changes (with respect to time). If, after SS1 is initiated, the block identifies that the drive does not brake along the configured ramp, then at SBR_OK the block sets a 0 signal - and as a consequence, for example STO is initiated or a mechanical brake can be closed.
SAFE_V	DInt	**Safe velocity actual value [mm/min]** This is supplied from the F_SAFE_POS block; the motor encoder is the signal source, which is read-in via the SI part of the drive. If, after SS1 is initiated, the block identifies that the drive does not brake along the configured ramp, then at SBR_OK the block sets a 0 signal - and as a consequence, for example STO is initiated or a mechanical brake can be closed.
EXECUTE	Bool	**Starting monitoring...** The block becomes active with a rising edge at this input, i.e. braking ramp monitoring is started (e.g. interconnection with a bit "SS1 active" from the PROFIsafe telegram 902)
ACK	Bool	**Acknowledging** If a fault occurs in normal operation, then this must be reset using ACK before the system can be restarted.

Outputs

Table 3-13

Name	Data type	Description
SBR_OK	Bool	**Status of the braking ramp monitoring** 1: The braking ramp is maintained - or monitoring is not active. 0: The drive brakes but not to achieve the minimum configured down ramp If this output changes to 0, then STO should be initiated or a mechanical brake closed.
RAMPING	Bool	**Braking ramp status** 1: Braking active
BUSY	Bool	**Status of the ramp monitoring** 1: Position and velocity limit monitoring active
POS_THRESHOLD	DInt	**Position limit value [mm]** Effective limit for the ramp monitoring regarding the

Name	Data type	Description
		position change
V_THRESHOLD	DInt	**Velocity limit value [mm/min]** Effective limit for the ramp monitoring regarding the velocity
ACK_REQ	Bool	**Acknowledgment request** If a fault has occurred, however it is no longer active, and can therefore be acknowledged, then the block indicates this using a 1 signal at ACK_REQ.
ERROR	Bool	**Error** This output is set if the block has been incorrectly parameterized, or if the block in operation detects that the SS1 braking ramp has been violated. The output remains set until an error is no longer active and has been acknowledged.
DIAG	Word	**Diagnostics word** Information about the function status and errors of the block are output here.

Structure of DIAG

Table 3-14

Bit No.	Description	Reset condition
0	SS1 braking ramp is not maintained	SAFE_V falls below V_STOP_MONITORING and a positive edge at ACK
1	Parameterizing error T_RAMP: Not a multiple integral of T_SAMPLE	The ratio between T_RAMP and T_SAMPLE is an integer multiple
2	Reserved	---
3	Parameterizing error V_MAX: V_MAX / (T_RAMP / T_SAMPLE) cannot be represented as integer multiple	The ratio between V_MAX and the number of cycles, given by T_RAMP and T_SAMPLE for the braking ramp, is an integer multiple.
4	T_SAMPLE <= 0	T_SAMPLE parameterized > 0
5	MAXTOL_V > V_MAX	MAXTOL_V parameterized <= V_MAX
6	T_RAMP < 0	T_RAMP parameterized >= 0
7	Reserved	---
8	Reserved	---
9	Reserved	---
10	Reserved	---
11	Error for internal calculation	Pos. edge at ACK, if the fault is no longer active
12	Reserved	---
13	Reserved	---
14	Reserved	---

3.5.3 Principle of operation

Parameterization

1. The actual velocity calculated by block F_SAFE_POS is interconnected to the SAFE_V input.
2. The braking ramp gradient is determined in the block using T_RAMP and V_MAX. Using T_SAMPLE and T_RAMP, the number of cycles are determined that are required to brake from V_MAX down to standstill. In each cycle, the maximum permissible velocity internally calculated in the block is then appropriately reduced.

When parameterizing, it must be ensured that the following relationships can be represented as integer multiples:

T_RAMP / T_SAMPLE

V_MAX / (T_RAMP/T_SAMPLE)

Further, the following relationship between the input variables must apply:

MAXTOL_V <= V_MAX

The permissible value ranges of the individual inputs should be taken from the table describing the inputs.

If not all of the specified preconditions are satisfied, then the block identifies this, and signals a parameterizing error with the appropriately set DIAG bits.

Safety note

The parameterization of V_MAX and T_RAMP must be selected so that when it is identified that the permissible traversing range has been exited (with the associated stop response), when an STO is initiated, the axis can be braked to standstill before the physical end of the traversing range is reached.

NOTE

The block only checks the parameterization when it is called for the 1st time. This increases the performance for further block operation.

As a consequence, it is not possible to reparameterize the system while it is in operation. The safety program must be regenerated and loaded each time that the block operating parameters are changed.

Ramp monitoring

3. Braking ramp monitoring is activated with a rising edge at EXECUTE.
4. If SAFE_V exceeds the internally calculated maximum permissible value, then output SBR_OK changes to 0, ERROR to 1 and DIAG bit 0 is set.

5. SBR_OK is also set to 0, if, in each cycle, SAFE_POS changes by more than the maximum position change per cycle internally calculated by the block. This establishes a two channel ramp monitoring configuration.
6. In this case, ERROR also changes to 1 - and DIAG bit 0 is set.
7. Monitoring is exited as soon as EXECUTE is reset to 0, and the internally calculated velocity ramp has reached a value of 0.
8. A tolerance value for the velocity and positioning monitoring can be parameterized via inputs MAXTOL_V and MAXTOL_POS. SBR_OK is then set to 0 if SAFE_V exceeds the internally calculated ramp + MAXTOL_V - or if the position increase with respect to the position at the instant of the selection is greater than the internally calculated maximum value + MAXTOL_POS.

WARNING

For a 0 signal at SBR_OK, STO must be immediately initiated or the mechanical brake(s) closed.

Acknowledging errors

9. DIAG and ERROR are reset to 0 using a positive edge at ACK, assuming that there are no active errors.
10. As soon as the block can be acknowledged, it indicates this using a 1 signal at output ACK_REQ. After a positive edge at ACK, ACK_REQ is reset to 0.
11. After SBR_OK has changed to a 0 signal, i.e. the braking ramp was not maintained, the block can only be acknowledged if the actual velocity at SAFE_V falls below the value at V_STOP_MONITORING. ACK_REQ then changes to a 1 signal.

3.6 F_BRAKE_TEST

3.6.1 Introduction

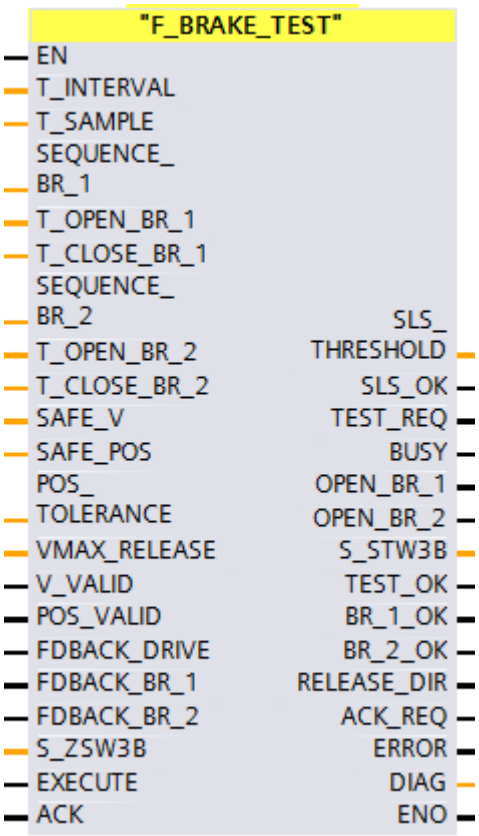
The fail-safe function block F_BRAKE_TEST is used to control the drive function SBT to test a motor holding brake or an external brake.

The torque setpoint and the test profiles are saved in the SI section of the drive in the Safe Brake Test (SBT). When requested, the block automatically coordinates the parameterized test sequences.

The functions of two independent brakes are then consecutively tested by establishing a torque against the closed brake.

If the brake test was unsuccessful, the block supports retraction logic with SDI and SLS. This means that only traversing/travel motion with reduced velocity, and for example for a hoisting gear application, only in the downward direction.

Figure 3-9



NOTE When using this block, blocks **F_BO_W (FC 176), F_W_BO (FC 177)** and **F_TP (FB 184)** must be available in the block folder. It is not permissible to renumber these! Blocks **LFAddDInt (FC 211)**, **LFSubDInt (FC 212)**, **LFMulDInt (FC 213)** and **LFDivDInt (FC 214)** are also required from this library. These may be renumbered, but not renamed.

3.6.2 Connections

All bool type variables listed in the following table are preassigned FALSE, all integer variables with 0, all TIME variables with T#0ms and all word variables with W#16#0.

Inputs

Table 3-15

Name	Data type	Description
T_INTERVAL	Time	**Test interval** After this time has elapsed, the block requests that a brake test is performed. This is signaled at output TEST_REQ using a 1 signal.
T_SAMPLE	DInt	**Sampling time [ms]** Here, the block sampling time, i.e. the interval in which the safety program is called (cyclic interrupt OB interval for the F-OB) is parameterized in ms.
SEQUENCE_BR_1	Word	**Configuration parameters** The test pattern to be executed and the brake type for brake 1 are defined according to the following schematic via this input: Bit 0: Test with test sequence 1 positive Bit 1: Test with test sequence 1 negative Bit 2: Test with test sequence 2 positive Bit 3: Test with test sequence 2 negative Bit 4: 0: external brake; 1: motor holding brake
T_OPEN_BR_1	DInt	**Opening time brake 1 [ms]** Within this time, the brake must completely open; otherwise a read back error is identified - and the test is exited as having not been successfully completed. In this case, DIAG bit 0 is also set.
T_CLOSE_BR_1	DInt	**Closing time brake 1 [ms]** Within this time, the brake must completely close; otherwise a read back error is identified - and the test is exited as having not been successfully completed. In this case, DIAG bit 0 is also set.
SEQUENCE_BR_2	Word	**Configuration parameters** The test pattern to be executed and the brake type for brake 2 are defined according to the following schematic via this input: Bit 0: Test with test sequence 1 positive Bit 1: Test with test sequence 1 negative Bit 2: Test with test sequence 2 positive Bit 3: Test with test sequence 2 negative Bit 4: 0: external brake; 1: motor holding brake
T_OPEN_BR_2	DInt	**Opening time brake 2 [ms]** Within this time, the brake must completely open; otherwise a read back error is identified - and the test is exited as having not been successfully completed. In this case, DIAG bit 1 is also set.
T_CLOSE_BR_2	DInt	**Closing time brake 2 [ms]** Within this time, the brake must completely close; otherwise a read back error is identified - and the test is exited as having not been successfully completed. In this case, DIAG bit 1 is also set.
SAFE_V	DInt	**Safe velocity actual value [mm/min]**

Name	Data type	Description
		is supplied from block F_SAFE_POS. If, for a brake test that has not been successfully completed, the actual velocity is greater than the upper limit parameterized at VMAX_RELEASE then output SLS_OK is reset and the machine is stopped. In this case, DIAG bit 2 is also set.
SAFE_POS	DInt	**Safe position actual value [mm]** is supplied from block F_SAFE_POS. This is required to monitor standstill during the brake test. If the axis moves by more than the value parameterized at POS_TOLERANCE, then the test is considered to have not been successfully completed, and is exited. In this case, DIAG bit 3 is also set.
POS_TOLERANCE	DInt	**Threshold for standstill detection [mm]** If the axis moves by more than this absolute value, then the test is considered to have not been successfully completed, and is exited. In this case, DIAG bit 3 is also set.
VMAX_RELEASE	DInt	**Retraction velocity [mm/min]** If the test was not successfully completed, then this value is output at SLS_THRESHOLD until a brake test has been successfully performed. VMAX_RELEASE must be parameterized in the range 1-2147483647. Otherwise, DIAG bit No. 4 is set
V_VALID	Bool	**Actual velocity valid** is supplied from block F_SAFE_POS. 1: Velocity is plausible 0: Velocity is not plausible, i.e. the increase of the deviation between the two encoders over time is outside the tolerance window. If a 0 signal is available here, then DIAG bit No. 6 is set.
POS_VALID	Bool	**Actual position valid** is supplied from block F_SAFE_POS. 1: Position is plausible 0: Position is not plausible, i.e. the discrepancy between the two encoders is outside the tolerance window. If a 0 signal is available here, then DIAG bit No. 5 is set.
FDBACK_DRIVE	Bool	**Brake control normal operation** 0: Close brake 1: Open brake
FDBACK_BR_1	Bool	**Feedback signal brake 1** 0: open 1: closed
FDBACK_BR_2	Bool	**Feedback signal brake 2** 0: open 1: closed
S_ZSW3B	WORD	**S120 Safety Info Channel – status word 3 (r10234)** **Bit 00: SBT_SELECTED** **Feedback signal drive – select SBT** 1: SBT selected 0: function not selected **Bit 02: SBT_ACTIVE_BR** **Feedback signal drive – active brake** Here, the drive signals the number of the brake currently be tested:

Name	Data type	Description
		0: brake 1 1: brake 2
		Bit 03: SBT_ACTIVE **Feedback signal drive – status SBT** 1: test running; drive establishes a torque against the closed brake 0: test not active; drive passive
		Bit 04: SBT_RESULT **Feedback signal drive – test result** 0: Brake faulty 1: Brake successfully tested
		Bit 05: SBT_FINISHED **Feedback signal drive – status test sequence** 0: Test running 1: Test completed
		Bit 06: SBT_CLOSE_BR **Brake control SBT for external brake** The drive issues the command to open/close external brakes via this input. 0: open ext. brake 1: close ext. brake
		Bit 07: SBT_FDBACK_DIR **Feedback signal drive – direction of torque buildup** Here, the drive signals the direction of the currently established torque: 0: positive 1: negative
EXECUTE	Bool	**Start brake test** The brake test is started via a positive edge at this input. After the test has been successfully completed, the time for the test interval is restarted and output TEST_OK is set again.
ACK	Bool	**Acknowledging** If a fault occurs in normal operation, then this must be reset using ACK before the system can be restarted. Acknowledgment is only realized with a positive edge at ACK; in fault-free (normal) operation has no effect.

Outputs

Table 3-16

Name	Data type	Description
SLS_THRESHOLD	DInt	**SLS limit [mm/min]** The presently maximum permissible traversing velocity is output here. This is 2147483647 in normal operation; if the brake test has not been successfully completed, then VMAX_RELEASE is output here. If VMAX_RELEASE has been parameterized <= 0, then an equivalent value of 1 is output here.
SLS_OK	Bool	**SLS limit status** 1: SAFE_V is less than/equal to SLS_THRESHOLD 0: SAFE_V has exceeded the value of SLS_THRESHOLD. If this output changes to 0, then a stop response should be initiated.
TEST_REQ	Bool	**Request brake test**

Name	Data type	Description
		1: brake test requested 0: no brake test requested
BUSY	Bool	**Test status** 1: Test running 0: Test not selected
OPEN_BR_1	Bool	**Control signal, external brake 1** 1: Open brake 0: Close brake
OPEN_BR_2	Bool	**Control signal, external brake 2** 1: Open brake 0: Close brake
S_STW3B	Word	**S120 Safety Control Channel – control word 3 (r10235)** **Bit 00: SBT_SELECT** **Drive communication: Brake test selected** Same conditions as BUSY output **Bit 01: SBT_START** **Drive communication: Start** 1: Start test sequence **Bit 02: SBT_BR_SELECT** **Drive communication: Brake selection** 0: brake 1 1: brake 2 **Bit 03: SBT_TORQUE_DIR** **Drive communication: Torque preselection** 0: positive 1: negative **Bit 04: SBT_SEQUENCE** **Drive communication: Select test sequence** 0: sequence 1 1: sequence 2 **Bit 05: SBT_FDBACK_DIR** **Drive communication: Status ext. brake** 0: open 1: closed
TEST_OK	Bool	**Test result status** 0: Test unsuccessful 1: Test successfully completed
BR_1_OK	Bool	**Status brake 1** 0: faulty or test still not performed 1: OK
BR_2_OK	Bool	**Status brake 2** 0: faulty or test still not performed 1: OK
RELEASE_DIR	Bool	**Status motion direction** is FALSE for a test that has not been successfully completed;
ACK_REQ	Bool	**Acknowledgment request** If a fault has occurred, however it is no longer active, and can therefore be acknowledged, then the block indicates this using a 1 signal at ACK_REQ.
ERROR	Bool	**Error** This output is set if the block has been incorrectly parameterized, or if the block in operation detects a potentially dangerous combination of input signals. The output re-

Name	Data type	Description
		mains set until an error is no longer active and has been acknowledged.
DIAG	Word	**Diagnostics word** Information about the function status and errors of the block are output here.

Structure of DIAG

Table 3-17

Bit No.	Description	Reset condition
0	Runtime error T_OPEN/T_CLOSE_BR_1 not maintained	Pos. edge at ACK, restart test, reset is realized if the test was successful
1	Runtime error T_OPEN/T_CLOSE_BR_2 not maintained	Pos. edge at ACK, restart test, reset is realized if the test was successful
2	SLS monitoring triggered SAFE_V exceeds V_MAX_RELEASE or V_VALID == 0 while the axis moves for an unsuccessful test	SAFE_V <= VMAX_RELEASE and V_VALID == 1 and pos. edge at ACK
3	no standstill during the test SAFE_POS changes during the test by more than POS_TOLERANCE	Pos. edge at ACK, restart test, reset is realized if the test was successful
4	Parameterizing error, value range	T_SAMPLE >= 1 and T_OPEN_BR_1 >= 1 and T_CLOSE_BR_1 >= 1 and T_OPEN_BR_2 >= 1 and T_CLOSE_BR_2 >= 1 and POS_TOLERANCE >= 0 and V_MAX_RELEASE >= 1
5	parameterizing error integer multiple	T_OPEN_BR_1 / T_SAMPLE can be represented as integer multiple and T_CLOSE_BR_1 / T_SAMPLE can be represented as integer multiple T_OPEN_BR_2 / T_SAMPLE can be represented as integer multiple and T_CLOSE_BR_2 / T_SAMPLE can be represented as integer multiple
6	Non-plausible feedback signal from SBT:	Pos. edge at ACK, restart test, reset is realized if the test was successful
7	Drive enable missing for active brake test	Pos. edge at ACK, restart test, reset is realized if the test was successful
8	No safe position available for active brake test	Pos. edge at ACK, restart test, reset is realized if the test was successful
9	Monitoring time: no feedback signal SBT_SELECTED within the monitoring time after selecting SBT	Pos. edge at ACK, restart test, reset is realized if the test was successful
10	Monitoring time external brake request initiated by SBT	Pos. edge at ACK, restart test, reset is realized if the test was successful
11	Error for internal calculation	Pos. edge at ACK, if the fault is no longer active
12	Reserved	---

Bit No.	Description	Reset condition
13	Reserved	---
14	Warning: not a safe position, SBT not possible	POS_VALID = 1
15	Warning: not a safe velocity, SBT not possible	V_VALID = 1

3.6.3 Principle of operation

Parameterization

When parameterizing, it must be ensured that the following relationships can be represented as integer multiples:

T_OPEN_BR_1 / T_SAMPLE

T_CLOSE_BR_1 / T_SAMPLE

T_OPEN_BR_2 / T_SAMPLE

T_CLOSE_BR_2 / T_SAMPLE

The permissible value ranges of the individual inputs should be taken from the table describing the inputs.

If not all of the specified preconditions are satisfied, then the block identifies this, and signals a parameterizing error with the appropriately set DIAG bits.

NOTE

The block only checks the parameterization when it is called for the 1st time. This increases the performance for further block operation.

As a consequence, it is not possible to reparameterize the system while it is in operation. The safety program must be regenerated and loaded each time that the block operating parameters are changed.

Interface to SINAMICS S120

The interface between F_BRAKE_TEST and SINAMICS S120 is described in the following. Communication runs in the standard telegram using status/control word S_ZSW3B(Safety Info Channel status word 3)/S_STW3B (Safety Control Channel control word 3). SBT selection should be interconnected to "SBT via SCC (p10235)". The signals are interconnected to the block directly via the control/status word; the assignment can be taken from the following tables and the interface description of this block.

Communication direction, F_BRAKE_TEST -> SINAMICS S120

Table 3-18

Bit	Meaning	Remarks		Parameter
0	Select brake test	1	Brake test selected	r10235.0
		0	Brake test deselected	
1	Start brake test	1	Start brake test requested	r10235.1
		0	Start brake test not requested	
2	Brake selection	1	Test brake 2 selected	r10235.2
		0	Test brake 1 selected	
3	Select direction of rota-	1	Negative direction selected	r10235.3

Bit	Meaning	Remarks		Parameter
	tion	0	Positive direction selected	
4	Select test sequence	1	Test sequence 2 selected	r10235.4
		0	Test sequence 1 selected	
5	Status external brake	1	External brake closed	r10235.5
		0	External brake open	
6…15	Reserved	--	--	--

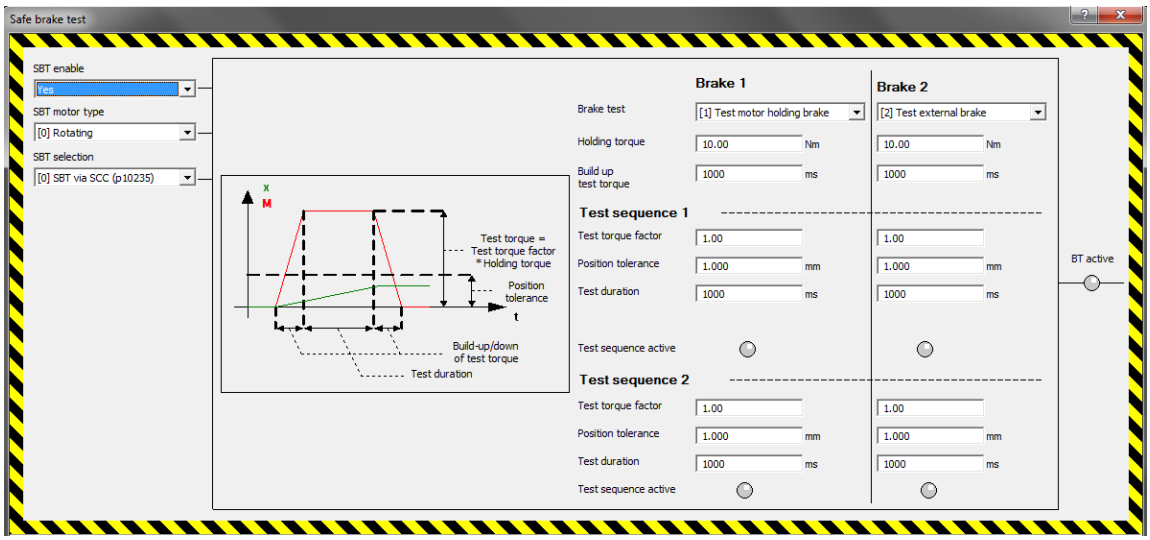
Communication direction, SINAMICS S120 -> F_BRAKE_TEST

Table 3-19

Bit	Meaning	Remarks		Parameter
0	Brake test	1	Brake test selected	r10234.0
		0	Brake test deselected	
1	Setpoint input dri-ve/external	1	Setpoint input for drive	r10234.1
		0	Setpoint input external (control)	
2	Active brake	1	Test brake 2 active	r10234.2
		0	Test brake 1 active	
3	Brake test active	1	Test active	r10234.3
		0	Test inactive	
4	Brake test result	1	Test successfully completed	r10234.4
		0	Test unsuccessful	
5	Brake test exited	1	Test performed	r10234.5
		0	Test incomplete	
6	External brake request	1	Close brake	r10234.6
		0	Open brake	
7	Actual load sign	1	Sign negative	r10234.7
		0	Sign positive	
8…13	Reserved	--	--	--
14	Acceptance test SLP(SE) deselected	1	Acceptance test SLP(SE) deselect-ed	r10234.14
		0	Acceptance test SLP(SE) deselect-ed	
15	Acceptance test mode selected	1	Acceptance test mode selected	r10234.15
		0	Acceptance test mode deselected	

Setting the safe brake test in the converter

Figure 3-10: Setting the safe brake test



The sequence in which the brakes should be tested must match the configuration in SINAMICS S120 and that at F_BRAKE_TEST. Otherwise, the block and SINAMICS S120 will output an error when performing the tests.

The parameters of the test sequences are set in SINAMICS S120; the selection as to which test sequences are to be performed and how are specified at F_BRAKE_TEST.

Test sequence and error handling

1. After the time parameterized at T_INTERVALL elapses, the block is requested to perform a brake test via output TEST_REQ. This is started using a rising edge at EXECUTE, BUSY is set to 1.
2. The test sequence for the particular brake is parameterized using input SEQUENCE_BR_1 or SEQUENCE_BR_2.
3. SEQUENCE_BR_1/2 is specified, bit coded:
   Bit 0: Test with test sequence 1 positive
4. Bit 1: Test with test sequence 1 negative
5. Bit 2: Test with test sequence 2 positive
6. Bit 3: Test with test sequence 2 negative
7. Bit 4: 0: external brake; 1: motor holding brake
8. Brake 1 is always tested first, followed by brake 2
9. The test is canceled when the first error/fault occurs ERROR changes to 1, BUSY is reset to 0.
10. At BR_1_OK or BR_2_OK, a 0 signal indicates that the test for this brake was not successful; output TEST_OK is set to 0.
11. These signals are only set to 1 after the test has been successfully completed.
12. As long as the test was not successfully completed, the velocity parameterized at input VMAX_RELEASE is output at SLS_THRESHOLD - and TEST_OK outputs a 0 signal.
13. RELEASE_DIR is set to 0 for an unsuccessful test. By appropriately controlling the SDI safety function on the drive side, it is possible to only allow traversing

in the safe direction, i.e. for a hoisting gear slowly downwards. RELEASE_DIR outputs a 1 signal again as soon as the test has been successfully completed.

14. If SAFE_V exceeds the value of SLS_THRESHOLD, then SLS_OK changes to 0 and DIAG bit 2 is set.
15. If the test for both brakes was successfully completed, then the maximum value (maximum DINT value = 2147483647) is again output at SLS_THRESHOLD.
16. DIAG and ERROR are reset to 0 using a positive edge at ACK, assuming that there are no active errors.
17. As soon as the block can be acknowledged, it indicates this using a 1 signal at output ACK_REQ. After a positive edge at ACK, ACK_REQ is reset to 0.
18. If a brake test was unsuccessful, a new brake test can only be started after acknowledgment using a positive edge at ACK. EXECUTE must again be selected to start.
19. DIAG bit 15 is set if a 0 signal is present at V_VALID.
20. Further, ERROR changes to 1 and SLS_OK to 0 if the block is presently in the retraction mode.
21. To exit this state, V_VALID must first be again set to a 1 signal using F_SAFE_POS by acknowledging.
22.
23. If the test is started using EXECUTE = 1, then the block initially signals this at the BUSY output. The SCC/SIC (Safety Control Channel/Safety Information Channel) is interconnected directly to the relevant input or output of the block as word. The internal signal processing in the block takes the appropriate bits for the brake test from the SIC (in the text, this are symbolically designated to make it easy to understand) - and processes them. The corresponding control signals for the brake test are then output, combined via word SCC. To ensure a good understanding of the situation, the internal signals from SCC or SIC words are designated with SBT_... in the following.
24. In the block, depending on the parameterized test sequence, outputs SBT_BR_SELECT, SBT_TORQUE_DIR and SBT_SEQUENCE are switched.
25. If the test has been started, then an appropriate feedback signal from the drive must be available at SBT_SELECTED. The user must also establish this signal interconnection.
26. The drive provides feedback about the brake presently being tested at input SBT_ACTIVE_BR. This feedback is used to check the plausibility. The block sets ERROR and DIAG bit 6 if the control signals contradict one another.

WARNING

For RELEASE_DIR = 0, it is not permissible that any potentially hazardous motion is executed. For hoisting gear, this would be motion upwards - for travel gear, motion in the positive or negative direction. For hoisting gear, drive function SDI is predominantly used to inhibit a specific traversing direction; for travel gear, the SOS drive function is used to inhibit any direction.

It is absolutely crucial that, depending on the application, the block output is interconnected with the correct signal to control the drive.

Otherwise, inadmissible motion is possible, which cannot be identified internally in the block.

Safety note

The parameterization of input V_MAX_RELEASE must be adapted according to the permissible safely reduced velocity, derived from the application-specific risk assessment.

Safety note

Parameter "T_INTERVAL" defines in which cyclic intervals a brake test is required. The value, which must be configured here, depends on the specific application - and is also dependent on the specific risk assessment and the actual hardware architecture of the safety function.

NOTE A brake test is requested at each stop-start transition of the CPU.

NOTE Block F_SAFE_POS provides a 0 signal at POS_VALID via output ERROR = 1. All other blocks, i.e. also F_BRAKE_TEST, indicate the status using an error code; however, in order to avoid being confronted by a flood of messages, in this case ERROR is not again set to a 1 signal, assuming that the block is not performing a brake test at this instant in time. ERROR is also set to 1 if a brake test is active, and a 0 signal is available at input POS_VALID.

Testing an external brake

If a 0 signal is available at SEQUENCE_BR_1/2.BIT4, then an external brake is tested according to the following schematic:

27. If a 1 signal is available at SBT_CLOSE_BR, depending on the state of SBT_ACTIVE_BR, the block either switches OPEN_BR_1 or OPEN_BR_2 inactive, i.e. the brake presently being tested is closed. A 1 signal must be available at feedback channel FDBACK_BR_1/ FDBACK_BR_2 within the time parameterized at T_CLOSE_BR_1/T_CLOSE_BR_2.
28. If this is not the case, then the test is canceled as described above. ERROR and DIAG bit 0/1 (depending on the brake presently being tested) change to 1.
29. After T_CLOSE_BR_1/T_CLOSE_BR_2 expires - and if there is a 1 signal at FDBACK_BR_1/ FDBACK_BR_2 - the drive is signaled that the brake is closed via SBT_FDBACK_BR = 1; the drive then executes the test profile.
30. During the test the system monitors as to whether the value at input SAFE_POS changes by more than the value parameterized at POS_TOLERANCE. The test is canceled as described above if this change is higher. ERROR and DIAG bit 3 change to a 1 signal.
31. Once the drive completes the test, the command to open the brake is given at the block input via SBT_CLOSE_BR using a 0 signal.
32. A 1 signal is again available at output OPEN_BR_1/ OPEN_BR_2.
33. A 0 signal must be available at feedback channel FDBACK_BR_1/ FDBACK_BR_2 after the time parameterized at input T_OPEN_BR_1/ T_OPEN_BR_2.
34. If this is not the case, then the test is canceled as described above. ERROR and DIAG bit 0/1 (depending on the brake presently being tested) change to 1.

35. After T_OPEN_BR_1/ T_OPEN_BR_2 expires - and if there is a 0 signal at FDBACK_BR_1/ FDBACK_BR_2 - then drive is signaled that the brake is open via SBT_FDBACK_BR = 0.
36. If the brake was successfully tested, then the drive signals this using SBT_FINISHED = 1.
37. When a test has been successfully completed, a 1 signal is available at SBT_RESULT.
38. If necessary, this test pattern is repeated for the second brake - or depending on SEQUENCE_BR_2.BIT4 for the second brake - the following test pattern is applied:

Testing a motor holding brake

If a 1 signal is available at SEQUENCE_BR_1/2.BIT4, then a motor holding brake is tested according to the following procedure at the drive:

39. In this operating mode, the drive directly controls the brake. This means that the drive independently executes its test profile; the block ignores SBT_CLOSE_BR.
40. During the test the system monitors as to whether the value at input SAFE_POS changes by a maximum of POS_TOLERANCE. The test is canceled as described above if this change is higher. If the brake was successfully tested, then the drive signals this using SBT_FINISHED = 1.
41. When a test has been successfully completed, a 1 signal is available at SBT_RESULT.

Test completed

42. If the test is still running for brake 2, however, the configured sequences for brake 1 have been executed already without any errors, then BR_1_OK has a 1 signal, while BR_2_OK and TEST_OK still have a 0 signal.
43. If the test was successfully completed for all of the configured test sequences, then this is signaled at output TEST_OK with a 1 signal, BR_2_OK then also has a 1 signal.
44. The time monitoring for when the next test is due (T_INTERVAL) is restarted, and then the block sets output BUSY back to 0.

Acknowledging errors

45. DIAG and ERROR are reset to 0 using a positive edge at ACK, assuming that there are no active errors.
46. As soon as the block can be acknowledged, it indicates this using a 1 signal at output ACK_REQ. After a positive edge at ACK, ACK_REQ is reset to 0.
47. If a brake test was unsuccessful, then this must first be acknowledged using a positive edge at ACK - before a new test can be started using EXECUTE.

3.6.4 Application example for safely controlling external brakes

In the following function example, the external brakes at F-DO channel A20.0 ("BRAKE1") and A20.1 ("BRAKE2") are to be controlled from F_BRAKE_TEST for the brake test - and safety function STO triggered.

The signal for STO (designated here as “STO_select") is low active; i.e. 1 means that STO is not active; 0 means that at least one safety function requests an STO.

The brake feedback signals are wired to the two standard inputs E1.0 ("FDBACK_BRAKE1") and E1.1 ("FDBACK_BRAKE2"); a 1 signal means that the brake is closed, while a 0 signal means that the particular brake is open.

In this particular example, 100ms is used as the monitoring time for opening and closing the brakes. This time also depends on the response time for your safety functions specified from the risk assessment.

WARNING

Safety note
The parameterization of inputs T_OPEN_BR_x and T_CLOSE_BR_x - as well as FDB_TIME - used in this example - must be adapted to address the required response time of the safety functions for the specific application.

It is not permissible that the monitoring time exceeds the required response time.

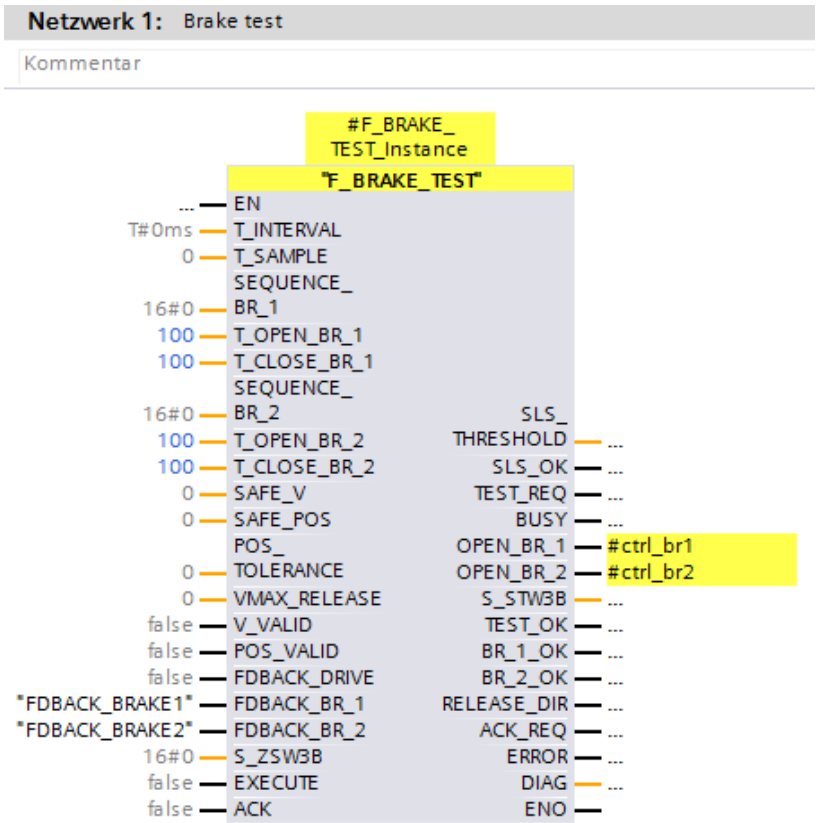
In the following code example, only the relevant interconnections have been made for the above description of the application for reasons of transparency.

In order that a program is created that can actually run, block F_BRAKE_TEST must be parameterized according to the description in Chapter 3.6.3.

The example is subdivided into three networks.

Block F_BRAKE_TEST is called in the first network. This does not directly control the brakes, but transfers the control commands via temporary variables #ctrl_br1 and #ctrl_br to networks 2 and 3.

Figure 3-11

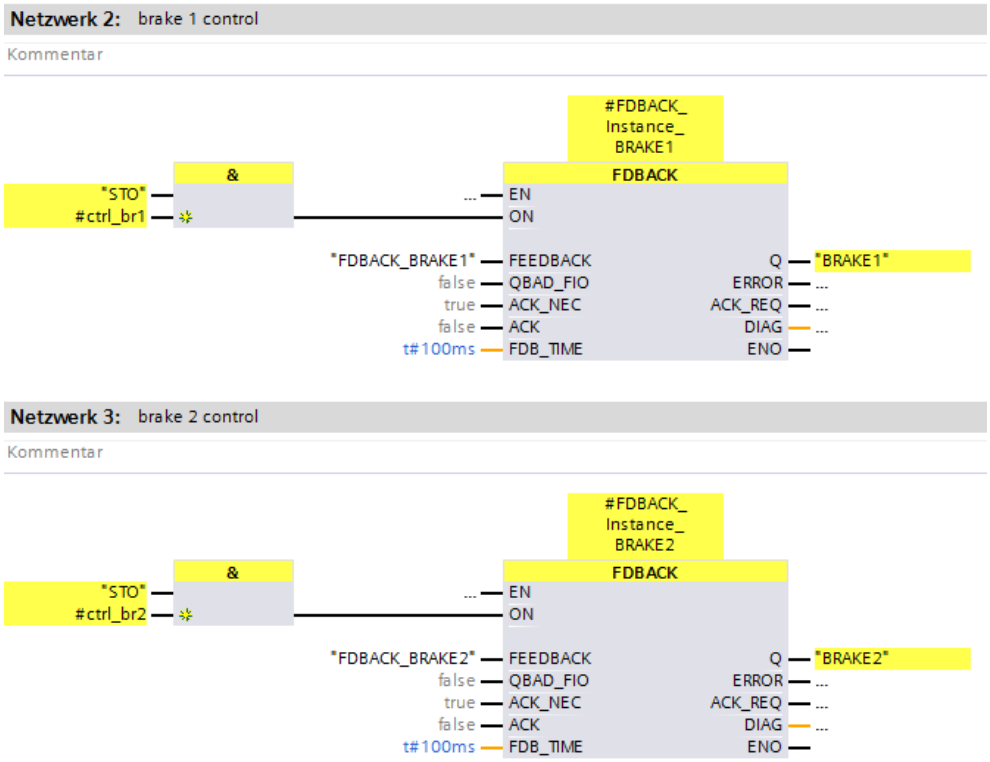


In the 2nd and 3rd networks these signals, together with signal “STO_select", are connected to a F_FDBACK function block. Block F_FDBACK is included in the STEP7 Safety Advanced library under number FB216, and is used to monitor the feedback circuit.

You can find additional information on this block in the online help in the TIA Portal.

Assuming that there is no feedback circuit fault for the brake, and the logic operation at the ON input of the F_FDBACK has a 1 signal, the brakes at output A20.0 ("BRAKE1") and A20.1 ("BRAKE2") are opened.

Figure 3-12



3.7 F_LOAD_MONITOR

3.7.1 Introduction

The fail-safe function block F_LOAD_MONITOR has the function to guarantee safety-related overload and slack cable detection.

The block allows various versions of the actual load value to be read in, e.g.:

1. Qualified (safe) measurement source - as well as safe evaluation (e.g. F-AI module)
2. Two diverse (non-safety-related) measurement sources (e.g. motor torque via SINAMICS and weighing cell via AI module) - the plausibility of the encoder values is checked using this block.

NOTE **Safety note**

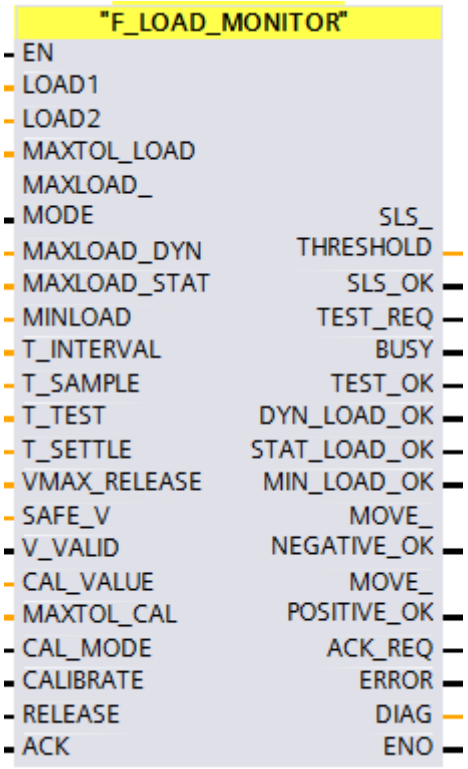
The encoder and evaluation units used must be evaluated in accordance with the specific application.

A retraction logic becomes available if slack cable or overload is detected during operation. When a slack cable is detected, retraction is monitored so that upward retraction is only permissible with reduced velocity. For overload, retraction is only permissible downward.

The block provides the option of making a distinction between static and dynamic loads, for example that can occur when quickly lifting loads.

To check the correct functioning of the measurement equipment, after a parameterizable interval expires, the system requests that the block is calibrated.

Figure 3-13



NOTE When using this block, block **F_BO_W (FC 176)** and block **F_TP (FB 184)** must be available in the block folder. It is not permissible to renumber these! Blocks **LFAddDInt (FC 211)**, **LFSubDInt (FC 212)**, **LFMulDInt (FC 213)** and **LFDivDInt (FC 214)** are also required from this library. These may be renumbered, but not renamed.

3.7.2 Connections

All bool type variables listed in the following table are preassigned FALSE, all integer variables with 0, all TIME variables with T#0ms and all word variables with W#16#0.

Inputs

Table 3-20

Name	Data type	Description
LOAD1	DInt	**Load channel 1 [%]** 10000 = 100.00%
LOAD2	DInt	**Load channel 2 [%]** 10000 = 100.00%
MAXTOL_LOAD	DInt	**Tolerance window load monitoring [%]** 10000 = 100.00%
MAXLOAD_MODE	Bool	**Monitoring mode** 0: Monitoring for static load 1: Monitoring for dynamic load
MAXLOAD_DYN	DInt	**Max. dynamic load [%]** 10000 = 100.00%
MAXLOAD_STAT	DInt	**Max. static load [%]** 10000 = 100.00%
MINLOAD	DInt	**Min. load** 10000 = 100.00%
T_INTERVALL	Time	**Test interval** After this time has elapsed, the block requests that the measurement equipment is tested. This is signaled at output TEST_REQ using a 1 signal.
T_SAMPLE	DInt	**Sampling time [ms]** Here, the block sampling time, i.e. the interval in which the safety program is called (cyclic interrupt OB interval of the F-OB) is parameterized in ms.
T_TEST	DInt	**Test duration [ms]**
T_SETTLE	DInt	**Settling time [ms]**
VMAX_RELEASE	DInt	**Retraction velocity [mm/min]** For an overload/underload condition, this value is output at SLS_THRESHOLD
SAFE_V	DInt	**Safe velocity actual value [mm/min]** is supplied from block F_SAFE_POS.
V_VALID	Bool	**Actual velocity valid** is supplied from block F_SAFE_POS. 1: Velocity is plausible 0: Velocity is not plausible, i.e. the increase of the deviation between the two encoders over time is outside the tolerance window.
CAL_VALUE	DInt	**Calibration value (%)** 10000 = 100.00% Relevant for testing the measurement equipment

Name	Data type	Description
MAXTOL_CAL	DInt	**Calibration tolerance [%]** 10000 = 100.00%
CAL_MODE	Bool	**Test mode** 0: Test with constant load value 1: Test with defined load step
CALIBRATE	Bool	**Starting the measurement equipment test** The calibration test is started using a positive edge at this input. After the test has been successfully completed, the time for the test interval is restarted.
RELEASE	Bool	**Retracting** If the block detects an overload/underload condition, by selecting this input, the axis can be moved with the velocity parameterized at VMAX_RELEASE in the direction enabled by the block using MOVE_POSITIVE_OK/MOVE_NEGATIVE_OK. Retraction motion is immediately stopped as soon as this input has a 0 signal.
ACK	Bool	**Acknowledging** If a fault occurs in normal operation, then this must be reset using ACK before the system can be restarted. Acknowledgment is only realized with a positive edge at ACK; in fault-free (normal) operation has no effect.

Outputs

Table 3-21

Name	Data type	Description
SLS_THRESHOLD	DInt	**SLS limit [mm/min]** The presently maximum permissible traversing velocity is output here. This is 2147483647 in normal operation; when overload/underload is detected, VMAX_RELEASE is output here. If VMAX_RELEASE has been parameterized <= 0, then an equivalent value of 1 is output here.
SLS_OK	Bool	**SLS limit status** 1: SAFE_V is less than/equal to SLS_THRESHOLD 0: SAFE_V has exceeded the value of SLS_THRESHOLD. If this output changes to 0, then a stop response should be initiated.
TEST_REQ	Bool	**Request to test the measuring equipment** 1: T_INTERVAL expired 0: Test not necessary
BUSY	Bool	**Test status** 1: Test running 0: Test not selected
TEST_OK	Bool	**Test result status** 0: Test faulty or test still not performed 1: Test successfully completed
DYN_LOAD_OK	Bool	**Status dyn overload** 0: Overload detected 1: Load OK
STAT_LOAD_OK	Bool	**Status stat. overload** 0: Overload detected 1: Load OK

Name	Data type	Description
MIN_LOAD_OK	Bool	**Underload status** 0: Slack cable detected 1: Load OK
MOVE_NEGATIVE_OK	Bool	**Negative motion permitted** If a 0 signal is available at this output, then it is not permissible that the machine continues to move in the negative direction. The output is set to 0 as soon as the block detects a slack cable condition.
MOVE_POSITIVE_OK	Bool	**Positive motion permitted** If a 0 signal is available at this output, then it is not permissible that the machine continues to move in the positive direction. The output is set to 0 as soon as the block detects a slack cable condition.
ACK_REQ	Bool	**Acknowledgment request** If a fault has occurred, however it is no longer active, and can therefore be acknowledged, then the block indicates this using a 1 signal at ACK_REQ.
ERROR	Bool	**Error** This output is set if the block has been incorrectly parameterized, or if the block in operation detects a potentially dangerous combination of input signals. The output remains set until an error is no longer active and has been acknowledged.
DIAG	Word	**Diagnostics word** Information about the function status and errors of the block are output here.

Structure of DIAG

Table 3-22

Bit No.	Description	Reset condition
0	Discrepancy error load monitoring	LOAD1 and LOAD2 within MAXTOL_LOAD and positive edge at ACK
1	Overload detected	LOAD1 and LOAD2 less than MAXLOAD_STAT or MAXLOAD_DYN (depending on MAXLOAD_MODE) – MAXTOL_LOAD and positive edge at ACK
2	Slack cable detected	LOAD1 and LOAD2 greater than MINLOAD + MAXTOL_LOAD and positive edge at ACK
3	Parameterizing error, load limits	MINLOAD < MAXLOAD_STAT <= MAXLOAD_DYN
4	Settling process, calibration inadmissibly long	Restart test
5	Inadmissibly high load fluctuation during calibration	Restart test
6	Parameter error test times	T_TEST > T_SETTLE > 0 and both times integer multiple of T_SAMPLE
7	Retraction velocity exceeded	SAFE_V <= SLS_THRESHOLD and positive edge at ACK
8	Parameterizing error, value range	0 < VMAX_RELEASE <= 2147483647 and 0<= MAXLOAD_DYN / MAXLOAD_STAT / MINLOAD / CAL_VALUE / MAXTOL_LOAD / MAXTOL_CAL <= 10000 parameterized

Bit No.	Description	Reset condition
9	Actual velocity invalid	Active velocity again valid and positive edge at ACK
10	Invalid value range input variables	LOAD1, LOAD2 in the range 0 to 10000 and positive edge at ACK
11	Error for internal calculation	Pos. edge at ACK, if the fault is no longer active
12	Reserved	---
13	Reserved	---
14	Reserved	---
15	Reserved	---

3.7.3 Scaling the input variables

The block expects that load limits or load actual values are entered as percentage to two decimal places, i.e. a value of 10000 corresponds to 100%. The user must carry out the scaling corresponding to the reference variable of the module used.

For example, for F-AI modules the reference variable is 27648. STEP7 Safety Advanced provides block "F_SCALE" specifically for this purpose.

If hardware is used with other reference variables, then the user is responsible for programming the scaling himself.

Safety note

Users must correctly calculate the load limit values in compliance with the requirements laid down in EN528. Users must appropriately interconnect the calculated limit values at the block.

3.7.4 Principle of operation

Parameterization

When parameterizing, it must be ensured that the following relationships can be represented as integer multiples:

T_TEST / T_SAMPLE

T_SETTLE / T_SAMPLE

Further, the following relationship between the input variables must apply:

MINLOAD < MAXLOAD_STAT <= MAXLOAD_DYN

T_TEST > T_SETTLE > 0

The permissible value ranges of the individual inputs should be taken from the table describing the inputs.

If not all of the specified preconditions are satisfied, then the block identifies this, and signals a parameterizing error with the appropriately set DIAG bits.

NOTE

The block only checks the parameterization when it is called for the 1st time. This increases the performance for further block operation.

As a consequence, it is not possible to reparameterize the system while it is in operation. The safety program must be regenerated and loaded each time that the block operating parameters are changed.

Load monitoring

1. If two independent sources are used for measuring the force, then after scaling these should be interconnected to inputs LOAD1 or LOAD2. If one measurement source is sufficient, then this is interconnected to both inputs.
2. If the difference between the two inputs is greater than the value parameterized at MAXTOL_LOAD, then ERROR = 1 and an error code is output at DIAG.
3. In addition, the velocity parameterized at VMAX_RELEASE is output at SLS_THRESHOLD.
4. If both values again lie within the parameterizable window using MAXTOL_LOAD, then output ERROR and DIAG can be set to 0 with a positive edge at ACK.
5. Using input MAXLOAD_MODE, a distinction can be made between monitoring for static overload (MAXLOAD_MODE = 0) and dynamic overload (MAXLOAD_MODE = 1).
6. For MAXLOAD_MODE =0, as soon as the value at LOAD1 or LOAD2 exceeds the value parameterized at MAXLOAD_STAT, this error is signaled to STAT_LOAD_OK using a 0 signal.
7. In addition, ERROR is set to 1 and an error code is output at DIAG.
8. For MAXLOAD_MODE =1, as soon as the value at LOAD1 or LOAD2 exceeds the value parameterized at MAXLOAD_DYN, this error is signaled to STAT_LOAD_OK using a 0 signal.
9. In addition, ERROR is set to 1 and an error code is output at DIAG.
10. As long as one of these errors is active, then the velocity parameterized at VMAX_RELEASE is output at SLS_THRESHOLD.
11. The response when MINLOAD is fallen below is essentially the same.

Retracting

12. The retraction function of the block can be activated using a 1 signal at input RELEASE. Further traversing in the positive direction is no longer permissible; the block signals this with a 0 signal at MOVE_POSITIVE_OK. Through an appropriate interconnection with the drive, users must ensure that in this case retraction is only possible downwards.
13. To facilitate retraction, DYN_LOAD_OK or STAT_LOAD_OK is reset to 1 with a rising edge at RELEASE; the drive stop response should be selected using a suitable user interconnection.
14. If, during retraction, the value of SAFE_V exceeds the value of SLS_THRESHOLD, then SLS_OK changes to 0.
15. If, in both cases LOAD1 and LOAD2 are again less than the active limit - MAXTOL_LOAD, then ERROR and DIAG bit 1 can be reset to 0 with a positive edge at ACK.
16. The maximum velocity is again output at SLS_THRESHOLD. (Maximum DINT value = 2147483647)

17. VMAX_RELEASE must lie in the range 1 – 2147483647; if values less than 1 are parameterized, the block detects this and signals it with DIAG bit No. 8. ERROR changes to 1. 1 is then output for the retraction velocity as equivalent value.
18. If, during retraction V_VALID = 0, then the retraction velocity can no longer be monitored in a safely-related way. Therefore, selection using RELEASE = 1 has no effect, and retraction motion is stopped. DIAG bit 9 and ERROR change to 1, SLS_OK has a 0 signal.
19. To exit this state, V_VALID must first be again set to a 1 signal using F_SAFE_POS by acknowledging.

WARNING

Safety note

The signal for RELEASE must be generated in a safety-related fashion, e.g. by using a key-operated switch or similar device.

WARNING

Safety note

The parameterization of input V_MAX_RELEASE must be adapted according to the permissible safely reduced velocity, derived from the application-specific risk assessment.

WARNING

The interconnection of output MOVE_POSITIVE_OK must match the selection of drive function SDI for a positive direction. For MOVE_POSITIVE_OK = 0, motion in the positive direction must no longer be possible.
The same is true when interconnecting output MOVE_NEGATIVE_OK and inhibiting the negative direction of motion.
It is absolutely crucial that the block outputs are interconnected with the correct signals to control the drive.

Otherwise, inadmissible motion towards the end stops is possible, which cannot be identified internally in the block.

Testing the measuring equipment

20. After the time that can be parameterized at T_INTERVALL has elapsed, the force sensor must be tested; the block flags this using a 1 signal at TEST_REQ.
21. The test is started using a positive edge at CALIBRATE, and output BUSY changes to 1.
22. Depending on input CAL_MODE, as test variable, a constant load or a defined load step is expected.

Safety note

Parameter "T_INTERVAL" defines in which cyclic intervals the measuring equipment must be tested. The value, which must be configured here, depends on the specific application - and is also dependent on the specific risk assessment and the actual hardware architecture of the safety function.

Case a): Test with constant load

23. If a 0 signal is available at input CAL_MODE, then within T_SETTLE, the measured load at LOAD1 and LOAD2 must assume the calibration value that can be parameterized at CAL_VALUE - taking into account the tolerance parameterized at MAXTOL_CAL.
24. If this is not the case, then ERROR changes to a 1 signal and at DIAG bit 4 is set.
25. The measured load value at LOAD1 and LOAD2 must not deviate from CAL_VALUE by more than MAXTOL_CAL for the time parameterized at T_TEST.
26. If this is not the case, then ERROR changes to a 1 signal and at DIAG bit 5 is set.
27. After T_TEST expires - and there is a valid load value - BUSY is reset to 0, and output TEST_OK is set to a 1 signal.
28. If T_TEST is parameterized to be <= T_SETTLE, then DIAG bit 5 and ERROR are set to 1.

Case b): Test with defined load step

29. The load value must execute a defined load stroke if a 1 signal is available at input CAL_MODE. In so doing, the signal at LOAD1 and LOAD2 must assume the expected stroke of CAL_VALUE within T_SETTLE.
30. If this is not the case, then ERROR changes to a 1 signal and at DIAG bit 4 is set.
31. For the duration of T_TEST, the measured range (stroke) must not deviate by more than MAXTOL_CAL from the expected range that can be parameterized at CAL_VALUE.
32. If this is not the case, then ERROR changes to a 1 signal and at DIAG bit 5 is set to 1.
33. If, after T_SETTLE expires, the signal level measured at LOAD1 and LOAD2 is not higher than the initial value (before the test stroke was started) by the value CAL_VALUE (taking into account MAXTOL_CAL), then ERROR is set to 1 and DIAG bit 4 is set.
34. After T_TEST expires - and there is a valid value for the load step - BUSY is reset to 0, and output TEST_OK is set to a 1 signal.
35. If T_TEST is parameterized to be <= T_SETTLE, then at DIAG bit 6 and ERROR are set to 1.
36. A successful test is signaled at block output TEST_OK using a 1 signal. BUSY is reset to 0. TEST_OK remains set to 1 until the next time that TEST_REQ changes to 1, or a new test is started.

Acknowledging errors:

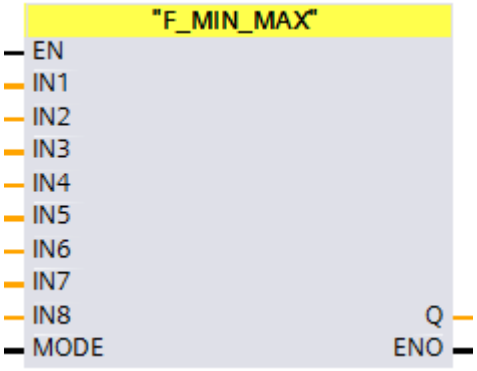
37. DIAG and ERROR are reset to 0 using a positive edge at ACK, assuming that there are no active errors.
38. As soon as the block can be acknowledged, it indicates this using a 1 signal at output ACK_REQ.
39. After a positive edge at ACK, ACK_REQ is reset to 0.

3.8 F_MIN_MAX

3.8.1 Introduction

The fail-safe F_MIN_MAX function executes a minimum/maximum value evaluation from up to 8 DINT values. The function can be used, for example, to select the presently most restrictive active SLS limit.

Figure 3-14



3.8.2 Connections

Inputs

Table 3-23

Name	Data type	Description
IN1	DInt	Operand 1 for evaluation
IN2	DInt	Operand 2 for evaluation
IN3	DInt	Operand 3 for evaluation
IN4	DInt	Operand 4 for evaluation
IN5	DInt	Operand 5 for evaluation
IN6	DInt	Operand 6 for evaluation
IN7	DInt	Operand 7 for evaluation
IN8	DInt	Operand 8 for evaluation
MODE	Bool	Selects minimum/maximum evaluation 0: Minimum evaluation 1: Maximum evaluation

Outputs

Table 3-24

Name	Data type	Description
Q	DInt	Depending on the particular MODE, minimum or maximum value of the 8 inputs

3.8.3 Principle of operation

Parameterization

1. The block is realized as function. This means that when called, all inputs must be interconnected. If a minimum/maximum value evaluation is to be carried out for less than 8 signals, then the signal sources should be interconnected a multiple number of times so that all of the inputs are occupied at the block.

Evaluating the minimum/maximum value

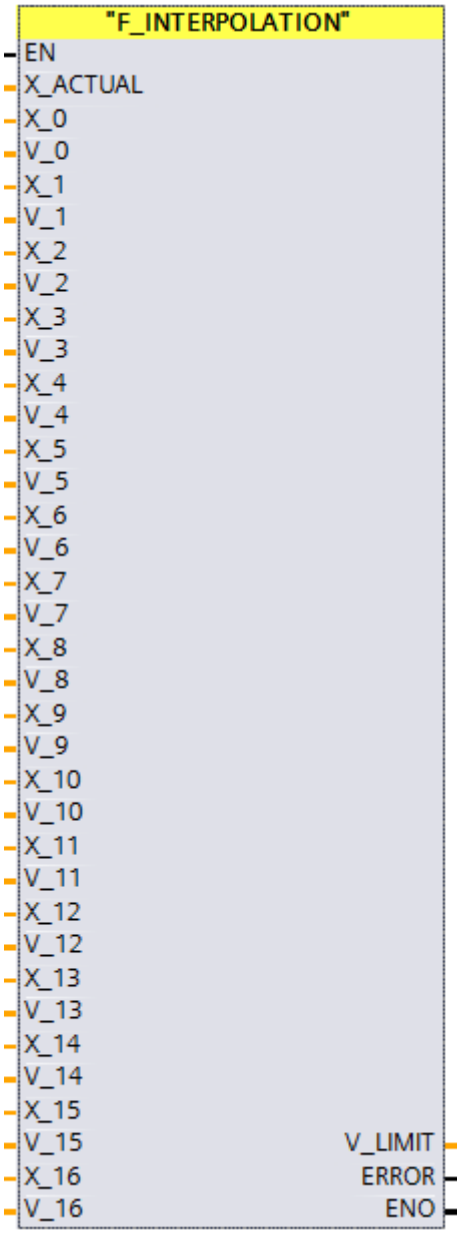
2. If, at input MODE, there is a 1 signal, then the block carries out a maximum evaluation from the 8 inputs IN1-IN8. The highest of these up to 8 DINT values are made available at output Q.
3. If, at input MODE, there is a 0 signal, then the block carries out a minimum evaluation; this means the lowest of these 8 DINT values is output at Q.

3.9 F_INTERPOLATION

3.9.1 Introduction

The fail-safe F_INTERPOLATION function implements a linear interpolation across 17 interpolation points. It is used as subordinate function of block F_ENDZONE (Chapter3.4) to map the envelope curve.

Figure 3-15



NOTE When using this block, blocks **LFAddDInt (FC 211)**, **LFSubDInt (FC 212)**, **LFMulDInt (FC 213)** and **LFDivDInt (FC 214)** are also required from this library. These may be renumbered, but not renamed.

3.9.2 Connections

Inputs

Table 3-25

Name	Data type	Description
X_ACTUAL	DInt	Actual position [mm]
X_0	DInt	Position 0 [mm]
V_0	DInt	Velocity at position 0 [mm/min]
X_1	DInt	Position 1 [mm]
V_1	DInt	Velocity at position 1 [mm/min]
X_2	DInt	Position 2 [mm]
V_2	DInt	Velocity at position 2 [mm/min]
X_3	DInt	Position 3 [mm]
V_3	DInt	Velocity at position 3 [mm/min]
X_4	DInt	Position 4 [mm]
V_4	DInt	Velocity at position 4 [mm/min]
X_5	DInt	Position 5 [mm]
V_5	DInt	Velocity at position 5 [mm/min]
X_6	DInt	Position 6 [mm]
V_6	DInt	Velocity at position 6 [mm/min]
X_7	DInt	Position 7 [mm]
V_7	DInt	Velocity at position 7 [mm/min]
X_8	DInt	Position 8 [mm]
V_8	DInt	Velocity at position 8 [mm/min]
X_9	DInt	Position 9 [mm]
V_9	DInt	Velocity at position 9 [mm/min]
X_10	DInt	Position 10 [mm]
V_10	DInt	Velocity at position 10 [mm/min]
X_11	DInt	Position 11 [mm]
V_11	DInt	Velocity at position 11 [mm/min]
X_12	DInt	Position 12 [mm]
V_12	DInt	Velocity at position 12 [mm/min]
X_13	DInt	Position 13 [mm]
V_13	DInt	Velocity at position 13 [mm/min]
X_14	DInt	Position 14 [mm]
V_14	DInt	Velocity at position 14 [mm/min]
X_15	DInt	Position 15 [mm]
V_15	DInt	Velocity at position 15 [mm/min]

Outputs

Table 3-26

Name	Data type	Description
V_LIMIT	DInt	Actual velocity limit [mm/min]
ERROR	Bool	Fault is active

3.9.3 Principle of operation

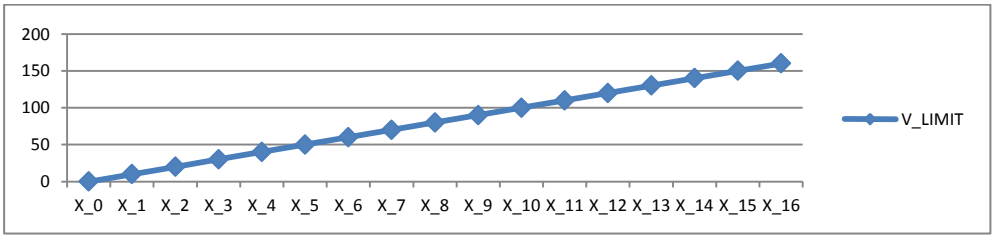
Parameterization

1. To start, the block checks the parameterization by checking that the X values of the interpolation points are valid. The X value of the subsequent interpolation point may not be less than the X value of the previous interpolation point.

Interpolation

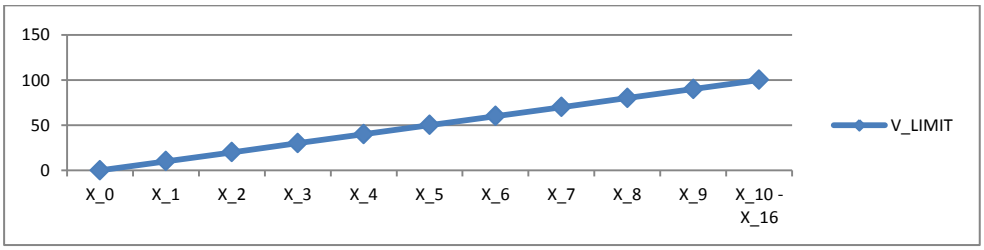
2. If X_ACTUAL is identical to an X value of an interpolation point, then the associated V value of the interpolation point is directly output at V_LIMIT.
3. If X_ACTUAL lies between two interpolation points, then the value is calculated using linear interpolation and output at V_LIMIT. This is shown in the following diagram.
4.

Figure 3-16



5. Is less than 17 interpolation points are required, all interpolation points above the last required point must be parameterized with its value. 11 interpolation points are required in the following diagram. As a consequence, interpolation points X_10/V_10 – X_16/V_16 have the same value.
6.

Figure 3-17



7. If an error occurs when parameterizing - or for an internal calculation, then this is output using ERROR = 1; the user program must then appropriately respond to this.

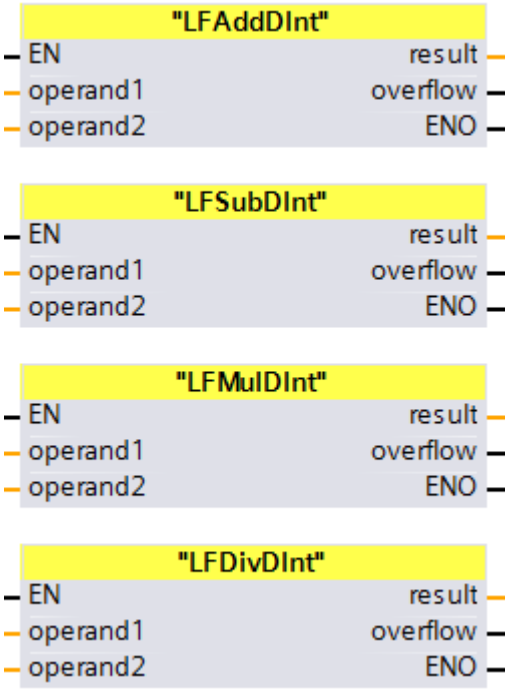
3.10 LFAddDInt/LFSubDInt/LFMulDInt/LFDivDInt

3.10.1 Introduction

For addition, subtraction, multiplication and division mathematical operations, the four fail-safe blocks check whether the result has exceeded the permissible value range of data type double integer (DINT). The functions are used within the blocks described in this library to execute and check mathematical calculations.

8.

Figure 3-18



3.10.2 Connections

Inputs

Table 3-27

Name	Data type	Description
operand1	DInt	Operand 1 for evaluation
operand2	DInt	Operand 2 for evaluation

Outputs

Table 3-28

Name	Data type	Description
result	DInt	Result ("0" for overflow)
overflow	Bool	1: Overflow 0: No overflow

3.10.3 Principle of operation

LFAddDInt

1. The block adds operand1 and operand2. If the result lies in the valid value range, the result is output at result - and output overflow is set to false.
2. If the valid value range after the mathematical operation is violated, at output result, a 0 is output and output overflow is set to true.
3.
4. result := operand1 + operand2

LFSubDInt

5. The block subtracts operand2 from operand1. If the result lies in the valid value range, the result is output at result - and output overflow is set to false.
6. If the valid value range after the mathematical operation is violated, at output result, a 0 is output and output overflow is set to true.
7.
8. result = operand1 - operand2

LFMulDInt

9. The block multiplies operand1 with operand2. If the result lies in the valid value range, the result is output at result - and output overflow is set to false.
10. If the valid value range after the mathematical operation is violated, at output result, a 0 is output and output overflow is set to true.
11.
12. result = operand1 * operand2

LFDivDInt

13. The block divides operand1 by operand2. If the result lies in the valid value range, the result is output at result - and output overflow is set to false.
14. If the valid value range after the mathematical operation is violated, at output result, a 0 is output and output overflow is set to true.
15.
16. result = operand1 / operand2

4 Interaction between the blocks

4.1 Overview

This chapter explains significant points, which must be taken into account when using fail-safe function blocks for storage and retrieval machines. The necessary block interconnection options are also shown as example.

The block package has a modular structure, and can be individually configured to address the particular application.

The blocks execute an autonomous subfunction. Depending on the specific machine in which they used, not all of the blocks are always required from the library.

If additional functions are required to specifically control an application, then users must realize this by adding additional fail-safe functions themselves. The signals of these functions are then interconnected with the RBG blocks.

Safety note

The safety-related times must be parameterized and the inputs and outputs interconnected corresponding to all of the directives applicable for the specific system. Further, they must be carefully checked at the system to ensure that they are in full compliance with the specific requirements.

4.2 Signal flow between components

The signal flow between the block interfaces, which can directly interact with one another, is shown in the following overview. For reasons of transparency, the additional inputs that are not connected are parameterized according to the above description, but are not interconnected in the following overview. This is because they do not exchange any information or data between the blocks, but are individually parameterized for each block.

4.2.1 Automation task

The signal flow between the block interfaces, which can directly interact with one another, is shown in the following overview. For reasons of transparency, the additional inputs that are not connected are parameterized according to the above specification, but are not interconnected in the following overview. This is because they do not exchange any information or data between the blocks, but are individually parameterized for each block.

In the following overview, the block interconnection monitors a hoisting gear so that it only traverses within a defined range. To achieve this, either block F_SLP_MONITOR or block F_ENDZONE is used. When F_ENDZONE is used, there is also the option that the machine may only approach the end zones with a reduced velocity.

The safe position and velocity required for the blocks mentioned above are supplied from block F_SAFE_POS.

Further, the hoisting gear is monitored for overload and slack cable using F_LOAD_MONITOR.

A block F_BRAKE_TEST is responsible for testing the correct functioning of the hoisting gear brakes.

If F_LOAD_MONITOR, F_BRAKE_TEST or F_SLP_MONITOR/F_ENDZONE identifies that a limit value has been violated, then the SLS threshold is also set to the value parameterized at the block.

Block F_SBR_MONITOR monitors as to whether the drive brakes along the configured down ramp after it selects an SS1. If this is not the case, then the signal to initiate STO is generated.

By ANDing all of the relevant enable signals of the blocks, the signal to initiate a stop response (e.g. SS1) can be formed for the drive.

For the retraction function of blocks F_ENDZONE, F_LOAD_MONITOR and F_SLP_MONITOR, using an AND logic operation of the corresponding MOVE_POSITIVE_OK/MOVE_NEGATIVE_OK signals, only that direction is permitted that allows the storage and retrieval machine to move away from the end zone.

If the brake test was unsuccessful, using block F_BRAKE_TEST, via output RELEASE_DIR, the hoisting gear can be prevented from moving upwards.

4.3 Response in the case of an error

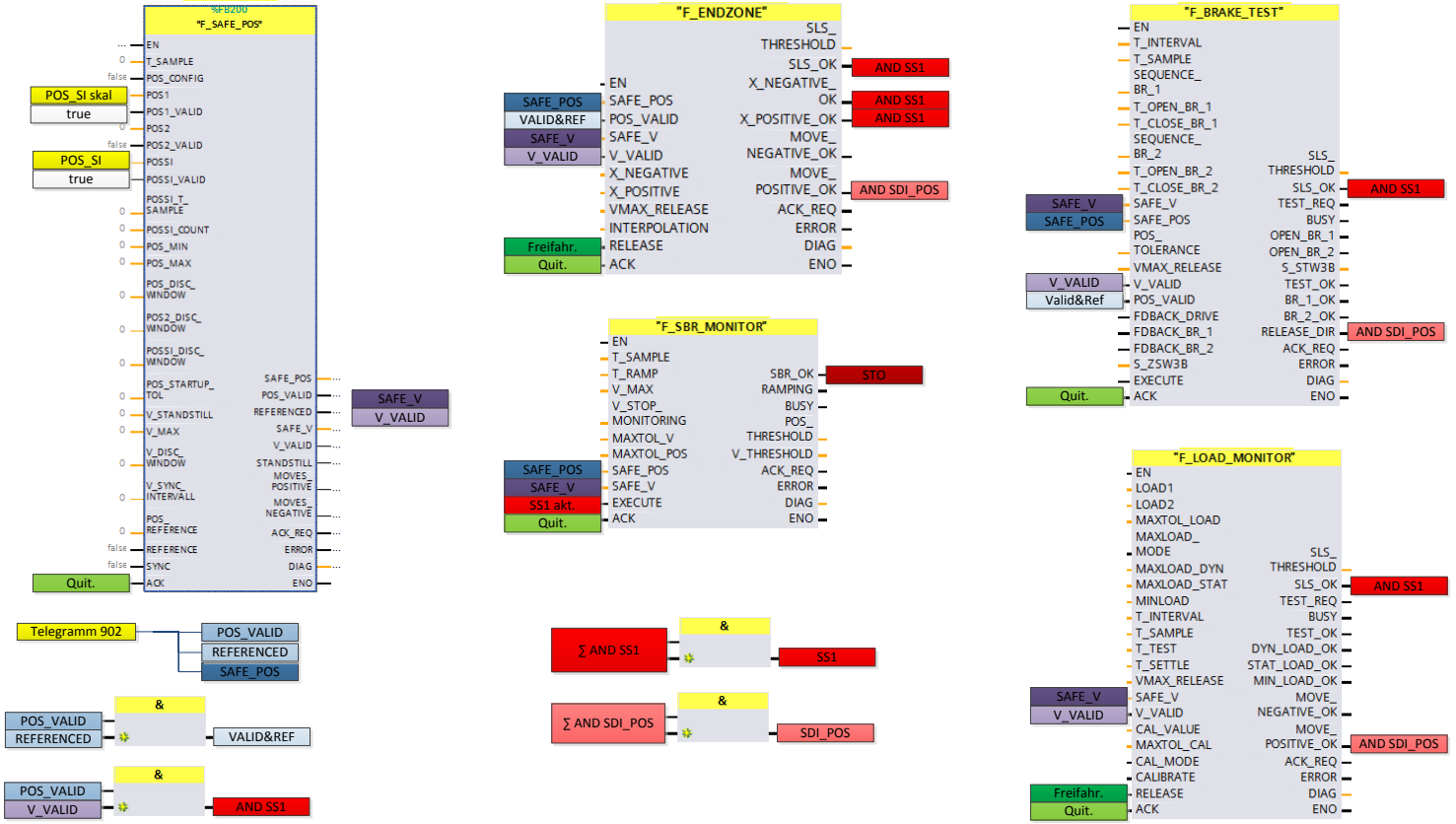
If, when parameterizing the block, or as a result of an invalid input assignment, as a result of the process, a error occurs at a block, then each block in the library, with the exception of F_MIN_MAX, signals this using output ERROR = 1.

With the exception of F_MIN_MAX, the blocks in the library have a DIAG output; this allows more precise diagnostics based on the error code that is output.

4.4 Block interconnection

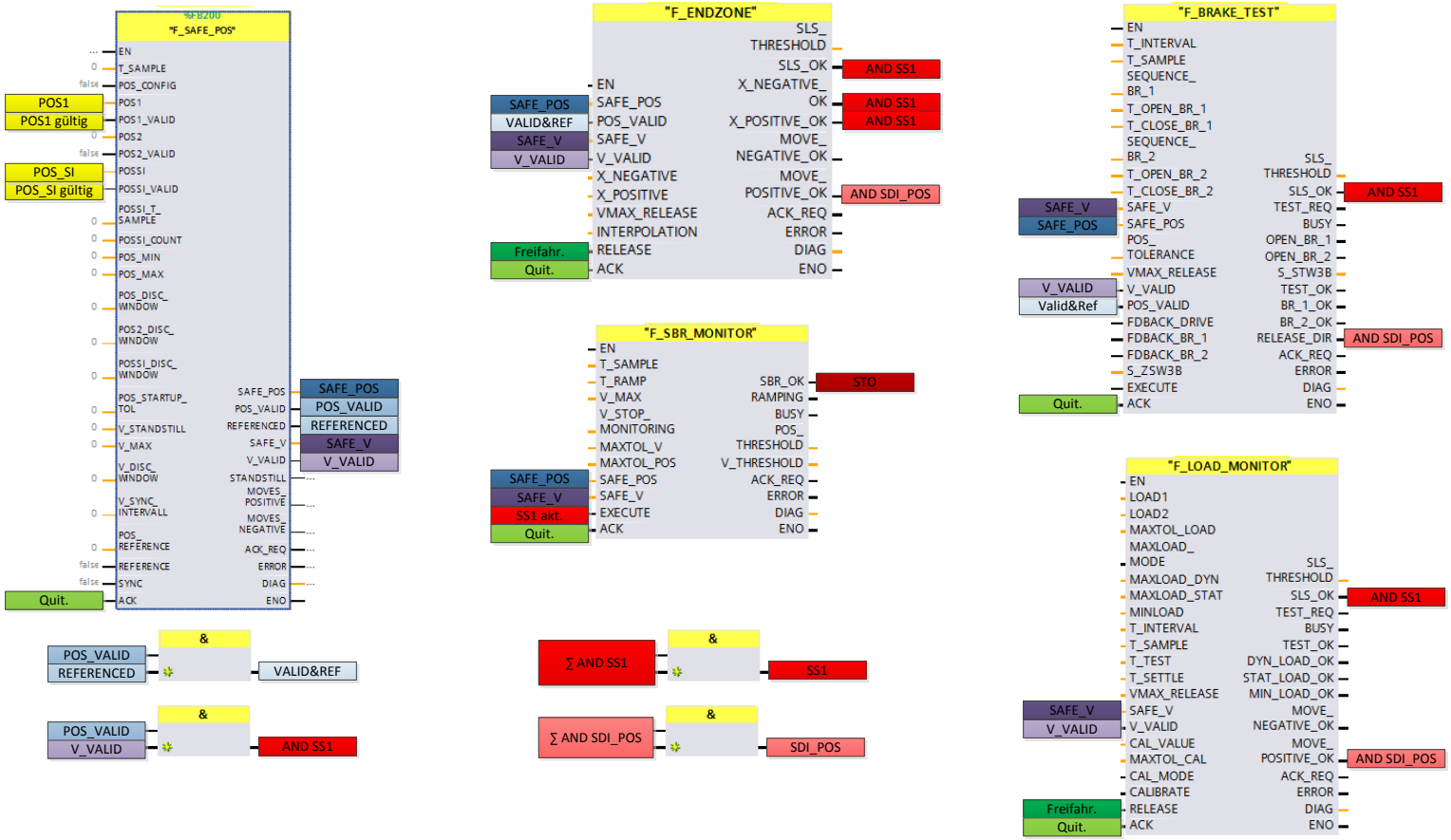
4.4.1 1-Encoder variant

Figure 4-1: Block interconnection



4.4.2 2- and 3- encoder variant

Figure 4-2: Block interconnection for 2- and 3- encoder variant



4.4.3 Additionally required blocks

The following blocks in the STEP7 Safety Advanced library are called in the fail-safe function blocks - and must therefore be available in the block folder:

See Chapter 3.1.2

4.4.4 Additional information

Information about configuring and parameterizing the hardware - as well as a description of handling STEP7 and the graphic editor (F-FBD or F-LAD) of SIMATIC safety - are described in the manuals listed:

- SIMATIC Safety - Configuring and Programming
  https://support.industry.siemens.com/cs/ww/en/view/54110126
- SINAMICS S120 Function Manual Safety Integrated
  Fehler! Hyperlink-Referenz ungültig.

5 Abbreviations

Table 5-1

abbreviation	Description
CPU	Central Processing Unit
CU	Control Unit
DB	Data block
DINT	Double integer; 32-bit data type
DQ	Digital output
F-AI	Basic positioner; drive function
F-CPU	Fail-safe analog module
FMEA	Fail-safe central processing unit
HTL	Failure mode and effects analysis
HW	Type of incremental encoder
I-DB	Hardware
INT	Instance data block
PL	Integer; 16-bit data type
SBT	Performance level
SDI	Safe Brake Test
SI	Safe Direction
SIL	Safety Integrated
SIN/COS	Safe Integrity Level
SLS	Sine-cosine; type of incremental encoder
SLU	Safely Limited Speed
SOS	Safe Length Unit
SRS	Safe Operating Stop
SS1	Safety Requirements Specification
SS2	Safe Stop 1
SSI	Safe Stop 2
STARTER	Synchronous serial interface; type of absolute encoder
STO	Drive engineering tool

6 Support

Application Center

If you have any questions about the use of products described in this manual that have not been answered, then contact the Application Center in D-91056 Erlangen

Fehler! Hyperlink-Referenz ungültig.

or your local Siemens contact in the local office and business location.

Fehler! Hyperlink-Referenz ungültig.

Training center

We offer courses to help you get started with the S7 automation system. Please contact your regional Training Center, or the central Training Center in D-90327 Nuremberg.

Phone: +49 (0)911 895–3200

Fehler! Hyperlink-Referenz ungültig.

***SIMATIC* documentation in the Internet/ Siemens Intranet**

• Documentation is available at no charge in the Internet at:

Fehler! Hyperlink-Referenz ungültig.

Contact the Knowledge Manager listed there to quickly locate the documentation that you require.

7 Appendix

Block runtimes

Table 7-1

Function block	CPU 1516F-3 PN/DP	CPU 1517F-3 PN/DP	CPU 1518F-4 PN/DP
F_SAFE_POS	707 µs	91 µs	44 µs
F_SLP_MONITOR	95 µs	16 µs	9 µs
F_ENDZONE	910 µs	105 µs	54 µs
F_BRAKE_TEST	500 µs	76 µs	39 µs
F_SBR_MONITOR	521 µs	58 µs	29 µs
F_LOAD_MONITOR	760 µs	86 µs	46 µs
Complete function set without F_SLP_MONITOR	**3.398 µs**	**416 µs**	**212 µs**
Complete function set without F_ENDZONE	**2.583 µs**	**327 µs**	**167 µs**

8 Related literature

Table 8-1

	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Download page of the entry https://support.industry.siemens.com/cs/ww/en/view/101167223
\3\	SINAMICS S120: Safe Position – SP https://support.industry.siemens.com/cs/ww/en/view/109746390

9 History

Table 9-1

Version	Date	Modifications
V1.0	07/2014	First Edition
V2.0	02/2016	Migration to TIA-Portal
V2.1	04/2017	Revision encoder combinations and F_SAFE_POS