SIEMENS

Application Example • 05/2016

Specification of Limit Values for Safely Limited Speed (SLS) from a Non-Safety HMI

SIMATIC Safety Integrated

https://support.industry.siemens.com/cs/ww/en/view/67634251

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 http://www.siemens.com/industrialsecurity.

Table of Contents

1 Task

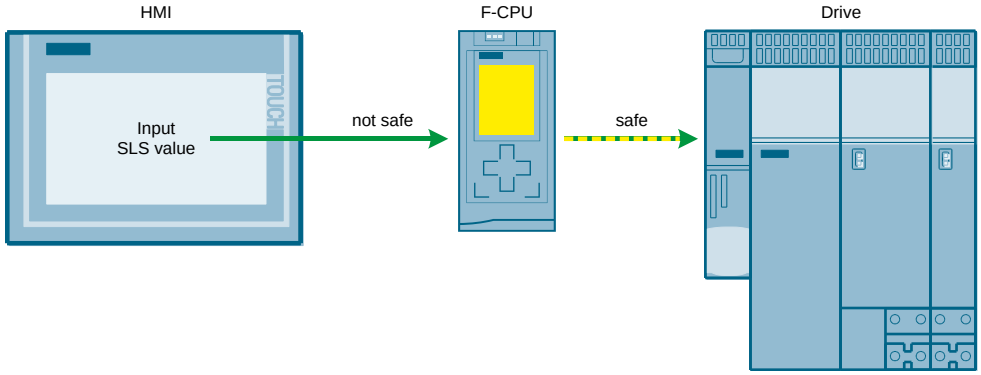
Introduction

Human Machine Interfaces (HMIs) are convenient, essential components in everyday industrial use. In order to use this convenience for operator control and monitoring of processes and plants even in safety-related applications, additional measures are required. This application example shows you how a non-safety HMI is directly involved in the safety function of an application.

Overview of the automation task

The figure below provides an overview of the automation task.

Figure 1-1



Description of the automation task

The limit value for Safely Limited Speed (in the following referred to as the “SLS value”) is to be transferred to a drive with a safety function. When SLS is requested, the actual speed value is compared to the SLS value. If the actual speed value exceeds the SLS value, the drive safely shuts down.

The SLS value is to be specified using a non-safety HMI and transferred to the drive via an F-CPU. As the connection between the HMI and the F-CPU is considered to be non-safe, data corruption could result in an unwanted SLS value being transferred to the F-CPU and then the drive, which can cause dangerous situations.

The core task of this application example is to safely determine an SLS value possibly corrupted during the transfer between the HMI and the controller.

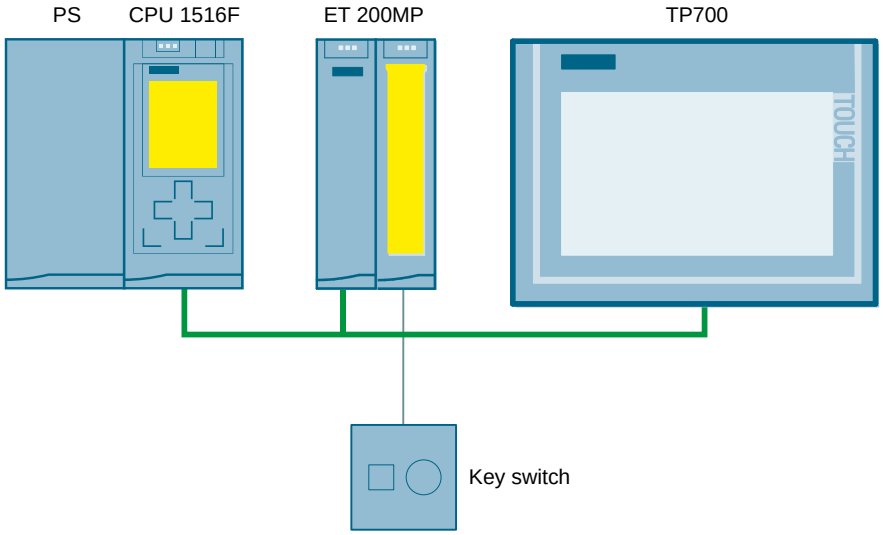
2 Solution

2.1 Overview

Diagrammatic representation

The diagrammatic representation below shows the most important components of the solution:

Figure 2-1



Advantages

The solution presented here offers the following advantages:

- User-friendly transfer of safety-related values for the drive using a non-safety HMI
- The safety concept can also be applied to other tasks
- The F-CPU and the drive with safety functions are certified safety components, i.e., an SLS value in the F-CPU is safely transferred to the drive with safety functions.

Scope

The measures for controlling the effects of errors due to the value transfer process described in this document end with the release of an SLS value checked for data corruption set by the HMI.

Processing the SLS value in the drive is not further discussed in this application example.

In this respect, the following applies:

If the correct SLS value exists safely in the F-CPU, it can be safely transferred to the drive via PROFIsafe mechanisms.

Required knowledge

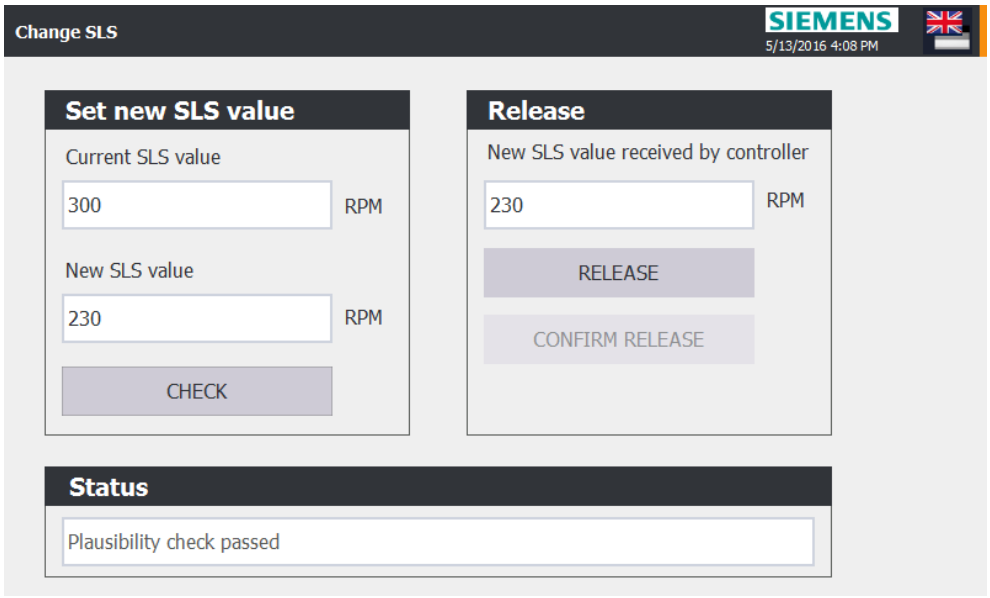
The following knowledge is required:

- Basics of functional safety
- Basics of STEP 7 programming
- Basics of WinCC

2.2 Description of the core functionality

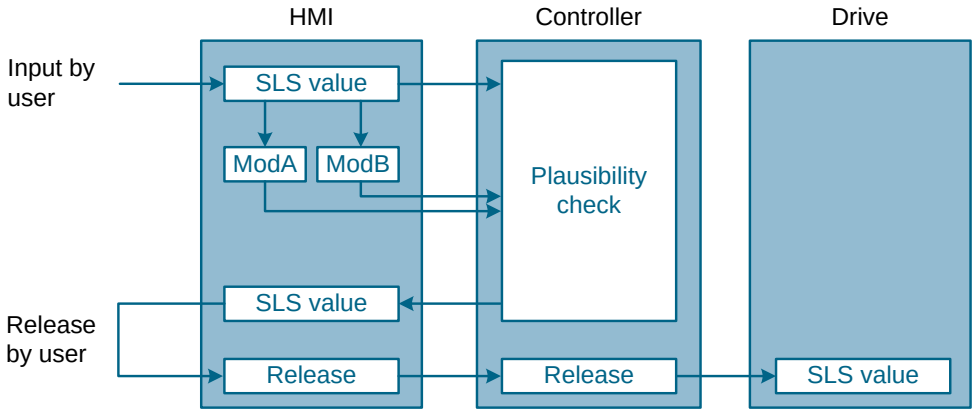
An HMI allows the operator to set a new SLS value. This value is deliberately modified in the HMI. Aside from the SLS value, this modification generates two additional values that are also transferred to the F-CPU. These modified values are used for diagnostics to rule out data corruption.

Figure 2-2



The modifications by the HMI are undone in the safety program. If the two remodified values match the transferred SLS value, data corruption can be ruled out.

Figure 2-3

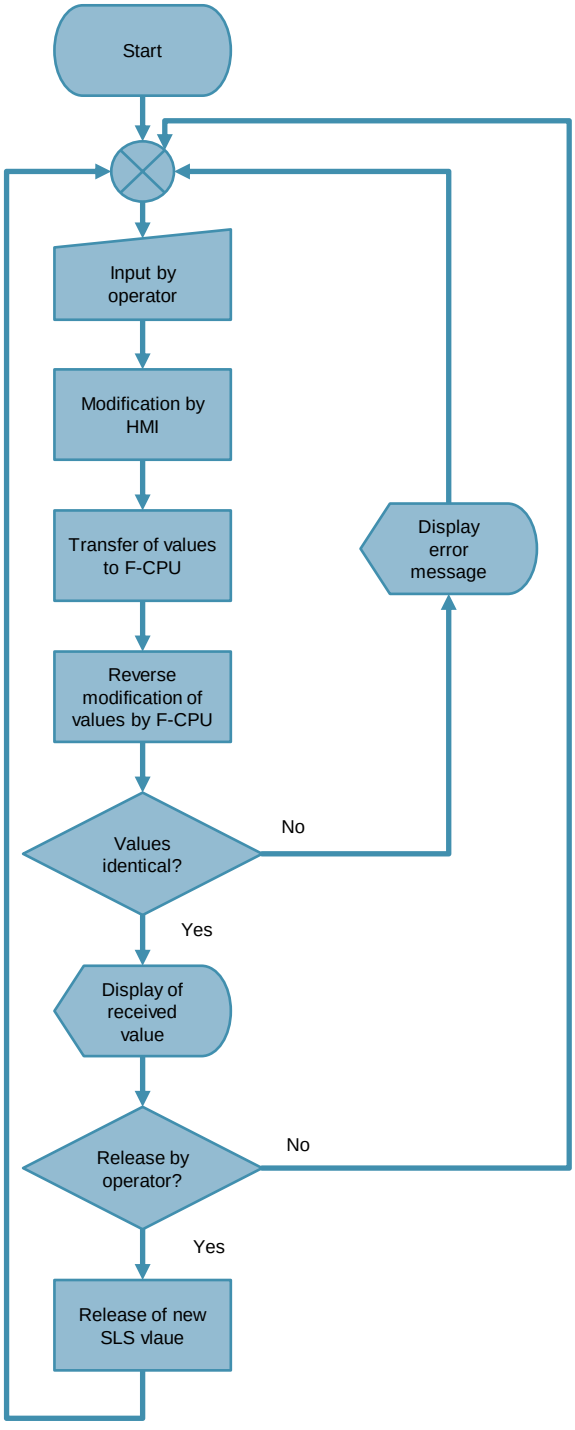


If any data corruption is detected, this information is displayed on the HMI. Only if the three values are identical (correct data transfer) will the HMI display the SLS value to the user for release and the new SLS value can be released by the operator.

Then the new SLS value is transferred from the controller to the drive via the fail-safe PROFIsafe PROFINET profile.

Sequence of the core functionality

Figure 2-4



2.3 Hardware and software components

2.3.1 Validity

This application is valid for

- STEP 7 V13 or higher
- WinCC V13 or higher
- S7-1500

2.3.2 Components used

This application was created with the following components:

Hardware components

Table 2-1

Component	No.	Article number	Note
Power supply	1	6EP1332-4BA00	PM 1507 70 W
Fail-safe S7 CPU	1	6ES7516-3FN00-0AB0	CPU 1516F-3 PN/DP FW 1.8
SIMATIC Memory Card	1	6ES7954-8LC02-0AA0	SMC 4MB
Interface module for ET 200MP	1	6ES7155-5AA00-0AB0	
Fail-safe digital input module	1	6ES7526-1BH00-0AB0	F-DI 16x24VDC
Front connector	1	6ES7592-1AM00-0XB0	Front connector, screw-type, 40-pin
S7-1500 mounting rail	2	6ES7590-1AE80-0AA0	Length: 482 mm
SIMATIC HMI TP700 Comfort	1	6AV2124-0GC01-0AX0	7”
Key-operated switch	1	3SU1100-4BF11-1FA0	Ronis lock, 1NC, 1NO

Note The functionality was tested with the listed hardware components. Similar products not included in the above list can also be used. In this case, please note that changes to the sample code (e.g., different addresses) may become necessary.

Software components

Table 2-2

Component	No.	Article number	Note
STEP 7 Professional	1	6ES7822-1AA03-0YA5	V13 SP1
STEP 7 Safety Advanced	1	6ES7833-1FA13-0YA5	V13 SP1
WinCC Advanced	1	6AV2102-0AA03-0AH5	V13 SP1

Sample files and projects

The following list contains all files and projects that are used in this example.

Table 2-3

Component	Note
67634251_SLS_over_HMI_DOC_V20_en.pdf	This document
67634251_SLS_over_HMI_CODE_V20.zip	This zip file contains the STEP 7 project.

2.4 Achievable SIL or PL

The safety concept described here is suitable to achieve **SIL 3** according to IEC 62061 or **PL e** according to ISO 13849-1.

3 Basics

3.1 Functional safety

From the perspective of the object to be protected, safety cannot be segregated. The causes of danger and also the technical measures to avoid them can vary widely. This is the reason that a differentiation is made between various types of safety, e.g. by specifying the particular cause of a potential hazard. For instance, the term “electrical safety” is used if protection has to be provided against electrical hazards and the term “functional safety” is used if the safety is dependent on the correct function.

In order to achieve the functional safety of a machine or plant, the safety-relevant parts of the protective and control systems must function correctly and must respond in the event of a fault in such a way that the system remains in a safe state or is brought into a safe state.

To achieve this, specifically qualified technology is required, which fulfills the requirements described in the relevant standards. The requirements to achieve functional safety are based on the following basic goals:

- Avoiding systematic faults
- Controlling systematic faults
- Controlling random faults or failures

The measure for the level of achieved functional safety is the probability of the occurrence of dangerous failures, the fault tolerance and the quality that should be guaranteed by avoiding systematic faults. Various terminology is used to express this in the standards:

- In IEC 62061: “Safety Integrity Level” (SIL)
- In ISO 13849-1: “Performance Level” (PL)

For more information on functional safety, please refer to \3\.

4 Principle of Operation

4.1 Complete overview

Safety program overview

The following figure shows the blocks and instructions involved in the safety program.

Figure 4-1 Safety program

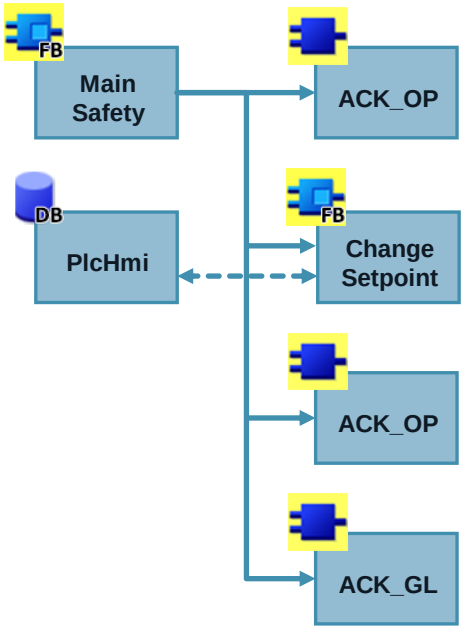


Table 4-1

Block	Function
MainSafety	Called cyclically and calls all the function blocks and instructions involved in the safety function.
ACK_OP (1)	Used to evaluate the release signal from the HMI.
ChangeSetpoint	Core element of this application example in the controller. This is where the plausibility check and the release of the SLS value are performed.
ACK_OP (2)	Used to evaluate the reintegration signal from the HMI.
ACK_GL	Used to reintegrate passivated F-channels.
PlcHmi	Forms the interface between the HMI and the controller.

HMI overview

The following figure shows the elements in the HMI involved in the safety function.

Figure 4-2 HMI

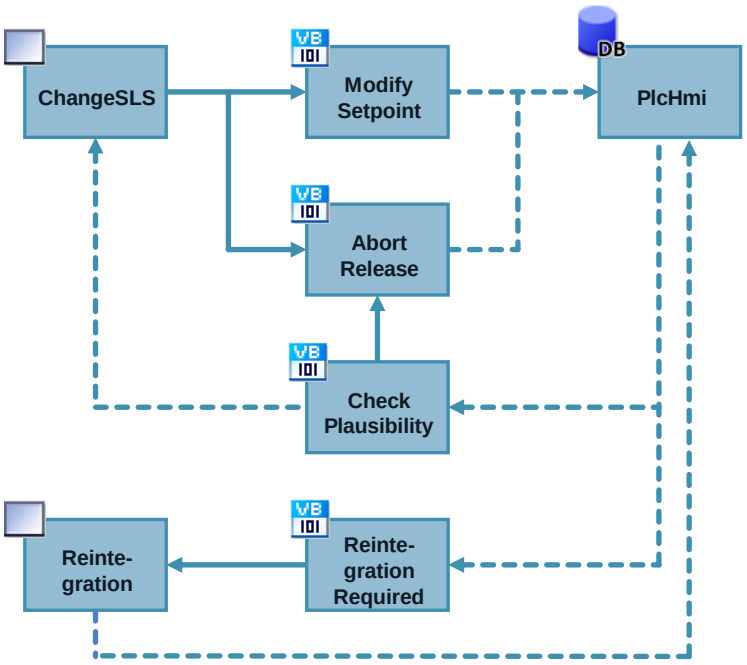


Table 4-2

Element	Function
ChangeSLS	Screen that allows the user to change and release the SLS value.
Reintegration	Screen that allows the user to reintegrate passivated F-channels.
ModifySetpoint	Script that allows the user to modify the SLS value and transfer the values to the “PlcHmi” data block.
AbortRelease	Script that allows the user to abort a started release sequence.
CheckPlausibility	Script that compares the SLS value transferred from the controller to the entered SLS value.
ReintegrationRequired	Script that activates the “Reintegration” screen if a passivated channel is detected.
PlcHmi	Forms the interface between the HMI and the controller.

4.2 “ChangeSetpoint” function block

The “ChangeSetpoint” function block is the core element of this application example in the controller. This is where the plausibility check and the release of the SLS value are performed.

Parameter

Figure 4-3

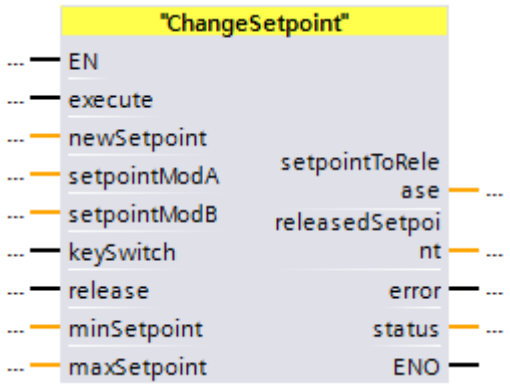


Table 4-3

Parameter	Declaration	Type	Description
execute	IN	Bool	Signal to execute the block
newSetpoint	IN	Int	New SLS value entered by the operator
setpointModA	IN	Int	Modification A of the SLS value entered
setpointModB	IN	Int	Modification B of the SLS value entered
keySwitch	IN	Bool	Key-operated switch for authorization
release	IN	Bool	Release signal from HMI
minSetpoint	IN	Int	Min allowed SLS value
maxSetpoint	IN	Int	Max allowed SLS value
setpointToRelease	OUT	Int	Received, checked SLS value displayed to the operator for release
releasedSetpoint	OUT	Int	Released SLS value
error	OUT	Bool	An error was detected
status	OUT	Word	Status information, see chapter 7.4

Functions

The following functions are implemented in the “ChangeSetpoint” function block:

- Monitoring of the key-operated switch
- Plausibility check of the received SLS value
- Check for specified upper and lower limit
- Compare to the last released SLS value
- Release the SLS value

4.3 Modification of the SLS value in the HMI

In the HMI input field, the user enters the desired SLS value. A “ModifySetpoint” script is used to generate two additional values from the SLS value:

Modification A

The first value modified from the SLS value is generated by

- subtracting 100 from the SLS value
- and then inverting all the bits of this integer value

Modification B

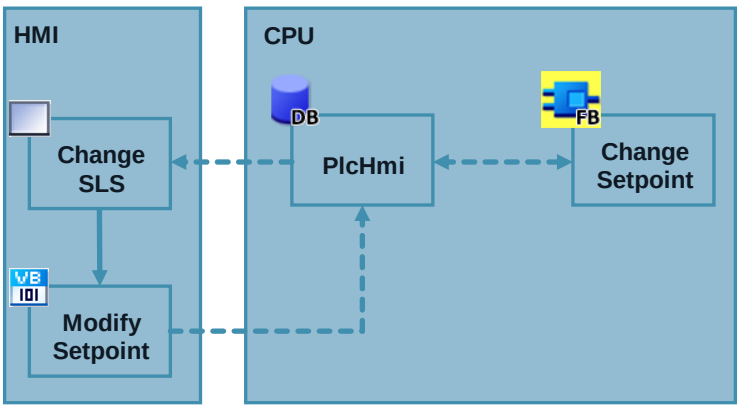
The second value modified from the SLS value is generated by

- adding 100 to the SLS value
- and then inverting all the bits of this integer value

There are therefore now three non-safety values:

- The SLS value entered by the operator
- Two modifications of the entered SLS value

Figure 4-4



When the SLS value has been entered by the operator and modified, these three values are written to the “PlcHmi” data block and therefore transferred to the safety program.

4.4 Plausibility check in the controller

Both the plausibility check and the release of a new SLS value are performed in the “ChangeSetpoint” function block as part of the safety program.

4.4.1 Check key-operated switch

The SLS value can only be changed when the key-operated switch is in the required position. If this is not the case, an error message will be displayed and the block will be prematurely aborted.

4.4.2 Undo modification

Undo modification consists of the following steps:

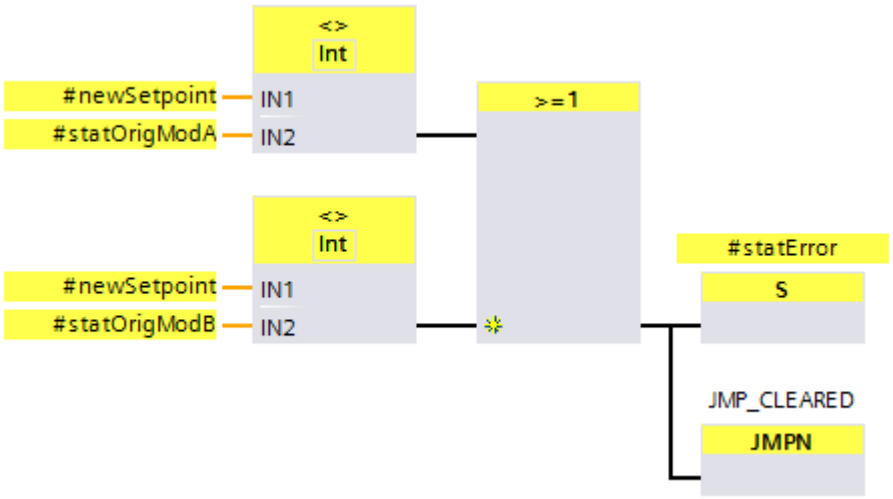
- Invert all the bits of the modified value
- Undo the offset by adding or subtracting 100

Note ADD, SUB and NEG instructions can theoretically result in an overflow. This causes the SIMATIC S7-1500 automation systems to STOP. If you want to catch an overflow and prevent the systems from STOPPING, this function has to be added (see also \6\).

4.4.3 Compare for equality

If no error has occurred, undo modification means that the two resulting values are equal to the SLS value.

Figure 4-5



If an error has occurred, the FB will be aborted at this point. As a result, no incorrect value is displayed to the user on the HMI for release. Therefore, an accidental release is simply not possible.

WARNING

Pending errors are automatically reset when transferring a new SLS value. For applications beyond the scope of this application example, you have to reassess whether this causes potential hazards.

4.4.4 Check limits

When configuring, an upper and lower limit are defined within which authorized staff can change the SLS value. It should not be possible to change these limits during operation; therefore, they are defined as constants in the safety program and transferred to the “ChangeSetpoint” block.

If these limits are violated, the FB will be aborted at this point and an error bit will be set. As a result, no illegal value is displayed to the user on the HMI for release. Therefore, an accidental release is simply not possible.

When reentering an SLS value within the limits, the error will be automatically reset.

Note Please note that MIN_SETPOINT and MAX_SETPOINT are signed number values; the absolute values will not be compared!

If SLS_SET is to be, for example, between -100 and -500, the following applies:
MIN = -500 and
MAX = -100.

If you mix up MIN and MAX (i.e., MIN = -100; MAX = -500), the error bit will always be set.

Note Safety-related minimum and maximum values prepared here are defined by the user during commissioning. Possible critical values are intercepted by the SINAMICS drive through a setpoint limitation.

4.4.5 Compare to the last released SLS value

The check additionally checks whether the absolute value of the new SLS value is higher than the absolute value of the last released SLS value. If this is the case, a warning will be displayed on the HMI.

4.5 Release of the SLS value

Key-operated switch

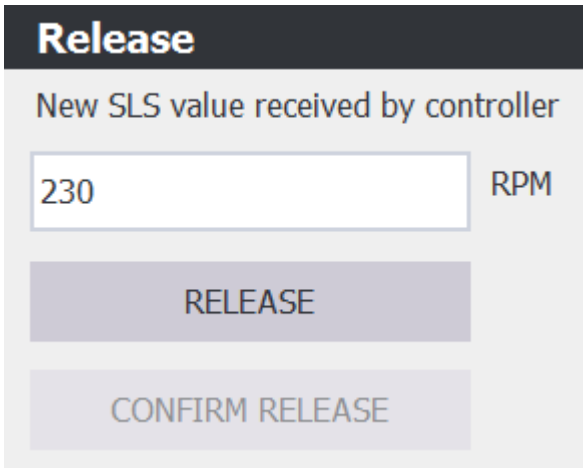
Only authorized staff should be allowed to change the SLS value with the aid of a key. The key-operated switch is monitored by an F-DI module. If a new SLS value is received by the “ChangeSetpoint” FB and the key-operated switch is not in the required position, an error message will be displayed on the HMI and the value cannot be released.

Second plausibility check in the HMI

If no error has occurred, the new SLS value will be saved in the “setpointToRelease” tag in the “PlcHmi” data block. The value to be released is displayed to the operator on the HMI for checking.

The value to be released, “setpointToRelease”, is continuously monitored by the HMI. When the value changes, the “CheckPlausibility” script is called in the HMI where a second plausibility check is performed in the HMI. It checks whether the value to be released transferred from the controller matches the value entered by the operator. If this is the case, the value can be released by the operator. Otherwise, the release buttons are grayed out and cannot be selected.

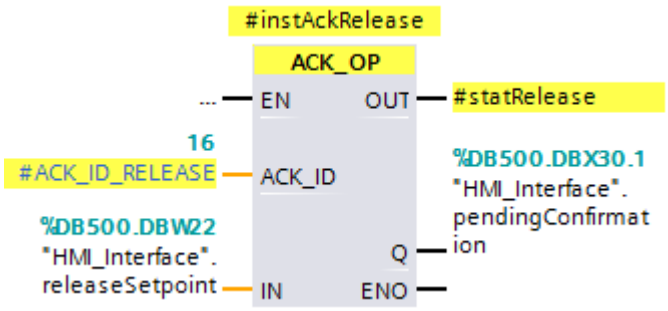
Figure 4-6



Release

The release is based on the use of the certified ACK_OP block from the STEP 7 Safety block library.

Figure 4-7



This instruction allows fail-safe acknowledgment from an operator control and monitoring system. This acknowledgment consists of two steps:

- Change in/out IN to value 6 (defined by the instruction) for exactly one cycle
- Change in/out IN to value at input ACK_ID for exactly one cycle within one minute but at the earliest after one second

The value at the ACK_ID input is stored as a constant in both the safety program and the HMI.

Pressing the “Release” button writes the value 6 to the IN in/out. A second later, the operator can press the “Confirm release” button, which writes the defined ACK_ID to the IN in/out.

Figure 4-8

If the “Confirm release” button is not pressed within one minute, the button will be grayed out and the release must be retriggered.

If an error is detected, both buttons will be grayed out and the value cannot be released. This also applies if an error is detected during the release.

Once the new SLS value has been released by the operator, the new SLS value is written to the “releasedSetpoint” output and displayed on the HMI as the current SLS value. This tag can then be transferred to the drive.

4.6 Measures on the possible errors

Overview

This chapter describes an FMEA (failure mode and effects analysis) for the application example described here. It covers the following errors:

Table 4-4

Communication errors	Description	Chapter
Corruption	**Corrupted SLS value** The SLS value entered on the HMI is transferred to the F-CPU as a corrupted value.	4.6.1
	Corruption of the SLS value and the modified value Aside from the SLS value, the two modified values are also transferred to the F-CPU as corrupted values.	4.6.2
	Incorrect input or acquisition of the entered SLS value An incorrect value is modified and transferred to the F-CPU.	4.6.3
Failure	**Power failure on the HMI**	4.6.4
	Power failure on the F-CPU	4.6.5

WARNING

Pending errors are automatically reset when transferring a new SLS value. For applications beyond the scope of this application example, you have to reassess whether this causes potential hazards.

Basis

The FMEA described here follows DIN EN 61784-3, in particular Table 1 of this standard (“Overview of the effectiveness of the various measures on the possible errors”).

4.6.1 Corruption of the SLS value

Error description

Transfer errors can result in the entered SLS value being incorrectly transferred to the “PlcHmi” data block and transferred to the safety program.

Error detection

After entering the SLS value on the HMI, it is initially stored in the HMI as an internal tag and modified twice. Therefore, there are three values:

- SLS value as an internal tag
- Modification A of the SLS value
- Modification B of the SLS value

These three values are transferred to the safety program, which undoes the modifications. If an error has occurred, undo modification means that the two resulting values are **not equal** to the SLS value.

This allows the user to safely detect that the entered SLS value was incorrectly transferred.

Error response

If an error is detected, the following actions are performed:

- The “error” output of the “ChangeSetpoint” FB is set.
- The error is specified at the “status” output.
- An error message is displayed on the HMI.
- The release buttons remain grayed out. Therefore, a release is not possible.
- The “ChangeSetpoint” function block is prematurely aborted. Therefore, a release on the HMI is not possible, even in the event of an error.

4.6.2 Corruption of the SLS value and the modified values

Error description

Transfer errors can result in both the entered SLS value and the modifications being incorrectly transferred to the “PlcHmi” data block and transferred to the safety program.

In this error scenario, two variants are conceivable:

- **Variant 1:**
  All three values are corrupted such that they differ after the remodification.
- **Variant 2:**
  All three values are corrupted such that they are **identical** after the remodification, but do not match the entered value.

Variant 1 – error detection

If an error has occurred, undo modification means that the two resulting values are **not equal** to the SLS value, regardless of whether one, two or all three values were corrupted.

This allows the user to safely detect that the entered SLS value was incorrectly transferred.

Variant 1 – example

Assumption: SLS value 2000 is entered on the HMI.

If no error has occurred, this would result in the following values:

Table 4-5 Transfer without error

Tag	Value
SLS value	**2000**
Modification A in the HMI	Subtract 100: 1900 (76C hex) Invert all 16 bits of 1900: F893 hex
Modification B in the HMI	Add 100: 2100 (834 hex) Invert all 16 bits of 2100: F7CB hex
Remodification A in the safety program	Two’s complement of F893 hex: 076D hex Minus 1: 76C hex Add 100 (undo offset): **2000**
Remodification B in the safety program	Two’s complement of F7CB hex: 835 hex Minus 1: 834 hex Subtract 100 (undo offset): **2000**

Result: SLS value and remodifications are identical. SLS value was correctly transferred.

4.6 Measures on the possible errors

If an error has occurred, this would result in the following values:

Table 4-6 Transfer with error

Tag	Value
SLS value	Corrupted to **2001** (instead of 2000)
Modification A in the HMI	Subtract 100: 1901 (76D hex) Invert all 16 bits except, for example, bit 1: F890 hex
Modification B in the HMI	Add 100: 2101 (835 hex) Invert all 16 bits except, for example, bit 2: F7CE hex
Remodification A in the safety program	Two's complement of F890 hex: 770 hex Minus 1: 76F hex Add 100 (undo offset): **2003**
Remodification B in the safety program	Two's complement of F7CE hex: 832 hex Minus 1: 831 hex Subtract 100 (undo offset): **1997**

Result: SLS value and remodifications differ. SLS value was incorrectly transferred.

Variant 1 – error response

If an error is detected, the following actions are performed:

- The "error" output of the "ChangeSetpoint" FB is set.
- The error is specified at the "status" output.
- An error message is displayed on the HMI.
- The release buttons remain grayed out. Therefore, a release is not possible.
- The "ChangeSetpoint" function block is prematurely aborted. Therefore, a release on the HMI is not possible, even in the event of an error.

Variant 2 – error detection

This error **cannot** be detected by the "ChangeSetpoint" function block. Error detection is nevertheless ensured:

- The value to be released transferred from the controller is displayed to the user for checking.
- The value to be released transferred from the controller is additionally checked in the HMI. If this value does not match the entered value, it cannot be released. The two buttons remain grayed out.

Variant 2 – error response

If an error is detected, the following actions are performed:

- An error message is displayed on the HMI.
- The two buttons remain grayed out. A release is not possible.

4.6.3 Incorrect input or acquisition of the entered SLS value

Error description

The following error causes are possible:

- The value is entered incorrectly by the operator.
- The entered value is acquired incorrectly by the HMI.

In both cases, the modifications are made for an incorrect SLS value. Therefore, the remodifications are identical to the transferred SLS value and no error is detected.

Error detection

The value to be released is displayed to the operator for checking.

4.6.4 Power failure on the HMI

Error description

The HMI fails due to missing supply voltage.

Error detection

No screen on the HMI.

Error response

The STEP 7 program of the F-CPU continues to run. The SLS value in the F-CPU remains valid. A hazardous state cannot occur as a new SLS value can only be requested and released using the (currently unavailable) HMI.

4.6.5 Power failure on the F-CPU

Error description

The F-CPU fails due to missing supply voltage.

Error detection

No program processing.

Error response

A hazardous state cannot occur as all values go to the safe state. After voltage recovery and RUN mode of the F-CPU, the SLS values must be requested again.

4.6.6 Diversity of the software modules

Requirement

The software modules used for encoding/decoding within the transfer/retransfer process and software modules used for displaying safety-related parameters for the user must use diversity at least for the functions to prevent systematic failures.

Implementation

Diversity is implemented in the bit pattern: on the one hand, by the modification in the HMI and, on the other hand, by the remodification in the safety program.

4.7 Access protection

F-CPU access protection

The F-CPU and access to the F-program require a password. The prepared password is: **siemens**

Key-operated switch

Only authorized persons are allowed to specify SLS values on the HMI. To ensure this, a key-operated switch must be in a certain position during the whole process. The signal of the key-operated switch is assigned to an F-DI on a two-channel basis. For the release of the new SLS value, the F-program checks whether the signal of the key-operated switch is present:

Table 4-7

Signal of the key-operated switch	Evaluation in the F-program	Response
0	Permission for releasing an SLS value does **not** exist.	**F-program:** Entered SLS value is not processed. **HMI:** Displays that no permission exists.
1	Permission for transferring an SLS value does exist.	The SLS value displayed on the HMI can be released.

4.8 Passivation and reintegration of the F-channel

Effects of passivating the F-channel

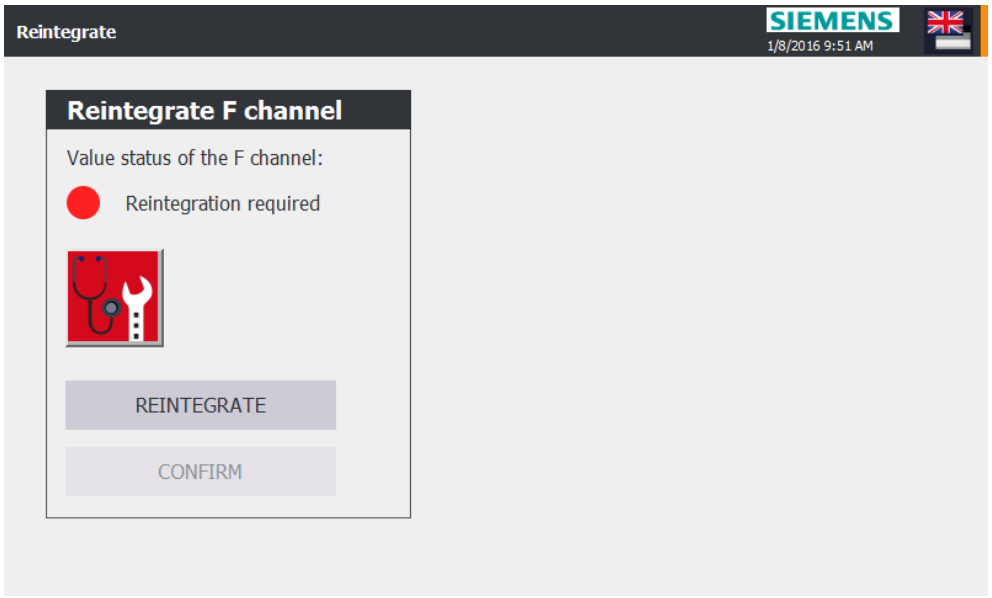
If the channel to which the key-operated switch is connected is passivated, it will have the substitute value “0”. Due to this, it is no longer possible to specify SLS values. The fact that substitute values are output for a channel is indicated by its value status. In this example, this is the “keySwitchVS” tag.

When the reason for the channel’s passivation has been cleared, the channel must be reintegrated.

Implementing reintegration

Reintegration is triggered using the HMI. To do this, the value status of the key-operated switch (“keySwitchVS” tag from the “PlcHmi” data block) is continuously monitored. When a value changes, the “ReintegrationRequired” script is called. If, at this time, the signal is “0”, the channel is passivated and the “Reintegrate” screen is activated.

Figure 4-9

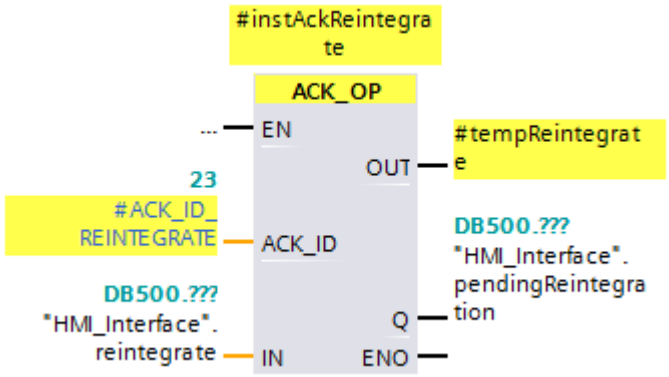


To facilitate finding the error cause, this screen includes a diagnostics indicator that takes the user directly to the diagnostic message of the controller.

4.8 Passivation and reintegration of the F-channel

Reintegration is based on the use of the certified ACK_OP and ACK_GL blocks from the STEP 7 Safety block library.

Figure 4-10



The ACK_OP instruction allows fail-safe acknowledgment from an operator control and monitoring system. This acknowledgment consists of two steps:

- Change in/out IN to value 6 (defined by the instruction) for exactly one cycle
- Change in/out IN to value at input ACK_ID for exactly one cycle within one minute but at the earliest after one second

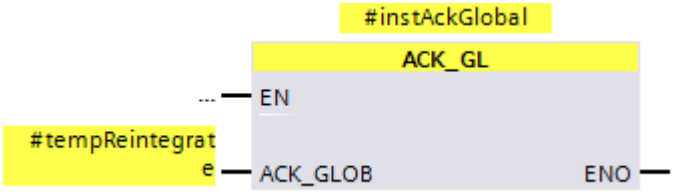
The value at the ACK_ID input is stored as a constant in both the safety program and the HMI.

Pressing the “Reintegrate” button writes the value 6 to the IN in/out. A second later, the operator can press the “Confirm” button, which writes the defined ACK_ID to the IN in/out.

If the “Confirm” button is not pressed within one minute, the button will be grayed out and reintegration must be retriggered.

The output signal of ACK_OP is assigned to the input of ACK_GL, which reintegrates all the passivated channels or modules when there is a positive signal.

Figure 4-11



5 Configuration and Project Engineering

The supplied project does not require any further configuration. If you want to reproduce the application example with other components, this chapter shows the most important settings.

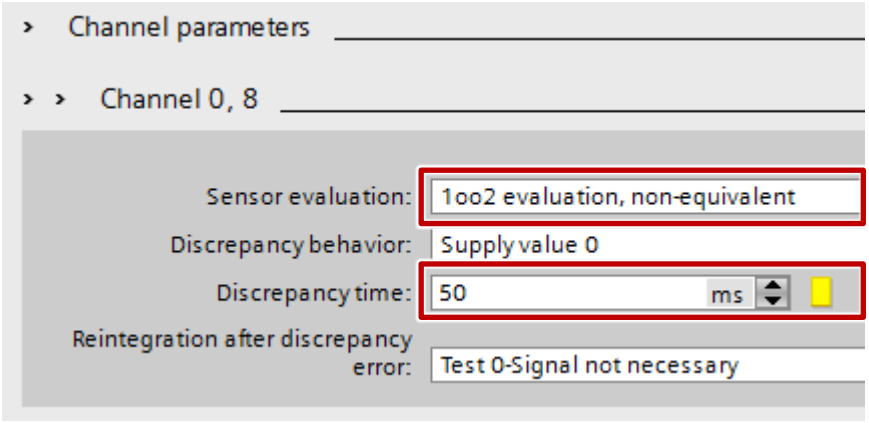
NOTICE	**The settings shown below contribute to achieving the required safety level. Changes to the settings can lead to the loss of the safety function.**

F-DI channel parameters

The key-operated switch is monitored using channel pair 0, 8 of the F-DI. The evaluation of the encoder must be set to “1oo2 evaluation, non-equivalent” in order to detect discrepancies between the two channels and therefore achieve the required safety level.

Due to the inertia of the key-operated switch, an allowed discrepancy time of 50 ms is set.

Figure 5-1

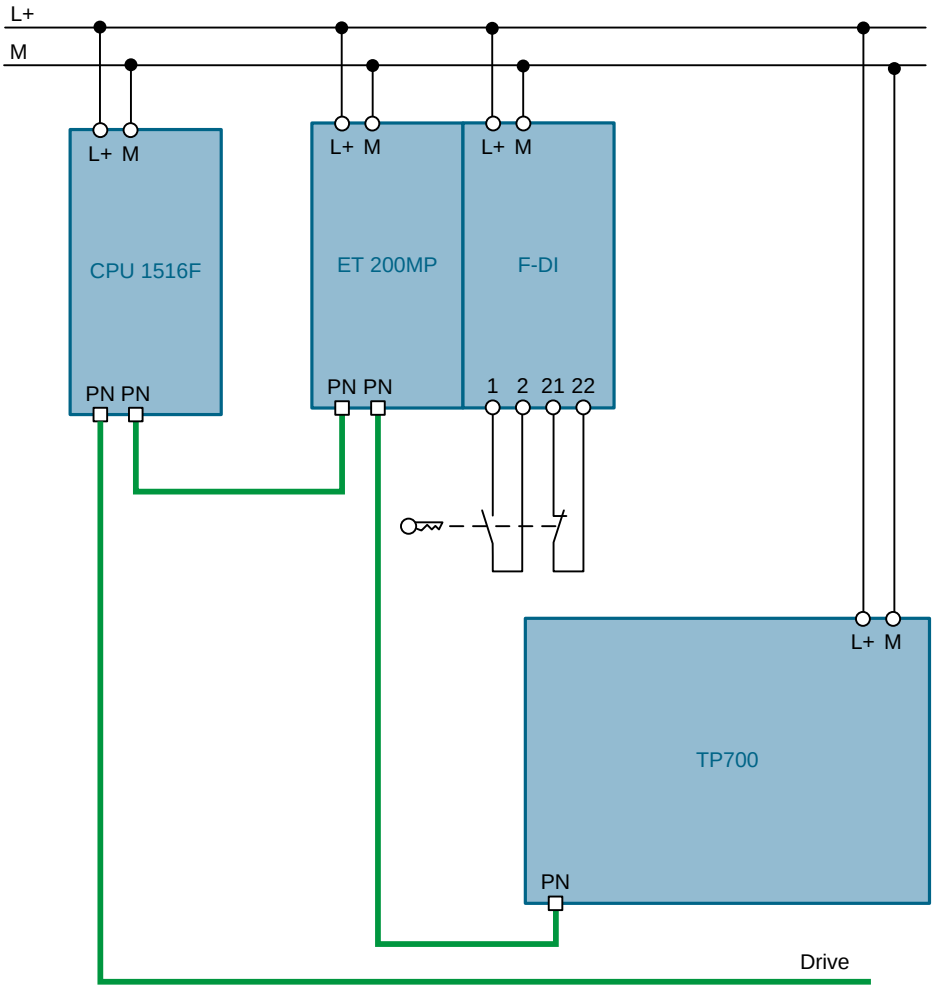


6 Installation and Startup

6.1 Installing the hardware

The figure below shows the rough hardware configuration of the application.

Figure 6-1



1. Mount the components to DIN rails.
2. Connect the components to the 24 V DC power supply.
3. Connect the PROFINET ports of the components with each other.
4. Connect the NO contact of the key-operated switch to terminals 1/2 of the F-DI module.
5. Connect the NC contact of the key-operated switch to terminals 21/22 of the F-DI module.

6.2 Installing the software (download)

6.2.1 Preparation

1. Download the “67634251_SLS_over_HMI_CODE_V20.zip” project file. For the download link, see \2\.
2. Save the zip file to any directory on your computer and extract it.
3. Set the IP address of the PG/PC so that the PG/PC is in the same subnet as the components.
4. Use an Ethernet cable to connect the PG/PC to the Ethernet interface of the switch.

The following IP addresses were used for this application example:

CPU S7-1516F
IP address: 192.168.0.1
Subnet mask: 255.255.255.0

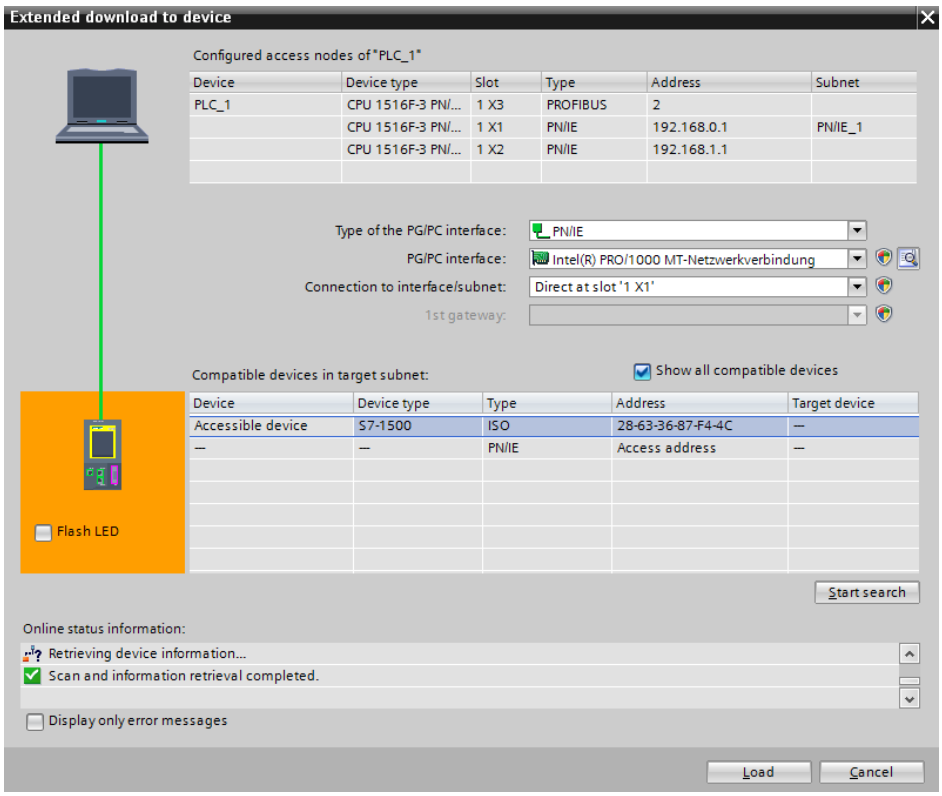
IM 155-5PN ST
IP address: 192.168.0.2
Subnet mask: 255.255.255.0

TP700
IP address: 192.168.0.3
Subnet mask: 255.255.255.0

6.2.2 Downloading the S7 project to the CPU S7-1516F

1. Open “TIA Portal V13”.
2. Go to the Project view.
3. In the menu bar in TIA Portal, click “Project > Open”.
4. Click “Browse” and open the extracted project.
5. Set the CPU S7-1516F to STOP.
6. In the Project tree, right-click “PLC_1 [CPU1516F-3 PN/DP]” and then select “Download to device > Hardware and software (only changes)”.
7. Select the appropriate interface and click “Start search”.

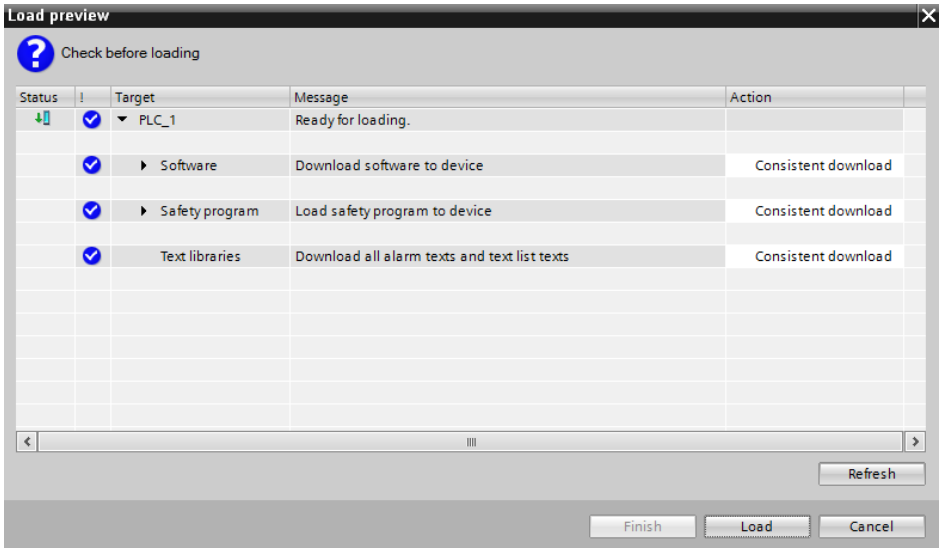
Figure 6-2



8. Select the CPU based on the MAC address and then click “Load”.

Note The IP address and the device name are automatically assigned when downloading the project to the CPU.

Figure 6-3



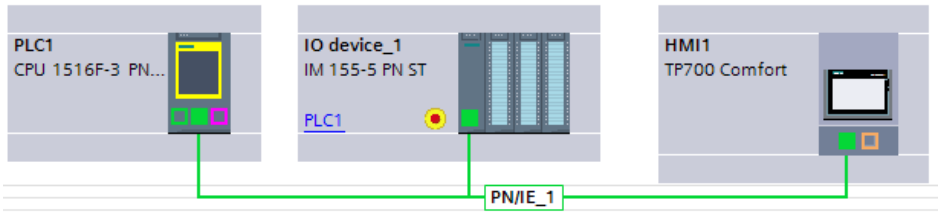
9. Confirm the dialog by clicking “Load”.
10. Click “Finish” when loading is complete.

6.2.3 Assigning device names

The device name of the CPU is automatically assigned during loading. The device name of the ET 200MP has to be assigned manually. To do this, proceed as follows:

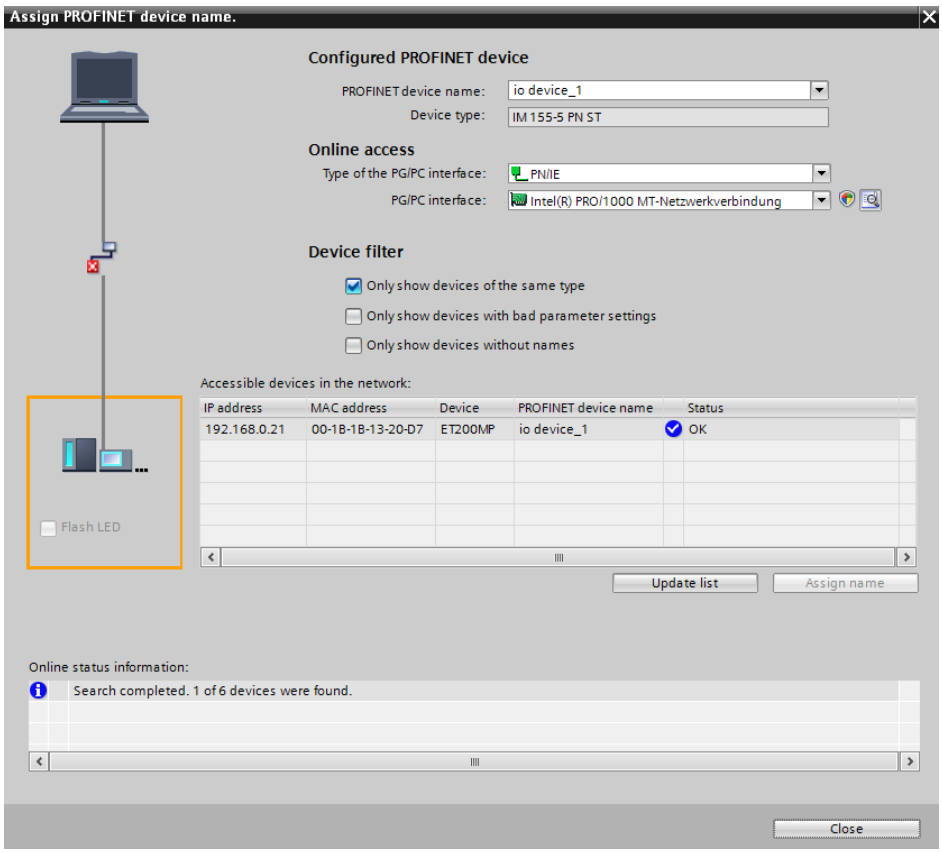
1. In the Project tree, open “Devices & networks”.
2. Right-click the ET 200MP and select “Assign device name”.

Figure 6-4 Devices & networks



3. Click “Update list” and select the ET 200MP based on the MAC address.
4. Now click “Assign name” and close the window when the status is labeled as “OK”.

Figure 6-5 Assign device name



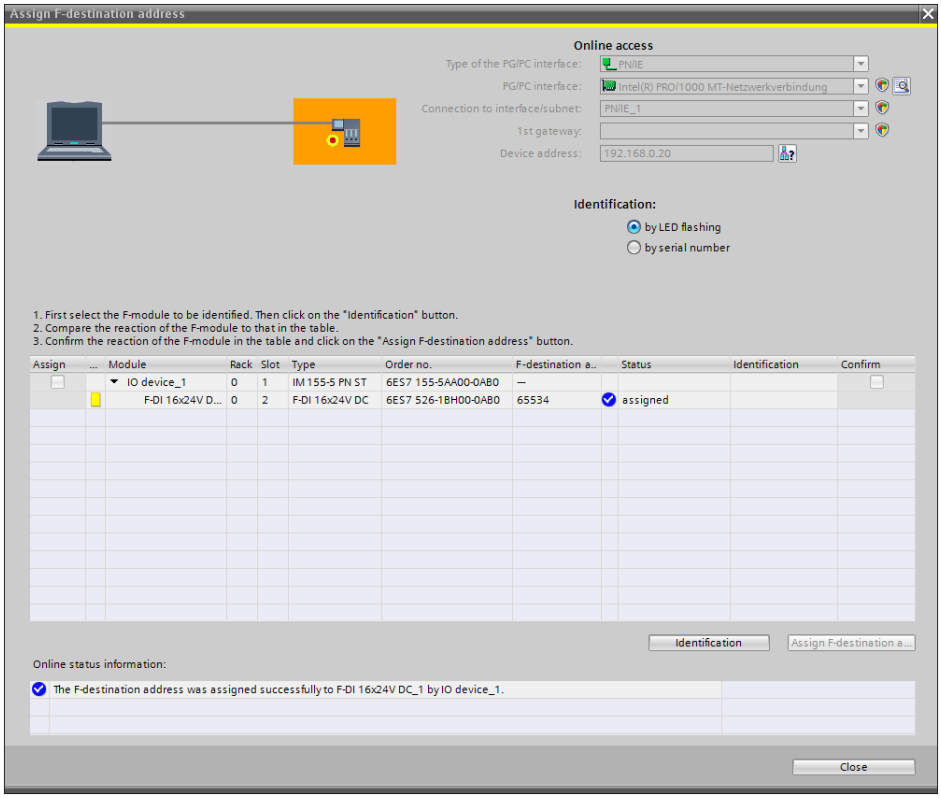
6.2.4 Assigning F-destination addresses

To establish secure communication between the F-CPU and the fail-safe modules of the ET 200MP, F-destination addresses must be assigned to the modules.

Note As the F-address is saved in the electronic coding element, the following steps are only required if the coding element has not already been assigned an F-address or a different F-address.

1. In the Project tree, open “Devices & networks”.
2. Right-click the ET 200MP station and select “Assign F-destination address”, see Figure 6-4.
3. Check the check box of the first fail-safe module and click the “Identification” button.
4. When the LEDs of the F-DI simultaneously flash green every second, check the “Confirm” check box.
5. Then click the “Assign F-destination address” button and confirm the dialog by clicking “Yes”.

Figure 6-6



6. Then you can close the window.

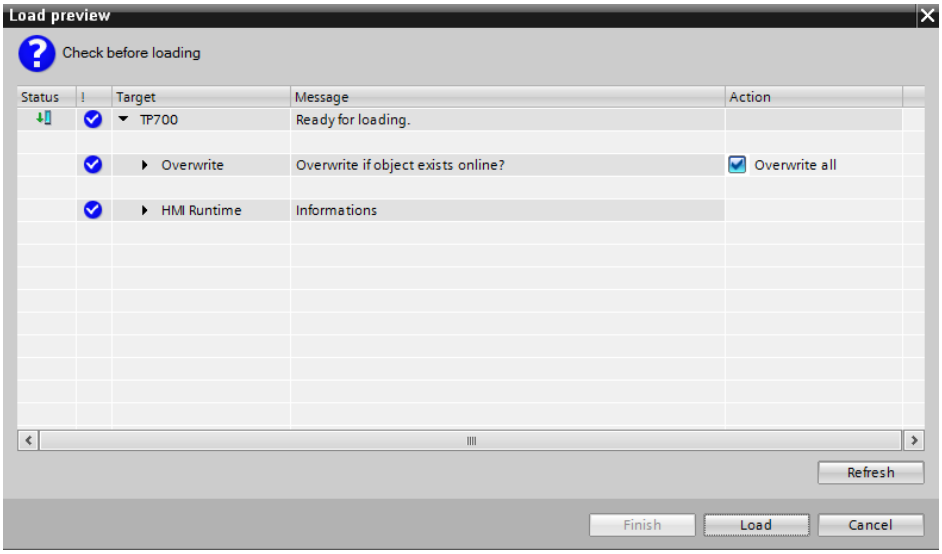
Note All red LEDs of the ET 200MP station should go off after assigning the F-destination address. If this is not the case, there may be a wiring error.

7. Now set the CPU S7-1516F to RUN.

6.2.5 Downloading the WinCC project to the HMI

1. Make sure that the HMI is in transfer mode or automatic transfer is allowed.
2. In the Project tree, right-click “TP700 [TP700 Comfort]” and then select “Download to device > Software (only changes)”.
3. Select the appropriate interface and click “Start search”.
4. Select the HMI based on the IP or MAC address and then click “Load”.

Figure 6-7



8. Check the “Overwrite all” check box and click “Load”.

7 Operation of the Application

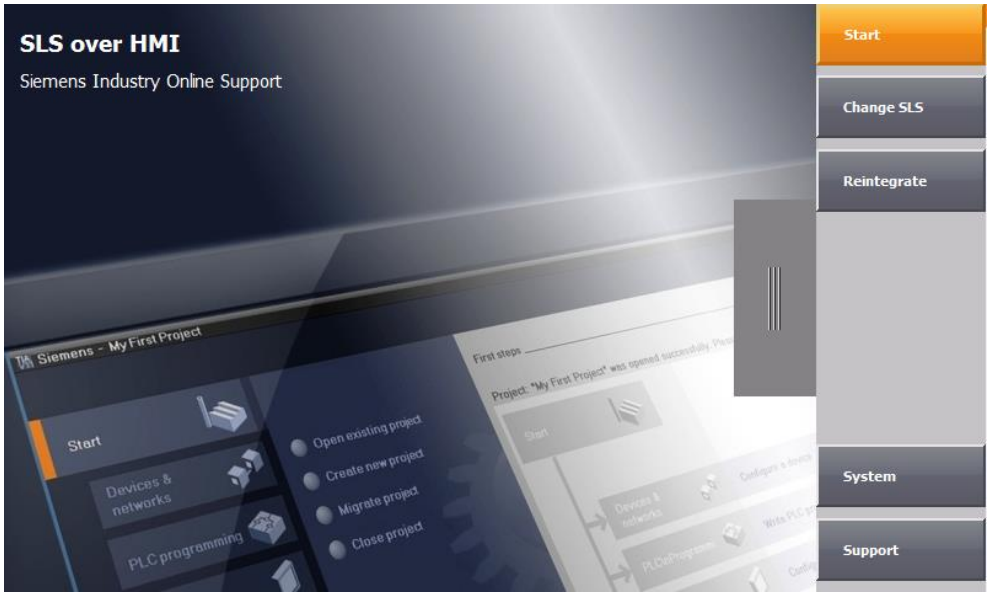
7.1 Overview

The user interface of the application example consists of two screens:

- Change SLS
- Reintegrate

A slide-in screen is used for navigating on the HMI. Tapping the right screen edge opens the slide-in screen and the user can select the desired screen.

Figure 7-1



7.2 Change SLS value

Proceed as follows to release a new SLS value.

Table 7-1

No.	Action	Comment
1.	Tap the right screen edge to open the slide-in screen with the navigation bar.	
2.	In the navigation bar, press “Change SLS”.	The “Change SLS” screen opens.
3.	Press the “New SLS value” text field and enter the desired value.	
4.	Press “Check”.	The SLS value is modified and written to the data block. If the plausibility check in the controller was successful, the entered value is displayed in the “New SLS value received by controller” text field.
5.	Press the “Release” button.	
6.	After one second – do not wait longer than one minute –, press the “Confirm release” button.	The new SLS value has been released and is now displayed as the current SLS value.

7.3 Reintegrating the passivated channel

Passivation of the channel of the key-operated switch automatically opens the “Reintegrate” screen. Proceed as follows to read the diagnostic messages out of the controller and reintegrate the passivated channel.

Table 7-2

No.	Action	Comment
1.	Press the diagnostics indicator.	A diagnostics window opens.
2.	In the diagnostic messages, navigate to the F-DI module.	
3.	Clear the error.	
4.	Close the diagnostics window.	If the error has been cleared, the “Reintegrate” button will be enabled.
5.	Press the “Reintegrate” button.	
6.	After one second – do not wait longer than one minute –, press the “Confirm” button.	The channel is reintegrated and the value status is displayed in green.

7.4 Diagnostics

Via the “status” output, the “ChangeSetpoint” function block provides status and error messages. On the HMI, these messages are displayed on the “Change SLS” screen in plain text.

Table 7-3

Status	Meaning	Remedy
16#0000	Job complete: new SLS value successfully released.	
16#7000	Not processing any job	
16#7003	Plausibility check OK	
16#7004	New SLS value higher than current SLS value.	
16#7005	New SLS value identical to current SLS value.	Enter a different SLS value.
16#8001	New SLS value below allowed minimum value.	Enter an SLS value within the allowed limits.
16#8002	New SLS value above allowed maximum value.	Enter an SLS value within the allowed limits.
16#8003	Plausibility error: the remodified values do not match the SLS value.	Check the communication between the HMI and the controller, in particular the “PlcHmi” data block. Then select the “Check” button again.
16#8004	Authorization missing	Operate the key-operated switch.

8 Links & Literature

Table 8-1

	Topic
\1\	Siemens Industry Online Support http://support.industry.siemens.com
\2\	Download page of the entry https://support.industry.siemens.com/cs/ww/en/view/67634251
\3\	Functional Safety at Siemens www.siemens.com/safety-integrated
\4\	Safety Evaluation Tool (SET) http://siemens.com/safety-evaluation-tool
\5\	SIMATIC Safety – Configuring and Programming https://support.industry.siemens.com/cs/ww/en/view/54110126
\6\	FAQ “How can an overflow of arithmetic operations in the fail-safe program be intercepted?” https://support.industry.siemens.com/cs/ww/en/view/109482083

9 History

Table 9-1

Version	Date	Modifications
V1.0	03/2013	First version
V2.0	05/2016	Migration to STEP 7 V13, S7-1500 and ET 200MP