

SIEMENS

Ingenuity for life

Industry Online Support

Home

Block for evaluating the diagnostics interface signal

SITOP PSU6200

<https://support.industry.siemens.com/cs/ww/en/view/109760217>

Siemens
Industry
Online
Support



Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.

Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.

By using the application examples you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.

The Siemens terms of use (<https://support.industry.siemens.com>) shall also apply.

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 constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place. For additional information on industrial security measures that may be implemented, please visit <https://www.siemens.com/industrialsecurity>.

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

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: <https://www.siemens.com/industrialsecurity>.

Table of contents

Legal information	2
1 Introduction	5
1.1 Overview.....	5
1.2 Principle of operation.....	5
1.2.1 Application with S7-1500.....	6
1.2.2 Application with S7-1200.....	7
1.2.3 Application with S7-300 and S7-400	8
1.2.4 Effect diagram	9
1.2.5 Signal profile.....	10
1.2.6 Functional sequences of the FB "LPSU6200_SignalEvaluationS71500" and FB "LPSU6200_SignalEvaluationS71200"	11
1.2.7 Internally used instructions for S7-1500 CPUs and S7-300 / S7-400 CPUs.....	12
1.2.8 Internally used instructions for S7-1200 CPUs	13
1.2.9 PSU6200 telegram content	14
1.3 Components used	19
2 Engineering	20
2.1 Interface description	20
2.1.1 PLC data type for S7-1200 and S7-1500	20
2.1.2 PLC data type and HMI data type for S7-300 / S7-400	22
2.1.3 FB "LPSU6200_SignalEvaluationS71500"	24
2.1.4 FB "LPSU6200_SignalEvaluationS71200"	26
2.1.5 FB "LPSU6200_SignalEvaluationS7300"	28
2.1.6 Faceplate "fpComAdvDetailS7xyPSU6200"	30
2.1.7 Faceplates "fpComAdvOverviewS7xy00PSU6200"	31
2.2 Projecting Facilities	32
2.2.1 Creating a TIA Portal project.....	32
2.2.2 Integration of a S7-CPU into the user project	34
Add S7-CPU via hardware catalog	34
Adjust hardware configuration for S7-CPU	35
Integration of a digital input module into the user project	36
Assign the sub-process image (TPA) of the digital input module to the cyclic interrupt OB.....	38
2.2.3 Integration of an HMI operating device	39
Adding an HMI operating device	39
Configuring the HMI operating device	41
2.3 Creating communication.....	42
2.3.1 Creating an HMI connection.....	42
2.3.2 Creating PLC variables	44
2.4 Integration of the function blocks in the user program	45
2.4.1 Opening the library "LPSU6200_V15.0-0_V1-1-0" in STEP 7	45
2.4.2 Copying Data Types and Function Modules to the User Project	47
2.4.3 Create a cyclic interrupt OB for S7-1200 and S7-1500.....	48
2.4.4 Create a cyclic interrupt OB for S7-300 / S7-400.....	50
2.4.5 Creating a Global Data Block for Data Exchange	54
2.4.6 Call the function block in the user program.....	57
2.5 Integrating Faceplates for S7-1200 and S7-1500	58
2.5.1 Create an HMI tag	58
2.5.2 Faceplate "fpComAdvDetailS71500PSU6200" and " fpComAdvDetailS71200PSU6200"	60
2.5.3 Faceplate "fpComAdvOverviewS71500" and "fpComAdvOverviewS71200"	63
2.6 Integrating Faceplates for S7-300 / S7-400	65

Table of contents

2.6.1	Creating HMI tag	65
2.6.2	Faceplate "fpComAdvDetailS7300PSU6200"	67
2.6.3	Faceplate "fpComAdvOverviewS7300PSU6200"	70
2.7	Compiling and Downloading.....	72
2.7.1	User program of the S7-CPU	72
2.7.2	HMI operating device	75
3	Operation	80
3.1	Faceplate "fpComAdvDetailS7xy00PSU6200"	80
3.1.1	Overview.....	80
3.1.2	Header area	80
3.1.3	Status area	81
3.1.4	Navigation area	82
3.1.5	Information area	82
3.2	Faceplate "fpComAdvOverviewS7xy00PSU6200".....	86
3.2.1	Overview.....	86
4	Error handling	87
4.1	Status display FB "LPSU6200_SignalEvaluationS7xy00"	87
4.2	Display of hashes ("#####") in the faceplate	88
5	Appendix	89
5.1	Service und support	89
5.2	Links and Literature	90
5.3	Change documentation	90

1 Introduction

1.1 Overview

The library "LPSU6200_V15.0-0_V1_0_0" provides functions and screen blocks for the SITOP PSU6200. The function blocks are used to read out the device data of the power supply. You can use the screen blocks to clearly display the device data on an HMI operating device.

The pre-assembled function blocks (FBs) save you time and money in engineering. You also do not have to worry about the serial communication of the SITOP PSU6200 with your controller.

1.2 Principle of operation

The diagnostics interface "COM" of the PSU6200 provides a signal that transmits the following device data as a Manchester code.

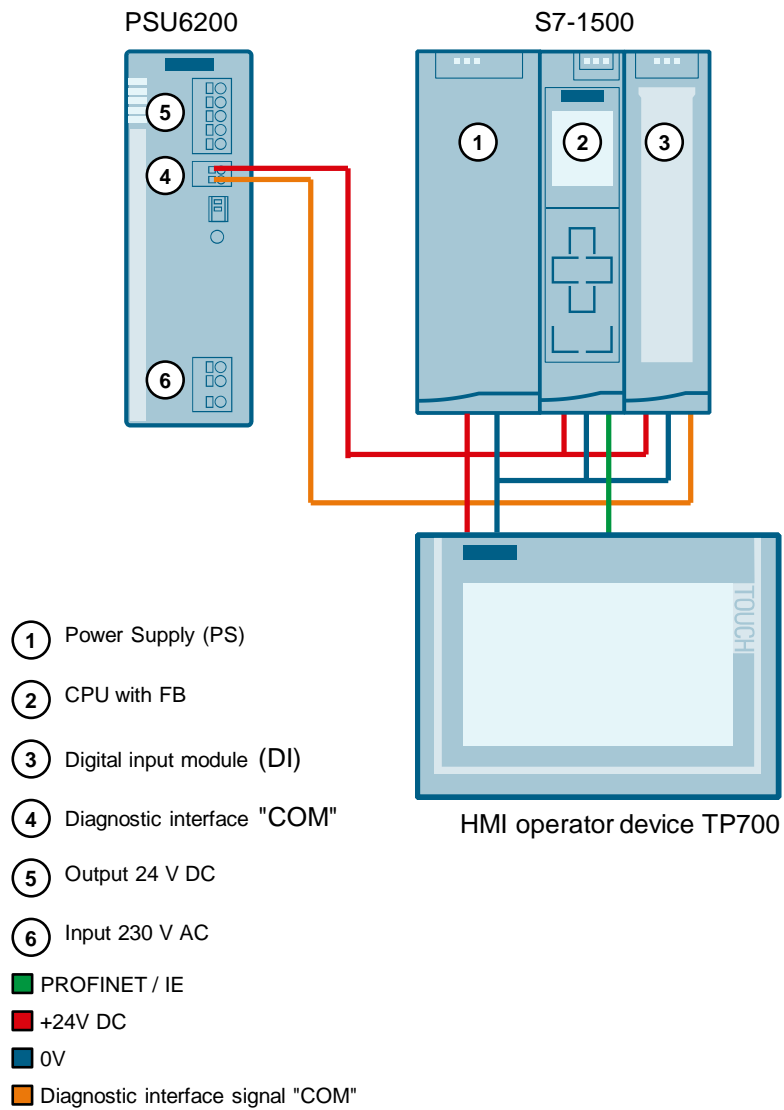
- Voltage and current values
- Nominal data
- Device information
- Temperature and operating hours

The signal of the diagnostics interface "COM" is read in and evaluated by the S7-CPU via a digital input. The aforementioned device data is stored in the S7-CPU in a pre-defined data structure (PLC data type). The HMI device accesses this predefined data structure to display the device data.

1.2.1 Application with S7-1500

The following figure shows the connection of the PSU6200 to a S7-1500 digital input module.

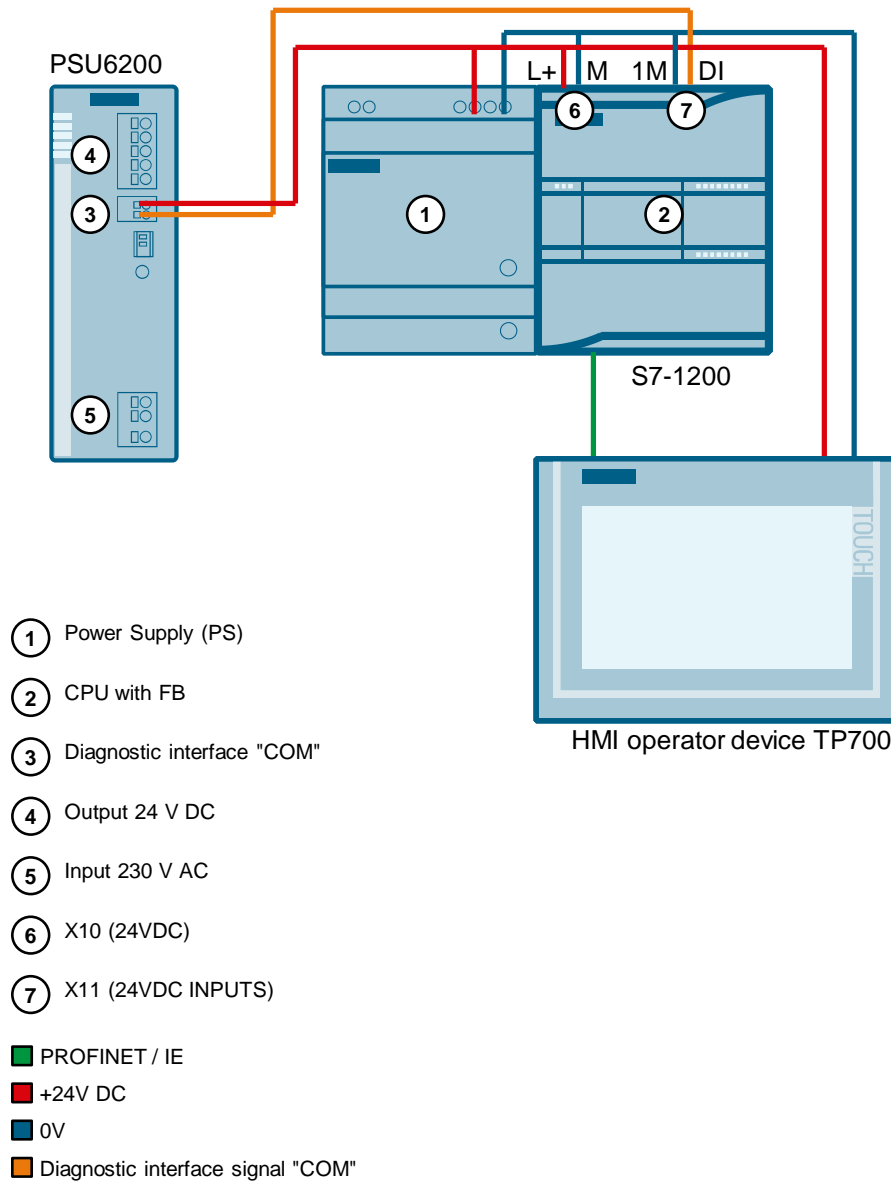
Figure 1-1



1.2.2 Application with S7-1200

The following figure shows the connection of the PSU6200 to a digital input of the S7-1200 CPU.

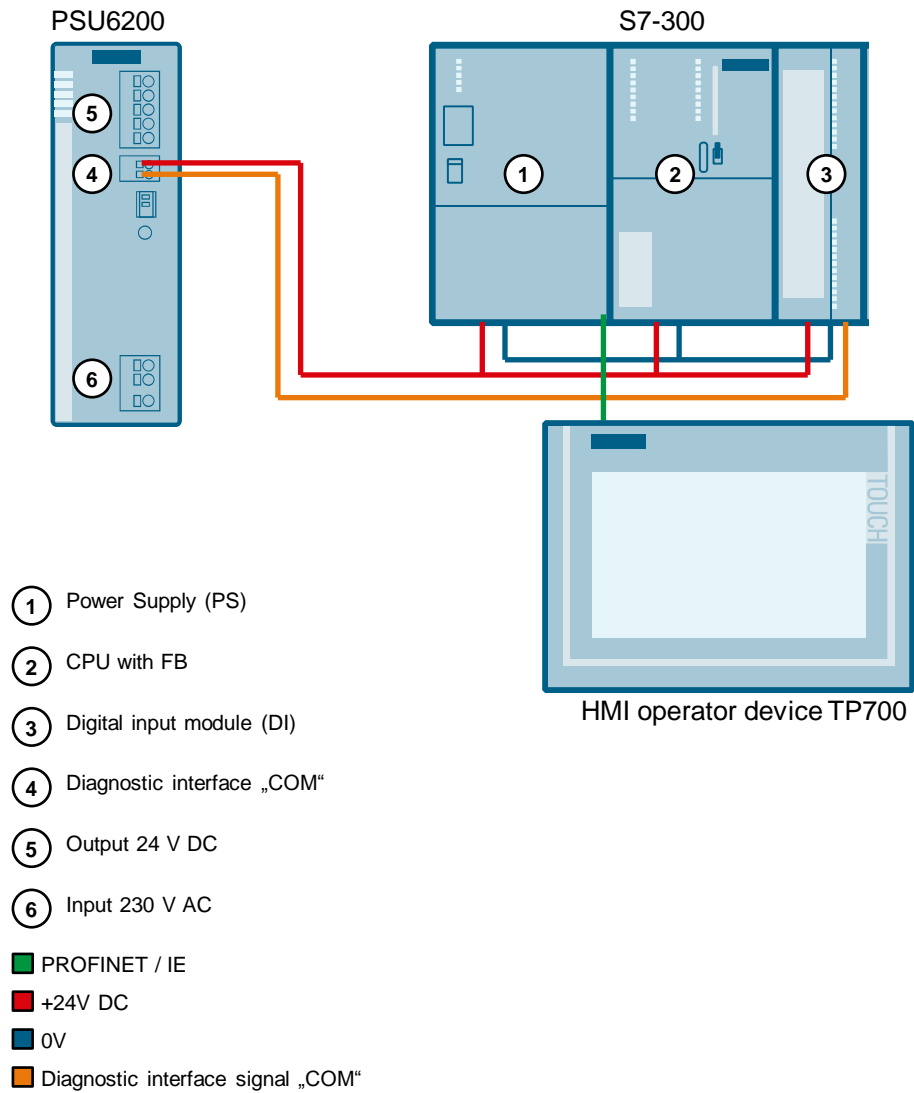
Figure 1-2



1.2.3 Application with S7-300 and S7-400

The following figure shows the connection of the PSU6200 to a S7-300 digital input module.

Figure 1-3



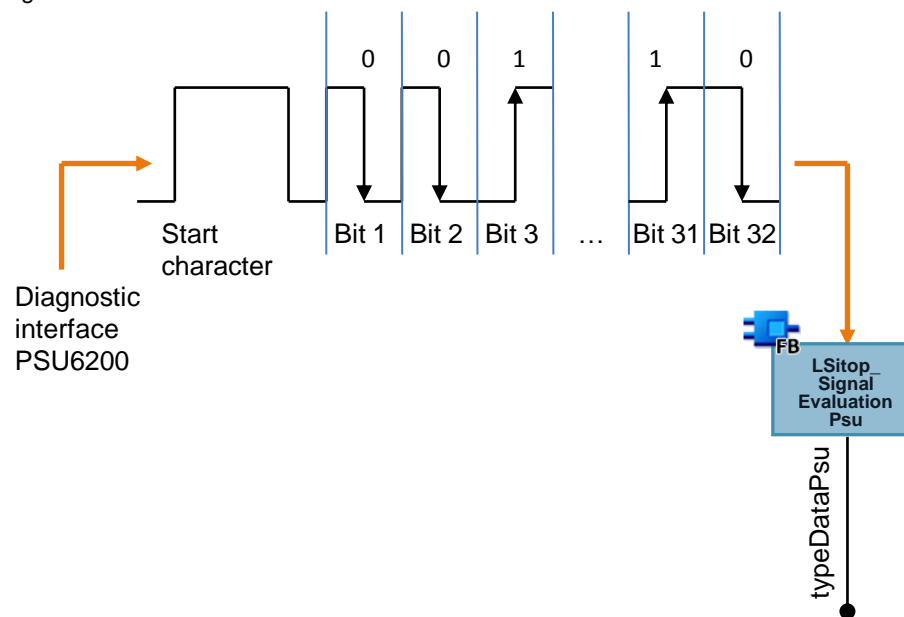
1.2.4 Effect diagram

The library provides the function block for following CPUs, respectively:

- S7-1500 CPUs
- S7-1200 CPUs
- S7-300 / S7-400 CPUs

Call the FB in the in the user program of S7-CPU, in order to evaluate the signal of the diagnostics interface "COM". The FB reads the signal of the diagnostics interface "COM" via a digital input. At its output, voltage and current values, nominal data, device data, temperature and operating hours are stored in a predefined data structure (PLC data type).

Figure 1-4



1.2.5 Signal profile

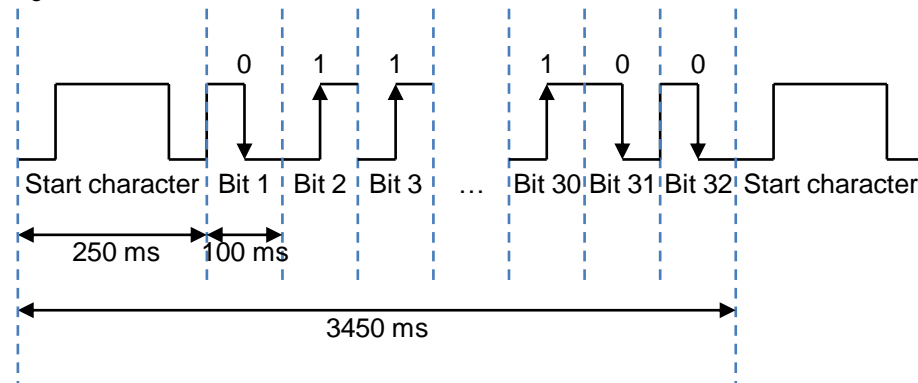
[Figure 1-5](#) shows the signal profile. A signal telegram consists of a start character and 32 bits.

Start character is structured as follows:

- 50 ms low level
- 150 ms high level
- 50 ms low level

The individual bits are transferred as a Manchester code. A bit has a duration of 100 ms, whereby the high level and the low level are each 50 ms long. A "0" is represented by a falling edge, i.e. a "0" is represented by a high level followed by a low level. A "1" is represented by a rising edge, i.e. a "1" is represented by a low level followed by a high level.

Figure 1-5

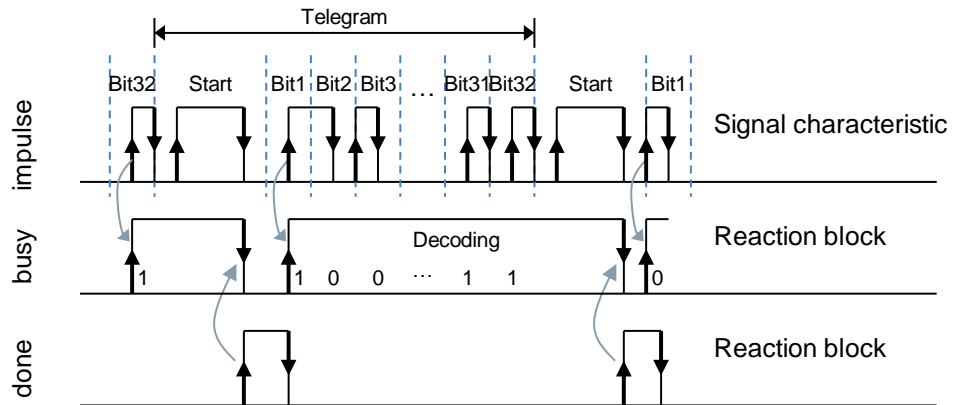


1.2.6 Functional sequences of the FB "LPSU6200_SignalEvaluationS71500" and FB "LPSU6200_SignalEvaluationS71200".

Figure 1-6 shows the graphic representation of the function effects of the following function blocks:

- FB "LPSU6200_SignalEvaluationS71500"
- FB "LPSU6200_SignalEvaluationS71200"
- FB "LPSU6200_SignalEvaluationS7300"

Figure 1-6



Note The instance DB of the FB "LPSU6200_SignalEvaluationS71500", FB "LPSU6200_SignalEvaluationS71200" and FB "LPSU6200_SignalEvaluationS7300" is generated when the function block is called. You need a separate instance DB for each call of the function module.

1.2.7 Internally used instructions for S7-1500 CPUs and S7-300 / S7-400 CPUs

The following instructions are used internally in the function block for the S7-1500 CPUs and S7-300 / S7-400 CPUs.

- **TIME_TCK:** The instruction TIME_TCK reads the system time of the CPU. The system time is a time counter that counts from 0 to a maximum of 2147483647 ms. In the event of an overflow, counting starts again from "0". The time grid and the accuracy of the system time is 1 ms. In the function block the system time is used to calculate the cycle time and to determine the length of the high and low levels.
The TIME_TCK instruction is found in the "Instructions" task card under "Extended instructions > Date and time-of-day > Clock functions".
- **TON:** The instruction generates a switch-on delay. In the function block, the switch-on delay is used when generating the error messages.
The TON instruction can be found in the Task Card "Instructions" under "Basic instructions > Timer operations".

Compare time values

To calculate the cycle time and the length of the high and low levels, the system time read must be greater than the system time read and stored in the last cycle, i.e. comparison expressions are used in the FB to compare the contents of two variables of the data type TME.

Calculate cycle time

To calculate the cycle time, the read system time is stored in each cycle and subtracted from the newly read system time in the next cycle. The cycle time must not exceed 10 ms so that every high level in the signal path of the diagnostics interface "COM" can be detected. If the cycle time exceeds 10 ms, the function block outputs an error with the value 16#8001 at the output "status".

Calculating the length of a high level

To calculate the length of a high level, the system time is read and stored if a positive edge is detected at the "impulse" input. If a negative edge is subsequently detected, the system time is read and stored again. The system time stored on a positive edge is subtracted from the system time stored on a negative edge.

Calculating the length of a low level

To calculate the length of a low level, the system time is read and stored if a negative edge is detected at the "impulse" input. If a positive edge is subsequently detected, the system time is read and stored again. The system time stored on a negative edge is subtracted from the system time stored on a positive edge.

1.2.8 Internally used instructions for S7-1200 CPUs

The following instructions are used internally in the function block for the S7-1200 CPUs:

- **RD_SYS_T**: With the instruction **RD_SYS_T** you read out the current date and time (module time) of the CPU clock. In the function block the time is used to calculate the cycle time and to determine the length of the high and low levels. The **RD_SYS_T** instruction can be found in the "Instructions" task card under "Extended instructions > Date and time-of-day".
- **TON**: The instruction generates a switch-on delay. In the function block the switch-on delay is used when generating the error messages. The **TON** instruction can be found in the Task Card "Instructions" under "Basic instructions > Timer operations".
- **T_DIFF**: The statement calculates the difference between two time points by subtracting two DT format time points. The **T_DIFF** instruction can be found in the "Instructions" task card under "Extended instructions > Date and time-of-day".

Compare time values

To calculate the cycle time and the length of the high and low levels, the time read must be greater than the time read and stored in the last cycle, i.e. comparison expressions are used in the FB to compare the contents of two variables of data type "DTL" to greater.

Calculate cycle time

To calculate the cycle time, the read time is stored in each cycle and subtracted from the newly read time in the next cycle. The cycle time must not exceed 10 ms so that every high level in the signal path of the diagnostics interface "COM" can be detected. If the cycle time exceeds 10 ms, the function block outputs an error with the value 16#8001 at the output "status".

Calculate the length of an impulse

To calculate the length of a high level, the time is read and stored if a positive edge is detected at the "impulse" input. If a negative edge is subsequently detected, the time is read again and stored. The time stored on a positive edge is subtracted from the time stored on a negative edge.

Calculate the length of a pause

To calculate the length of a low level, the time is read and stored if a negative edge is detected at the "impulse" input. If a positive edge is subsequently detected, the time is read again and stored. The time stored on a negative edge is subtracted from the time stored on a positive edge.

Note

Since the time is used in the S7-1200 to calculate the cycle time and the length of the high level and low level, the output "dataPsu" must not be evaluated if the clock time is set by the time synchronization.

1.2.9 PSU6200 telegram content

Overview and Header

The PSU6200 sends 4 telegrams with the following content via the diagnostics interface "COM":

- Bit 1 to 6: Header
- Bit 2 to 13: Status information of the PSU6200
- Bit 14 to 32: Telegram-specific user content

The header and the status information of the PSU6200 are contained in all 4 telegrams. The following net capacities are transmitted with the 4 telegrams:

- Voltage and current values
- Nominal data and device information
- Device information
- Temperature and operating hours

The following figure shows the structure of the telegrams.

Figure 1-7

Bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14 through 32
	Header						Status information						Telegram-specific user content	
	Device type ¹⁾		R ²⁾	Content type										
Telegram 1	0	1	1	0	0	0	x	x	x	x	x	x	x	Voltage and Current values
Telegram 2	0	1	1	0	0	1	x	x	x	x	x	x	x	Nominal data and device information
Telegram 3	0	1	1	0	1	0	x	x	x	x	x	x	x	Device information
Telegram 4	0	1	1	0	1	1	x	x	x	x	x	x	x	Temperature and operating hours

¹⁾ Device type: PSU6200

²⁾ R: Reserved

Status information

The following table shows the contents of the status information.

Table 1-1

Bit	Meaning	Value range (binary)
1 to 6	Header	See Figure 1-7
7	DC Okay	0: Output voltage too low 1: Output voltage ok
8	Service life	0: Operating hours < 90% of the nominal value 1: Device age > 90 % of nominal value
9 to 10	Utilization	00 _{bin} = 0 _{dez} : Utilization by output current < 30 % 01 _{bin} = 1 _{dez} : Capacity utilization by output current 30 % to 60 % 10 _{bin} = 2 _{dez} : Capacity utilization by output current 61 % to 90 % 11 _{bin} = 3 _{dez} : Utilization by output current > 90 %
11	Overtemperature shutdown	0: Device in normal operation 1: Device switched off saving due to overtemperature
12	Position of the DIP switch "HV" (DC voltage level increase)	0: OK threshold low 1: OK threshold high
13	Position of the DIP switch "PO" Parallel operation	0: Stand-alone operation 1: Parallel operation
14 through 32	Telegram-specific user content	

Useful content of telegram 1

In telegram 1, current and voltage values are transmitted. This telegram is identified in the header by the content type 00_{bin} = 0_{dez}.

The following table shows the user content of telegram 1.

Table 1-2

Bit	Meaning	Value range (binary)
1 to 6	Header	See Figure 1-7
7 to 13	Status information	See Table 1-1
14 to 23	Current output voltage	0 to 102.3 V
24 to 30	Current output voltage	0 to 63.5 A
31	Reserved	0
32	Parity	The parity bit serves as a supplementary bit to supplement the number of bits occupied by 1 (including the parity bit) in the telegram as even or odd. 0: Parity sum even 1: Parity sum uneven

Useful content of telegram 2

In telegram 2, nominal and device data are transmitted. This telegram is identified in the header by the content type $01_{\text{bin}} = 1_{\text{dez}}$.

Table 1-3

Bit	Meaning	Value range (binary)
1 to 6	Header	See Figure 1-7
7 to 13	Status information	See Table 1-1
14 to 15	Family property	$00_{\text{bin}} = 0_{\text{dez}}$: AC/DC $01_{\text{bin}} = 1_{\text{dez}}$: DC/DC $10_{\text{bin}} = 2_{\text{dez}}$: BAT $11_{\text{bin}} = 3_{\text{dez}}$: Reserve
16 to 18	Nominal output voltage	$000_{\text{bin}} = 0_{\text{dez}}$: 12 V $001_{\text{bin}} = 1_{\text{dez}}$: 24 V $010_{\text{bin}} = 2_{\text{dez}}$: Reserve $011_{\text{bin}} = 3_{\text{dez}}$: 48 V $100_{\text{bin}} = 4_{\text{dez}}$: Reserve $101_{\text{bin}} = 5_{\text{dez}}$: Reserve $110_{\text{bin}} = 6_{\text{dez}}$: Reserve $111_{\text{bin}} = 7_{\text{dez}}$: Reserve
19 to 21	Nominal output current	$000_{\text{bin}} = 0_{\text{dez}}$: 5 A $001_{\text{bin}} = 1_{\text{dez}}$: 10 A $010_{\text{bin}} = 2_{\text{dez}}$: 20 A $011_{\text{bin}} = 3_{\text{dez}}$: 40 A $100_{\text{bin}} = 4_{\text{dez}}$: 12 A $101_{\text{bin}} = 5_{\text{dez}}$: Reserve $011_{\text{bin}} = 6_{\text{dez}}$: Reserve $111_{\text{bin}} = 7_{\text{dez}}$: Reserve
22	Input phases	0: 1-phase device 1: 3-phase device
23 to 26	Year of manufacture	0 to 15 years from 2019
27 to 29	Product version	0 to 7
30	Coating	0: Circuit boards not coated 1: Circuit boards coated
31	Manufacturer site	0: Manufacturer site Vienna 1: Manufacturer site Sibiu
32	Parity	The parity bit serves as a supplementary bit to supplement the number of bits occupied by 1 (including the parity bit) in the telegram as even or odd. 0: Parity sum even 1: Parity sum uneven

Useful content of telegram 3

In telegram 3, device data is transmitted. This telegram is identified in the header by the content type $10_{\text{bin}} = 2_{\text{dez}}$.

Table 1-4

Bit	Meaning	Value range (binary)
1 to 6	Header	See Figure 1-7
7 to 13	Status information	See Table 1-1
14 to 17	Month of construction	$0000_{\text{bin}} = 0_{\text{dez}}$: Reserve $0001_{\text{bin}} = 1_{\text{dez}}$: January $0010_{\text{bin}} = 2_{\text{dez}}$: February $0011_{\text{bin}} = 3_{\text{dez}}$: March $0100_{\text{bin}} = 4_{\text{dez}}$: April $0101_{\text{bin}} = 5_{\text{dez}}$: May $0110_{\text{bin}} = 6_{\text{dez}}$: June $0111_{\text{bin}} = 7_{\text{dez}}$: July $1000_{\text{bin}} = 8_{\text{dez}}$: August $1001_{\text{bin}} = 9_{\text{dez}}$: September $1010_{\text{bin}} = 10_{\text{dez}}$: October $1011_{\text{bin}} = 11_{\text{dez}}$: November $1100_{\text{bin}} = 12_{\text{dez}}$: December $1101_{\text{bin}} = 13_{\text{dez}}$: Reserve $1110_{\text{bin}} = 14_{\text{dez}}$: Reserve $1111_{\text{bin}} = 15_{\text{dez}}$: Reserve
18 to 20	Software version	0 to 7
21 to 23	Device version: Digit 11 of the article number	0 to 7
24 to 26	Device innovations Digit 12 of the article number	0 to 7
27	AC/DC or battery	0: Power supply with AC or DC input 1: Battery
28 to 30	Variants (Number of channels) Digit 14 of the article number	$000_{\text{bin}} = 0_{\text{dez}}$: A (one channel) $001_{\text{bin}} = 1_{\text{dez}}$: B (two channels) $010_{\text{bin}} = 2_{\text{dez}}$: C (four channels) $011_{\text{bin}} = 3_{\text{dez}}$: D (eight channels) $100_{\text{bin}} = 4_{\text{dez}}$: X (Not relevant) $101_{\text{bin}} = 5_{\text{dez}}$: Reserve $110_{\text{bin}} = 6_{\text{dez}}$: Reserve $111_{\text{bin}} = 7_{\text{dez}}$: Reserve
31	Variants (type of terminals): Digit 15 of the article number	0: Y (Standard screw terminals) 1: X (Standard spring-loaded terminals / Push In)
32	Parity	The parity bit serves as a supplementary bit to supplement the number of bits occupied by 1 (including the parity bit) in the telegram as even or odd. 0: Parity sum even 1: Parity sum uneven

Useful content of telegram 4

The temperature and operating time are transmitted in telegram 4. This telegram is identified in the header by the content type $11_{\text{bin}} = 3_{\text{dez}}$

Table 1-5

Bit	Meaning	Value range (binary)
1 to 6	Header	See Figure 1-7
7 to 13	Status information	See Table 1-1
14 to 18	Reserve	0
19 to 20	Temperature status	00 _{bin} = 0 _{dez} : Low 01 _{bin} = 1 _{dez} : High 10 _{bin} = 2 _{dez} : derating 11 _{bin} = 3 _{dez} : Device switched off due to overtemperature in the storage tank
21	Reserve	0
22	Phase error	0: Normal operation 1: Phase error present
23	Glitch overvoltage	1: Overvoltage glitch 2-200 ms occurred at output since previous transmission of bit
24	Glitch undervoltage	1: Undervoltage glitch 2-200 ms at output occurred since previous transmission of bit
25 to 31	Operating hours	0-127000 hours
32	Parity	The parity bit serves as a supplementary bit to supplement the number of bits occupied by 1 (including the parity bit) in the telegram as even or odd. 0: Parity sum even 1: Parity sum uneven

1.3 Components used

This application example was created with these hardware and software components.

Table 1-6

Component	Quantity	Article number	Alternative
S7-1500			
CPU 1513-1 PN	1	6ES7513-1AL01-0AB0	Any S7-1500 CPU, ET 200SP CPU and ET 200pro CPU
Digital input module DI 32x24V DC HF	1	6ES7-1BL00-0AB0	Other digital input modules or digital inputs of a decentralized periphery
S7-1200			
CPU 1211C	1	6ES7211-1AE40-0XB0	Any S7-1200 CPU
S7-300			
CPU 315-2 PN/DP	1	6ES7315-2EH14-0AB0	<ul style="list-style-type: none"> Any S7-300 / S7-400 CPU IM 151-8(F) PN/DP CPU IM 154-8(F/FX) PN/DP CPU
DI 8/DO 8x24VDC/0,5A	1	6ES7323-1BH01-0AA0	Other digital input modules and digital output modules or digital inputs and digital outputs of a decentralized periphery.
HMI operator panel			
TP700 Comfort	1	6AV2124-0GC01-0AX0	All Comfort Panel ≥ 7" and PC station with WinCC RT Advanced ≥ V15 also usable
Engineering software			
STEP 7 V15	1	6ES7822-1AA05-0YA5	For the project planning of the S7-CPU you need STEP 7 Professional V15 or higher.
WinCC Engineering V15	1	6AV210.-...5-0	
SIMATIC WinCC Advanced V15	1	6AV2101-0AA05-0AA5	Package V15 Download license also possible
SIMATIC WinCC Runtime Advanced 4096 PowerTags (V15)	1	6AV2104-0HA05-0AA0	Package V15 Download license also possible

2 Engineering

2.1 Interface description

2.1.1 PLC data type for S7-1200 and S7-1500

The following PLC data types are predefined data structures in which the user contents of the telegrams are stored.

- "typeDataS71500PSU6200"
- "typeDataS71200PSU6200"

The following table shows the structure of the PLC data types.

Table 2-1

Parameters	Data type	Start value	Description
deviceName	String	PSU6200	Device name
stateOutputVoltageOK	Bool	FALSE	FALSE: Output voltage too low TRUE: Output voltage ok
stateOperatingHours	Bool	FALSE	FALSE: Operating hours < 90 % of the nominal value TRUE: Operating hours > 90 % of the nominal value
outputCurrentLoad	bytes	16#01	16#00: Utilization < 30 % of nominal output current 16#01: Utilization 30 % to 60 % of nominal output current 16#02: Utilization 61 % to 90 % of nominal output current 16#03: Utilization > 90 % of nominal output current
overTemperatureShutdown	Bool	TRUE	FALSE: Device in normal operation TRUE: Device switched off saving due to overtemperature
stateCOM	Bool	TRUE	DIP switch "COM" TRUE: Diagnostics interface "COM" activated FALSE: Diagnostics interface "COM" deactivated
stateHV	Bool	FALSE	DIP switch "HV" FALSE: Threshold low TRUE: Threshold high
statePO	Bool	FALSE	DIP switch "PO" FALSE: Stand-alone operation TRUE: Parallel operation
outputVoltage	Real	0.0	Output voltage
outputCurrent	Int	0	Output current

Parameters	Data type	Start value	Description
nominalOutputVoltage	Int	24	Nominal output voltage
nominalOutputCurrent	Int	10	Nominal output current
productState	Int	1	Product version
firmwareVersion	Int	1	Firmware version
stateDeviceTemperature	Int	0	0: Device temperature normal 1: Device temperature high 2: Device temperature critical 3: Overtemperature
phaseError	Bool	0	FALSE: no phase error TRUE: Phase error Note Only for 3-phase devices
operatingHours	DInt	1000	Operating hours in 1000 h Value range: 0-127000 h
articleNumber	String	6EP3333-7SB00-0AX0	Article number
serialNumber	String	Q6/L1	Part of serial number <ul style="list-style-type: none"> • Manufacturer site • Year of manufacture • Month of construction
counterGlitchOvervoltage	Int	0	Counter interference pulse overvoltage (output voltage).
counterGlitchUndervoltage	Int	0	Counter interference pulse undervoltage (output voltage)
stateNoCommunication	Bool	0	FALSE: Communication via the diagnostic interface "COM" is active. TRUE: Communication via the diagnostic interface "COM" is not active.

2.1.2 PLC data type and HMI data type for S7-300 / S7-400

The PLC data type "typeDataS7300PSU6200" is a predefined data structures in which the user contents of the telegrams are stored.

The faceplates "fpComAdvDetailS7300" und "fpComAdvOverviewS7300" use the HMI data type "typeDataS7-300PSU6200HMI" to access the data of the PLC data type "typeDataS7300PSU6200" and show the data of the PLC data type on HMI operator device.

The PLC data type "typeDataS7300PSU6200" and the HMI data type "typeDataS7300PSU6200HMI" have the same structure.

Using an HMI connection for the absolute from the HMI data type to the PCL data type in S7-300 / S7-400 CPU.

The following table shows the structure of the PLC data type "typeDataS7300PSU6200" and HMI data type "typeDataS7300PSU6200HMI".

Table 2-2

Parameters	Data type	Offset	Start value	Description
deviceName	String	0.0	PSU6200	Device name
stateOutputVoltageOK	Bool	256.0	FALSE	FALSE: Output voltage too low TRUE: Output voltage ok
stateOperatingHours	Bool	256.1	FALSE	FALSE: Operating hours < 90 % of the nominal value TRUE: Operating hours > 90 % of the nominal value
outputCurrentLoad	bytes	257.0	16#01	16#00: Utilization < 30 % of nominal output current 16#01: Utilization 30 % to 60 % of nominal output current 16#02: Utilization 61 % to 90 % of nominal output current 16#03: Utilization > 90 % of nominal output current
overTemperatureShutdown	Bool	258.0	TRUE	FALSE: Device in normal operation TRUE: Device switched off saving due to overtemperature
stateCOM	Bool	258.1	TRUE	DIP switch "COM" TRUE: Diagnostics interface "COM" activated FALSE: Diagnostics interface "COM" deactivated
stateHV	Bool	258.2	FALSE	DIP switch "HV" FALSE: Threshold low TRUE: Threshold high

Parameters	Data type	Offset	Start value	Description
statePO	Bool	258.3	FALSE	DIP switch "PO" FALSE: Stand-alone operation TRUE: Parallel operation
outputVoltage	Real	260.0	0.0	Output voltage
outputCurrent	Int	264.0	0	Output current
nominalOutputVoltage	Int	266.0	24	Nominal output voltage
nominalOutputCurrent	Int	268.0	10	Nominal output current
productState	Int	270.0	1	Product version
firmwareVersion	Int	272.0	1	Firmware version
stateDeviceTemperature	Int	274.0	0	0: Device temperature normal 1: Device temperature high 2: Device temperature critical 3: Overtemperature
phaseError	Bool	276.0	0	FALSE: no phase error TRUE: Phase error Note Only for 3-phase devices
operatingHours	DInt	278.0	1000	Operating hours in 1000 h Value range: 0-127000 h
articleNumber	String	282.0	6EP3333-7SB00-0AX0	Article number
serialNumber	String	538.0	Q6/L1	Part of serial number <ul style="list-style-type: none"> Manufacturer site Year of manufacture Month of construction
counterGlitchOvervoltage	Int	794.0	0	Counter interference pulse overvoltage (output voltage).
counterGlitchUndervoltage	Int	796.0	0	Counter interference pulse undervoltage (output voltage)
stateNoCommunication	Bool	798.0	0	FALSE: Communication via the diagnostic interface "COM" is active. TRUE: Communication via the diagnostic interface "COM" is not active.

2.1.3 FB "LPSU6200_SignalEvaluationS71500"

The FB "LPSU6200_SignalEvaluationS71500" reads the signal of the diagnostics interface "COM" at the input "impulse" in order to evaluate the signal course and to store the contents of the 4 telegrams in a predefined data structure at the output "dataPsu".

The [Figure 2-1](#) shows the call of the FB "LPSU6200_SignalEvaluationS71500" in the user program.

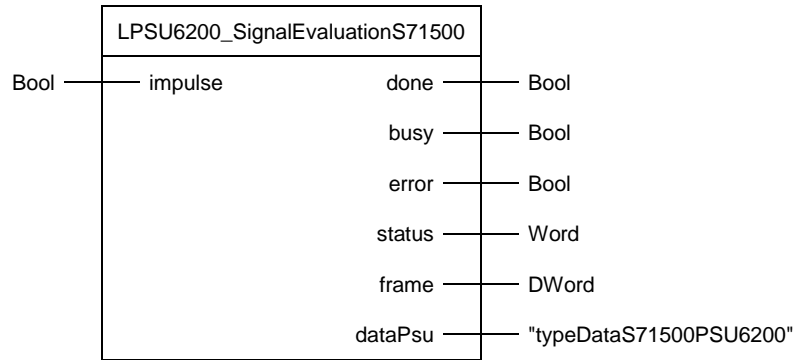
Call up the FB "LPSU6200_SignalEvaluationS71500" in the user program of the S7-CPU in a cyclic interrupt OB with a maximum of 10 ms.

The cycle time must not exceed 10 ms so that every high level in the signal path of the diagnostics interface "COM" is detected.

Note

If the cycle time exceeds 10 ms, the FB "LPSU6200_SignalEvaluationS71500" outputs an error with the value 16#8001 at the output "status".

Figure 2-1



The following table shows the parameters of the FB "LPSU6200_SignalEvaluationS71500".

Table 2-3

Name	P type	Data type	Comment
impulse	IN	Bool	Input via which the signal of the diagnostics interface "COM" is read in.
done	OUT	Bool	1: A telegram was evaluated completely and without error. The data are stored in a predefined data structure at the output "dataPsu". The data of the output "dataPsu" can be transferred. The output "done" is set to the value "1" for one cycle. 0: Telegram evaluation is running, or no signal detected at input "impulse".
busy	OUT	Bool	1: FB "LPSU6200_SignalEvaluationS71500" is active 0: If done = 1, then a telegram was evaluated completely and without error. The content of the telegram is stored in a predefined data structure at the output "psuData". The data of the output "psuData" can be transferred.
error	OUT	Bool	1: An error occurred during the processing of the routine. The output "error" is set to the value "1" for one cycle. 0: No error
status	OUT	Word	Status display If error = 1, the "status" output indicates the error number for a cycle. If error = 0, the value 16#0000 is displayed at the "status" output.
frame	OUT	DWord	Binary code of the received telegram.
dataPsu	OUT	"typeDataS71500PSU6200"	Predefined data structure in which the status information and the user contents of the telegrams are stored. Detailed information on the structure of the predefined data structure can be found in the section 2.1.1 .

Note

The instance DB of the FB "LPSU6200_SignalEvaluationS71500" is generated when the FB "LPSU6200_SignalEvaluationS71500" is called. For each call of the FB "LPSU6200_SignalEvaluationS71500" you need a separate instance DB. The FB "LPSU6200_SignalEvaluationS71500" must not be called several times with the same instance DB.

2.1.4 FB "LPSU6200_SignalEvaluationS71200"

The FB "LPSU6200_SignalEvaluationS71200" reads the signal of the diagnostics interface "COM" at the input "impulse" in order to evaluate the signal course and to store the contents of the 4 telegrams in a predefined data structure at the output "dataPsu".

The [Figure 2-2](#) shows the call of the FB "LPSU6200_SignalEvaluationS71200" in the user program.

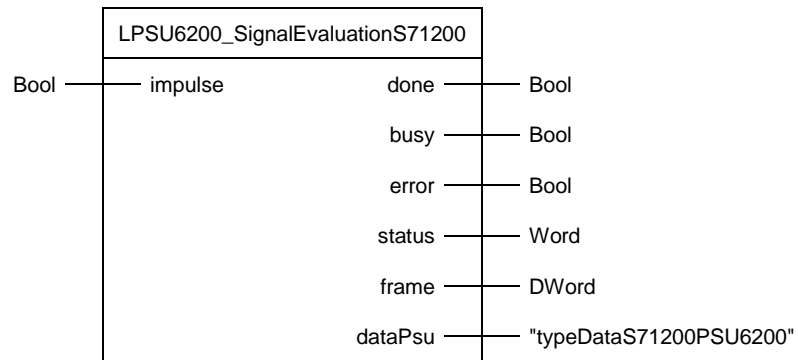
Call up the FB "LPSU6200_SignalEvaluationS71200" in the user program of the S7-CPU in a cyclic interrupt OB with a maximum of 10 ms.

The cycle time must not exceed 10 ms so that every high level in the signal path of the diagnostics interface "COM" is detected.

Note

If the cycle time exceeds 10 ms, the FB "LPSU6200_SignalEvaluationS71200" outputs an error with the value 16#8001 at the output "status".

Figure 2-2



The following table shows the parameters of
FB "LPSU6200_SignalEvaluationS71200".

Table 2-4

Name	P type	Data type	Comment
impulse	IN	Bool	Input via which the signal of the diagnostics interface COM is read in.
done	OUT	Bool	1: A telegram was evaluated completely and without errors. The data are stored in a predefined data structure at the output dataPsu". The data of the output "dataPsu" can be transferred. The output "done" is set to the value "1" for one cycle. 0: Telegram evaluation is running, or no signal detected at input "impulse".
busy	OUT	Bool	1: FB "LPSU6200_SignalEvaluationS71200" is active 0: If done = 1, then a telegram was evaluated completely and without error. The content of the telegram is stored in a predefined data structure at the output "psuData". The data of the output "psuData" can be transferred.
error	OUT	Bool	1: An error occurred during the processing of the routine. The output "error" is set to the value "1" for one cycle. 0: No error
status	OUT	Word	Status display If error = 1, the "status" output indicates the error number for a cycle. If error = 0, the value 16#0000 is displayed at the "status" output.
frame	OUT	DWord	Binary code of the received telegram.
dataPsu	OUT	"typeDataS71200PSU6200"	Predefined data structure in which the status information and the user contents of the telegrams are stored. Detailed information on the structure of the predefined data structure can be found in the section 2.1.1 .

Note

The instance DB of the FB "LPSU6200_SignalEvaluationS71200" is generated when the FB "LPSU6200_SignalEvaluationS71200" is called. For each call of the FB "LPSU6200_SignalEvaluationS71200" you need a separate instance DB. The FB "LPSU6200_SignalEvaluationS71200" must not be called several times with the same instance DB.

2.1.5 FB "LPSU6200_SignalEvaluationS7300"

The FB "LPSU6200_SignalEvaluationS7300" reads the signal of the diagnostics interface "COM" at the input "impulse" in order to evaluate the signal course and to store the contents of the 4 telegrams in a predefined data structure at the output "dataPsu".

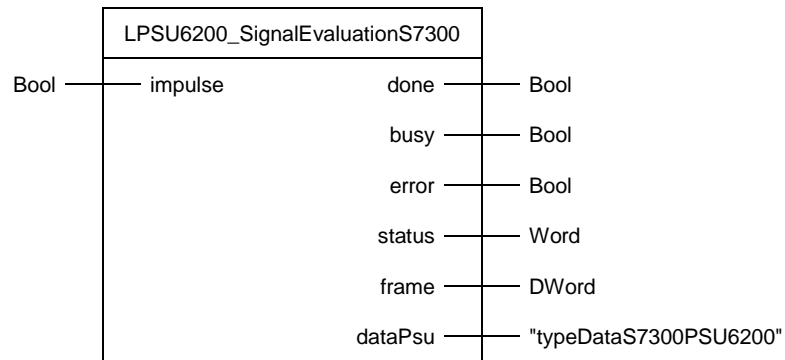
The [Figure 2-3](#) shows the call of the FB "LPSU6200_SignalEvaluationS7300" in the user program.

Call up the FB "LPSU6200_SignalEvaluationS7300" in the user program of the S7-CPU in a cyclic interrupt OB with a maximum of 10 ms.

The cycle time must not exceed 10 ms so that every high level in the signal path of the diagnostics interface "COM" is detected.

NOTE If the cycle time exceeds 10 ms, the FB "LPSU6200_SignalEvaluationS7300" outputs an error with the value 16#8001 at the output "status".

Figure 2-3



The following table shows the parameters of
FB "LPSU6200_SignalEvaluationS7300".

Table 2-5

Name	P-Type	Data Type	Comment
impulse	IN	Bool	Input via which the signal of the diagnostics interface "COM" is read in.
done	OUT	Bool	1: A telegram was evaluated completely and without error. The data are stored in a predefined data structure at the output "dataPsu". The data of the output "dataPsu" can be transferred. The output "done" is set to the value "1" for one cycle. 0: Telegram evaluation is running, or no signal detected at input "impulse".
busy	OUT	Bool	1: FB "LPSU6200_SignalEvaluationS7300" is active. 0: If done = 1, then a telegram was evaluated completely and without error. The content of the telegram is stored in a predefined data structure at the output "psuData". The data of the output "psuData" can be transferred.
error	OUT	Bool	1: An error occurred during the processing of the routine. The output "error" is set to the value "1" for one cycle. 0: No error
status	OUT	Word	Status display If error = 1, the "status" output indicates the error number for a cycle. If error = 0, the value 16#0000 is displayed at the "status" output.
frame	OUT	DWord	Binary code of the received telegram.
dataPsu	OUT	"typeDataS7300PSU6200"	Predefined data structure in which the status information and the user contents of the telegrams are stored. Detailed information on the structure of the predefined data structure can be found in the section 2.1.2 .

NOTE

The instance DB of the FB "LPSU6200_SignalEvaluationS7300" is generated when the FB "LPSU6200_SignalEvaluationS7300" is called. For each call of the FB "LPSU6200_SignalEvaluationS7300" you need a separate instance DB. The FB "LPSU6200_SignalEvaluationS7300" must not be called several times with the same instance DB.

2.1.6 Faceplate "fpComAdvDetailS7xyPSU6200"

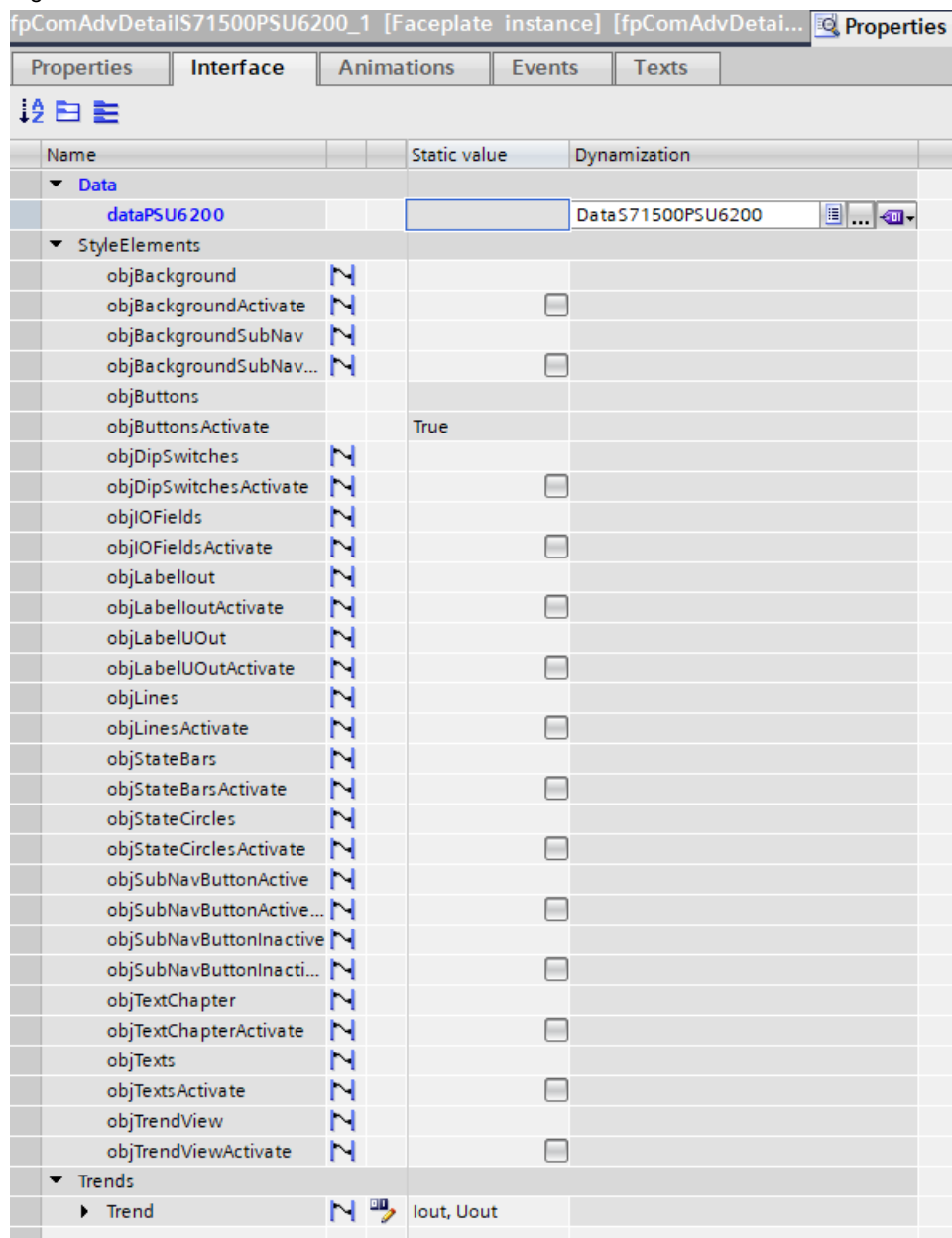
The following faceplates provide different categories with different properties for parameterization via the interface:

- "fpComAdvDetailS71500PSU6200"
- "fpComAdvDetailS71200PSU6200"
- "fpComAdvDetailS7300PSU6200"

The categories fulfil the following purposes:

- Data: Property for the variable binding
- StyleElements: Properties for changing the style of the screen block
- Trends: Properties for parameterizing the trends for the trend display

Figure 2-4



2.1.7 Faceplates "fpComAdvOverviewS7xy00PSU6200"

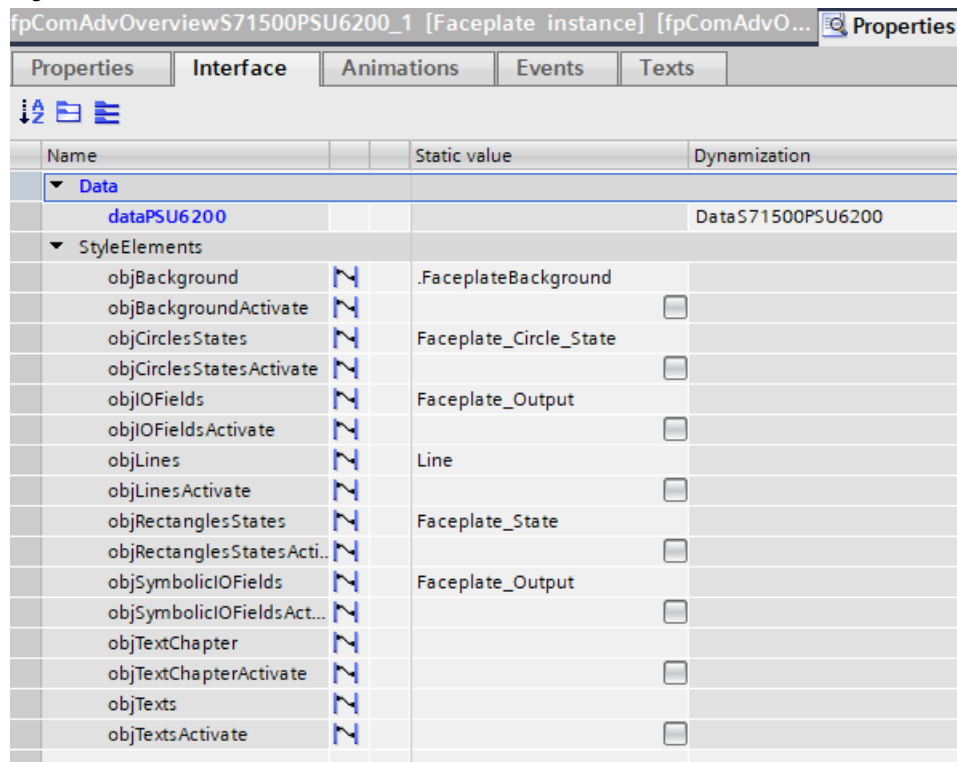
The following faceplates provide different categories with different properties for parameterization via the interface:

- "fpComAdvOverviewS71500PSU6200"
- "fpComAdvOverviewS71200PSU6200"
- "fpComAdvOverviewS7300PSU6200"

The categories fulfil the following purposes:

- Data: Property for the variable binding
- StyleElements: Properties for changing the style of the screen block

Figure 2-5

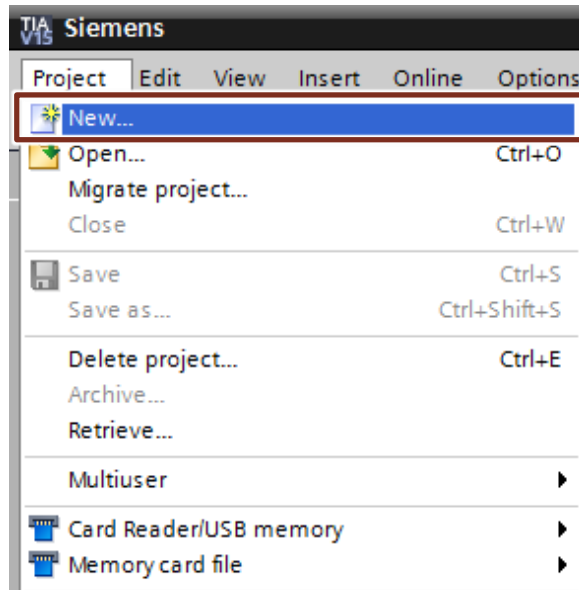


2.2 Projecting Facilities

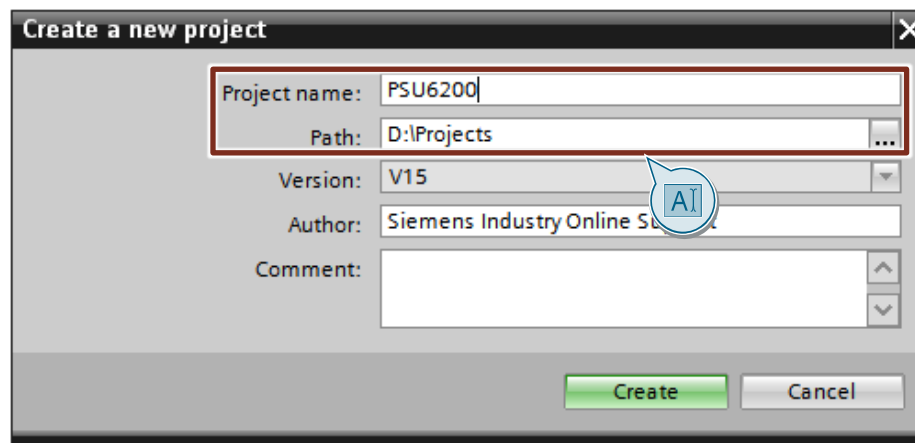
2.2.1 Creating a TIA Portal project

1. Open the TIA Portal.
2. Switch to the "Project view".

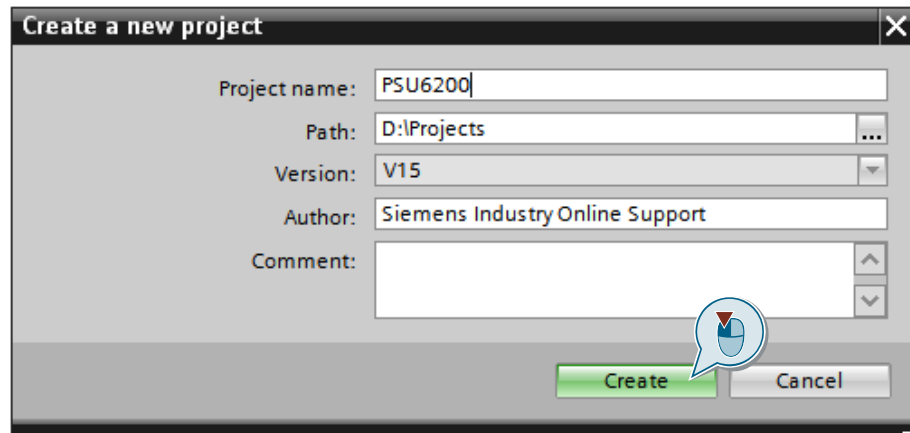
From the "Project" menu, select "New".
The "Create a new project" dialog opens.



3. Enter your project name and path or accept the proposed data.



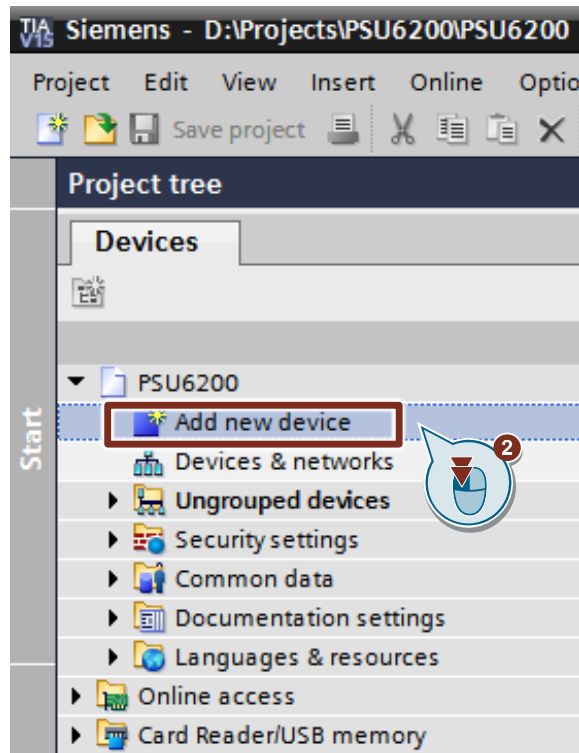
4. Click on the "Create" button. The new project is created and displayed in the project tree.



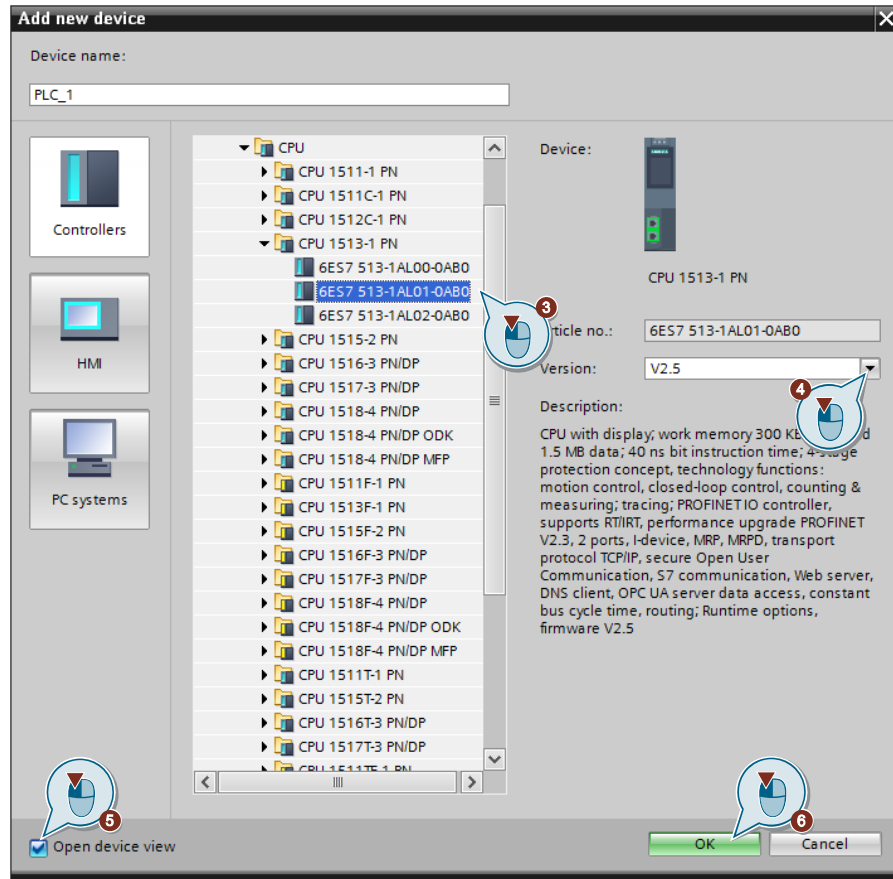
2.2.2 Integration of a S7-CPU into the user project

Add S7-CPU via hardware catalog

1. Open the "Project tree".
2. Double-click the "Add new device" command to add a new PLC device.
The dialog box "Add new device" opens.

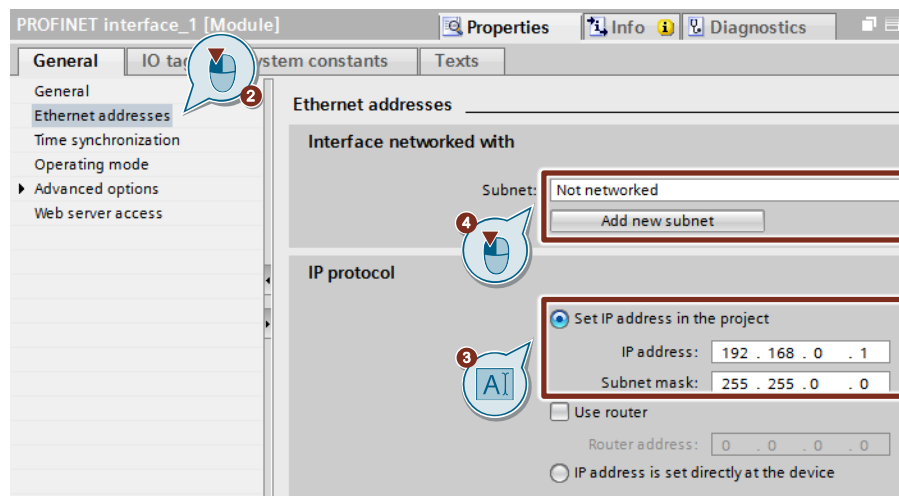


3. Under "Controllers", select the part number of the desired S7 CPU.
4. Set the firmware version via the drop-down list of the dialog.
5. Activate the option box "Open device view".
6. Click the "OK" button to add the selected S7 CPU.
The dialog box closes and the "Device view" of the S7 CPU opens in the hardware and network editor.



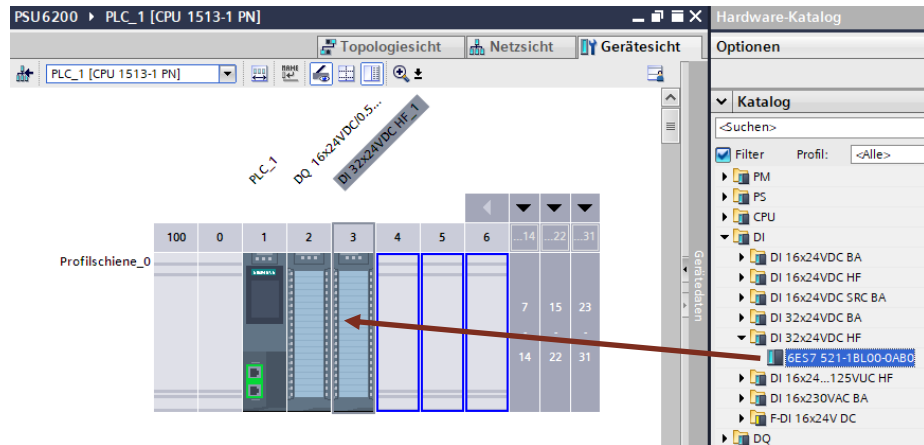
Adjust hardware configuration for S7-CPU

1. Mark the S7-CPU in the graphical area of the device view. The "Properties" of the S7-CPU are displayed in the inspector window.
2. In the "General" tab, navigate to "PROFINET interface [X1] > Ethernet addresses".
3. Enter the IP address and subnet mask of the S7 CPU.
4. Click on the "Add new subnet" button or select an existing subnet from the drop-down list.



Integration of a digital input module into the user project

1. Open the device view of the S7 CPU or the decentralized periphery, e.g. ET 200SP.
2. Select the desired digital input module in the "Hardware catalog".
3. Paste the selected digital input module via drag & drop into the slot where it is located in the actual hardware setup.



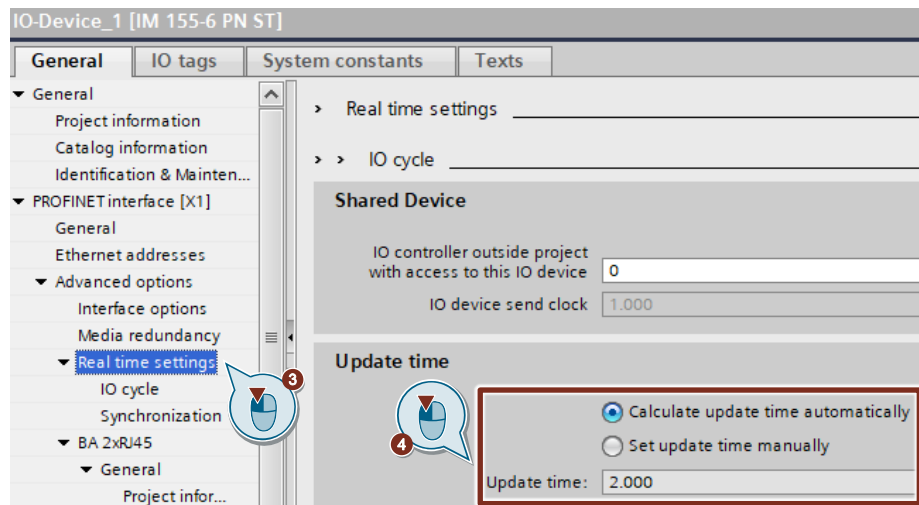
4. The preset address of the digital inputs is displayed in the table area of the device view. You can change it if necessary.

Device overview							
Module	Rack	Slot	I address	Q address	Type		
	0	100					
	0	0					
PLC_1	0	1			CPU 1513-1 PN		
PROFINET-Schnittstelle 1	0	1 X1			PROFINET interface		
DI 16x24VDC HF_1	0	2	0...1		DI 16x24VDC HF		
	0	3					

Update the Update time

If you read in the diagnostics interface "COM" of the PSU6200 via a digital input module of the decentralized periphery, e.g. ET 200SP, the update time must not exceed 16 ms. You can have the update time calculated automatically or set it manually:

1. Open the device view of the decentralized periphery, e.g. ET 200SP.
2. In the graphical area of the device view, select the Interface Module (IM) of the ET 200SP. The "Properties" of the decentralized periphery are displayed in the inspector window.
3. Navigate in the "General" tab to "PROFINET Interface [X1] > Advanced Options > Real-Time Settings".
4. Set the update time (max. 16 ms).

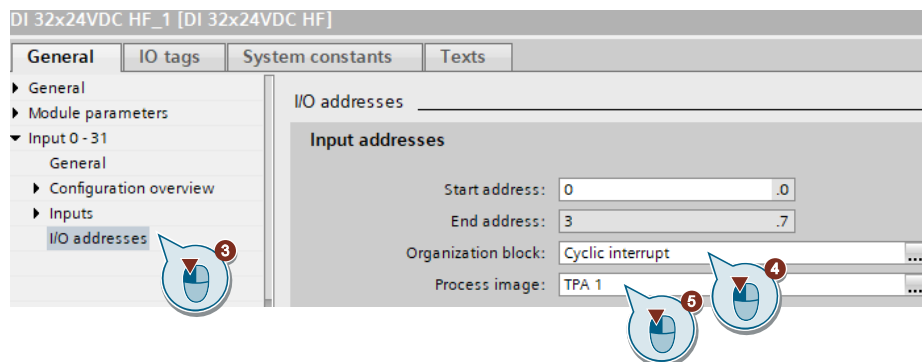


Assign the sub-process image (TPA) of the digital input module to the cyclic interrupt OB

The process image contains the image of the digital input module and digital output module and is divided accordingly into an input process image and an output process image. The process image can be divided into individual sub-process images that can be updated either automatically or by the user program.

An input is the image of the corresponding bit on the digital input module. The interrogation of an input is equivalent to the interrogation of the bit directly on the module. After the CPU start-up and before the first processing of OB 1, the operating system transfers the signal states of the digital input module into the input process image. Then the OB 1 is processed. Following termination of OB1, a new cycle begins with updating of the process image. Since the OB 1 cycle can be very long, depending on the user program, and can be interrupted several times by the cyclic interrupt OB, you must assign a sub-process image of the digital input module to the cyclic interrupt OB. Thus, the sub-process image is automatically updated when the cyclic interrupt OB is called.

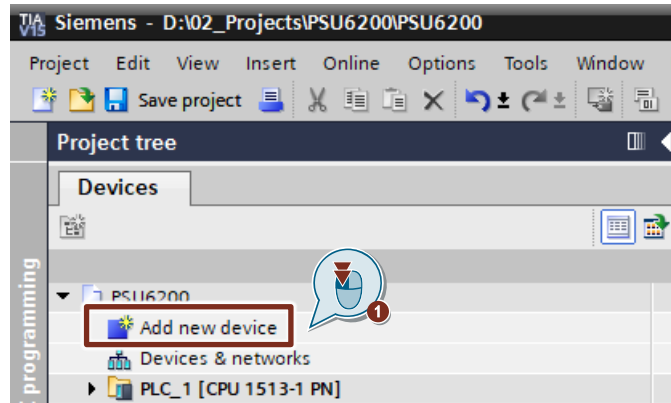
1. Open the device view of the S7 CPU or the decentralized periphery, e.g. ET 200SP.
2. Select the digital input module in the graphical area of the device view. The "Properties" of the digital input module are displayed in the Inspector window.
3. In the "General" tab, navigate to "Input 0 - x > I/O addresses".
4. Select the cyclic interrupt OB under "Organization block".
5. Under "Process image", select the desired sub-process image, e.g. TPA 1.



2.2.3 Integration of an HMI operating device

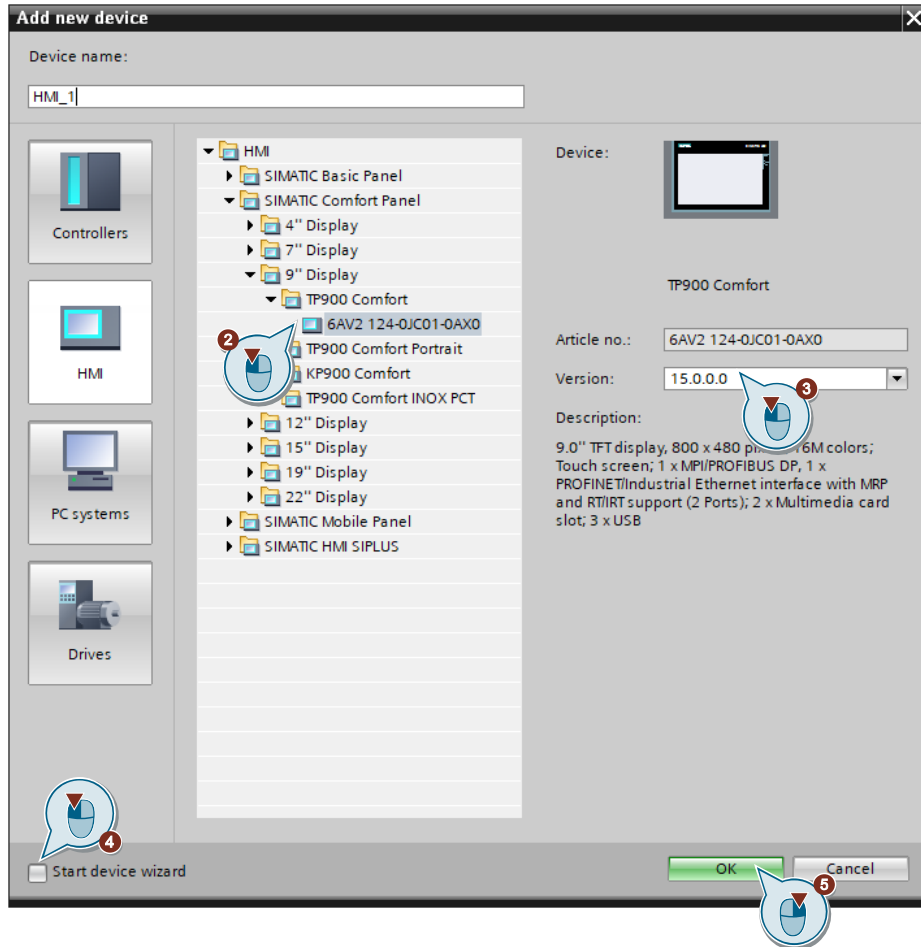
Adding an HMI operating device

1. Add a new device to your project via "Project tree > Add new device".
The dialog box "Add new device" opens.



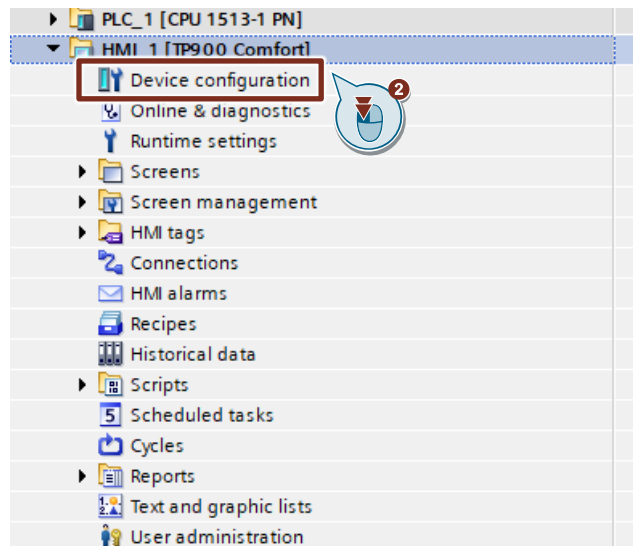
2. Select the desired HMI operating device under "HMI" and assign a name.
3. **Optional**
If necessary, set the software version via the drop-down list of the dialog.
4. Deactivate the check box "Start device wizard".

- Click on the "OK" button to add the selected HMI operating device. The dialog box closes, and the inserted HMI operating device is displayed in the project tree.

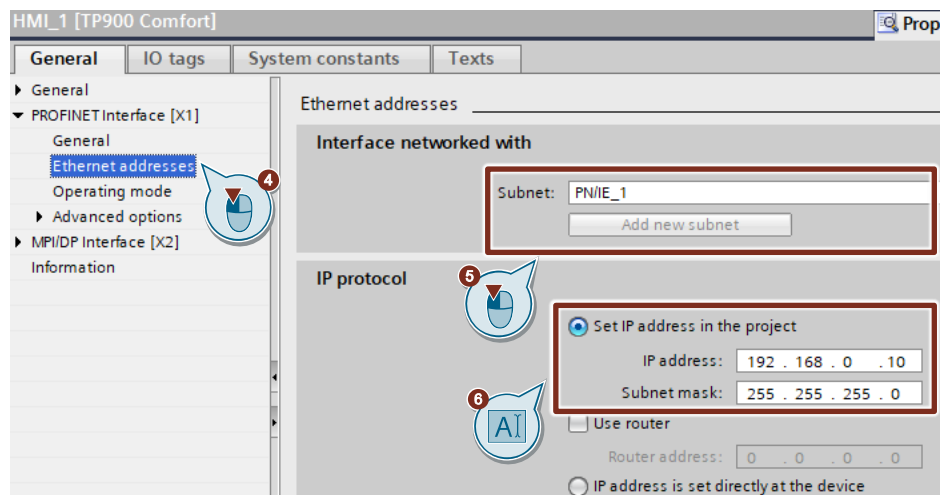


Configuring the HMI operating device

1. Use the "Project tree" to navigate to the device folder of the HMI operating device.
2. Double-click on the "Device configuration" command to open the hardware and network editor in the "Device view" of the HMI operating device.



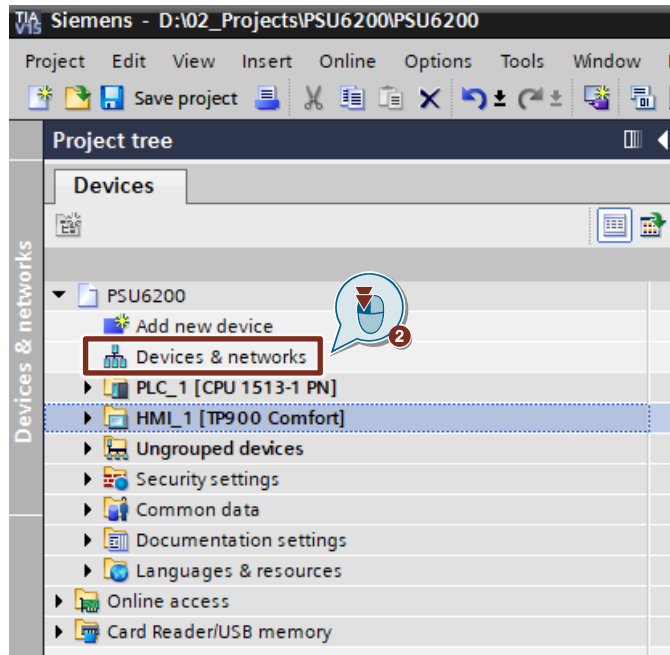
3. Mark the HMI operating device in the graphical area of the "Device view". The "Properties" of the HMI operating device are displayed in the Inspector window.
4. In the "General" tab, navigate to "PROFINET interface [X1] > Ethernet addresses".
5. Enter the IP address and subnet mask of the HMI device.
6. Use the drop-down list to select the subnet to which the S7-CPU is connected.



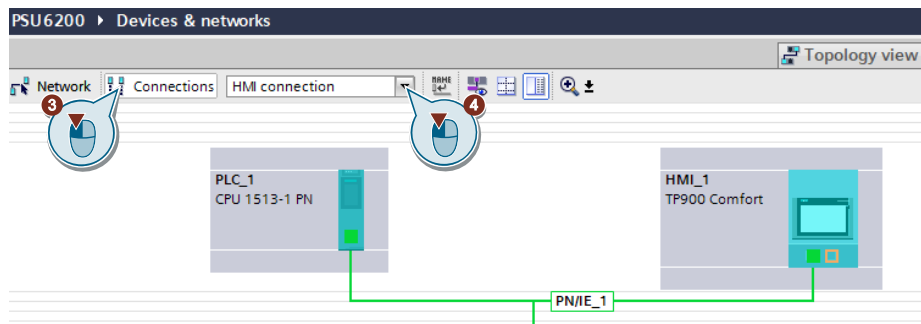
2.3 Creating communication

2.3.1 Creating an HMI connection

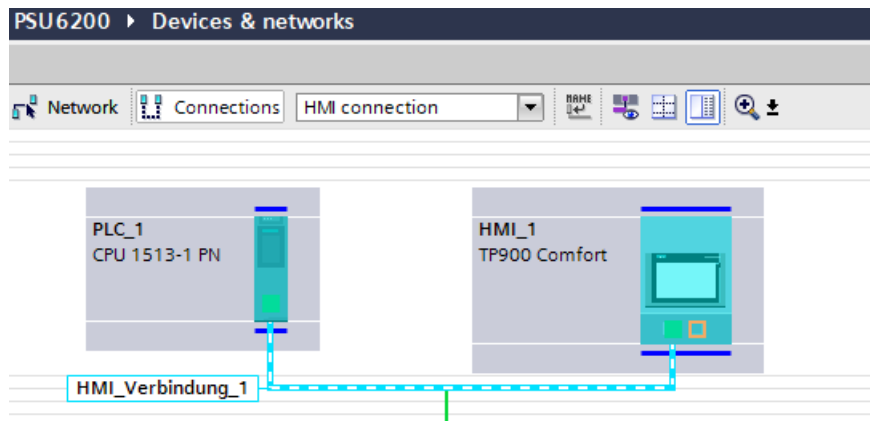
1. Open the "Project tree".
2. Double-click the Devices & Networks command to open the Network view graphical area.



3. In the toolbar, click the "Connections" icon. This enables the connection mode.
4. Select the connection "HMI connection" in the adjacent drop-down list. In the network view, all S7 CPUs and HMI devices that can be used for an HMI connection are highlighted in color.



5. You can now have the connection path automatically defined or explicitly define a connection path via specific interfaces:
 - Having connection path defined automatically:
Select the CPU from which a connection is to be set up. Move the mouse to the target component. Confirm the connection end point with another mouse click.
 - Select explicit connection path from interface to interface:
Click on the subnet interface in the device from which you want to start a connection. Then move the mouse pointer while holding the mouse button down to the desired interface in the target device and release the mouse button.
6. In the network view, the created HMI connection is displayed in the graphical and tabular area (tab "Connections").



The screenshot shows the 'Connections' tab in the 'Devices & networks' window. It displays a table with the following data:

Local connection name	Local end point	Local ID (hex)	Partner ID (hex)	Partner	Connection type
HMI_Verbindung_1	HMI			PLC_1 [CPU 1513-1 PN]	HMI connection

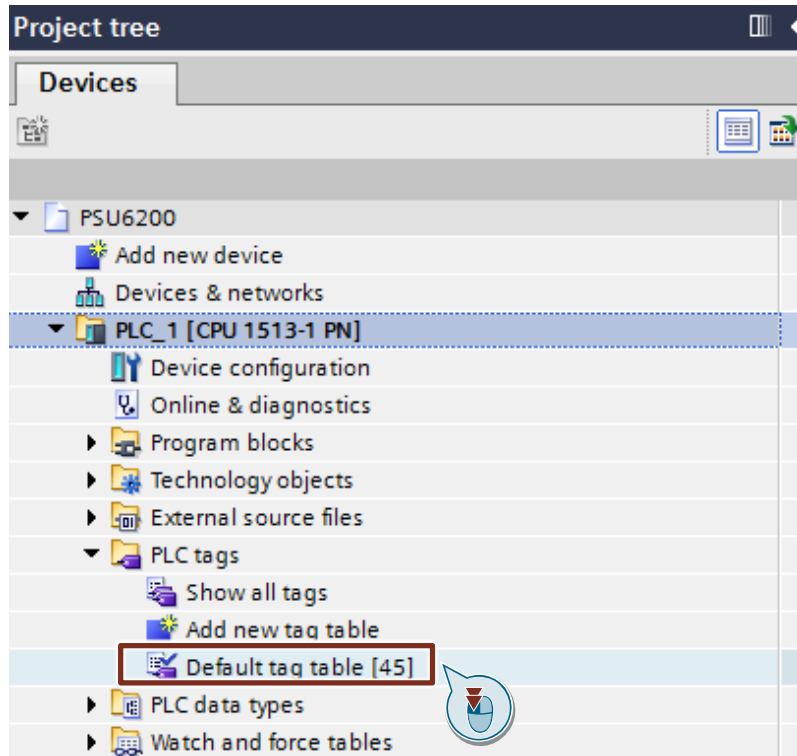
7. The connection settings are displayed in the Inspector window. If necessary, change the name of the connection (e.g. "S7_1500").

The screenshot shows the 'Inspector' window for the 'HMI_Verbindung_1' connection. It displays the following configuration details:

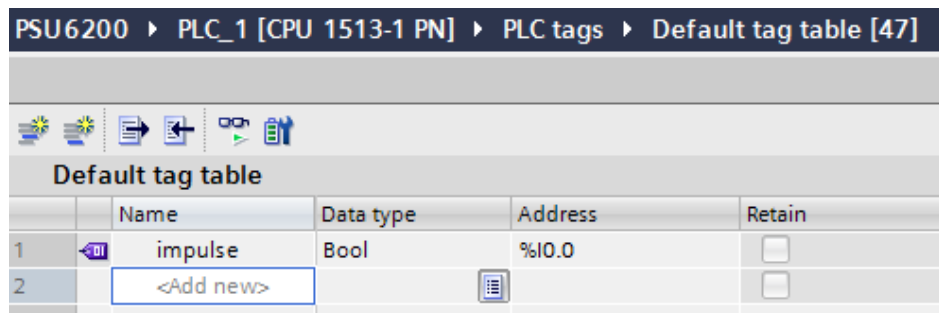
- Name:** HMI_Verbindung_1
- Connection path:** Local (HMI_1) to Partner (PLC_1 [CPU 1513-1 PN])
- End point:** HMI_1
- Interface:** HMI_1 IE_CP_1, PROFINET Schnittstelle_1[X1]
- Interface type:** Ethernet
- Subnet:** PNIE_1
- Address:** 192.168.0.10

2.3.2 Creating PLC variables

1. Navigate in the "Project tree" to the device folder of the S7-CPU.
2. In the "PLC tags" folder, double-click on the "Default tag table".
The default tag table opens in the work area.



3. In the column "Address" enter the address of the digital input via which the diagnostics interface "COM" is read in.
4. Select the "Bool" data type in the "Date type" column.
5. Enter a name of your choice for the PLC variable in the column "Name".



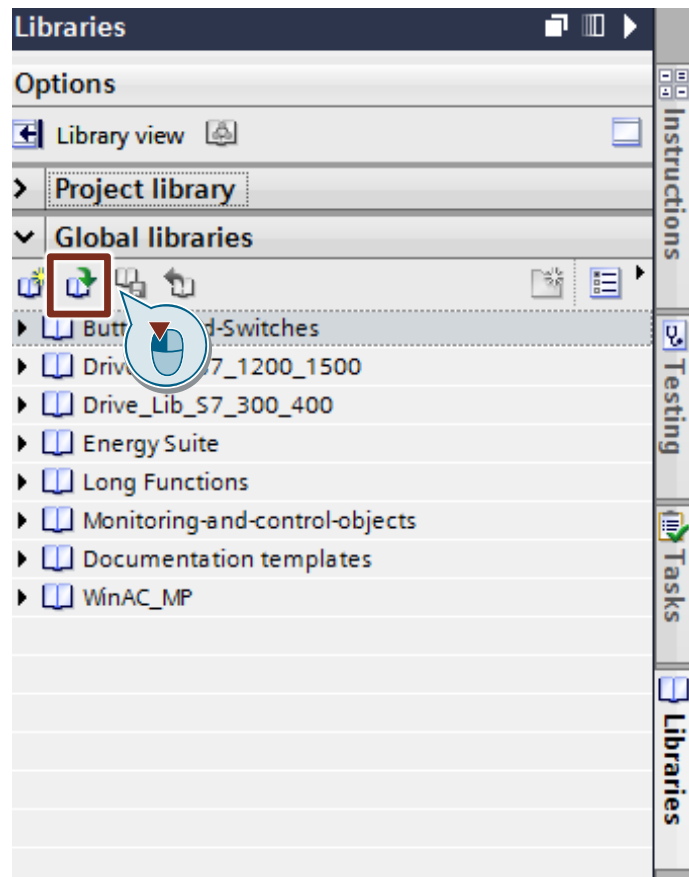
2.4 Integration of the function blocks in the user program

2.4.1 Opening the library "LPSU6200_V15.0-0_V1-1-0" in STEP 7

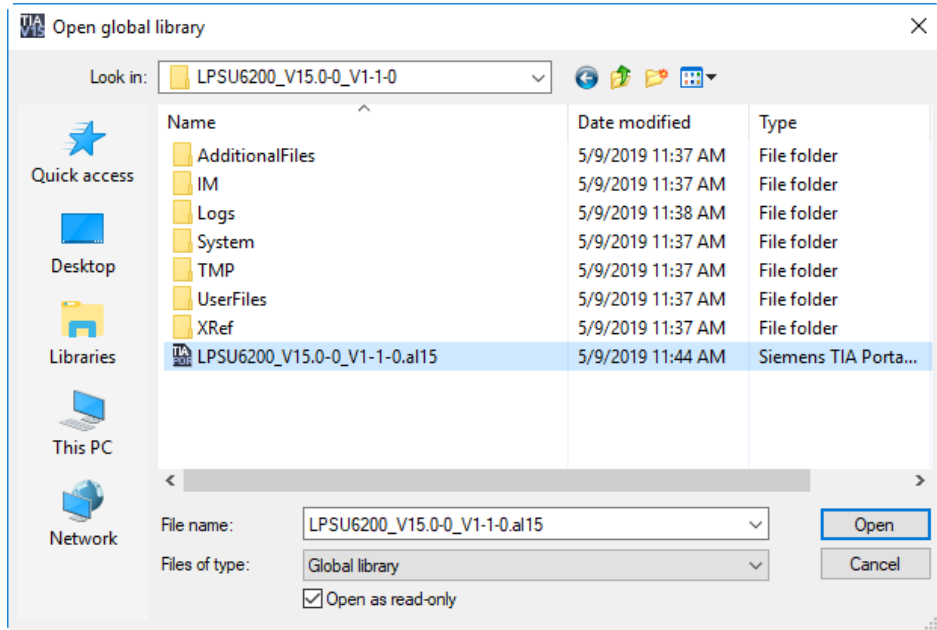
Note

For this section you must have downloaded the library "LPSU6200_V15.0-0_V1-1-0.zip" and unpacked it into a directory of your choice.

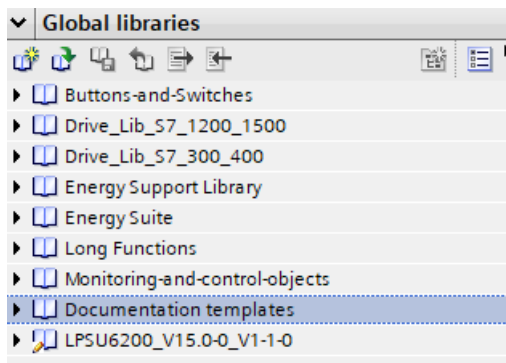
1. Click the "Libraries" Task Card and open the "Global libraries" palette.
2. Click the "Open global library" button to open the "LPSU6200_V15.0-0_V1-1-0" library. The corresponding dialog opens.



3. Select the global library "LPSU6200_V15.0-0_V1-1-0.al15" and confirm your selection with the "Open" button.

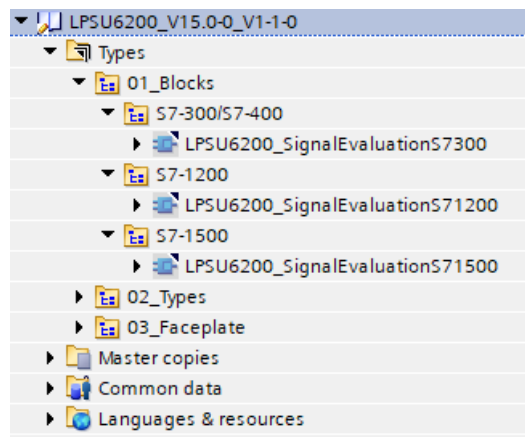


4. The library "LPSU6200_V15.0-0_V1-1-0.al15" is opened and displayed under the palette "Global libraries".

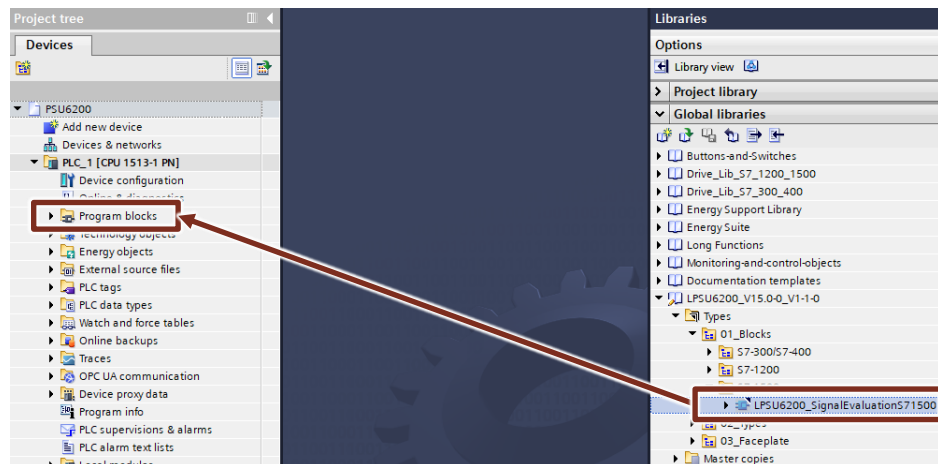


2.4.2 Copying Data Types and Function Modules to the User Project

- In the library you find under "Types > 01_Blocks" the function blocks for communication for the following S7-CPU:
 - S7-1500 CPUs: Function block "LPSU6200_SignalEvaluationS71500"
 - S7-1200 CPUs: Function block "LPSU6200_SignalEvaluationS71200"
 - S7-300 / S7-400 CPUs: Function block "LPSU6200_SignalEvaluationS7300"



- Insert the function block for your S7-CPU via drag & drop into the folder "Program blocks" of your device, e.g. S7-1500 CPU.

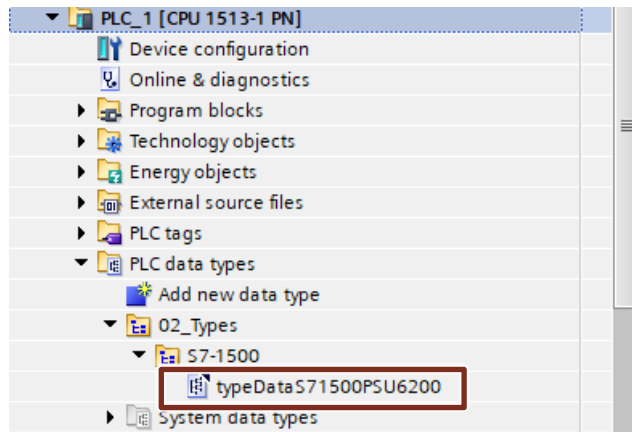


Note

The function blocks each use their own user-defined PLC data type (UDT – User defined Data type).

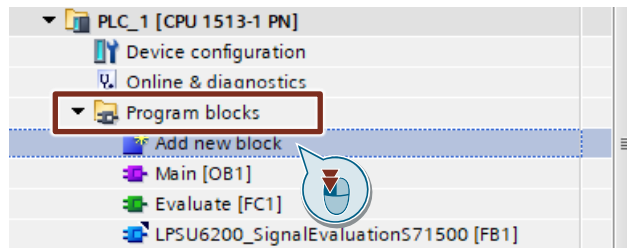
- S7-1500 CPUs: "typeDataS71500PSU6200"
- S7-1200 CPUs: "typeDataS71200PSU6200"
- S7-300 / S7-400 CPUs: "typeDataS7300PSU6200"

This is automatically inserted into the "PLC data types" folder of your S7-CPU when the respective function block is inserted.

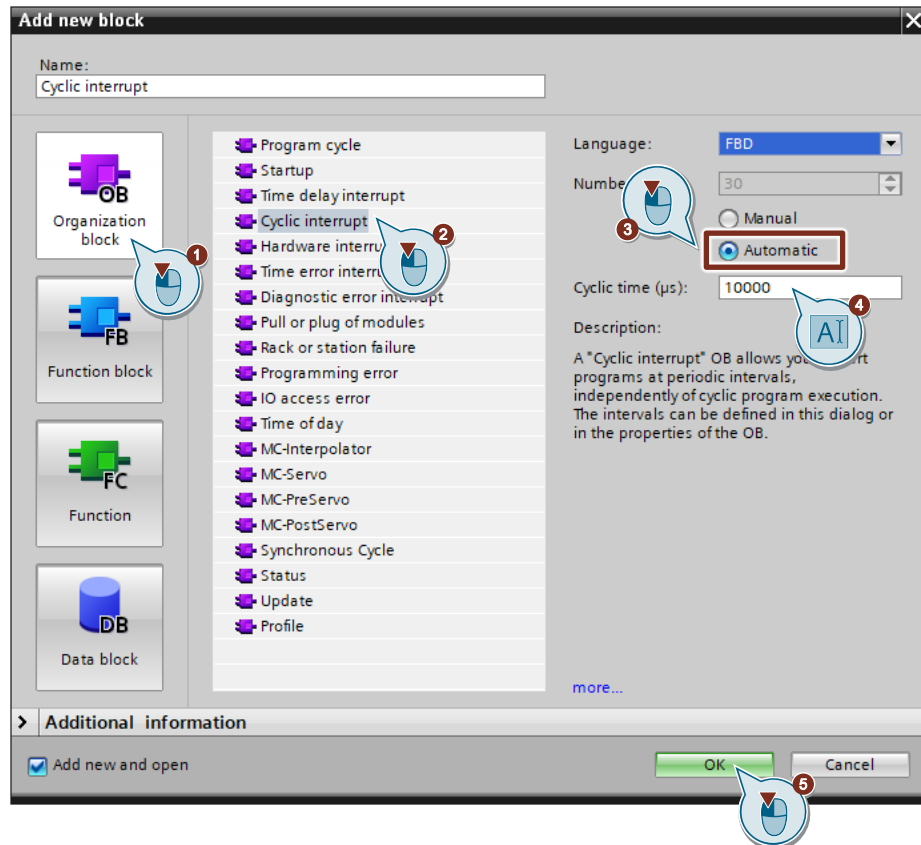


2.4.3 Create a cyclic interrupt OB for S7-1200 and S7-1500

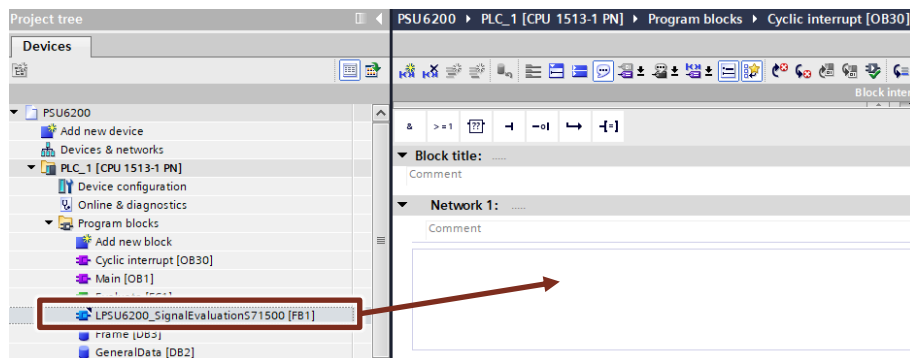
1. Navigate in the "Project tree" to the device folder of the S7-CPU.
2. Open the "Program blocks" folder and double-click the "Add new block" command to add a cyclic interrupt OB. The dialog "Add new block" opens.



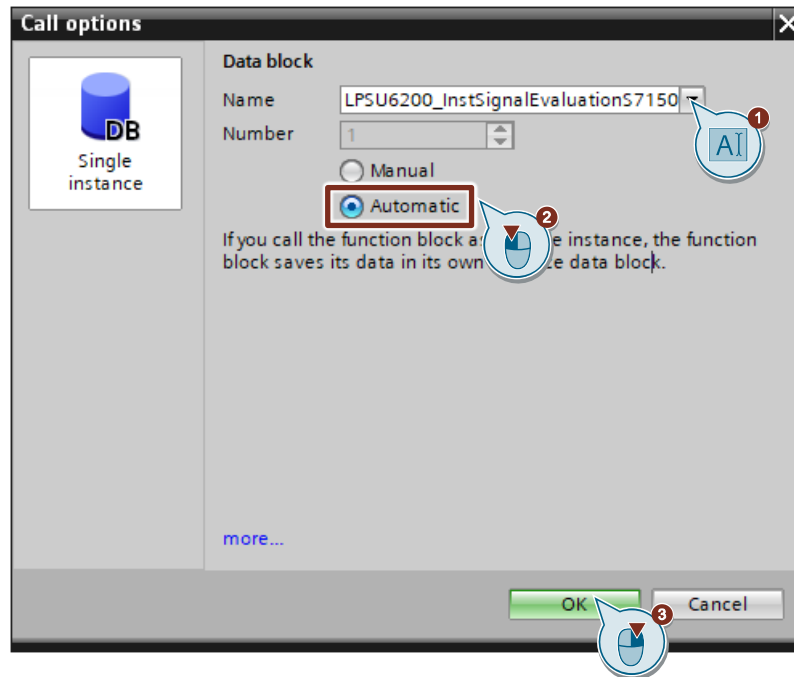
3. Make the following settings and then confirm your entries with the "OK" button.
 - Select the "Organization block" icon.
 - Select the entry "Cyclic interrupt" to create a cyclic interrupt OB.
 - Activate the "Automatic" radio button for automatic number assignment. The number of the cyclic interrupt OB is assigned by the TIA Portal.
 - Enter the cycle time. The cycle time is given in microseconds (μs). The cycle time for calling the FB "LPSU6200_SignalEvaluationS71500" and FB "LPSU6200_SignalEvaluationS71200" must not exceed 10 ms.
 - Accept the settings with "OK".



4. Double-click on the newly inserted cyclic interrupt OB to open the corresponding program editor.
5. Drag and drop the FB "LPSU6200_SignalEvaluationS71500" or FB "LPSU6200_SignalEvaluationS71200" from the project navigation into any network of the cyclic interrupt OB.

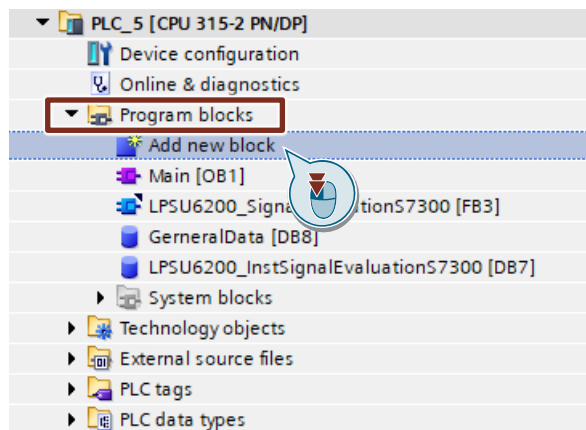


6. The dialog "Call options" for generating the instance DB of the FB "LPSU6200_SignalEvaluationS71500" or FB "LPSU6200_SignalEvaluationS71200" opens automatically.
7. Make the following settings and then confirm your entries with the "OK" button.
 - Enter the name of the instance DB.
 - Activate the "Automatic" radio button for automatic number assignment. The instance DB number is assigned by the TIA Portal.
 - Accept the settings with "OK".

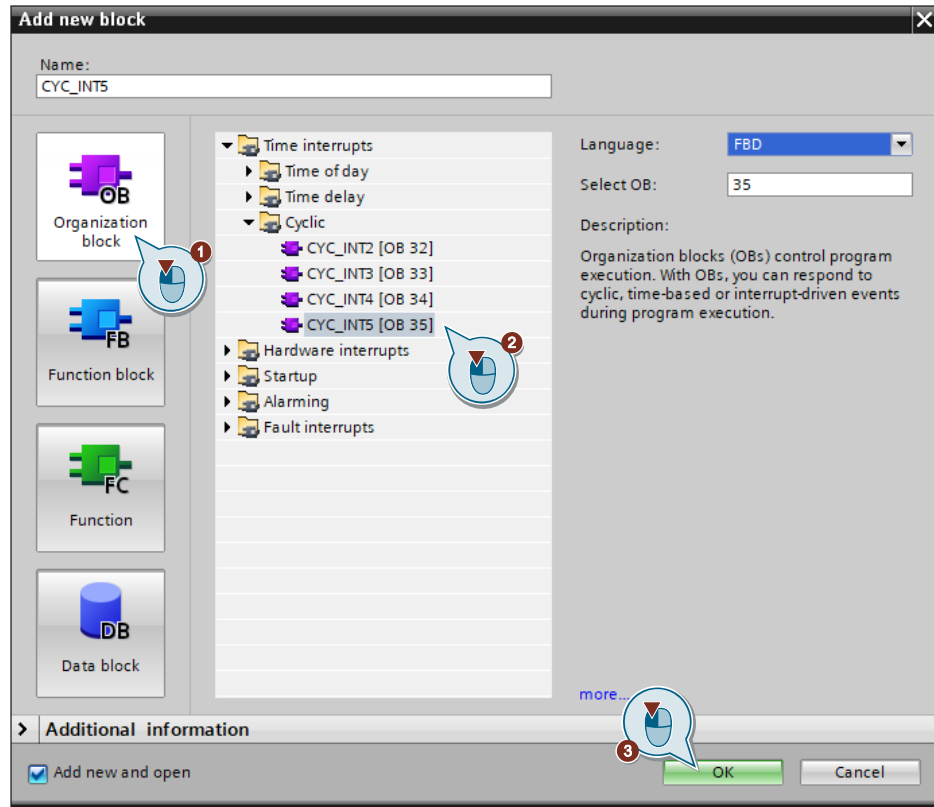


2.4.4 Create a cyclic interrupt OB for S7-300 / S7-400

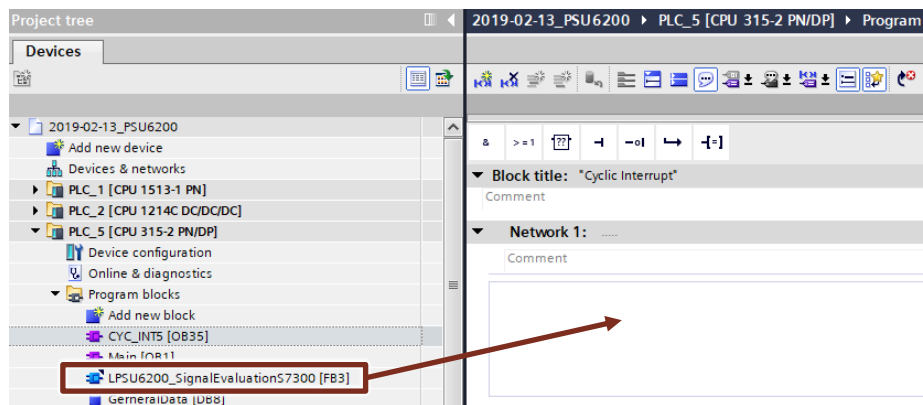
1. Navigate in the "Project tree" to the device folder of the S7-CPU.
2. Open the "Program blocks" folder and double-click the "Add new block" command to add a cyclic interrupt OB. The dialog "Add new block" opens.



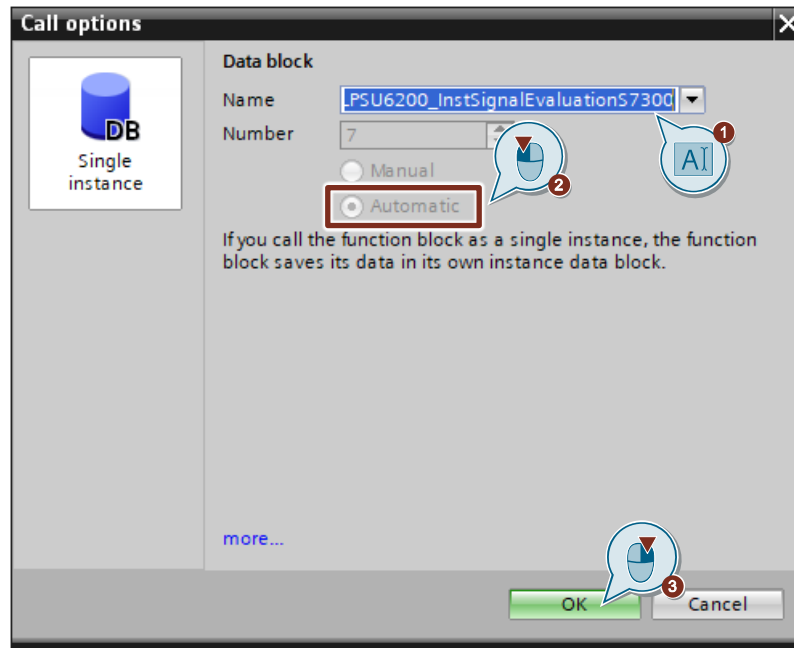
3. Make the following settings and then confirm your entries with the "OK" button.
 - Select the "Organization block" icon.
 - Select the entry "Time interrupts > Cyclic > CYC_INT5 [OB35]" to create a cyclic interrupt OB. The number of the cyclic interrupt OB is shown under "Select OB:". The OB35 is used in this example.
 - Accept the settings with "OK".



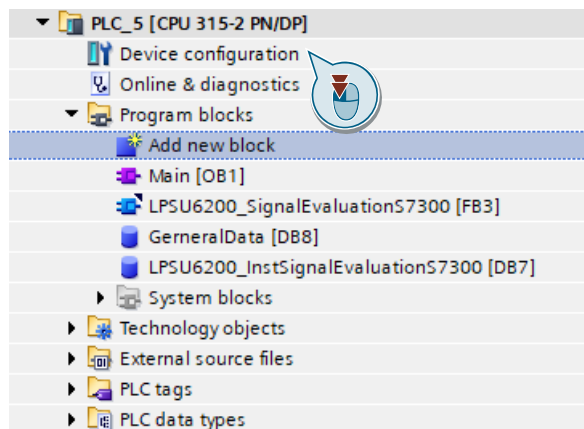
4. Double-click on the newly inserted cyclic interrupt OB to open the corresponding program editor.
5. Drag and drop the FB "LPSU6200_SignalEvaluationS7300" from the project navigation into any network of the cyclic interrupt OB.



6. The dialog "Call options" for generating the instance DB of the FB "LPSU6200_SignalEvaluationS7300" opens automatically.
7. Make the following settings and then confirm your entries with the "OK" button.
 - Enter the name of the instance DB.
 - Activate the "Automatic" radio button for automatic number assignment. The instance DB number is assigned by the TIA Portal.
 - Accept the settings with "OK".



8. Navigate in the "Project tree" to the device folder of the S7-CPU and double-click the "Device configuration" command. The "Device view" of the S7-300 opens in the hardware and network editor.



9. Mark the S7-300 CPU. The "Properties" of the S7-300 CPU are displayed in the inspector window.
10. In the "General" tab, navigate to "Interrupts > Cyclic interrupts".

11. Enter the value "10 ms" in the "Execution" column.

The screenshot shows the SIMATIC Manager interface for PLC_5 [CPU 315-2 PN/DP]. The 'Cyclic interrupts' table is displayed with the following data:

OB number	Priority	Execution	Phase offset	Unit
OB 32:	9	1000	0	ms
OB 33:	10	500	0	ms
OB 34:	11	200	0	ms
OB 35:	12	10	0	ms

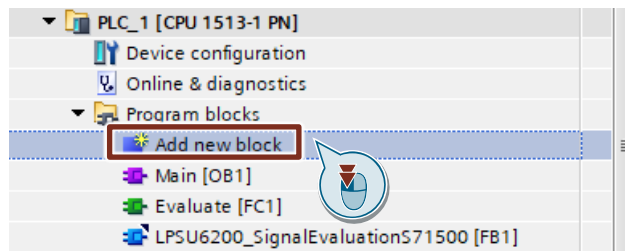
The 'Execution' column for OB 35 is highlighted with a red box, and a callout bubble containing 'AI' points to it. The 'Cyclic interrupts' menu item in the left sidebar is also highlighted with a blue dashed box and a callout bubble containing 'AI'.

2.4.5 Creating a Global Data Block for Data Exchange

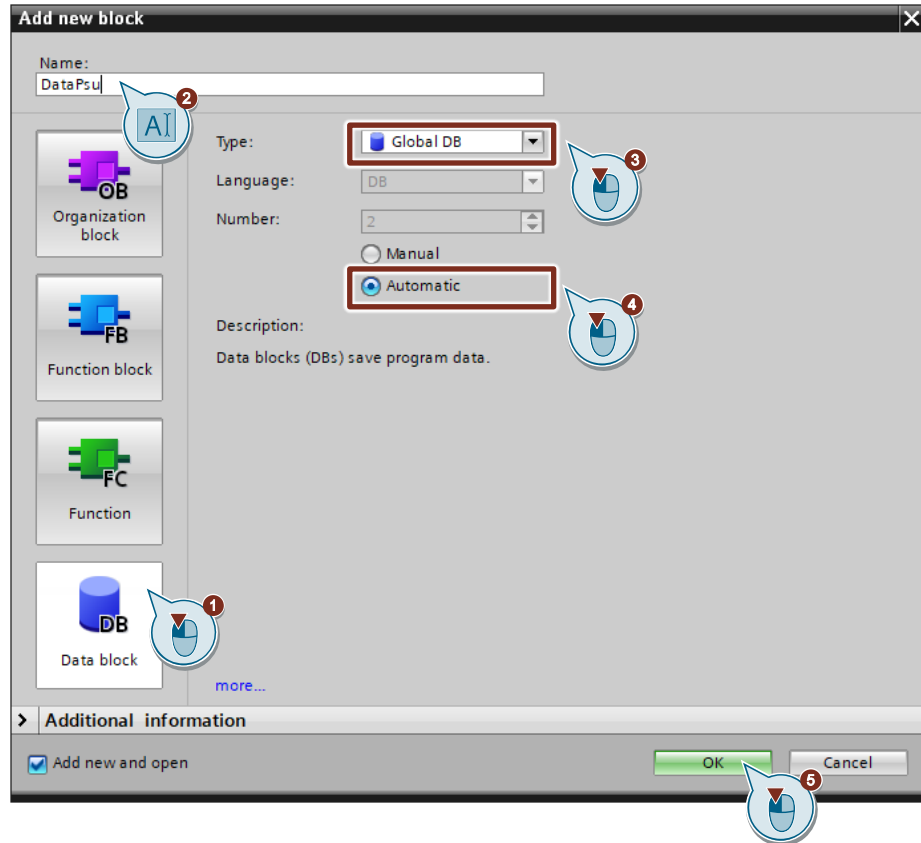
Note

This section shows you how to create a global data block for data exchange. The data block is used to store the data of the SITOP PSU6200 which was read out with the help of the FBs "LPSU6200_SignalEvaluationS71500" and "LPSU6200_SignalEvaluationS71200".

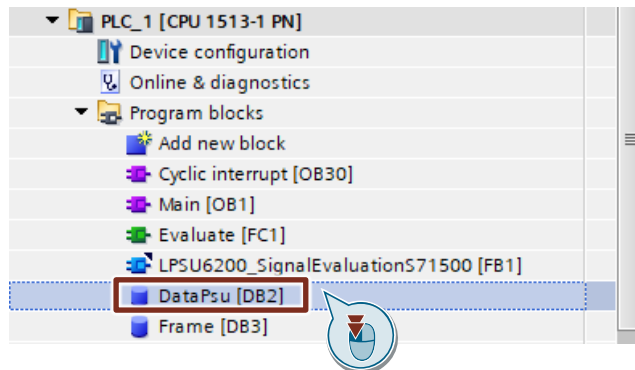
1. Navigate in the "Project tree" to the device folder of the S7-CPU.
2. Open the "Program blocks" folder and double-click the "Add new block" command to add a new data block. The dialog "Add new block" opens.



3. Make the following settings and then confirm your entries with the "OK" button.
 - Select the symbol "Data block".
 - Select "Global DB" as the type.
 - Enter the name of the DB.
 - Activate the "Automatic" radio button for automatic number assignment. The number of the global DB is assigned by the TIA Portal.



4. Double-click the newly inserted global data block to open it.



- Double-click "<Add new>" to create the variables accordingly [Table 2-6](#).

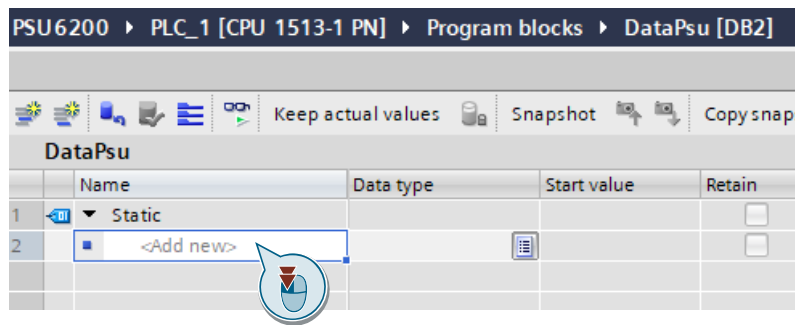


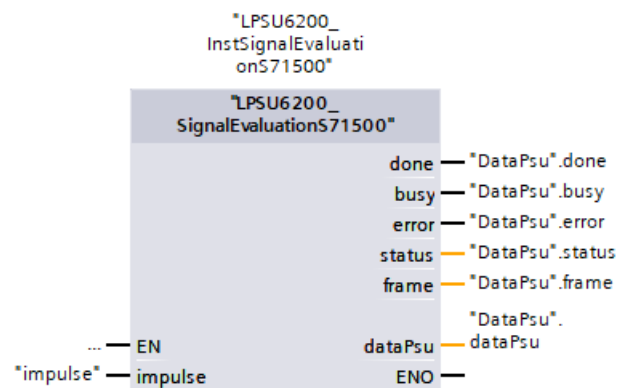
Table 2-6

Name	Data type	Interconnection at the FB
done	Bool	Output "done"
busy	Bool	Output "busy"
error	Bool	Output "error"
status	Word	Output "status"
frame	DWord	Output "frame"
dataPsu	typeS71500DataPsu6200	Output "dataPsu"

DataPsu			
	Name	Data type	Start value
1	Static		
2	done	Bool	false
3	busy	Bool	false
4	error	Bool	false
5	status	Word	16#0
6	frame	DWord	16#0
7	dataPsu	*typeDataS71500PSU6200*	
8	deviceName	String	'PSU6200'
9	stateOutputVoltageOK	Bool	false
10	stateOperatingHours	Bool	false
11	outputCurrentLoad	Byte	16#04
12	overTemperatureShutdown	Bool	false
13	stateCOM	Bool	true
14	stateHV	Bool	false
15	statePO	Bool	false
16	outputVoltage	Real	0.0
17	outputCurrent	Int	0
18	nominalOutputVoltage	Int	24
19	nominalOutputCurrent	Int	10
20	productState	Int	1
21	firmwareVersion	Int	1
22	stateDeviceTemperature	Int	0
23	phaseError	Bool	false
24	operatingHours	DInt	1000
25	articleNumber	String	'6EP3333-75B00-0AX0'
26	serialNumber	String	'Q6/L1'
27	counterGlitchOvervoltage	Int	0
28	counterGlitchUndervoltage	Int	0
29	stateNoCommunication	Bool	false

2.4.6 Call the function block in the user program

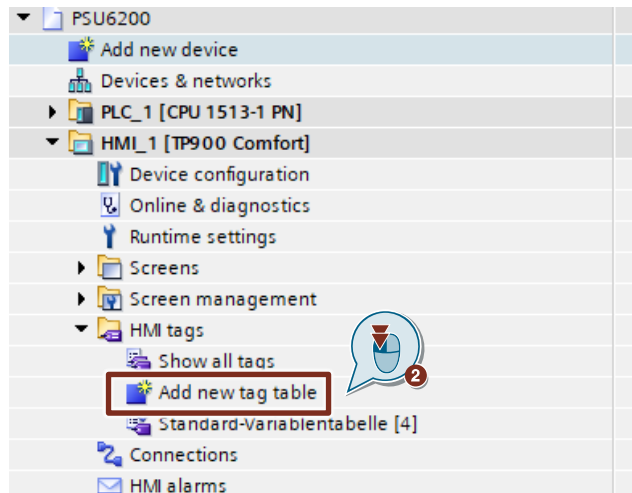
1. In the "Project tree" open the folder "Program blocks" of your S7-CPU.
2. Double-click on the cyclic interrupt OB to open the corresponding program editor.
3. Assign the PLC variable which refers to the address of the digital input via which the diagnostics interface "COM" is read in to the "impulse" input (see section [2.3.2](#)).
4. Assign the variables that you have created in the global data block to the outputs of the FB (see section [2.4.5](#)).



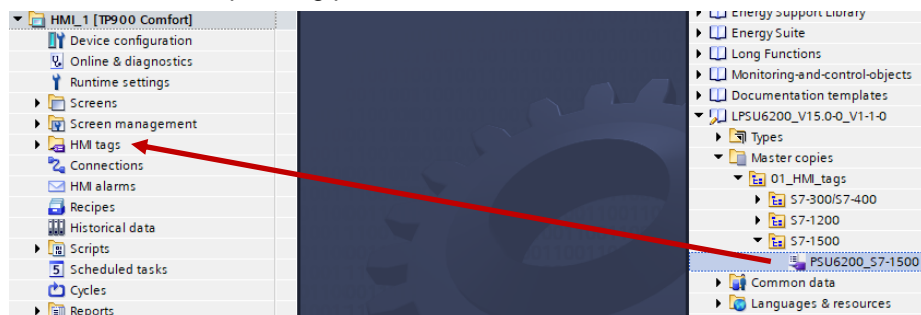
2.5 Integrating Faceplates for S7-1200 and S7-1500

2.5.1 Create an HMI tag

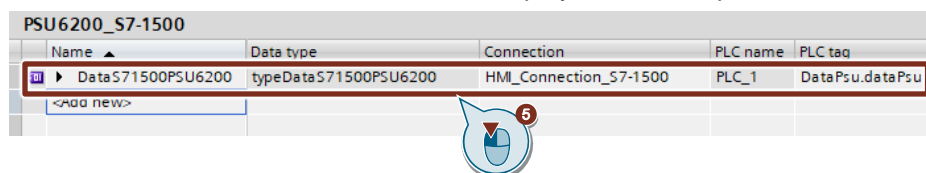
1. Navigate to the device folder of the HMI operating device in the "Project tree".
2. Open the "HMI tags" folder and double-click the "Add new tag table" command to create a new HMI tag table.



3. Name the tag table "PSU6200" and double-click the tag table to open it.
4. Create a variable "DataS71500PSU6200".
5. Alternatively, you can find the "PSU6200" tag table in the "LPSU6200_V15.0-0_V1-1-0" library in the folder "Master copies > 01_HM_tags".
6. Insert the "PSU6200" tag table by dragging and dropping into the "HMI tags" folder of the HMI operating panel

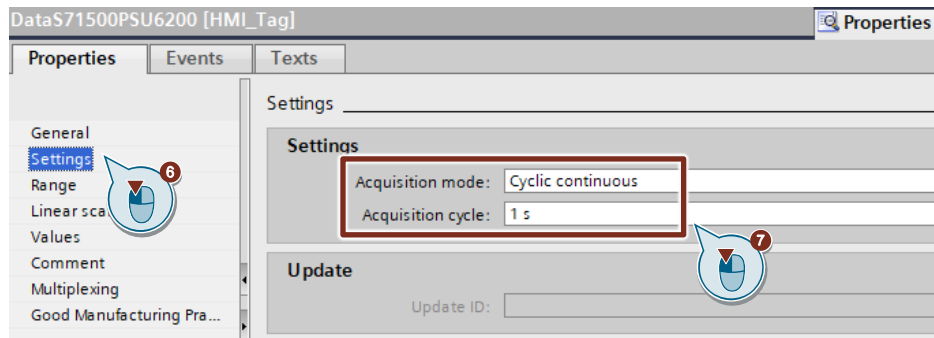


7. Select the variable "DataS71500PSU6200" in the tag table. The properties of the "DataS71500PSU6200" variable are displayed in the inspector window.



8. In the "General" tab, navigate to "Settings".

9. Select the following settings:
 - "Acquisition mode": "Cyclic continuous"
 - "Acquisition cycle": 1 s

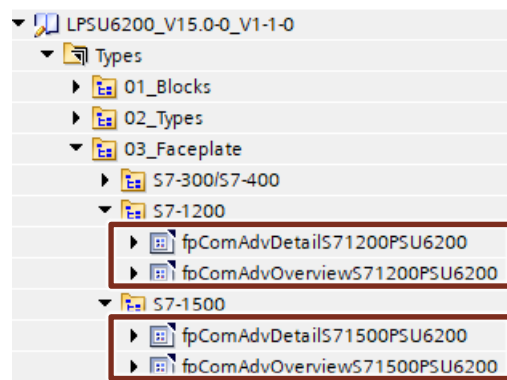


10. Connect the variable with the corresponding PLC variable of the data type "typeDataS71500PSU6200" or "typeDataS71200PSU6200".

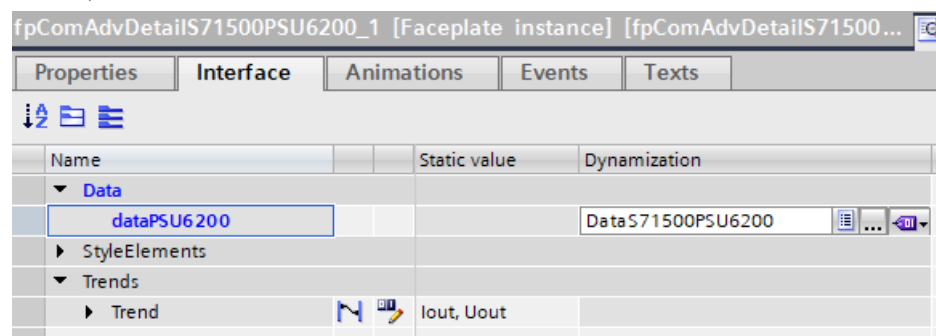
PSU6200_S7-1500					
Name ▲	Data type	Connection	PLC name	PLC tag	
◀ ▶ DataS71500PSU6200	typeDataS71500PSU6200	HMI_Connection_S7-1500	PLC_1	DataPsu.dataPsu	
<Add new>					

2.5.2 Faceplate "fpComAdvDetailS71500PSU6200" and "fpComAdvDetailS71200PSU6200"

1. Navigate in the "Project tree" to the folder "Screens".
2. Double-click on the screen in which you want to integrate the faceplate for displaying the SITOP PSU6200 device data. The screen opens.
3. Open folder "Types > 03_Faceplate" of the library "LPSU6200_V15.0-0_V1-1-0". There you can find the faceplates for the S7-1500 CPUs and S7-1200 CPUs.
 - fpComAdvDetailS71500PSU6200: Faceplate for communication with S7-1500 CPUs to graphically display device data as well as output voltage and output current of the PSU6200.
 - fpComAdvOverviewS71500PSU6200: Faceplate for communication with S7-1500 CPUs shows an overview of the most important device data of the PSU6200.
 - fpComAdvDetailS71200PSU6200: Faceplate for communication with S7-1200 CPUs to graphically display device data as well as output voltage and output current of the PSU6200.
 - fpComAdvOverviewS71200PSU6200: Faceplate for communication with S7-1200 CPUs shows an overview of the most important device data of the PSU6200.



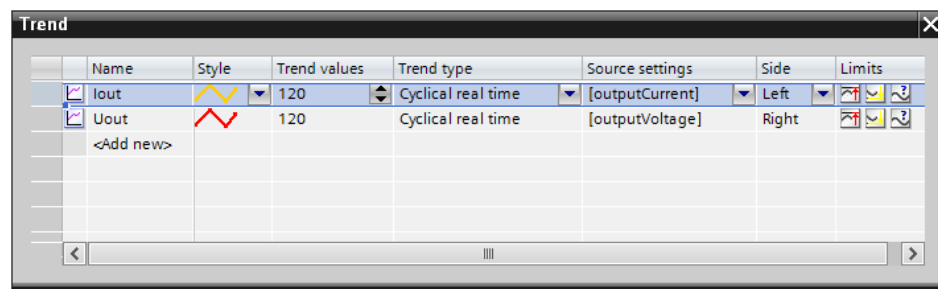
4. Insert an instance of the faceplate "fpComAdvDetailS71500PSU6200" by dragging and dropping into the opened screen.
5. Highlight the faceplate in the screen and open the interface of the faceplate in the "Properties > Interface" tab.
6. Connect the created HMI tag to the property "dataPSU6200" (category: "Data").



7. Create two curves for the voltage and current values via the "Trend" property (category: "Trends") with the following settings.

Table 2-7

Parameters	Value
"Name"	<ul style="list-style-type: none"> • Iout • Uout
"Style"	<ul style="list-style-type: none"> • "Line color": yellow, red • "Mode": "Interpolated" • "Line width": of your choice • "Line style": "Solid"
"Trend values"	At least 120
"Trend Type"	"Cyclical real time"
"Source Settings"	<ul style="list-style-type: none"> • "Process values": Variables "outputCurrent" and "outputVoltage" • "Cyclic": 1 s
"Side"	<ul style="list-style-type: none"> • Left • Right



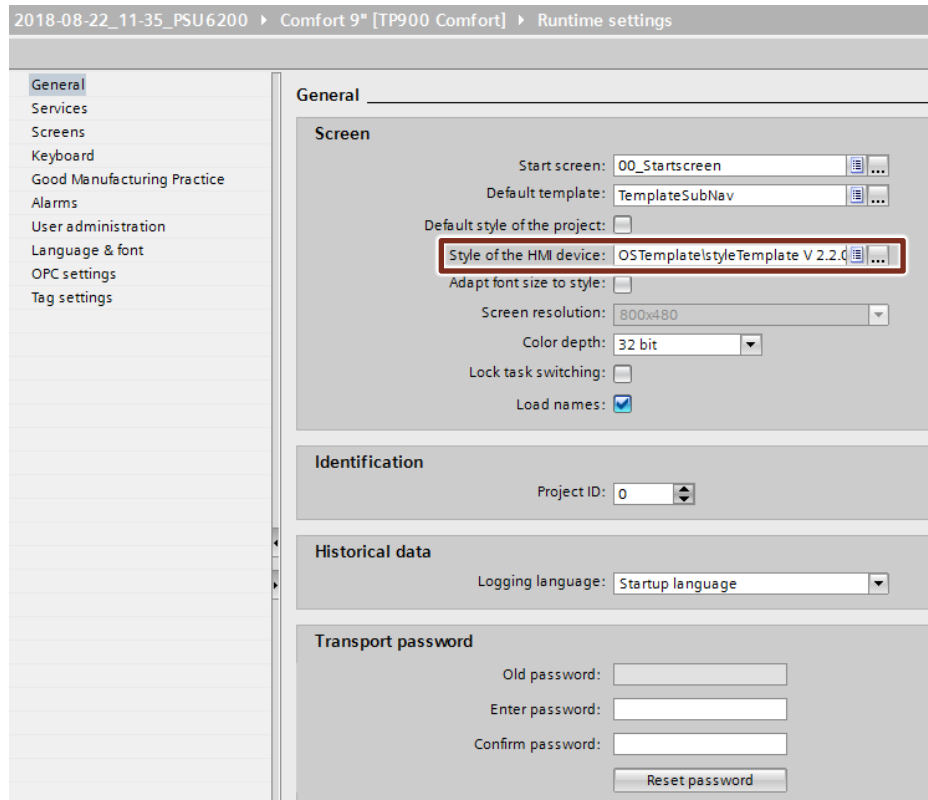
8. **Optional: Perform this step if you would like to adjust the style of the faceplate to your own style.**

In the "Style Elements" category, you can find the properties to change individual styles of objects.

- Select the desired object styles.
- Accept the respective object style by activating the option box under the respective property.

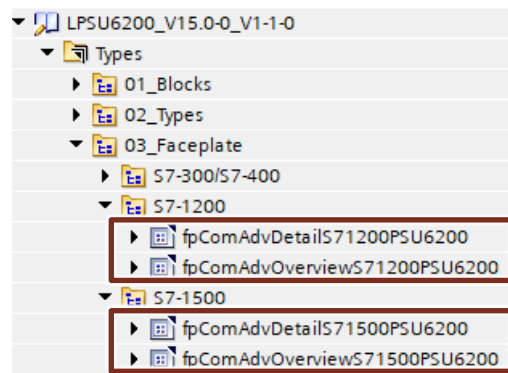
Note

A current style for the HMI device must be activated in the runtime settings under "General > Screen > Style of the HMI device".

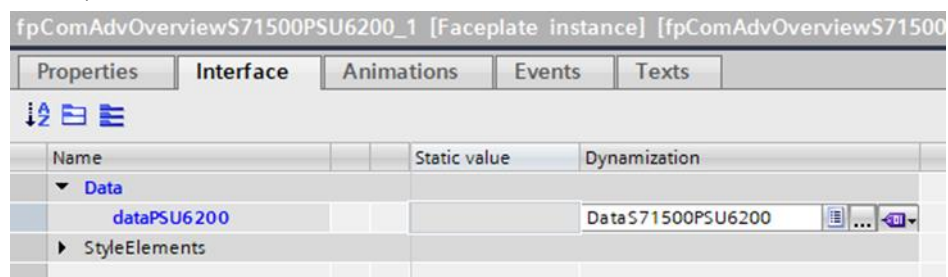


2.5.3 Faceplate "fpComAdvOverviewS71500" and "fpComAdvOverviewS71200"

1. Navigate in the "Project tree" to the folder "Screens".
2. Double-click on the screen in which you want to integrate the faceplate for displaying the SITOP PSU6200 device data. The screen opens.
3. Open folder "Types > 03_Faceplate" of the library "LPSU6200_V15.0-0_V1-1-0". There you can find the faceplates for the S7-1500 CPUs and S7-1200 CPUs.
 - fpComAdvDetailS71500PSU6200: Faceplate for communication with S7-1500 CPUs to graphically display device data as well as output voltage and output current of the PSU6200.
 - fpComAdvOverviewS71500PSU6200: Faceplate for communication with S7-1500 CPUs shows an overview of the most important device data of the PSU6200.
 - fpComAdvDetailS71200PSU6200: Faceplate for communication with S7-1200 CPUs to graphically display device data as well as output voltage and output current of the PSU6200.
 - fpComAdvOverviewS71200PSU6200: Faceplate for communication with S7-1200 CPUs shows an overview of the most important device data of the PSU6200.



4. Insert an instance of the faceplate "fpComAdvOverviewS71500PSU6200" by dragging and dropping into the opened screen.
5. Highlight the faceplate in the screen and open the interface of the faceplate in the "Properties > Interface" tab.
6. Connect the created HMI tag to the property "dataPSU6200" (category: "Data").



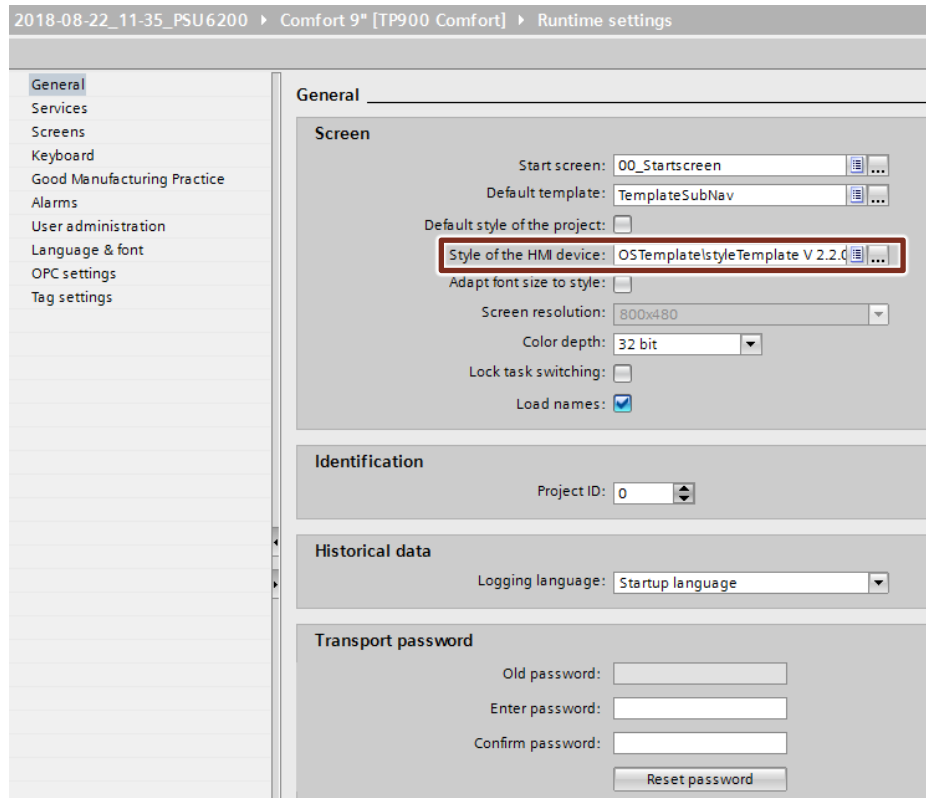
7. **Optional: Perform this step if you would like to adjust the style of the faceplate to your own style.**

In the "Style Elements" category, you can find the properties to change individual styles of objects.

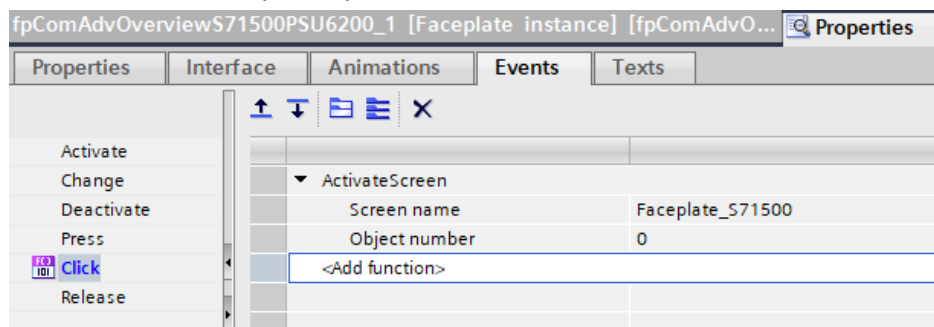
- Select the desired object styles.
- Accept the respective object style by activating the option box under the respective property.

Note

A current style for the HMI device must be activated in the runtime settings under "General > Screen > Style of the HMI device".



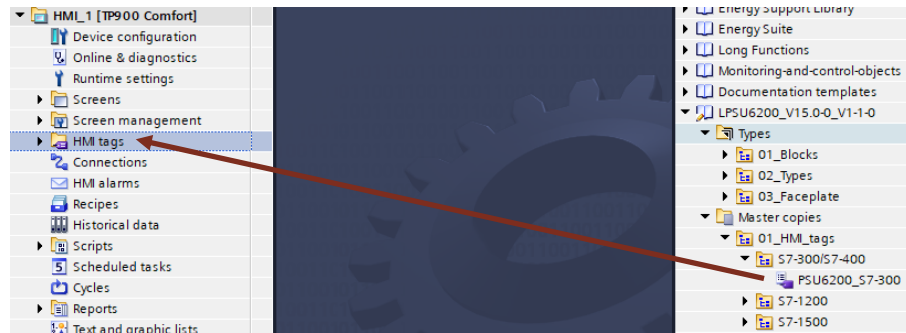
8. Mark the faceplate in the screen and open the events of the faceplate via the register "Properties > Events".
9. Connect the "ActivateScreen" function to the event "Click", to open the screen that includes the faceplate "fpComAdvDetailS71500".



2.6 Integrating Faceplates for S7-300 / S7-400

2.6.1 Creating HMI tag

1. Open the the global library "LPSU6200_V15.0-0_V1-1-0" (see section [2.4.1](#)).
2. The "PSU6200" tag table is available in the folder "Master copies > 01_HMI_tags > S7-300/S7-400".
3. Insert the "PSU6200" tag table by dragging and dropping into the "HMI tags" folder of the HMI operating panel.



4. Open the "HMI tags" folder and double-click the "PSU6200" tag table to open it in the workspace.
5. In the "Connection" column, select the HMI connection that you have created between HMI operating panel and S7-300 / S7-400.
6. In the "Address" column, set the start address of the PLC data type "typeDataS7300PSU6200".

PSU6200_S7-300					
Name	Data type	Connection	PLC name	PLC tag	Address
▶ DataS7300PSU6200	typeDateS7300PSU6200HMI V 1.1.0	HMI_Connection_S7-300	PLC_3	<Undefined>	%DB2.DBX8.0
<input type="button" value="Add new"/>					



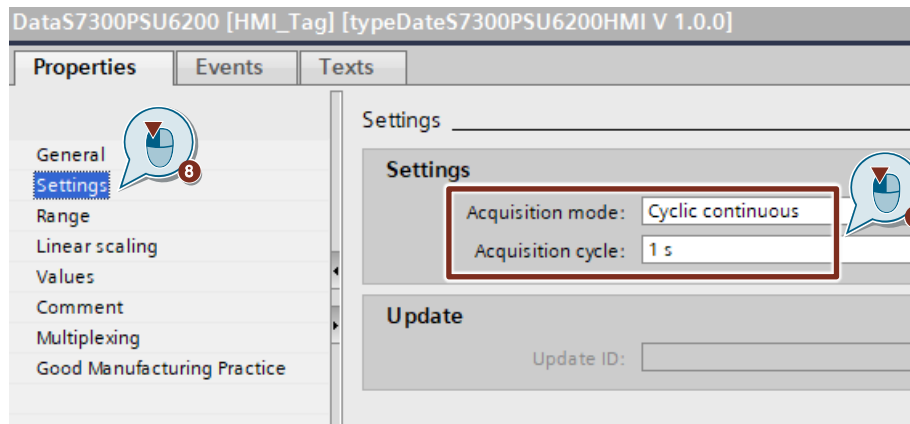
7. Select the variable "DataS7300PSU6200" in the tag table. The properties of the "DataS7300PSU6200" variable are displayed in the inspector window.

PSU6200_S7-300					
Name	Data type	Connection	PLC name	PLC tag	Address
▶ DataS7300PSU6200	typeDateS7300PSU6200HMI V 1.1.0	HMI_Connection_S7-300	PLC_3	<Undefined>	%DB2.DBX8.0
<input type="button" value="Add new"/>					



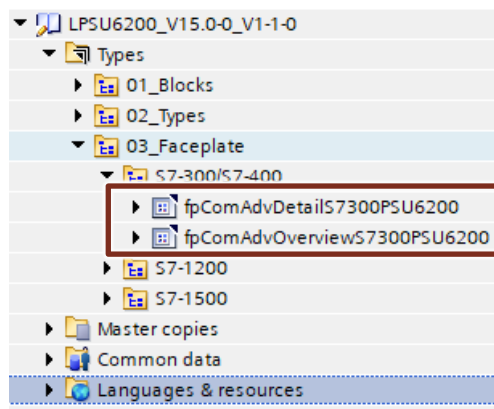
8. In the "General" tab, navigate to "Settings".

9. Select the following settings:
- "Acquisition mode": "Cyclic continuous"
 - "Acquisition cycle": 1 s

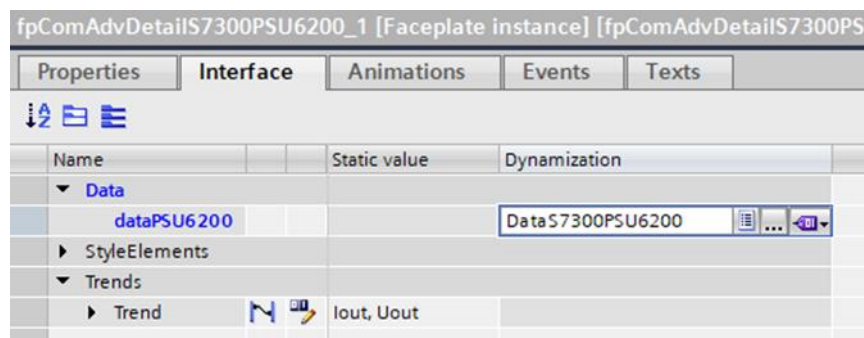


2.6.2 Faceplate "fpComAdvDetailS7300PSU6200"

1. Navigate in the "Project tree" to the folder "Screens".
2. Double-click on the screen in which you want to integrate the faceplate for displaying the SITOP PSU6200 device data. The screen opens.
3. Open folder "Types > 03_Faceplate" of the library "LPSU6200_V15.0-0_V1-1-0". There you can find the faceplates for the S7-300 / S7-400 CPUs.
 - fpComAdvDetailS7300PSU6200: Faceplate for communication with S7-300 CPUs to graphically display device data as well as output voltage and output current of the PSU6200.
 - fpComAdvOverviewS7300PSU6200: Faceplate for communication with S7-300 CPUs shows an overview of the most important device data of the PSU6200.



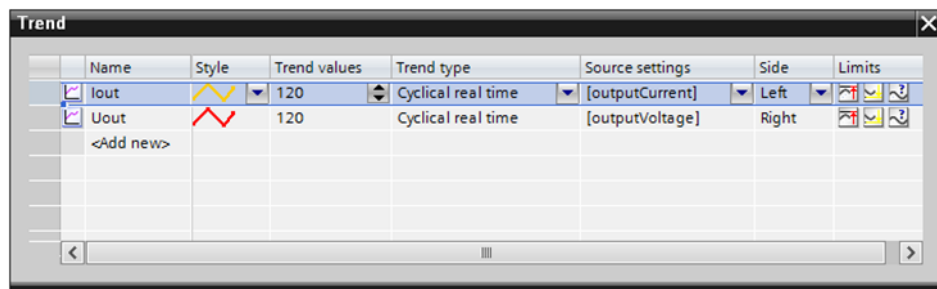
4. Insert an instance of the faceplate "fpComAdvDetailS7300PSU6200" by dragging and dropping into the opened screen.
5. Highlight the faceplate in the screen and open the interface of the faceplate in the "Properties > Interface" tab.
6. Connect the created HMI tag to the property "dataPSU6200" (category: "Data").



7. Create two curves for the voltage and current values via the "Trend" property (category: "Trends") with the following settings.

Table 2-8

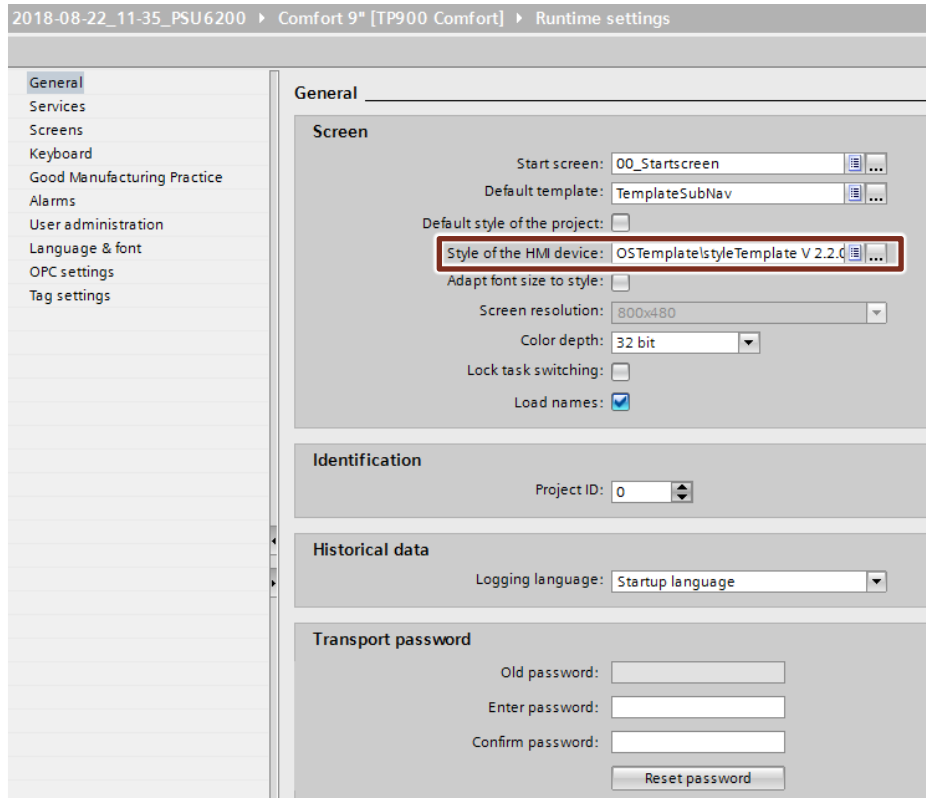
Parameters	Value
"Name"	<ul style="list-style-type: none"> • Iout • Uout
"Style"	<ul style="list-style-type: none"> • "Line color": yellow, red • "Mode": "Interpolated" • "Line width": of your choice • "Line style": "Solid"
"Trend values"	At least 120
"Trend Type"	"Cyclical real time"
"Source Settings"	<ul style="list-style-type: none"> • "Process values": Variables "outputCurrent" and "outputVoltage" • "Cyclic": 1 s
"Side"	<ul style="list-style-type: none"> • Left • Right



8. **Optional: Perform this step if you would like to adjust the style of the faceplate to your own style.**
 In the "Style Elements" category, you can find the properties to change individual styles of objects.
 - Select the desired object styles.
 - Accept the respective object style by activating the option box under the respective property.

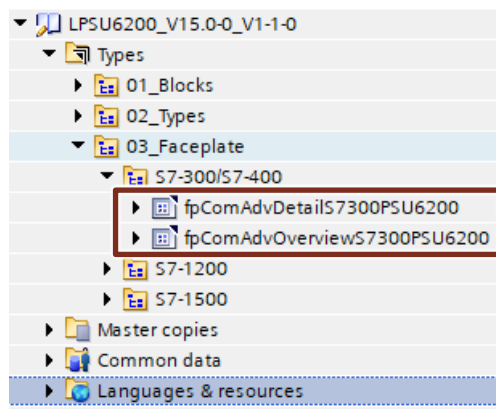
Note

A current style for the HMI device must be activated in the runtime settings under "General > Screen > Style of the HMI device".

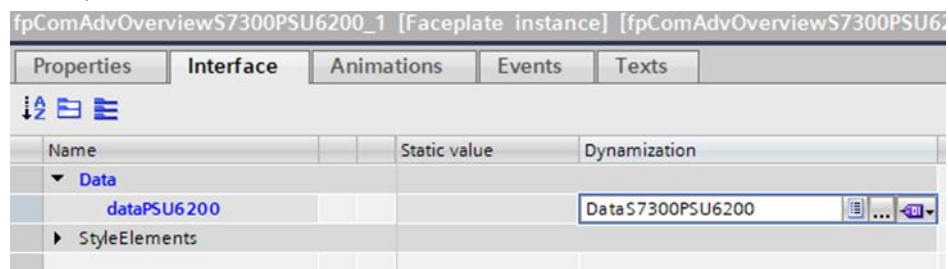


2.6.3 Faceplate "fpComAdvOverviewS7300PSU6200"

1. Navigate in the "Project tree" to the folder "Screens".
2. Double-click on the screen in which you want to integrate the faceplate for displaying the SITOP PSU6200 device data. The screen opens.
3. Open folder "Types > 03_Faceplate" of the library "LPSU6200_V15.0-0_V1-1-0". There you can find the faceplates for the S7-300 / S7-400 CPUs.
 - fpComAdvDetailS7300PSU6200: Faceplate for communication with S7-300 CPUs to graphically display device data as well as output voltage and output current of the PSU6200.
 - fpComAdvOverviewS7300PSU6200: Faceplate for communication with S7-300 CPUs shows an overview of the most important device data of the PSU6200.



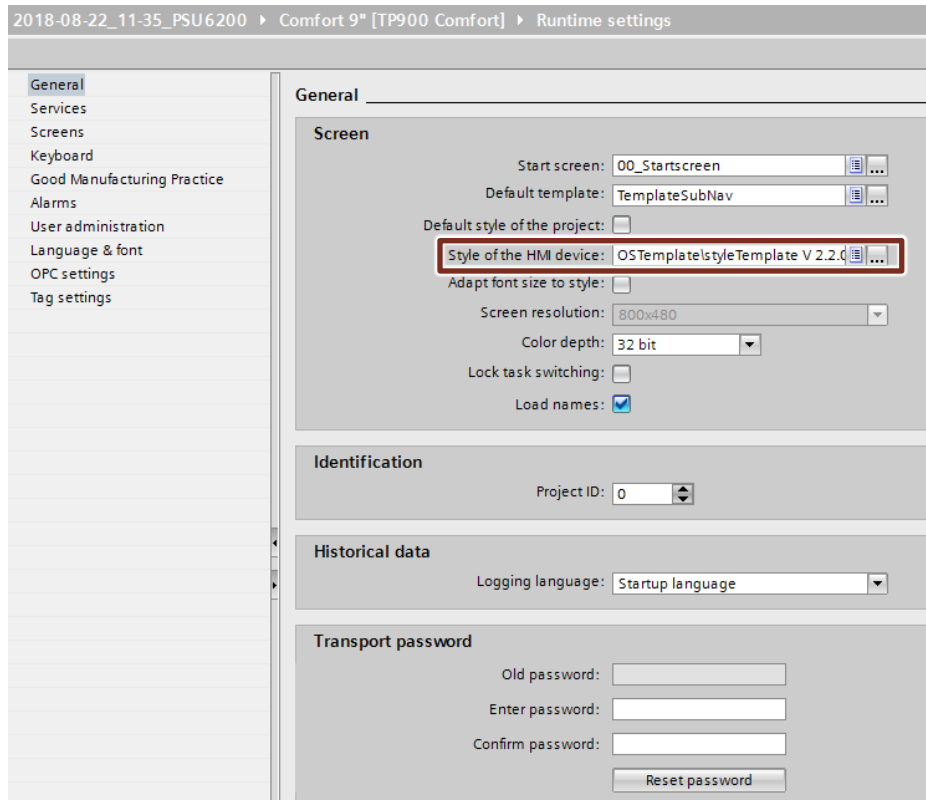
4. Insert an instance of the faceplate "fpComAdvOverviewS7300PSU6200" by dragging and dropping into the opened screen.
5. Highlight the faceplate in the screen and open the interface of the faceplate in the "Properties > Interface" tab.
6. Connect the created HMI tag to the property "dataPSU6200" (category: "Data").



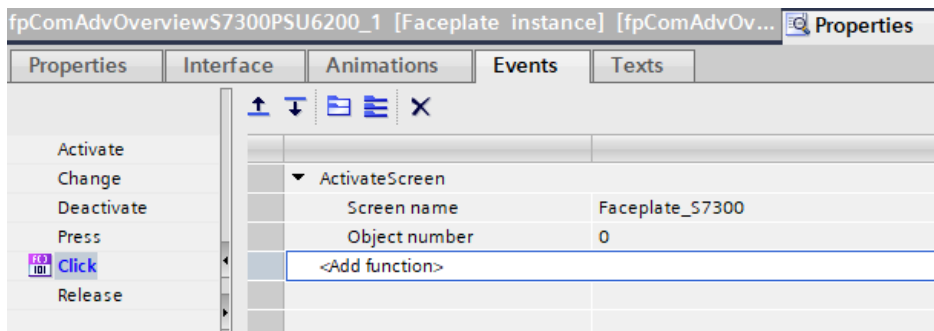
7. **Optional: Perform this step if you would like to adjust the style of the faceplate to your own style.**
In the "Style Elements" category, you can find the properties to change individual styles of objects.
 - Select the desired object styles.
 - Accept the respective object style by activating the option box under the respective property.

Note

A current style for the HMI device must be activated in the runtime settings under "General > Screen > Style of the HMI device".



8. Mark the faceplate in the screen and open the events of the faceplate via the register "Properties > Events".
9. Connect the "ActivateScreen" function to the event "Click", to open the screen that includes the faceplate "fpComAdvDetailS7300".

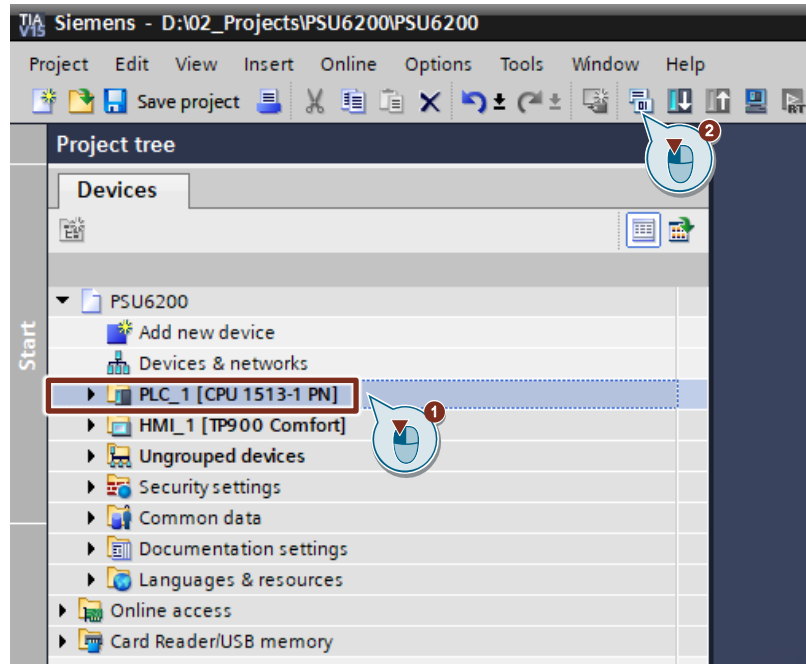


2.7 Compiling and Downloading

2.7.1 User program of the S7-CPU

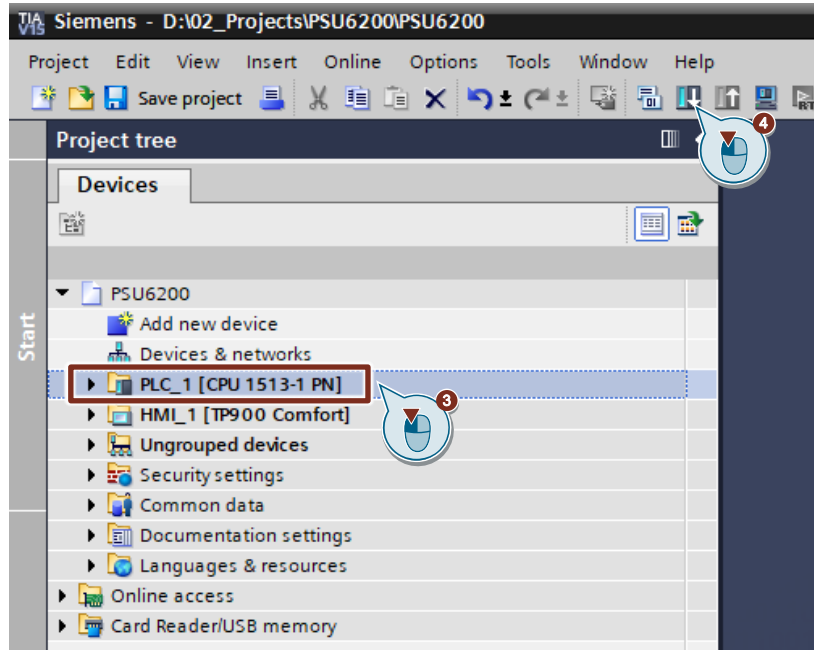
Compiling

1. Select the S7 CPU in the "Project tree".
2. Click on the "Compile" button in the toolbar.



Downloading

1. Make sure that your engineering PC and your S7 CPU are in the same subnet.
2. On the S7-CPU display, set the IP address and subnet mask which you entered in the hardware configuration.
3. Select the S7 CPU in the "Project tree".
4. Click on the "Download to device" button in the toolbar to load the hardware configuration and the software into the S7 CPU.

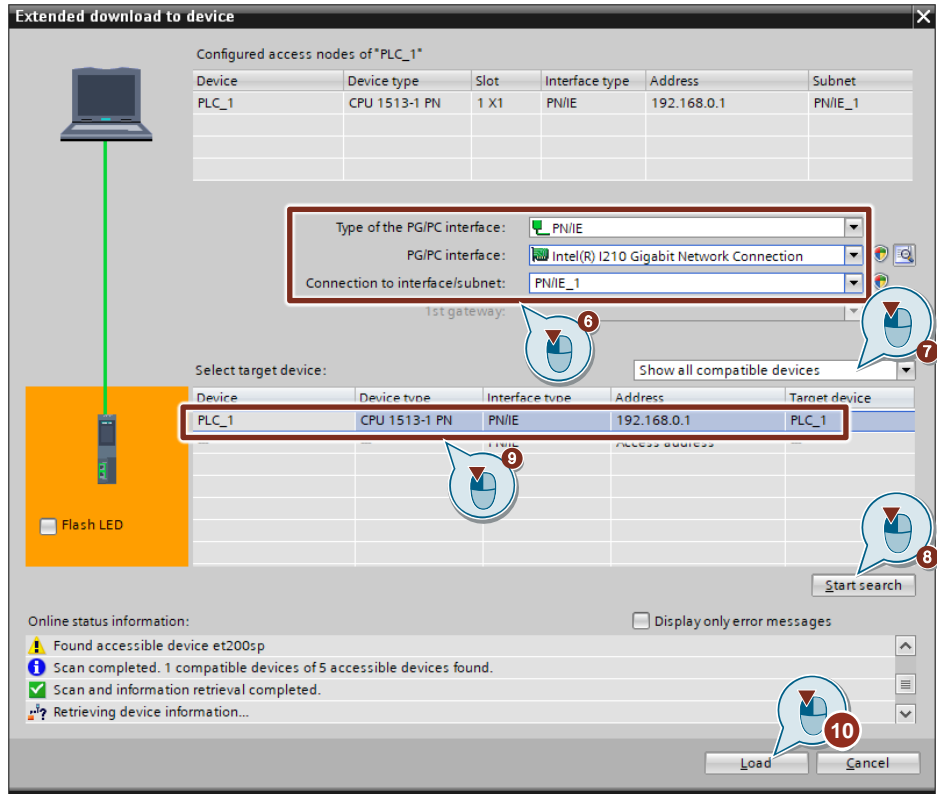


5. The "Extended download to device" or "Load preview" dialog opens automatically

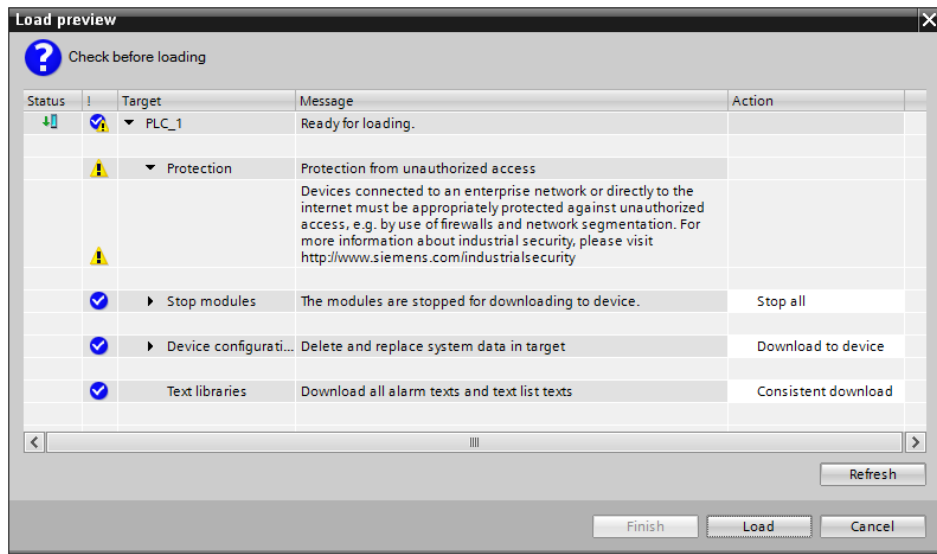
Note

The "Extended download to device" dialog is only opened automatically if the access path from the PG/PC to the S7 CPU has to be reset.

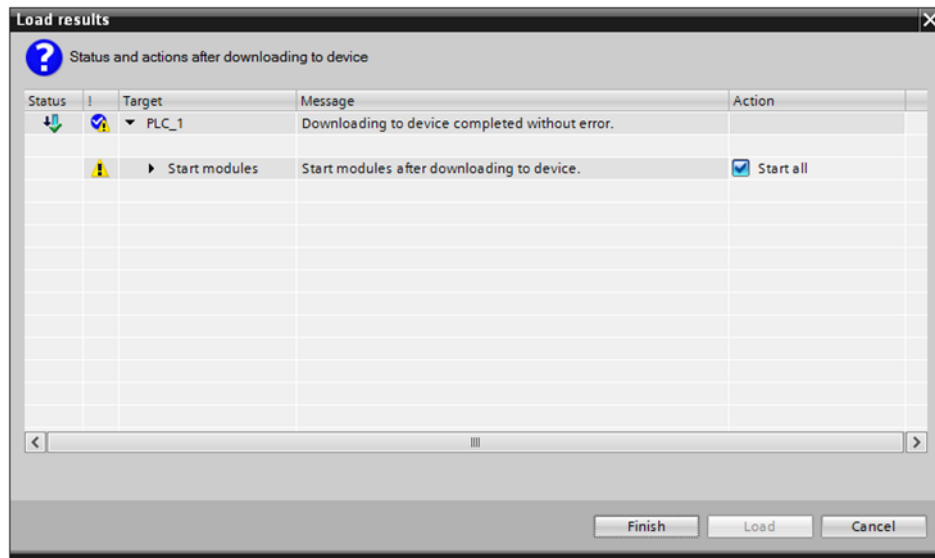
6. In the "Extended download to device" dialog, make the following settings to access the S7-CPU via TCP/IP:
 - Type of PG/PC interface: PN/IE
 - PG/PC interface: PG/PC network card
 - Connection to interface/subnet: Subnet of the S7-CPU, e.g. PN/IE_1
7. Select the "Show all compatible devices" option.
8. Click the "Start search" button.
9. Select the S7 CPU as the target device.
10. Click the "Load" button



11. In the "Load preview" dialog, click the "Load" button to start the load process.



12. In the "Load results" dialog, click the "Finish" button to finish loading the results.



2.7.2 HMI operating device

Download the configuration to your HMI operating device or start the simulation.

Note

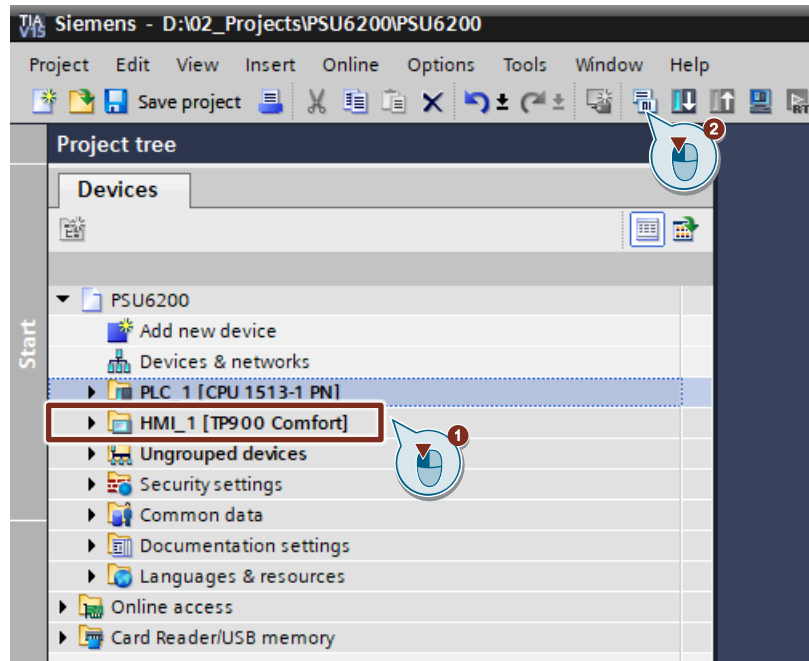
If you only simulate the device, make sure that the PG/PC interface is set correctly.

Follow the following link of the FAQ "Why is communication between the Runtime simulation and the S7-1200/S7-1500 not possible if the PG/PC interface is not set correctly?"

<https://support.industry.siemens.com/cs/ww/en/view/38717202>

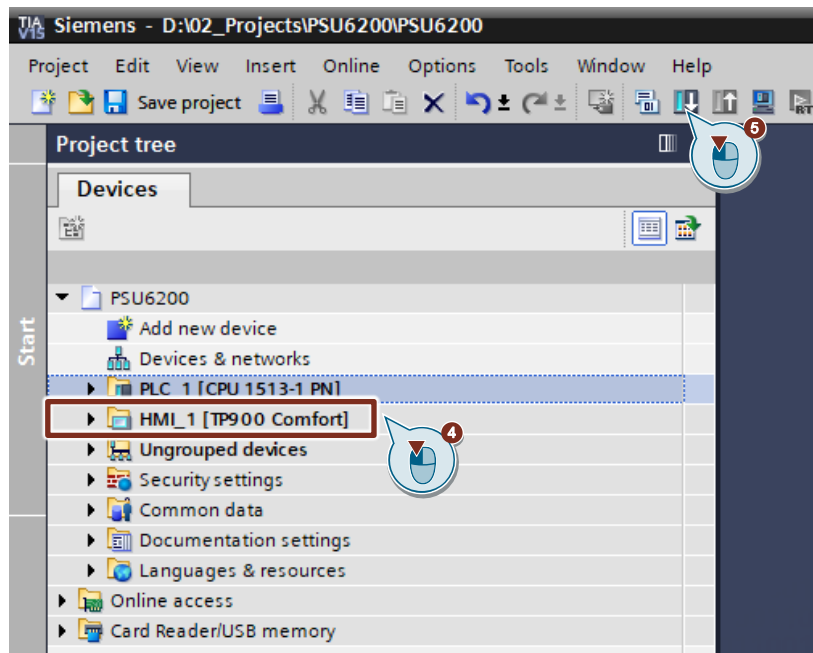
Compiling

1. Select the HMI operating device in the "Project tree".
2. Click on the "Compile" button in the toolbar.



Downloading

1. Make sure that your engineering PC and the HMI operating device are connected to the same subnet.
2. Set the IP address and subnet mask that you have entered in the hardware configuration on the HMI operating device ("Control Panel > Transfer Settings > Properties > PN_X1").
3. Switch the HMI operating device to transfer mode via the "Start Center".
4. Select the HMI operating device in the "Project tree".
5. Click on the "Download to device" button in the toolbar to load the project engineering into the HMI operating device.

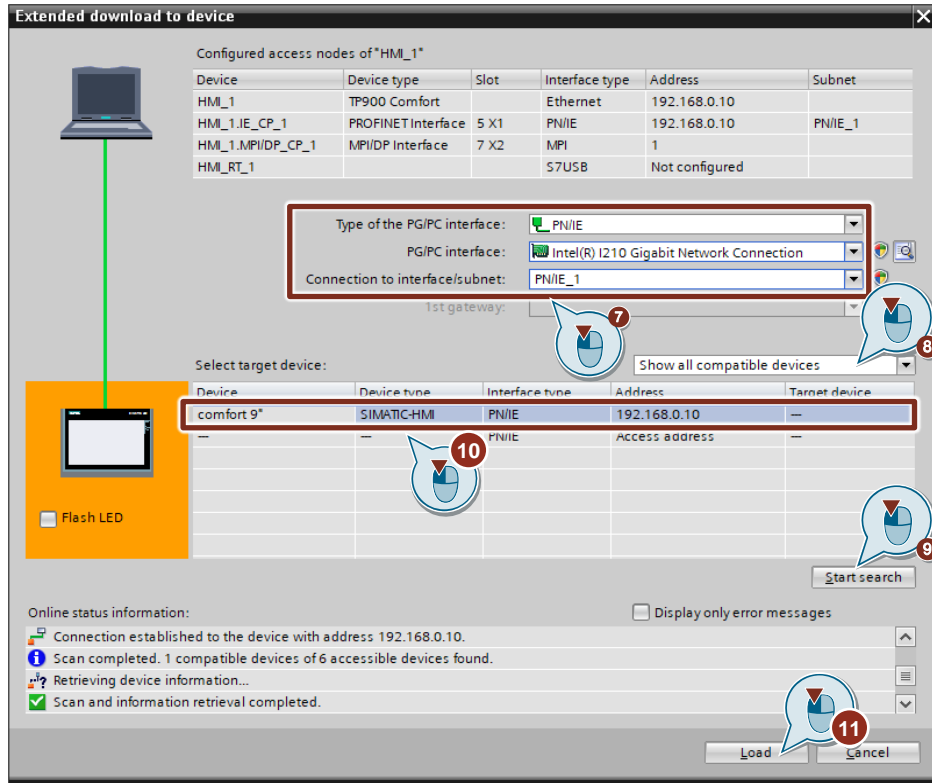


6. The "Extended download to device" or "Load preview" dialog opens automatically.

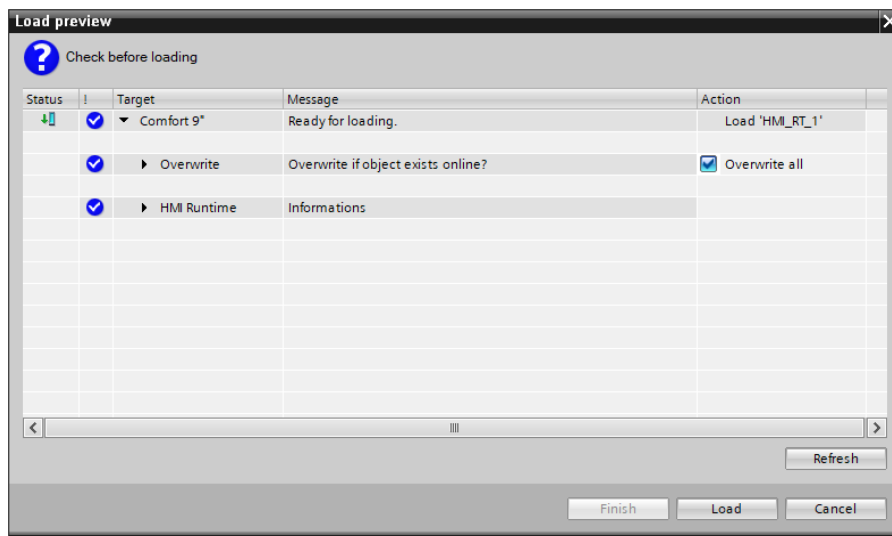
Note

The "Extended download to device" dialog is only opened automatically if the access path from the PG/PC to the HMI operating device has to be reset.

7. In the "Extended download to device" dialog, make the following settings to access the HMI device via TCP/IP:
 - Type of PG/PC interface: PN/IE
 - PG/PC interface: PG/PC network card
 - Connection to interface/subnet: Subnet of the HMI, e.g. PN/IE_1
8. Select the "Show all compatible devices" option.
9. Click the "Start search" button.
10. Select the HMI device as target device.
11. Click the "Load" button.



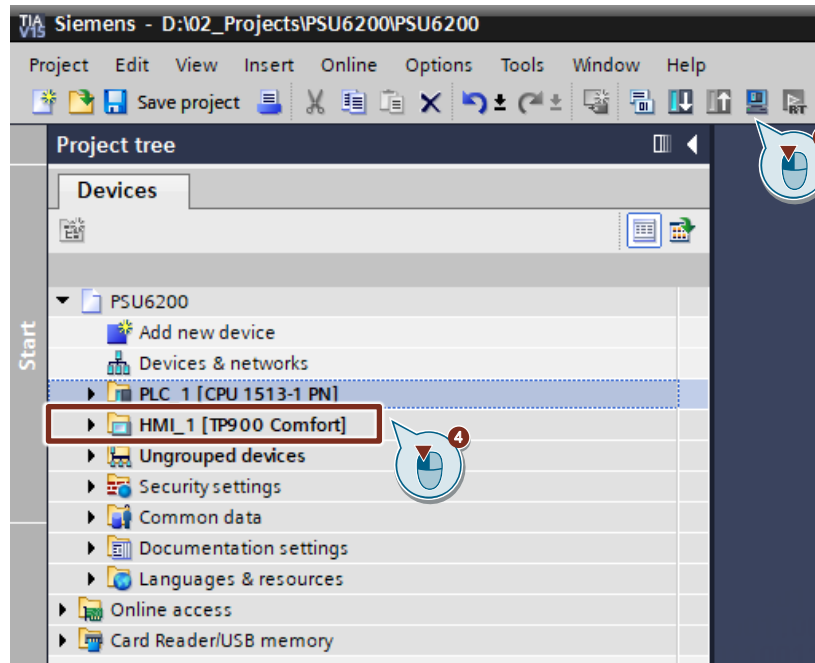
12. In the "Load preview" dialog, click the "Load" button to start the load process.



13. The Runtime starts on the HMI device, automatically.

Start simulation

1. Make sure that your engineering PC and the S7 CPU are connected to the same subnet.
2. Open the "Control Panel" of the Engineering PC.
3. Under "Set PG/PC Interface (32-bit)", select the network card via which the visualization is to communicate with the S7 CPU.
4. Select the HMI operating device in the project navigation.
5. Click on the "Start simulation" button in the toolbar.



6. The runtime simulation is started.

3 Operation

3.1 Faceplate "fpComAdvDetailsS7xy00PSU6200"

3.1.1 Overview

Figure 3-1

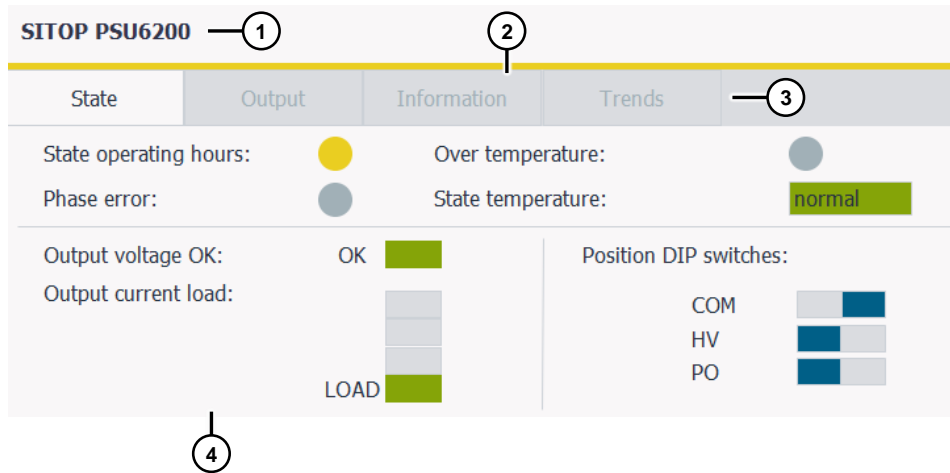


Table 3-1

No.	Area	Description
1	Header area	Displays "SITOP + <Device name>" as header
2	Status area	Status line for colored display of the current device status
3	Navigation area	Navigation between the four data areas <ul style="list-style-type: none"> • Status: Device data with status information • Output: Device data to the output of the power supply • Information: General device information • Trends: Trend displays with current and voltage value of the power supply output
4	Information area	Area to display the selected device information in the navigation area

3.1.2 Header area

In the heading area "SITOP + <device name>" is displayed as the heading of the faceplate. If more than one screen faceplate is used, you know immediately by the device name in the header which device information is shown.

3.1.3 Status area

The status area displays the following states:

- Device shutdown due to overtemperature
- Phase error
- Utilization of nominal output current
- Device temperature status
- Device operating hours status
- Output voltage status

The table below describes the color states in the status area.

Table 3-2

Status	Description	Status bar		
		Color (R,G,B) Background	Color (R,G,B) Frame	Flashing
Device temperature status	Normal	133,164,8	133,164,8	No
	High	234,206,33	234,206,33	No
	derating	234,206,33	234,206,33	No
	Overtemperature	202,51,51	202,51,51	No
Utilization of nominal output current	Iout < 30 %	133,164,8	133,164,8	No
	Iout = [30-60 %]	133,164,8	133,164,8	No
	Iout = [61-90 %]	133,164,8	133,164,8	No
	Iout > 90 %	234,206,33	234,206,33	No
Device operating hours status	Operating hours < 90 % of nominal value	133,164,8	133,164,8	No
	Operating hours > 90% of nominal value	234,206,33	234,206,33	No
Phase error	No	161,176,183	161,176,183	No
	Yes	202,51,51	202,51,51	No
Device shutdown due to overtemperature	No	161,176,183	161,176,183	No
	Yes	202,51,51	202,51,51	Yes
Output voltage status	Not OK	161,176,183	161,176,183	No
	OK	133,164,8	133,164,8	No
Communication status	OK	161,176,183	161,176,183	No
	Not OK	202,51,51	202,51,51	No

3.1.4 Navigation area

The device information for the SITOP PSU6200 is divided into four different topic areas in the navigation area.

- Status: Parameters for device status
- Output: Output parameters
- Information: Informative device information
- Trends: Voltage and current trend of the output

Use the buttons to switch between the topic areas. Depending on the button selected, the device information displayed in the information area changes.

The color of the button indicates which topic area was selected.

Table 3-3

Active topic area	Inactive topic area
Status	Status

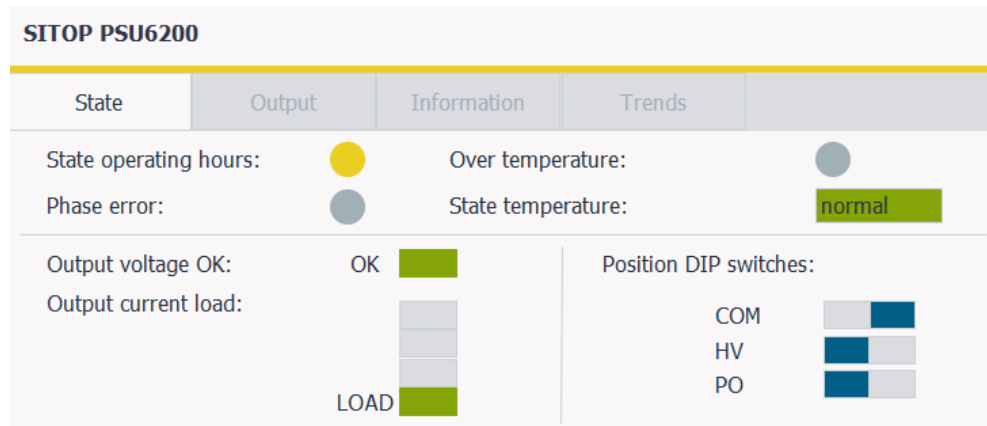
3.1.5 Information area

The information area clearly displays all device data of the SIOTP PSU6200 via status displays (circuits), switches, I/O fields and trend displays. Depending on the topic area, these objects are used in different ways.

Status

The "State" information area shows the status parameters of the PSU6200.

Figure 3-2



The following status parameters are displayed:

- Output voltage OK
- Operating hours status
- Output utilization
 - I_{out} < 30 %
 - I_{out} = [30-60 %]
 - I_{out} = [61-90 %]
 - I_{out} > 90 %
- Phase error
- DIP switch position
 - COM tag
 - HV tag
 - PO tag
- Shutdown due to overtemperature
- Temperature status
 - "normal"
 - "high"
 - "derating"
 - "Over temperature"

When coloring the status elements, the color constellation from the following applies [Table 3-2](#).

Output

The information area "Output" displays the parameters of the output of the SITOP PSU6200.

Figure 3-3

SITOP PSU6200			
State	Output	Information	Trends
Output voltage:	23.9 V		
Output current:	0 A		
Counter Glitch overvoltage:	0	Counter Glitch undervoltage:	0

The following output parameters are displayed:

- Output voltage
- Output current
- Counter interference pulse overvoltage
- Counter interference pulse undervoltage

Information

The information area "Information" displays the informative and static device data of the SITOP PSU6200.

Figure 3-4

SITOP PSU6200			
State	Output	Information	Trends
Device name:	PSU6200	Device type:	24 V / 20 A
Serial number:	Q6/L0	Product state:	0
Article number:	6EP3336-7SB00-3AX0	Firmware version:	0
Operating hours:	< 1000 h		

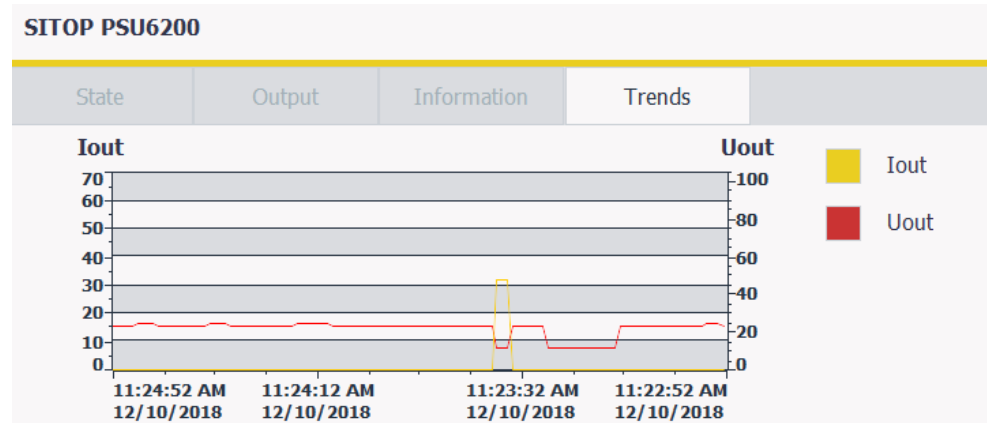
You see the following device information:

- Device name
- Serial number
- Article number
- Operating hours
- Device type
- Product version
- Firmware version

Trends

The information area "Trends" shows a trend display which graphically displays the output voltage and the output current of the SITOP PSU6200.

Figure 3-5



By default, the trend lines have the following colors:

- Current ("Iout"): yellow
- Voltage (Uout): red

3.2 Faceplate "fpComAdvOverviewS7xy00PSU6200"

3.2.1 Overview

Figure 3-6

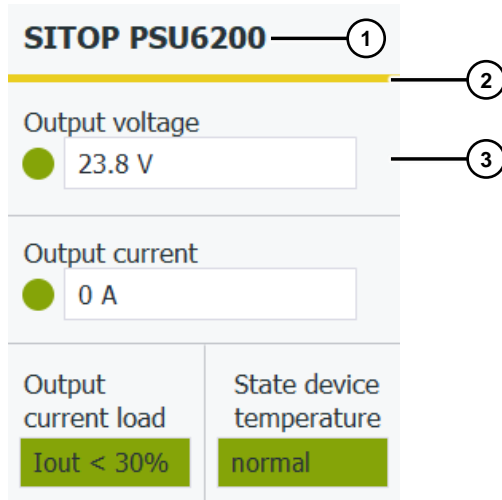


Table 3-4

No.	Area	Description
1	Header area	Displays "SITOP + <Device name>" as header
2	Status area	Status line for colored display of the current device status
3	Information area	Area for displaying the most important device data: <ul style="list-style-type: none"> • Output voltage • Output current • Output utilization • Device temperature status

The following faceplates provide an overview of the most important device data:

- "fpComAdvOverviewS71500PSU6200"
- "fpComAdvOverviewS71200PSU6200"
- "fpComAdvOverviewS7300PSU6200"

Click on the faceplate to display further device data and trends of the output voltage and output current in the following faceplates:

- "fpComAdvS71500PSU6200"
- "fpComAdvS71200PSU6200"
- "fpComAdvS7300PSU6200"

4 Error handling

4.1 Status display FB "LPSU6200_SignalEvaluationS7xy00"

Table 4-1

Value at output "status"	Meaning	Help/Note
16#8001	No signal change was detected at input "impulse" for at least 6 seconds.	<ul style="list-style-type: none"> • Check whether the diagnostics interface "COM" is connected to the digital input. • Check whether you indicate the right digital input on the "pulses" input. • Check that the input voltage is connected to the PSU6200.
16#8002	Cycle time of 10 ms exceeded	Call the function block with a maximum of 10 ms.
16#8003	Error when evaluating the frame content	High level and low level could not be properly detected so that an error occurred when evaluating the frame content.
16#8004	Frame length outside the value range	Value range: 3275 ms to 3625 ms
16#8005	Device type not detected	Permissible device type: PSU6200
16#8006	Frame unknown	Permissible content type: 0 to 3
16#8007	Communication error or device replacement	Article number change detected

4.2 Display of hashes ("#####") in the faceplate

Figure 4-1

SITOP #####				
State	Output	Information	Trends	
Device name:	#####	Device type:	#### / ####	
Serial number:	#####	Product family:	AC/DC	
Article number:	#####	Product version:	####	
Betriebsstunden:	#####	Software version:	####	

If hashes ("#####") are displayed in the faceplate for device data, there is a communication problem between the HMI operating device and the S7 CPU.

Check the following items:

- Check the connection cables for proper wiring and damage.
- For WinCC RT Advanced, check whether the PG/PC interface was correctly configured. If the PG/PC interface is set incorrectly, communication between the runtime simulation and the S7-1500 / S7-1200 is not possible. (see FAQ [38717202](#)). You find the settings under "Control Panel > Set PG/PC interface".
- Check whether the devices are in the same network.
- Check if the IP address of a device is blocked in the network router.

5 Appendix

5.1 Service und support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:

<https://support.industry.siemens.com/>

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts. Please send queries to Technical Support via Web form:

www.siemens.com/industry/supportrequest

SITRAIN – Training for Industry

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that's tailored to the customer's specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:

www.siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:

<https://support.industry.siemens.com/cs/sc>

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for Apple iOS, Android and Windows Phone:

<https://support.industry.siemens.com/cs/ww/en/sc/2067>

5.2 Links and Literature

Table 5-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to the article page of the application example https://support.industry.siemens.com/cs/ww/en/view/109760217
\3\	SIMATIC STEP 7 Basic/Professional V15 und SIMATIC WinCC V15 https://support.industry.siemens.com/cs/ww/en/view/109755202

5.3 Change documentation

Table 5-2

Version	Date	Change
V1.0	01/2019	First version
V1.1	02/2019	Update for S7-300 / S7-400
V1.2	05/2019	Additional PLC data type to display status and default values in case of communication with diagnostic interface "COM" is not active