SCE Training Curriculum
for Integrated Automation Solutions
Totally Integrated Automation (TIA)

Siemens Automation Cooperates with Education

TIA Portal Module 060-010

PROFIBUS with Master CPU 315F-2 PNDP and Slave ET 200S

Cooperates with Education

Automation

Suitable SCE trainer packages for these training curriculums:

SIMATIC controllers

- **SIMATIC S7-300 with CPU 314C-2PN/DP**
  Order no.: 6ES7314-6EH04-4AB3
- **SIMATIC S7-300 with CPU 314C-2PN/DP (upgrade)**
  Order no.: 6ES7314-6EH04-4AB4
- **SIMATIC S7-300 with CPU 315F-2PN/DP**
  Order no.: ES7315-2FH14-4AB1
- **SIMATIC ET 200S with CPU IM151-8 F PN/DP**
  Order no.: 6ES7151-8FB00-4AB1

SIMATIC STEP 7 Software for Training

- **SIMATIC STEP 7 Professional V11 - Single license**
  Order no.: 6ES7822-1CC01-4YA5
- **SIMATIC STEP 7 Professional V11 - Classroom license (up to 12 users)**
  Order no.: 6ES7822-1AA01-4YA5
- **SIMATIC STEP 7 Professional V11 - Upgrade license (up to 12 users)**
  Order no.: 6ES7822-1AA01-4YE5
- **SIMATIC STEP 7 Professional V11 - Student license (up to 20 users)**
  Order no.: 6ES7822-1AC01-4YA5

Please note that these training packages are replaced with successor packages when necessary.
An overview of the currently available SCE packages is provided under: siemens.com/sce/tp

Advanced Training

For regional Siemens SCE advanced training courses, please get in touch with your regional SCE contact person siemens.com/sce/contact

Additional information regarding SCE

siemens.com/sce

Information regarding usage

This SCE training curriculum for the end-to-end automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)" specifically for training purposes for public educational facilities and R&D facilities. Siemens AG does not guarantee the contents.

This document is to be used only for initial training on Siemens products/systems; i.e., it can be copied entirely or partially and given to those being trained for usage within the scope of their training. Distributing or copying this document and sharing its contents is permitted within public training and advanced training facilities for training purposes.

Exceptions require written permission from the Siemens AG contact person: Roland Scheuerer roland.scheuerer@siemens.com.

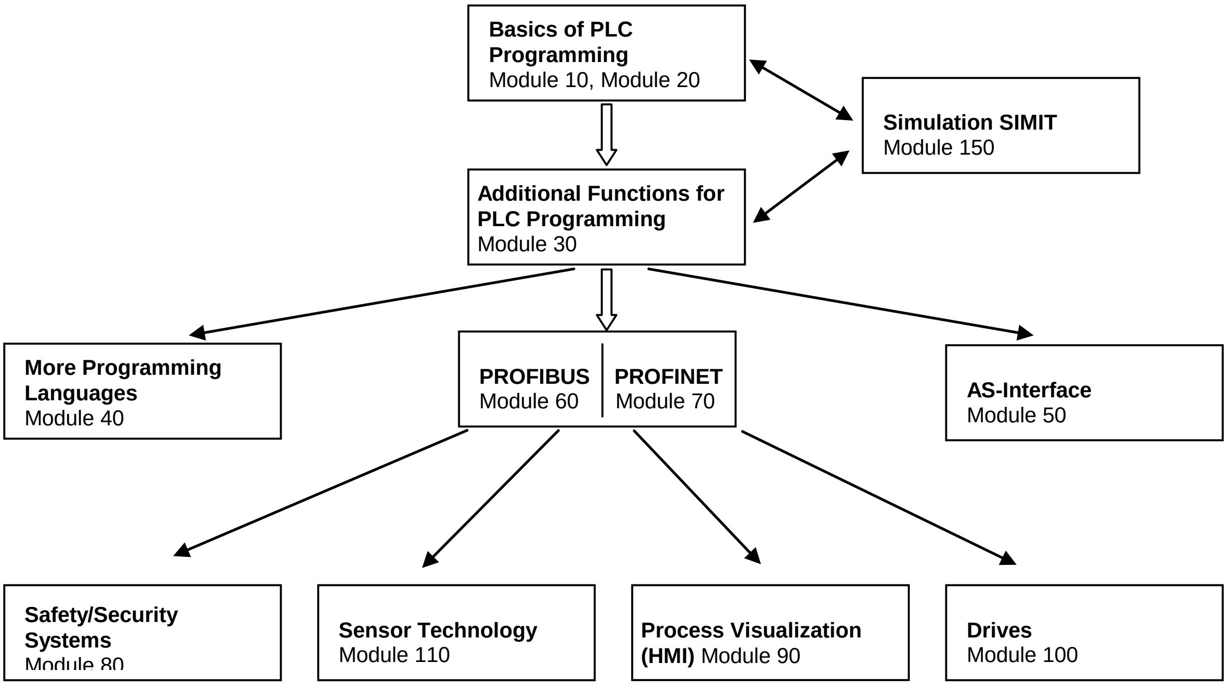
Offenders will be held liable. All rights including translation are reserved, particularly if a patent is granted or a utility model or design is registered.
Usage for industrial customer courses is expressly prohibited. We do not consent to the training curriculum being used commercially.

We would like to thank Michael Dziallas Engineering Corporation and all other involved persons for their support during the preparation of this training curriculum.

PAGE:

1. PREFACE

Regarding its content, module SCE_EN_060-010 is part of the training unit **‘PROFIBUS’**.



Training Objective:

This module provides the reader with information on how to commission the CPU 315F-2PN/DP as master and the ET 200S as slave on PROFIBUS. The module demonstrates the general procedure based on a brief example

Requirements:

To successfully work through this module, the following knowledge is assumed:

- Proficiency in working with Windows
- Advanced knowledge of PLC programming with the TIA Portal (e.g. module SCE_EN_020-010_R1201_Startup Programming with SIMATIC S7-300 and module SCE_EN_030...)

Hardware and software needed

1 PC Pentium 4, 1.7 GHz 1 (XP) – 2 (Vista) GB RAM, free disk storage approx. 2 GB operating system Windows XP Professional SP3/Windows 7 Professional/Windows 7 Enterprise/Windows 7 Ultimate/Windows 2003 Server R2/Windows Server 2008 Premium SP1, Business SP1, Ultimate SP1

2 Software: STEP 7 Professional V11 SP1 (Totally Integrated Automation (TIA) Portal V11)

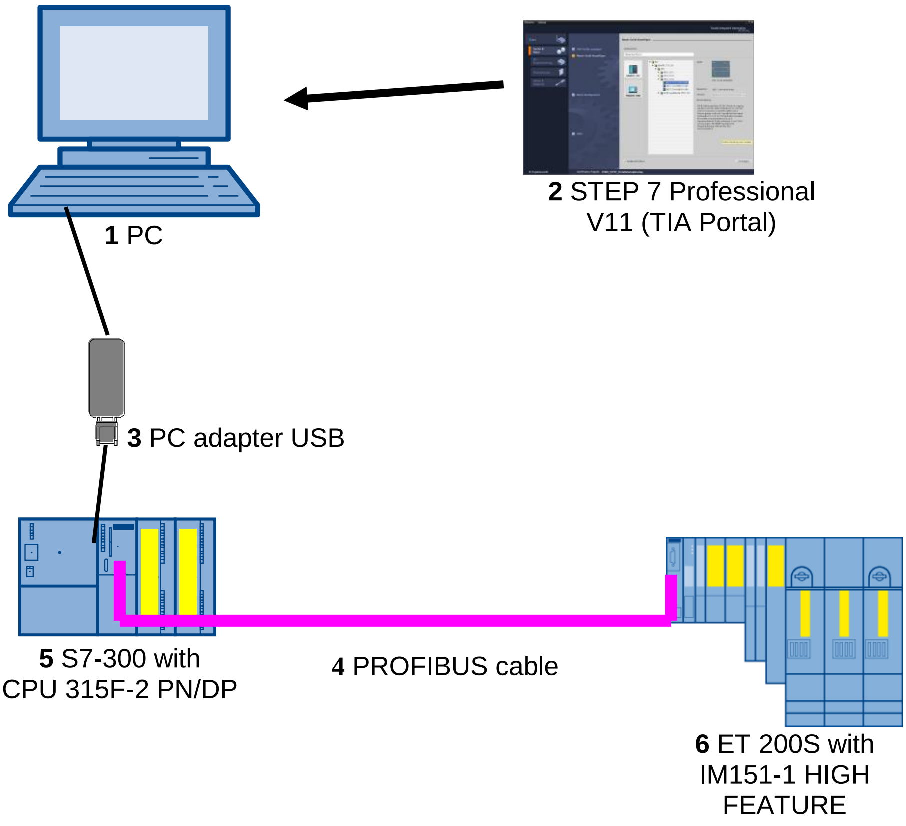
3 PROFIBUS interface for the PC, e.g. PC Adapter USB (alternatively, connection via Ethernet interface is also possible)

4 PROFIBUS cable with 2 PROFIBUS connectors (with PG socket at rear!)

5 PLC SIMATIC S7-300; for example, CPU 315F-2PN/DP with 16DI/16DO signal module. The inputs have to be brought out to a panel.

6 Distributed I/O device ET 200S for PROFIBUS with 8 digital inputs and 8 digital outputs.

- Interface module: IM 151-1 HIGH FEATURE
- Power module: PM-E RO DV24V / 8A
- Electronic module 4DI HF DC24V (the inputs must be brought out to a panel)
- Electronic module 4DI HF DC24V (the inputs must be brought out to a panel)
- Power module: PM-E RO DV24V / 8A
- Electronic module 4DO HF DC24V/0.5A
- Electronic module 4DO HF DC24V/0.5A


1 PC
2 STEP 7 Professional V11 (TIA Portal)
3 PC adapter USB
5 S7-300 with CPU 315F-2 PN/DP
4 PROFIBUS cable
6 ET 200S with IM151-1 HIGH FEATURE

2. NOTES ON USING THE CPU 315F-2 PN/DP

The CPU 315F-2 PN/DP is a CPU which is supplied with 2 integrated interfaces.

- The first interface is a combined MPI/PROFIBUS DP interface which can be used as master or slave on PROFIBUS DP to connect distributed IO/field devices with very fast response times. The CPU can be programmed via MPI or via PROFIBUS DP.
- The second interface is an integrated PROFINET interface with 2 ports. This allows you to use the CPU as PROFINET IO controller to operate distributed IO devices on PROFINET. You can also use this interface to program the CPU.
- Failsafe IO devices can also be used with both interfaces.
- In addition, CPU 315F-2 PN/DP features a web server that allows remote diagnostics to be conducted using any browser, even without STEP 7 software.

Notes:

- In this module, the CPU 315F-2 PN/DP is used as master on PROFIBUS.
- A Micro Memory Card is required in order to operate this CPU.
- Parameters for the addresses of the input and output modules can be assigned for this CPU.

3. NOTES ON USING THE ET200S WITH IM 151-1 HIGH FEATURE

SIMATIC ET 200S is a highly modular, distributed IO device. It can be operated with various interface modules. Here is a selection of these:

IM 151-1 BASIC, IM 151-1 STANDARD and IM 151-1 FO STANDARD for the connection of max. 63 IO modules (all types, except PROFIsafe) to PROFIBUS DP; alternatively, bus connection with RS 485 D-Sub connector or using integrated fiber-optic connection.

IM 151-1 HIGH-FEATURE for connection of max. 63 IO modules (all types, including isochronous mode for PROFIsafe) to PROFIBUS DP; bus connection with RS485 D-Sub connector.

IM 151-3 PN for connection of max. 63 IO modules (all types, including isochronous mode for PROFIsafe) to PROFINET IO controller; bus connection with RJ45 connector.

IM 151-3 PN HF (HIGH FEATURE) for connection of max. 63 IO modules (all types, including isochronous mode for PROFIsafe) to PROFINET IO controller; bus connection with 2x RJ45 connectors.

IM151-8 F-CPU PN/DP, IM 151-7/F-CPU, IM 151-7/CPU and IM 151-7/CPU FO for connection of max. 63 IO modules (all types, PROFIsafe only with IM151-8 F-CPU PN/DP and IM151-7/F-CPU) to PROFINET or PROFIBUS DP (alternatively, bus connection with RS 485 D-Sub connector or using integrated fiber-optic connection). With integrated CPU to preprocess the process data.

The following IO modules, for example, can be used here:

Power modules to group load and encoder supply voltages separately and to monitor these voltages

Digital electronic modules to connect digital sensors and actuators

Analog electronic modules to connect analog sensors and actuators

Sensor module to connect IQ-Sense sensors

Technology modules Electronic modules with integrated technological functions, such as counting, positioning, data exchange, etc.

Frequency converter and motor starter modules

Notes:

- In this module, the IM151-1 HF interface module is used as PROFIBUS slave.
- The PROFIBUS address is set in binary code at 8 switches on the IM151-1 HF interface module. The bottom switch must be set to OFF. A number is assigned to each of the other switches. These numbers are added together to make up the PROFIBUS station address. A modified setting for the PROFIBUS address is only applied when power is restored. For this reason, the IM151-1 HF interface module must be switched off and then on again.

4. COMMISSIONING THE PROFIBUS (MASTER CPU 315F-2 PN/DP / SLAVE ET200S)

Below, a description is provided on how to commission a PROFIBUS network with the CPU 315F-2 PN/DP as master and the ET 200S as slave.
To test the configuration, a program is written in which an indicator light P1 is activated when the S1 button is pressed. Another indicator light P2 is activated if two buttons, S1 and S2, are pressed at the same time.

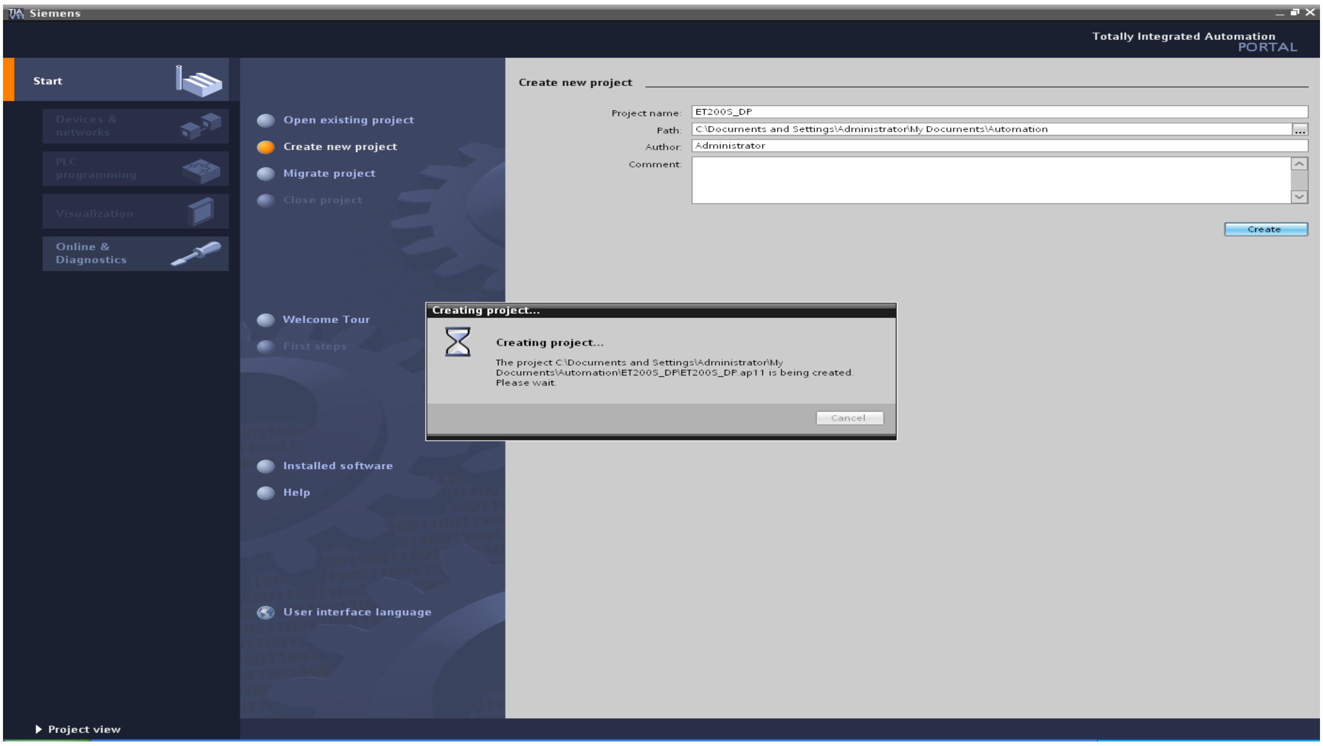
Assignment list:

%I0.0	S1	Selector switch S1 (NO)
%I0.4	S2	Selector switch S2 (NO)
%Q0.0	P1	Indicator light P1
%Q0.4	P2	Indicator light P2

4.1. Creating a project and configuring hardware

1. The central tool is the **‘Totally Integrated Automation Portal’**, which is opened with a double-click. (→ TIA Portal V11)

2. Programs for SIMATIC S7-300 are managed in projects. Such a project is now created in the portal view. (→ Create new project → ET 200S_DP → Create)



Create new project

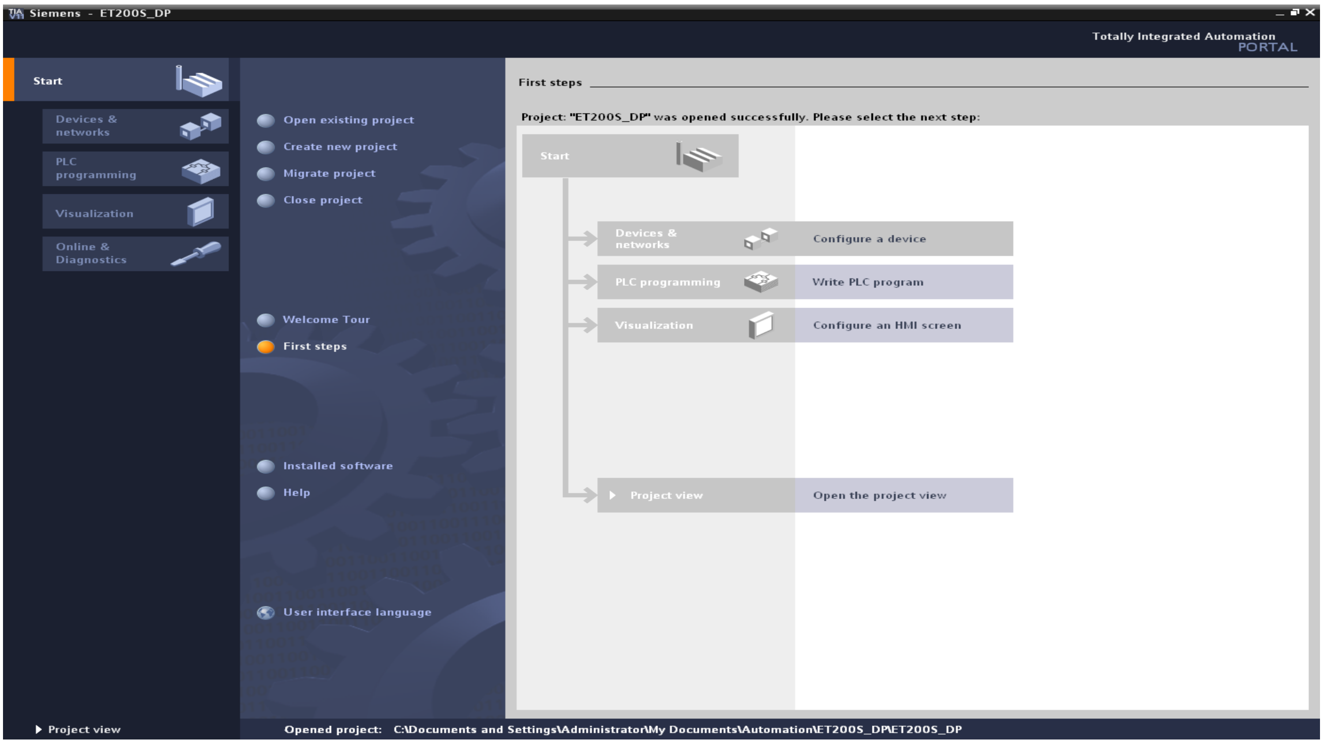
Project name: ET200S_DP
Path: C:\Documents and Settings\Administrator\My Documents\Automation
Author: Administrator
Comment:

Creating project...

Creating project...

The project C:\Documents and Settings\Administrator\My Documents\Automation\ET200S_DP\ET200S_DP.ap11 is being created. Please wait.

3. **'First steps'** for configuring are now suggested. We first want to **'Configure a device'**.
(→ First steps → Configure a device)

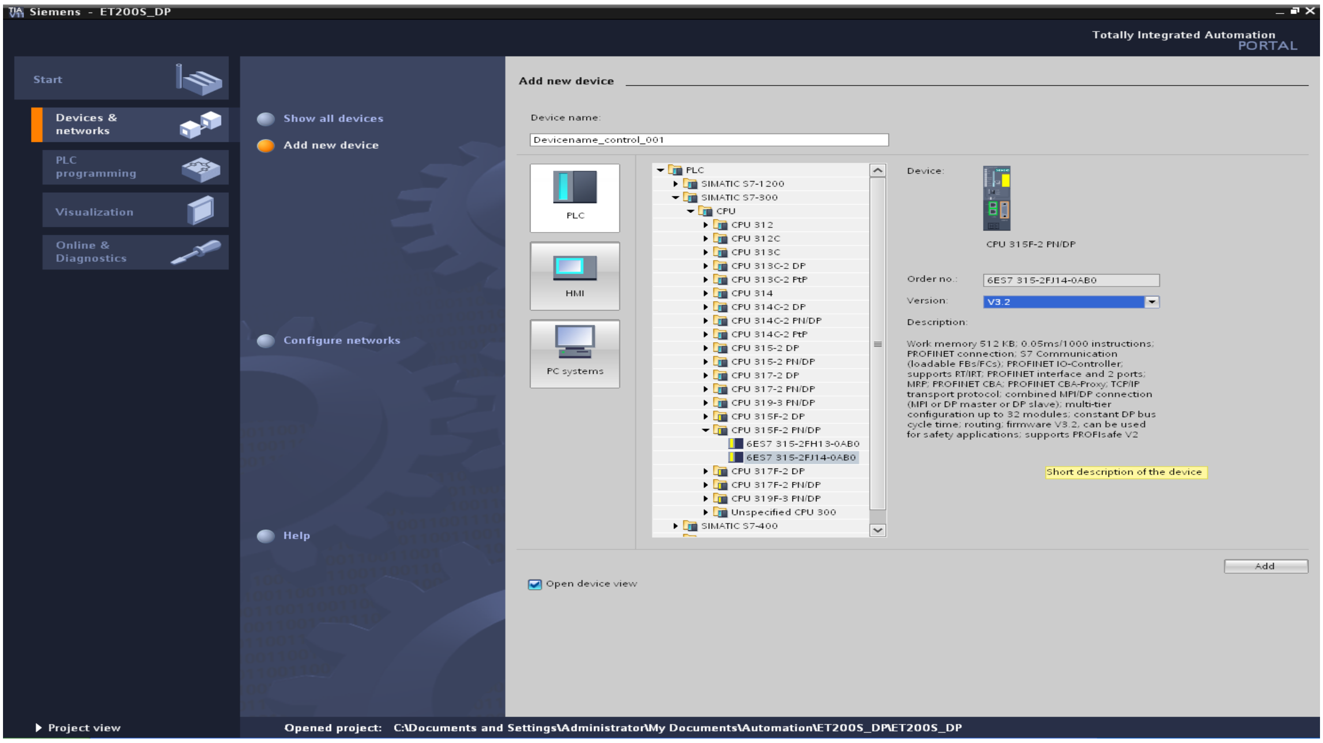


First steps

Project: "ET200S_DP" was opened successfully. Please select the next step:

Opened project: C:\Documents and Settings\Administrator\My Documents\Automation\ET200S_DP\ET200S_DP

4. The next step is **‘Add new device’** with the **‘Device name control 001’**.
   Choose the **‘CPU 315F-2 PN/DP’** with the appropriate order number from the catalog. (→ Add new device → Control 001 → PLC → SIMATIC S7-300 → CPU → CPU 315F-2 PN/DP → 6ES7 315-2FJ14-0AB0 → V3.2 → Add)



Add new device

Device name: Devicename_control_001

Device:

Order no.: 6ES7 315-2FJ14-0AB0

Version: V3.2

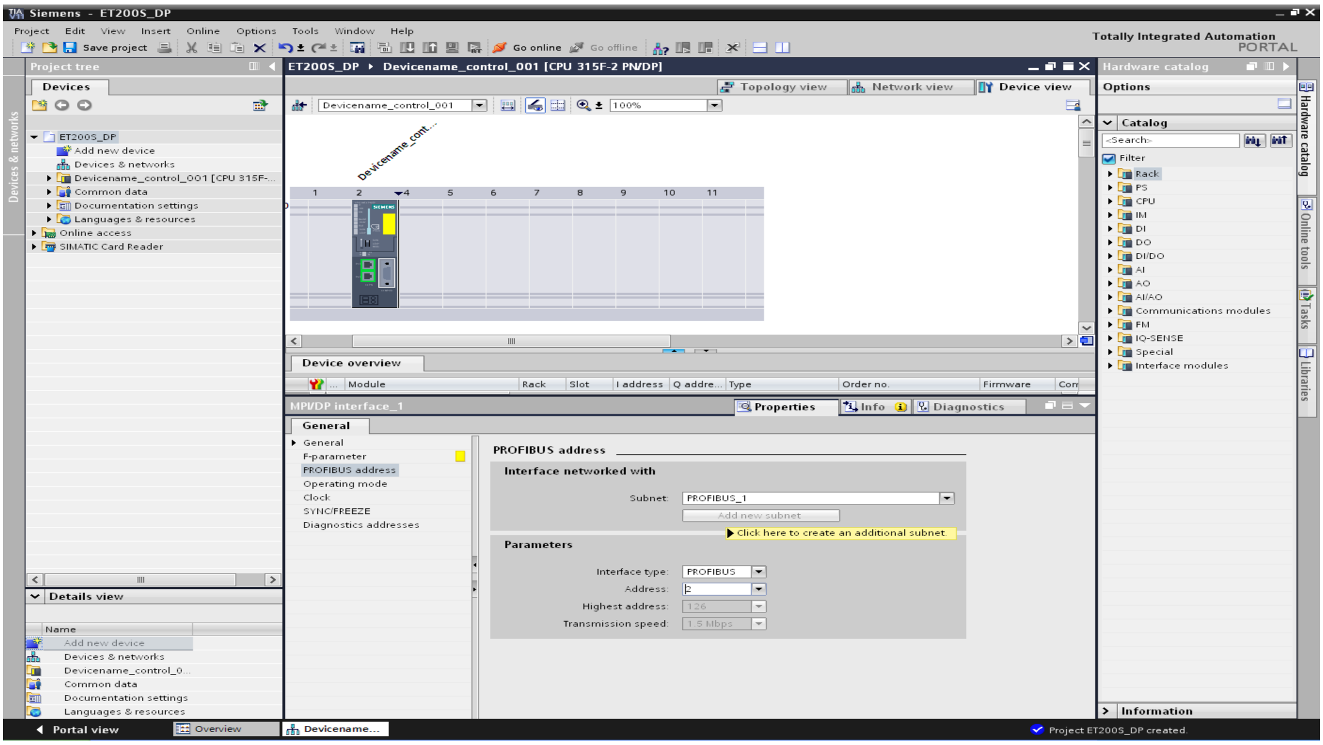
Description:

Work memory 512 KB; 0.05ms/1000 instructions; PROFINET connection; S7 Communication (loadable FBs/FCs); PROFINET IO-Controller; supports RT/IRT; PROFINET interface and 2 ports; MRP; PROFINET CBA; PROFINET CBA-Proxy; TCP/IP transport protocol; combined MPI/DP connection (MPI or DP master or DP slave); multi-tier configuration up to 32 modules; constant DP bus cycle time; routing; firmware V3.2, can be used for safety applications; supports PROFIsafe V2

☑ Open device view

Opened project: C:\Documents and Settings\Administrator\My Documents\Automation\ET200S_DP\ET200S_DP

5. The software now switches automatically to the project view containing the opened hardware configuration in the device view. Additional modules can be added from the hardware catalog (to the right!).
   To ensure that the software will access the correct CPU later, the PROFIBUS address of the CPU must be set. In addition, this interface is connected with a subnet.
   (→ Properties → General → MPI/PROFIBUS address → Add new subnet → PROFIBUS_1 → Interface type: PROFIBUS → address: 2)



PROFIBUS address

Interface networked with

Subnet: PROFIBUS_1

Add new subnet

Click here to create an additional subnet.

Parameters

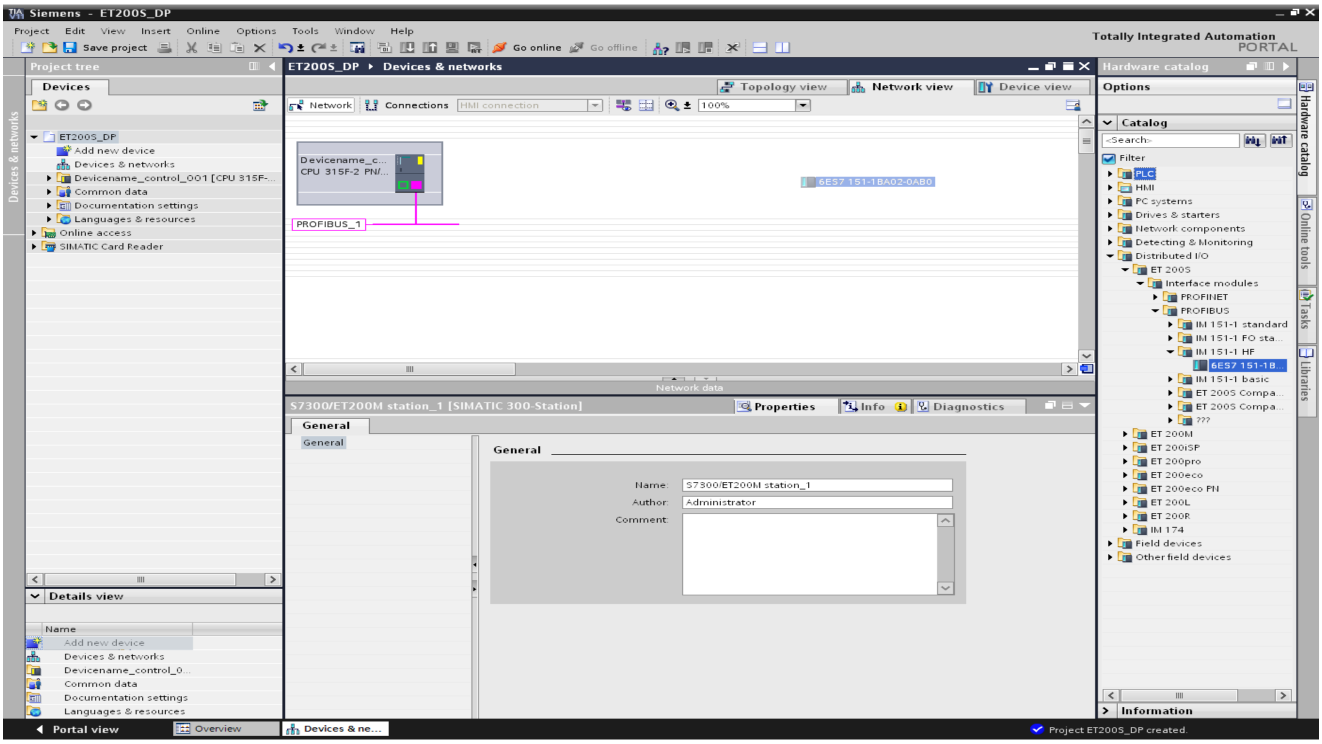
Interface type: PROFIBUS

Address: 2

Highest address: 126

Transmission speed: 1.5 Mbps

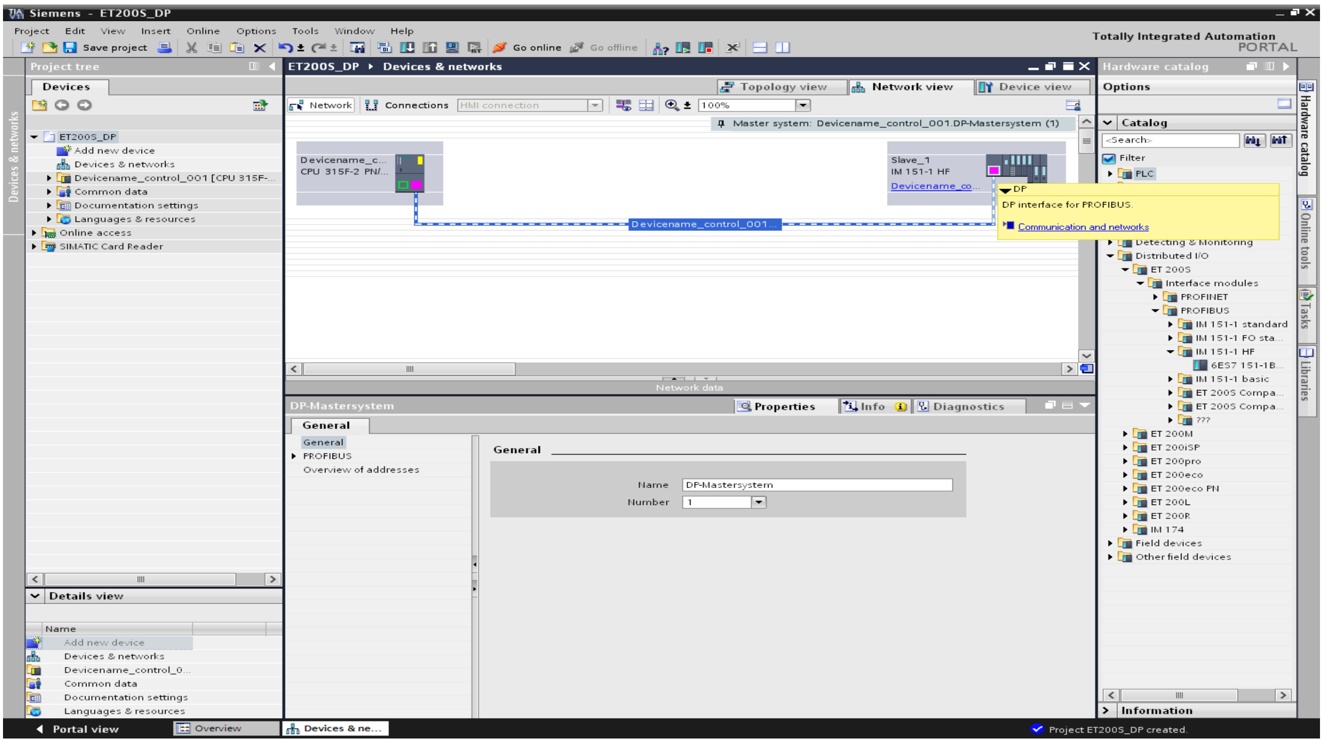
6. To connect the ET 200S with the CPU 315-2 PN/DP, you must go to the **‘network view’**. The relevant PROFIBUS module can be moved to the network view by means of drag-and-drop. (→ Network view → Distributed I/O → ET 200S → Interface modules → PROFIBUS → IM151-1 HF → 6ES7 151-1BA02-0AB0).



General

Name: S7300/ET200M station_1
Author: Administrator
Comment:

7. Then, the DP interface of the CPU 315-2 PN/DP and of the IM 151-1 HF are connected using the mouse. (→ DP → DP)



DP interface for PROFIBUS.

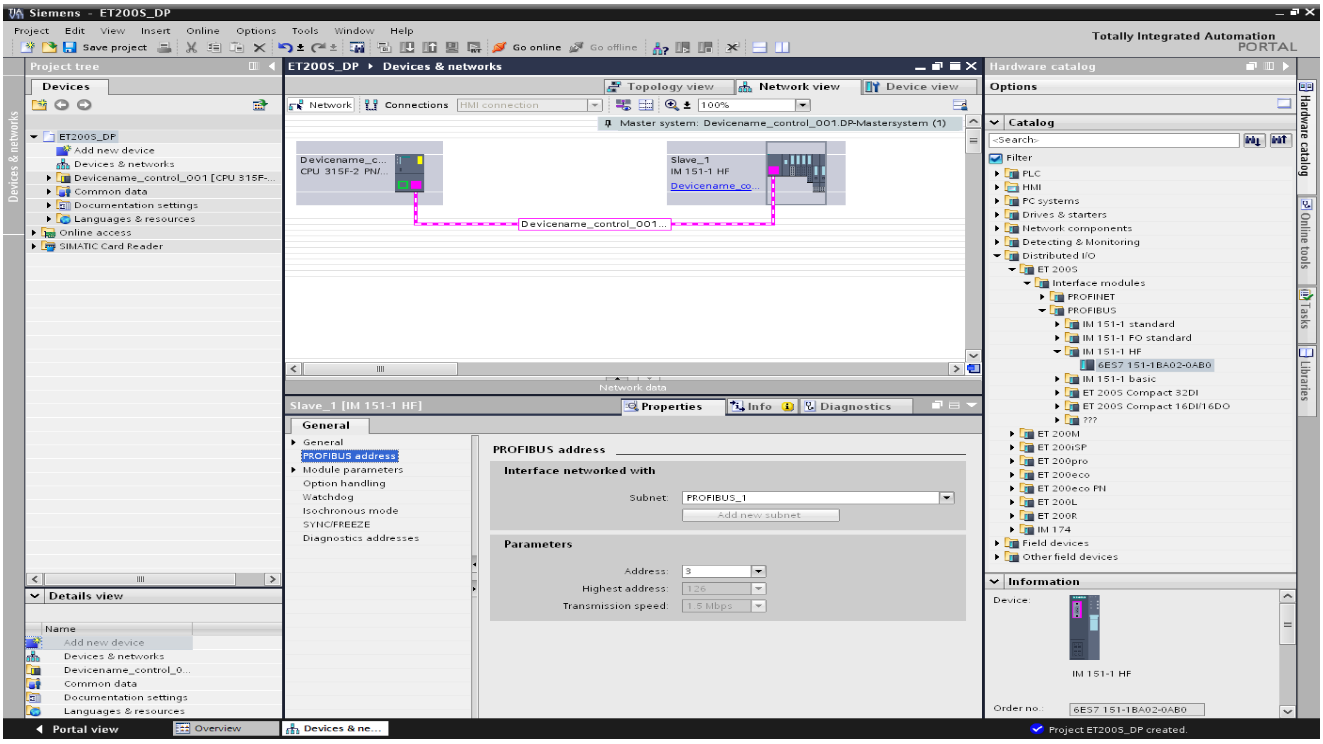
General

Name DP-Mastersystem

Number 1

Project ET200S_DP created.

8. The **'PROFIBUS interface'** is then set in the properties of the **'IM 151-1 HF'** and an address matching the setting at the IM151-1 HF interface module is assigned. (→ IM 151-1 HF → Properties → General → PROFIBUS address → Address: 3)



PROFIBUS address

Interface networked with

Subnet: PROFIBUS_1

Add new subnet

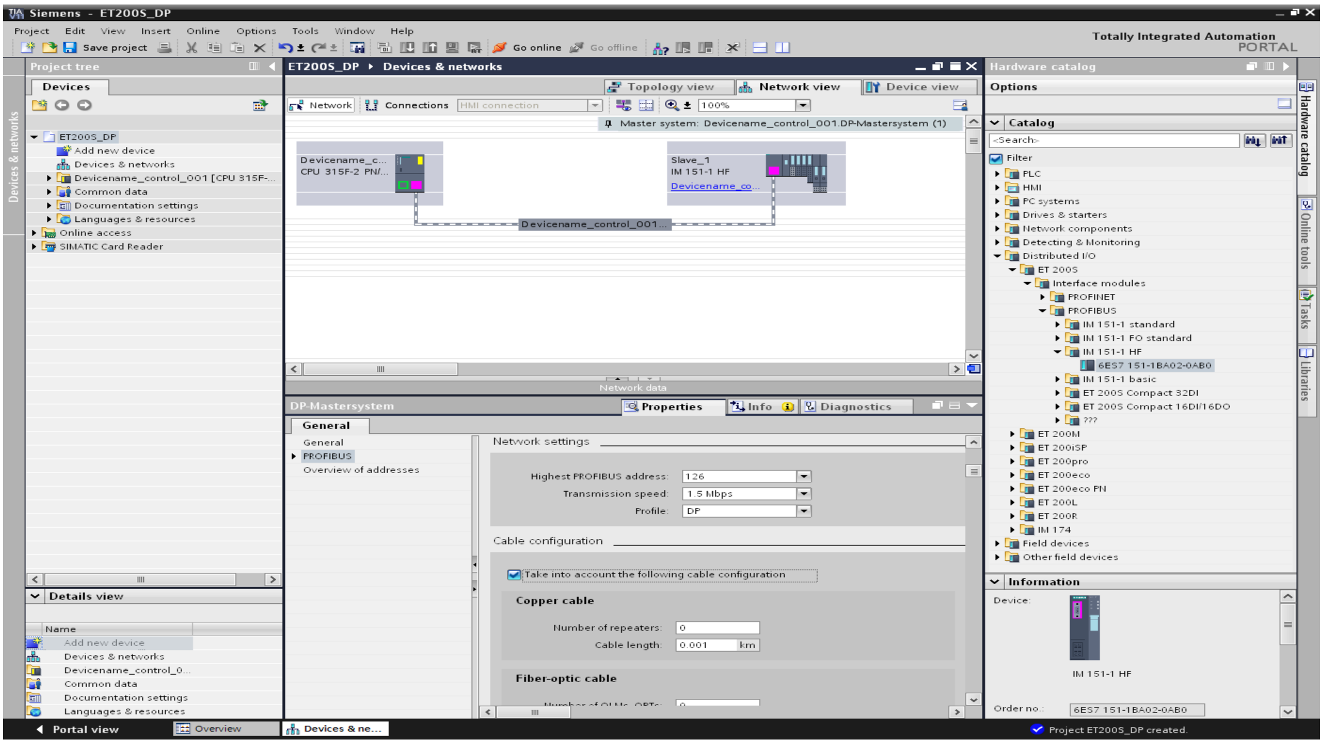
Parameters

Address: 3

Highest address: 126

Transmission speed: 1.5 Mbps

9. Network settings such as **‘Highest PROFIBUS address’**, **‘Transmission speed’** etc. for the entire **‘DP master system’** can be made here. (→ DP master system → Highest PROFIBUS address: 126 → Transmission speed: 1.5 Mbps)



DP-Mastersystem – General – PROFIBUS

Network settings

Highest PROFIBUS address: 126

Transmission speed: 1.5 Mbps

Profile: DP

Cable configuration

☑ Take into account the following cable configuration

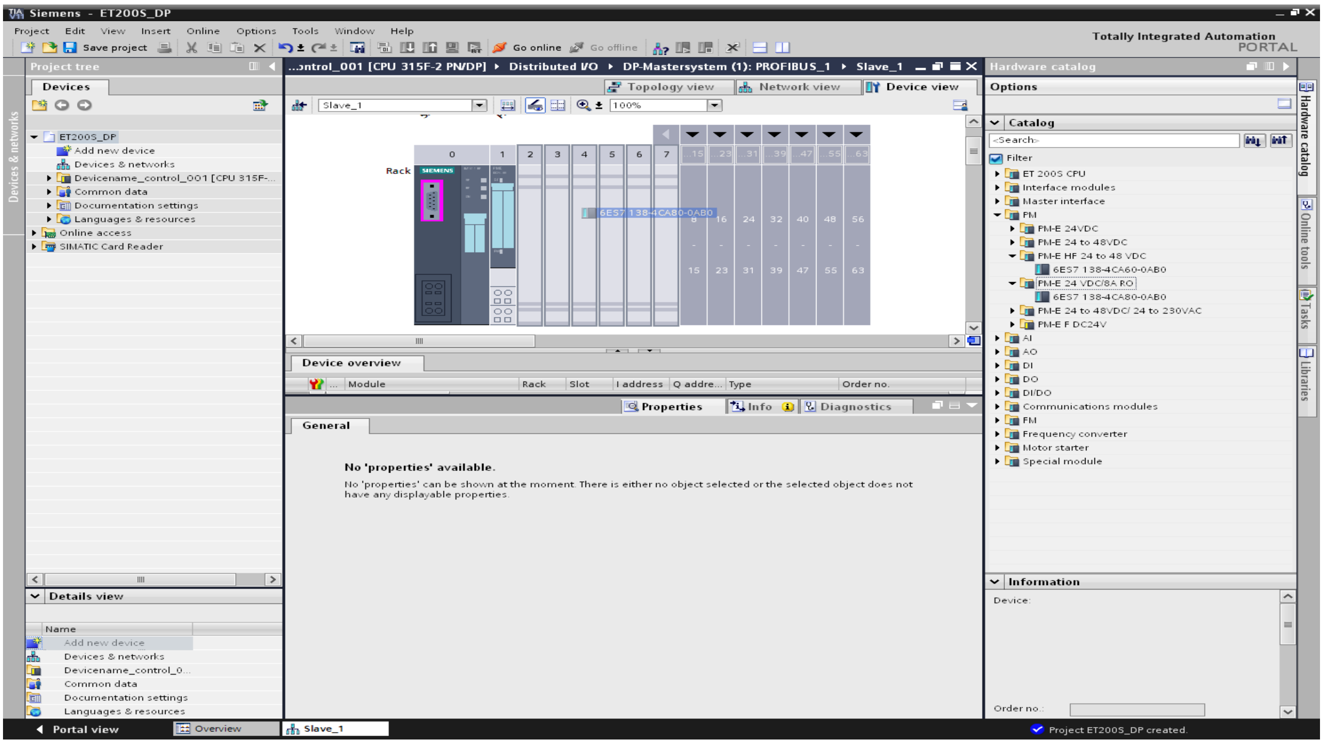
Copper cable

Number of repeaters: 0

Cable length: 0.001 km

Fiber-optic cable

10. In the device view, all other modules which are also inserted in the real ET 200S can now be selected from the hardware catalog and inserted into the configuration table. To do this, click on the name of the corresponding module, keep the mouse button pressed, and drag it to the appropriate column of the configuration table. We start with the 2 power modules **'PM-E DC24V / 8A RO'** which are dragged to slots 1 and 4**.** (→ PM → PM-E DC24V / 8A RO → 6ES7 138-4CA80-0AB0)



No 'properties' available.

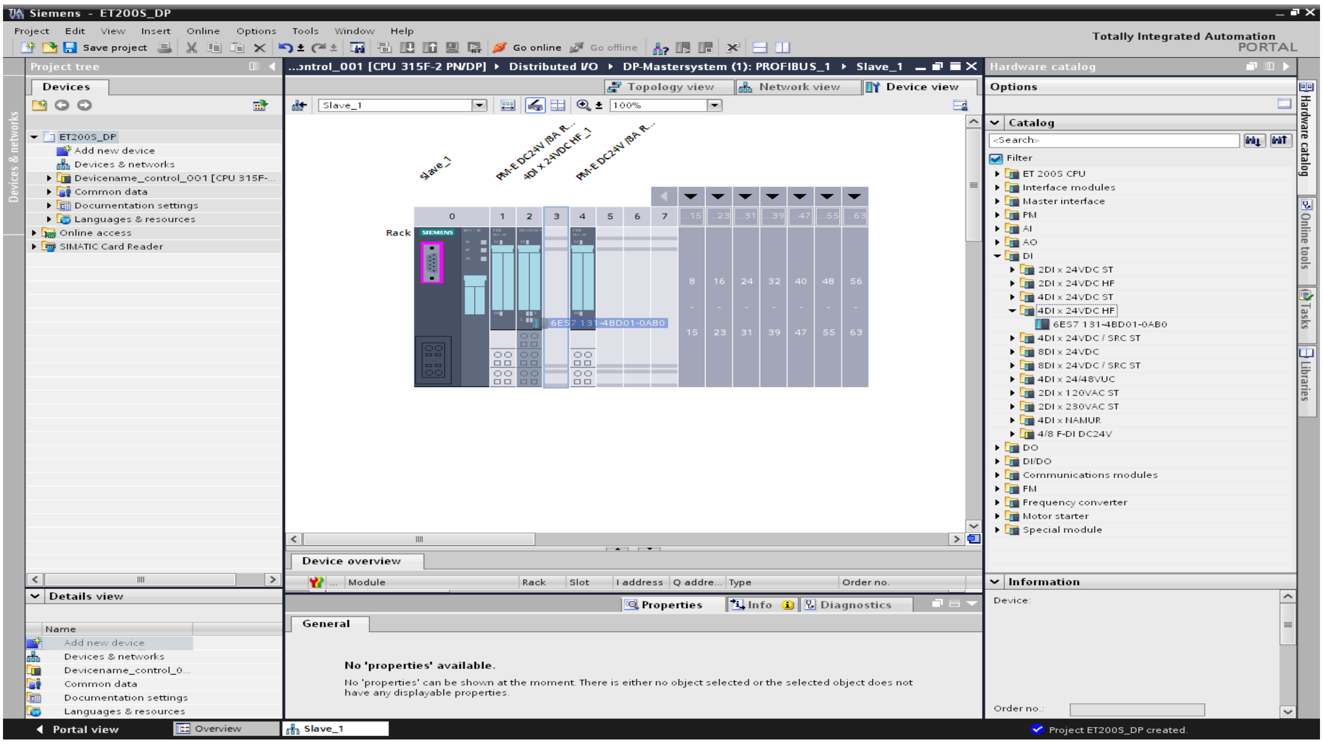
No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties.

Project ET200S_DP created.

Notes:

When several power modules are used, we have the option of switching the power supply to specific areas on and off. In the event of an emergency stop, for example, the power supply to the output modules can be switched off separately from the power supply to the input modules.

11. We drag the input module **'4DI x DC24V HF'** to slots 2 and 3. (→ DI → 4DI x DC24V HF → 6ES7 131-4BD01-0AB0)

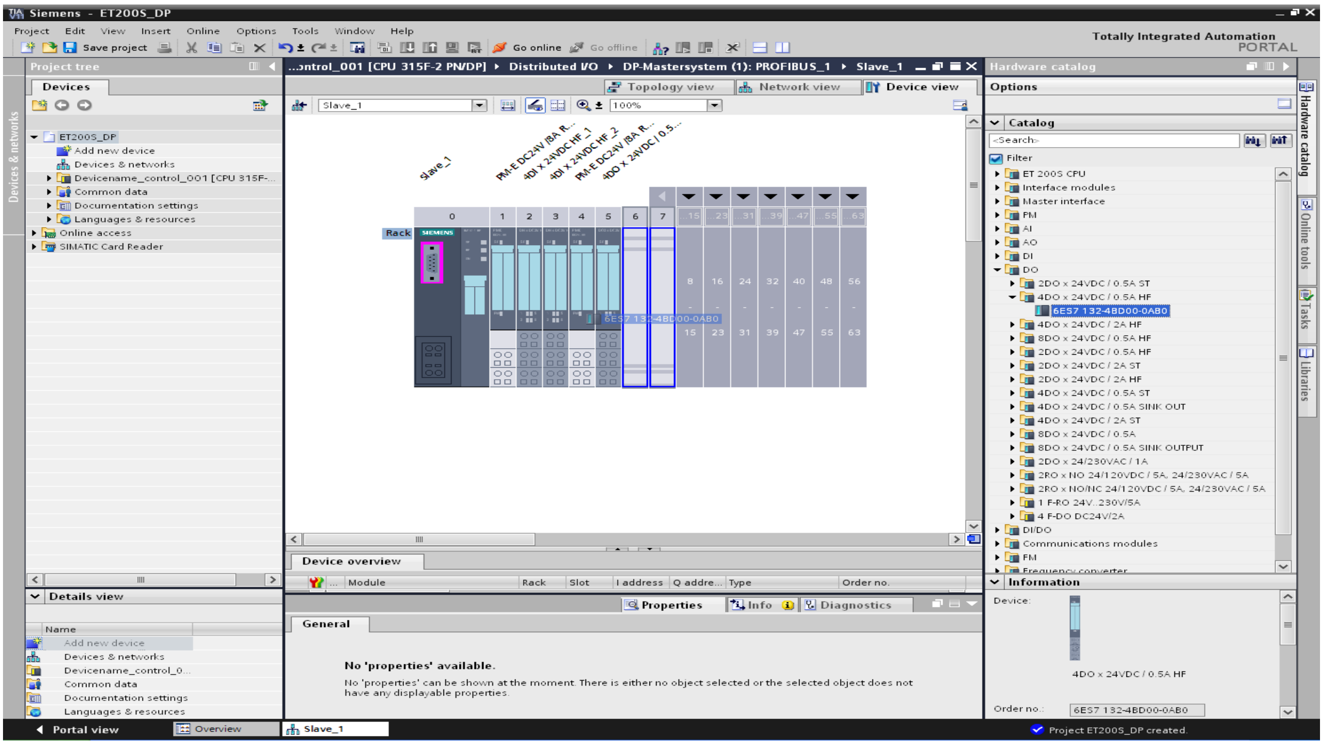


No 'properties' available.

No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties.

Project ET200S_DP created.

12. We drag the output module **'4DO x DC24V / 0.5A HF'** to slots 5 and 6. (→ DO → 4DO x DC24V / 0.5A HF → 6ES7 132-4BD00-0AB0)

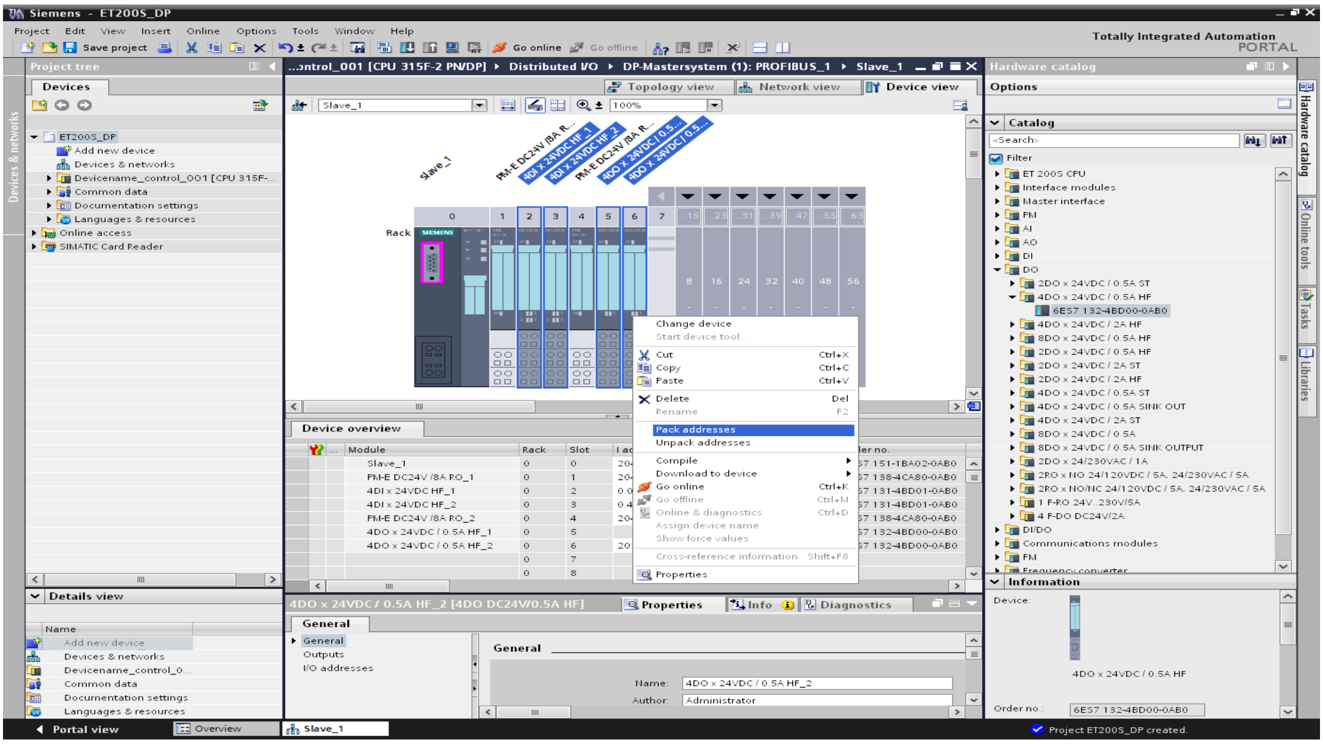


No 'properties' available.

No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties.

Project ET200S_DP created.

13. Addresses can be packed so that a new byte address does not need to be started with every module. To do this, select the modules in question and right-click on **‘Pack addresses’**. (→ Pack addresses)



Device overview

Module	Rack	Slot	I ad...	...ler no.
Slave_1	0	0	20...	...S7 151-1BA02-0AB0
PM-E DC24V /8A RO_1	0	1	20...	...S7 138-4CA80-0AB0
4DI x 24VDC HF_1	0	2	0.0...	...S7 131-4BD01-0AB0
4DI x 24VDC HF_2	0	3	0.4...	...S7 131-4BD01-0AB0
PM-E DC24V /8A RO_2	0	4	20...	...S7 138-4CA80-0AB0
4DO x 24VDC / 0.5A HF_1	0	5		...S7 132-4BD00-0AB0
4DO x 24VDC / 0.5A HF_2	0	6	20...	...S7 132-4BD00-0AB0
	0	7		
	0	8		

4DO x 24VDC / 0.5A HF_2 [4DO DC24V/0.5A HF]

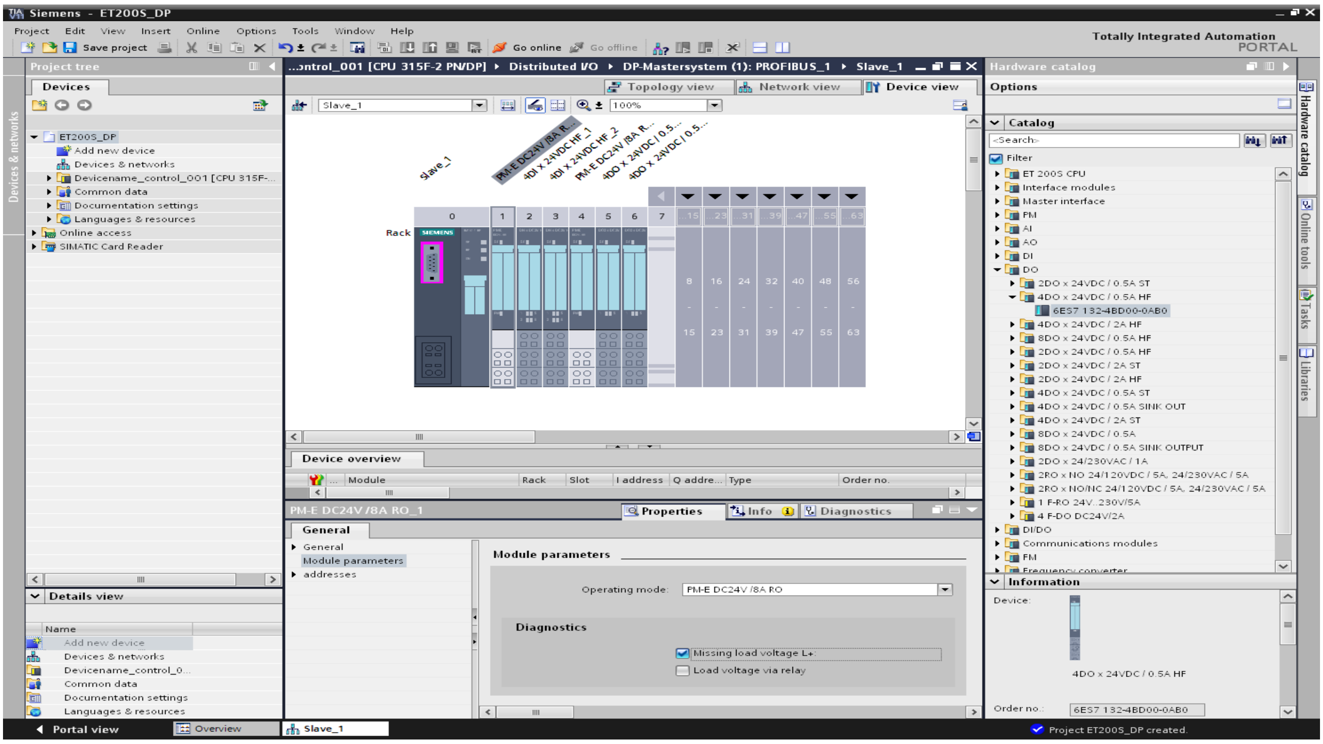
General

Name: 4DO x 24VDC / 0.5A HF_2

Author: Administrator

Order no.: 6ES7 132-4BD00-0AB0

14. **‘Properties’** for the modules can also be set in the **‘Module parameters’**. E.g. **‘Diagnostics, missing load voltage L+’** for the first power module. (→ PM-E DC24V / 8RA RO_1 → General → Module parameters → Diagnostics → Missing load voltage L+)



Module parameters

Operating mode: PM-E DC24V /8A RO

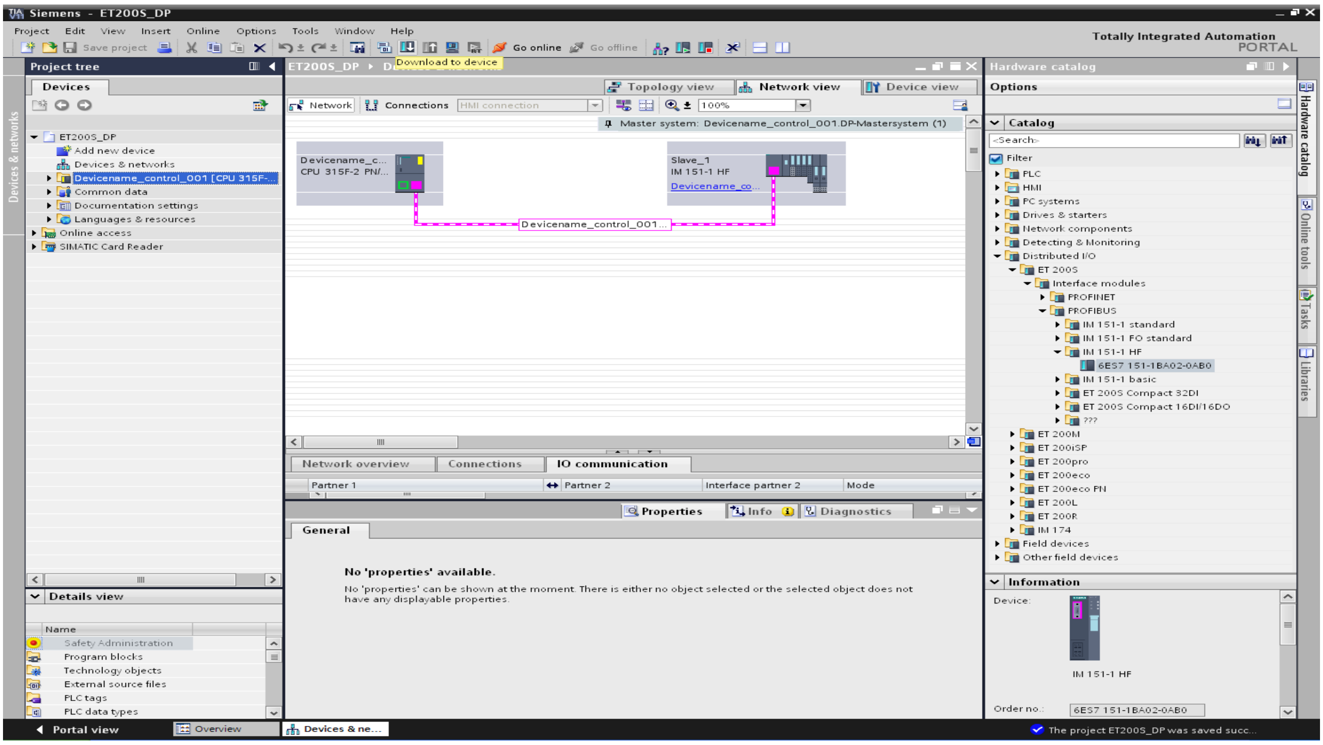
Diagnostics

☑ Missing load voltage L+

□ Load voltage via relay

15. Now, we save the project with the settings made so far. (→ Save project)

16. To load your entire program to the CPU, first select the **'Control 001'** folder and then click the Download to device icon ⬇. (→ Control 001 → ⬇)

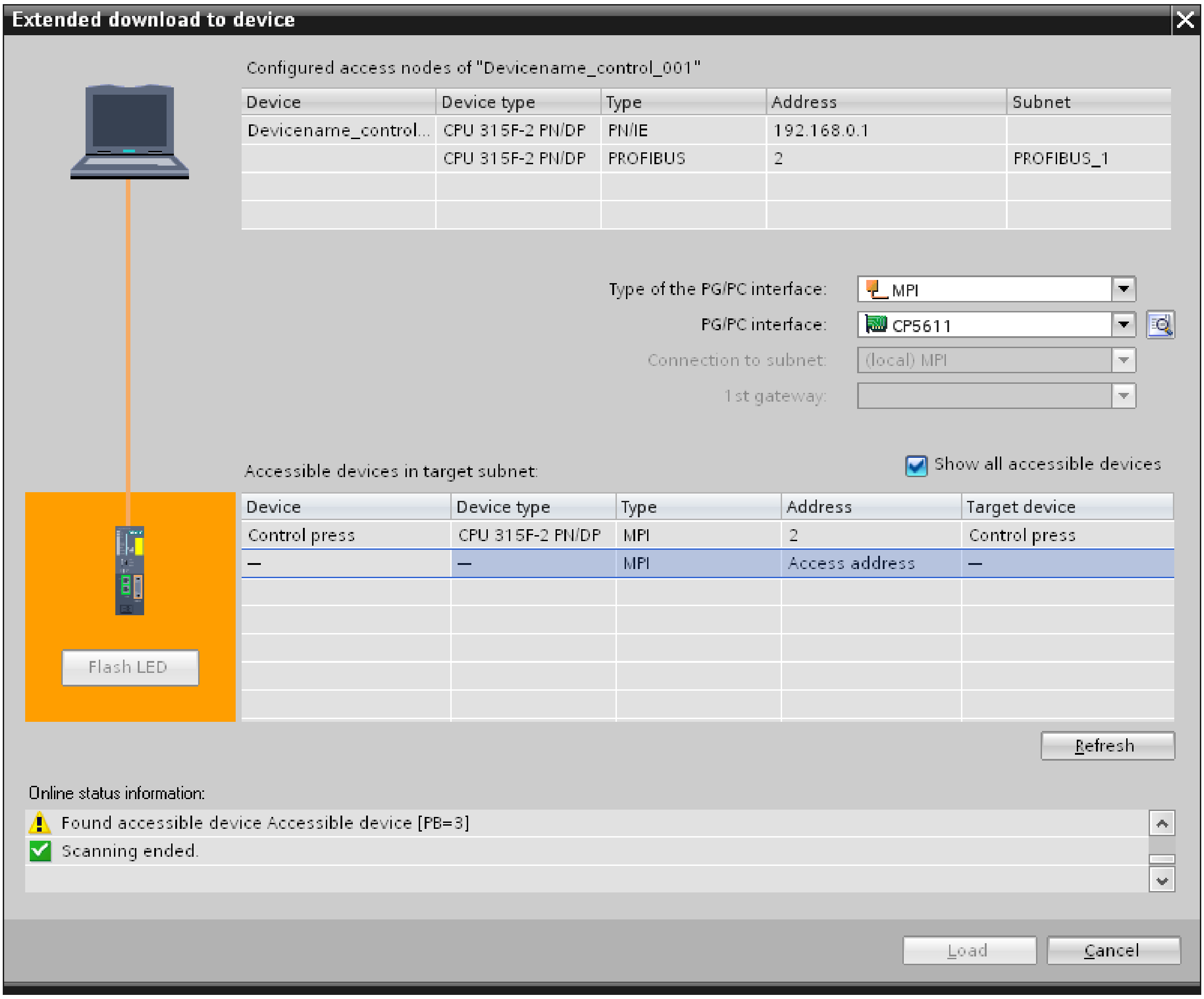


No 'properties' available.

No 'properties' can be shown at the moment. There is either no object selected or the selected object does not have any displayable properties.

The project ET200S_DP was saved succ...

17. For an initial commissioning, you select **‘MPI’** as the PG/PC interface type and then the PC adapter as the PG/PC interface in the following dialog. After you have also activated the **‘Show all accessible devices’** option, click **‘Refresh’** to see an **‘S7-300’** with the matching MPI address and select it as target device. Next, click **‘Load’**. (→ Type of the PG/PC interface: MPI → PG/PC interface: PC adapter → Show all accessible devices → Refresh → Device MPI 2 → Load)



Extended download to device

Configured access nodes of "Devicename_control_001"

Device	Device type	Type	Address	Subnet
Devicename_control...	CPU 315F-2 PN/DP	PN/IE	192.168.0.1	
	CPU 315F-2 PN/DP	PROFIBUS	2	PROFIBUS_1

Type of the PG/PC interface: MPI

PG/PC interface: CP5611

Connection to subnet: (local) MPI

1st gateway:

Accessible devices in target subnet:

☑ Show all accessible devices

Device	Device type	Type	Address	Target device
Control press	CPU 315F-2 PN/DP	MPI	2	Control press
—	—	MPI	Access address	—

Online status information:

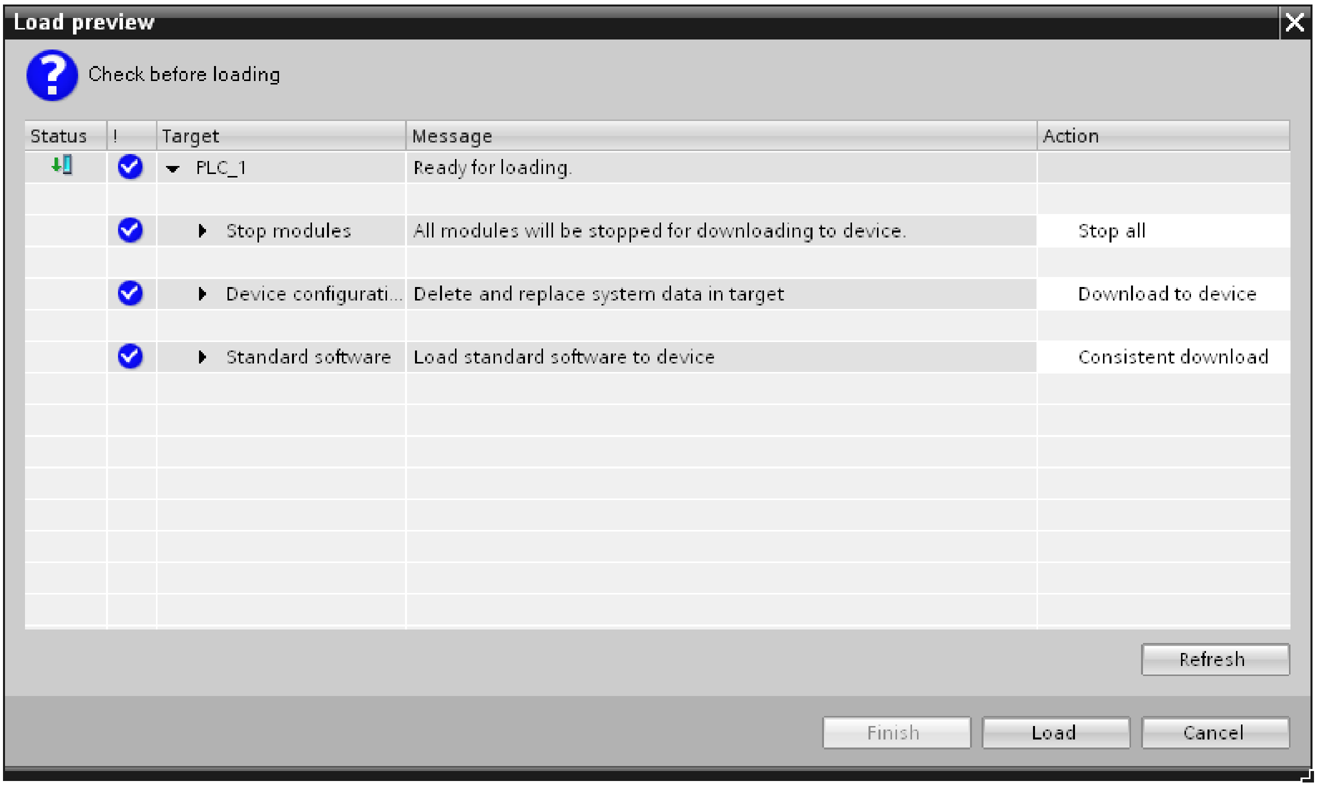
Found accessible device Accessible device [PB=3]

Scanning ended.

Notes:

If the PLC was loaded before with **‘PROFIBUS’** as interface setting, the **‘PG/PC interface PROFIBUS’** must be set as type here.

18. The configuration is now compiled automatically, and an overview of the steps to be performed is displayed once again for checking before the program is loaded. Click **‘Load’** to start loading the program. (→ Load)

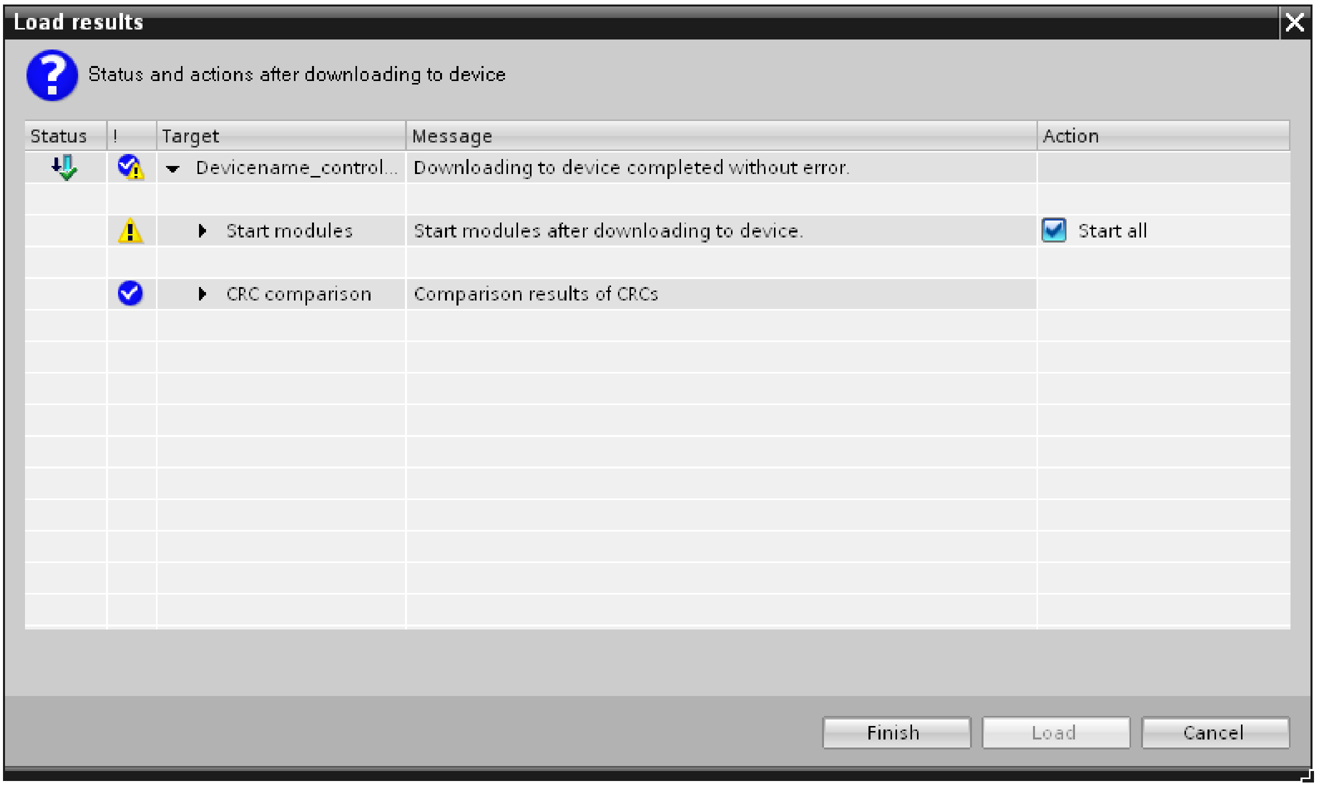


Load preview

Check before loading

Status	!	Target	Message	Action
	✓	PLC_1	Ready for loading.	
	✓	Stop modules	All modules will be stopped for downloading to device.	Stop all
	✓	Device configurati...	Delete and replace system data in target	Download to device
	✓	Standard software	Load standard software to device	Consistent download

19. The successful load result is displayed in a window. Now, click **'Start all'** and then **'Finish'** to set the CPU to Run mode again. (→ Start all → Finish)



Load results

Status and actions after downloading to device

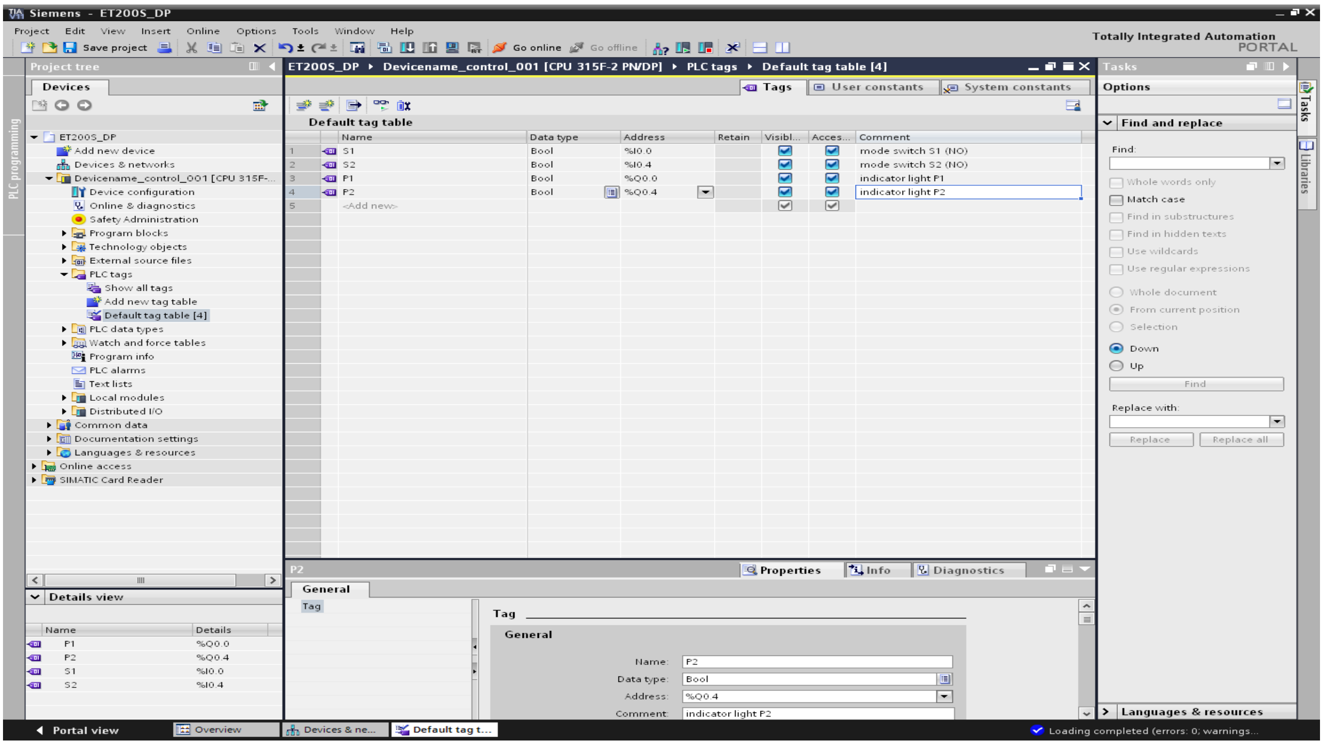
Status	!	Target	Message	Action
		Devicename_control...	Downloading to device completed without error.	
		Start modules	Start modules after downloading to device.	☑ Start all
	✓	CRC comparison	Comparison results of CRCs	

4.2. Generating and testing a program

20. Because modern programming uses tags and not absolute addresses, the **global PLC tags** must be defined here as an initial step.

These global PLC tags are descriptive names with a comment for each input and output used in the program. The global PLC tags can then be accessed later during programming via their names. These global tags can be used in all blocks anywhere in the program.

For this purpose, select **‘Control 001 [CPU 315F-2 PN/DP]’** and then **‘PLC tags’** in the project tree. Double-click the **‘Default tag table’** to open it and enter the names for the inputs and outputs as shown below. (→ Control 001 [CPU 315F-2 PN/DP] → PLC tags→ Default tag table)



Default tag table

	Name	Data type	Address	Retain	Visibl...	Acces...	Comment
1	S1	Bool	%I0.0		☑	☑	mode switch S1 (NO)
2	S2	Bool	%I0.4		☑	☑	mode switch S2 (NO)
3	P1	Bool	%Q0.0		☑	☑	indicator light P1
4	P2	Bool	%Q0.4		☑	☑	indicator light P2
5	<Add new>				☑	☑	

Name	Details
P1	%Q0.0
P2	%Q0.4
S1	%I0.0
S2	%I0.4

P2

General

Tag

Tag

General

Name: P2

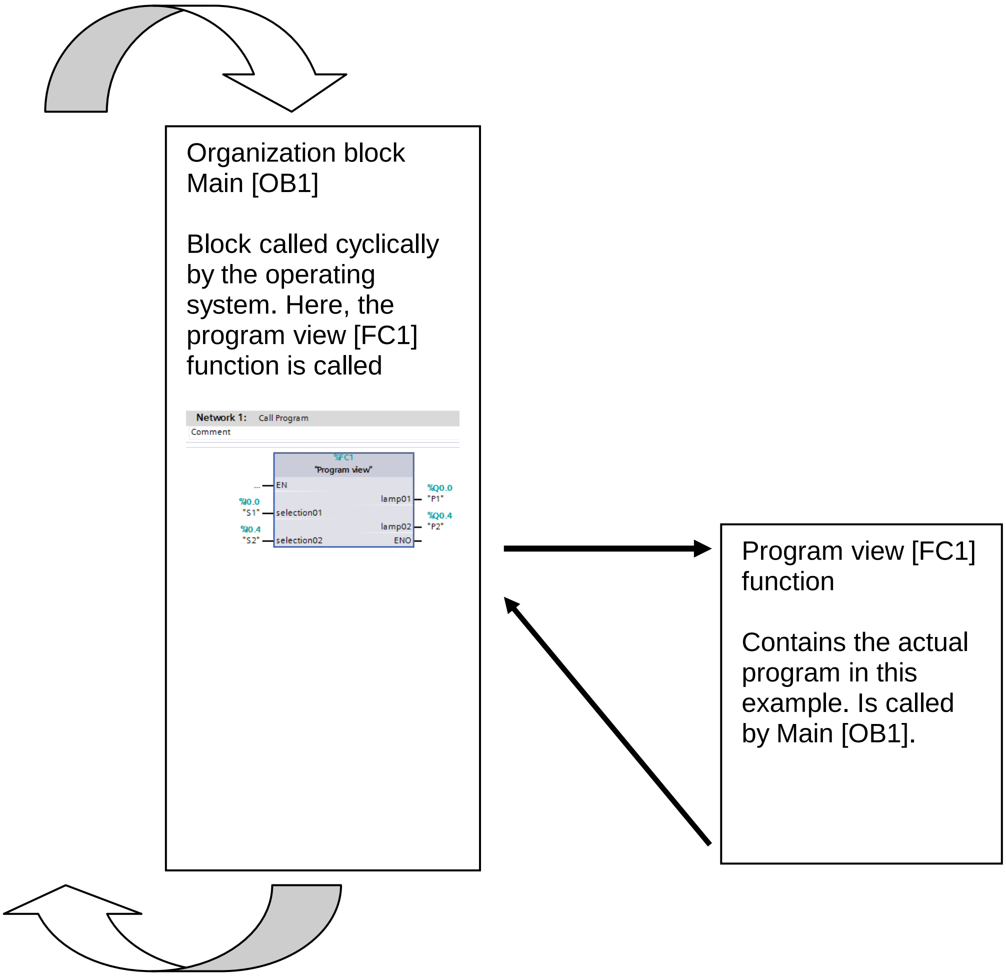
Data type: Bool

Address: %Q0.4

Comment: indicator light P2

21. The program sequence is written to so-called blocks. The organization block Main [OB1] is already provided as standard. This block represents the interface to the CPU operating system and is automatically called and cyclically processed by this operating system.
    From this organization block, additional blocks can be called in turn for structured programming, e.g., the program view [FC1] function.
    The purpose is to break down an overall task into partial tasks. This makes it easier to solve these tasks and to test their functionality.

Program structure of the example:



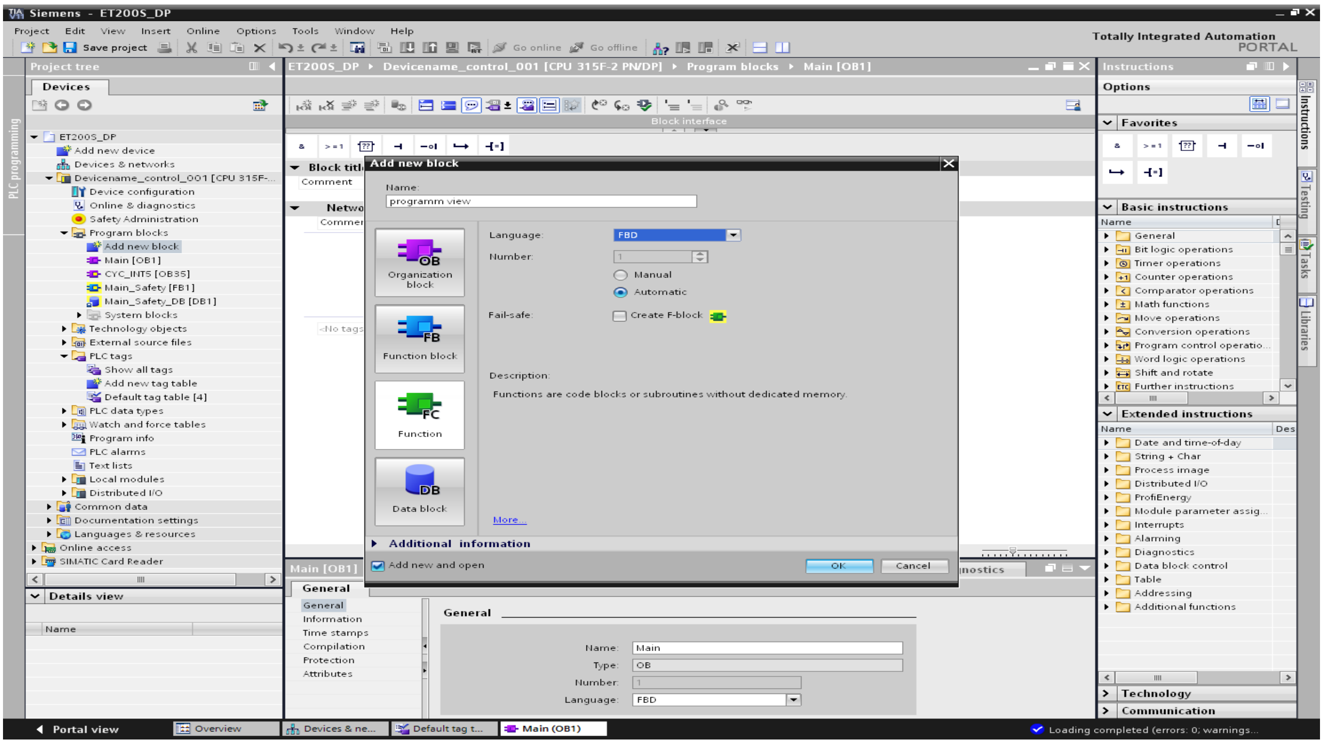
Organization block
Main [OB1]

Block called cyclically by the operating system. Here, the program view [FC1] function is called

Program view [FC1] function

Contains the actual program in this example. Is called by Main [OB1].

22. To create the program view [FC1] function, select **'Control 001 [CPU 315F-2 PN/DP]'** in the project tree and then **'Program blocks'**. Afterwards, double-click **'Add new block'**. Select **'Function (FC)'** and assign the name **'program view'**. Specify the **'FBD'** function block diagram as programming language. The numbering will be automatic. Since this FC1 will be called using its symbolic name later in any case, the number does not play an important role. Click **'OK'** to accept your entries.
(→ Control 001[CPU 315F-2 PN/DP]' → Program blocks → Add new block → Function (FC) → program view → FBD → OK)



Add new block

Name: programm view

Language: FBD

Number: 1

○ Manual

● Automatic

Fail-safe: □ Create F-block

Description:

Functions are code blocks or subroutines without dedicated memory.

More...

Additional information

☑ Add new and open

General

Name: Main

Type: OB

Number: 1

Language: FBD

23. The **‘program view [FC1]’** block then opens automatically. The interface of the block must be declared before the program can be written. In the interface declaration, the local tags known only in this block are defined.

The tags are divided into two groups:

- Block parameters that form the block interface for the call in the program.

Type	Designation	Function	Available in
Input parameters	Input	Parameters whose values are read by the block.	Functions, function blocks, and some types of organization blocks
Output parameters	Output / Return	Parameters whose values are written by the block.	Functions and function blocks
In/out parameters	InOut	A parameter whose value is read by the block when it is called and is written back by the block to the same parameter after it is processed.	Functions and function blocks

- Local data that is used for saving intermediate results.

Type	Designation	Function	Available in
Temporary local data	Temp	Tags that are used to store temporary intermediate results. Temporary data is retained for only one cycle.	Functions, function blocks, and organization blocks
Static local data	Static	Tags that are used for saving static intermediate results in the instance data block. Static data is retained until it is overwritten, which may be after several cycles.	Function blocks

24. The following tags are required in our example for declaration of the local tags.

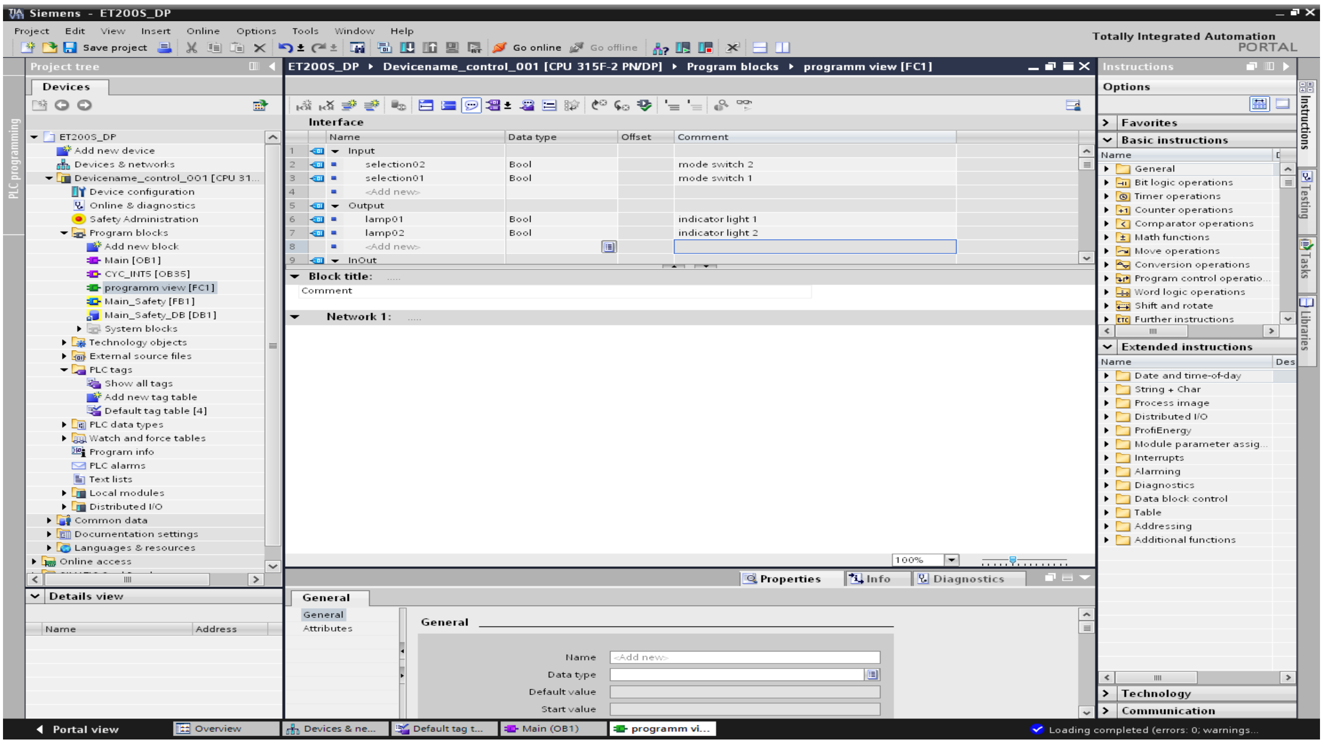
Input:

selection01	Selector switch 1
selection02	Selector switch 2

Output:

lamp01	Indicator light 1
lamp02	Indicator light 2

In this case, all of the tags are ‘Bool’ type tags. That is, they are binary tags that can only have the state ‘0’ (false) or ‘1’ (true).
All local tags should also be provided with a sufficiently descriptive comment for better understanding.



ET200S_DP ▸ Devicename_control_001 [CPU 315F-2 PN/DP] ▸ Program blocks ▸ programm view [FC1]

Interface

	Name	Data type	Offset	Comment
1	▾ Input			
2	selection02	Bool		mode switch 2
3	selection01	Bool		mode switch 1
4	<Add new>			
5	▾ Output			
6	lamp01	Bool		indicator light 1
7	lamp02	Bool		indicator light 2
8	<Add new>			
9	▾ InOut			

Block title:

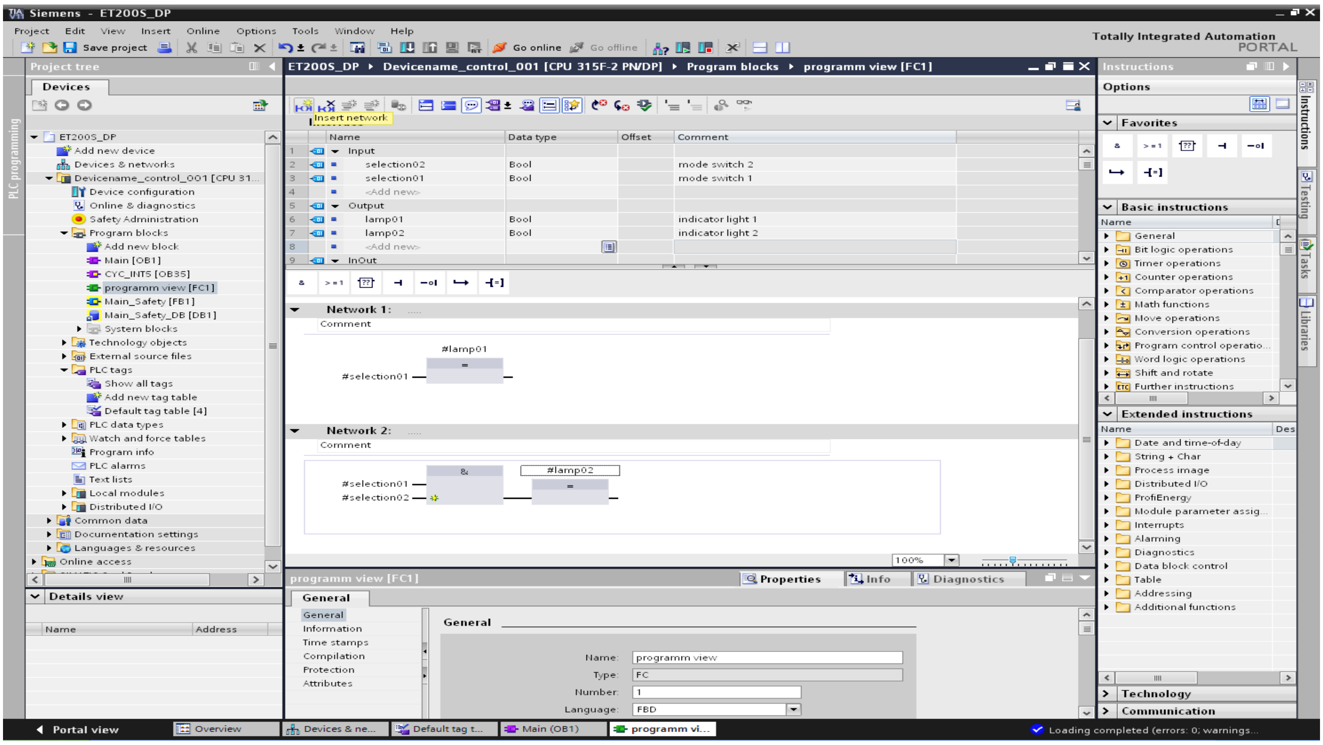
Comment

Network 1:

Loading completed (errors: 0; warnings...

Note:
To avoid confusion with the PLC tags, it is helpful to write the local tags with lowercase letters.

25. Once the local tags have been declared, you can start to create the program shown here. To provide a better overview, we program in networks. A new network can be inserted by clicking on the symbol **‘Insert network’**. Like the block itself, each network should be documented in the title line at least. If a longer text is needed for the description, the **‘Comment’** field can be used in addition.



ET200S_DP ▸ Devicename_control_001 [CPU 315F-2 PN/DP] ▸ Program blocks ▸ programm view [FC1]

Insert network

	Name	Data type	Offset	Comment
1	Input			
2	selection02	Bool		mode switch 2
3	selection01	Bool		mode switch 1
4	<Add new>			
5	Output			
6	lamp01	Bool		indicator light 1
7	lamp02	Bool		indicator light 2
8	<Add new>			
9	InOut			

Network 1:

Comment

#selection01 → = #lamp01

Network 2:

Comment

#selection01, #selection02 → & → = #lamp02

programm view [FC1]

General

Name: programm view

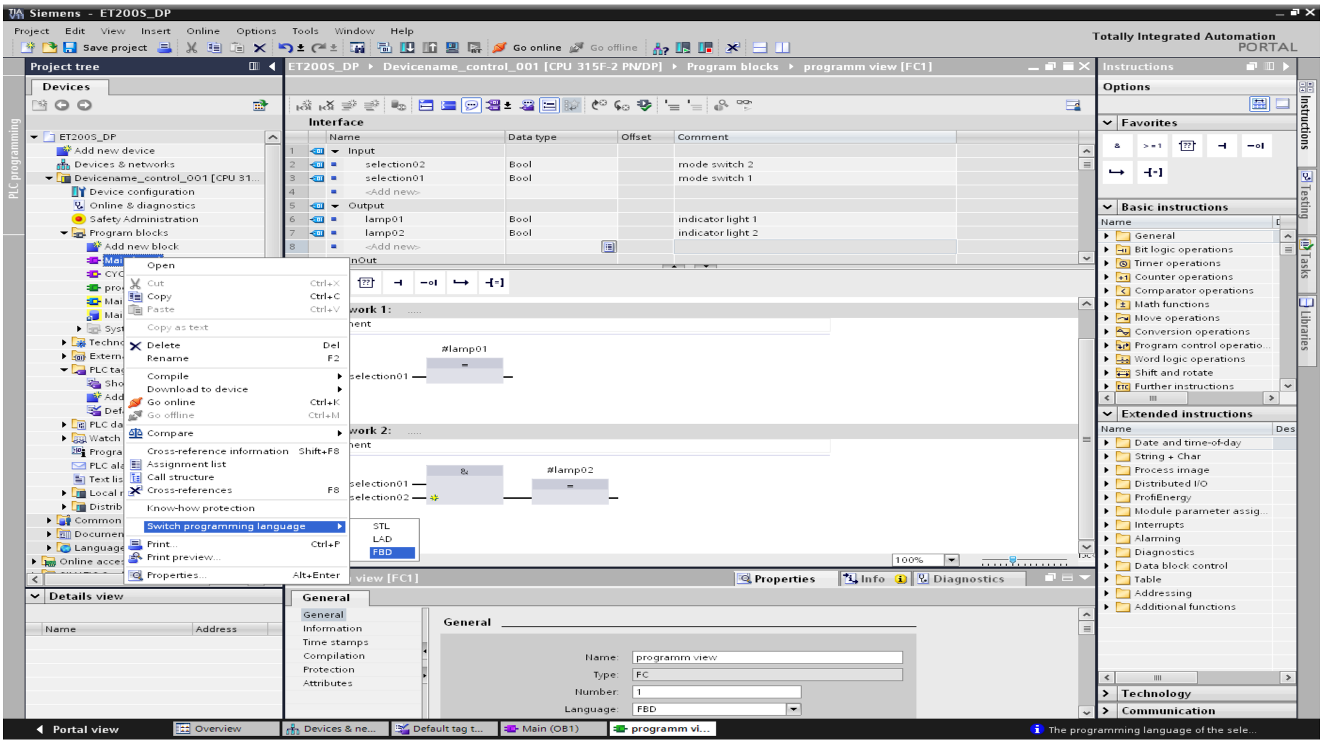
Type: FC

Number: 1

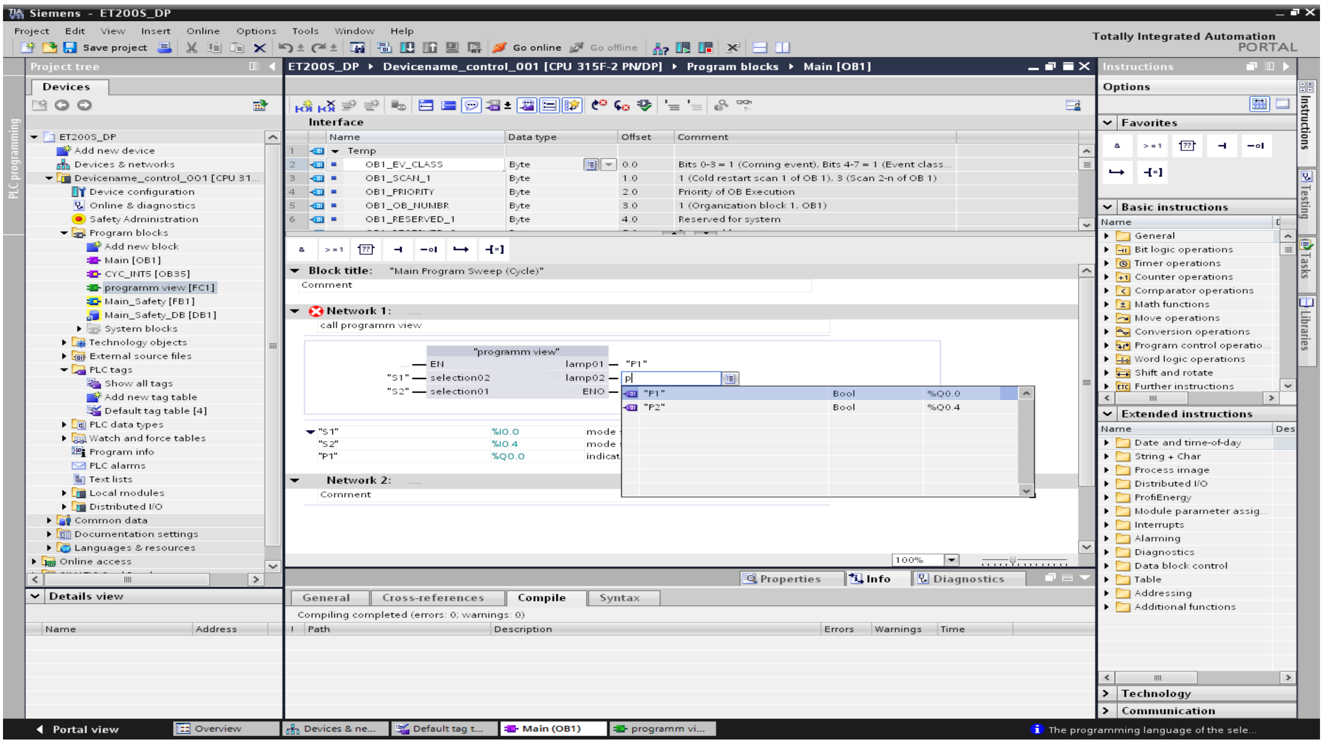
Language: FBD

Loading completed (errors: 0; warnings...

26. Before we double-click to open the block **‘Main[OB1]’**, we select **‘FBD’** as its programming language.
(→ Main[OB1] → Switch programming language → FBD → Main[OB1])



27. The **‘program view’** block can then be moved to Network 1 of the Main[OB1] block by means of drag-and-drop. The interface parameters of the **‘program view’** block must now be connected to the global PLC tags as shown here. Don't forget to document the networks in the Main[OB1] block. (→ Program view [FC1])



ET200S_DP ▸ Devicename_control_001 [CPU 315F-2 PN/DP] ▸ Program blocks ▸ Main [OB1]

Interface

	Name	Data type	Offset	Comment
1	Temp			
2	OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class...
3	OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
4	OB1_PRIORITY	Byte	2.0	Priority of OB Execution
5	OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
6	OB1_RESERVED_1	Byte	4.0	Reserved for system

Block title: "Main Program Sweep (Cycle)"
Comment

Network 1: .....
call programm view

"programm view"
EN — lamp01 — "P1"
"S1" — selection02 — lamp02 — p
"S2" — selection01 — ENO

"P1"	Bool	%Q0.0
"P2"	Bool	%Q0.4

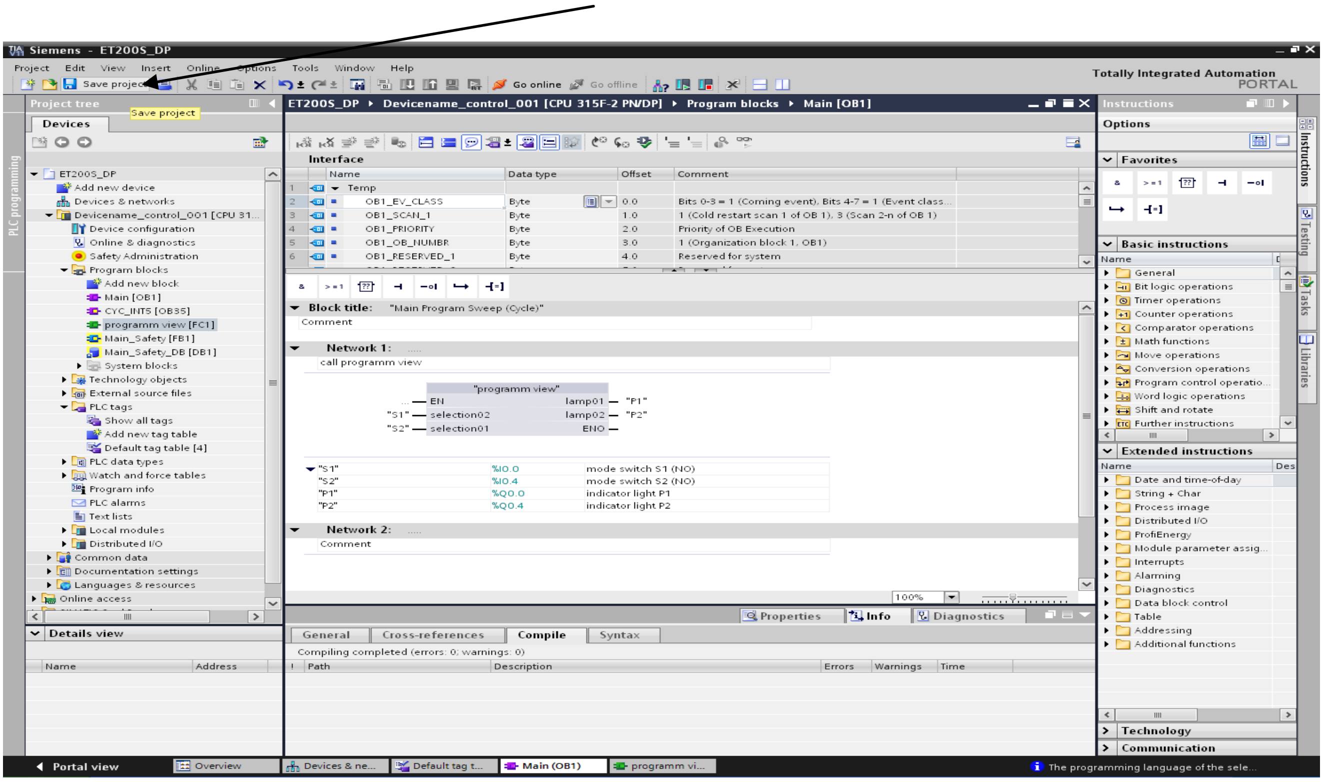
"S1"	%I0.0	mode
"S2"	%I0.4	mode
"P1"	%Q0.0	indicat

Network 2: .....
Comment

Compiling completed (errors: 0; warnings: 0)

28. The button Save project is then used to save the project once again.

("EMERGENCY STOP" → –o| → Save project)



Interface

	Name	Data type	Offset	Comment
1	Temp			
2	OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class...
3	OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
4	OB1_PRIORITY	Byte	2.0	Priority of OB Execution
5	OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
6	OB1_RESERVED_1	Byte	4.0	Reserved for system

Block title: "Main Program Sweep (Cycle)"
Comment

Network 1: .....
call programm view

"programm view"
... EN — lamp01 — "P1"
"S1" — selection02 — lamp02 — "P2"
"S2" — selection01 — ENO

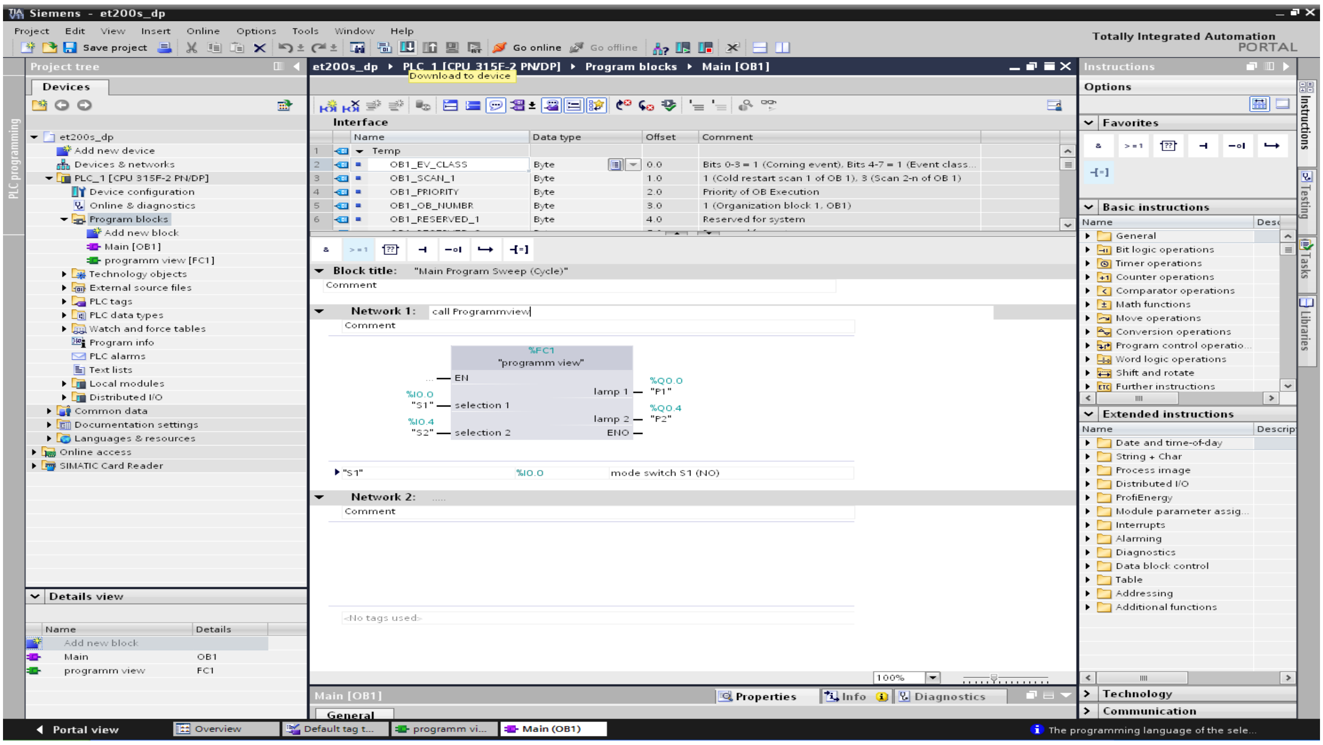
"S1"	%I0.0	mode switch S1 (NO)
"S2"	%I0.4	mode switch S2 (NO)
"P1"	%Q0.0	indicator light P1
"P2"	%Q0.4	indicator light P2

Network 2: .....
Comment

Compiling completed (errors: 0; warnings: 0)

!	Path	Description	Errors	Warnings	Time

29. To only load the program blocks to the CPU, first select the **'Program blocks'** folder and then click the Download to device icon . (→ Program blocks →)



Download to device

et200s_dp ▸ PLC_1 [CPU 315F-2 PN/DP] ▸ Program blocks ▸ Main [OB1]

Interface

	Name	Data type	Offset	Comment
1	▼ Temp			
2	OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class...
3	OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
4	OB1_PRIORITY	Byte	2.0	Priority of OB Execution
5	OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
6	OB1_RESERVED_1	Byte	4.0	Reserved for system

Block title: "Main Program Sweep (Cycle)"

Comment

Network 1: call Programmview

Comment

%FC1
"programm view"
EN
%I0.0 "S1" — selection 1
lamp 1 — %Q0.0 "P1"
%I0.4 "S2" — selection 2
lamp 2 — %Q0.4 "P2"
ENO

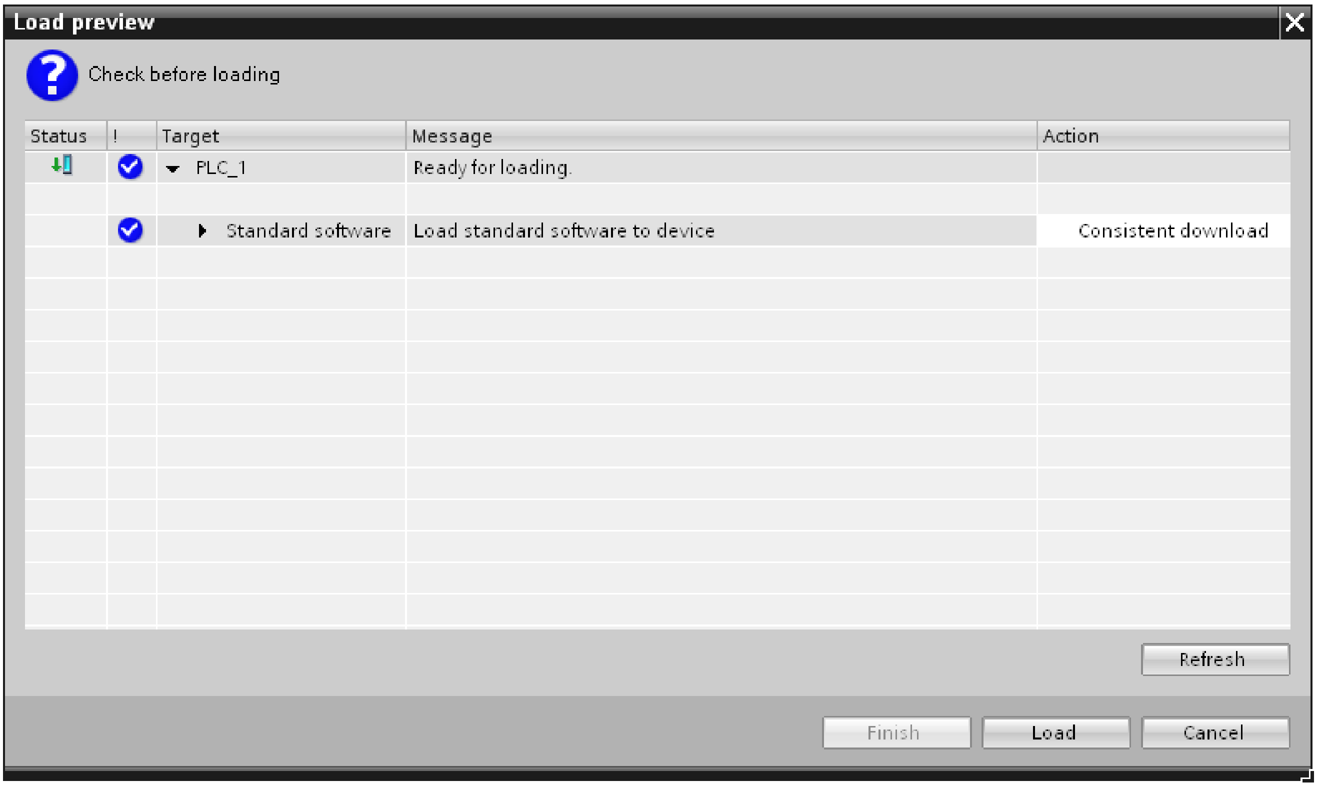
"S1" %I0.0 mode switch S1 (NO)

Network 2:

Comment

<No tags used>

30. The program is now compiled automatically, and an overview of the steps to be performed is displayed once again for checking before the program is loaded. Click **'Load'** to start loading the program. (→ Load)

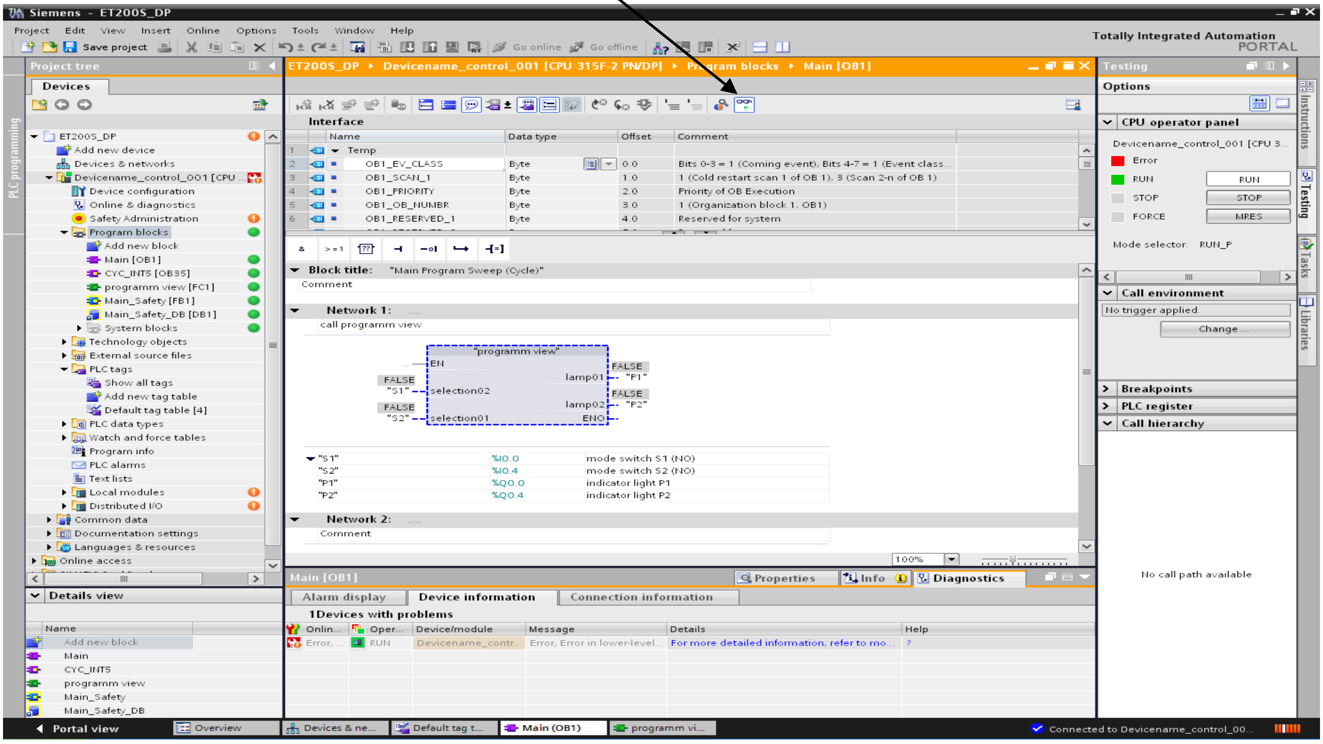


Load preview

Check before loading

Status	!	Target	Message	Action
	✓	PLC_1	Ready for loading.	
	✓	Standard software	Load standard software to device	Consistent download

31. Clicking the icon (Monitoring on/off) allows you to monitor the state of the input and output tags on the **'program view'** block. (→)



	Name	Data type	Offset	Comment
1	Temp			
2	OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class...
3	OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
4	OB1_PRIORITY	Byte	2.0	Priority of OB Execution
5	OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
6	OB1_RESERVED_1	Byte	4.0	Reserved for system

Block title: "Main Program Sweep (Cycle)"

Comment

Network 1: .....

call programm view

"programm view"
EN
FALSE "S1" — selection02
FALSE "S2" — selection01
lamp01 — FALSE "P1"
lamp02 — FALSE "P2"
ENO

"S1"	%I0.0	mode switch S1 (NO)
"S2"	%I0.4	mode switch S2 (NO)
"P1"	%Q0.0	indicator light P1
"P2"	%Q0.4	indicator light P2

Network 2: .....

Comment

1Devices with problems

Onlin...	Oper...	Device/module	Message	Details	Help
Error, ...	RUN	Devicename_contr...	Error, Error in lower-level...	For more detailed information, refer to mo...	?