

HARDWARE CONFIGURATION

OBJECTIVE

In this chapter, students are acquainted with how an automation system functions.

You will be able to configure the selected hardware in the hardware configuration of **PCS7** and check it for consistency. Important settings are parameterized so that the process control system **PCS7**, using this information, autonomously sets up all blocks needed in the automation stations for communication between encoders, actuators and the control level.

THEORY IN SHORT

When the hardware is configured, the 'real' components for recording measured values and reading out signals to influence the process are inserted and arranged in tabular representation. As Figure 1 shows, this process describes at which slot and in which sequence which input/output module is placed. In addition, the input and output signals are assigned to a defined memory location in the process image and the modules are parameterized during this process. If field busses are used, unique station addresses are specified also.

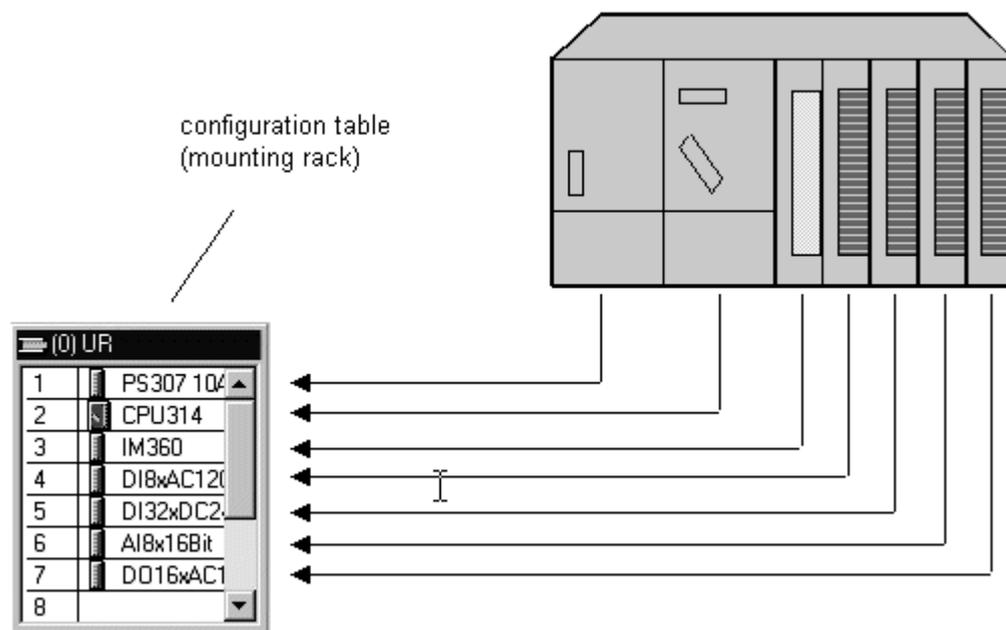


Figure 1: Mapping the real module to the configuration table [1]

When saving and compiling the settings, consistency will be checked internally first (do all modules match, are addresses assigned twice, etc.). As soon as the configuration is consistent within itself, the blocks needed for the communication of the process data are set up without any user action, and can then be loaded into the control system's **automation stations (AS)**. They will then have all the information needed to ascertain whether the installed configuration corresponds to the one that was configured. Then they make sure that the process data for further processing is available at the ASs in the desired scanning frequency. To most users it remains hidden that extensive provisions are made if there is an error: such as automatically sending suitable indications and alarms.

THEORY

DISTRIBUTED ARCHITECTURE OF PROCESS CONTROL SYSTEMS

Scalable process control systems such as **PCS7** cover a wide range of processes: from applications for small laboratory systems with a few encoders and actuators to plants with a hundred thousand measuring points. To cover such a wide range, special structures are needed. A typical component structure that can be easily expanded looks like this:

- On the process control level, an operator system is used consisting of one or several **operator stations (OS)**. With these OSs, the control room operators are able to run and monitor the plant.
- On the control level, one or several **automation stations (AS)** are located where the control and closed loop control functions are executed in real time. These systems are to perform their service regardless of OS failures. They consist of at least one power supply (PS), one CPU and if needed, communication modules (CP).
- On the field level, the states of the technical processes are recorded or the process is specifically influenced by means of encoders and actuators.

While -as Figure 2 shows- in the lab all components can run on a single computer system, starting with the order of magnitude of a technical college it is advisable to distribute the components in order to handle the complexity. To exchange data between these components, depending on the requirement different bus systems are used that, for example, possess the real time properties needed for process data communication.

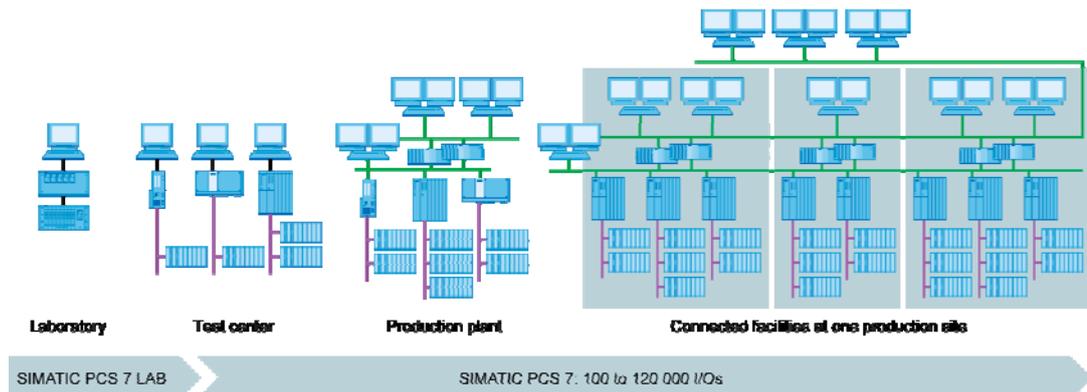


Figure 2: Scalable structure of the process control system PCS7 [2]

PROCESS IMAGE

As on any other computer, the automation programs are processed on the automation systems (AS) in the central processing unit (CPU). The control program is processed cyclically. If signals from the process are processed also, the program does not access the connected input and output modules directly. Instead, a process image is generated where, at the beginning of the cycle, all signals are read in and stored at once.

This has two reasons: First, accessing the process image requires appreciably less time since it is stored in the CPU's internal memory. Second, this ensures that the input information was collected within a defined time window –regardless of the execution time of the control program. This data consistency is attained by inputting, once for each cycle, the signals of the input modules in the process image of the inputs (PII). The program is then executed and the results written into the process image of the outputs (PIO).

After the entire program is executed, the data is written from the PIO into the output modules and only then read out to the process. Then, the process image of the inputs is updated again, as shown Figure 3 below.

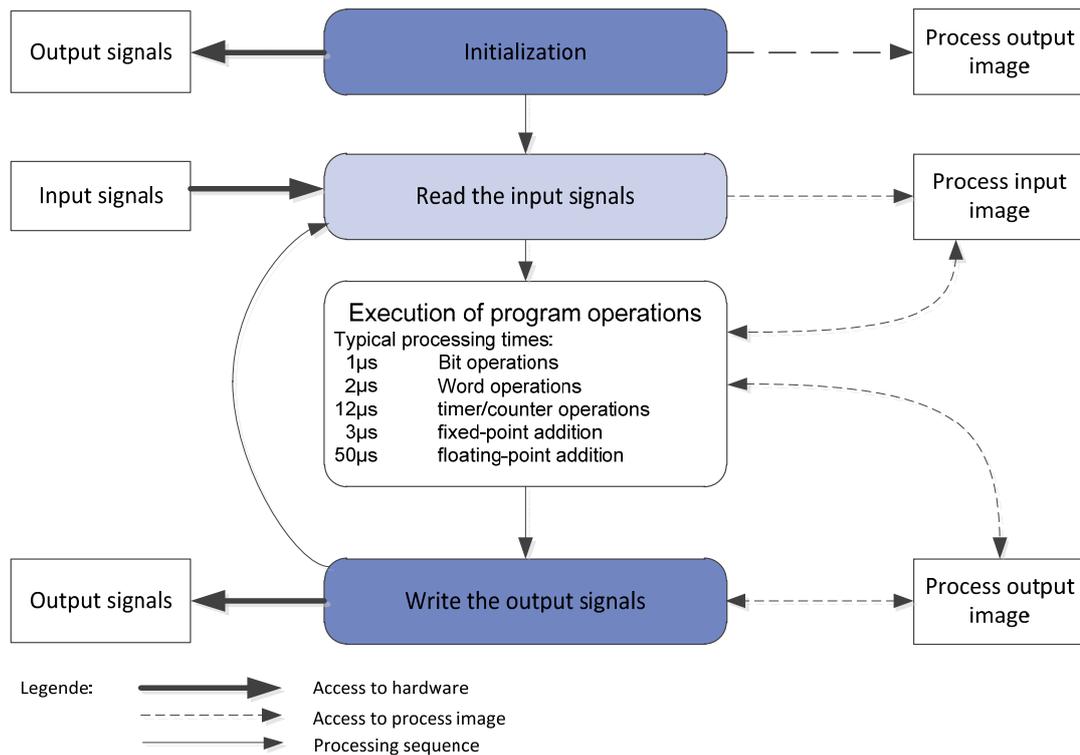


Figure 3: Reading and writing input and output signals during the PLC [4] processing cycle

CONNECTION TO THE PROCESS

With transducers, the process signals -such as temperature, pressure, level or through flow- recorded by the encoders are converted into an electrical signal. Unless the measuring device is interfaced directly by means of a fieldbus, the signal is usually converted into an electrical standardized signal. A standardized signal module can then record it on the automation system side.

Since in process industry plants, measured values from a handful up to several thousands are to be recorded, it has to be possible to select, to clearly assign and to parameterize the measuring module while the automation is planned. For that reason, the required signal modules are first arranged virtually in the hardware configuration. Also, memory is assigned in the process image to the signal modules. As soon as a signal module is added to the configuration, sufficient memory is assigned in the process image automatically. This automatic assignment can subsequently be changed manually; however, the size of the memory area in the CPU must be noted.

Depending on the signal type, different signal modules are used. For binary signals, DI and DO modules are used. The individual signals are organized bit by bit; that means each input/output signal is assigned one bit of the process image. However, the signal modules usually record 8, 16 or 32 signals simultaneously.

For analog signals, AI and AO modules are used. Analog modules are usually organized in words (16 bits). As shown in Figure 4, each analog input or output signal is assigned 16 bits of memory. In return, the analog input module converts the analog process signal into a digital form. Depending on the resolution, only the more significant positions are assigned, and a '0' is written to the less significant ones. Analog output modules convert the digital output value into an analog signal. In the case of analog signals, we not only differentiate the modules according to the number of signals, but also according to their resolutions; for example, 2x12 bits, 8x13 bits or 8x16 bits.

	Address	Symbol	Data type	Comment
1	MV 512	A1.T2.A1T...	WORD	current value filling level reactor R001
2	MV 514	A1.T2.A1T...	WORD	current value filling level reactor R002
3	MV 516	A1.T2.A1T...	WORD	current level temperature reactor R001
4	MV 518	A1.T2.A1T...	WORD	current value temperature reactor R002
5	MV 520			
6	MV 522			
7	MV 524			
8	MV 526			

Sorting:
 Display Columns R, D, M, C, CC

The symbols are updated with 'OK' or 'Apply'

Figure 4: Symbol table of an analog input module

DISTRIBUTED IO

If the distances of the inputs and outputs to the automation system are large, wiring can become very extensive and involved. Moreover, electro-magnetic interference may impair reliability. Using distributed IO devices is suitable in the case of such plants:

- The automation system is located centrally
- One or several IO devices (input and output modules) process in the distributed mode on location
- Data is transmitted between the IO and the automation system by means of Profibus DP [1]. To this end, the AS and the IO have to be equipped with corresponding communication modules.

For the plant described in the above chapter, the SIMATIC ET200M was selected as the distributed IO device. The IO modules of the proven automation system S7-300 are connected to an interface module (IM 153-x) that ensures the communication to the AS. Figure 5 shows a typical configuration. To the right of interface module IM 153-1 several digital and analog input and output modules are connected. The process signals coming from the field are sent directly to the routing level that is placed below the input and output modules. From there, short cables lead to the modules -which provides for quickly eliminating errors when wiring into the field.

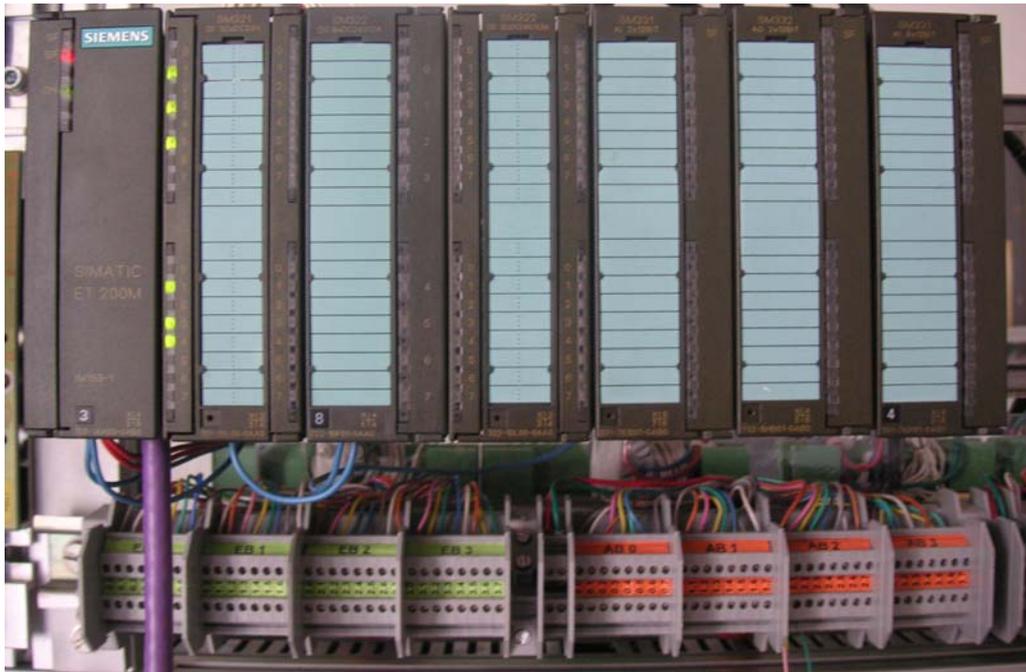


Figure 5: Modular distributed I/O device ET200M (Source: Laboratory Plant TU Dresden)

As Figure 6 below shows: in the hardware configuration, the SIMATIC ET200M is interfaced with a Profibus DP line of the AS. The hardware configuration automatically suggests addresses that are not yet used in the selected subnet. The ET200M slots are assigned input and output modules, as described below.

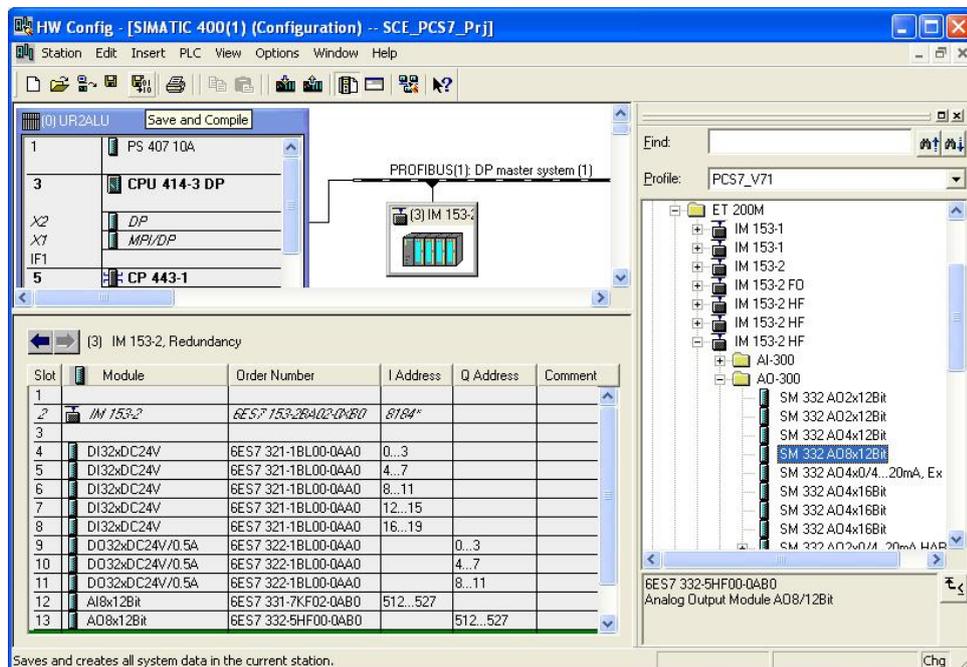


Figure 6: Hardware configuration of a modular distributed I/O device (ET200M)

LITERATURE

- [1] Online Help PCS 7. Siemens.
- [2] www.automation.siemens.com

STEP BY STEP INSTRUCTIONS

TASK

In this chapter, the **PCS7** project for a multi-purpose plant is set up, using a wizard.

Then, first the S7 station it contains is configured. The example uses the following: the SIMATIC S7-400 with the CPU414- 3DP, and the communication processor CP443-1 for interfacing with the Ethernet by means of the TCP/IP protocol.

The connection of the IO signals to control the actuators in the plant and to record the input signals is the ET200M. This modular field device is connected to the CPU with the fieldbus PROFIBUS DP.

The PC station as host computer with the **PCS7** software and **WinCC** for visualization has to be configured also. Any PC or laptop equipped with a standard Ethernet interface can be used.

With the Ethernet -by means of the TCP/IP protocol- the host computer as operator station (OS) is interfaced with the CPU as automation station (AS).

The project is developed on the host computer also. Thus, the host computer is operator station (OS) and engineering station (ES) in one.

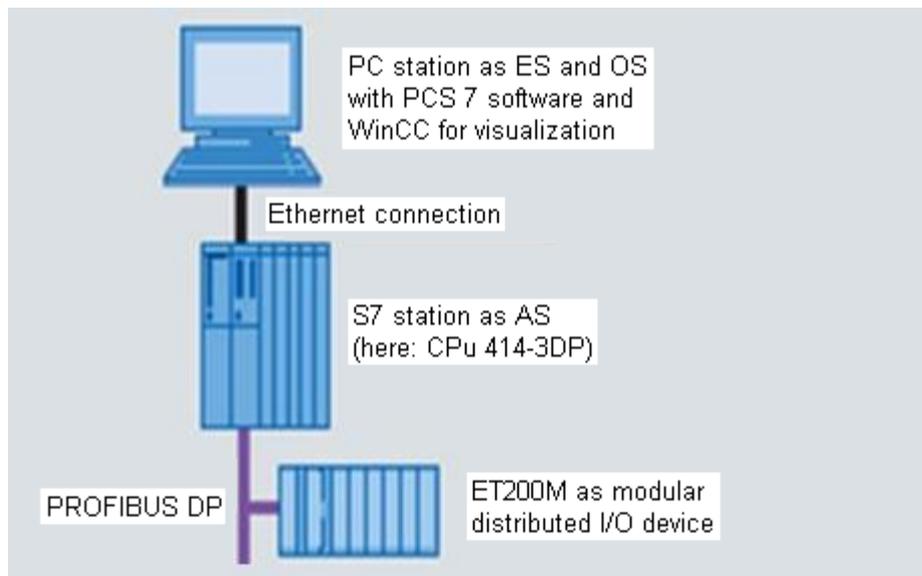


Figure 7: PCS structure of the multi-purpose plant



Note: The abbreviations for **Engineering Station (ES)**, **Operator Station (OS)** and **Automation Station (AS)** should be borne in mind, since these terms are used often in the **PCS7** software and also in this manual.



Note: Depending on the available hardware, the CPU414-3DP can be replaced with another CPU, a PC based SIMATIC PCS 7 AS RTX or the SIMATIC PCS 7 Box PC.

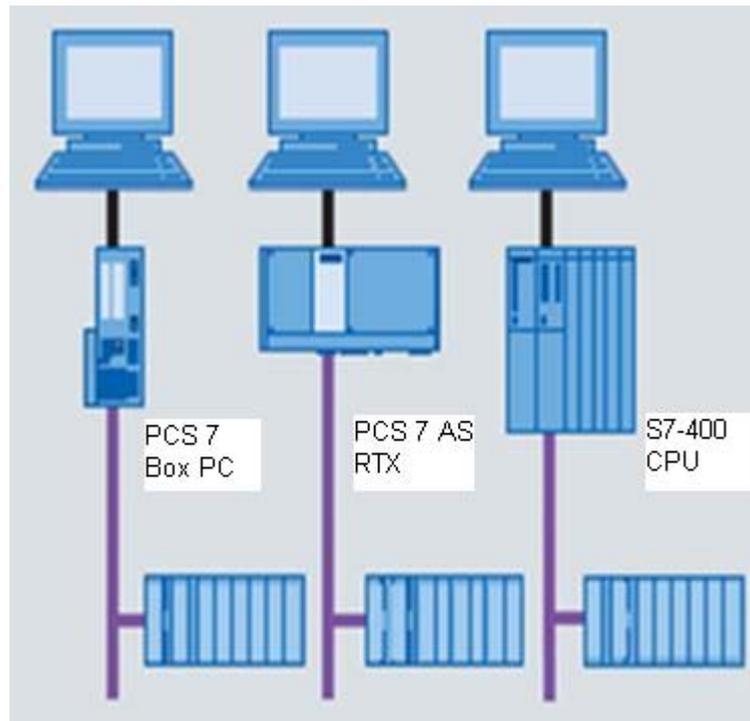


Figure 8: Various PCS structures with SIMATIC PCS 7 Box, SIMATIC PCS 7 AS RTX and SIMATIC S7-400 CPU 414-3DP as control system (from left)



Note: In the chapters below, the simulation software S7 PLCSIM is used to test the programs. Thus, in principle any controller can be configured.

OBJECTIVE

In this chapter, the student will learn the following:

- Setting up a **PCS7** project
- Configuring the hardware for an S7 station
- Configuring the hardware for a PC station with **WinCC**
- Networking the S7 station and the PC station

PROGRAMMING

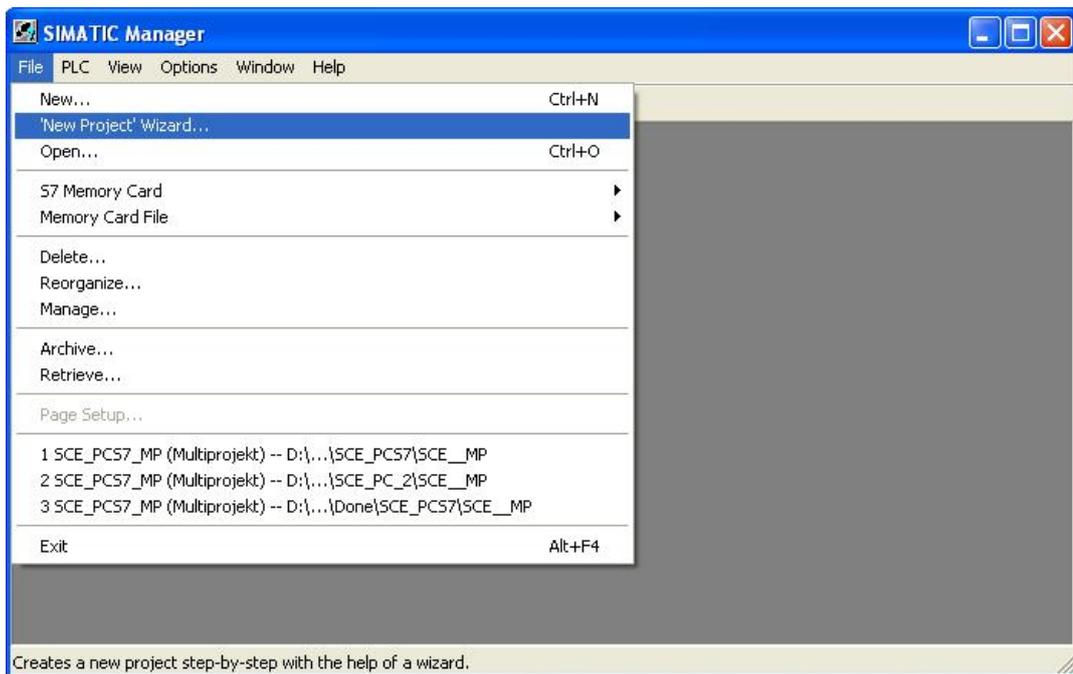
1. The central tool in the **PCS7** is the **SIMATIC Manager**. It is opened here with a double click. (→ SIMATIC Manager)



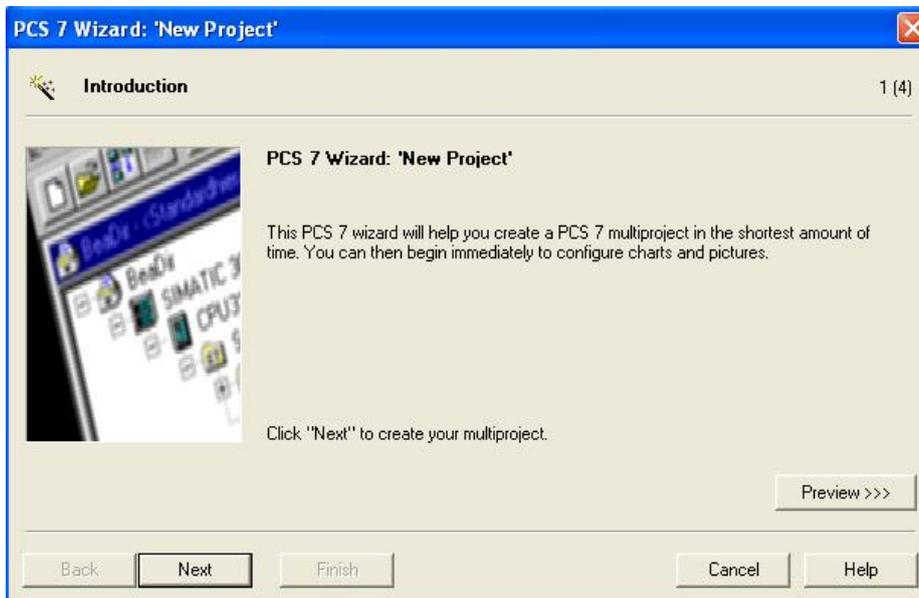
SIMATIC Manager

2. To create a **PCS7** project, using the wizard is recommended because it sets up the S7 station and the PC station at the same time.

(→ File → 'New Project' Wizard)

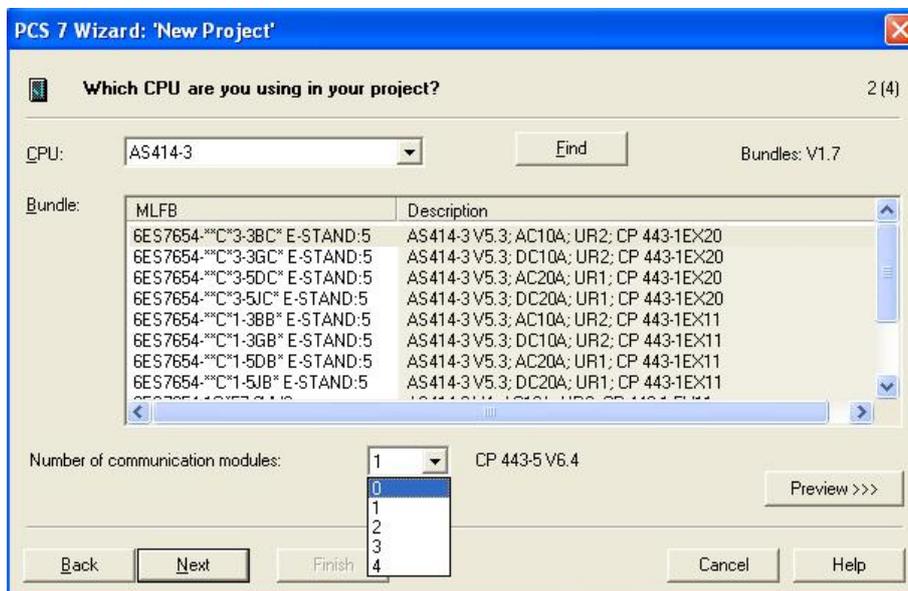


- Here, we are setting up the project as a multi-project. Thus, a master data library is set up also in addition to the S7 station and the PC station. It ensures that within a project the same version of blocks and planning templates (measuring point types) is always used. (→ Next)



- Next, we select the following: the AS configuration with the CPU we are using, the power supply and the communication processors for PROFIBUS and Ethernet. Since **PCS7** stations are usually ordered as a bundle, the bundles can be selected here based on their order numbers.

(→ AS414-3 → 6ES7*** → Number of communication modules CP443-5 → 0 → Next)

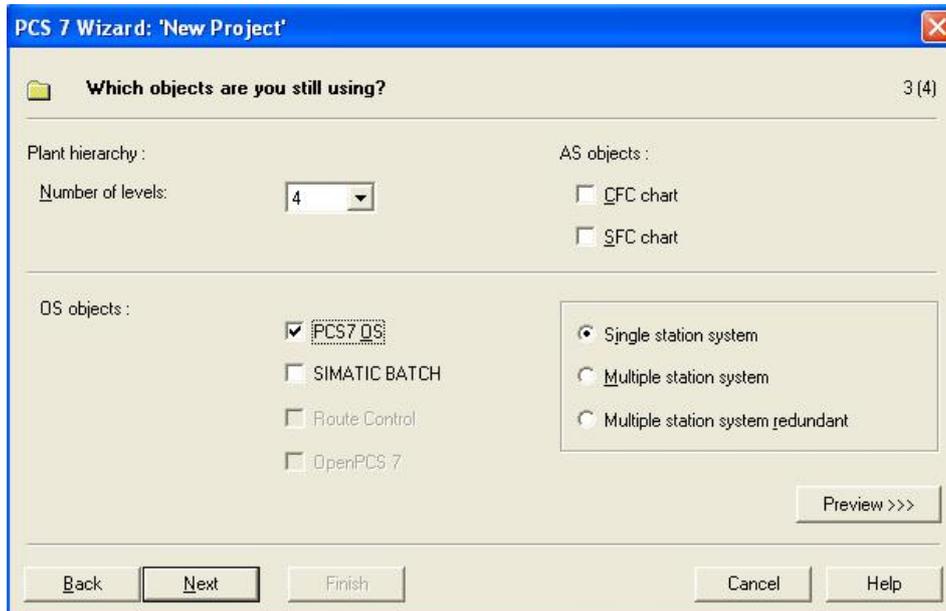


Note: The quantity of additional communication modules for PROFIBUS is added here in an additional selection to the bundle.

Since the bundles listed here often don't correspond 100% to the existing S7 station, it can happen that later, in the hardware configuration, individual components have to be added or exchanged.

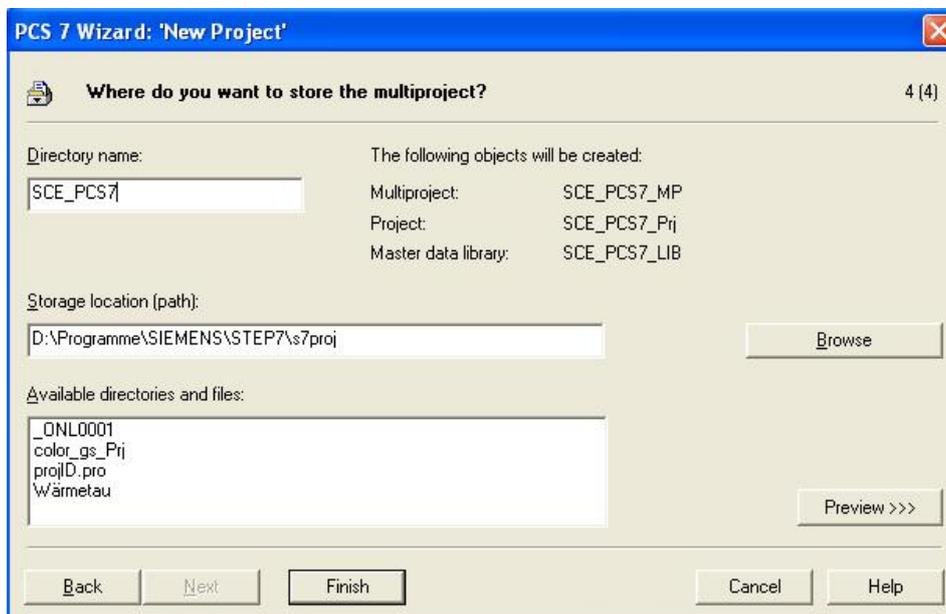
- Now, we select the number of hierarchical levels for the plant hierarchy (refer to chapter 'Plant Hierarchy'), and an OS object.

(→ 4 → PCS7 OS → Single station system → Next)



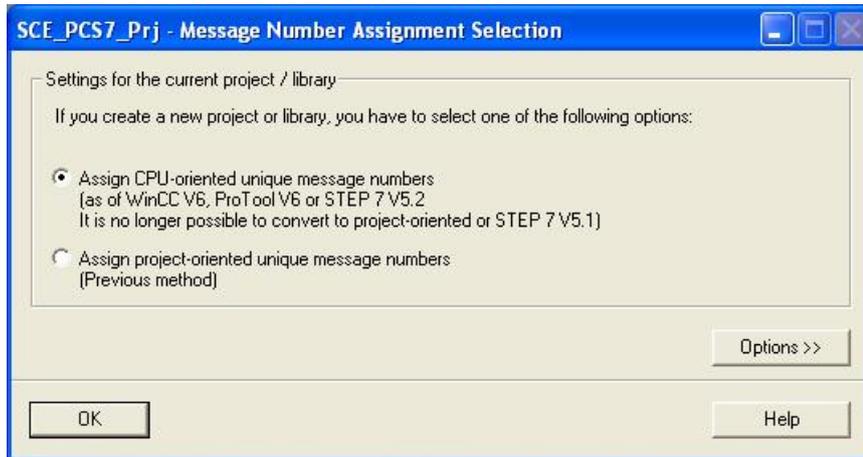
- In the window below, the storage location and the directory name (also project name) are specified and the project finished.

(→ Storage location: as required → Directory name: SCE_PCS7 → Finish)



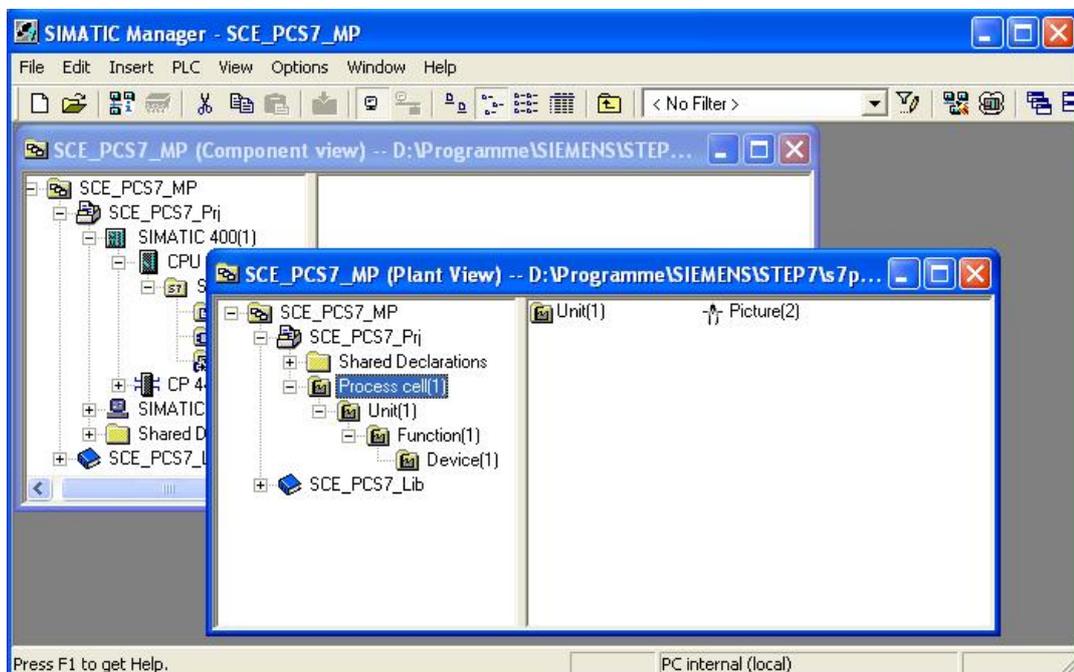
- As the message number concept, we select the current procedure where the message numbers are unique CPU-wide.

(→ Assign CPU-oriented unique message numbers → OK)



Note: In the case of the message system, message numbers are used for communication between AS and OS. In the OS, for each number a message with a message text and additional information is stored. If the AS sends a message number, the OS displays the associated message.

- After the project is finished, it is opened and displayed in the component view as well as the plant view. We can alternate between the views in the menu under View. (→ View)

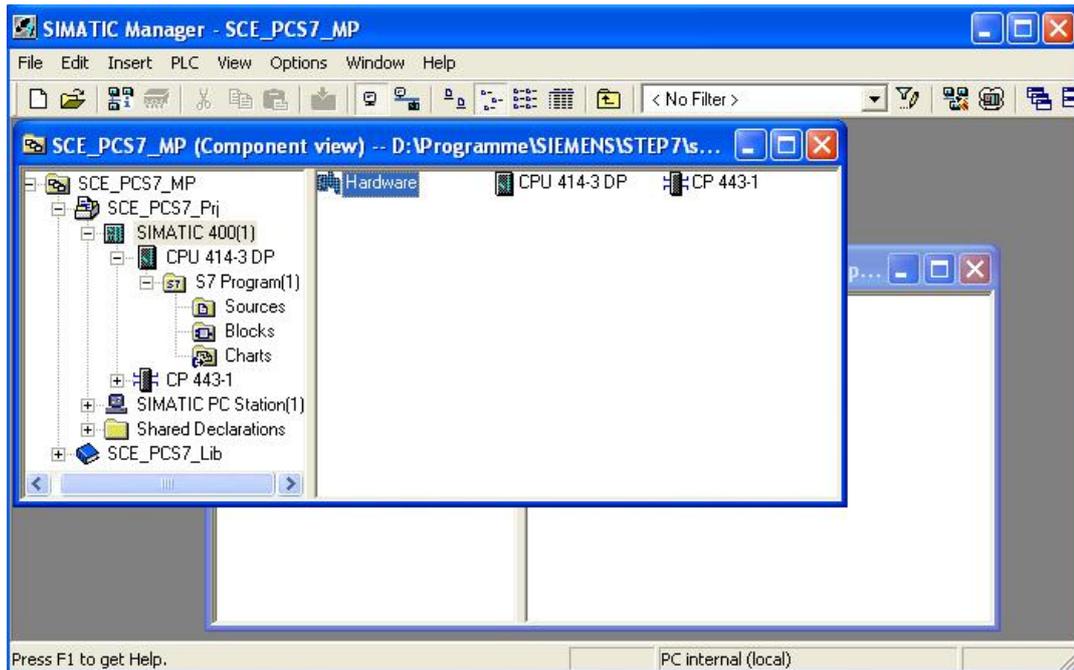




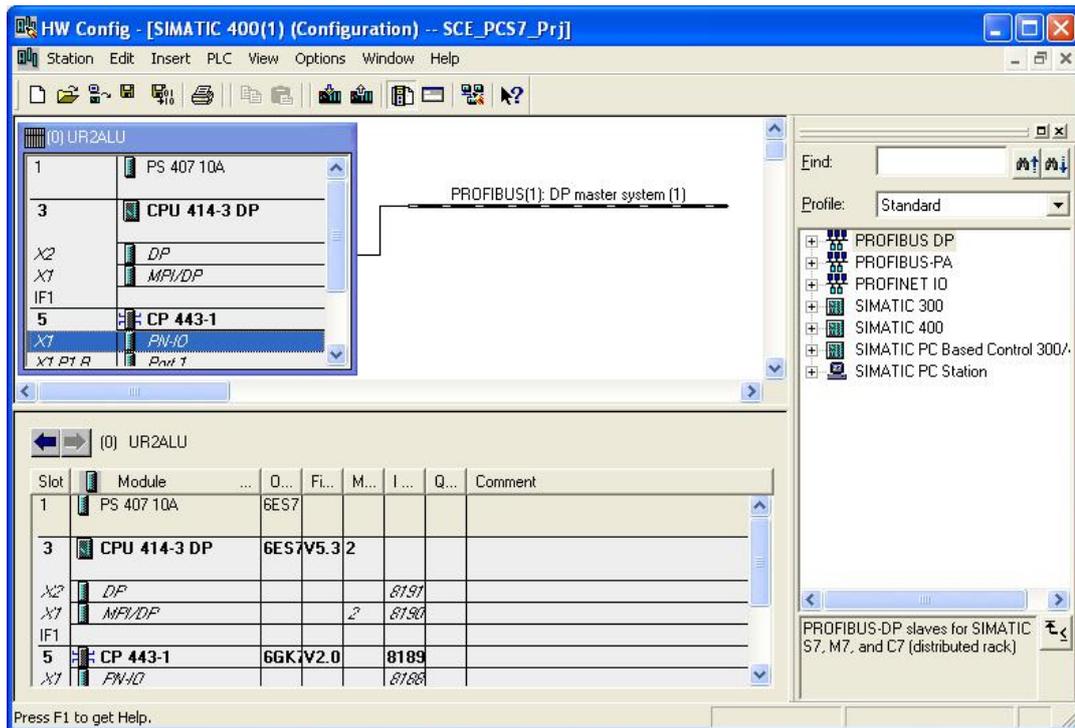
Note: The chapter 'Plant Hierarchy' below provides additional information regarding the component view and the plant view. In this chapter, we are using only the component view familiar from STEP7.

9. Next, in the component view we select the SIMATIC 400 station and with a double click open the hardware configuration.

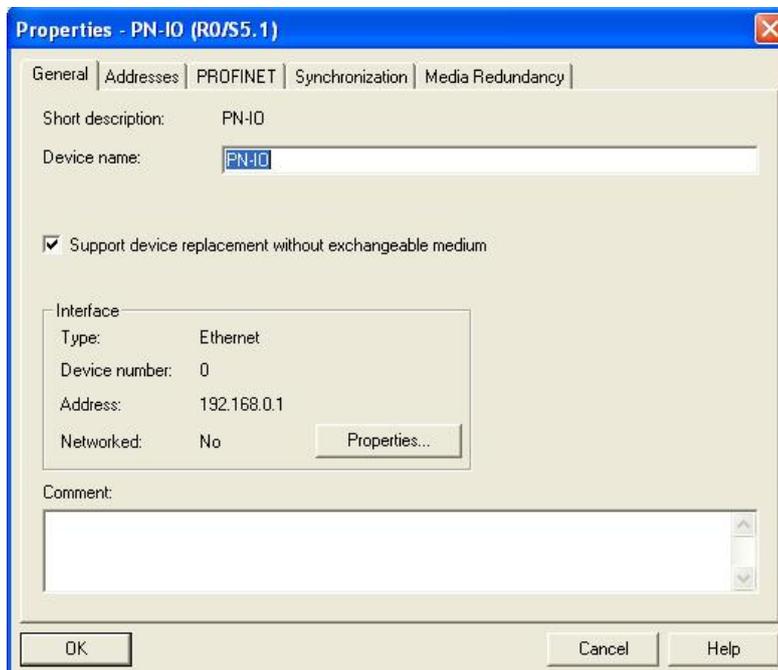
(→ Component view → SIMATIC 400(1) → Hardware)



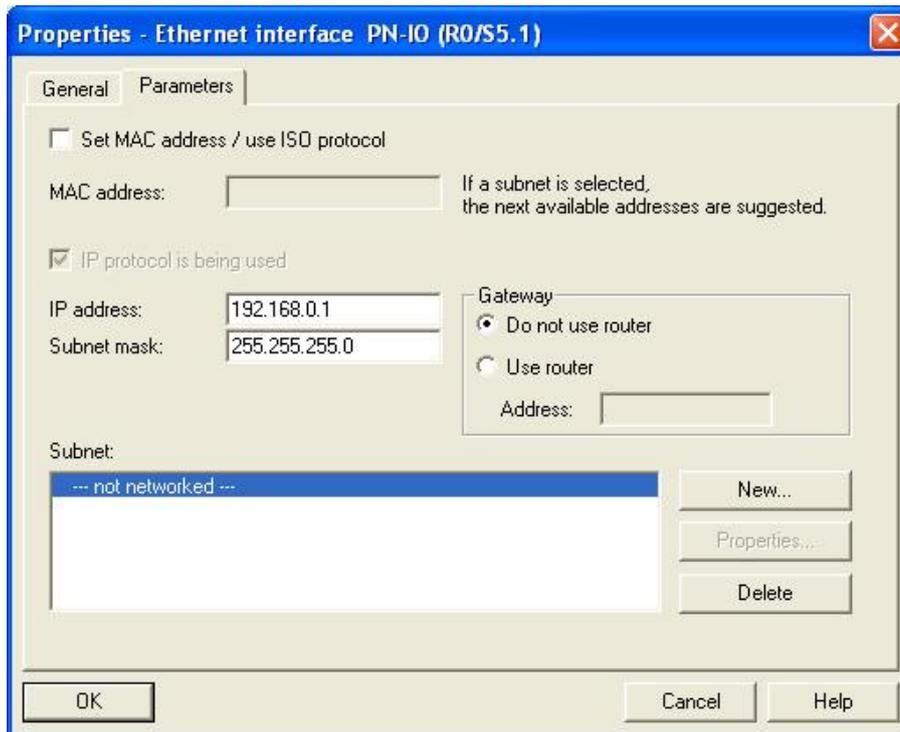
10. To make the settings for Ethernet networking, we select the PN IO interface with a double click on CP 443-1. (→ PN-IO)



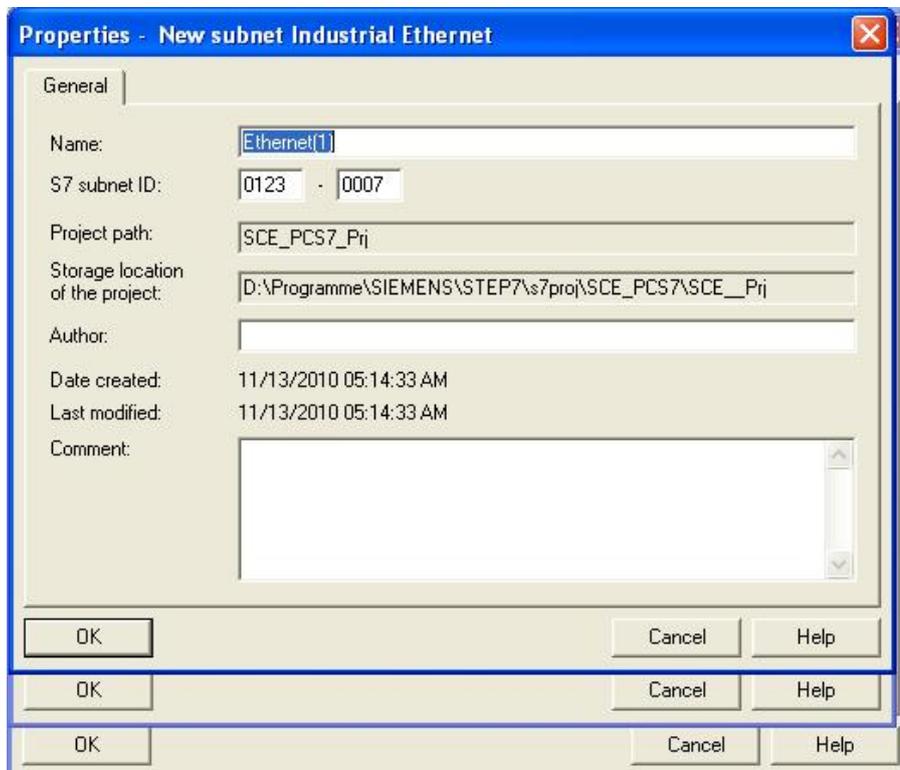
11. Here, we assign a device name and select the properties for the Ethernet interface (→ Properties)



12. In the parameters, we enter an IP address and subnet screen form, and set up a new subnet. (→ Parameters → 192.168.0.1 → 255.255.255.0 → New)

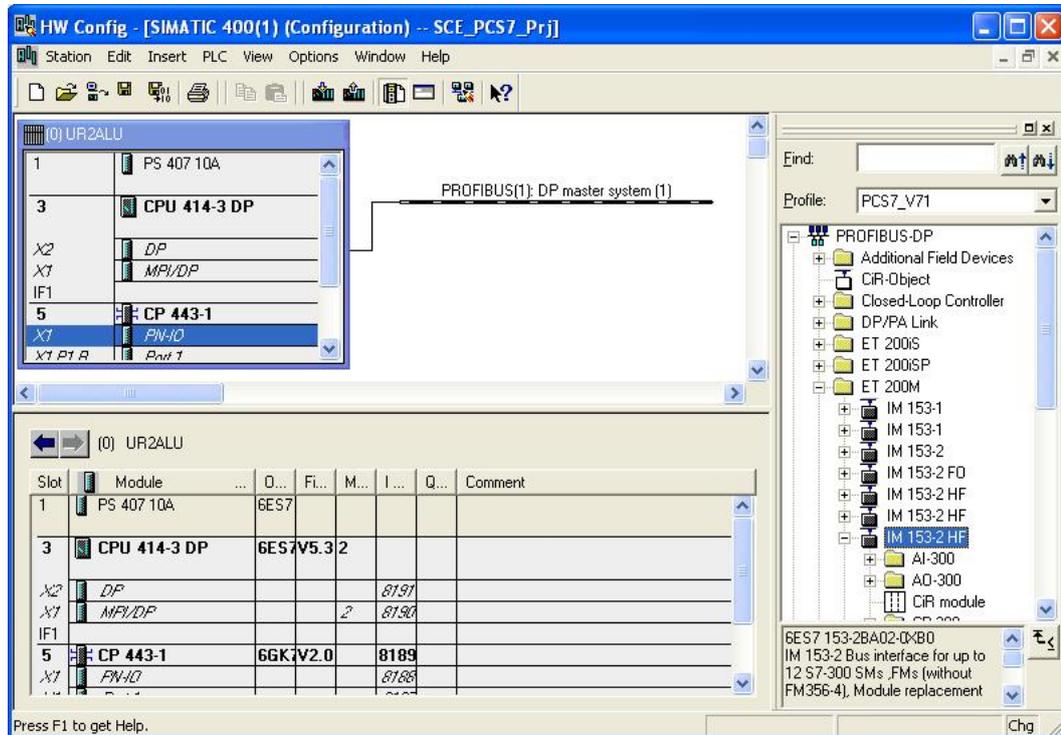


13. We then accept the subnet and the settings. (→ OK → OK → OK)



14. Next, we configure the ET200M as field device on the PROFIBUS. First, we have to select the matching interface module from the catalog in the folder PROFIBUS DP, ET200M, drag it to the CPU's master system and drop it there.

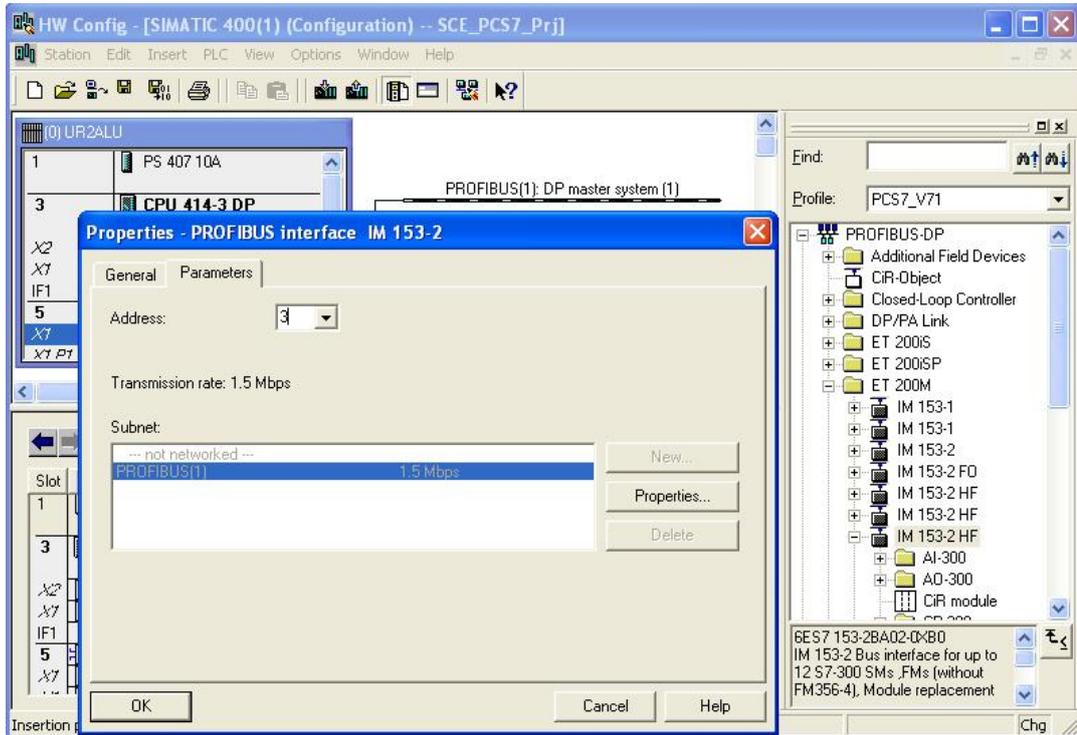
(→ PROFIBUS-DP → ET200M → IM 153-2 HF → PROFIBUS(1): DP master system(1))



Note: To select the exact interface module, you have to take note of the order numbers. They are printed on the interface module and are indicated in the footer of the hardware catalog after selecting a component.

If you don't have your own hardware available, follow the screenshot shown here.

15. In the selection below, we assign the PROFIBUS address for the interface module. (→ 3 → OK)



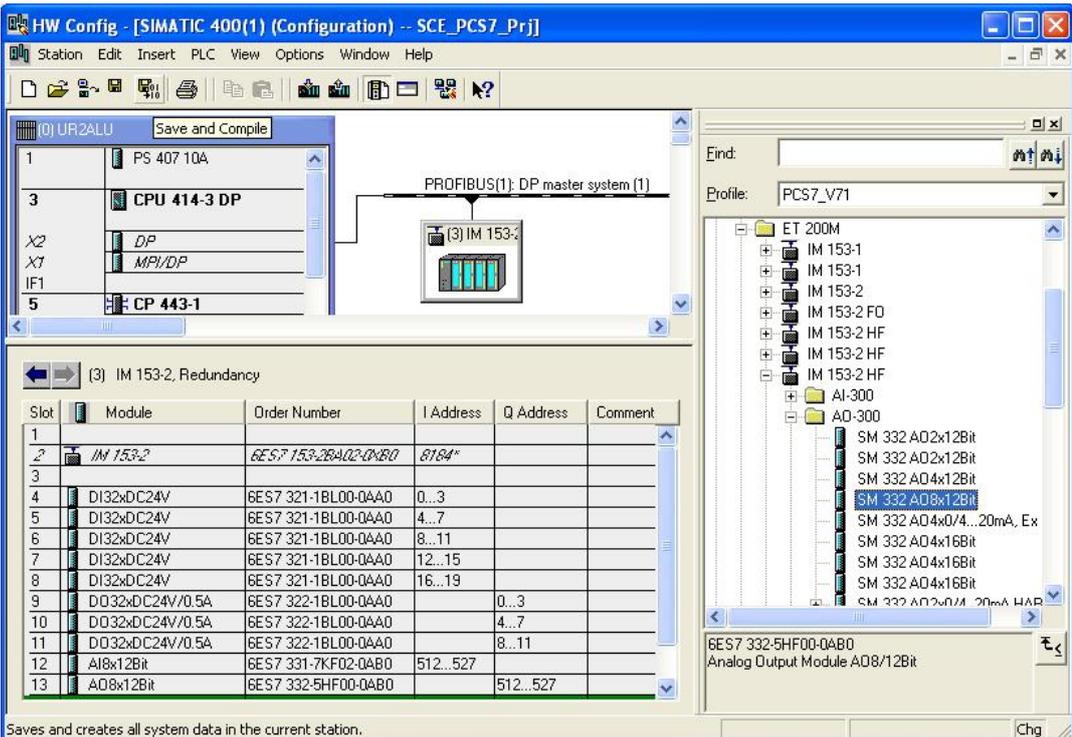
Note: The address set here also has to be set at the interface module in the binary code using a switch block.

16. Now -from the folders located below the interface module used- we enter the IO modules. This is done by dragging these modules to the respective slot within the ET200M and dropping them there.

The IO addresses of the individual modules should be set in their properties, as shown here.

Once your configuration is finished, accept it with the button  to save and compile.

(→ PROFIBUS DP → ET200M → IM 153-2 HF → DI-300 → DO-300 → AI-300 → AO-300
→ Set addresses → )



Slot	Module	Order Number	I Address	Q Address	Comment
1					
2	IM 153-2	6ES7 153-2BA02-0AB0	8184*		
3					
4	DI32xDC24V	6ES7 321-1BL00-0AA0	0...3		
5	DI32xDC24V	6ES7 321-1BL00-0AA0	4...7		
6	DI32xDC24V	6ES7 321-1BL00-0AA0	8...11		
7	DI32xDC24V	6ES7 321-1BL00-0AA0	12...15		
8	DI32xDC24V	6ES7 321-1BL00-0AA0	16...19		
9	DO32xDC24V/0.5A	6ES7 322-1BL00-0AA0		0...3	
10	DO32xDC24V/0.5A	6ES7 322-1BL00-0AA0		4...7	
11	DO32xDC24V/0.5A	6ES7 322-1BL00-0AA0		8...11	
12	AI8x12Bit	6ES7 331-7KF02-0AB0	512...527		
13	AO8x12Bit	6ES7 332-5HF00-0AB0		512...527	



Note: To select exactly the correct modules, take note of the order numbers. These are printed on the modules and indicated in the footer of the hardware catalog after selecting a component. If you don't have your own hardware available, follow the screenshot shown here.



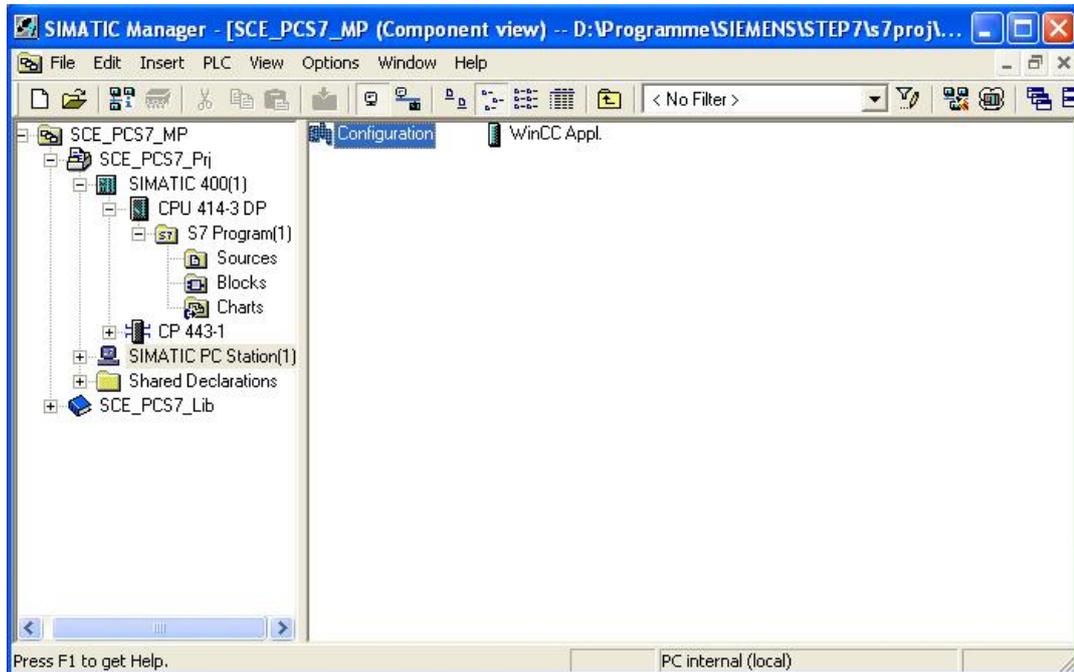
Note: Slot 3 remains empty since it is reserved for the expansion module with a multi-tier configuration.



Note: To use the specified symbol table, it is important to set the specified IO addresses here.

17. Next, in the component view of the SIMATIC Manager, we select the SIMATIC PC station and there, with a double click, we open the hardware configuration also.

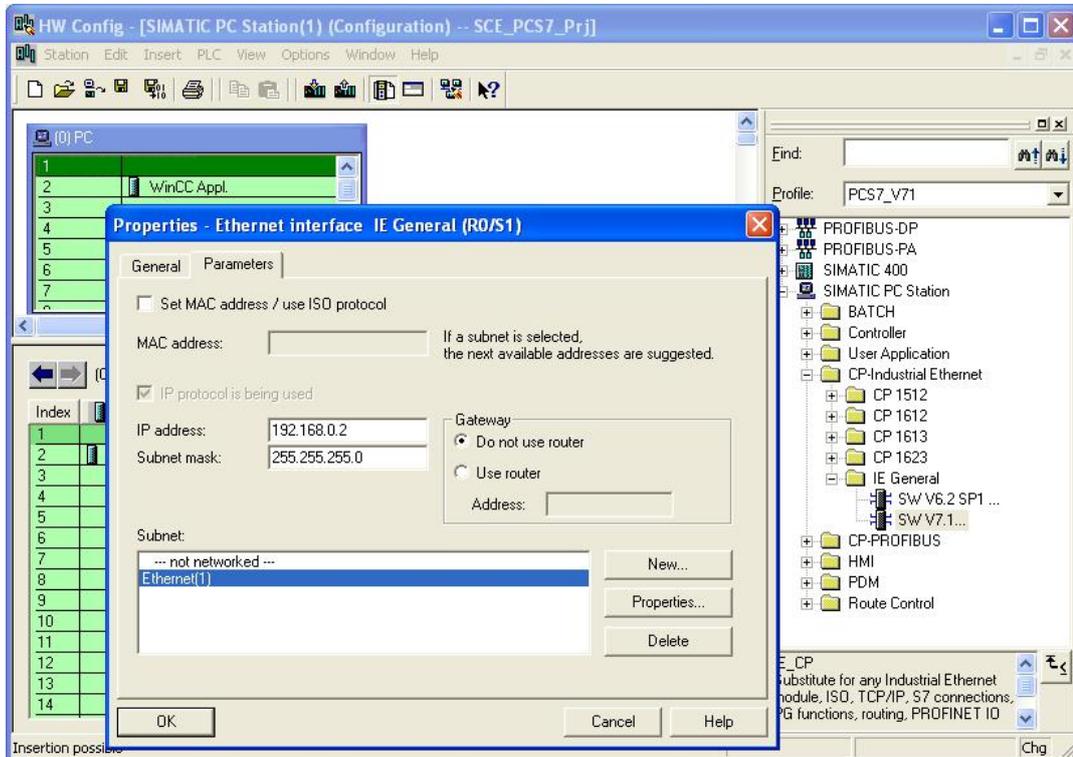
(→ Component view → SIMATIC PC Station(1) → Configuration)



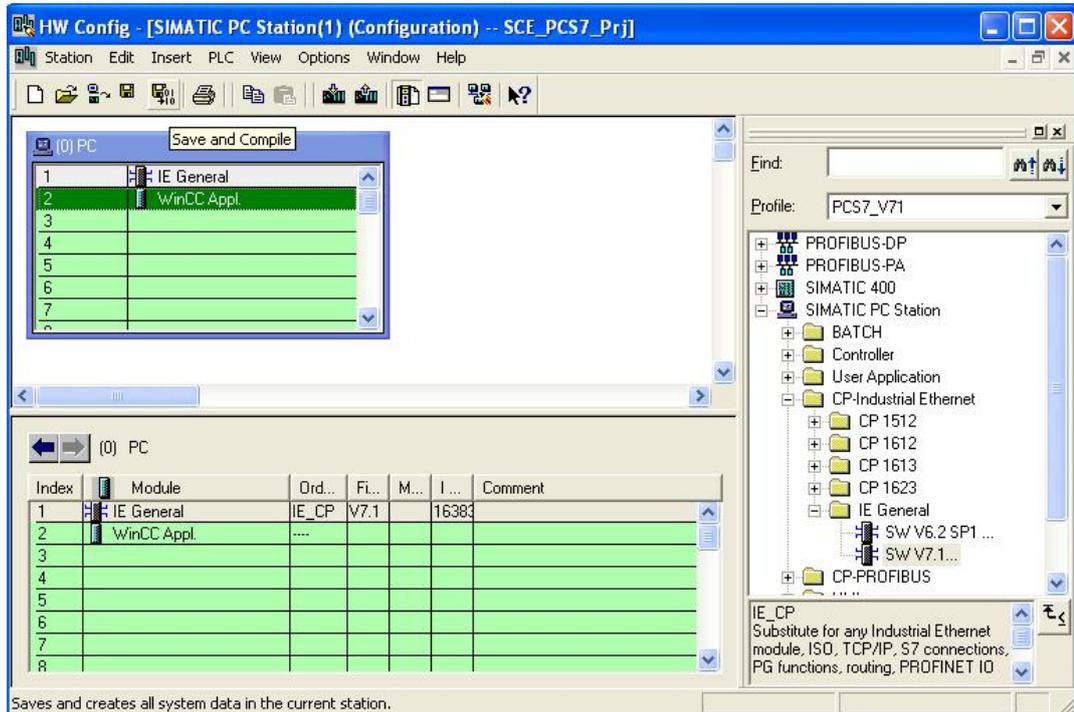
18. Within the PC station, first the Ethernet interface has to be entered. To this end, we drag the CP Industrial Ethernet in the V7.1 version from IE General to the first free slot in the PC station.

In the window that is then displayed, we network this interface with the Ethernet already set up in the S7 station, and enter the IP address and the subnet screen form.

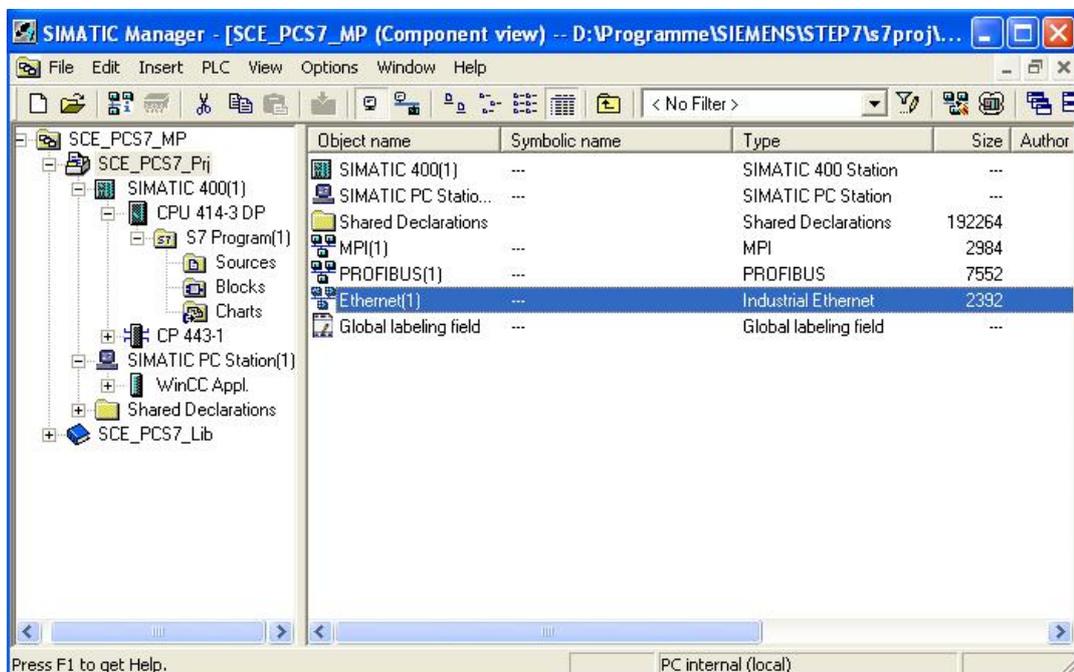
(→ SIMATIC PC Station → CP-Industrial Ethernet → IE General → SW V7.1 → Ethernet(1) → 192.168.0.2 → 255.255.255.0 → OK)



19. In the PC station, the **WinCC** application and an Ethernet interface are entered. We accept this configuration by clicking on the  button to Save and Compile. (→ )

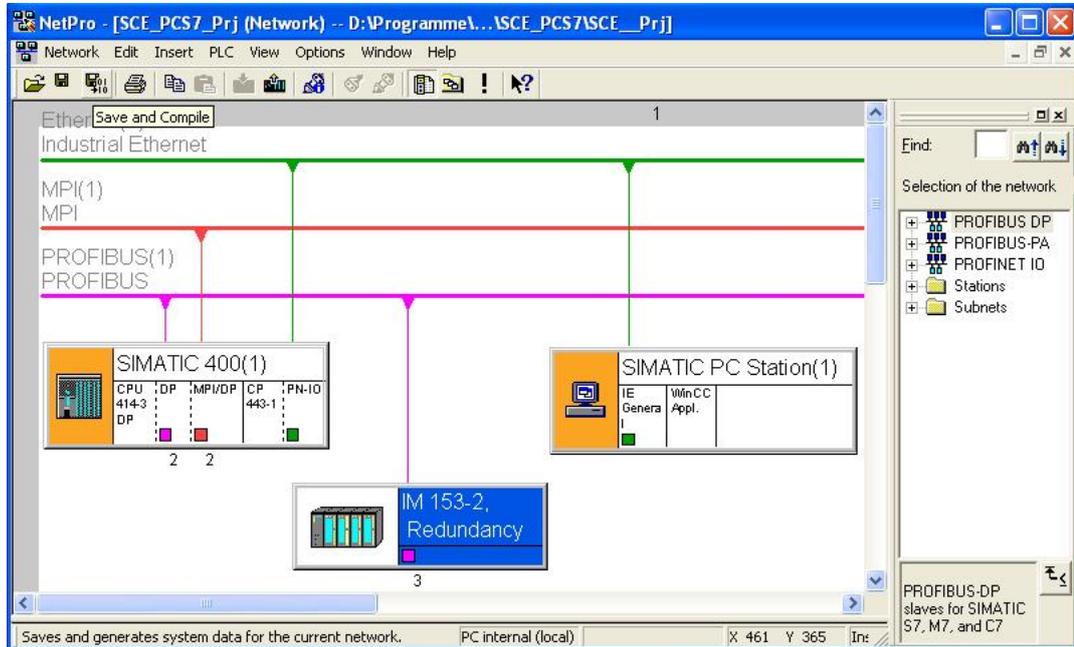


20. To check and compile the networking in our project, we open -in the component view of the SIMATIC Manager- the Ethernet with a double click. (→ Component view → SCE_PCS7_Prj → Ethernet(1))



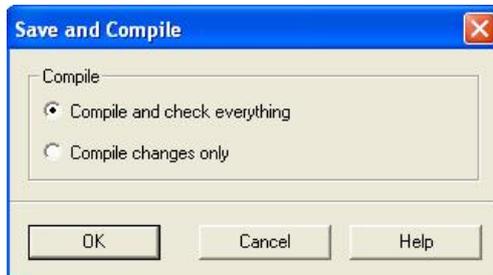
21. The tool **NetPro** provides us with a nice overview of the components and networks in our project. We can see that both stations are connected to each other by means of the Ethernet, and the ET200M is interfaced by means of PROFIBUS with the

SIMATIC400. We accept these networking settings by clicking on the button 'Save and Compile' (→ ) to Save and Compile. (→ )

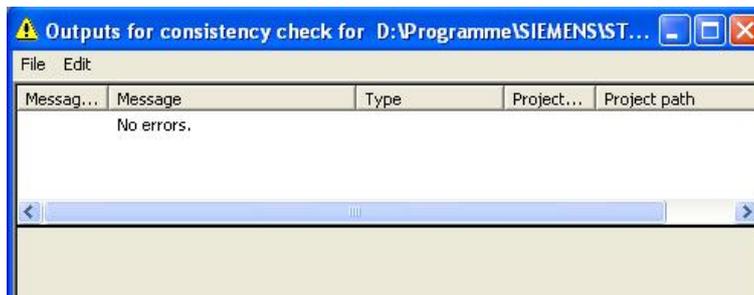


22. In the window that follows, we select Compile and check everything.

(→ Compile and check everything → OK)



23. The result of the compilation is displayed in a window. (→ )



EXERCISES

We are now going to apply what we learned from the theory chapters and the step by step instructions to the exercises. To this end, we are using the existing multi-project from the step by step instructions and expand it.



Note: The exercises can also be performed without having completely and correctly worked through the step by step instructions beforehand. To get to the required processing status, load the included project archive 'PCS7_SCE_0102_R1009.zip' with the function 'Dearchive' (→ File → Dearchive...).

For the following exercises it is assumed that the experimental plant is utilized in the context of a larger technical college plant. Additional needed automation stations are configured and implemented by means of a new subproject of the existing multi-project. The following new automation stations have to be processed:

The AS2 has the same equipment as the automation station existing in the project. However, different analog modules are used for a more accurate resolution of the analog input and output signals.

The AS3 is to be equipped with the 417-4 DP CPU to which the three ET200Ms are connected.

TASKS

1. Rename the 'SIMATIC 400 (1)' as AS1 and the 'SIMATIC PC STATION (1)' as ES/OS1.
2. Create a new project 'SCE_PCS7_Prj2' in your multi-project.
3. Insert a copy of AS1 in the new project and rename it as AS2.
4. Open the hardware configuration of the AS2 and interface it with a new Ethernet and a new Profibus subnet 'Profibus (2)'.
5. Replace the analog input and output modules by means of the context menu 'Exchange object' with modules that have a higher resolution:
 - AI8x12 Bit → AI8x16 Bit
 - AO8x12 Bit → AO4x16 Bit
6. Now, insert a second SIMATIC 400 in the project and name it AS3. Then open the hardware configuration of the AS3 and add the following components from the hardware catalog:
 - UR 2 (6ES7 400-1JA00-0AA0)
 - PS 407 10A (6ES7 407-0KA01-0AA0)
 - CPU 417-4 (6ES7 417-4XT05-0AB0)
 - CP 443-1 (6GK7 443-1EX20-0XE0)
7. Interface the CP with the existing Ethernet and for the DP connection, set up a new Profibus line 'Profibus (3)'.
8. Next, copy the ET200M from the AS1 and add it twice to the Profibus line. Observe the IO addresses and the Profibus addresses.
9. Now, configure an ET200M yourself from the hardware catalog. The order numbers in the parentheses clearly identify the modules.

- IM 153-1 (6ES7 153-1AA83-0XB0)
 - SM 321 DI16x24VDC (6ES7 321-1BH01-0AA0)
 - SM 321 DI16x24VDC (6ES7 321-1BH02-0AA0)
 - SM 323 DI8/DO8x24VDC/0.5A (6ES7 323-1BH01-0AA0)
 - SM 322 DO16x24VDC/0.5A (6ES7 322-1BH01-0AA0)
 - SM 322 DO8x24VDC/0.5A (6ES7 322-8BF00-0AB0)
 - SM 331 AI2x12 Bit (6ES7 331-7KB02-0AB0)
 - SM 332 AO2x12 Bit (6ES7 332-5HB01-0AB0)
10. Connect the Ethernet networks from both projects by going to 'Multi-project' → 'Match Project'. Check whether the IP addresses are unique.