



SIEMENS

SCE Training Curriculum

Siemens Automation Cooperates with Education (SCE) | 09/2015

PA Module P01-02 SIMATIC PCS 7 – Hardware Configuration

Cooperates
with Education

Automation

SIEMENS

Matching SCE Trainer Packages for these curriculum

- **SIMATIC PCS 7 Software block of 3 packages**
Order No. 6ES7650-0XX18-0YS5
- **SIMATIC PCS 7 Software block of 6 packages**
Order No. 6ES7650-0XX18-2YS5
- **SIMATIC PCS 7 Software Upgrade block of 3 packages**
Order No. 6ES7650-0XX18-0YE5 (V8.0 → V8.1) or 6ES7650-0XX08-0YE5 (V7.1 → V8.0)
- **SIMATIC PCS 7 Hardware Set including RTX Box**
Order No. 6ES7654-0UE13-0XS0

Please note that these trainer packages may be replaced with subsequent packages.

An overview of the available SCE packages is provided at: [siemens.com/sce/tp](https://www.siemens.com/sce/tp)

Continuing education

For regional Siemens SCE continuing education, contact your regional SCE contact partner.

[siemens.com/sce/contact](https://www.siemens.com/sce/contact)

Additional information relating to SIMATIC PCS 7 and SIMIT

In particular, Getting Started, videos, tutorials, manuals and programming guide.

[siemens.com/sce/pcs7](https://www.siemens.com/sce/pcs7)

Additional information relating to SCE

[siemens.com/sce](https://www.siemens.com/sce)

Note on Usage

The training curriculum for the integrated automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)" specifically for training purposes at public educational and R&D facilities. Siemens AG is not liable for the contents.

This document may only be used for initial training on Siemens products/systems. This means it may be copied entirely or partially and handed to trainees for use within the scope of their training. Passing on or copying this document and communicating its contents is permitted within public training and continuing education facilities for training purposes.

Exceptions require written permission by Siemens AG. Contact person: Roland Scheuerer
roland.scheuerer@siemens.com.

Violators are subject to damages. All rights including translation rights are reserved, particularly in the event a patent is granted or a utility model or design is registered.

Usage for industrial customer courses is explicitly not permitted. We do not agree to the commercial utilization of these documents.

We would like to thank the Technical University Dresden, particularly Prof. Dr. Leon Urbas and Annett Krause, MS, as well as the Michael Dziallas Engineering Corporation and those who provided support in preparing this SCE training document.

HARDWARE CONFIGURATION

TRAINING OBJECTIVE

In this chapter, the students learn how an automation system works. They will be able to configure the selected hardware in the hardware configuration of **PCS 7** and check it for consistency. Important settings will be parameterized so that from this data the process control system **PCS 7** automatically sets up all blocks necessary for communication between sensors, actuators and the control level in the automation stations.

THEORY IN BRIEF

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

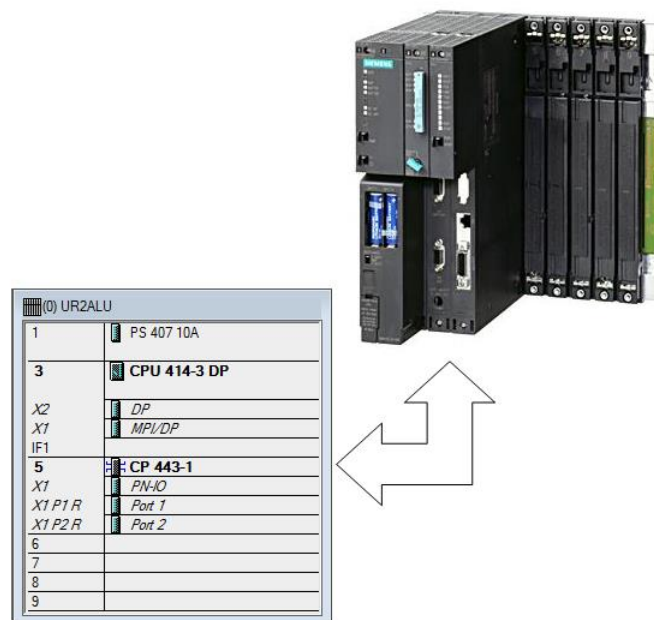


Figure 1: Mapping the real configuration to the configuration table [1]

When storing and compiling the settings that were made, the internal consistency is checked (do all modules match, have addresses been assigned twice, etc.). As soon as the configuration is consistent within itself, the blocks necessary for the communication of the process data are set up without user assistance and can then be loaded to the **automation stations (AS)** of the control system. These now have all the information to initially determine whether the installed configuration corresponds to the one that was configured. They then make sure that the process data are available at the ASs in the desired scanning frequency for further processing. For most users it is not apparent that extensive provisions are made in case an error occurs, such as the automatic transmission of suitable messages and alarms.

THEORY

DISTRIBUTED ARCHITECTURE OF PROCESS CONTROL SYSTEMS

Scalable process control systems such as **PCS 7** cover a large range of processes. Applications include small laboratory systems with only a few sensors and actuators as well as systems that have a hundred thousand process tags. To cover this broad spectrum, special structures are required. A typical component structure that can be easily expanded looks like this:

- On the process management level, an **operator system (OS)** consisting of one or several operator stations is used. By means of these OSs, the control operators can operate and monitor the system.
- On the control level, one or several **automation stations (AS)** are located on which the control functions and closed loop control functions are carried out in real time. These systems are to do their job regardless of OS failures. They consist of at least one power supply (PS) and a CPU as well as communication modules (CP), if necessary.
- On the field level, the states of the technical processes are recorded or the process is specifically influenced by means of sensors and actuators.

While all components can run on a single computer system in a lab—as seen in Figure 2—, distributing the components is advisable when the size of a test center is extensive to handle the complexity. Depending on the requirements, different bus systems that possess the necessary real time properties for process data communication, for example, are used to exchange data between these components.

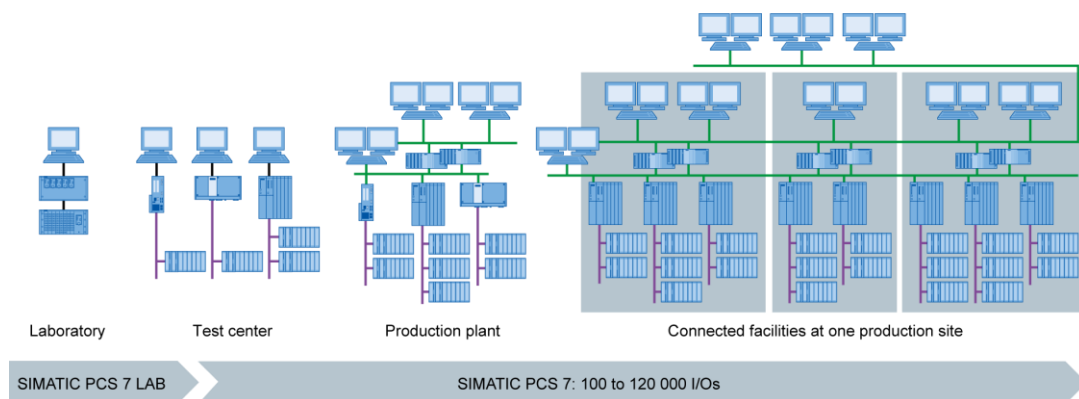


Figure 2: Scalable structure of the process control system PCS 7 [2]

PROCESS IMAGE

The automation programs are processed on the automation stations (AS) in the central processing unit (CPU) as in any other computer. The control programs are processed cyclically. If signals from the process are processed, the program does not directly access the connected input and output modules; instead, a so-called process image is generated where all signals are entered at once and stored at the beginning of the cycle.

There are two reasons for this: First, accessing the process image requires considerably less time because it is stored in the internal memory of the CPU. Second, it 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 entering the signals of the input modules once for each cycle in the process image of the inputs (PII). Next, the program is processed, and the results are written to the process image of the outputs (PIQ).

Once the entire program is processed, the data is written from the PIQ to the output modules and only then read out to the process. The process image of the inputs is then updated once again as shown Figure 3.

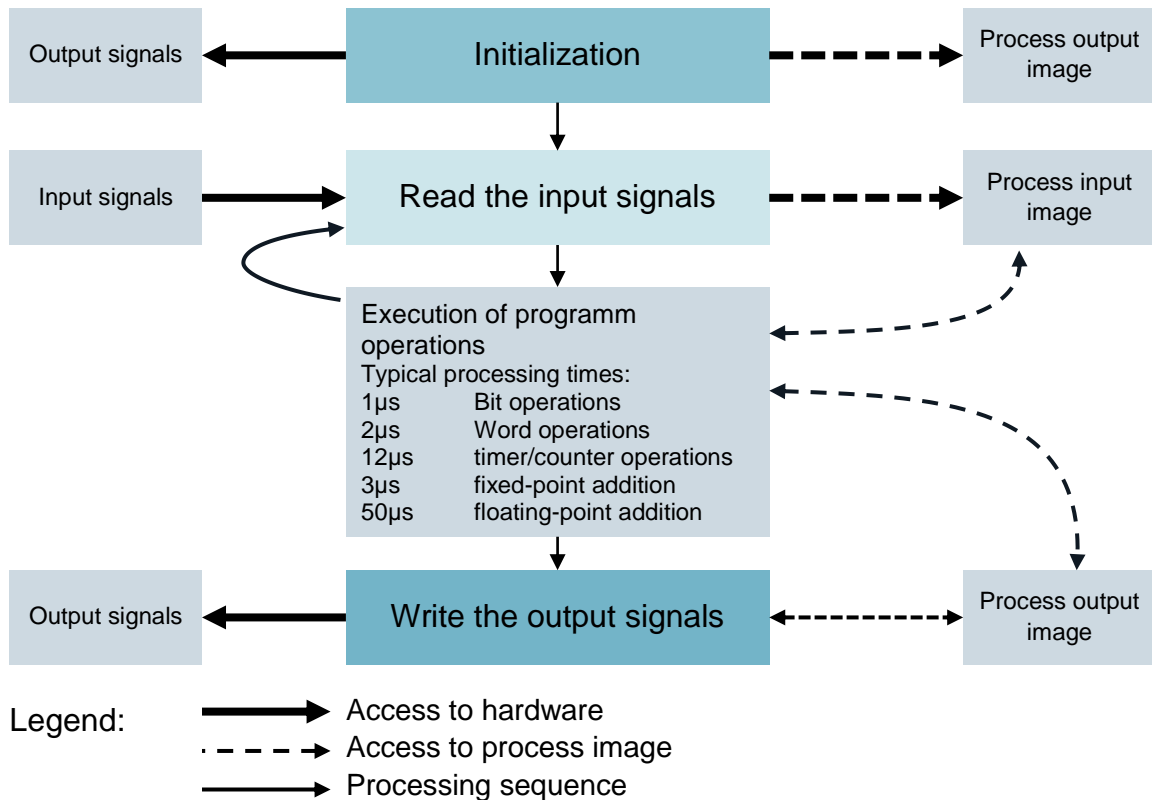


Figure 3: Reading and writing input and output signals during processing of the PLC cycle

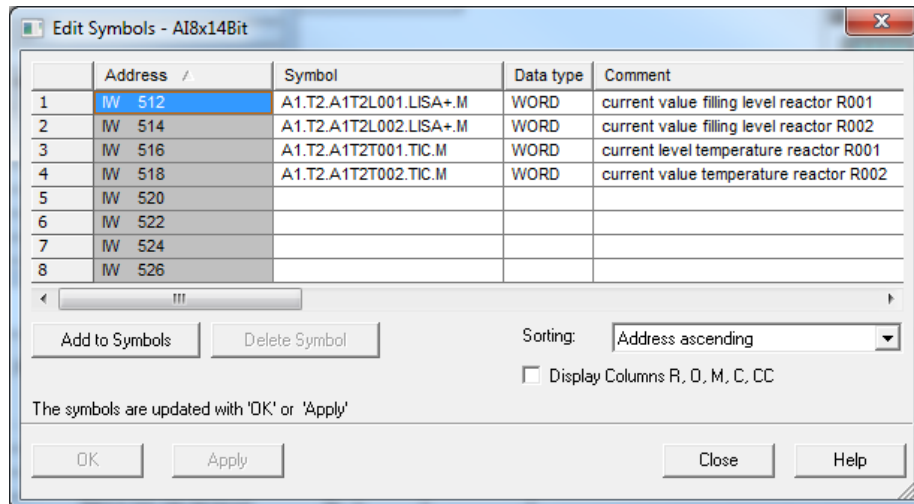
CONNECTION TO THE PROCESS

The process signals recorded by the sensors, such as temperature, pressure, level and flow rate, are converted into an electrical signal with transducers. If the measuring device is not connected directly via fieldbus, the signal is usually converted into an electrical standard signal. It can then be recorded on the side of the automation system by a standardized signal module.

Because process industry systems are to record a handful or up to several thousand measured values, it must be possible to select, clearly assign and parameterize the measuring module when automation is planned. This means the required signal modules are initially arranged virtually in the hardware configuration. At the same time, memory is assigned (as mentioned previously) in the process image to the signal modules. As soon as a signal module is included in the configuration, sufficient memory is automatically assigned in the process image. This automatic assignment can then be changed manually; however, the size of the CPU memory area must be taken into consideration.

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

For analog signals, AI (Analog Input) and AO modules (Analog Output) are used. Analog modules are usually organized in words (16 bits). Each analog input or output signal is assigned 16 bits of memory (as seen in Figure 4). To do this, the analog input module converts the analog process signal into a digital one. Depending on the resolution, only the higher order positions are assigned and '0' is written to the lower order positions. Analog output modules convert the digital output value into an analog signal. Regarding analog signals, modules are distinguished not only 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	IW 512	A1.T2.A1T2L001.LISA+.M	WORD	current value filling level reactor R001
2	IW 514	A1.T2.A1T2L002.LISA+.M	WORD	current value filling level reactor R002
3	IW 516	A1.T2.A1T2T001.TIC.M	WORD	current level temperature reactor R001
4	IW 518	A1.T2.A1T2T002.TIC.M	WORD	current value temperature reactor R002
5	IW 520			
6	IW 522			
7	IW 524			
8	IW 526			

Figure 4: Symbol table of an analog input module

DISTRIBUTED IO

In case sensors and signal sources are located far away from the automation system, wiring can become very extensive and complex. In addition, electro-magnetic interferences can limit reliability. The use of distributed IO devices makes sense for such systems:

- The automation system is at a central location.
- One or several IO devices (input and output modules) process distributed on site.
- Data is transferred between the periphery and the automation system [1] by means of PROFIBUS DP (distributed IO). To this end, the AS and the periphery have to be equipped with corresponding communication modules.

A SIMATIC ET200M was selected as the distributed IO device for the system described in the previous chapter. The IO modules of the proven S7-400 automation system are connected to an interface module (IM 153-x) that ensures communication to the AS. Figure 5 shows a typical configuration. On the right, several digital and analog input and output modules are connected to the interface module IM 153-1. The process signals coming from the field are connected directly to the routing level placed under the input and output modules; from there, short cables lead to the modules which means wiring faults to the field can be quickly corrected.

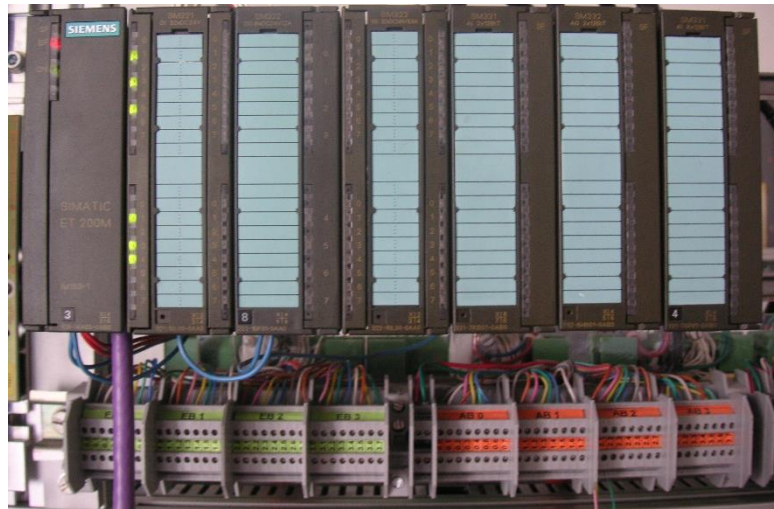


Figure 5: Distributed I/O device ET200M (Source: Laboratory Plant TU Dresden)

In the hardware configuration, the SIMATIC ET200M is connected directly to a PROFIBUS DP line of the AS as seen in Figure 6. The hardware configuration automatically recommends addresses that are not yet used in the selected subnet. The slots of the ET200M are assigned to input and output modules as described below.

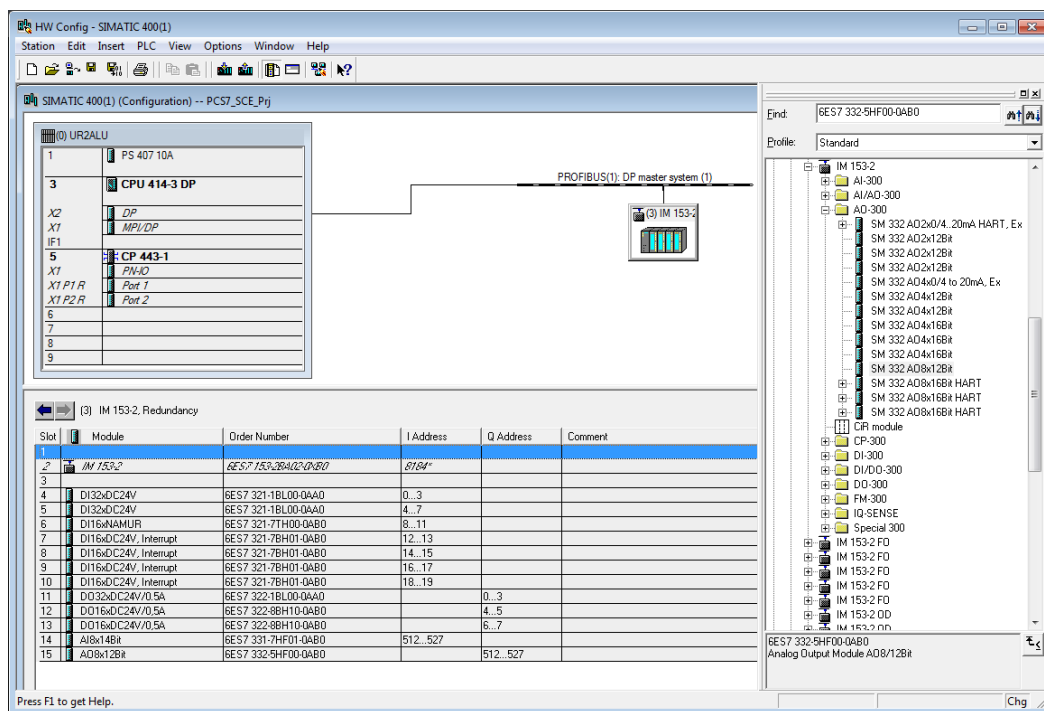


Figure 6: Hardware configuration of an ET200M

LITERATURE

- [1] Online Help for PCS 7. Siemens
- [2] www.automation.siemens.com
- [3] SIEMENS (2010): Configuring Hardware and Communication Connections STEP 7. A5E02789741-01. (<http://support.automation.siemens.com/WW/view/en/45531110>)

STEP BY STEP INSTRUCTIONS

TASK

In this chapter, the **PCS 7** project for the multi-purpose plant is to be set up with a wizard.

Next, the included S7 station is configured. In the example, a SIMATIC S7-400 is used with a CPU 414-3 DP and a CP443-1 communication processor for interfacing to an Ethernet by means of the TCP/IP protocol.

The IO signals for controlling the actuators in the plant and for recording the input signals are connected using the ET200M. This modular field device is connected to the CPU by means of the PROFIBUS DP fieldbus.

The PC station as master computer with the **PCS 7** software and **WinCC** for visualization has to be configured as well. You can use any PC or laptop with a standard Ethernet interface for this purpose.

The master computer as operator station (OS) with the CPU as automation station (AS) is interfaced to an Ethernet by means of the TCP/IP protocol.

The project is also developed on the master computer. This makes the master computer the operator station (OS) as well as the engineering station (ES).

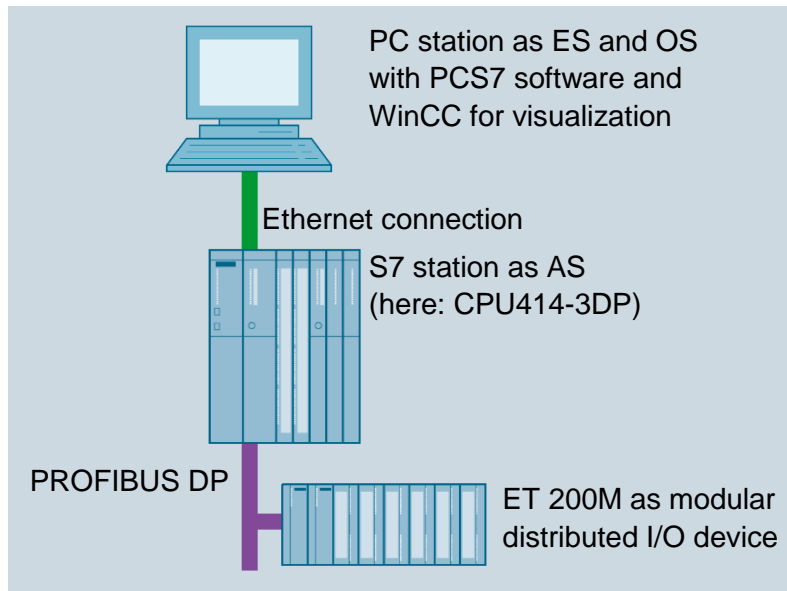


Figure 7: Plant configuration for multi-purpose plant



Note: The abbreviations **Engineering Station (ES)**, **Operator Station (OS)** and **Automation Station (AS)** should be committed to memory, because these terms are used frequently in the **PCS 7** software and in this document.



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.

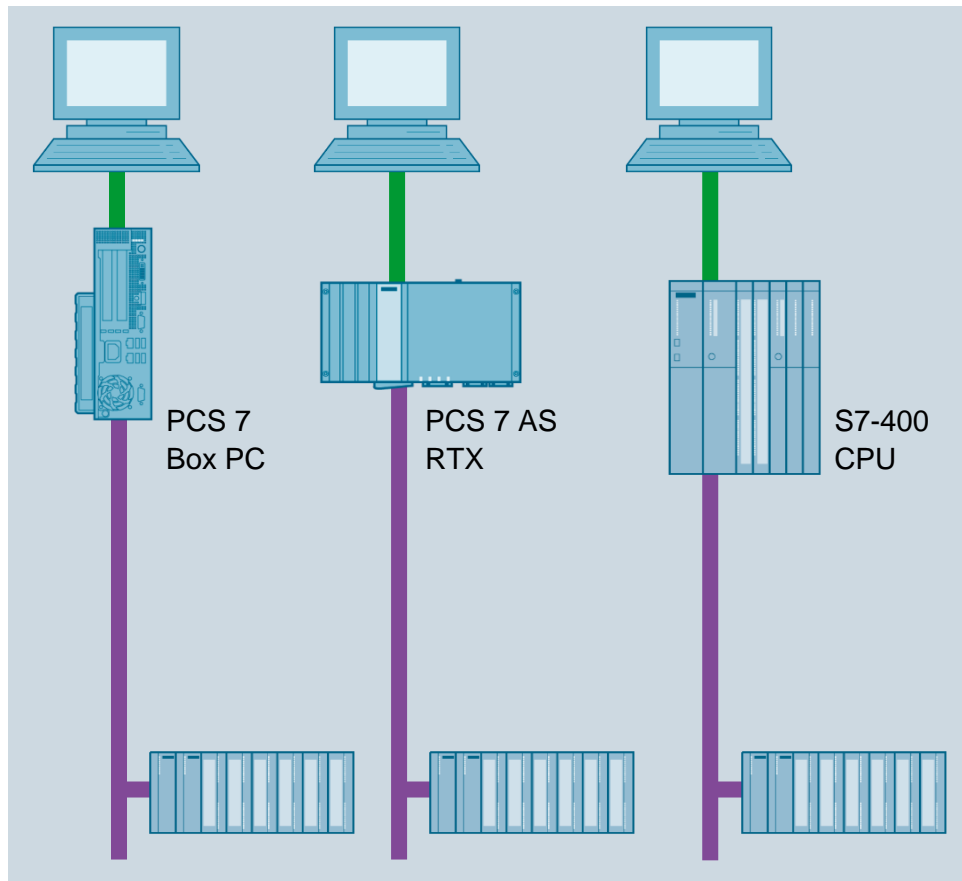


Figure 8: Various plant configurations with SIMATIC PCS 7 Box, SIMATIC PCS 7 AS RTX and SIMATIC S7-400 CPU 414-3DP as controller (from left)



Note: In the following chapters, the simulation software S7-PLCSIM is used for testing the programs. This means you can basically configure any controller.

TRAINING OBJECTIVE

In this chapter, the student learns the following:

- Creating a **PCS 7** project
- Generating the hardware configuration for an S7 station
- Generating the hardware configuration for a PC station with **WinCC**
- Networking the S7 station and the PC station

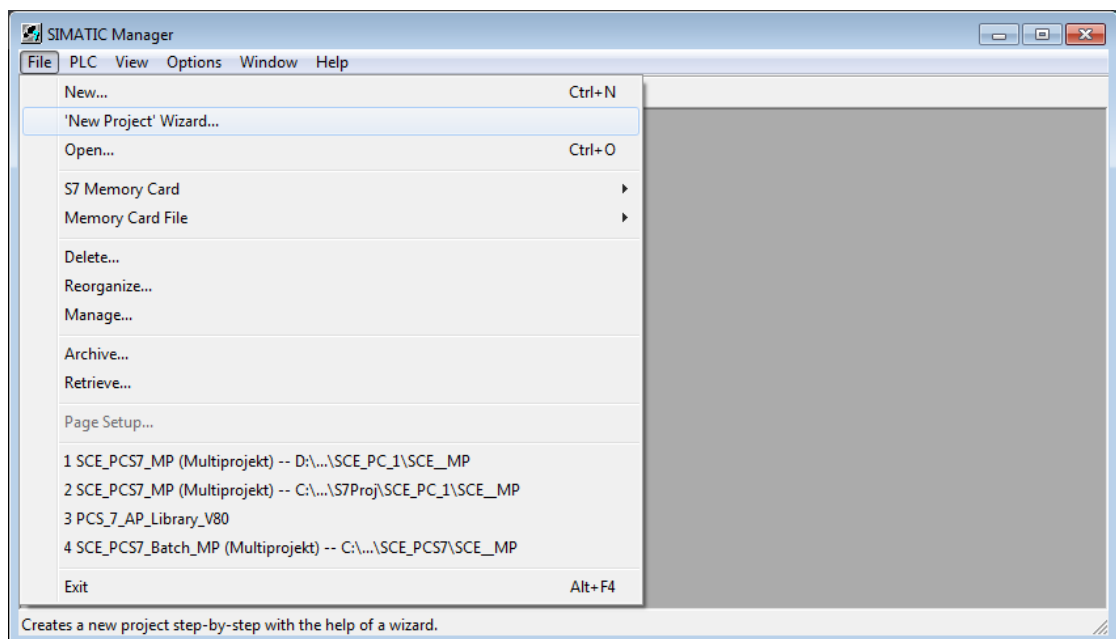
PROGRAMMING

1. The central tool in **PCS 7** is the **SIMATIC Manager**. Here, it is called with a double click.
(→ SIMATIC Manager)

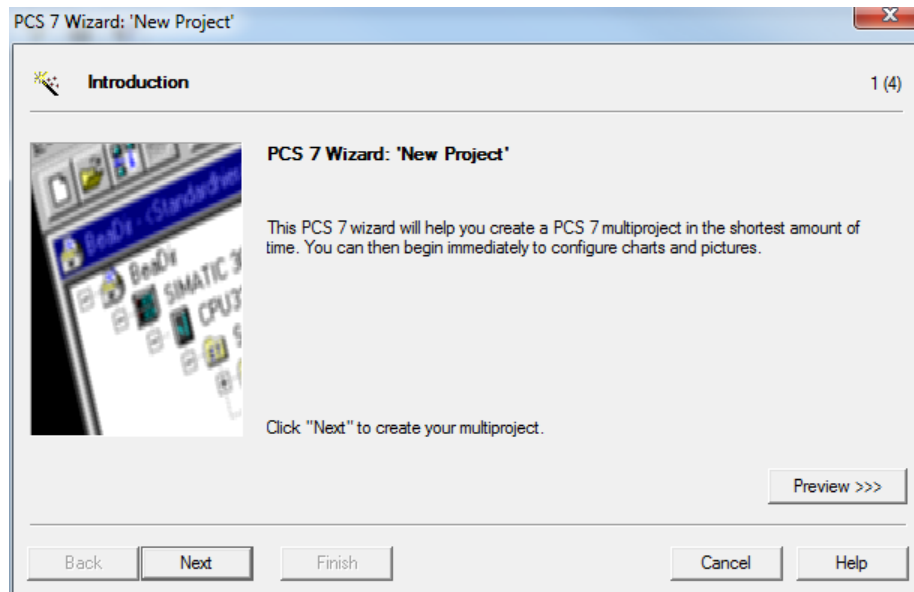


SIMATIC Manager

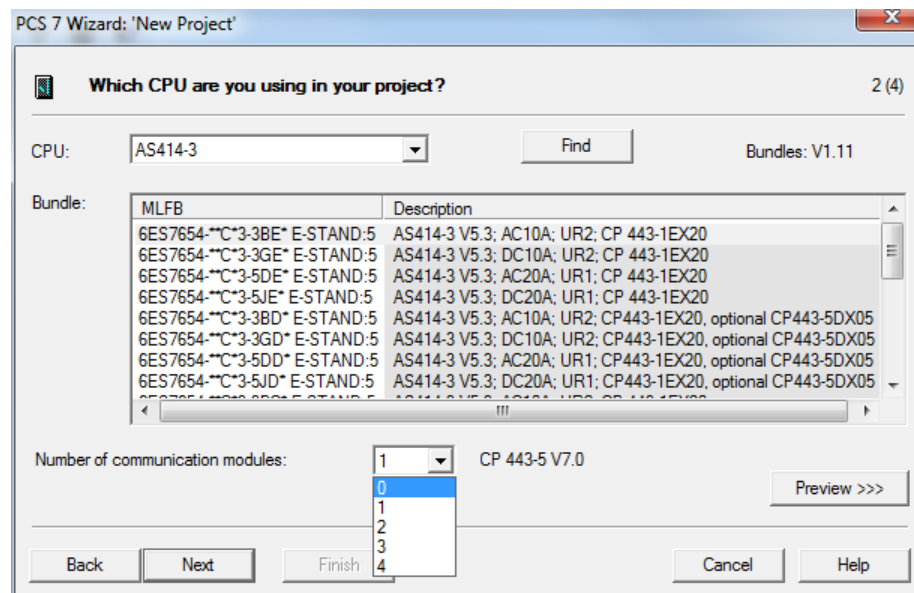
2. To create a **PCS 7** project, the use of a wizard is recommended because it sets up both the S7 station and the PC station in one step.
(→ File → 'New Project' Wizard...)



3. We are going to create the project as a multi-project. This means a master library is created in addition to the S7 station and the PC station. The library ensures that the same version of blocks and chart templates (process tag types) is always used within a project. (→ Next)



4. Next we select the configuration of the AS with the CPU used, the power supply and the communication processors for PROFIBUS and Ethernet. Because **PCS 7** stations are usually ordered as an entire station (bundle), the bundles can be selected based on their order numbers here. (→ CPU: AS414-3 → Bundle: 6ES7*** → Number of communication modules CP443-5: 0 → Next)



Note: The number of additional communication modules for PROFIBUS is included here in an additional selection.

Because the bundles listed often do not correspond always match the available S7 stations exactly, individual components may have to be added or exchanged later in the hardware configuration.

5. We now select the number of hierarchy levels for the plant hierarchy (refer to chapter Plant Hierarchy) and an OS object.

(→ 3 → PCS7 OS → Single station system → Next)

PCS 7 Wizard: 'New Project'

Which objects are you still using? 3 (4)

Plant hierarchy :
Number of levels: 3

AS objects :
☐ CFC chart
☐ SFC chart

OS objects :
☒ PCS7 OS
☐ SIMATIC BATCH
☐ Route Control
☐ OpenPCS 7

Single station system
Multiple station system
Multiple station system redundant

Preview >>>

Back Next Finish Cancel Help

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

(→ Storage location: any → Directory name: PCS7_SCE → Finish)

PCS 7 Wizard: 'New Project'

Where do you want to store the multiproject? 4 (4)

Directory name: PCS7_SCE

The following objects will be created:
Multiproject: PCS7_SCE_MP
Project: PCS7_SCE_Pj
Master data library: PCS7_SCE_LIB

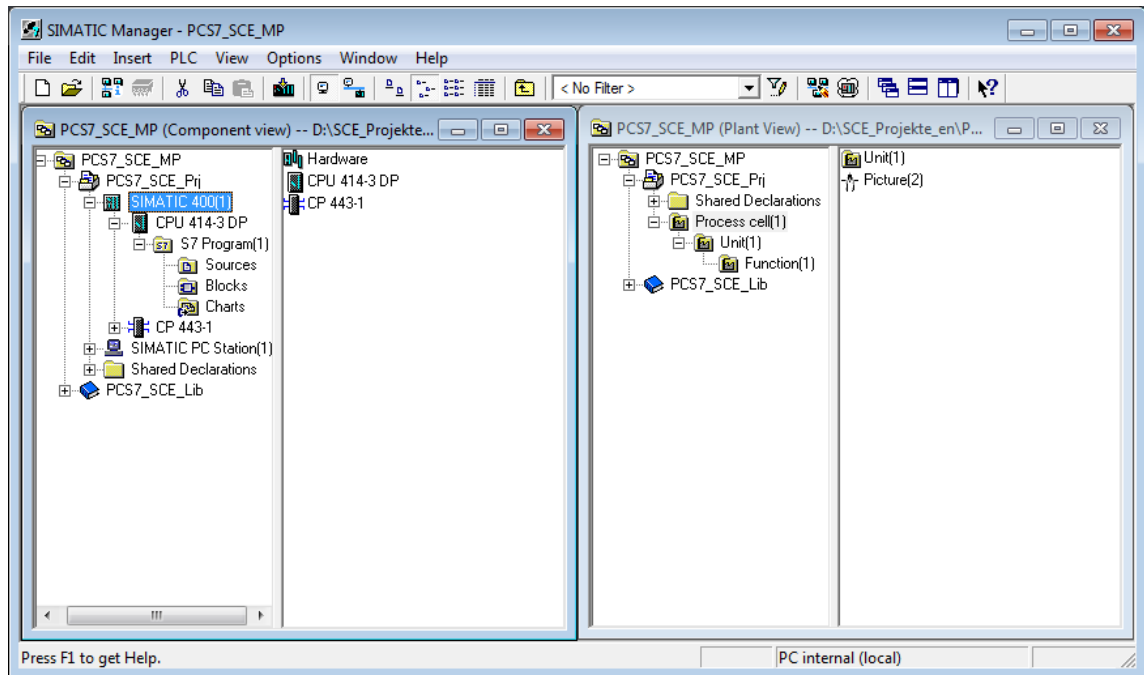
Storage location (path): C:\Program Files\SIEMENS\STEP7\proj Browse

Available directories and files:
_ONL0001
asasrv.ini
projID.pro
SCE_PC_1
SCE_PCS7

Preview >>>

Back Next Finish Cancel Help

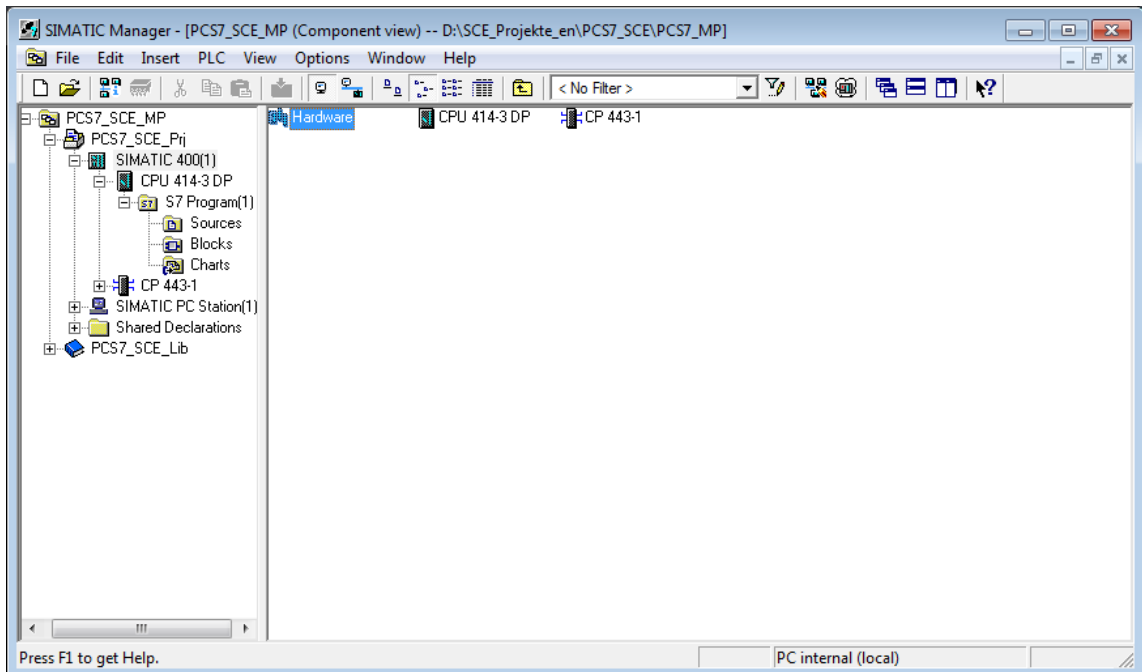
7. After the project is finished, it is opened and displayed in the Component view as well as the Plant view. We can switch between the views in the menu under View.
(→ View)



Note: Additional information about the Component view and Plant view is provided in the chapter below 'Plant Hierarchy'. In this chapter, only the Component view familiar from STEP7 is used.

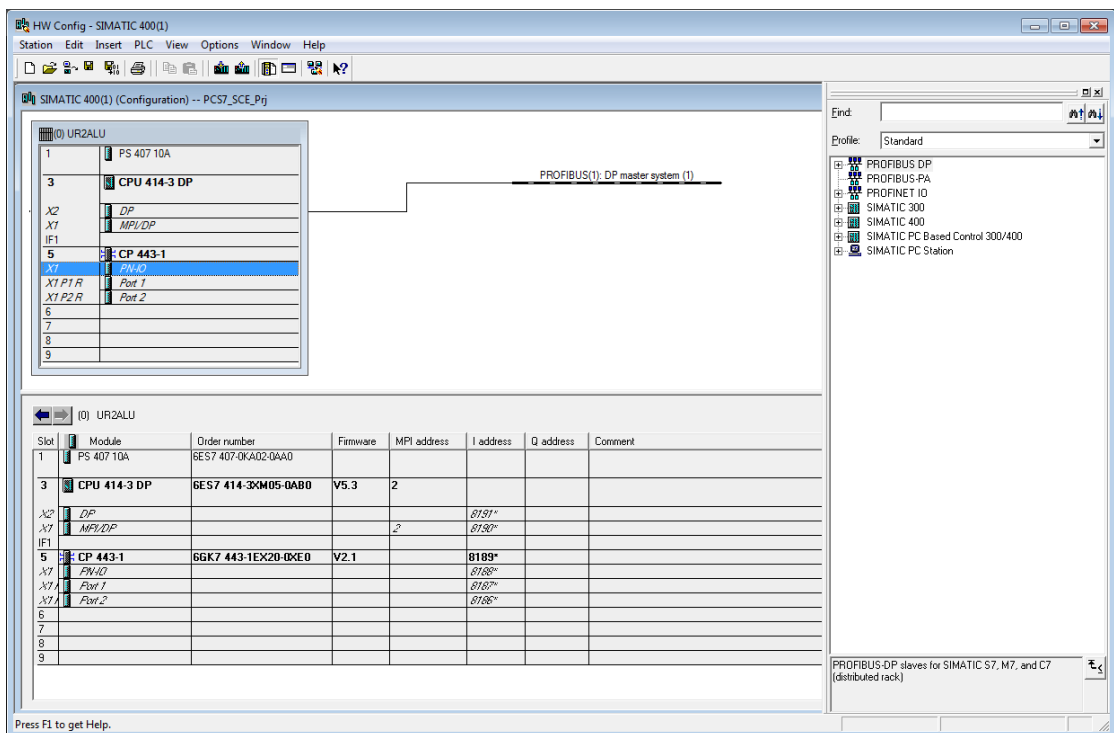
8. Next we select the SIMATIC 400 station in the Component view open the hardware configuration with a double click.

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



9. To make the settings for Ethernet networking, we select the PN-IO interface with a double click in CP 443-1.

(→ PN-IO)



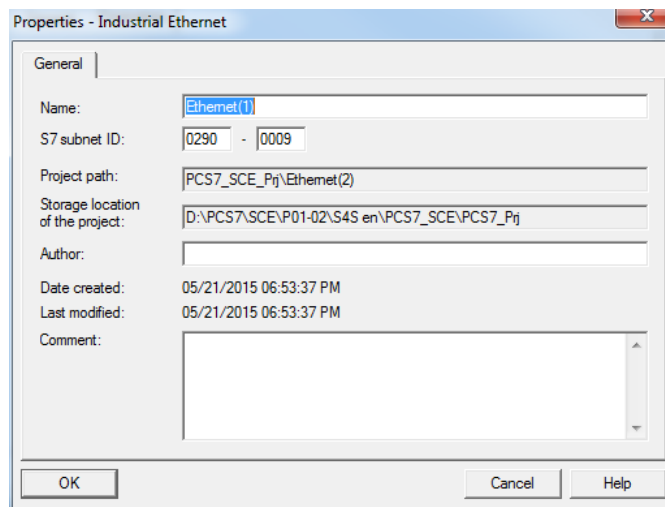
10. Here, we can assign a device name and the properties for the Ethernet interface.
(→ Properties)

The screenshot shows the 'Properties - PN-IO (R0/S5.1)' dialog box with the 'General' tab selected. The 'Short description' is 'PN-IO'. The 'Device name' field contains 'PN-IO'. A checkbox 'Support device replacement without exchangeable medium' is checked. The 'Interface' section shows 'Type: Ethernet', 'Device number: 0', 'Address: 192.168.0.1', and 'Networked: No'. There is a 'Properties...' button next to 'Networked'. A 'Comment' text area is empty. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

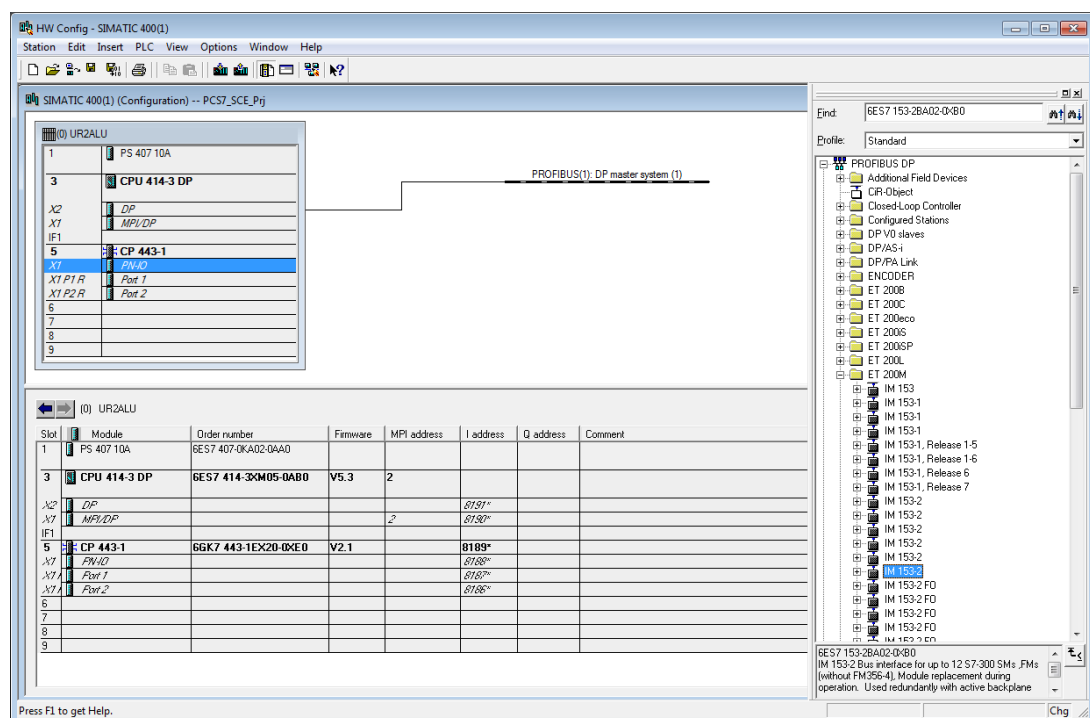
11. In the parameters, we enter an IP address and a subnet mask and set up a new subnet.
(→ Parameters → 192.168.0.1 → 255.255.255.0 → New)

The screenshot shows the 'Properties - Ethernet interface PN-IO (R0/S5.1)' dialog box with the 'Parameters' tab selected. The 'Set MAC address / use ISO protocol' checkbox is unchecked. The 'MAC address' field is empty, with a note: 'If a subnet is selected, the next available addresses are suggested.' The 'IP protocol is being used' checkbox is checked. The 'IP address' field contains '192.168.0.1' and the 'Subnet mask' field contains '255.255.255.0'. The 'Gateway' section has 'Do not use router' selected. The 'Subnet' list shows '— not networked —'. There are 'New...', 'Properties...', and 'Delete' buttons for the subnet list. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

12. Then we accept the subnet and the settings.
(→ OK → OK → OK)



13. Next, we configure the ET200M as a field device on the PROFIBUS. To this end, we first have to set the matching profile, then select the matching interface module in the catalog in the folder PROFIBUS DP/ET200M, and drag it to the CPU's master system.
(→PROFIBUS-DP →ET200M →IM 153-2 HF →PROFIBUS(1): DP master system(1))



Note: To exactly select the correct interface module, the order numbers have to be noted. They are printed on the interface module and in the footer of the hardware catalog after you have selected a component.

If you do not have your own hardware available, it is best to adhere to the representation shown here.

14. In the following selection, assign the PROFIBUS address for the interface module.
(→ 3 → OK)

Properties - PROFIBUS interface IM 153-2

General Parameters

Address: 3

Transmission rate: 1.5 Mbps

Subnet:

- not networked --
- PROFIBUS(1) 1.5 Mbps


New... Properties... Delete



OK Cancel Help

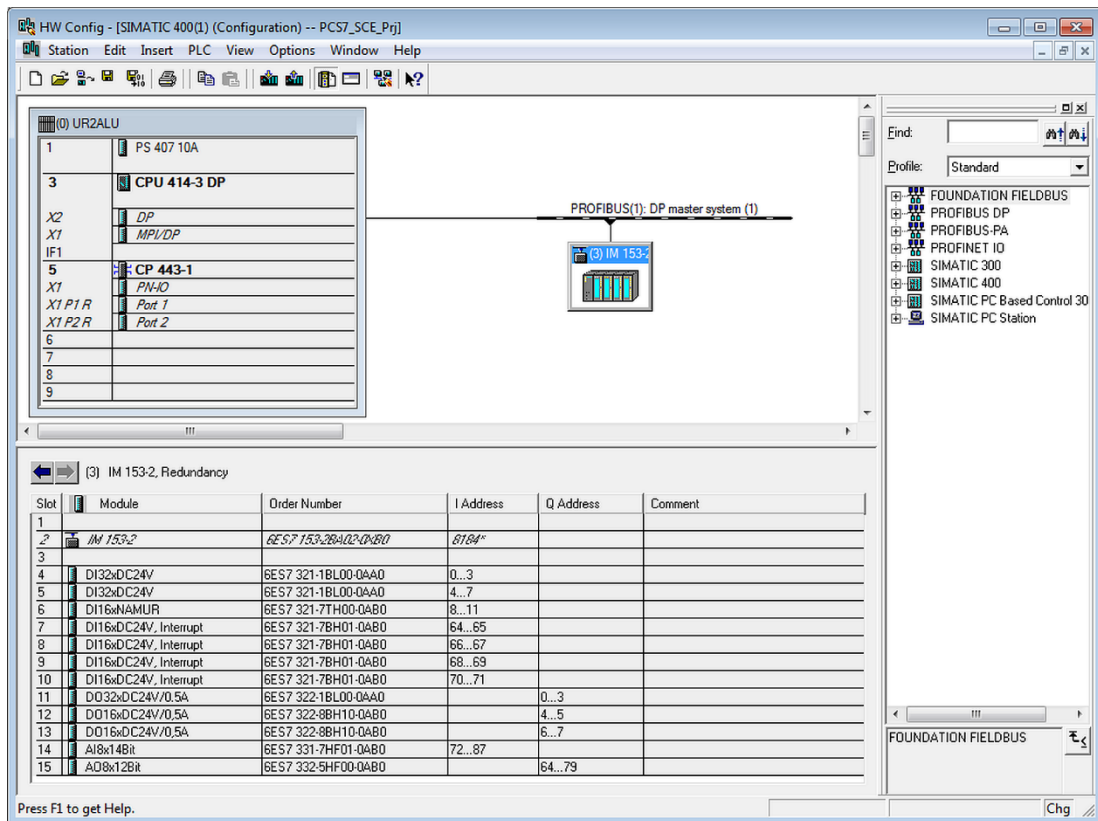


Note: The address set here must also be on the interface module in binary code using a switch block.

15. Now, we enter the IO modules from the folders below the interface module used. This is done by dragging these modules to the respective slot within the ET200M. The IO addresses of the individual modules should be set in their properties as shown here.

When your configuration is complete, accept it by using the button  for saving and compiling.

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



Slot	Module	Order Number	I Address	Q Address	Comment
1	IM 153-2	6ES7 153-2BA02-0AB0	8184		
2					
3					
4	DI32xDC24V	6ES7 321-1BL00-0AA0	0...3		
5	DI32xDC24V	6ES7 321-1BL00-0AA0	4...7		
6	DI16xNAMUR	6ES7 321-7TH01-0AB0	8...11		
7	DI16xDC24V, Interrupt	6ES7 321-7BH01-0AB0	64...65		
8	DI16xDC24V, Interrupt	6ES7 321-7BH01-0AB0	66...67		
9	DI16xDC24V, Interrupt	6ES7 321-7BH01-0AB0	68...69		
10	DI16xDC24V, Interrupt	6ES7 321-7BH01-0AB0	70...71		
11	DO32xDC24V/0.5A	6ES7 322-1BL00-0AA0		0...3	
12	DO16xDC24V/0.5A	6ES7 322-8BH10-0AB0		4...5	
13	DO16xDC24V/0.5A	6ES7 322-8BH10-0AB0		6...7	
14	AI8x14Bit	6ES7 331-7HF01-0AB0	72...87		
15	AO8x12Bit	6ES7 332-5HF00-0AB0		64...79	



Note: To exactly select the correct modules, the order numbers have to be noted. They are printed on the modules and in the footer of the hardware catalog after you have selected a component. If you do not have your own hardware available, it is best to adhere to the representation shown here.

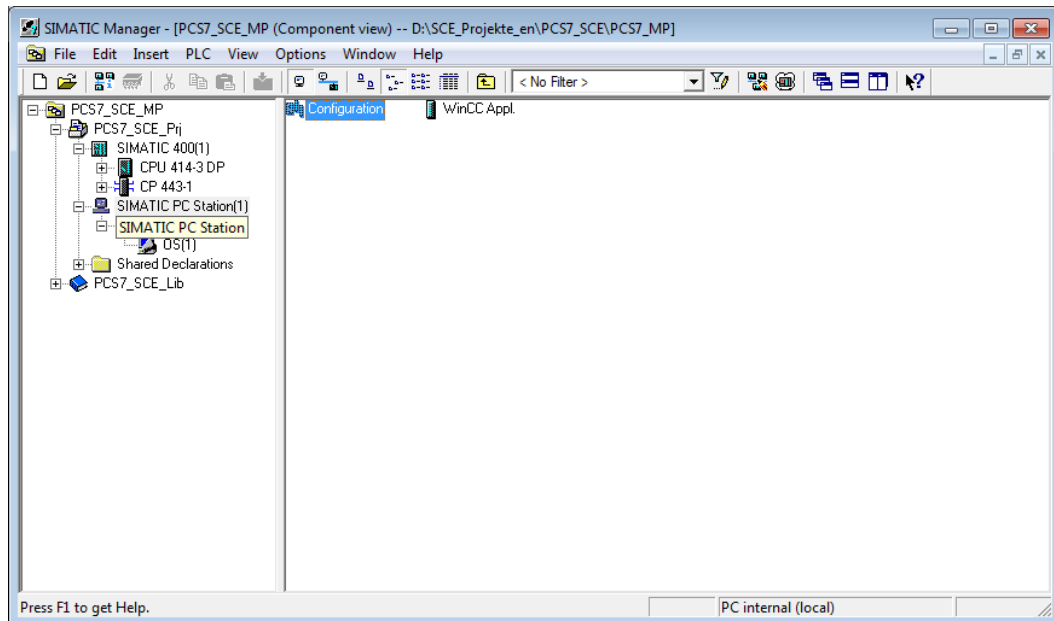
Note: You can facilitate the search for the correct modules by utilizing the search dialog on the very top of the catalog. Simply enter the order number you are looking for; then you can browse the entire catalog up and down.

Note: Slot 3 remains free. It is reserved for the extension module with a multi-tier configuration.

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

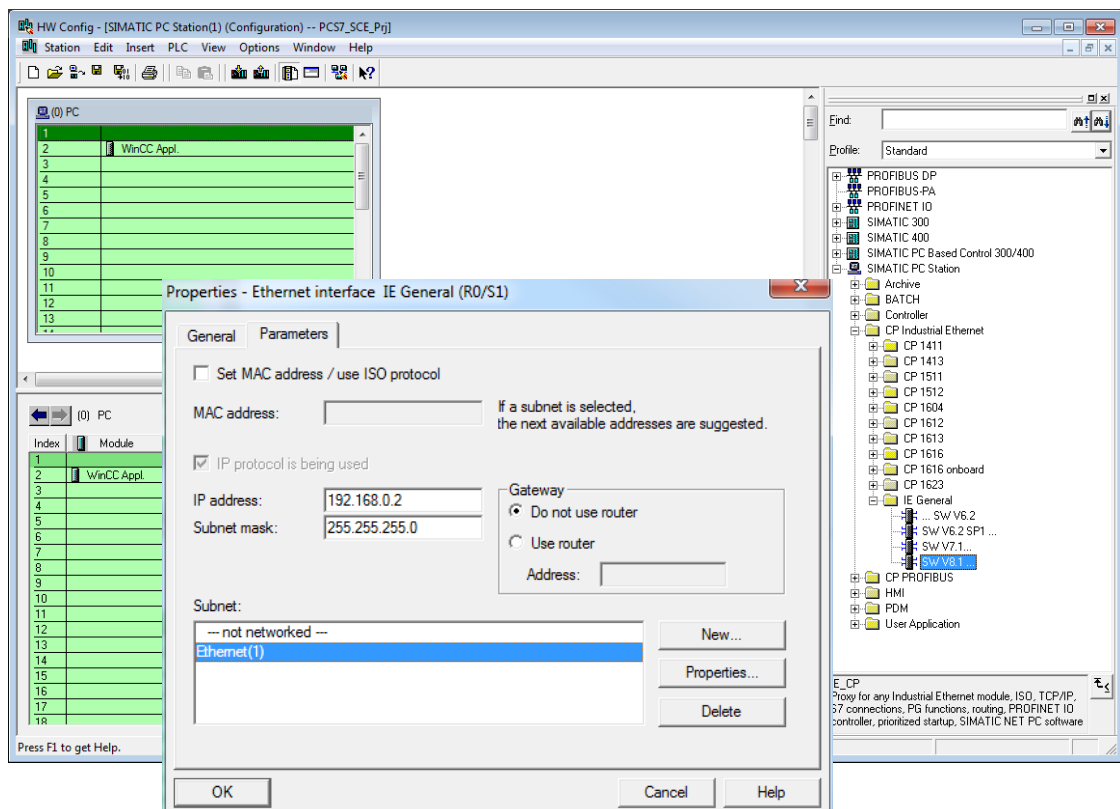
16. Next, in the SIMATIC Manager we select the SIMATIC PC station in the Component view and open the configuration with a double click.




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

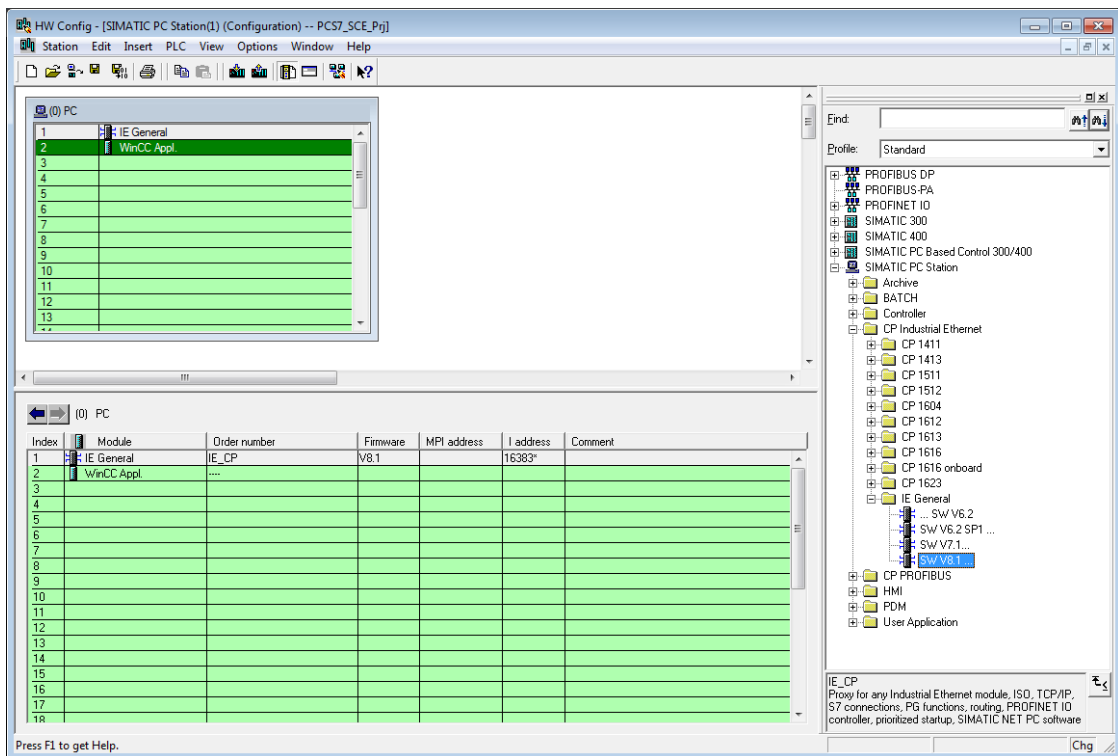


17. Within the PC station, the Ethernet interface must be entered first. To this end, we drag the CP Industrial Ethernet in version V8.1 from IE General to the first free slot in the PC station.

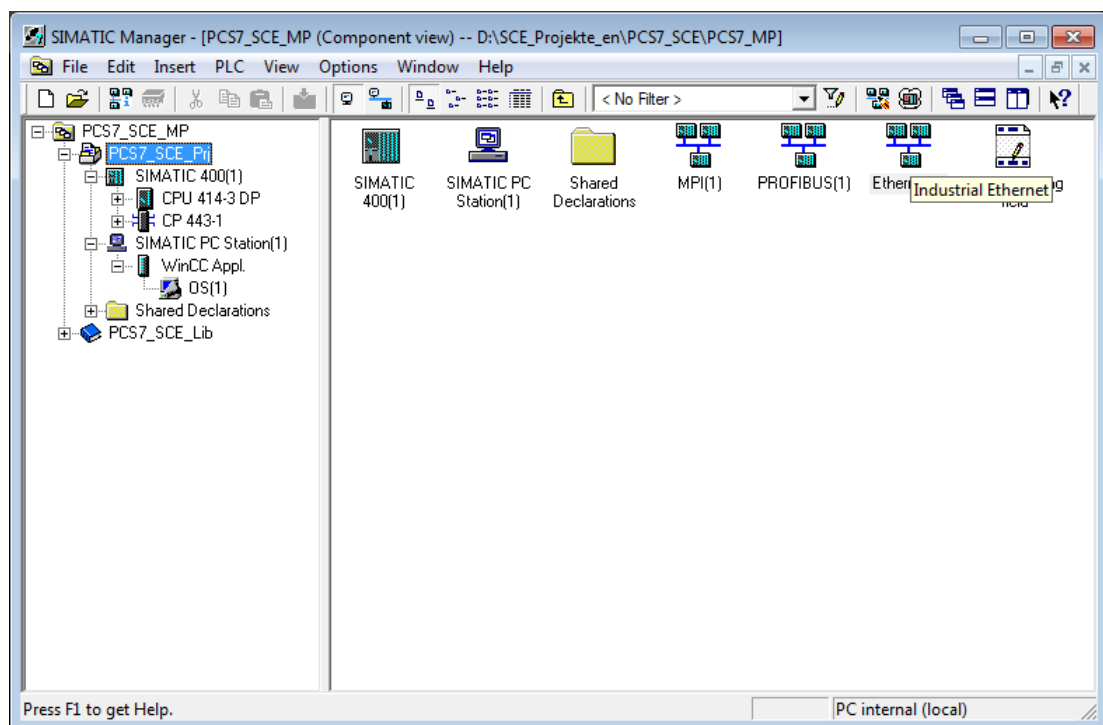
In the window that is displayed next, we connect this interface with the Ethernet network already set up in the S7 station, and enter the IP address and the subnet mask.
(→SIMATIC PC Station →CP Industrial Ethernet →IE General →SW V8.1 →Ethernet(1) → 192.168.0.2 → 255.255.255.0 → OK)





18. An Ethernet interface and the **WinCC** application are now entered in the PC station. We accept this configuration by clicking on  for saving and compiling.
(→  → )

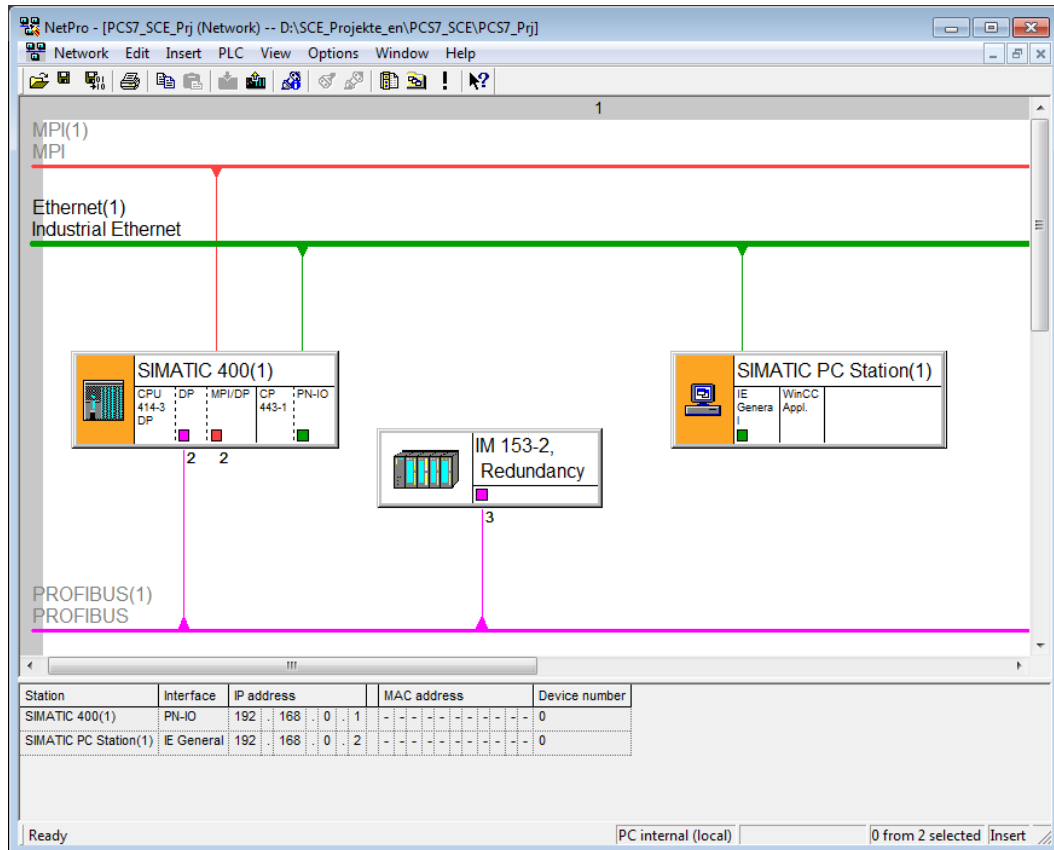


19. To check and compile the network in our project, we now open the Ethernet network with a double click in the Component view in SIMATIC Manager.
(→ Component view → SCE_PCS7_Prj → Ethernet(1))



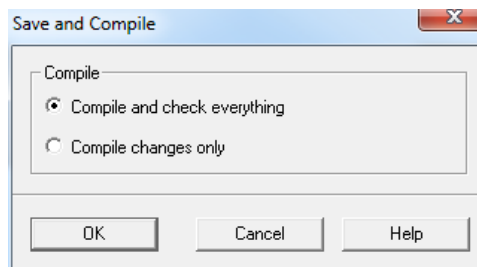
20. The **NetPro** tool provides a nice overview of the components and networks in our project. We see that both stations are connected to each other over Ethernet and the ET200M is interfaced with the SIMATIC S7-400 by means of PROFIBUS. We accept these network settings

by clicking on the button  for saving and compiling.
(→ )



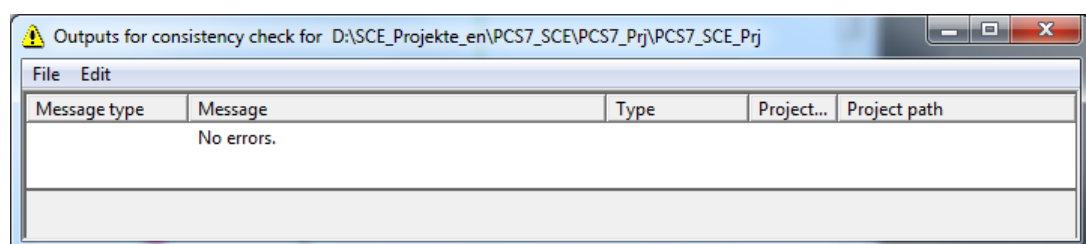
21. In the next window, we select 'Compile and check everything'.

(→ Compile and check everything → OK)



22. The result of the compilation is displayed in a window.

(→ )



EXERCISES

The exercises provide an opportunity to implement what we have learned from theory and the step by step instructions. To this end, the existing multi-project from the step by step instructions will be used and expanded.



Note: The exercises can be carried out without having to first work through the step by step instructions completely and correctly. To get to the required processing level, the project archive provided can be unzipped and opened using the function 'Retrieve'. (→ File → Retrieve...).

The project archive for the step by step instructions is called: PCS7_SCE_0102_R1503_en.zip

The following exercises are recommended especially for users of the PCS 7 trainer package, because an AS RTX Box is to be integrated here. The AS RTX Box is also an automation station and can handle the identical tasks as the configured S7-400.

For that reason, this exercise is not mandatory for execution of the overall project.

EXERCISES

1. With a right mouse click on the project, add the new AS by selecting "Add new object" and then "Preconfigured station ...". In the dialog field that opens, select the "PCS7 BOX" as CPU and then the "AS RTX" with order number 6ES7654-0UE13-0XX0. Follow the dialog without making any other settings.
2. Because the AS RTX Box is a PC based automation station, you should have a second SIMATIC PC station in the project. The stations should now be assigned plausible names; for example, SIMATIC S7-400(1) is named AS1, SIMATIC PC Station(1) is named OS and SIMATIC PC Station(2) is named AS2.
3. Now, network the AS RTX Box (=AS2) with the Ethernet(1) and with a new PROFIBUS master system PROFIBUS(2). To this end, you have to open the configuration of the AS2. So far, your AS has only one interface to PROFIBUS, "CP5611-CP5621". For this reason, add an IE General. Assign parameters to the Ethernet interface exactly as described in the step by step instructions. To parameterize the PROFIBUS interface, you have to open the Properties and add a new PROFIBUS network.
4. For the new AS to actually be able to handle the tasks of the AS1, you now need the identical ET200M. You have two options for adding the ET200M including the IO cards. The first option corresponds to the configuration as provided in the step by step instructions. The second option is copying the generated ET200M and inserting it with a right mouse click on the PROFIBUS(2) line.