24V production line

Hardware configuration

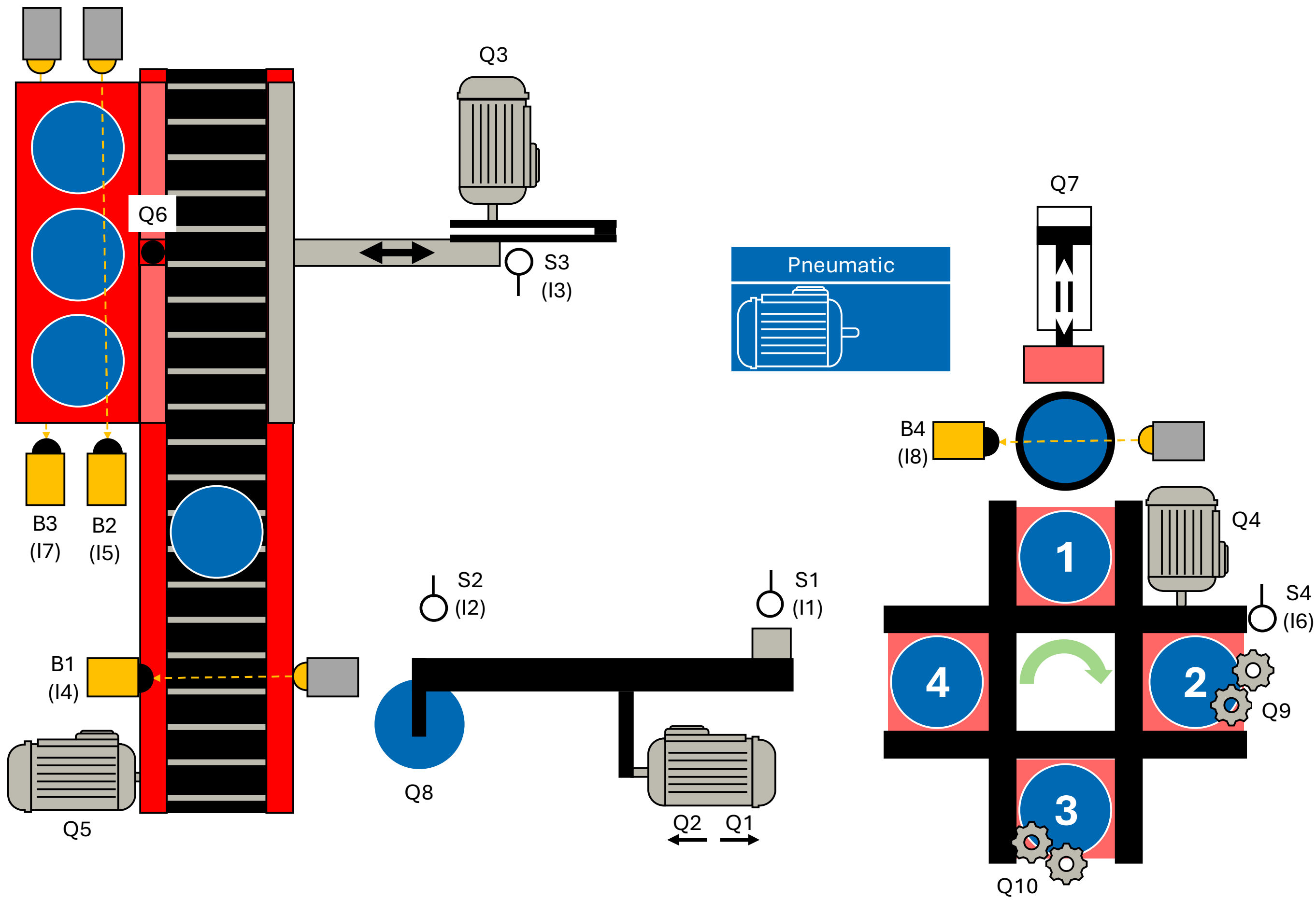
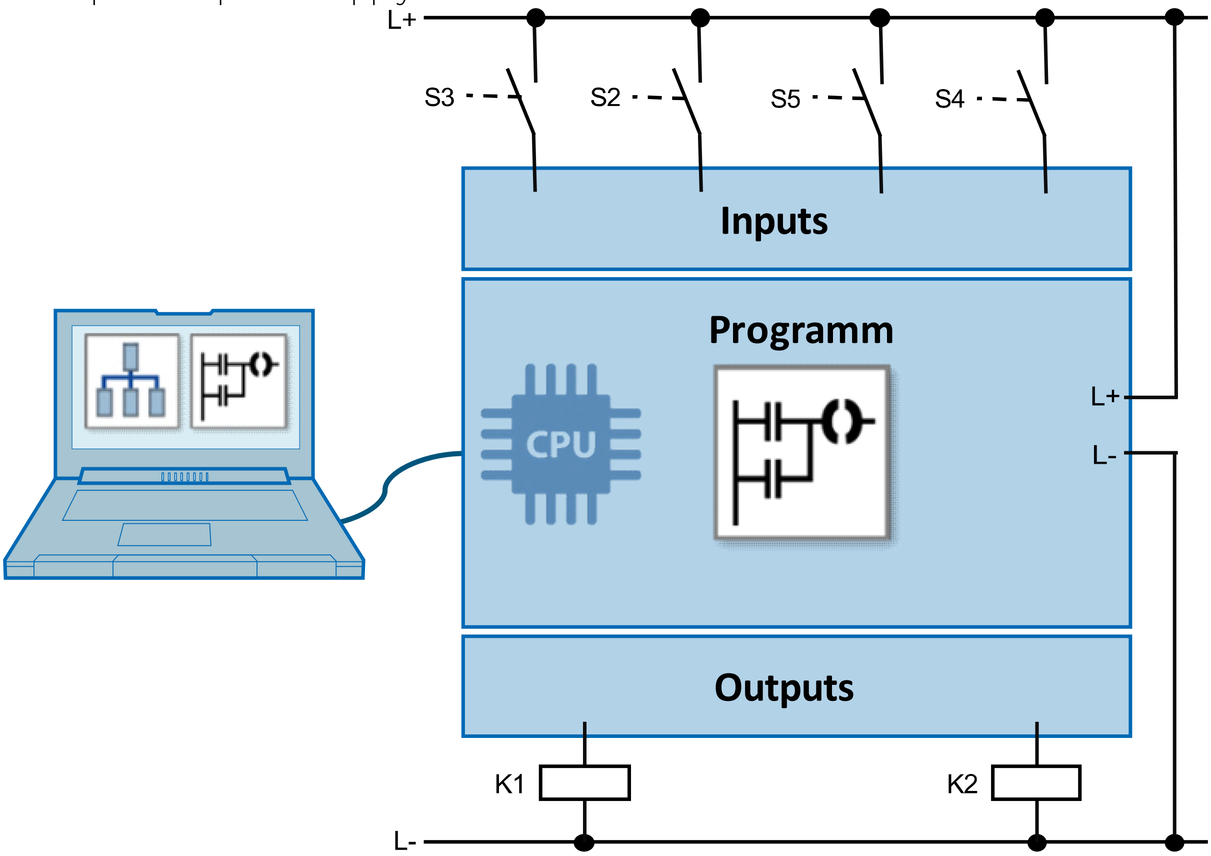


Table of contents

2 Hardware configuration

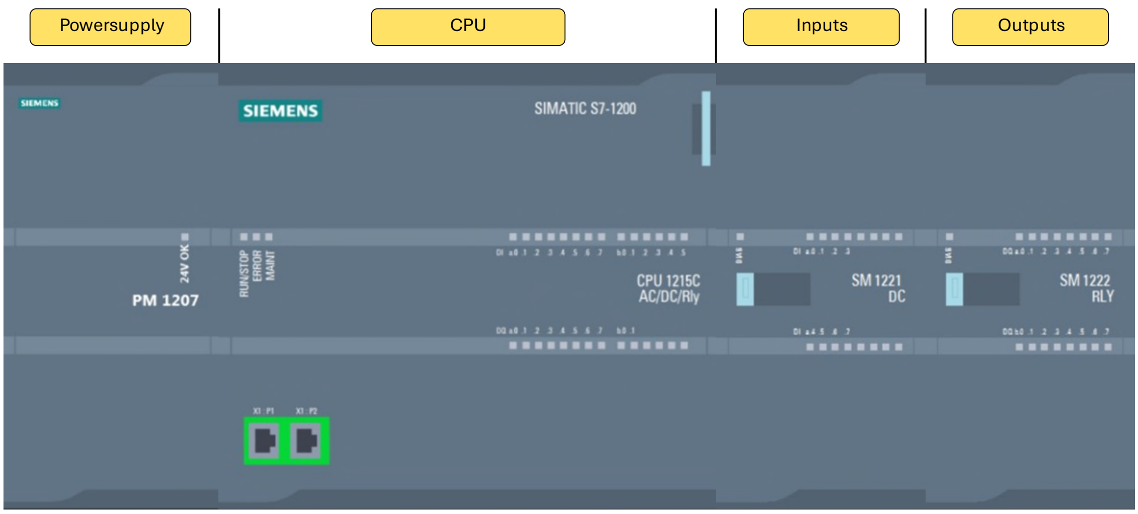
2.1 Components of a PLC

The model is controlled by a programmable logic controller. Such an automation system essentially consists of the central module, the input and output modules and a possible power supply module.



Picture 1 PLC - Mode of operation

The following figure shows the structure of an automation system using the Siemens SIMATIC S7-1200 as an example.



Picture 2 Structure of a Siemens S7-1200 controller

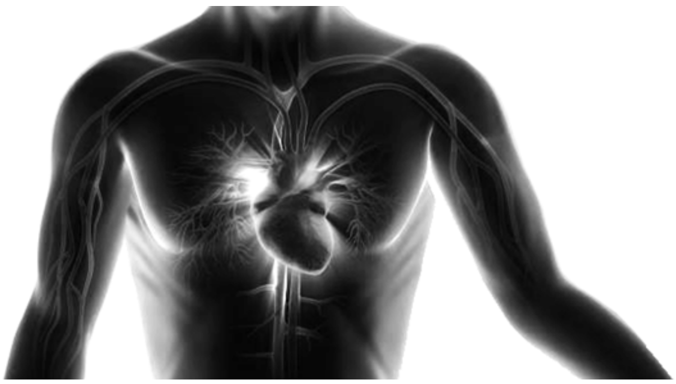
The components of an S7-1200 are explained below in analogy to the human body.

2.1.1 Power supply

The system power supply (PS) supplies the automation system with an internal voltage. An additional load power supply (PM) is required to supply the signaling devices, actuators and indicator lights.

Picture 3 S7-1200 Load power supply

In humans, the power supply corresponds to the cardiovascular system, which supplies all other organs with energy.

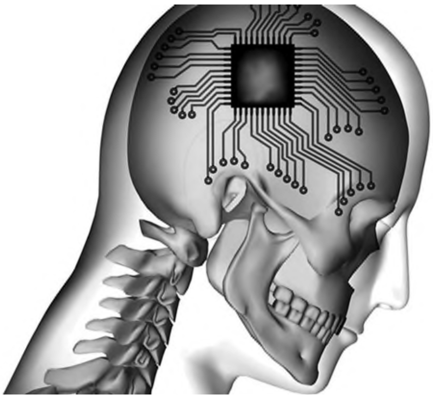


Picture 4 Cardiovascular system

2.1.2 Central assembly

The control unit of the central processing unit (CPU) processes the program stored in the program memory.
The status of the inputs is queried during program processing.
Depending on the signal status of the inputs and the program stored in the program memory, the outputs are then activated by the control unit.

Picture 5 S7-1200 CPU



Picture 6 Brain

In comparison to humans, this would be the brain, which processes all control sequences.

Signal modules form the interface between the process and the automation system. Digital and analog input and output modules are available.

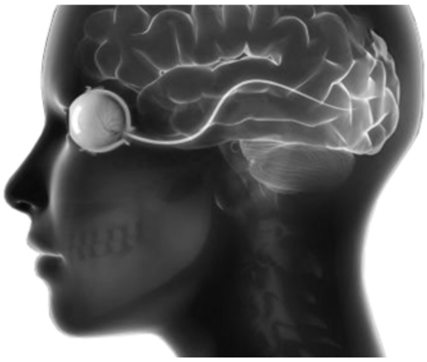
2.1.3 Input modules

The signal transmitters (sensors) are connected to the input modules (DI = Digital Input or DE Digital Inputs / AI = Analog Input or AE = Analog Inputs). These are, for example

- Control buttons and switches
- Contact and position feedback
- Counting pulses

The CPU uses these signals to record the current system status.

Picture 7 S7-1200 DI module



Picture 8 human senses

The input modules record the signals from the sensors and forward them to the CPU in a similar way to how the human eye forwards signals to the brain.

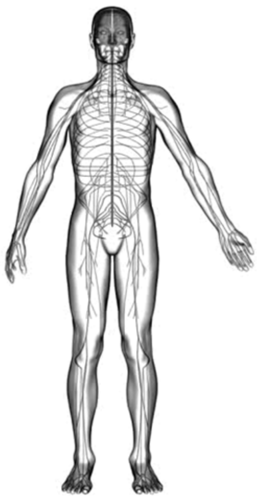
2.1.4 Output assemblies

Actuators and signal transmitters (actuators) are connected to the output modules (DO = Digital Output or DA = Digital Outputs / AO = Analog Output or AA = Analog Outputs).
These are for example:

- Indicator lights
- Contactor and valve controls
- Drive commands

Depending on the information processed, the CPU sends signals to the individual outputs, which then control the actuators and actuators and trigger reactions.

Picture 9 S7-1200 DO module

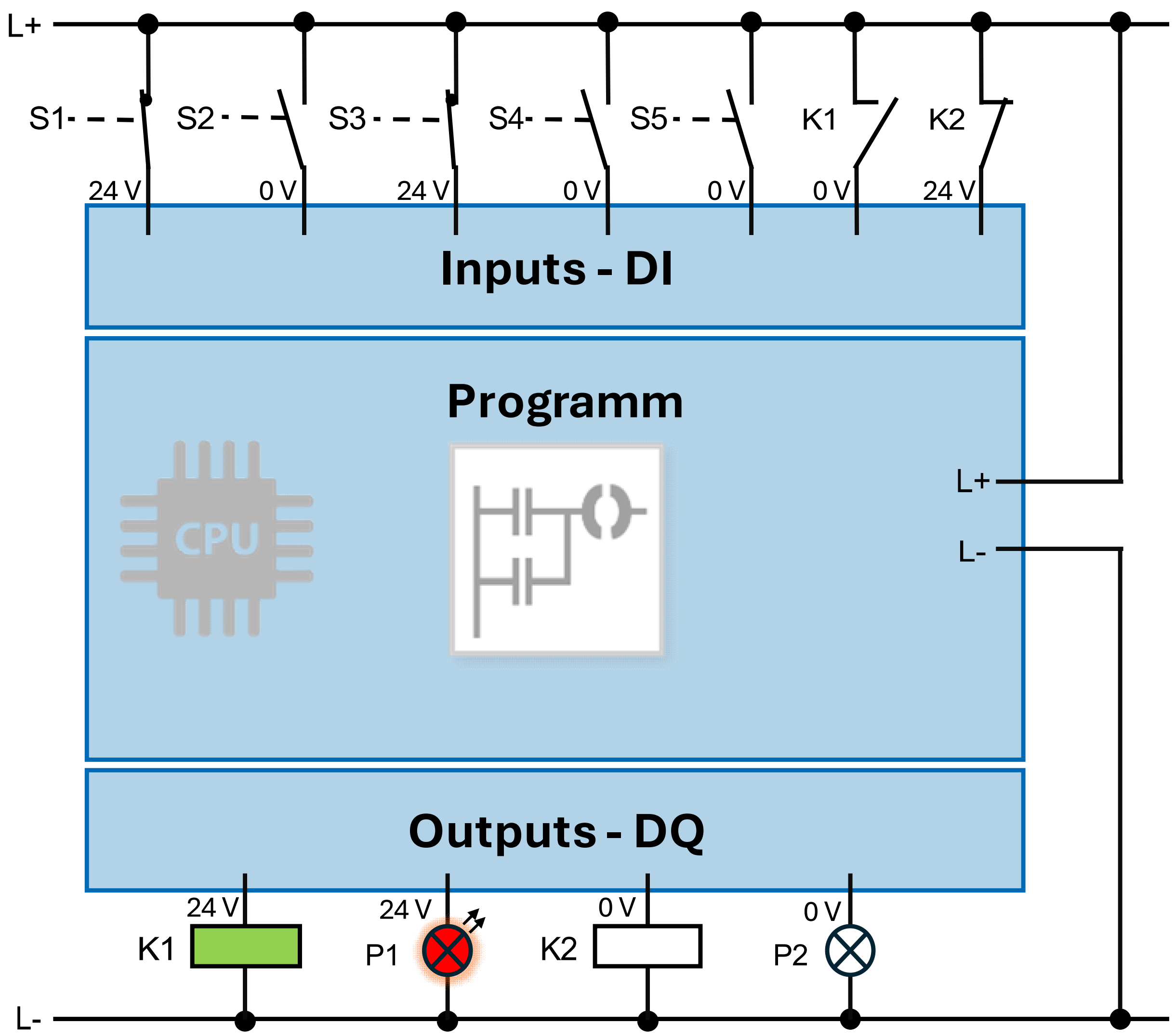


Picture 10 human limbs

In analogy to humans, it is the limbs that react to the commands of the brain.

2.1.5 Binary signals

In programmable logic controllers (PLCs), a lot of information is processed and controlled using binary, i.e. two-value, signals.
These are read into the PLC by digital input modules and transmitted via digital output modules.



Picture 11 Connection of binary signals

This signal status information is stored in one bit in the PLC.
The bit is the smallest unit of information technology.

2.1.6 Signal status information

Binary input signals

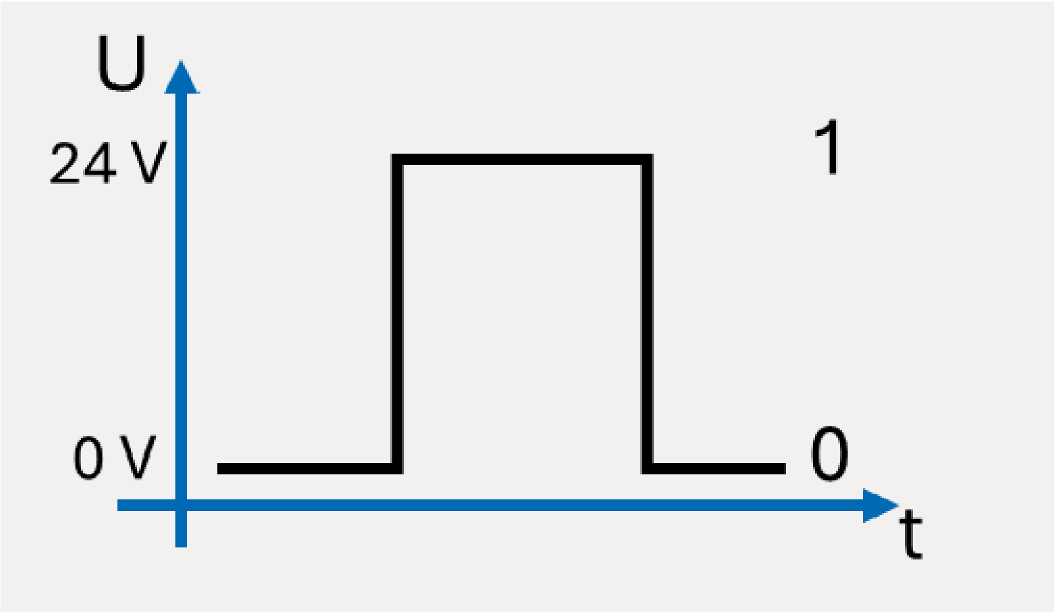
The status of a binary input signal is detected via the voltage applied.
A distinction can be made between two signal states.

- Voltage present = signal status "1" or "TRUE"
- Voltage is not present = signal status "0" or "FALSE"

Binary output signals

The same applies to the binary output signals.

- the output is controlled by the PLC:
  Signal status "1" or "TRUE" = voltage present
- the output is not controlled by the PLC:
  Voltage is not present = signal status "0" or "FALSE"



Picture 12 Binary signal

2.2 Hardware project planning

In the hardware configuration, the modules are configured in the same way as they exist in the real system.
The procedure will differ depending on the target system used (Siemens S7 300 / S7 1500, Beckhoff, etc...). However, the following basic steps must always be followed:

- Structure of the hardware components used in the programming software (e.g. TIA Portal or TwinCAT)
- Parameterization of the modules

  CPU
    - Communication addresses (e.g. IP address, other bus addresses)
    - Assembly labeling (name)

  Signal modules
    - Input / output addresses
    - Assembly labeling (name)

- Configuration can be translated without errors

The procedure is then described in detail using the example of an S7 1200 CPU in the TIA Portal, but PLC systems from other manufacturers can also be used (Rockwell, Schneider Electric, Mitsubishi Electric, ABB, Omron, Bosch-Rexroth, Beckhoff, ...).

2.2.1 TIA Portal

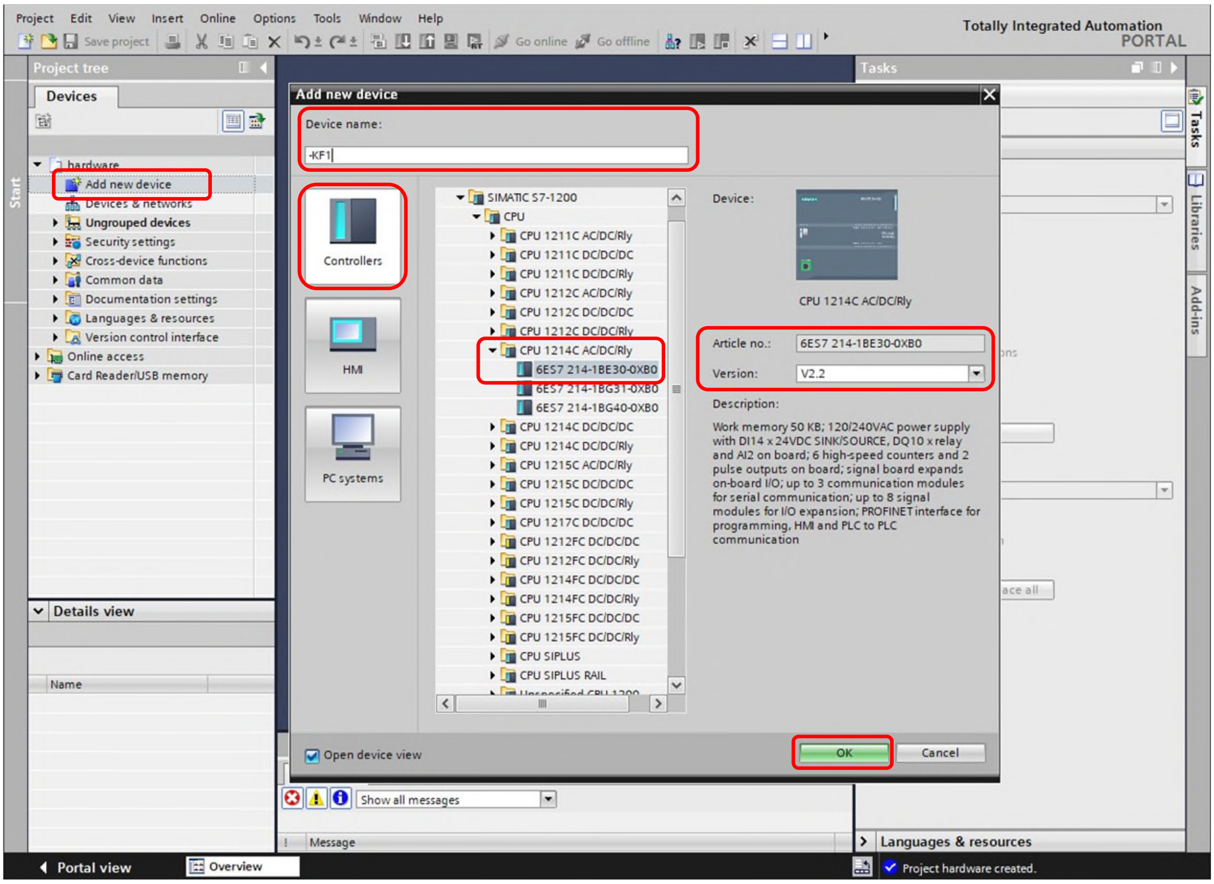
The following describes in detail how hardware configuration of an S7 1200 PLC can be carried out in the TIA Portal.

An empty TIA Portal project is used as the initial state, into which an S7 1200 controller is inserted as a new device.

You are supported by a wizard when selecting the devices.
After pressing the "Add new device" button, three device groups are available for selection:

- Controller
- HMI
- PC systems

After selecting a device group (here the "Controller" group), the device to be inserted can be selected from a tree structure using the article number. When inserting, ensure that the correct firmware version is selected.
It is advisable to assign a meaningful name to the device (e.g. the equipment identification number).



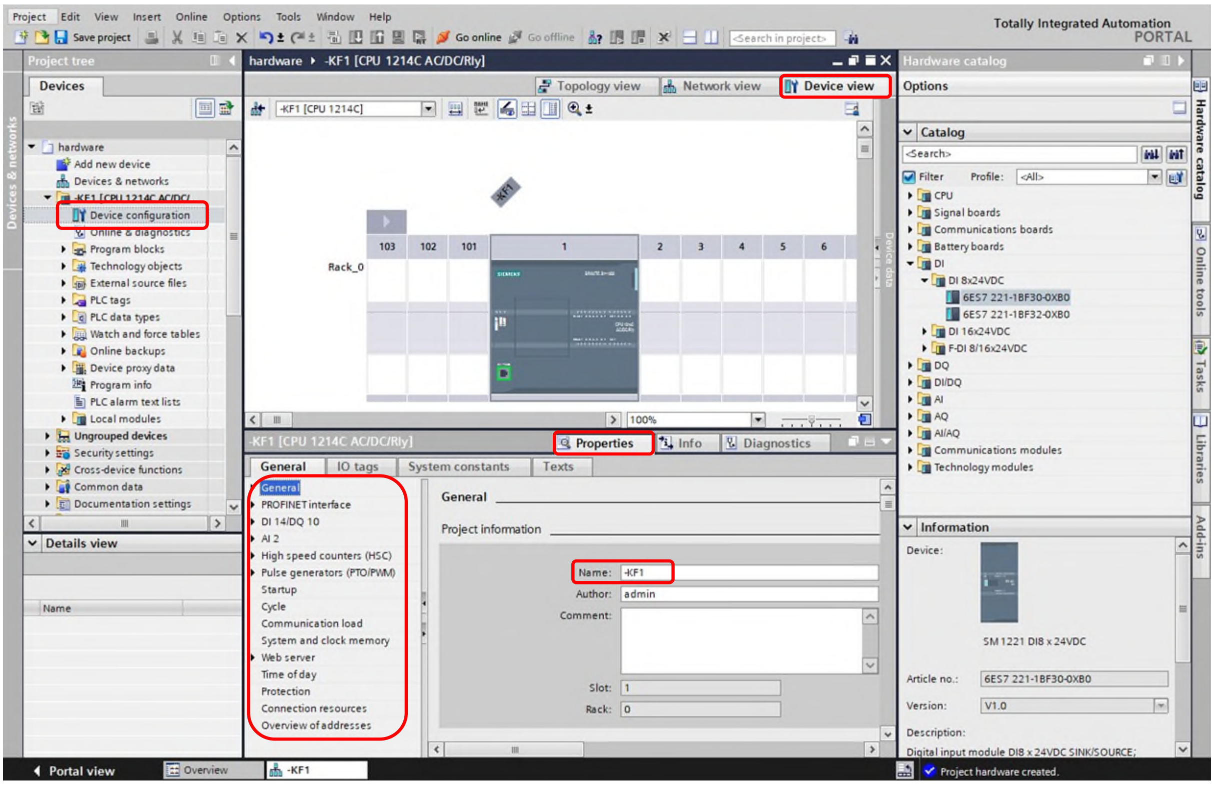
Picture 13 "Add new device" dialog

After adding the CPU, the device view opens.
This can also be opened again at any time in the project navigation, below the configured CPU, by double-clicking on "Device configuration".

The device view is used to configure and parameterize devices.

When configuring the device hardware, you determine which modules are used in your system. This includes the selection and arrangement of subracks and the modules within the subracks. You select the individual modules from the hardware catalog in the task cards. When parameterizing, you define properties for each (parameterizable) module (e.g. address).

If a hardware component is selected in the graphical area of the device view, the parameters of this module can be adjusted in the inspector window under "Properties". These are organized in a tree structure.



Picture 14 Device view

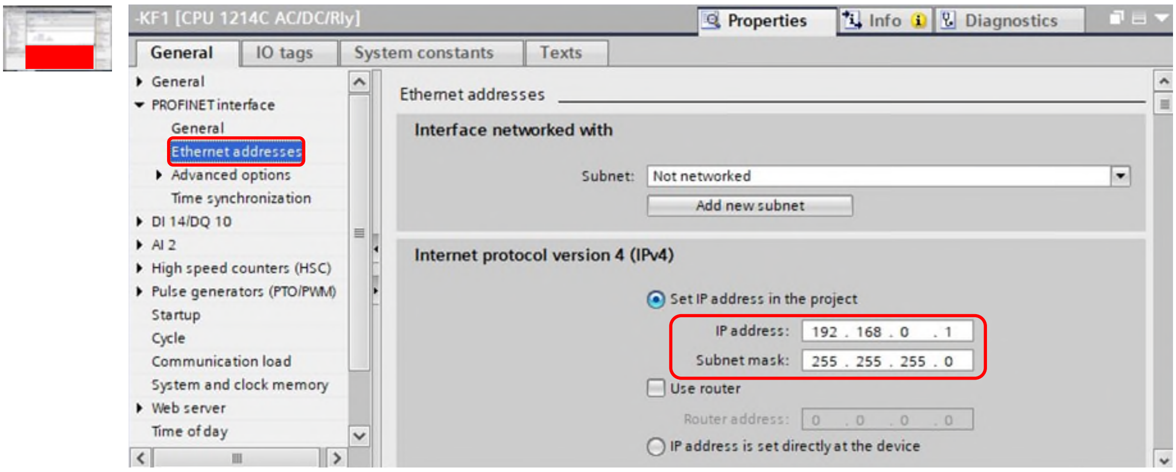
If the device has an equipment identification, this can be entered for the respective component under "General→ Project information→ Name".

GROLLMUS fischertechnik

Ethernet address and subnet mask

The Ethernet address is uniquely assigned and is required for communication via Ethernet or PROFINET.

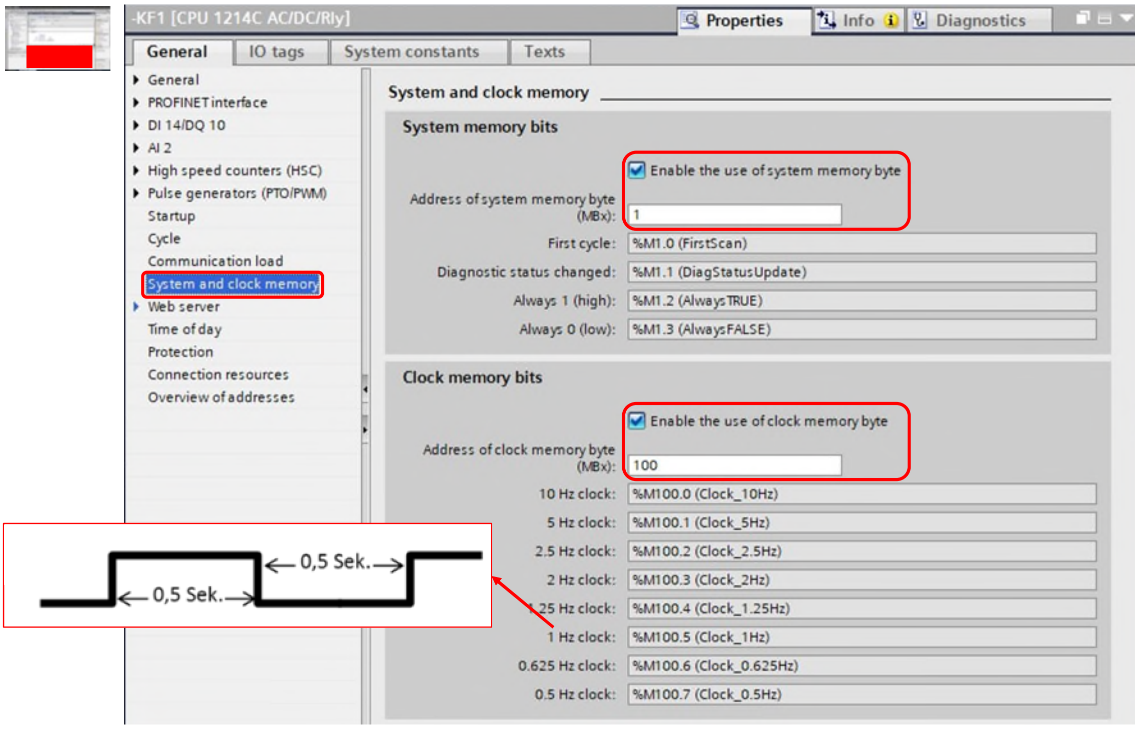
Networking to other stations (e.g. ET200SP IO device) takes place via the "Subnet" setting. By default, "not networked" or "PN/IE_1" can be selected here.



Picture 15 Ethernet address

System and clock flags

In the PLC properties under "System and clock flags", you can define and activate flag bytes for system flag bits and one for clock flag bits.



Picture 16 System and clock flags

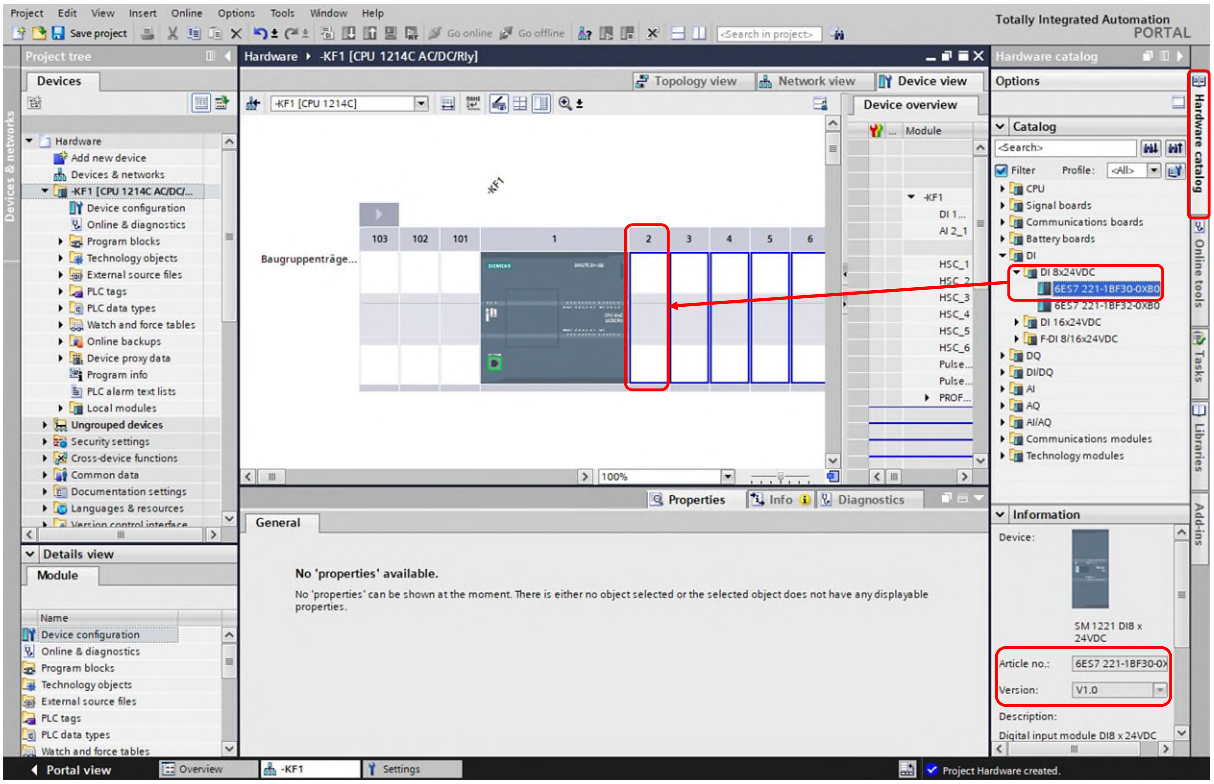
The individual bits in a clock memory byte have different fixed frequencies. The address of the marker byte is defined when the CPU is parameterized. Clock flags can be used for calculations or flashing displays, for example.

Adding and parameterizing peripheral modules

You can add further modules to the rack from the hardware catalog.
The following options are available:

- via drag & drop from the hardware catalog to a free valid slot
- Double-click on the slot selected in the rack in the hardware catalog
- via "Copy" and "Paste"

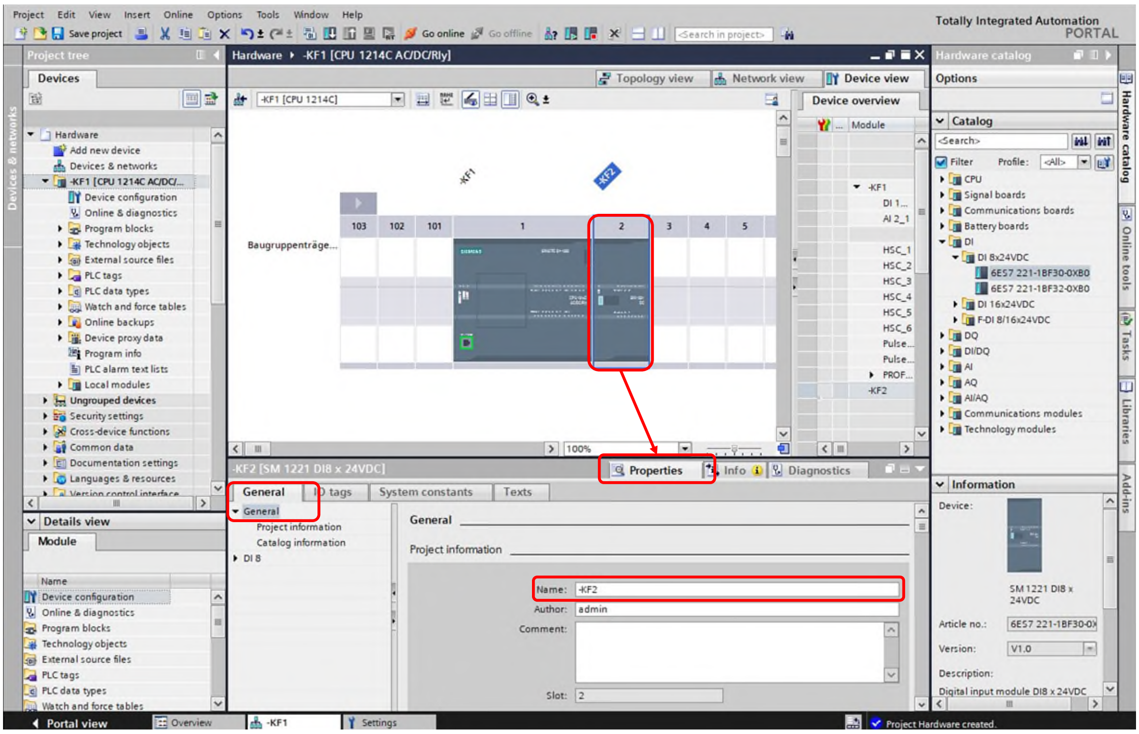
Possible slots are outlined in blue after selecting the module from the hardware catalog.



Picture 17 Add signal module

Siemens assemblies are identified by the article number. This is printed on each module. Before inserting the module, ensure that the correct firmware version is selected in the "Information" palette.

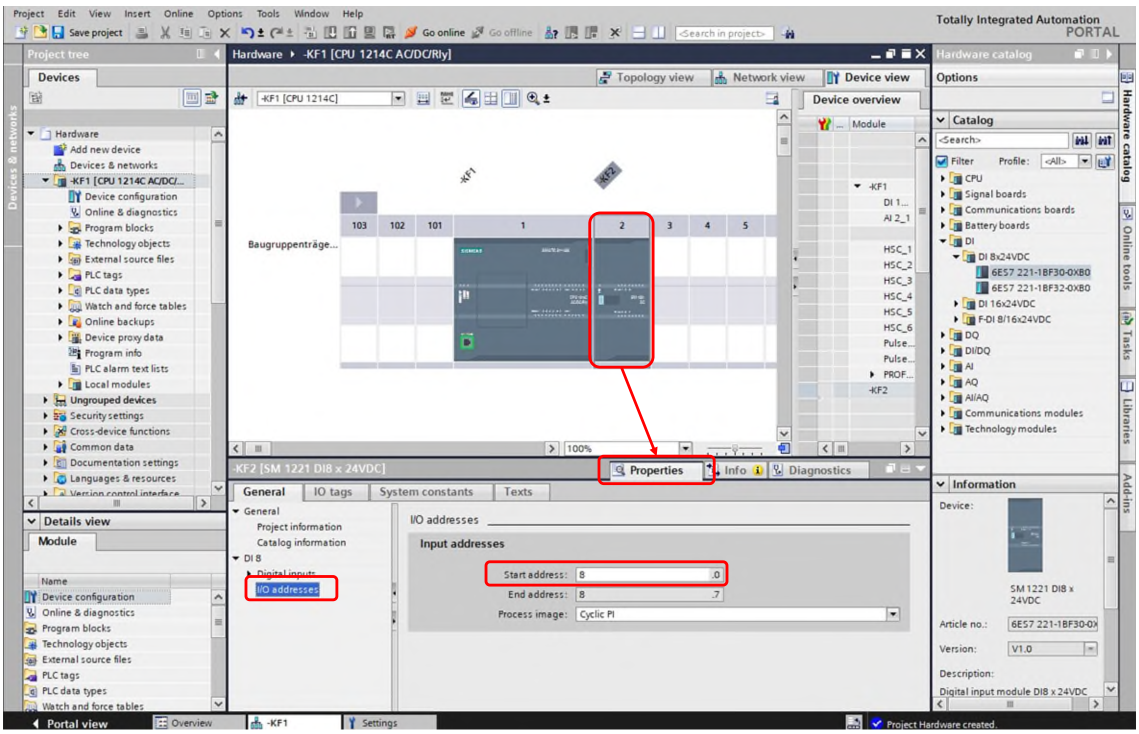
I/O addresses and other parameters are preset when plugging in.
If the module is selected, these parameters can be adjusted in the inspector window under "Properties".



Picture 18 Signal module properties→ General

Under "General", for example, the assembly name can be replaced with a meaningful name.

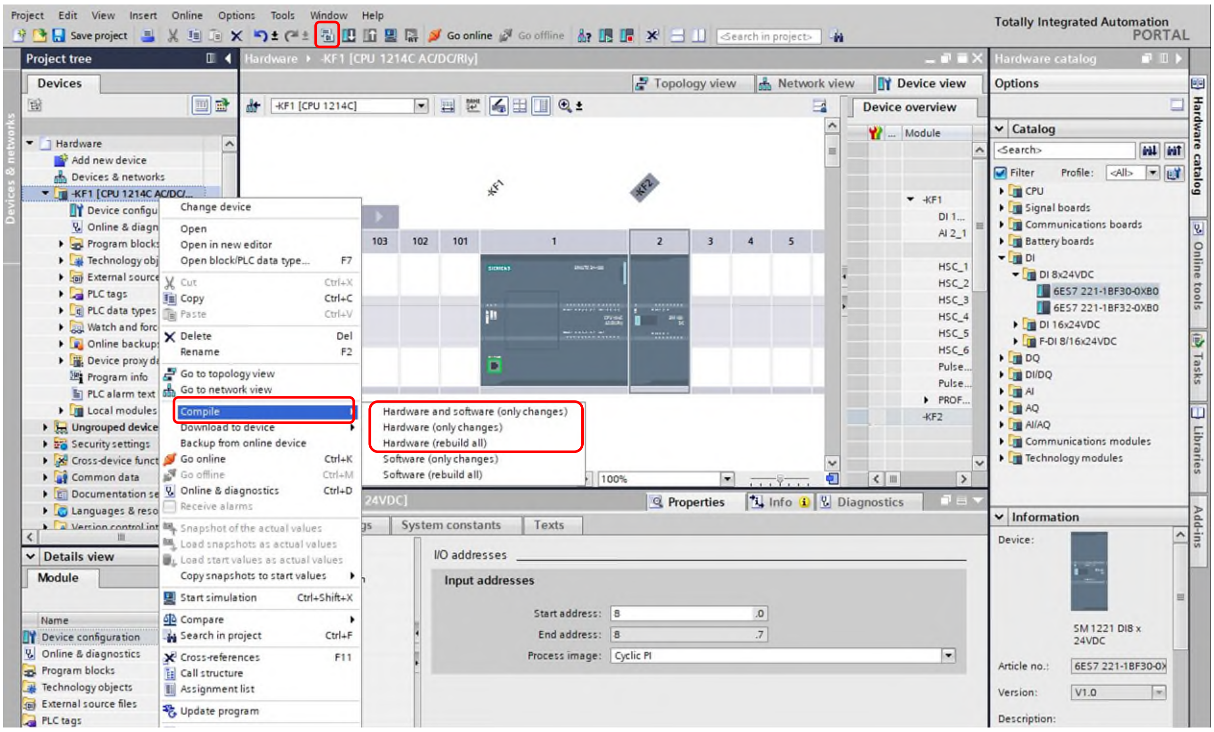
The I/O addresses can also be adjusted in the tree structure:



Picture 19 Signal module properties→ I/O addresses

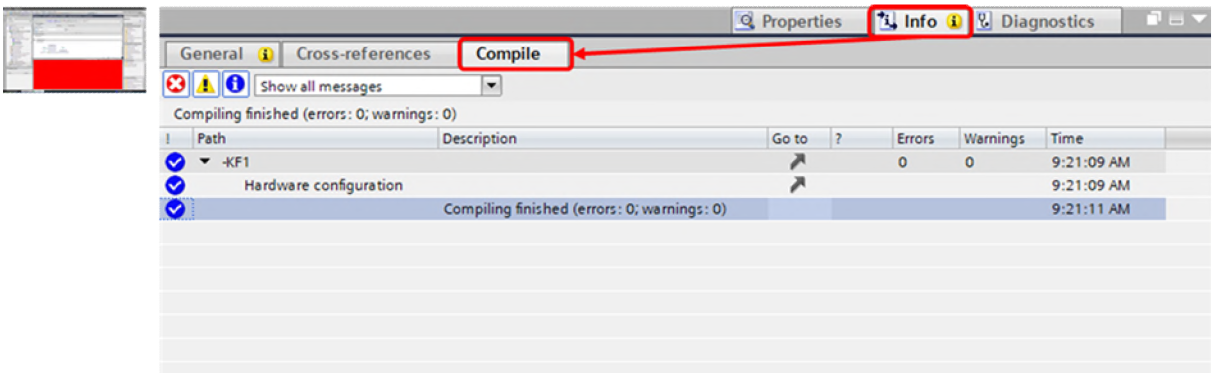
Translate project planning data - Hardware

Before the configuration data can be loaded into the PLC, the configuration must have been translated without errors. The project planning is checked for consistency during translation.
You can trigger the translation explicitly, e.g. via the context menu of the right mouse button of the device in the project navigation or via the button in the function bar of the program editor.



Picture 20 Translate

The result of the translation, including any errors or warnings that may have occurred, is displayed in the "Translate" tab in the inspector window.



Picture 21 Translate tab in the inspector window

The "Go to" column takes you to the error location. If the translation contains errors, the device cannot be loaded. If there are warnings, loading is generally possible.

It is nevertheless advisable to remove all warnings.

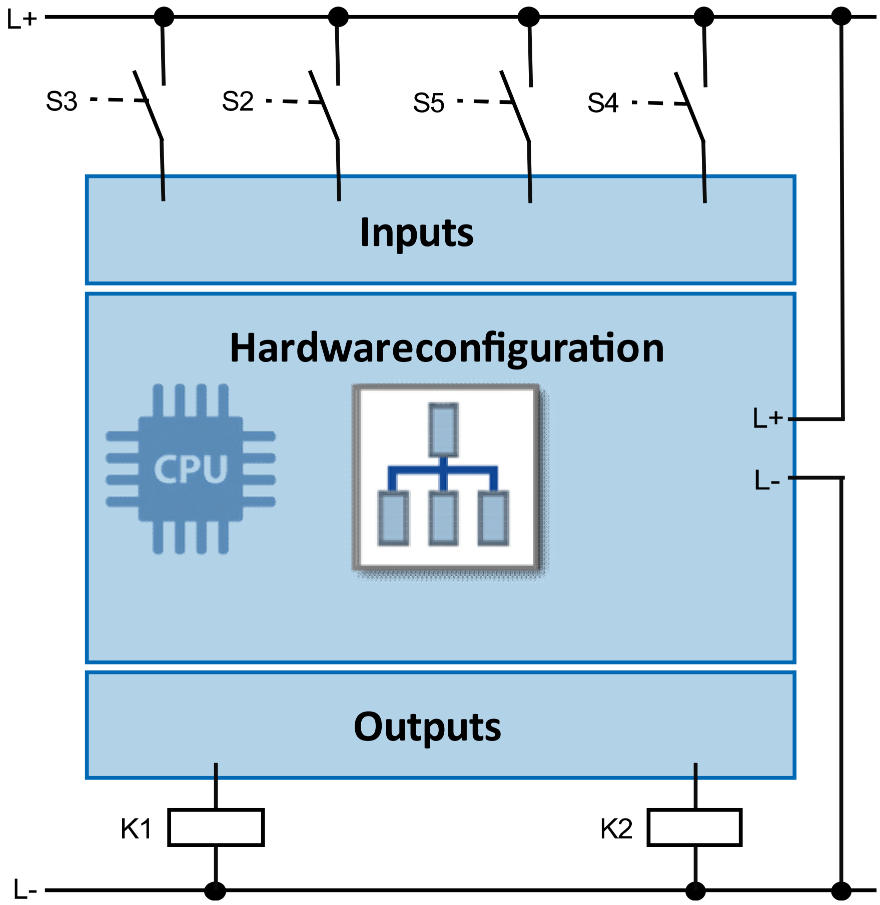
2.2.2 Exercise: Hardware configuration

Target:

I can carry out the project planning of the PLC hardware independently.

Task:

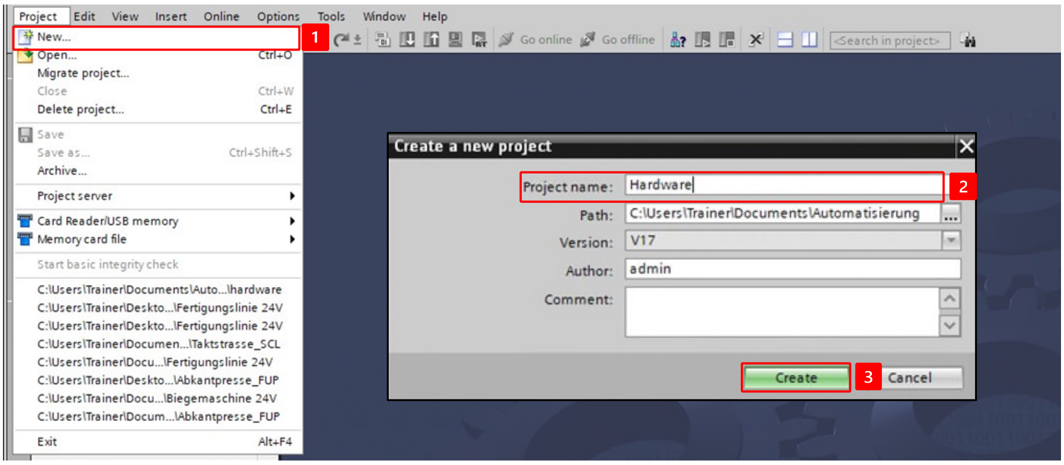
Configure the hardware according to your target system and translate the configuration data.



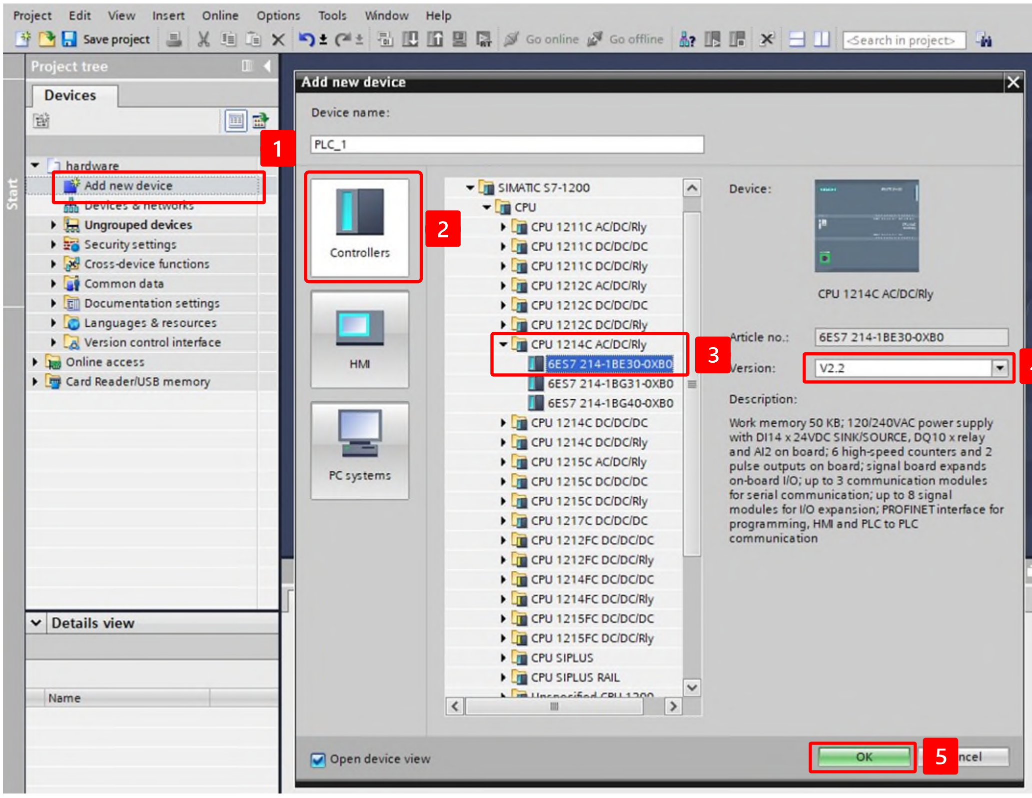
Procedure:

The procedure is shown below using the configuration of an S7 1214C AC/DC/RLY as an example.

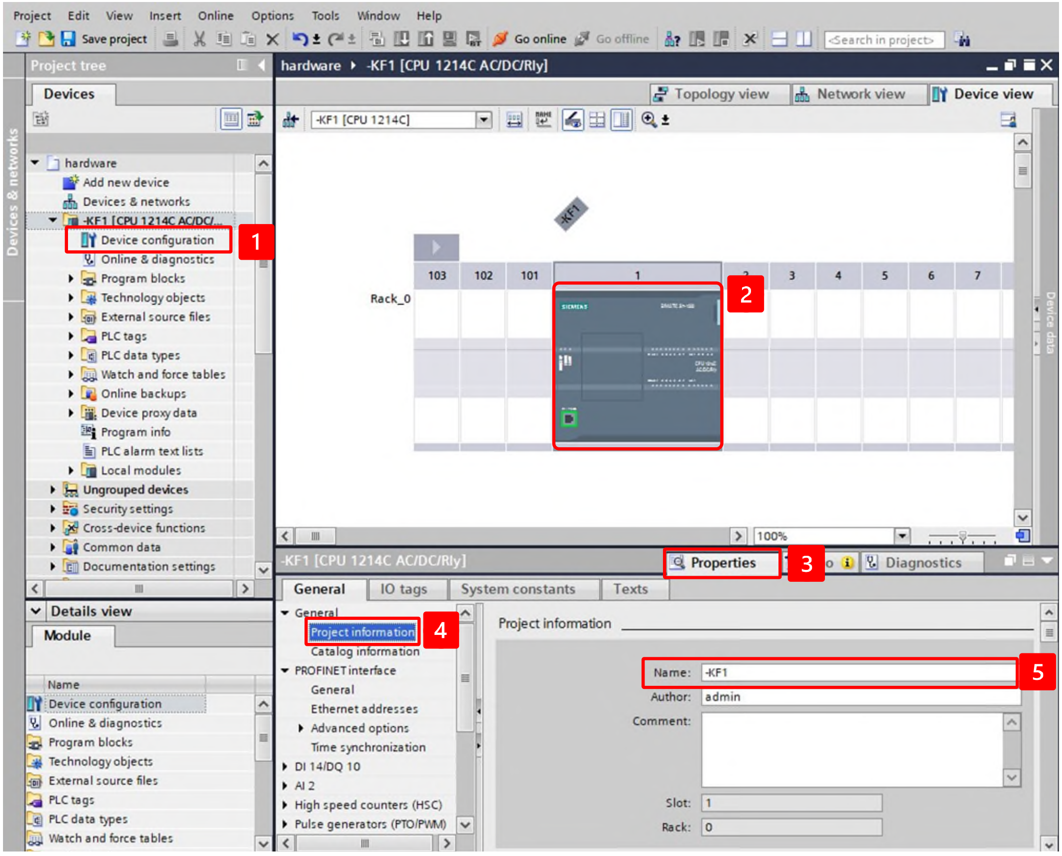
1. Create a new TIA Portal project and assign a meaningful name:



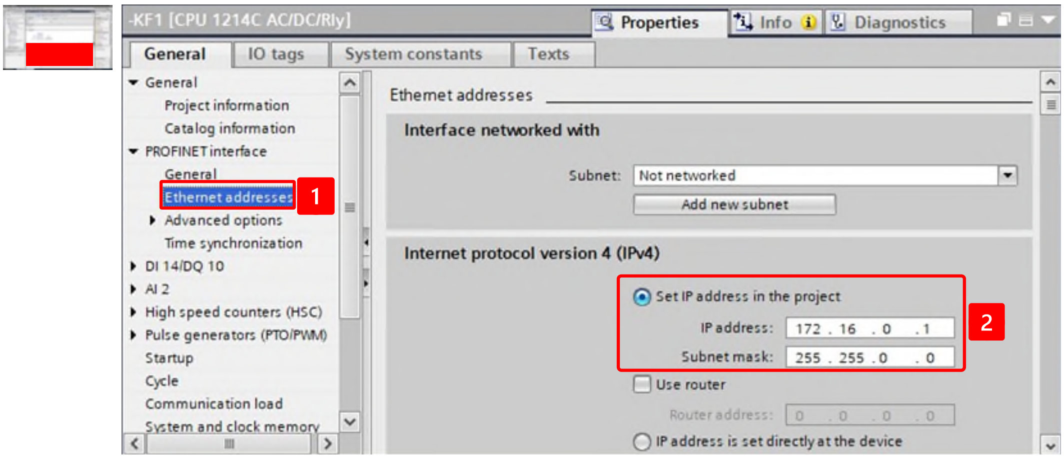
2. Add the CPU.
   Make sure you have the correct order number and firmware.



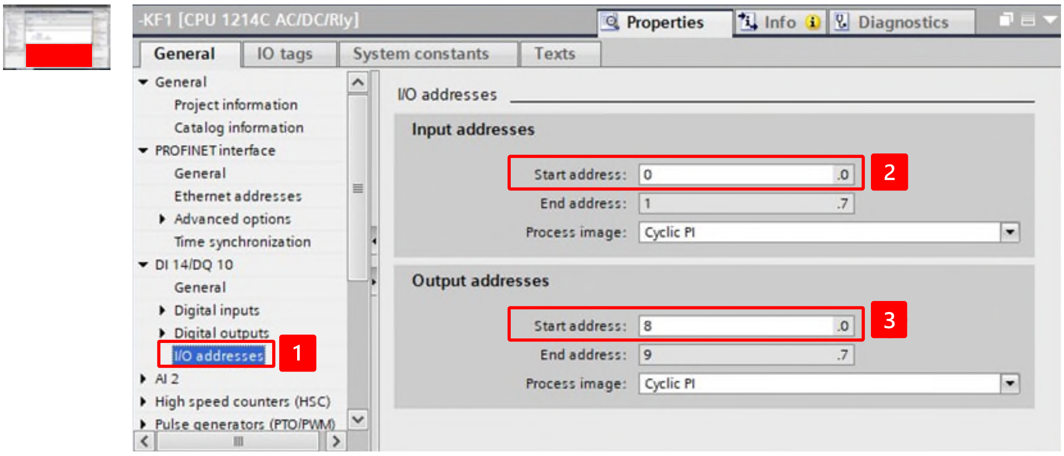
3. Customize the module labeling:



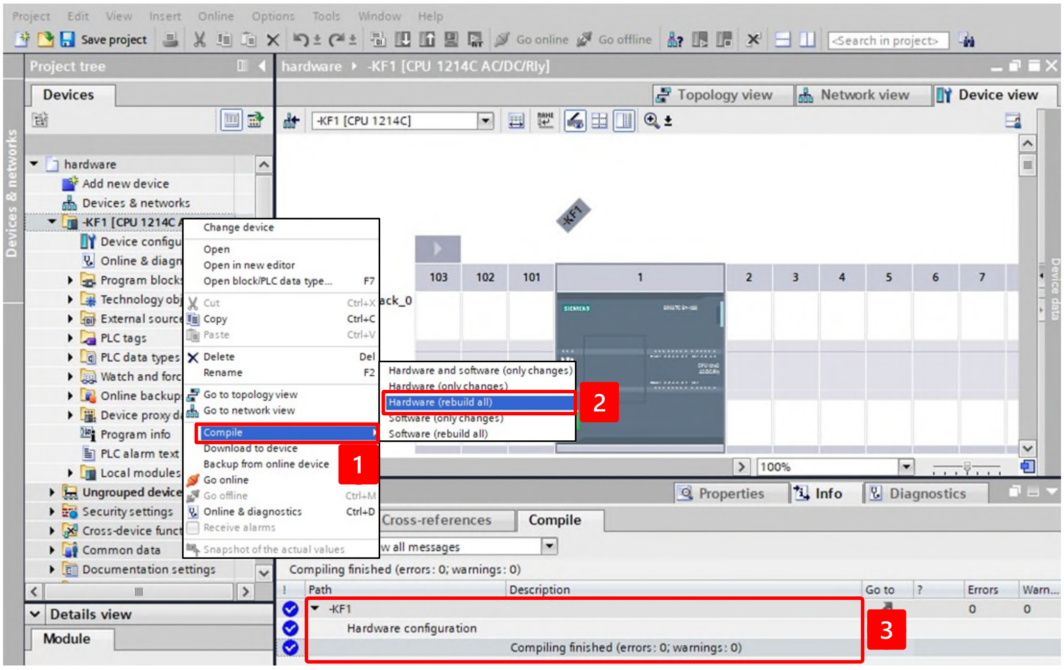
4. Assign unique network parameters:



5. Assign unique input and output addresses:



6. Translate your project planning:

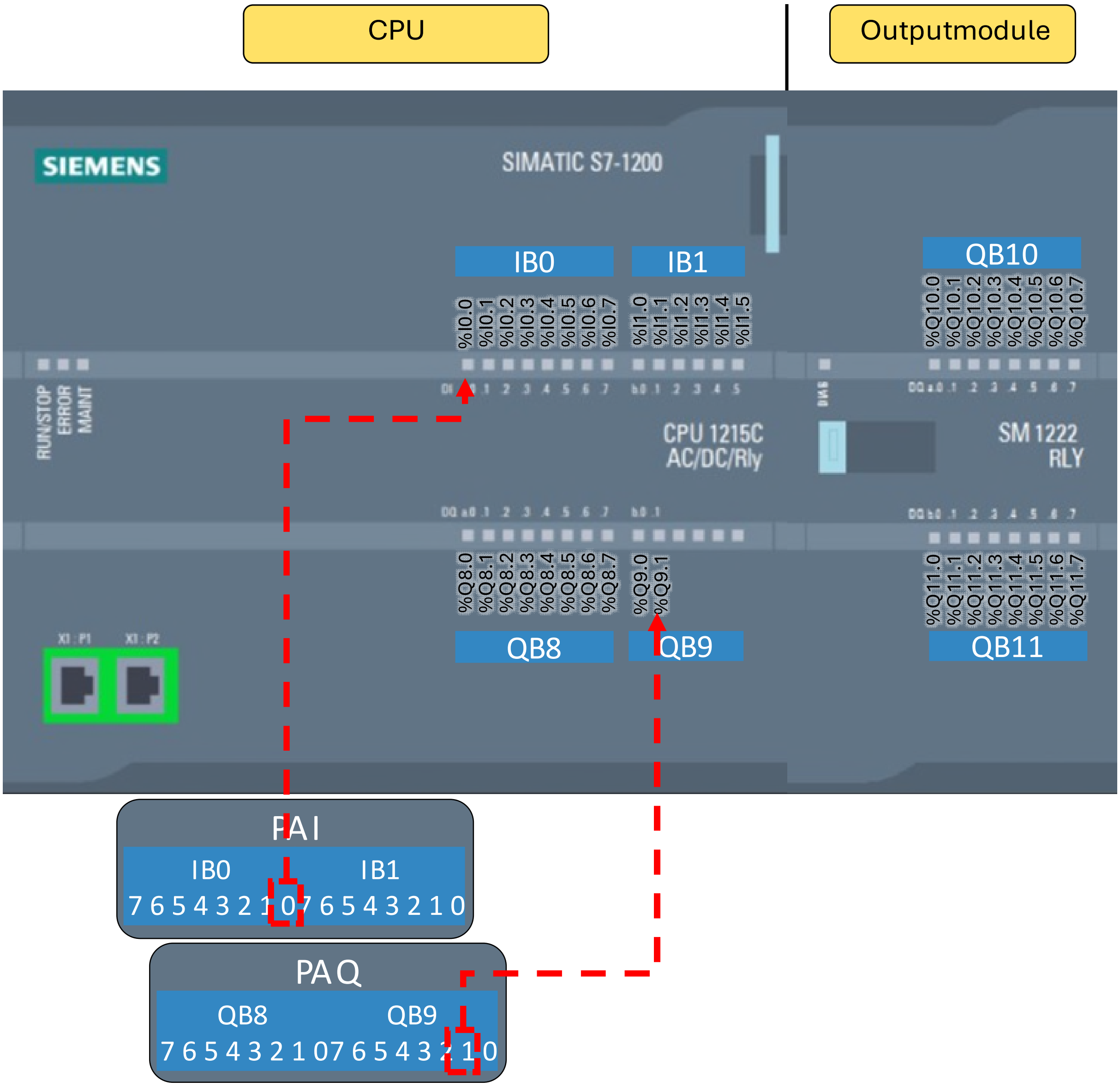


2.3 Assembly and memory addressing

2.3.1 Introduction

Configuration

In order to address the signals of the modules in the program, the modules must be clearly identifiable and addressable. To make this possible, assign them a unique start address in the configuration.



Picture 22 Addressing using the example of a Siemens S7 1200

Each channel of a module is assigned a fixed memory area in the PLC:

- Write inputs to the process image of the "PAE" inputs
- Read outputs from the process image of the "PAA" outputs

Access to the memory addresses in the program

The data in the memory can be addressed via the memory address.
Access can be both read and write.

Access to the memory addresses in the program is defined via an operand identifier and a parameter.

Operand identifiers are as follows:

Range	Labeling	
	German	international
Input	E	I
Output	A	Q
Flags / Memory	M	M

Table 1 Operand identifier

Inputs (I)

The status of the input channels is saved in the process image of the inputs (PAE).

Outputs (Q)

The status of the input channels is saved in the process image of the outputs (PAA).

Marker (M)

Flags are used to store internal states or intermediate results. Their function can be compared to that of auxiliary relays. A separate memory area must exist in the CPU for flags. Its size depends on the CPU type.

2.3.2 Symbolic addressing

Control programs should process and, if necessary, save data from the process. Variables are the means by which this data is recorded. When accessing variables, a distinction is made between symbolic and direct addressing.

Addressing refers to the specification of the storage location of the data. In the PLC, these are, for example, the areas inputs, outputs and flags.

Input and output addresses are assigned to the signal modules in the hardware configuration. The signals connected there can therefore be addressed absolutely.

Operand	Labeling	Parameters	
		Byte address	Bit address
I 1.0	I	1	0
Q 4.2	Q	4	2
M 31.7	M	31	7

Table 2 Absolute representation of bit variables

In addition to this absolute address, a name must be assigned to an operand. The name of a variable then refers to the associated absolute address. If a variable is preferably addressed by its name, this is referred to as symbolic addressing.

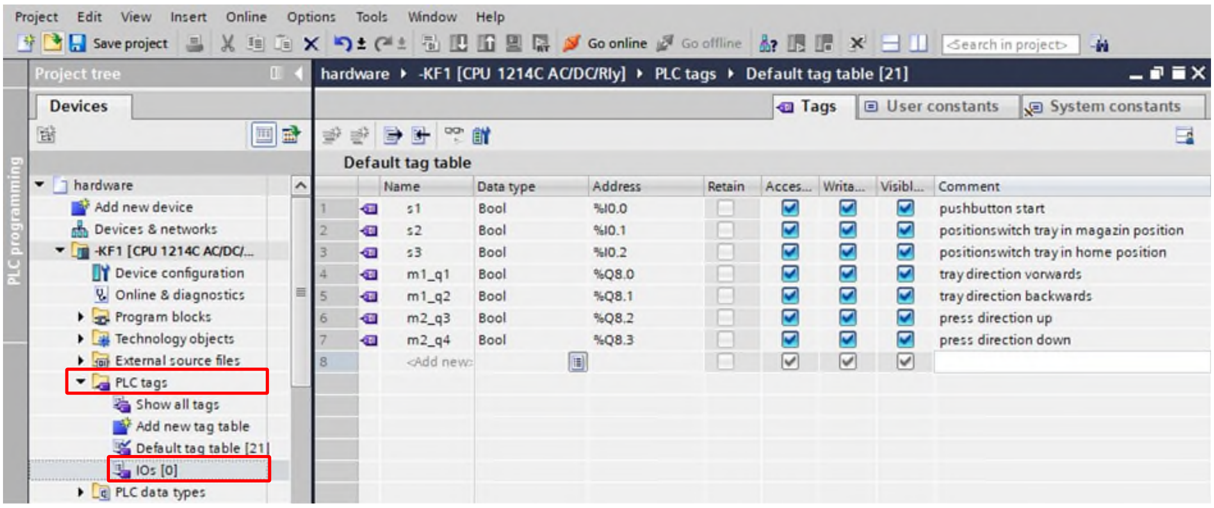
By using meaningful names and comments, you make your program easier to understand and read, which makes program creation and troubleshooting easier. The name can be derived from the equipment identifier, for example.

The assignment between symbolic and absolute addressing takes place in so-called variable tables. Variable tables contain the definitions of the PLC variables that are valid throughout the CPU.
Variable tables are characterized by:

- Common data storage: The variables are used in all editors
  .
- In addition to variables, constants can also be declared.
- Several variable tables can be created.

Siemens TIA Portal

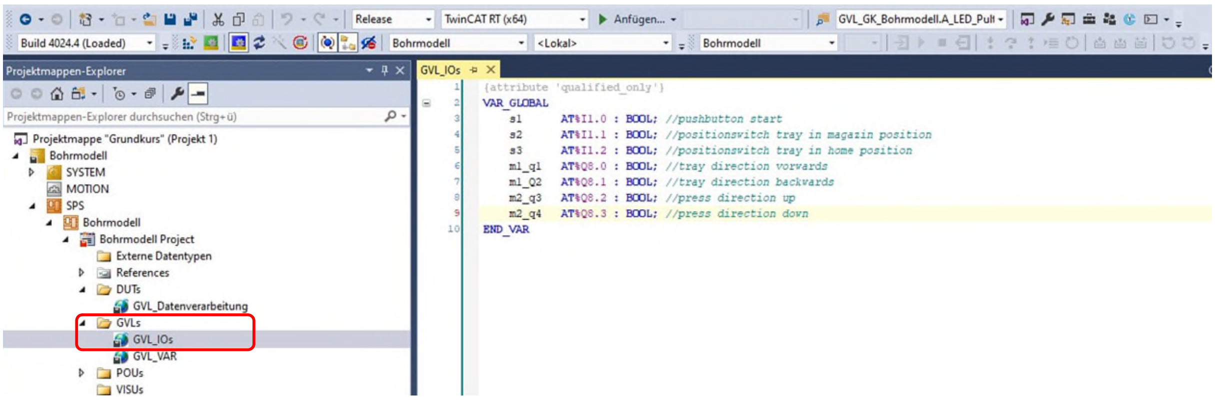
The variables are defined in the project navigation of your project below the corresponding PLC in the "PLC variables" folder in "Variable tables". One or more variable tables can be created and managed in this folder.



Picture 23 PLC variables in the TIA Portal

Beckhoff / Codesys

The variables are stored in "GVLs" (Global Variable Lists). For the sake of clarity, these GVLs should be stored in the "GVLs" folder, but can also be stored in any other folder within the PLC in the Solution Explorer.



Picture 24 PLC variables in TwinCAT (Beckhoff)

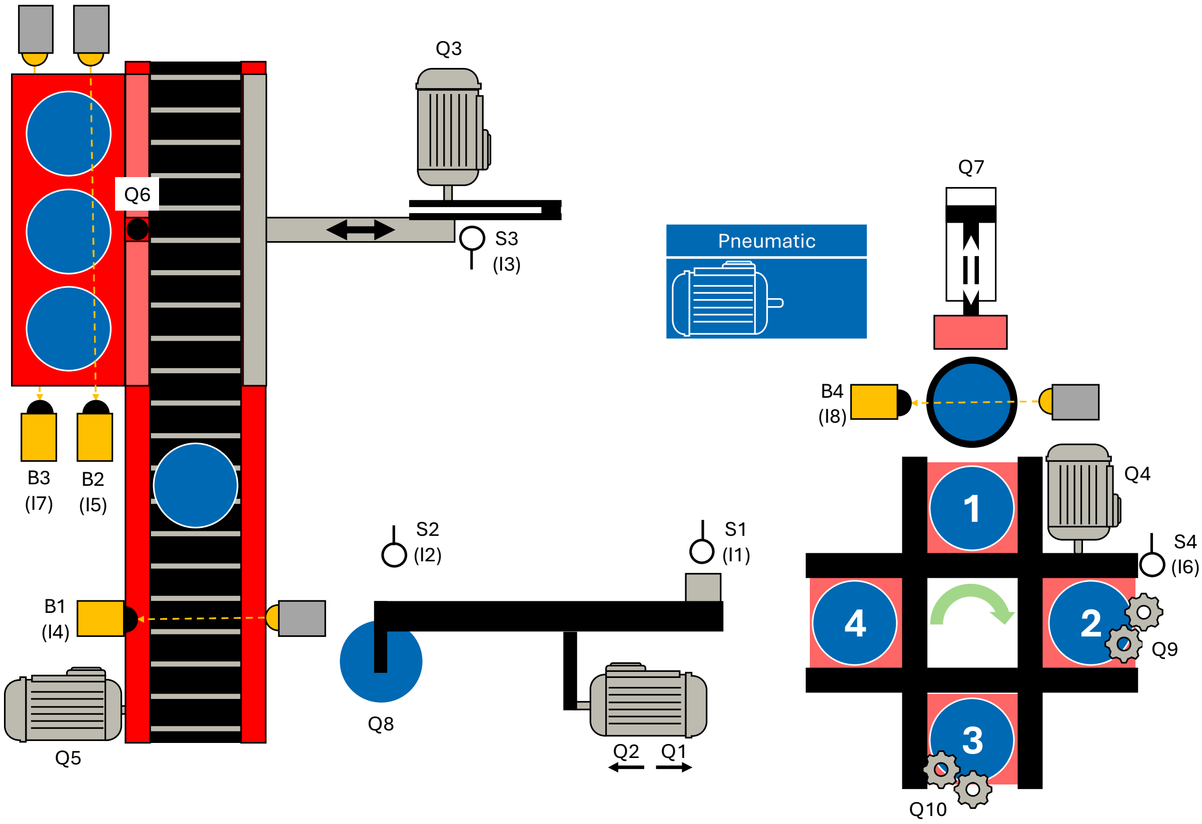
2.3.3 Exercise: Creating PLC variables

Target:

I can create and edit PLC variables.

Task:

Create a PLC variable table and create the input and output variables for the model.

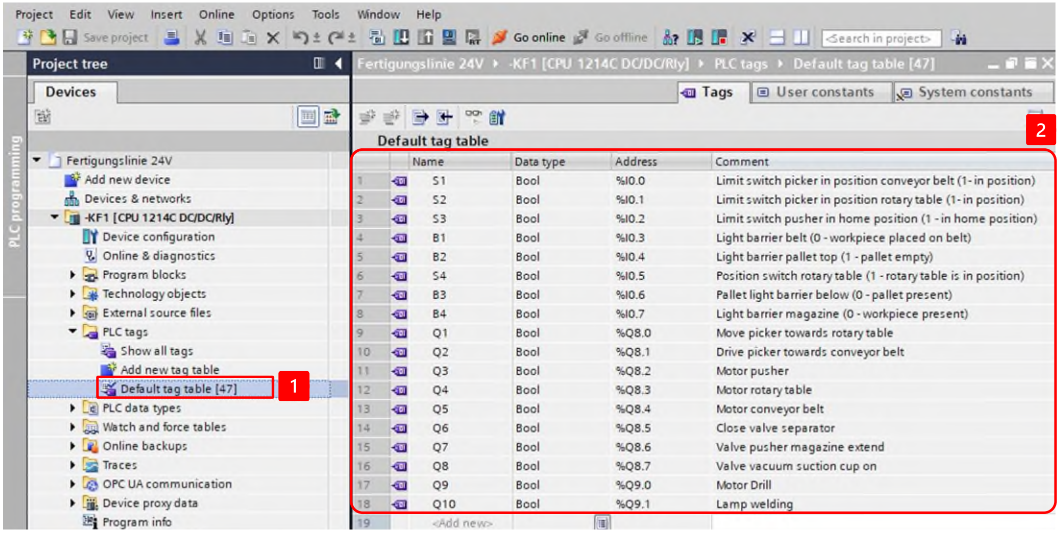


For quick and easy assignment, the absolute and symbolic addresses of the variables can be entered in the "Address" and "Symbol" columns in the "Model" section of the assignment plan.

Procedure:

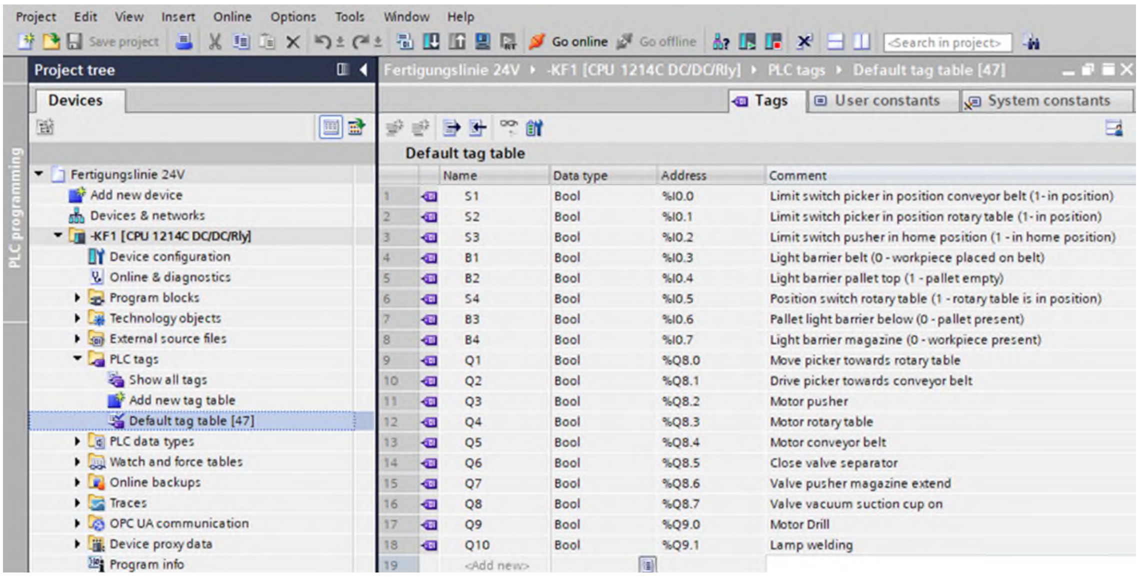
1. Use the "Add new variable table" button to add a new PLC variable table and assign a meaningful name. Or use an existing table (e.g. standard variable table).

2. Assign a meaningful name and comment for each sensor and actuator. Assign the addresses according to your hardware configuration.



Solution

Solution:



Default tag table

	Name	Data type	Address	Comment
1	S1	Bool	%I0.0	Limit switch picker in position conveyor belt (1 - in position)
2	S2	Bool	%I0.1	Limit switch picker in position rotary table (1 - in position)
3	S3	Bool	%I0.2	Limit switch pusher in home position (1 - in home position)
4	B1	Bool	%I0.3	Light barrier belt (0 - workpiece placed on belt)
5	B2	Bool	%I0.4	Light barrier pallet top (1 - pallet empty)
6	S4	Bool	%I0.5	Position switch rotary table (1 - rotary table is in position)
7	B3	Bool	%I0.6	Pallet light barrier below (0 - pallet present)
8	B4	Bool	%I0.7	Light barrier magazine (0 - workpiece present)
9	Q1	Bool	%Q8.0	Move picker towards rotary table
10	Q2	Bool	%Q8.1	Drive picker towards conveyor belt
11	Q3	Bool	%Q8.2	Motor pusher
12	Q4	Bool	%Q8.3	Motor rotary table
13	Q5	Bool	%Q8.4	Motor conveyor belt
14	Q6	Bool	%Q8.5	Close valve separator
15	Q7	Bool	%Q8.6	Valve pusher magazine extend
16	Q8	Bool	%Q8.7	Valve vacuum suction cup on
17	Q9	Bool	%Q9.0	Motor Drill
18	Q10	Bool	%Q9.1	Lamp welding
19	<Add new>			

The equipment identifier was used as the symbolic name.
The addresses were adapted according to the hardware configuration:

- Inputs in byte EB 0
- Outputs in byte AB 8 and 9