Bending machine 24V

Hardware configuration

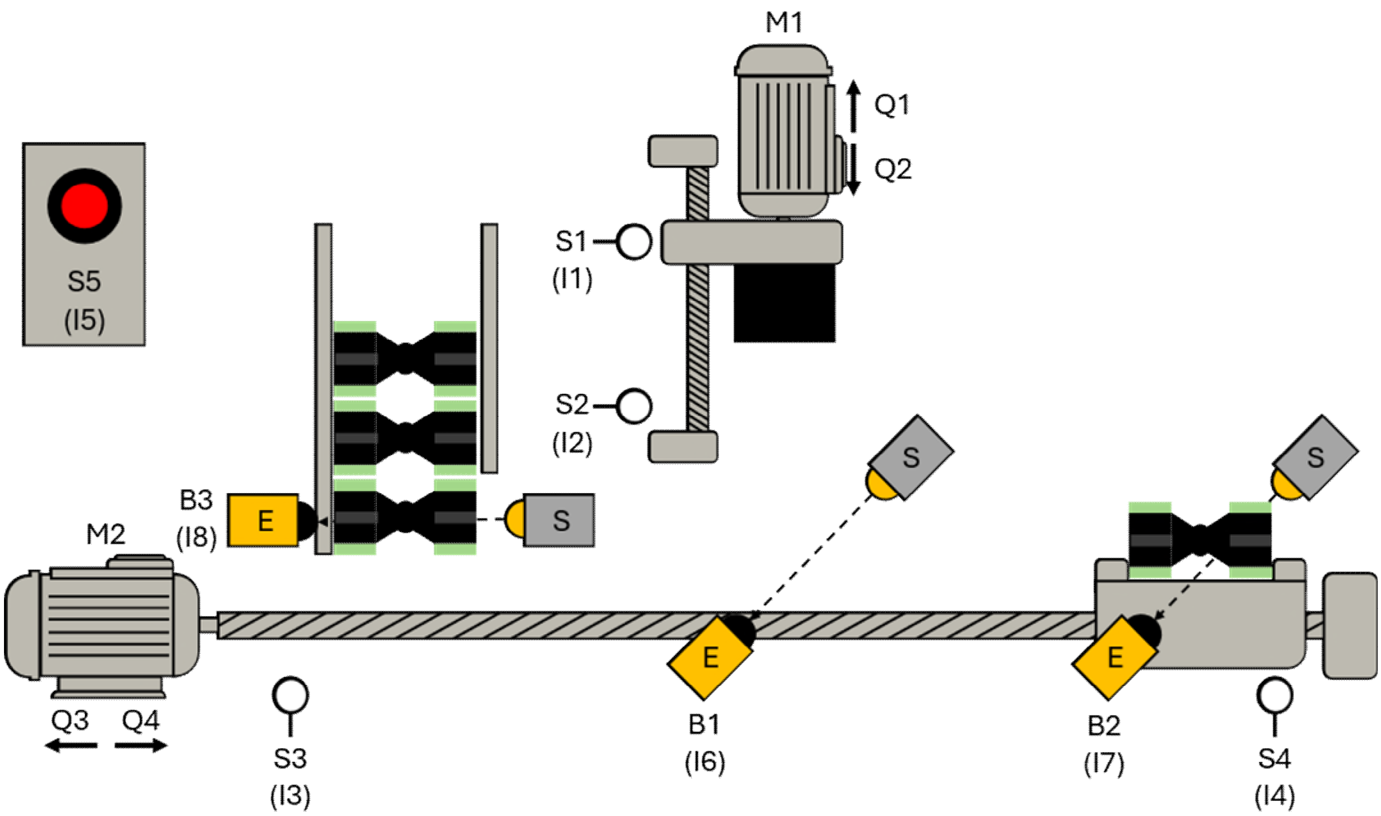


Table of contents

2 Hardware configuration 1
2.1 Components of a PLC 1
2.1.1 Power Supply 2
2.1.2 Central Module 3
2.1.3 Input Modules 4
2.1.4 Output Assemblies 5
2.1.5 Binary Signals 5
2.1.6 Signalzustandsinformation 7
2.2 Hardware Configuration 8
2.2.1 TIA 9
2.2.2 Exercise: Hardware Configuration 15
2.3 Assembly and Memory Addressing 19
2.3.1 Introduction 19
2.3.2 Symbolic Addressing 21
2.3.3 Exercise: Creating PLC Variables 23

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 assembly, the input and output modules and a possible power supply module.

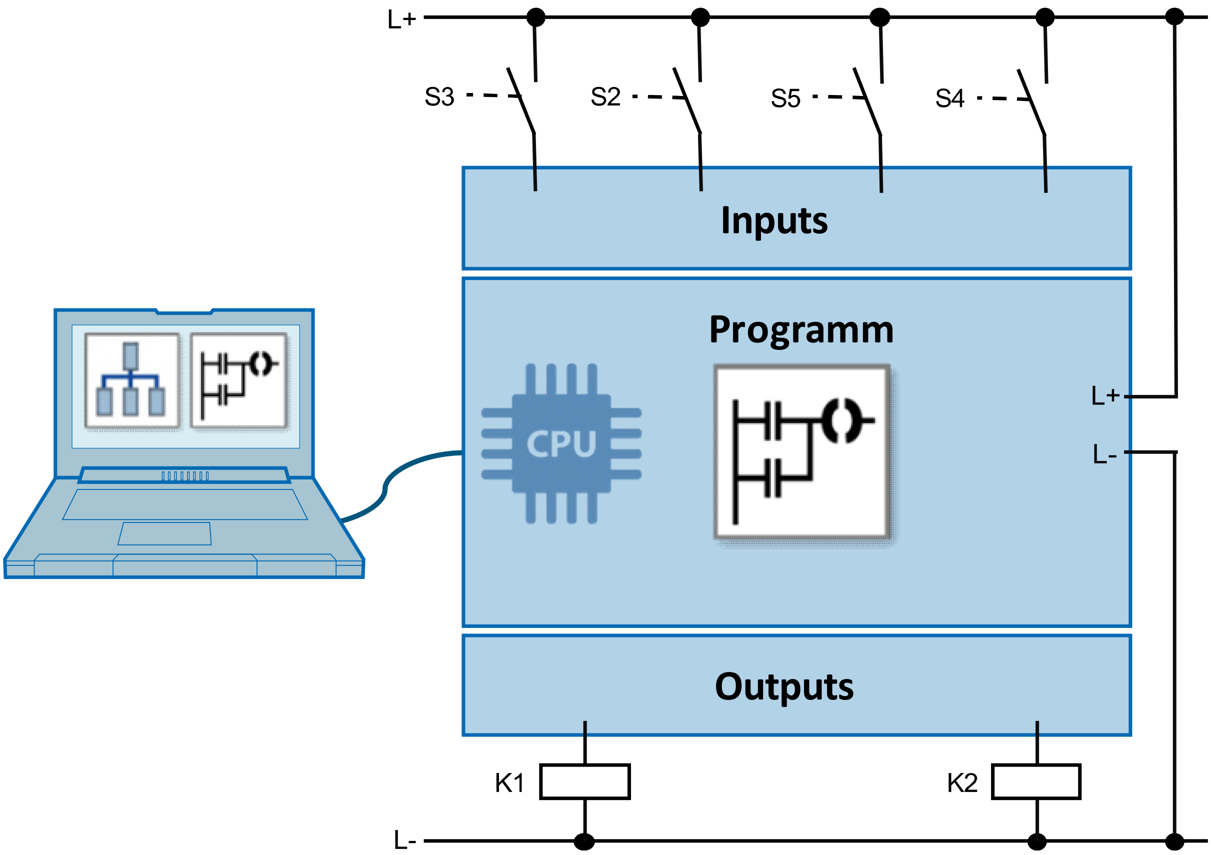


Figure 1 PLC - How it works

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

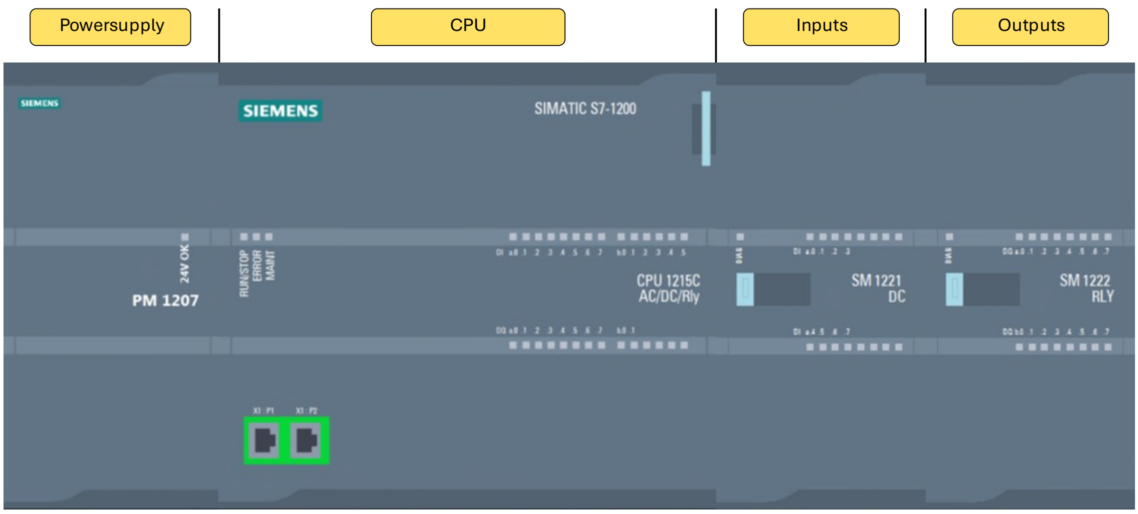


Figure 2 Structure of a Siemens S7-1200 controller

In the following, the components of an S7-1200 are explained 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 signal transmitters, actuators and light detectors.

Figure 3 S7-1200 Load Power Supply

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

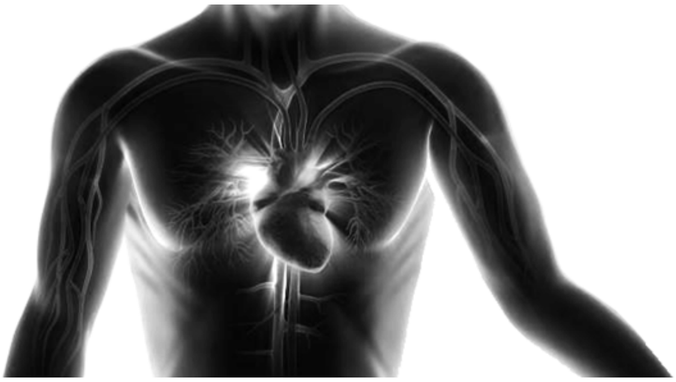


Figure 4 Cardiovascular system

2.1.2 Central Assembly

The control unit of the central module (CPU = Central Processing Unit) processes the program stored in the program memory.
During program processing, the status of the inputs is queried. Depending on the signal state of the inputs and the program stored in the program memory, the outputs are then controlled by the control unit.

Figure 5 S7-1200 CPU

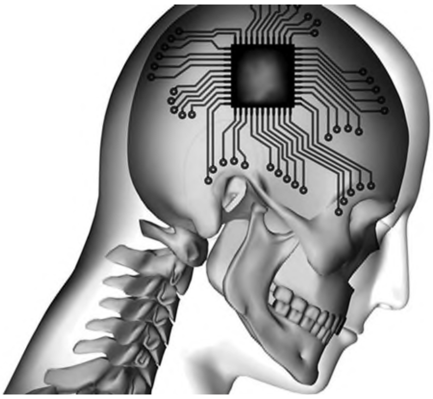


Figure 6 Brain

Compared to humans, this would be the brain, which processes all control processes.

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

2.1.3 Input Assemblies

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
- Counts

With the help of these signals, the CPU records the current system status.

Figure 7 S7-1200 DI Module

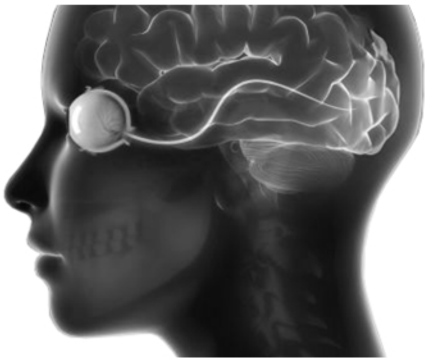


Figure 8: Human senses

The input assemblies capture the signals from the sensors and forward them to the CPU, similar to how the human eye transmits 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
- Contactor and valve controls
- Driving commands

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

Figure 9 S7-1200 DO Assembly

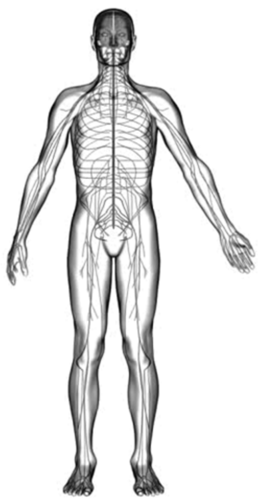


Figure 10 Human limbs

In analogy to humans, it is the limbs that respond to the brain's commands.

2.1.5 Binary Signals

In programmable logic controllers (PLC), a lot of information is processed and controlled with the help of binary, i.e. bivalent, signals.
These are read into the PLC by digital input modules and transmitted via digital output assemblies.

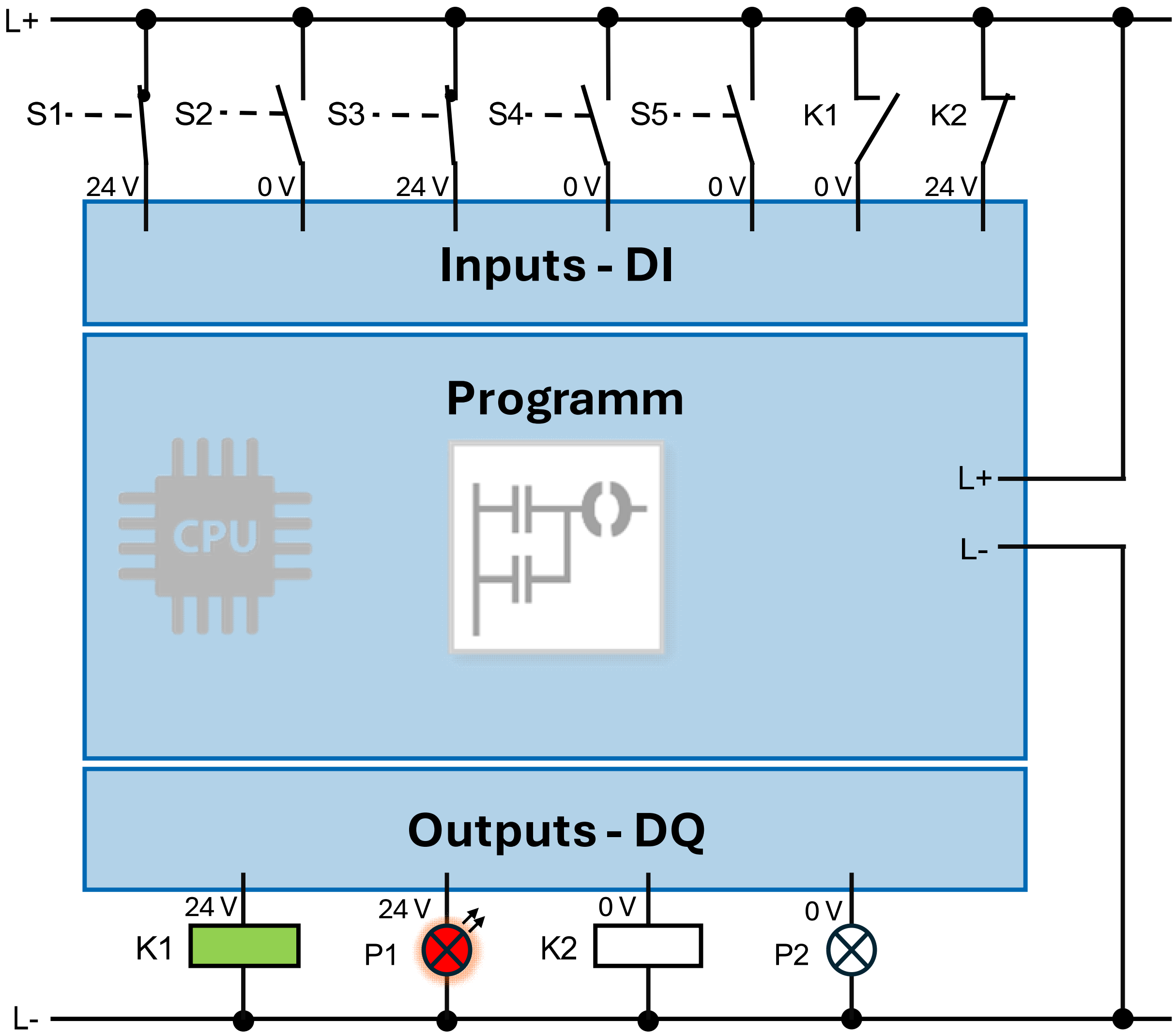


Figure 11 Connection of binary signals

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

2.1.6 Signalzustandsinformation

Binary Input Signals
The state of a binary input signal is detected by the applied voltage.
Two signal states can be distinguished.

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

Binary Output Signals
The same applies to the binary output signals.

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

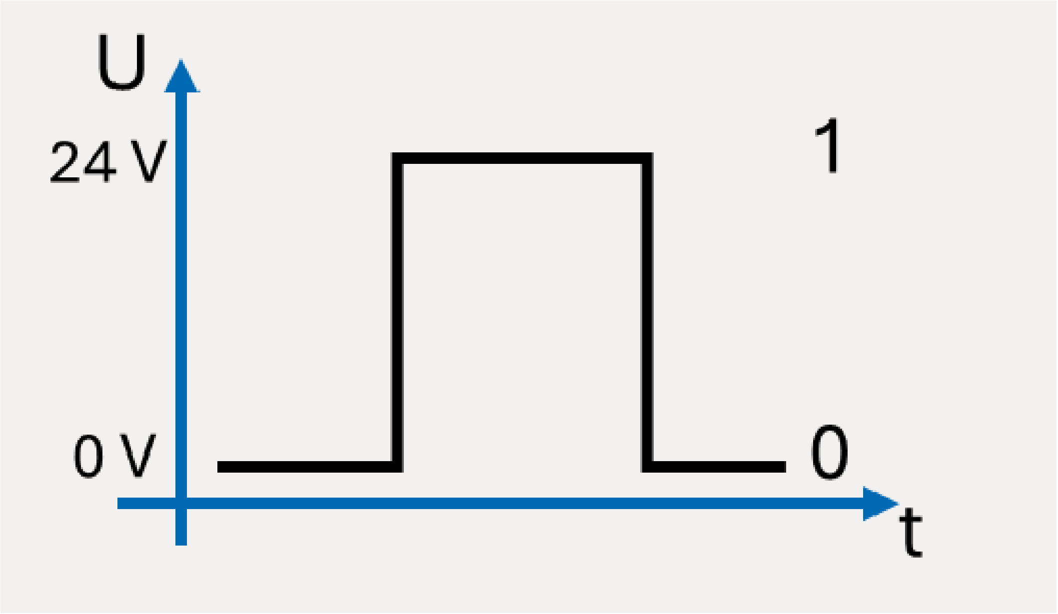


Figure 12 Binary Signal

2.2 Hardware configuration

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

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

  CPU
  - Communication addresses (e.g. IP address, other bus addresses)
  - Assembly Labeling (Name)

  Signalmodule
  - Entry/exit addresses
  - Assembly Labeling (Name)

- Configuration can be translated without errors

Subsequently, the procedure is 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

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, in which an S7 1200 controller is inserted as a new device.

You will be assisted in selecting the devices by a wizard.
After pressing the "Add new device" button, there are three device groups to choose from:

- Controller
- HMI
- PC Systems

After selecting a device group (in this case the "Controllers" group), the device to be inserted can be selected from a tree structure based on the article number. When inserting, care must be taken to select the correct firmware version.
It is advisable to give the device a meaningful name (e.g. the equipment identifier).

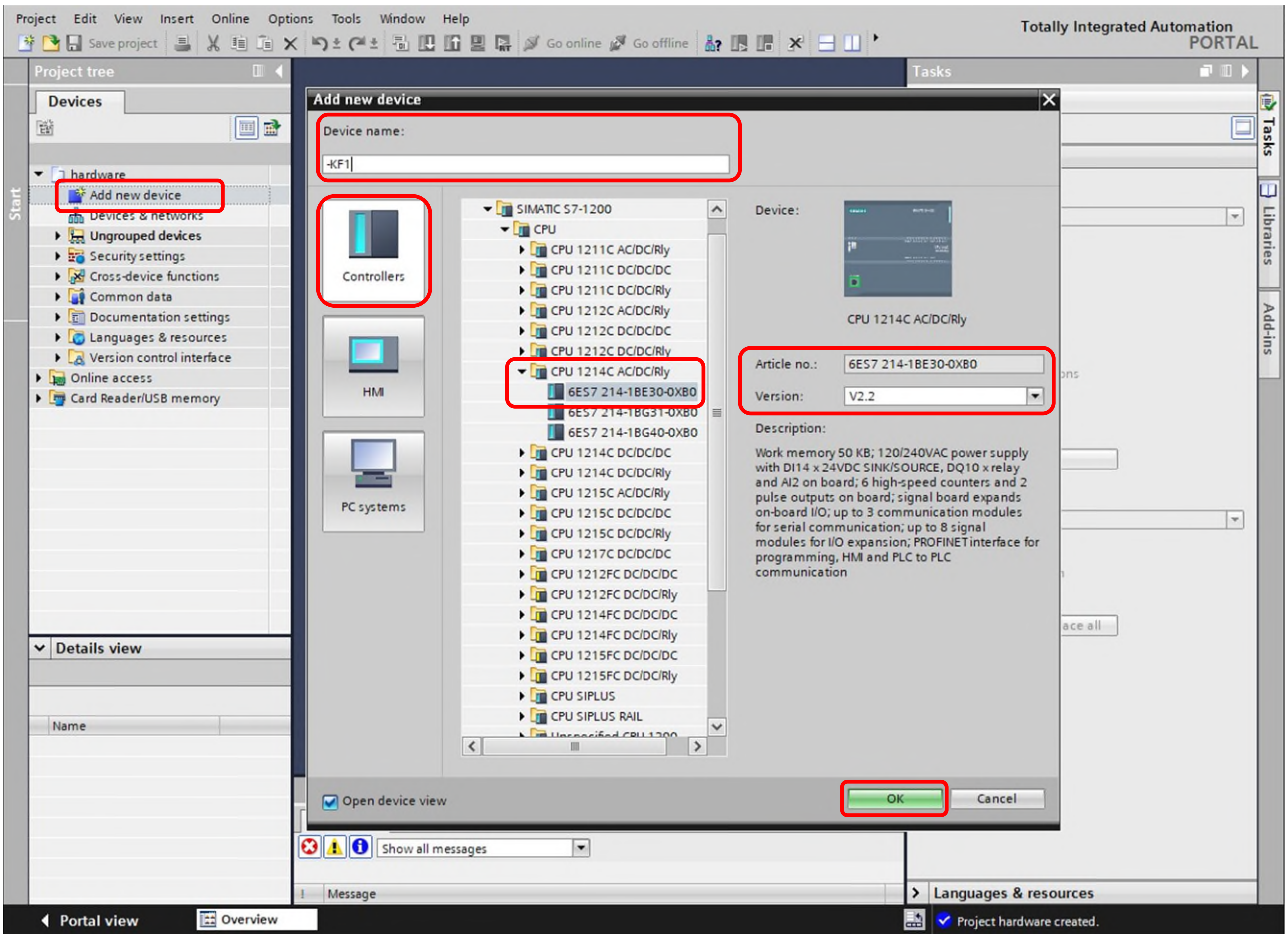


Figure 13 "Add New Device" Dialog

After adding the CPU, the device view opens.
It can also be opened 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 assemblies are used in your system. This includes the selection and arrangement of subracks as well as the assemblies within the subracks. You select the individual assemblies from the hardware catalog in the task cards. When parameterizing, you define properties for each (parameterizable) assembly (e.g. address).

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

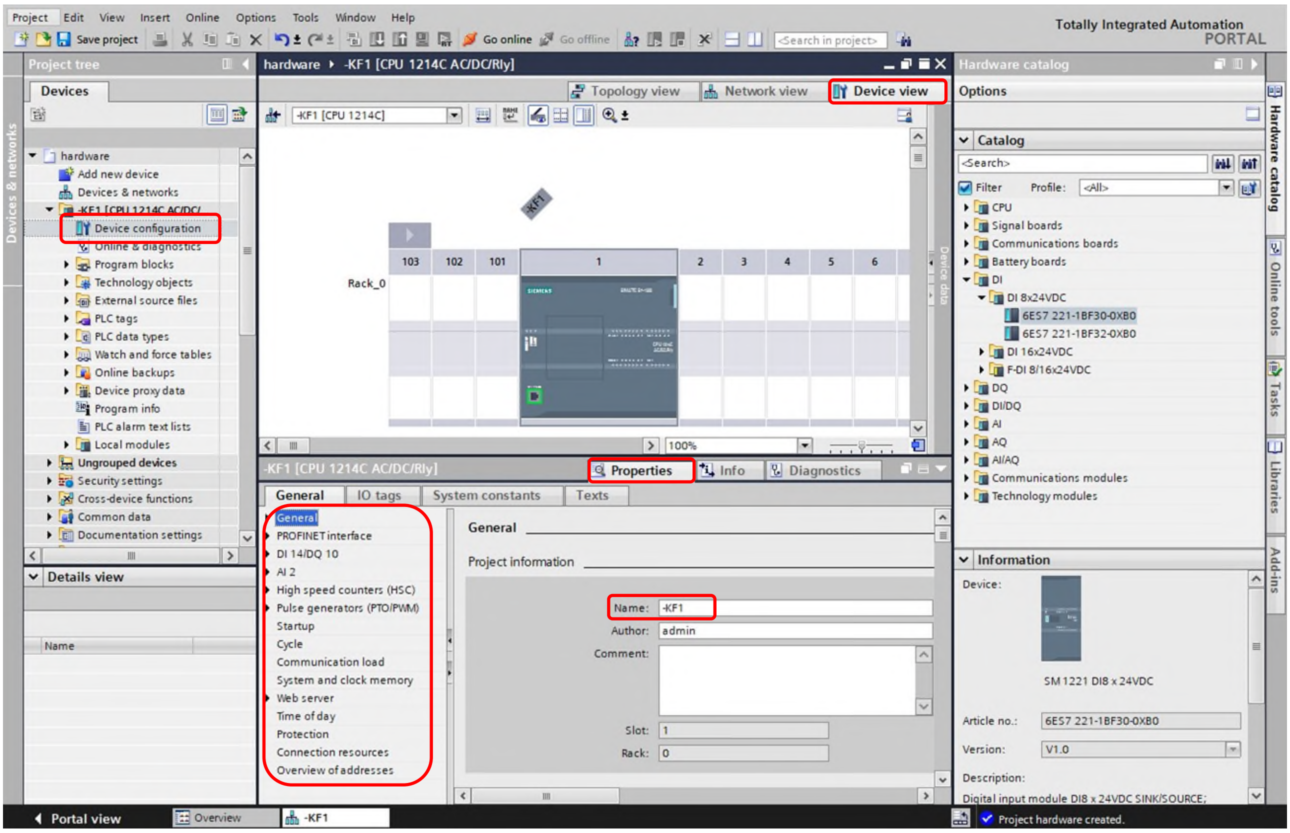


Figure 14 Device view

If the device has a component label, 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) is done via the "Subnet" setting. By default, "not networked" or "PN/IE_1" can be selected here.

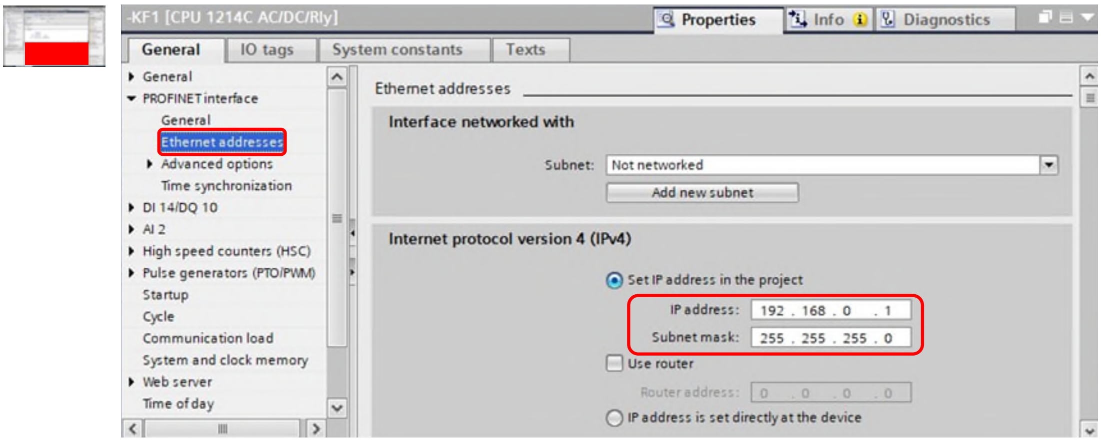


Figure 15 Ethernet Address

System und Taktmärk

In the properties of the PLC under "System and Clock Bits" you can define and activate marker bytes for system marker bits and one for clock bits.

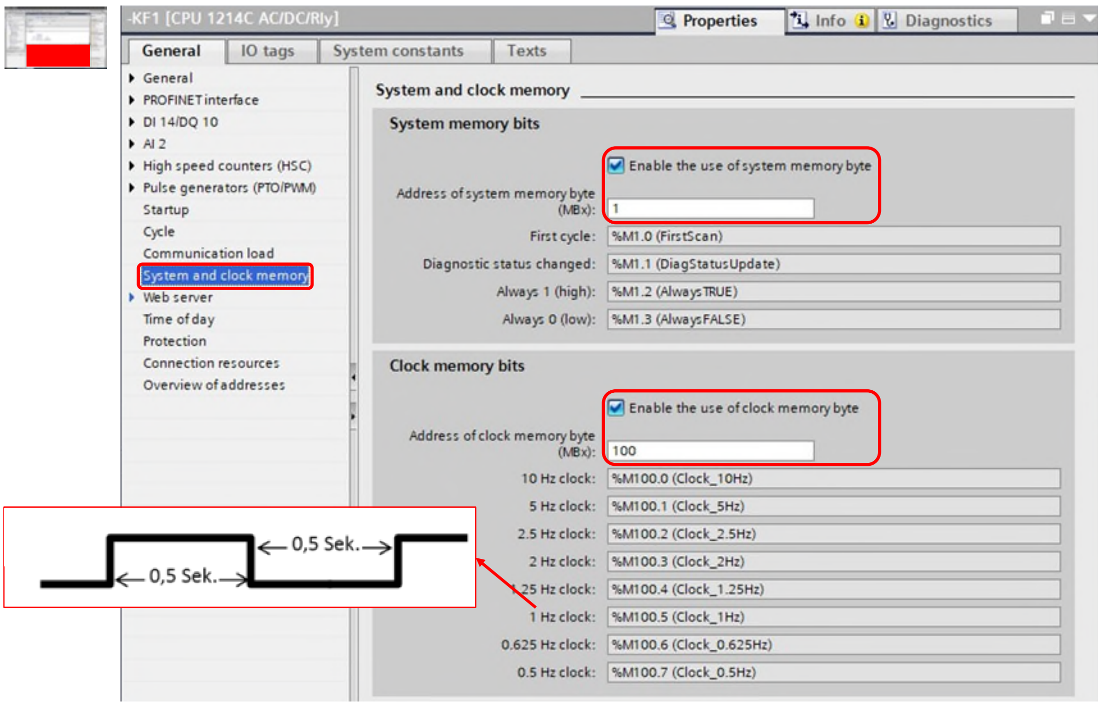


Figure 16 System and clock marker

In a clock byte, the individual bits have different fixed frequencies. The address of the marker byte is determined when parameterizing the CPU. Clock markers can be used, for example, for calculations or flashing displays.

Adding and parameterizing peripheral modules

From the hardware catalog, you can add additional assemblies to the rack.
There are the following ways to do this:

- by dragging and dropping from the hardware catalog to a free valid slot
- by double-clicking 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.

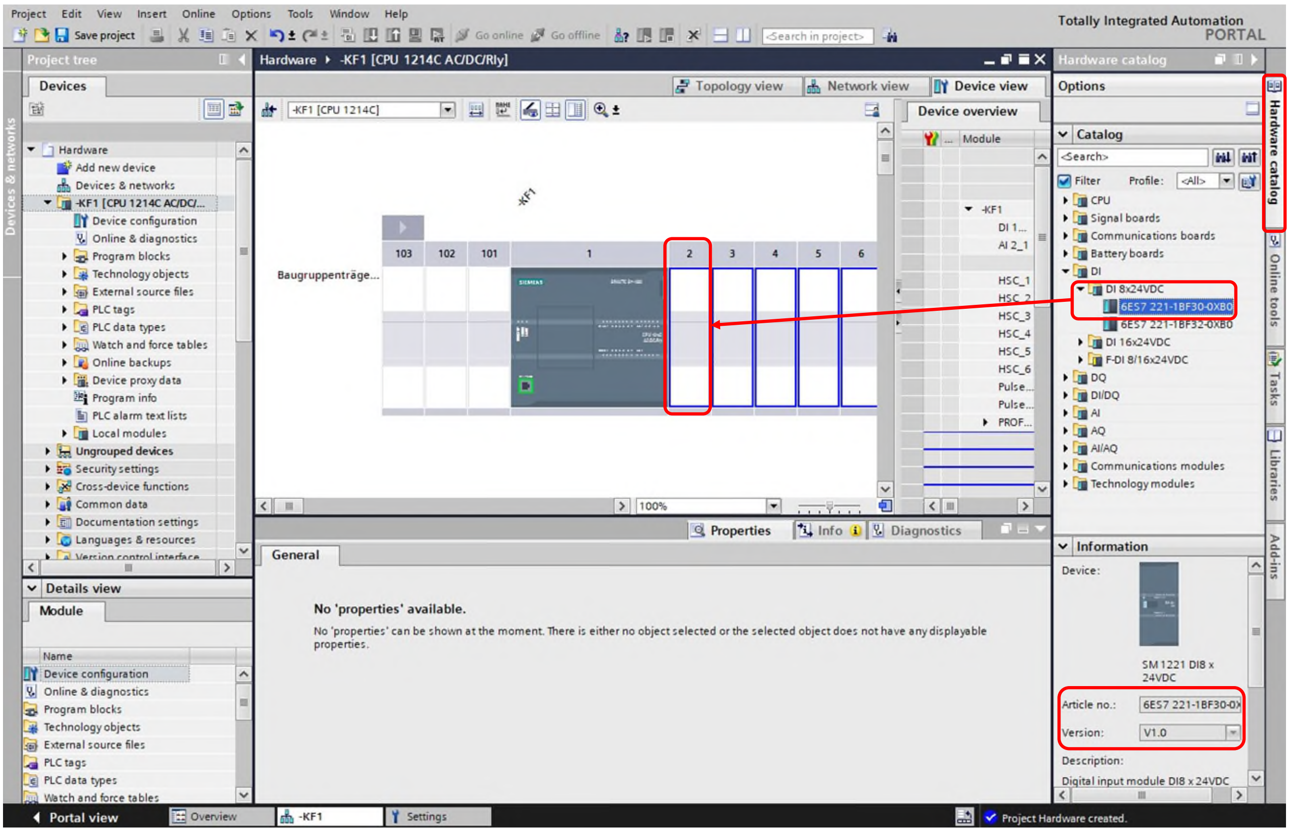


Figure 17 Adding the Signal Module

Siemens assemblies are identified by the article number. This is printed on each assembly. Before inserting the assembly, it is important to ensure that the firmware version is selected correctly in the "Information" palette.

When mating, I/O addresses and other parameters are pre-occupied.
If the assembly is selected, these parameters can be adjusted in the Inspector window under "Properties".

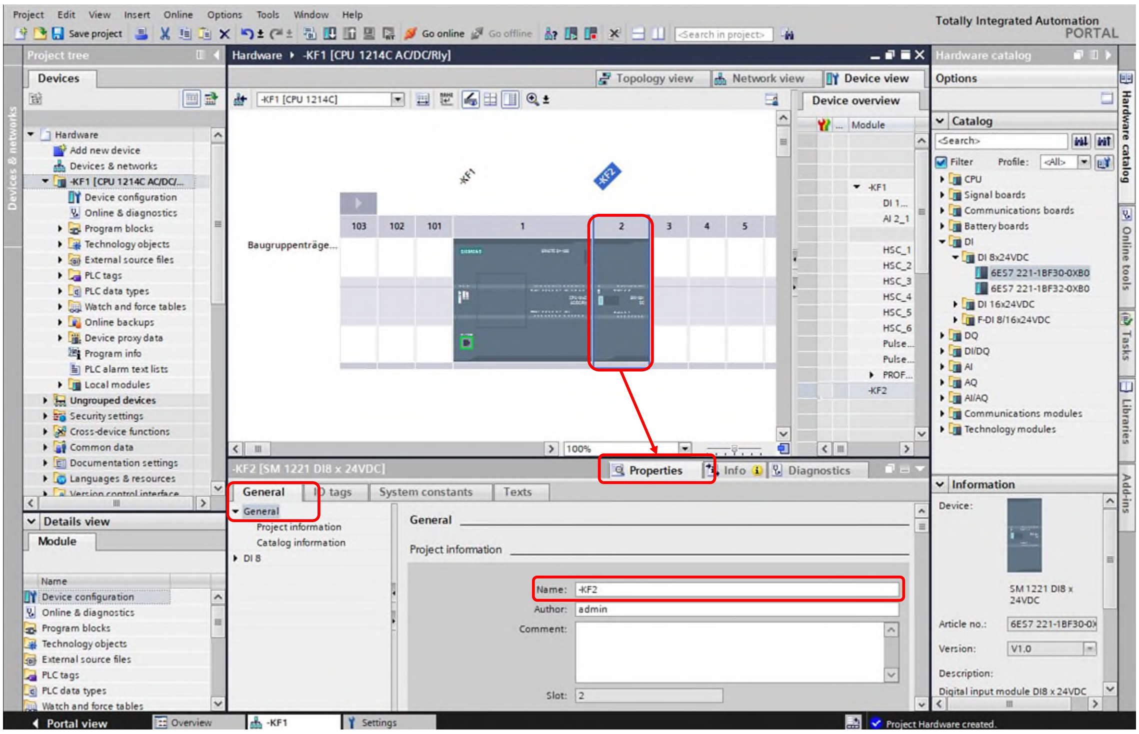


Figure 18 Signal Module Properties → General

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

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

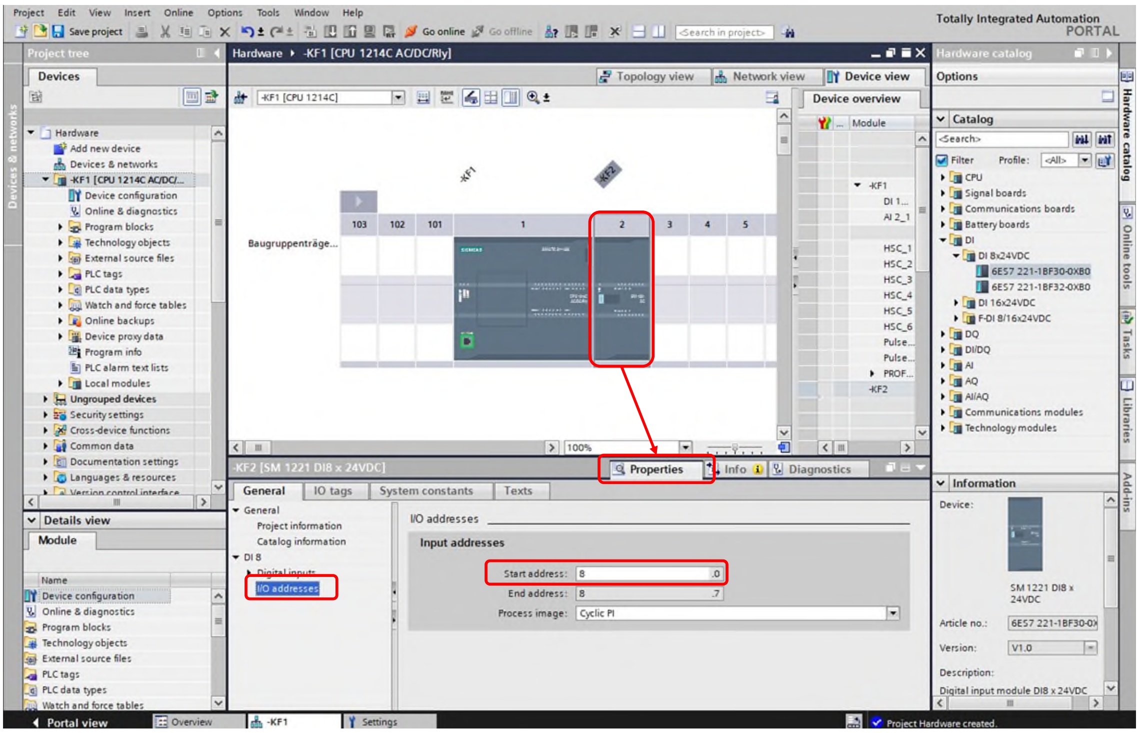


Figure 19 Signal Module I → /O Address Properties

Translating Project Planning Data - Hardware

Before the configuration data can be loaded into the PLC, the configuration must have been translated without errors. When translating, the project planning is checked for consistency.
You can initiate the translation explicitly, e.g. via the context menu of the device's right-click button in the project navigation or via the button in the function bar of the program editor.

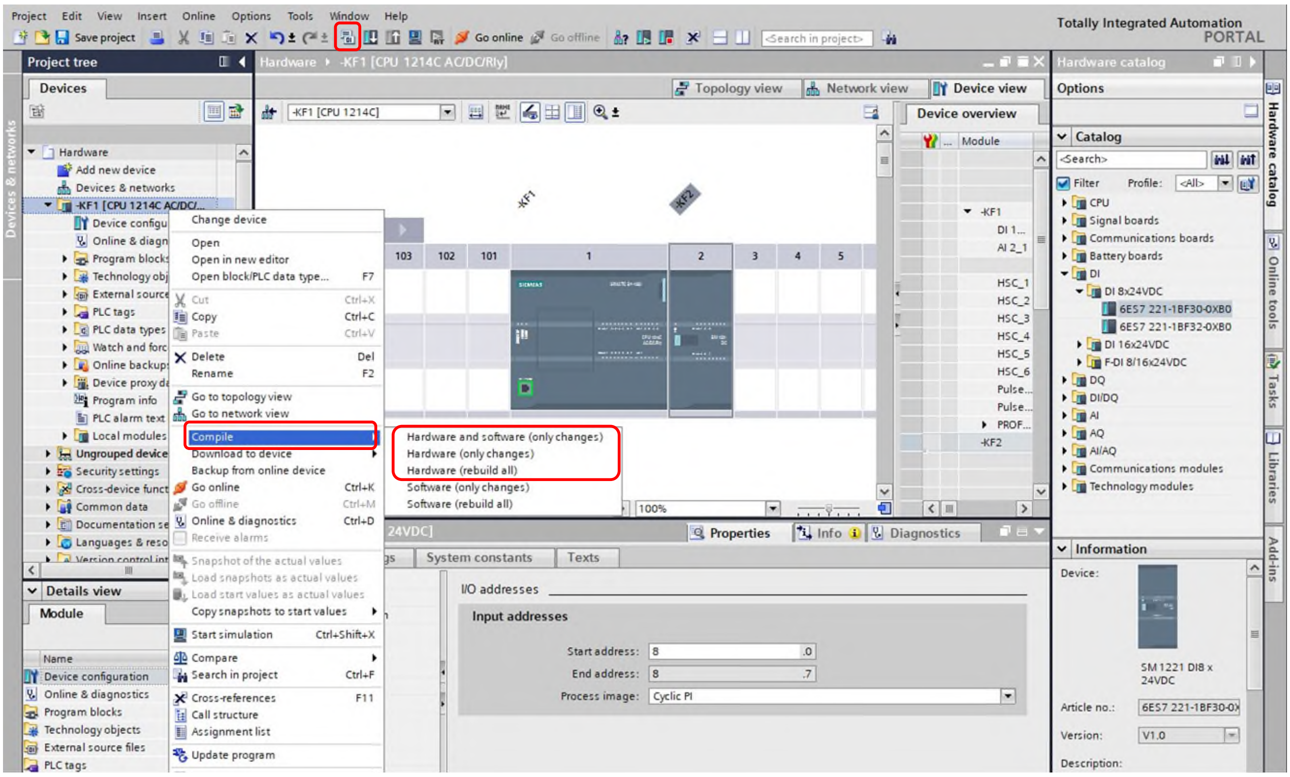


Figure 20 Translate

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

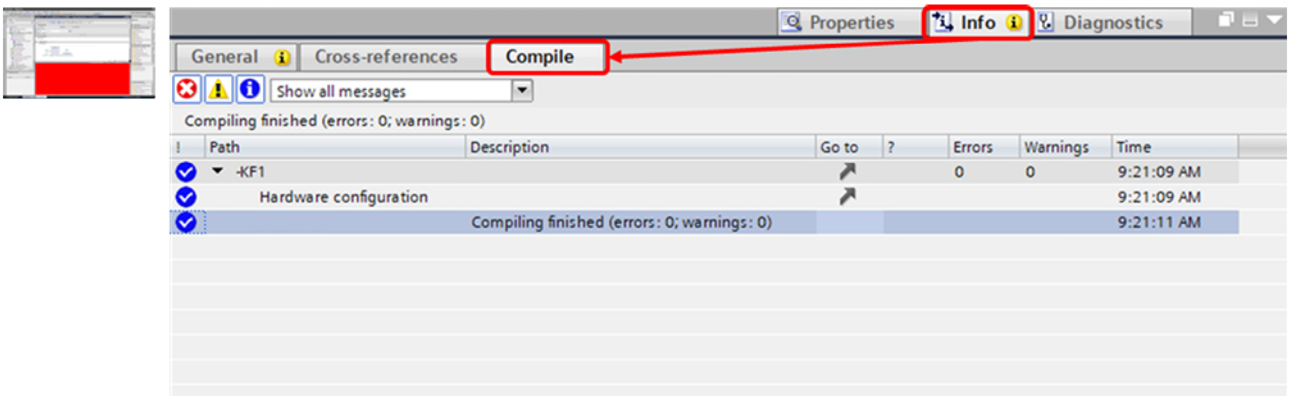


Figure 21 Translate tab in the inspector window

The column "Go to" takes you to the fault location. If the translation contains errors, the device cannot be charged. In the event of warnings, charging is generally possible.

Nevertheless, it is advisable to remove all warnings as well.

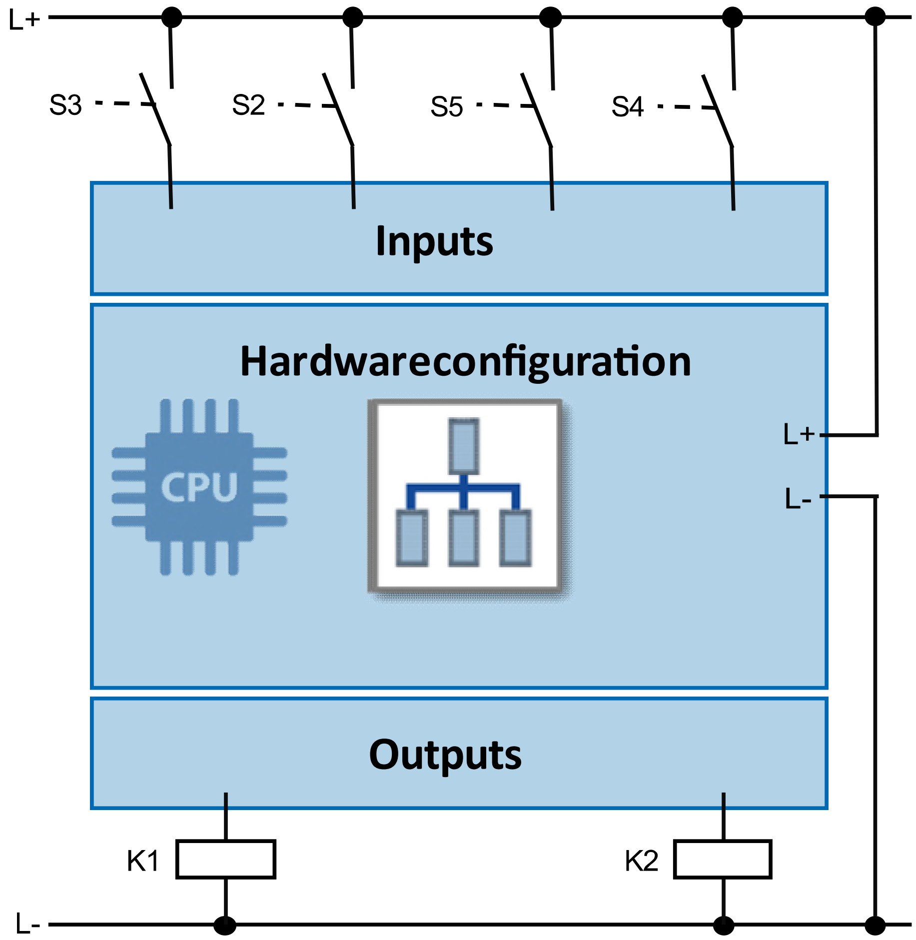
2.2.2 Exercise: Hardware configuration

Goal:

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

Task:

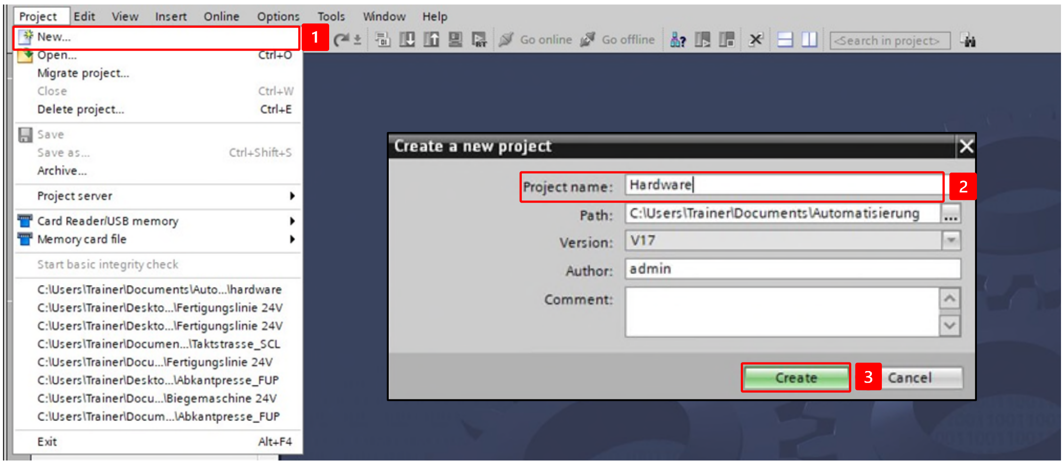
Configure the hardware according to your target system and translate the project planning data.



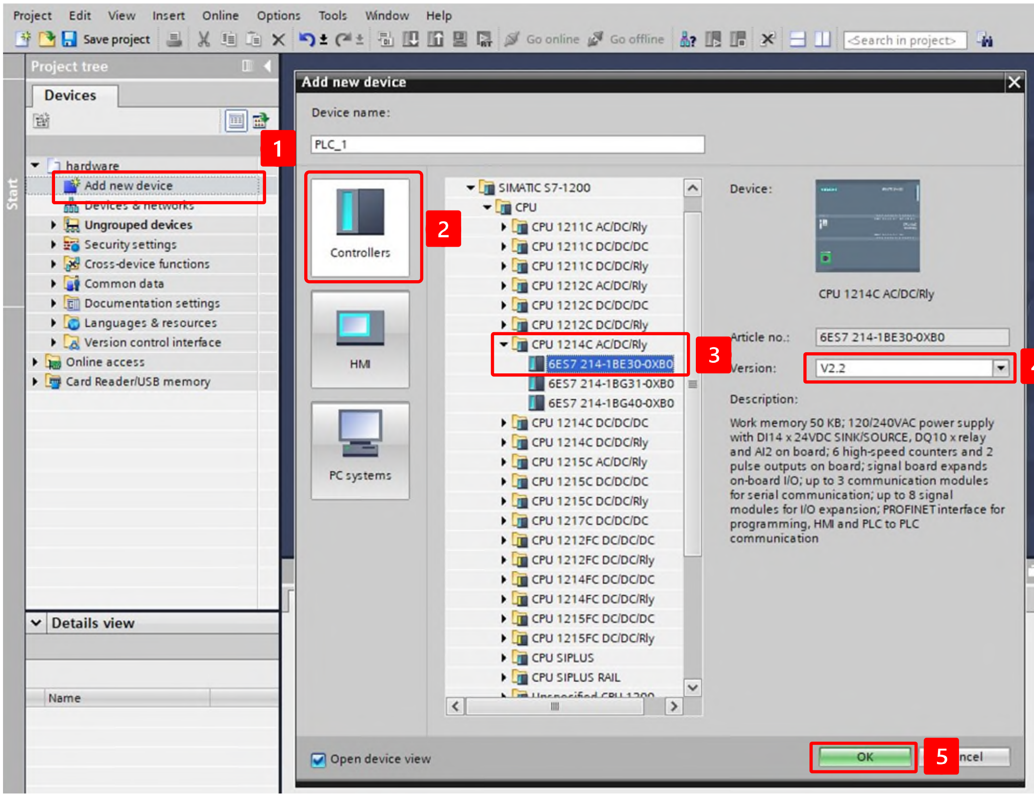
Procedure:

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

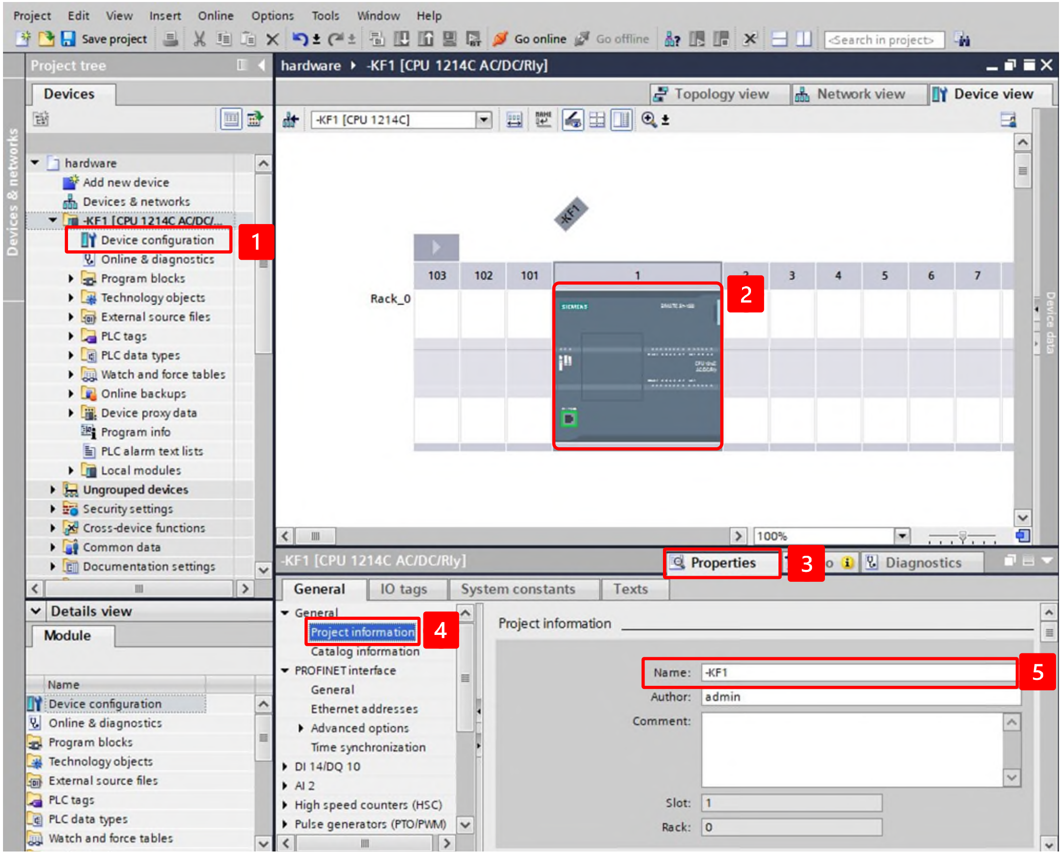
1. Create a new TIA-Portal project and give it a meaningful name:



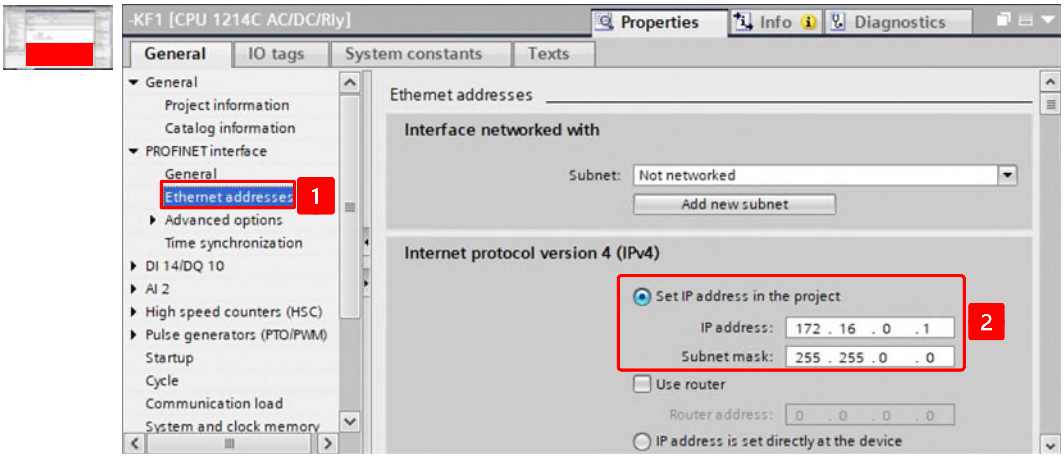
2. Add the CPU.
   Pay attention to the correct order number and firmware.



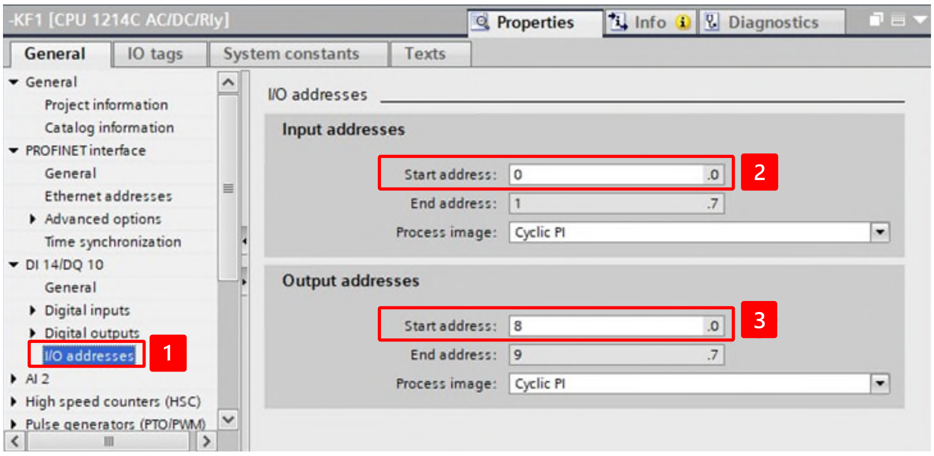
3. Customize the assembly labeling



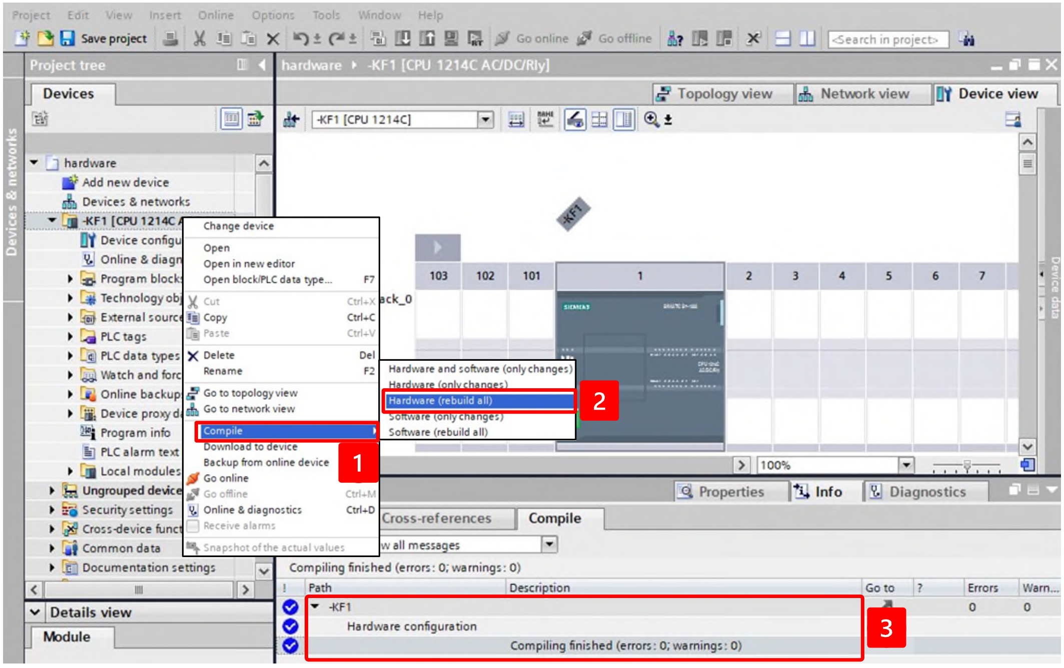
4. Assign unique network parameters



5. Enter unique entry and exit 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 assemblies in the program, the assemblies must be clearly identifiable and addressable. To make this possible, assign a starting address to each of them once in the configuration.

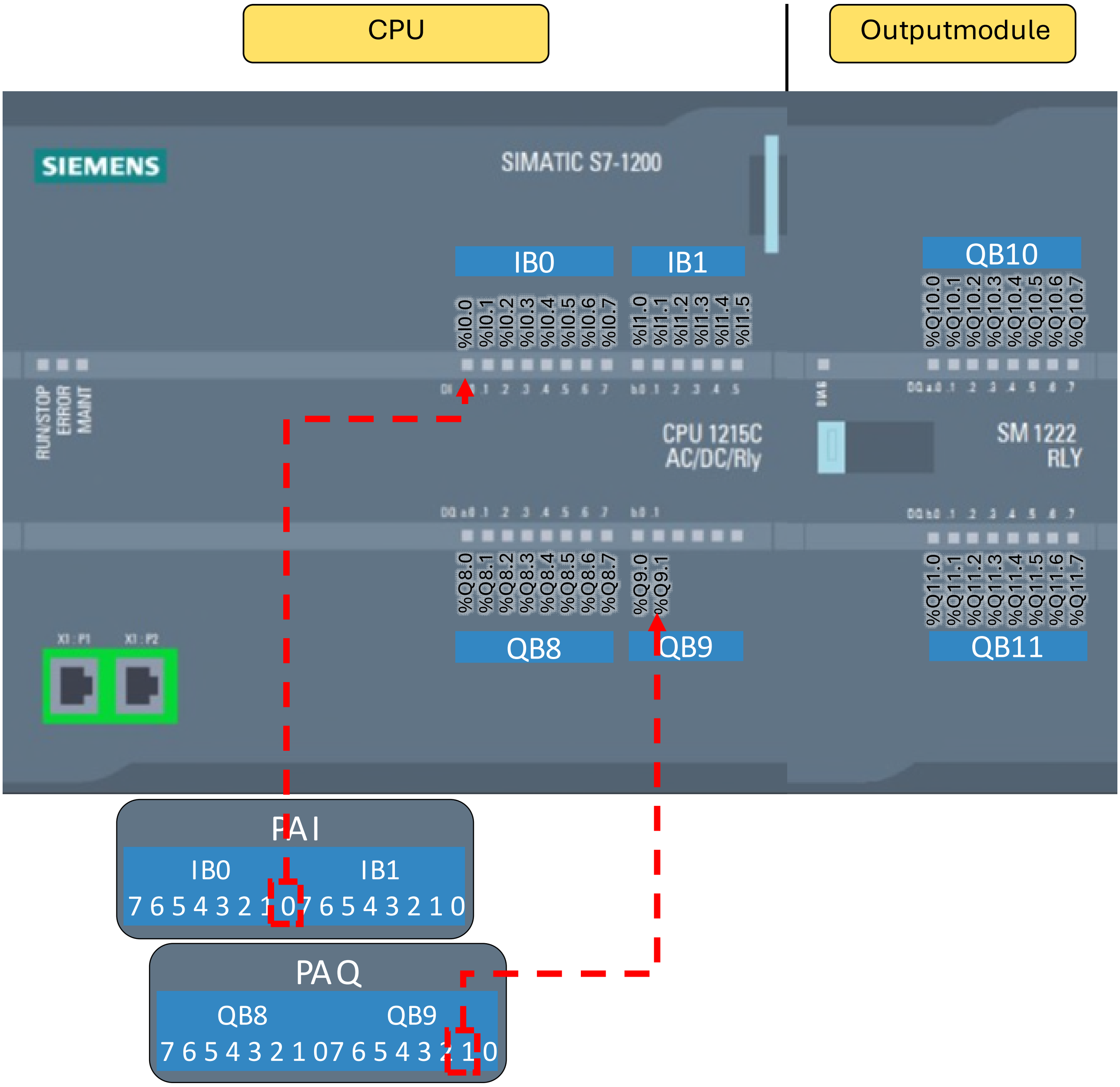


Figure 22 Addressing using the example of a Siemens S7 1200

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

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

Access to the memory addresses in the application
The memory address can be used to access the data in the memory.
Access can be both read and write.

Accesses to the memory addresses in the program are defined by an operand flag and a parameter.

Operand identifiers are as follows:

Bereich	Kennzeichnung	
	Deutsch	international
Eingang / Input	E	I
Ausgang / Output	A	Q
Merker / Memory	M	M

Table 1 Operand characteristics

Inputs (I)
The status of the input channels is stored in the process image of the inputs (PAE).

Outputs (Q)
The status of the input channels is stored in the process image of the outputs (PAA).

Merker (M)
Flags are used for storing internal states or intermediate results. Their function can be compared to that of auxiliary relays. For Merker, there must be a separate memory area in the CPU. Its size depends on the CPU type.

2.3.2 Symbolic addressing

Control programs are supposed to process data from the process and, if necessary, store it. The variables are the means to capture this data. When accessing variables, a distinction is made between symbolic and direct addressing.

Addressing is the indication of the location of the data. In the case of the PLC, these are, for example, the areas Inputs, Outputs and Markers.

In the hardware configuration, input and output addresses are assigned to the signal modules. The signals connected there are therefore absolutely addressable.

Operand	Marking	Parameter	
		Byteadresse	Bit location
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, an operand must be given a name. The name of a variable then refers to the corresponding absolute address. If a variable is preferably addressed by name, this is called symbolic addressing.

By using meaningful names and comments, you make your program easier to understand and read, making it easier for you to create and troubleshoot. The name can be derived from the equipment identifier, for example.

The assignment between the symbolic and absolute addressing is done in so-called variable tables. Variable tables contain the definitions of the CPU-wide PLC variables.
Variable tables are characterized by:

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

Siemens TIA-Portal

The variables are defined in Variable Tables in your project's project navigation below the corresponding PLC in the PLC Variables folder. In this folder, one or more variable tables can be created and managed.

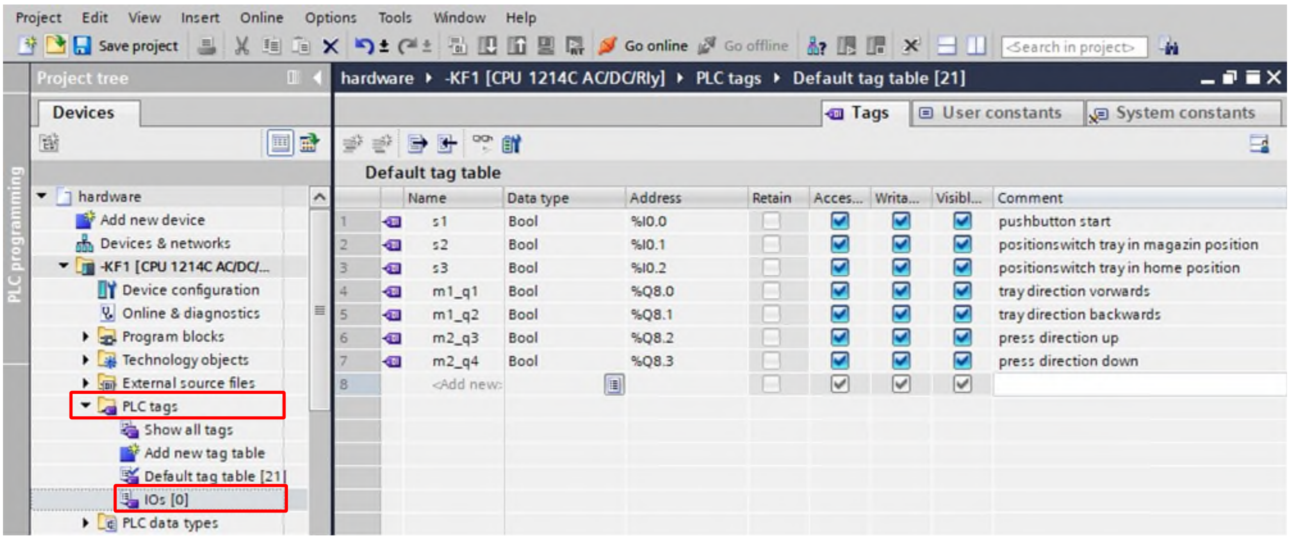


Figure 23 PLC variables in the TIA portal

Beckhoff / Codesys

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

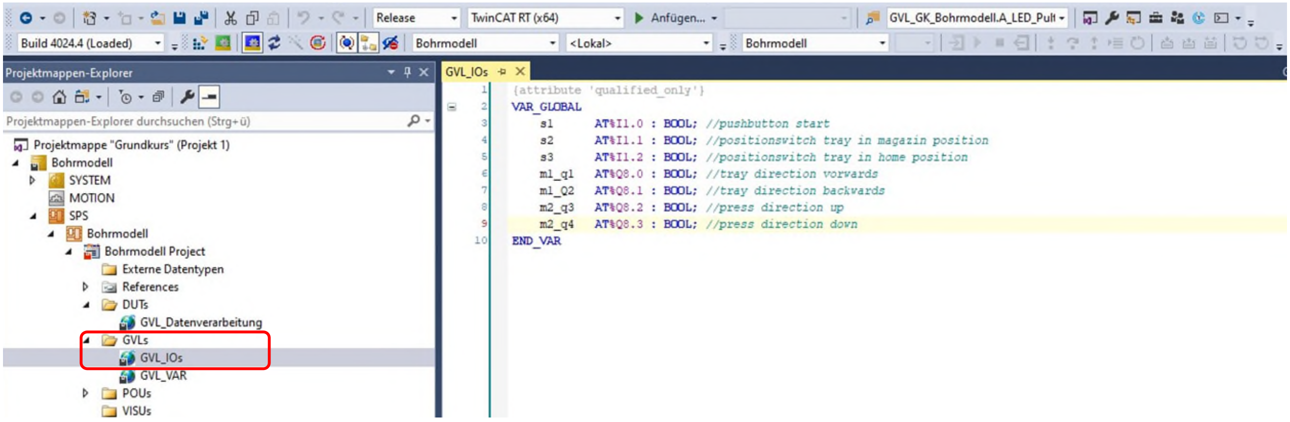


Figure 24 PLC variables in TwinCAT (Beckhoff)

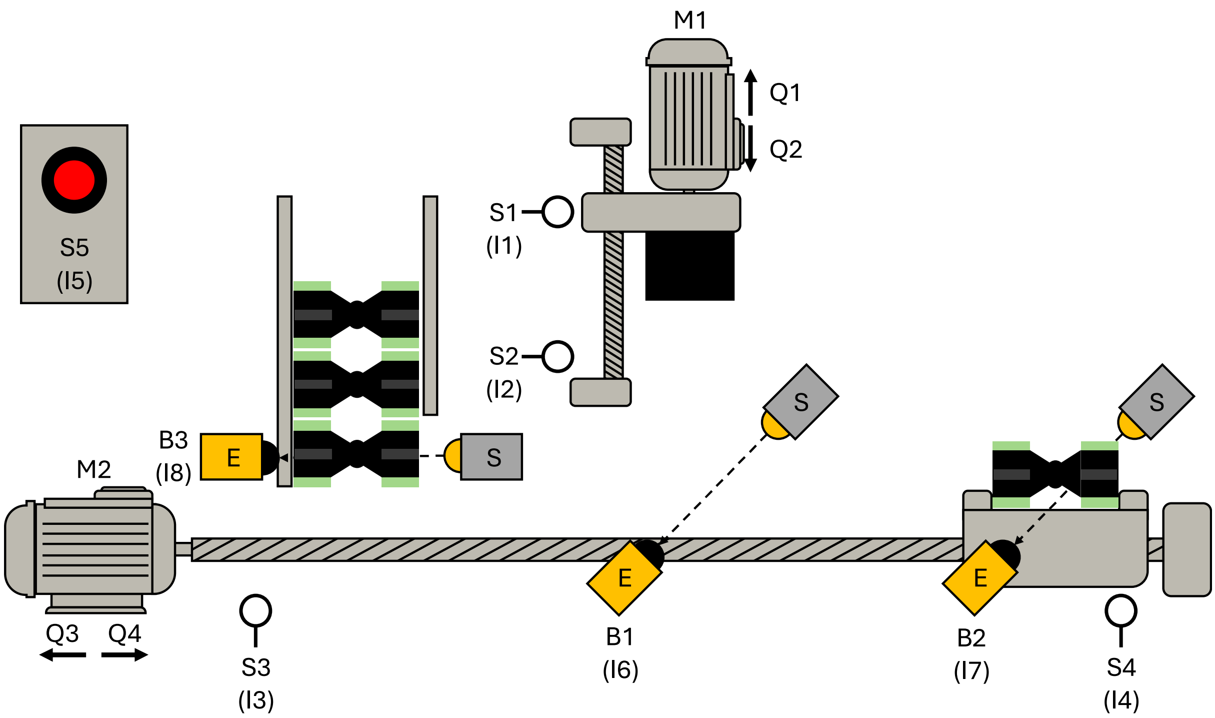
2.3.3 Exercise: Creating PLC Variables

Goal:

I can create and edit PLC variables.

Task:

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

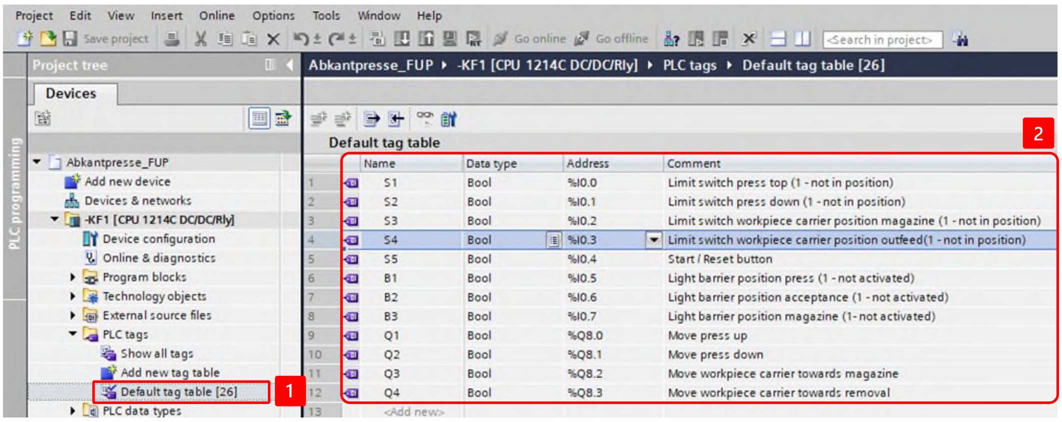


In a simple and quick assignment, the absolute and symbolic addresses of the variables can be entered in the allocation plan, from the chapter "Model", in the columns "Address" and "Symbol".

Procedure:

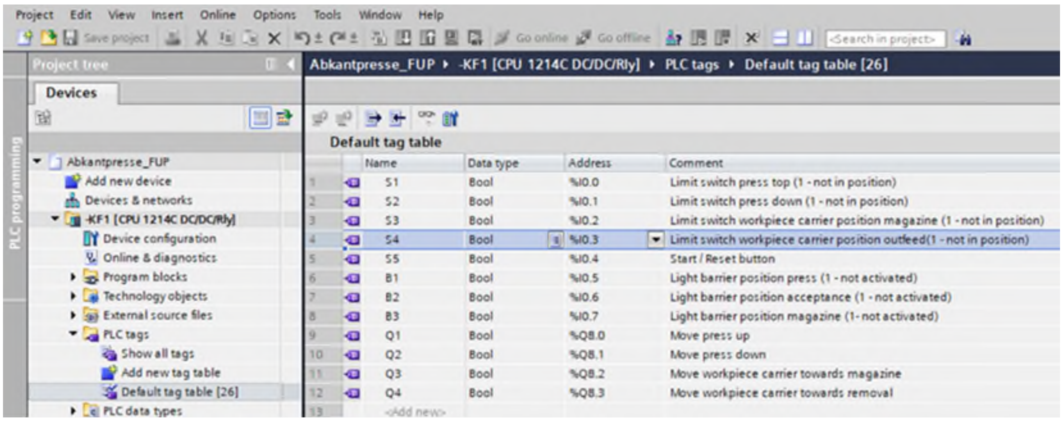
1. Use the "Add new variable table" button to add a new PLC variable table and assign a meaningful name. Or use an existing table (e.g. standard variable table).

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



Solution

Solution:



In the symbolic name, the equipment identifier was used. The addresses have been adjusted according to the hardware configuration:

- Inputs in bytes EB 0
- Outputs in bytes AB 8