

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

Siemens Automation Cooperates with Education

TIA Portal Module 020-011 Startup Programming with SIMATIC S7-1500

Cooperates
with Education

Automation

SIEMENS

Matching SCE training packages for these training curriculums

SIMATIC Controllers

- **SIMATIC S7-1500F with CPU 1516F-3 PN/DP**
Order number: 6ES7516-3FN00-4AB1

SIMATIC STEP 7 Software for Training

- **SIMATIC STEP 7 Professional V13 – Single license**
Order number: 6ES7822-1AA03-4YA5
- **SIMATIC STEP 7 Professional V13 – block of 12 class room license**
Order number: 6ES7822-1BA03-4YA5
- **SIMATIC STEP 7 Professional V13 – block of 12 upgrade license**
Order number: 6ES7822-1AA03-4YE5
- **SIMATIC STEP 7 Professional V13 – block of 20 student license**
Order number: 6ES7822-1AC03-4YA5

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

Multiple information for S7-1500

Such as Getting started, videos, tutorials, manuals and programming guidelines.
[siemens.com/sce/S7-1500](https://www.siemens.com/sce/S7-1500)

Continued Training

For regional Siemens SCE continued training, please contact your regional SCE contact person
[siemens.com/sce/contact](https://www.siemens.com/sce/contact)

Additional information regarding SCE

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

Information regarding Usage

This SCE 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 for public education facilities and R&D facilities. Siemens AG does not guarantee the contents.

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

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

Offenders will be held liable. All rights including translation are reserved, particularly if a patent is granted or a utility model or design is registered.

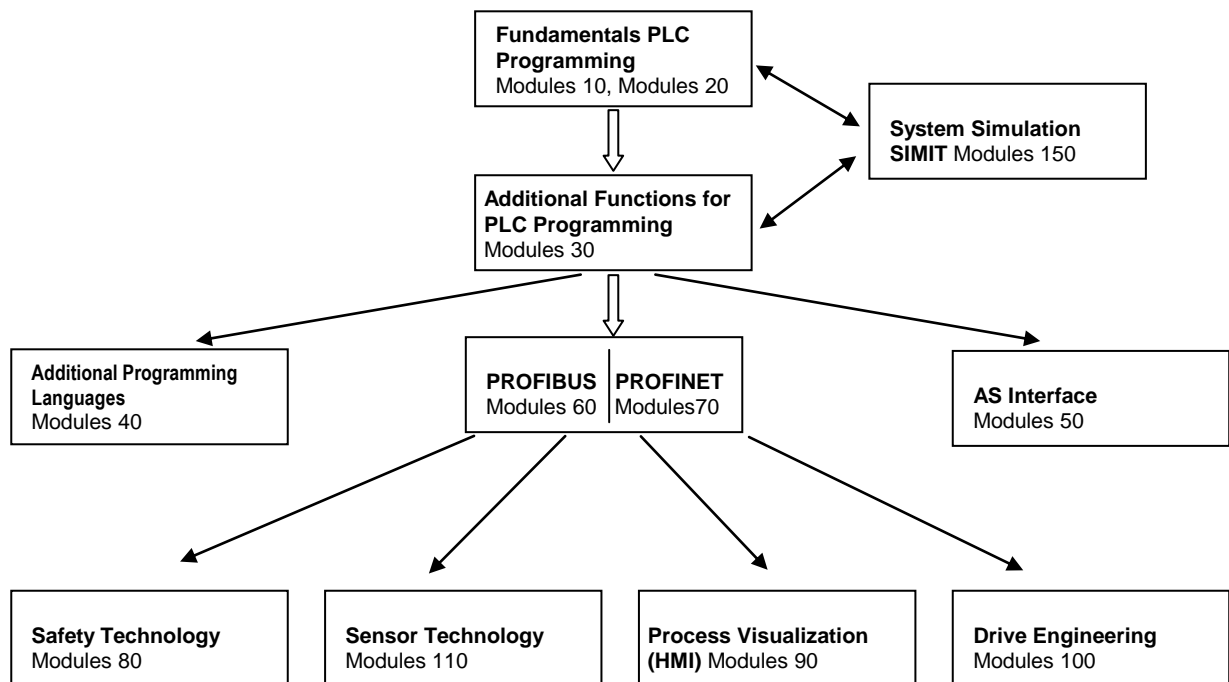
Usage for industrial customer courses is explicitly not permitted. We do not consent to the training curriculums being used commercially.

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

	PAGE
1. Preface	4
2. Programming Information for SIMATIC S7-1500	6
2.1 Automation System SIMATIC S7-1500	6
2.2 Programming Software STEP 7 Professional V13 (TIA Portal V13)	7
3. Installing the Software STEP 7 Professional V13 (TIA Portal V13).....	8
4. Connecting to the CPU via TCP/IP and Resetting to Factory Setting	9
5. What is a PLC and what are PLCs used for?	17
5.1 What does the term PLC mean?	17
5.2 How does the PLC control the process?	17
5.3 Where does the PLC obtain the information regarding process states?	18
5.4 What is the difference between break (NC) and make (NO) contacts?	18
5.5 How does the SIMATIC S7-1500 address individual input/output signals?	19
5.6 How is the program processed in the PLC?	20
5.7 What do logic operations look like in the PLC program?	21
5.7.1 AND Logic Operation	21
5.7.2 OR Logic Operation	23
5.7.3 Negation.....	24
5.8 How is the program generated? How does it get to the PLC memory?	25
6. Configuring and Operating the SIMATIC S7-1500	26
6.1 Module Spectrum.....	26
6.1.1 Sample Configuration	29
6.2 Operating and Display Elements of the CPU 1516-3 PN/DP	30
6.2.1 Frontal View of CPU 1516-3 PN/DP with Integrated Display	30
6.2.2 Status and Error Indicators	30
6.2.3 Operating and Connection Elements of the CPU 1516-3 PN/DP behind the front cover	31
6.2.4 SIMATIC Memory Card.....	32
6.2.5 Operating Mode Switch.....	32
6.2.6 Display of the CPU	33
6.3 Storage Areas of the CPU 1516-3 PN/DP and the SIMATIC Memory Card	35
7. Sample Task Press Control	37
8. Programming the Press with SIMATIC S7-1500	38
8.1. Portal View.....	38
8.2. Project View	39

1. Preface

Regarding its content, module SCE_DE_020-011 is part of the training unit '**Fundamentals of PLC Programming**' and describes the **fast entry** into programming the SIMATIC S7-1500 with the TIA Portal.



Training Objective

The reader will learn how to program the programmable logic controller (PLC) SIMATIC S7-1500 by using the programming tool TIA Portal. The training module provides the fundamentals and, with the steps below, shows how this is accomplished based on a detailed example.

- Installing the software and setting up the programming interface
- Explanation what a PLC is, and how it works
- Structure and operation of the PLC SIMATIC S7-1500
- Creating, loading and testing a sample program

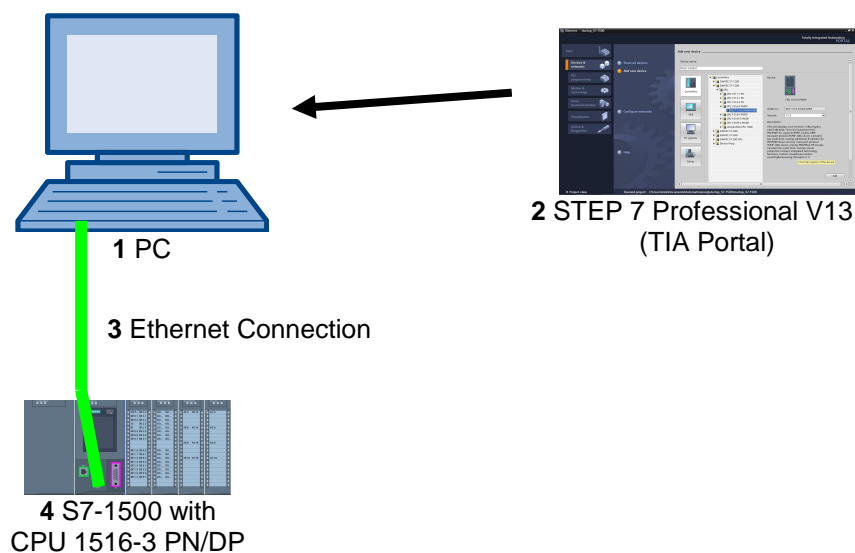
Prerequisites

The following knowledge is assumed to successfully work through this training module:

- Knowledge in handling Windows

Hardware and Software required

- 1 PC Intel® Celeron® Dual Core 2,2 GHz, 1.7 GHz 4 GB RAM, free disk storage approx. 5GB
Operating systems: Windows 7 (32/64Bit) Professional SP1/ Enterprise SP1 / Ultimate SP1,
Windows 8.1 (64Bit) Professional / Enterprise, Windows Server 2008 (64Bit) R2 StdE SP1
(full installation), Windows Server 2012 (64Bit) R2 StdE (full installation)
- 2 Software STEP 7 Professional V13 (Totally Integrated Automation (TIA Portal V13)
- 3 Ethernet connection between PC and CPU 1516-3 PN/DP
- 4 PLC SIMATIC S7-1500; for example, CPU 1516-3 PN/DP with signal modules for digital inputs
(DI) and digital outputs (DO).The inputs have to be brought to a panel.



2. Programming Information for SIMATIC S7-1500

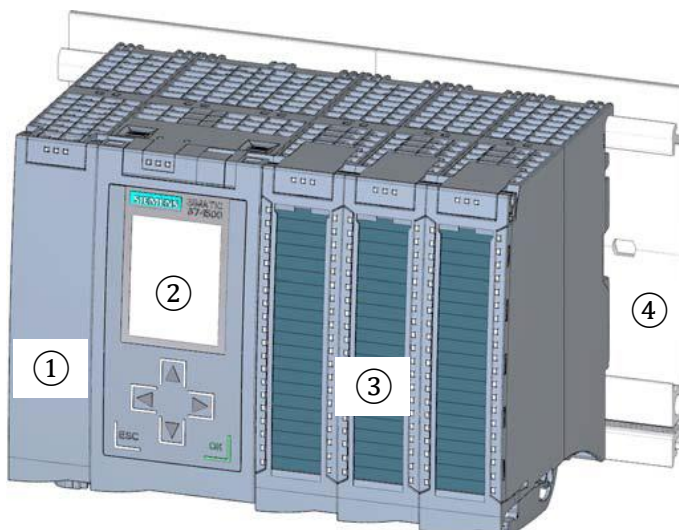
2.1 Automation System SIMATIC S7-1500

The automation system SIMATIC S7-1500 is a modular control system for the medium and upper performance range. An extensive module spectrum is available for optimum adaptation to the automation task.

The SIMATIC S7-1500 is the continued development of the SIMATIC S7-300 and S7-400 automation systems with the following performance features:

- Increased system performance
- Integrated motion control functionality
- PROFINET IO IRT
- Integrated display for machine-related operation and diagnosis
- STEP 7 language innovations while retaining proven functions

The S7-1500 controller consists of a power supply ①, a CPU with integrated display ②, and input/output modules for digital and analog signals ③. The modules are mounted on a channel with integrated mounting rail profile ④. If needed, communication processors and function modules for special tasks - such as stepper motor control- are used.



With the S7 program, the programmable logic controller (PLC) monitors and controls a machine or a process: the IO modules are polled in the S7 program by means of input addresses (%I), and output addresses (%O) are addressed.

The system is programmed with the software STEP 7 Professional V13.

2.2 Programming Software STEP 7 Professional V13 (TIA Portal V13)

The software STEP 7 Professional V13 (TIA Portal V13) is the programming tool for the following automation systems:

- SIMATIC S7-1500
- SIMATIC S7-1200
- SIMATIC S7-300
- SIMATIC S7-400
- SIMATIC WinAC

With STEP 7 Professional V13, the following functions can be utilized for automating a system:

- Configuring and parameterizing the hardware
- Specifying communication
- Programming
- Test, commissioning and service with the operational/diagnostic functions
- Documentation
- Creating visualization for SIMATIC basic panels with integrated WinCC Basic.
- With additional WinCC packages, visualization solutions for PCs and other panels can be created.

All functions are supported with detailed Online Help.

3. Installing the Software STEP 7 Professional V13 (TIA Portal V13)

STEP 7 Professional is provided on a DVD.

To install STEP 7 Professional, do the following:

1. Place the STEP 7 Professional DVD in the DVD drive.
2. The setup program starts automatically. If it doesn't, start it by double clicking on the file '**→ START.exe**'.
3. The setup program takes you through the entire installation process of STEP 7 Professional
4. To utilize STEP 7 Professional, a license key is required on your computer.

In the course of the installation, this license key can be transferred from an included USB stick to your computer. With the software '**Automation License Manager**' this license key can then be moved to other data carriers. This license key may also be located on a different computer and be requested by means of a network.

Note:



The provided license key '**STEP 7 Professional Combo**' includes the simultaneous activation of the software STEP 7 V5.5.

4. Connecting to the CPU via TCP/IP and Resetting to Factory Setting

To program the SIMATIC S7-1500 from the PC, the PG or a laptop, a TCP/IP connection or optionally a PROFIBUS connection is needed.

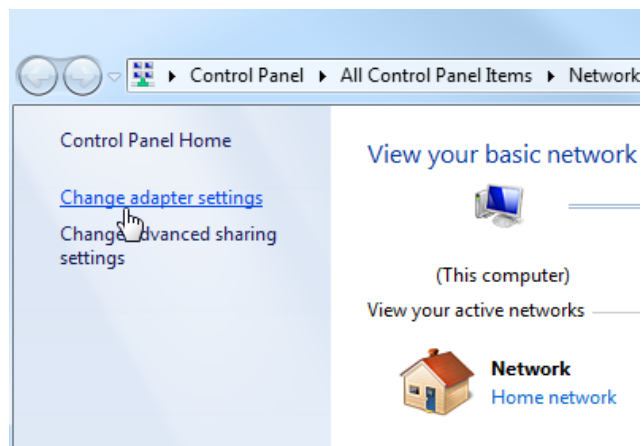
For the PC and the SIMATIC S7-1500 to communicate by means of TCP/IP, it is important that the IP addresses on both devices match.

First, we are setting the computer's IP address using the Windows 7 operating system.

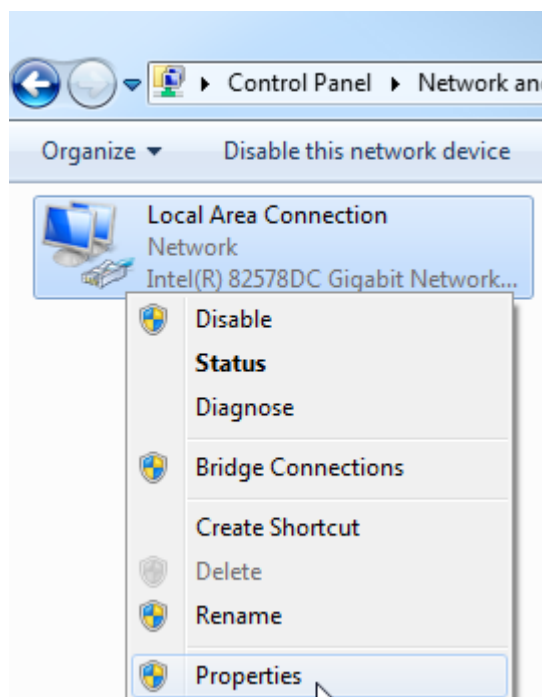
1. Locate the network symbol below in the task bar  and then click on '**Open Network and Sharing Center**'. (→  → Open Network and Sharing Center)



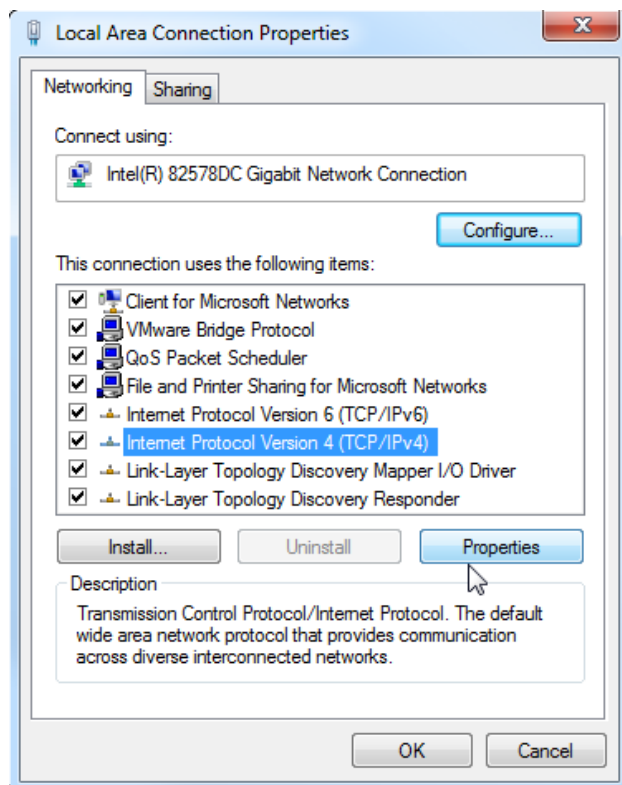
2. In the opened window of the Network and Sharing Center, click on '**Change Adapter Settings**'. (→ Change Adapter Settings)



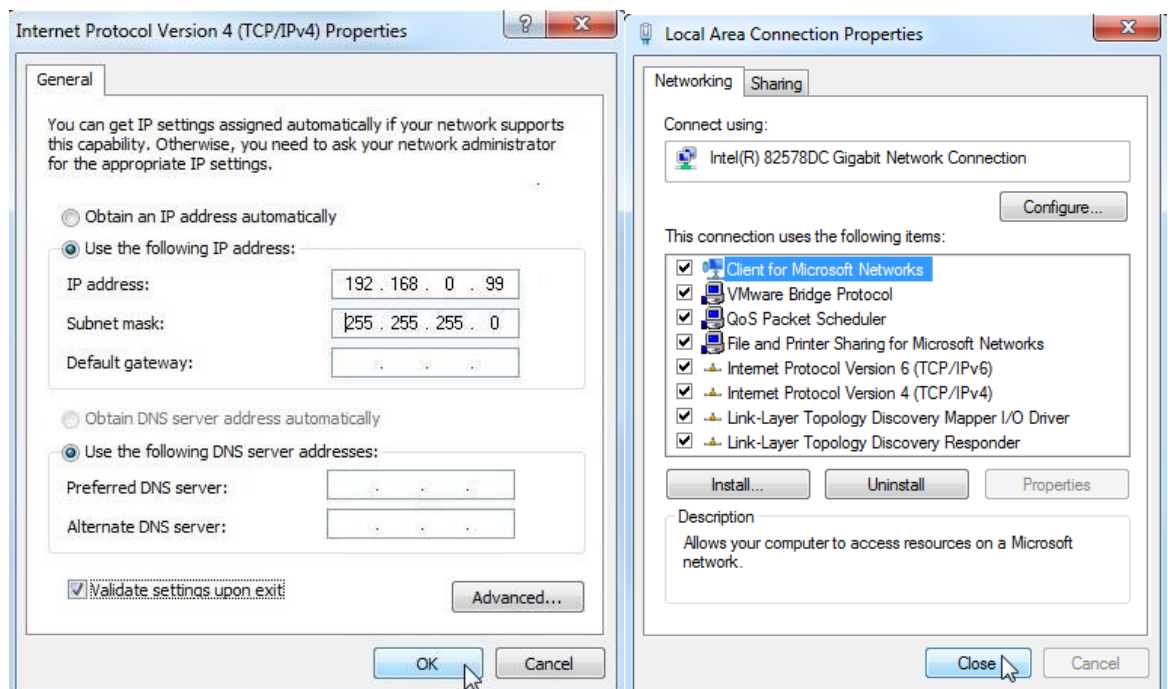
3. Select the '**Local Area Connection**' you want to use for establishing a connection with the controller and then click on '**Properties**'. (Local Area Connection → Properties)



4. Now, select the '**Properties**' for '**Internet Protocol Version 4 (TCP/IPv4)**'
(→ Internet Protocol Version 4 (TCP/IP) → Properties



5. You can now set the '**IP address**' and the '**Subnet mask**' and accept with '**OK**'. (→ Use the following IP address → IP address: 192.168.0.99 → Subnet mask 255.255.255.0 → OK → Close)



Information about networking on the Ethernet (additional information in Attachment V of the training curriculum):**MAC Address:**

The MAC address consists of a fixed and a variable part. The fixed part ("Basis MAC Address") indicates the manufacturer (Siemens, 3COM, ...). The variable part of the MAC address differentiates the various Ethernet stations and should be unique globally. A MAC address assigned by the factory is imprinted on each module.

Value range for the IP address:

The IP address consists of four decimal numbers in the value range from 0 to 255; they are separated from each other by a period: for example, 141.80.0.16

Value range for the subnet mask:

This mask is used for recognizing whether a station -or its IP address- belongs to the local subnet or whether it can be accessed only by means of a router.

The subnet mask consists of 4 decimal numbers in the value range from 0 to 255; they are separated from each other with a period; for example, 255.255.0.0

In their binary representation, the 4 decimal numbers of the subnet mask have to contain from the left a sequence of gapless values "1" and from the right a sequence of gapless values "0".

The values "1" determine the IP address area for the network number. The values "0" determine the IP address area for the station address.

Example:

Correct values:	255.255.0.0 decimal	= 1111 1111.1111 1111.0000 0000.0000 0000 binary
	255.255.128.0 decimal	= 1111 1111.1111 1111.1000 0000.0000 0000 binary
	255.254.0.0 decimal	= 1111 1111.1111 1110.0000 0000.0000 0000 binary
Wrong value:	255.255.1.0 decimal	= 1111 1111.1111 1111.0000 0001.0000 0000 binary

Value range for the gateway address (router):

The address consists of 4 decimal numbers in the value range from 0 to 255, separated from each other by a period; for example, 141.80.0.1.

Relationship IP addresses, address of the router and subnet mask:

The IP address and the router address are allowed to differ only at those positions where the subnet mask is "0".

Example:

You entered the following: for the subnet mask 255.255.255.0; for the IP address 141.30.0.5 and for the router address 141.30.128.1.

The IP address and the router address must have a different value only in the 4th decimal number. In the example, the third position already differs.

In the example, the following change has to be made alternatively:

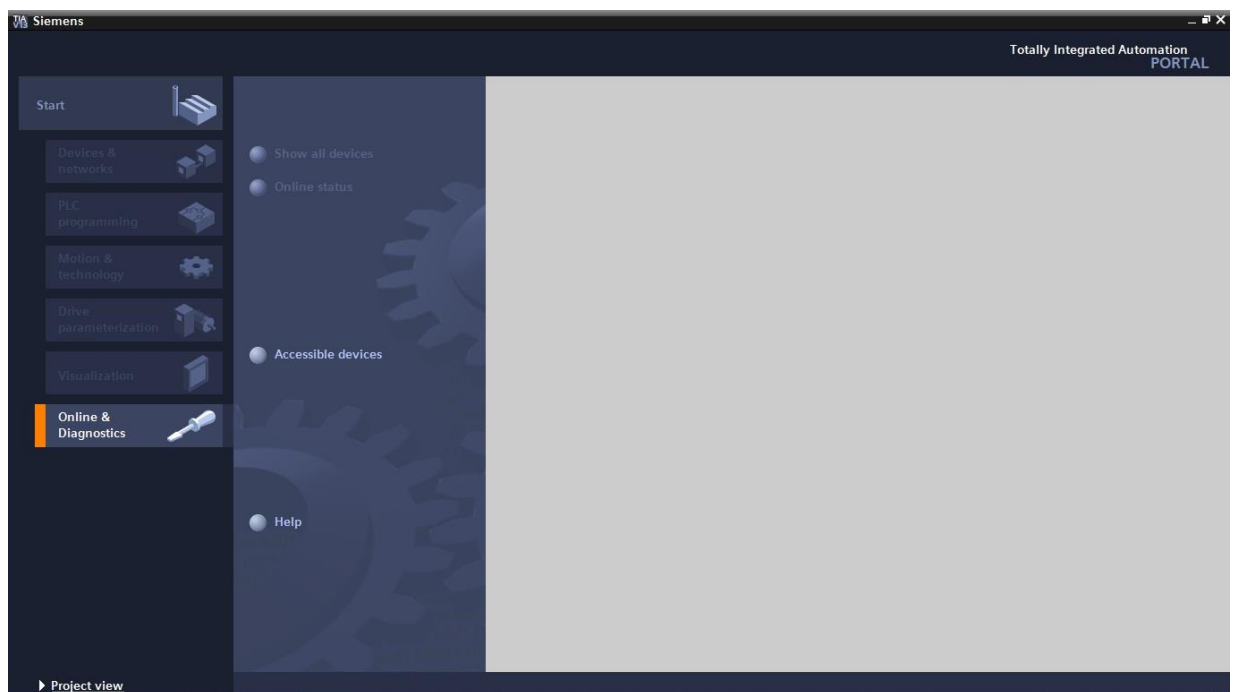
- the subnet mask to: 255.255.0.0 or
- the IP address to: 141.30.128.5 or
- the router address to: 141.30.0.1

The IP address of the SIMATIC S7-1500 is set as follows.

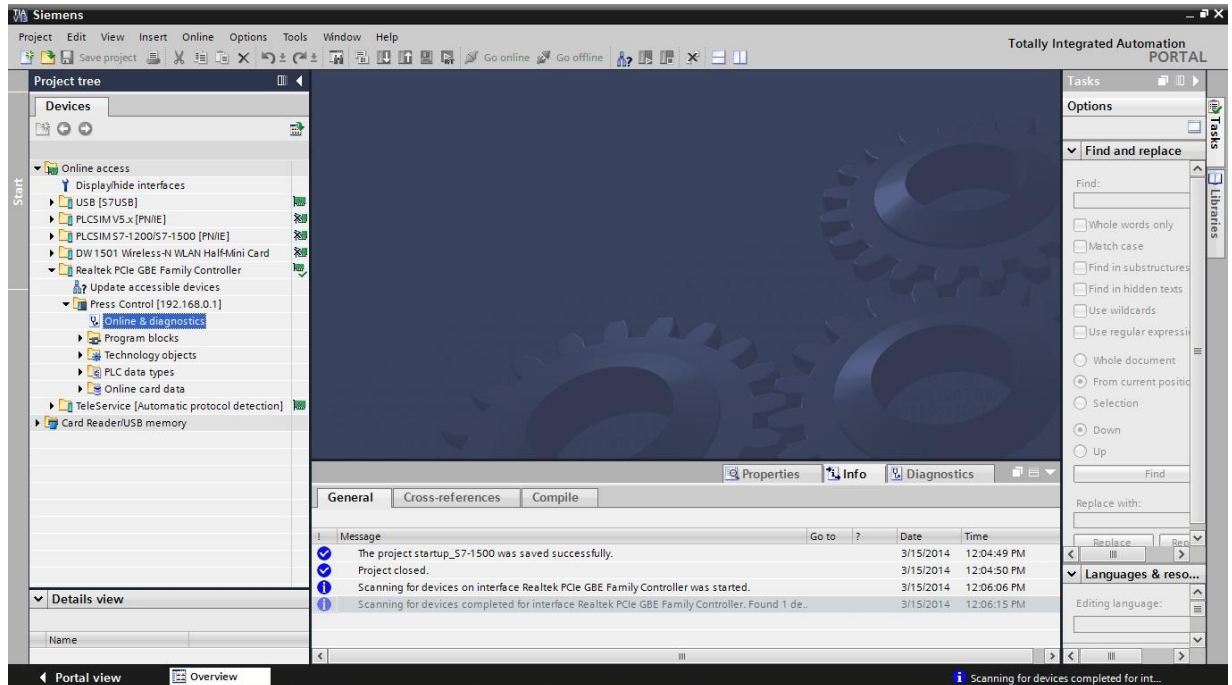
6. Select '**Totally Integrated Automation Portal**'; it is called here with a double click.
(→ TIA Portal V13)



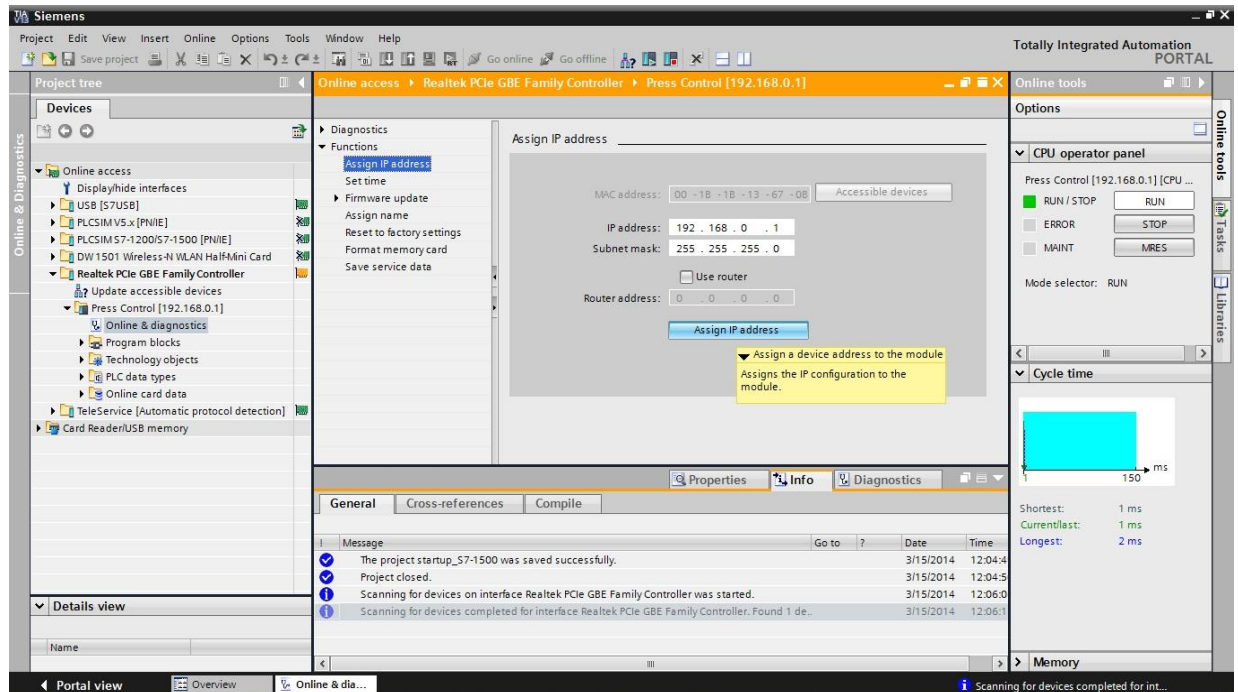
7. Select '**Online & Diagnostics**' and then open the '**Project view**'.
(→ Online & Diagnostics → Project view)



8. In the project tree under '**Online access**', select the network card that was set previously. If you click here on '**Update accessible devices**' you will now see the IP address (provided it is set) or the MAC address (if the IP address is not yet assigned) of the connected SIMATIC S7-1500. Here, select '**Online & Diagnostics**'. (→ Online access → ... Network Connection → Update accessible devices → → Online & Diagnostics)



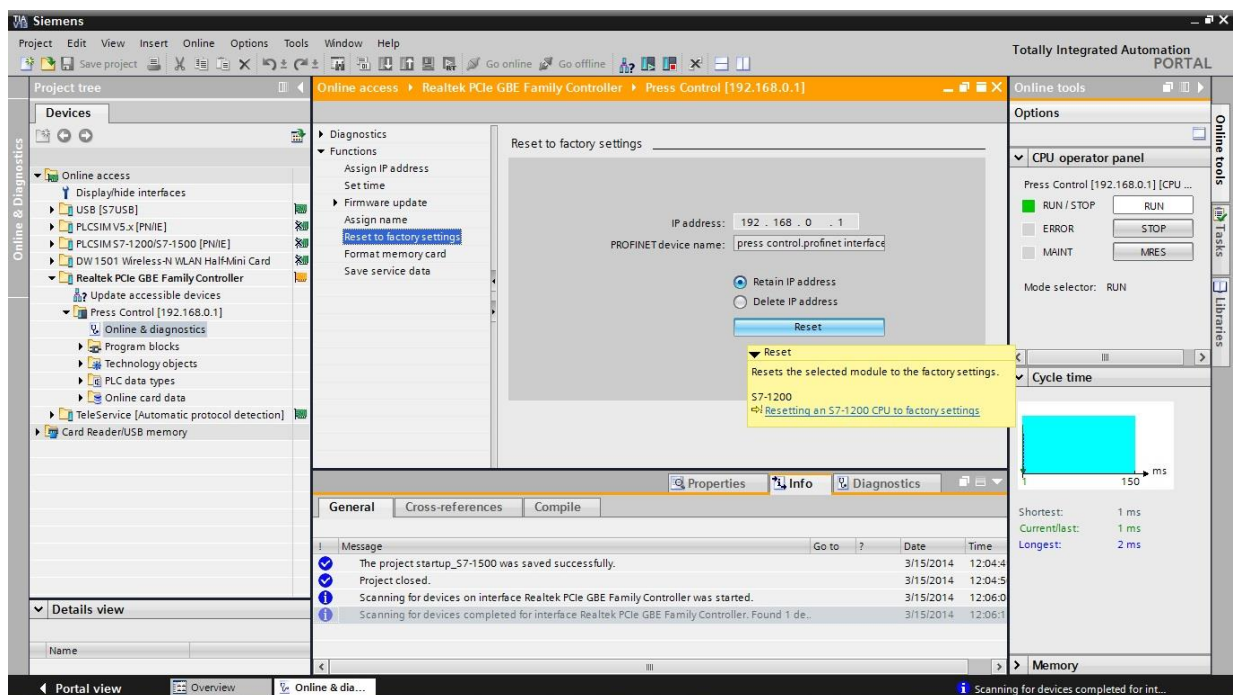
- Under **'Functions'**, the option **'Assign IP address'** is located. Here, enter the **'IP address'** and **'Subnet mask'**. Now, click on **'Assign IP address'**; this new address is then assigned to your SIMATIC S7-1500. (→ Functions → Assign IP address → IP address: 192.168.0.1 → Subnet mask: 255.255.255.0 → Assign IP address)



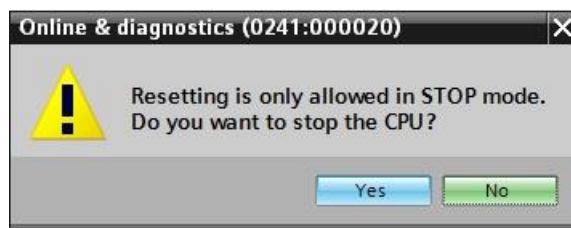
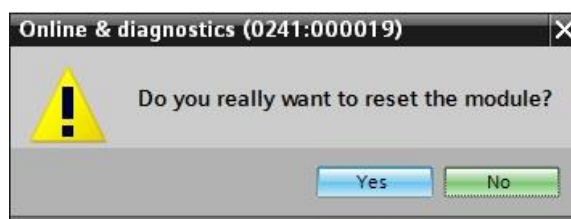
Note:

The SIMATIC S7-1500 IP address can also be set by means of the display at the CPU.

10. If there should be problems with the acceptance of the IP address, or if you want to reset the controller, select '**Functions**', '**Reset to factory settings**', and then click on '**Reset**'.
(→ Functions → Reset to factory settings → Reset)



11. Confirm the question 'Do you really want to reset the module?' with '**OK**'; if necessary, stop the CPU. (→ OK → Yes)



Note:

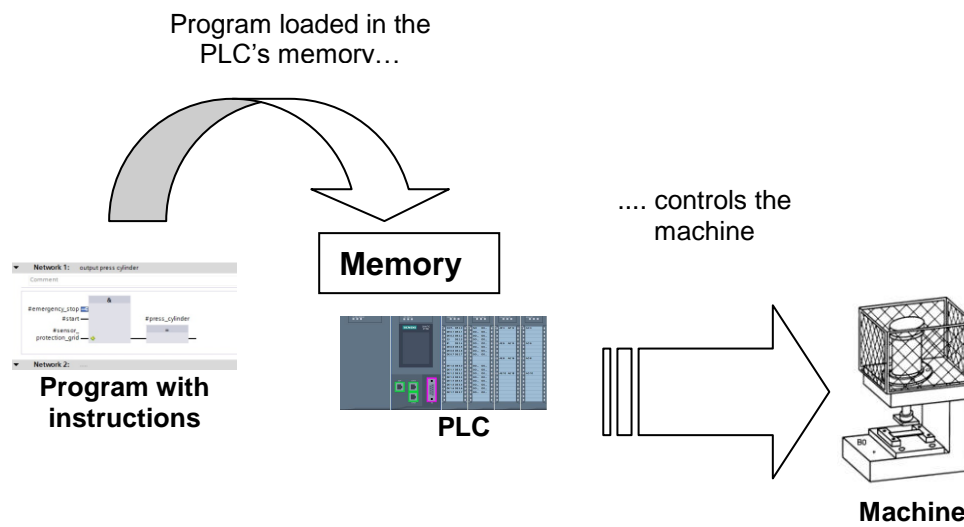
A reset to the factory setting is possible also by means of the display or of the operating mode switch.

5. What is a PLC and what are PLCs used for?

5.1 What does the term PLC mean?

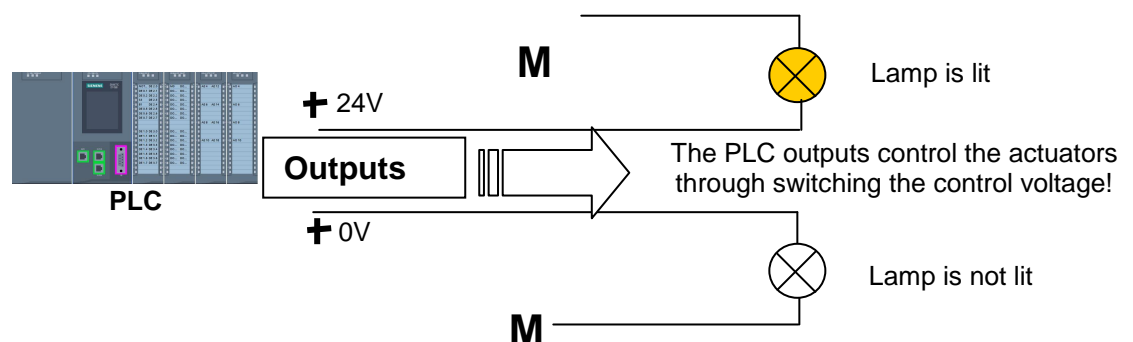
PLC is the abbreviation for **P**rogrammable **L**ogic **C**ontroller. This describes a device that controls a process (for example, a printing press for printing newspapers, a filling system for filling cement into bags, a press for pressing plastic parts, etc.).

These actions are carried out corresponding to the instructions of a program that is located in the device's memory.



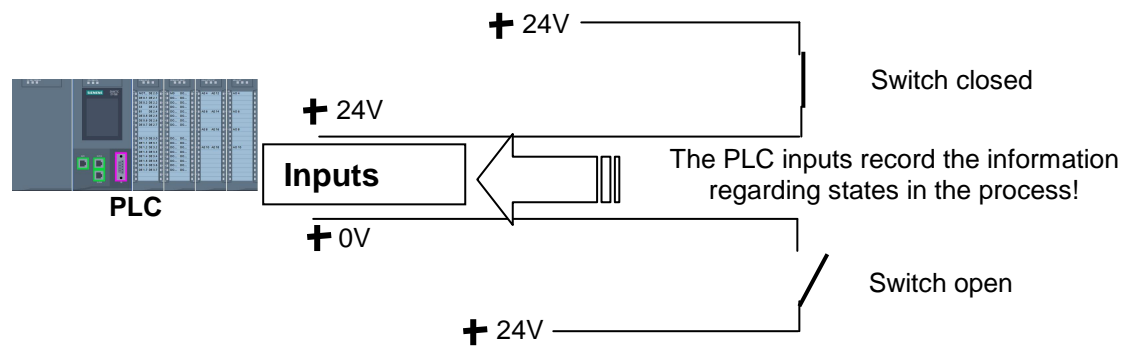
5.2 How does the PLC control the process?

The PLC controls the process as follows: so-called **actuators** are wired with a control voltage of -for example- 24V by means of PLC connections called **outputs**. This allows for motors being switched on and off, valves opened and closed, and lamps switched on and off.



5.3 Where does the PLC obtain the information regarding process states?

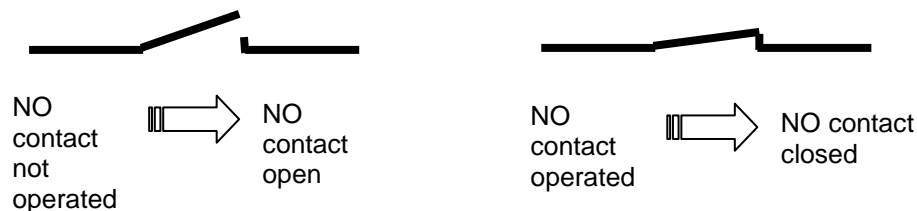
The PLC receives information regarding the process from the so-called **signal transmitters** that are connected to the PLC's **inputs**. These signal transmitters can be sensors that detect whether a work piece is located at a certain position, or simple switches and buttons that can be open or closed. Here, we differentiate between **NC contacts** (normally closed contacts) that are closed without being operated, and **NO contacts** (normally open contacts) that are open without being operated.



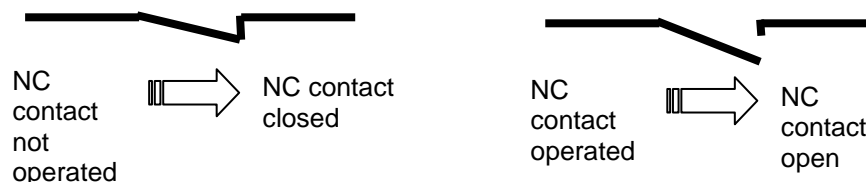
5.4 What is the difference between break (NC) and make (NO) contacts?

Regarding signal transmitters, we differentiate between **NC contacts** and **NO contacts**.

The switch shown here is a **normally open contact** (make contact); i.e., it is closed if it was operated.



The switch shown here is a **normally closed contact** (break contact); i.e., it is closed if it was not operated.



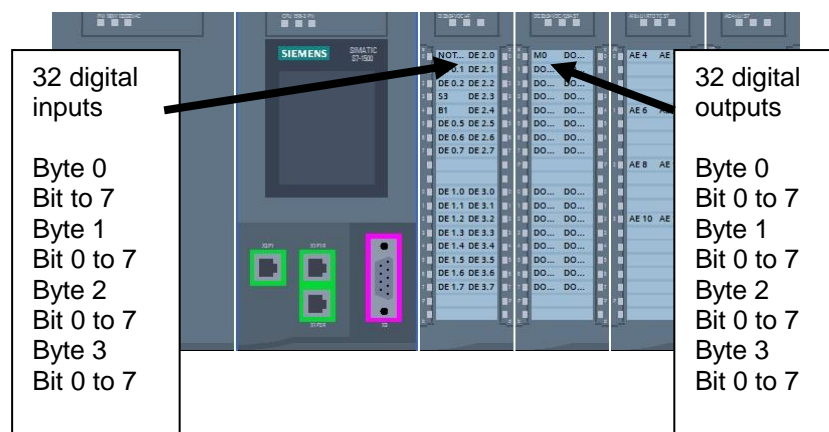
5.5 How does the SIMATIC S7-1500 address individual input/output signals?

Specifying a certain input or output within the program is called 'addressing'.

The inputs and outputs of the PLCs are usually combined into groups of eight for digital input and digital output modules. This group of eight is called **byte**. Each such group is assigned a number as the so-called **byte address**.

To address a single input or output within a byte, each byte is divided into individual **bits**. These are numbered starting with Bit 0 to Bit 7. This is how we get the **bit address**.

The PLC shown here now has a signal module with the input bytes 0 to 3 as well as the output bytes 0 to 3.



To address the fifth digital input, for example, we enter the following address:

%I 0 . 4

%I identifies here the address type as input, **0** the byte address and **4** the bit address. The byte address and the bit address are always separated by a period.

Note:

For the bit address, a **4** is specified at the 5th input because counting starts with 0.

To address the 10th output, we specify the following address:

%O 1 . 1

%O identifies here the address type as output, **1** the byte address and **1** the bit address. The byte address and the bit address are always separated by a period.

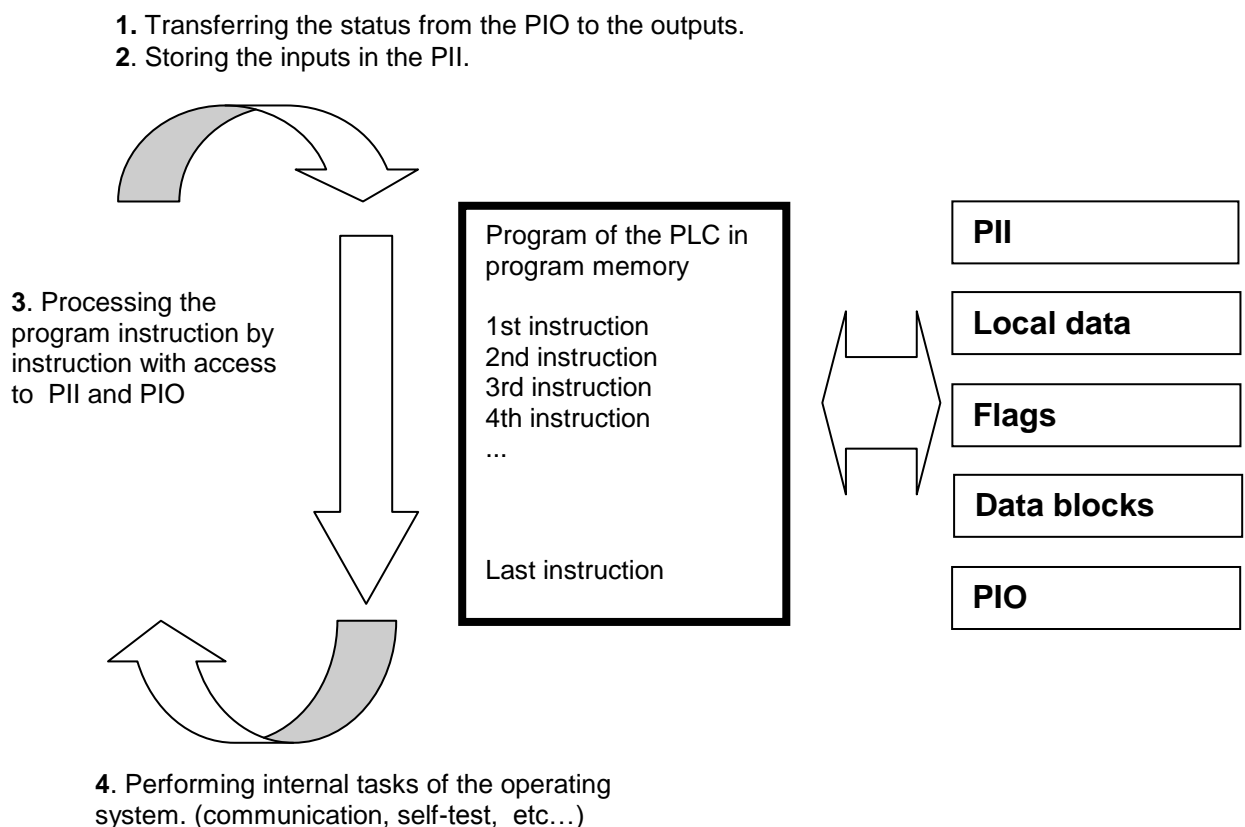
Note:

For the bit address, a **1** is specified at the 10th output, because counting starts with 0.

5.6 How is the program processed in the PLC?

Programs are processed in a PLC cyclically in the following sequence:

1. First, the status is transferred from the process image of the outputs (**PIO**) to the outputs; these are then switched on or switched off.
2. Next the **processor** (which is practically the PLC's brain) polls whether the individual inputs carry voltage. The status of the inputs is stored in the process image of the inputs (**PII**). For the inputs that carry voltage the information 1 or 'high' is stored, for those that do not carry voltage the information 0 or 'low' is stored..
3. This processor processes the program stored in the program memory. The program consists of a list of logic operations and instructions that are processed one after the other. For the required input information the PII entered previously is accessed, and the results of the logic operation are written to the process image of the outputs (**PIO**). If required, the processor also accesses other memory areas; for example, for local data of subprograms, data blocks and flags.
4. Finally, internal tasks of the operating system such as self-test and communication are carried out. Then the process continues with Item 1.



Note:

The time the processor needs for this sequence is called cycle time. It in turn depends on the number and type of instructions and the processor capability.

5.7 What do logic operations look like in the PLC program?

Logic operations are used to specify conditions for switching an output.

In the PLC program, these conditions can be generated with the programming languages ladder diagram (**LAD**) or function block diagram (**FBD**).

For reasons of clarity, we are using the **FBD**.

A large number of logic operations is available that can be used in PLC programs.

The **AND** as well as the **OR** logic operation and the **NEGATION** of an input are primarily used and will be briefly explained below using examples.

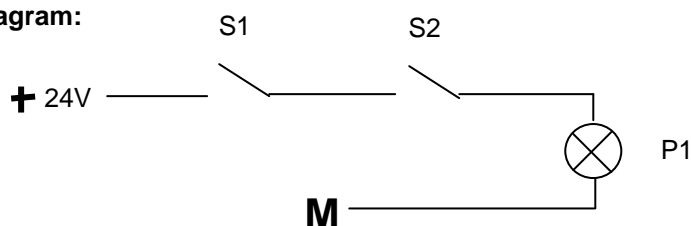
Note: Information regarding additional logic operations is available quickly and clearly arranged in Online Help.

5.7.1 AND Logic Operation

Example of an AND logic operation:

A lamp is to be lit when two switches are operated simultaneously as NO contact.

Circuit diagram:



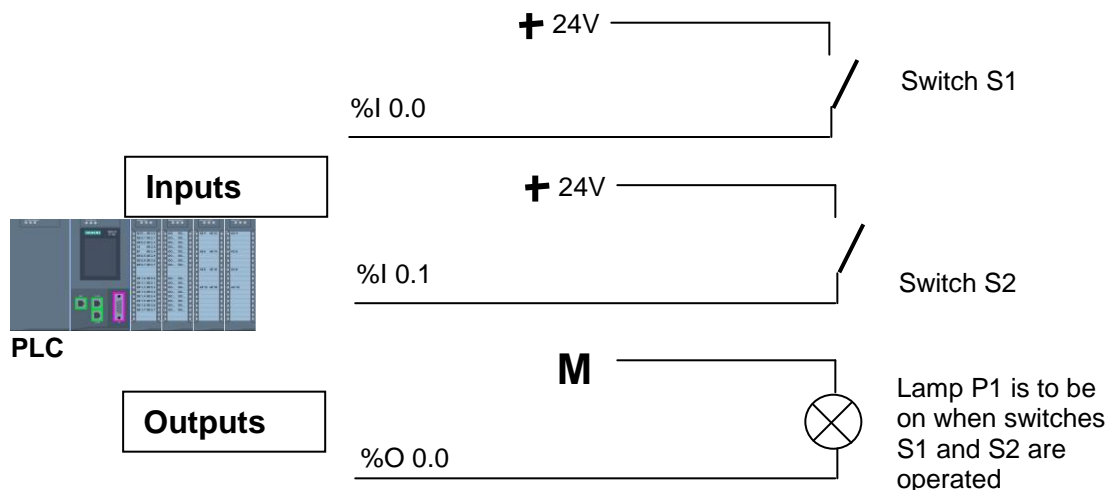
Explanation:

The lamp is lit exactly when both switches are operated; i.e., if switches S1 **and** S2 are operated, lamp P1 is lit.

Wiring the PLC:

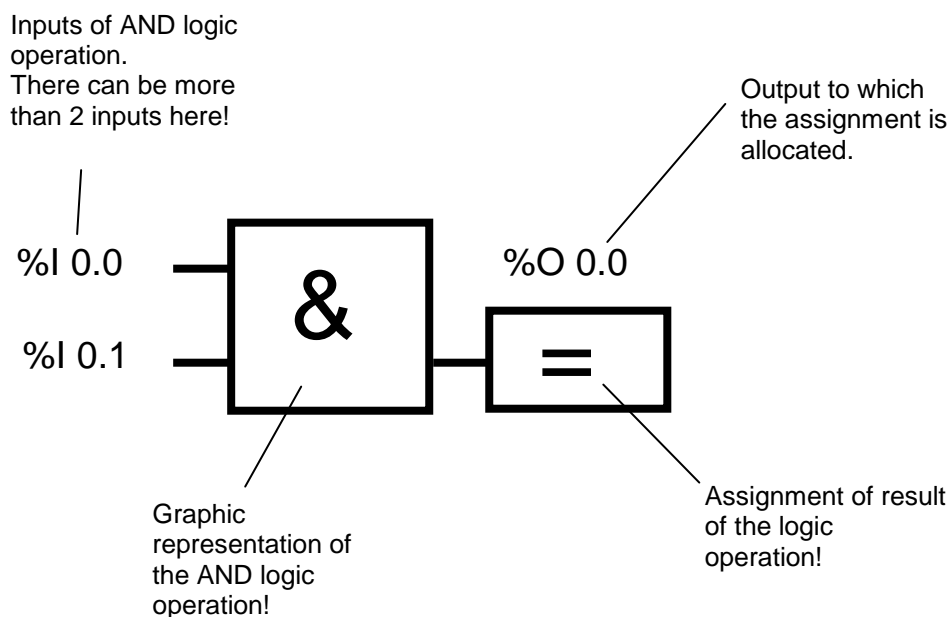
To implement this logic in a PLC program, both switches have to be connected to PLC inputs, of course. Here, S1 is wired to the input %I 0.0 and S2 to input %I 0.1.

In addition, lamp P1 has to be connected to an output, for example %O 0.0.



AND logic operation in FBD:

In FBD, the AND logic operation is programmed using graphic representation; it looks like this:

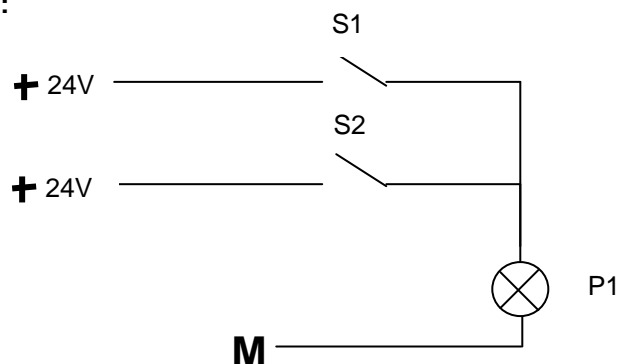


5.7.2 OR Logic Operation

Example of an OR logic operation:

A lamp is to be lit when one or both of two switches is operated as NO contact.

Circuit diagram:



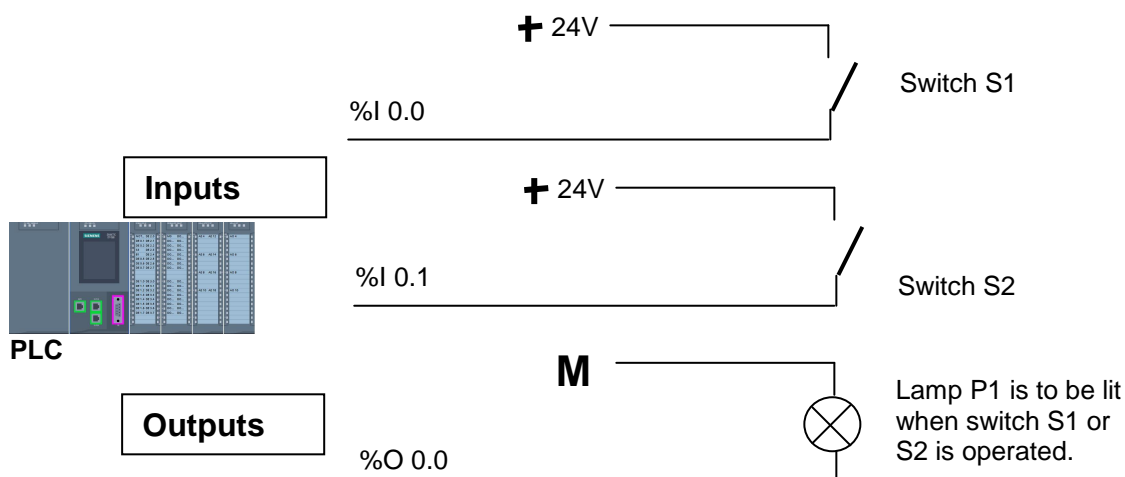
Explanation:

The lamp is lit exactly when one or both switches are operated; i.e., if switch S1 **or** S2 is operated, lamp P1 is on.

PLC wiring:

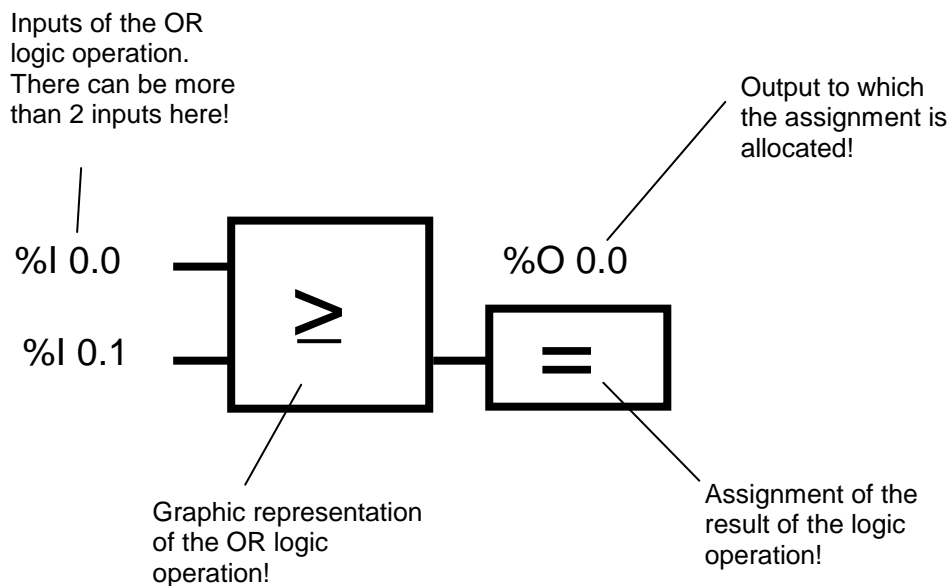
To implement this logic in a PLC program, both switches have to be connected to the inputs of the PLC, of course. Here, S1 is wired to input %I 0.0 and S2 to input %I 0.1.

In addition, lamp P1 has to be connected to an output; for example, %O 0.0.



OR logic operation in FBD:

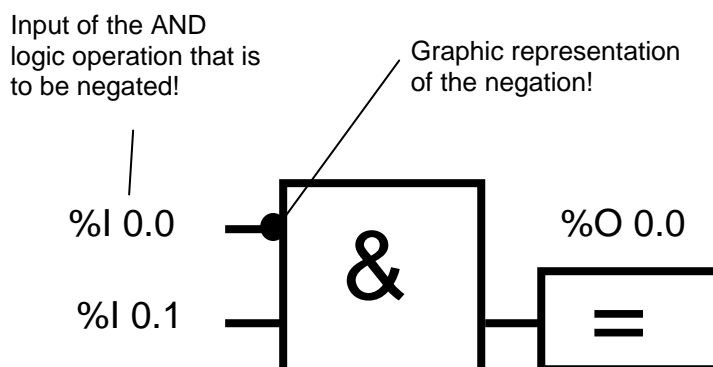
In FBD, the OR operation is programmed through graphic representation and looks like this:



5.7.3 Negation

In logic operations it is often necessary to poll whether a **NO contact** was **NOT operated** or whether a **NC contact** was **operated**, and there is no voltage at the corresponding input for that reason. . This is done by programming a **Negation** at the input of the AND or OR logic operation.

In FBD, the negation of an input is programmed at an AND logic operation with the following graphic representation:

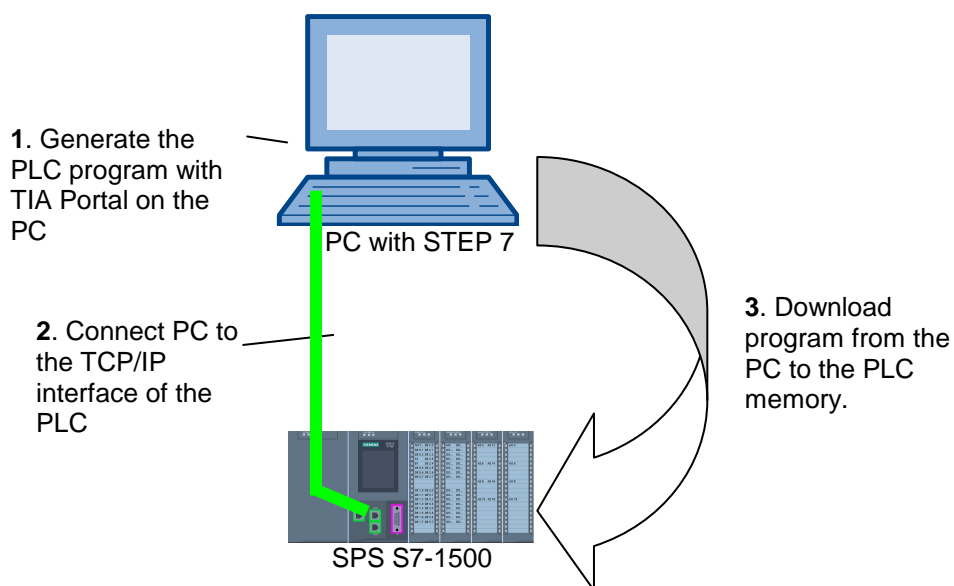


Voltage is applied to output %O 0.0 exactly when %I 0.0 is not wired and %I 0.1 is wired.

5.8 How is the program generated? How does it get to the PLC memory?

The PLC program is generated with the TIA Portal on a PC and stored there temporarily. After the PC is connected to the TCP/IP interface of the PLC, the program can then be downloaded to the PLC's memory with a load function.

The PC is now no longer needed for further program processing in the PLC.



Note: The exact sequence is described step by step in the chapters below.

6. Configuring and Operating the SIMATIC S7-1500

6.1 Module Spectrum

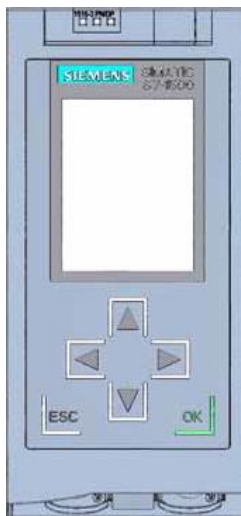
The SIMATIC S7-1500 is a modular automation system and offers the following module spectrum:

Central modules CPUs with integrated display

The CPUs execute the user program and have different performance capability. Furthermore, by means of the backplane the additional modules are supplied with the integrated system power supply.

Additional features and functions of the CPU:

- Communication via Ethernet
- Communication via PROFIBUS/PROFINET
- HMI communication for operator control and monitoring devices
- Web server
- Integrated technology functions (i.e. PID controllers, motion control...)
- System diagnosis
- Integrated security (for example, know-how protection, copying protection, access protection, integrity protection)



System power supply modules PS (rated input voltages 24V DC to 230VAC/DC)

With connection to the backplane provide the configured modules with the internal supply voltage.



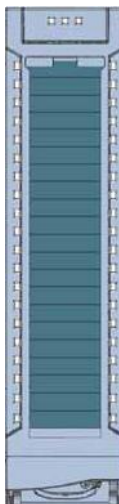
Load power supply modules PM (rated input voltages 120/230VAC)

Possess no connection to the backplane of the S7-1500 automation system. The following are supplied with the load power supply with DC24V: the system power supply of the CPU, input and output circuits of the IO modules, sensors and actuators.



IO modules

For digital input (DI)/digital output (DQ)/analog input (AI)/analog output (AQ)



Technology modules TM

As incremental encoders and pulse generators with/without direction level



Communication modules CM

For serial communication RS232 / RS422 / RS 485, PROFIBUS and PROFINET



SIMATIC Memory Card

Up to 2GByte maximum for storing the program data as well as simpler CPU exchange in maintenance cases.



6.1.1 Sample Configuration

The configuration of an automation system S7-1500 shown below is used in this training curriculum as a program example.

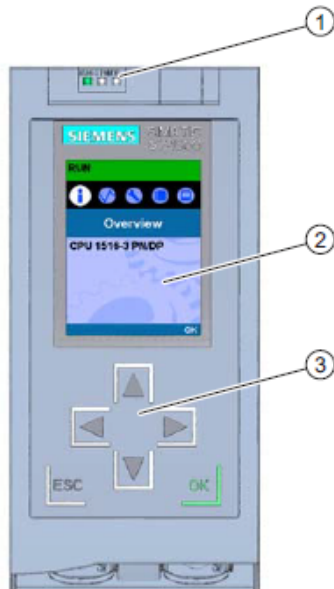


- ① Load power module PM with input 120/230VAC, 50Hz / 60Hz, 190W and output 24VDC / 8A
- ② Central module CPU 1516-3 PN/DP with integrated PROFIBUS and PROFINET interfaces
- ③ IO module 32x digital input DI 32x24VDC HF
- ④ IO module 32x digital output DQ 32x24VDC/0.5A ST
- ⑤ IO module 8x analog input AI 8xU/I/RTD/TC ST
- ⑥ IO module 4x analog output AQ 4xU/I ST

6.2 Operating and Display Elements of the CPU 1516-3 PN/DP

The figure below shows the operating and display elements of the CPU 1516-3 PN/DP. Arrangement and number of elements deviate at other CPUs from this picture.

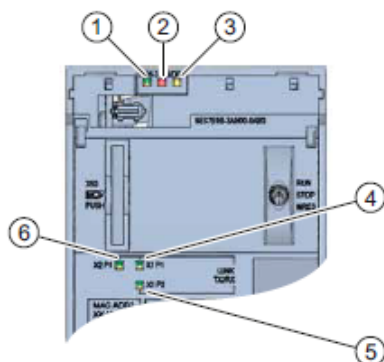
6.2.1 Frontal View of CPU 1516-3 PN/DP with Integrated Display



- ① LED displays for the current operating mode and diagnostic status of the CPU
- ② Display
- ③ Operator control buttons

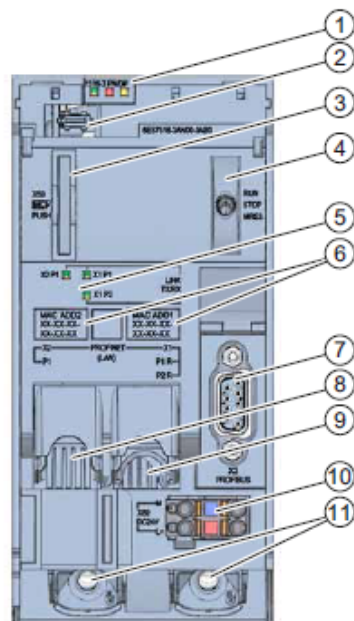
6.2.2 Status and Error Indicators

The CPU is equipped with the following LED indicators:



- ① RUN/STOP LED (yellow/green LED)
- ② ERROR LED (red LED)
- ③ MAINT LED (yellow LED)
- ④ LINK RX/TX-LED for X1 P1 port (yellow/green LED)
- ⑤ LINK RX/TX-LED for X1 P2 port (yellow/green LED)
- ⑥ LINK RX/TX-LED for X2 P1 port (yellow/green LED)

6.2.3 Operating and Connection Elements of the CPU 1516-3 PN/DP behind the front cover



- ① LED displays for the current operating mode and diagnostics status of the CPU
- ② Display connection
- ③ Slot for the SIMATIC memory card
- ④ Mode selector switch
- ⑤ LEDs for the 3 ports of the PROFINET interfaces X1 and X2
- ⑥ MAC addresses of the interfaces
- ⑦ PROFIBUS interface (X3)
- ⑧ PROFINET interface (X2) with 1 port
- ⑨ PROFINET interface (X1) with 2-port switch
- ⑩ Connection for power supply
- ⑪ Fixing screws

Note:

The front panel with the display can be removed/inserted during operation.

6.2.4 SIMATIC Memory Card

A SIMATIC Micro Memory Card is used as memory module. This card is a pre-formatted memory card compatible with the Windows file system. It is available in different memory sizes and can be used for the following purposes:

- Transportable data carrier
- Program card
- Firmware update card

To operate the CPU, the MMC **must** be inserted because the CPUs do not have an integrated load memory. A commercially available SD card reader is necessary to write to/read the SIMATIC memory card with the PG/PC. It is used, for example, to copy data with the Windows Explorer directly to the SIMATIC memory card.

Note:

We recommend inserting the SIMATIC memory card into the CPU or removing it only in the POWER OFF mode.

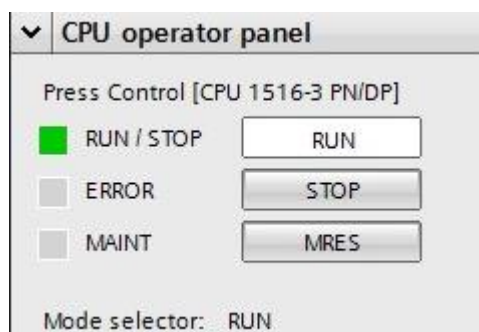
6.2.5 Operating Mode Switch

With the operating mode switch, we can set the CPU's current operating mode. The operating mode switch is designed as toggle switch with three switching positions:

Position	Meaning	Explanation
RUN	RUN mode	The CPU is executing the user program.
STOP	STOP mode	The CPU is not executing the user program.
MRES	Memory reset	Position for CPU memory reset.

Under Online & Diagnostics, the operating mode (**STOP** and **RUN**) can also be switched with the button on the CPU operator panel of the STEP 7 Professional V13 software.

In addition, the operator panel includes a button **MRES** for a general reset and displays the CPU's status LEDs.



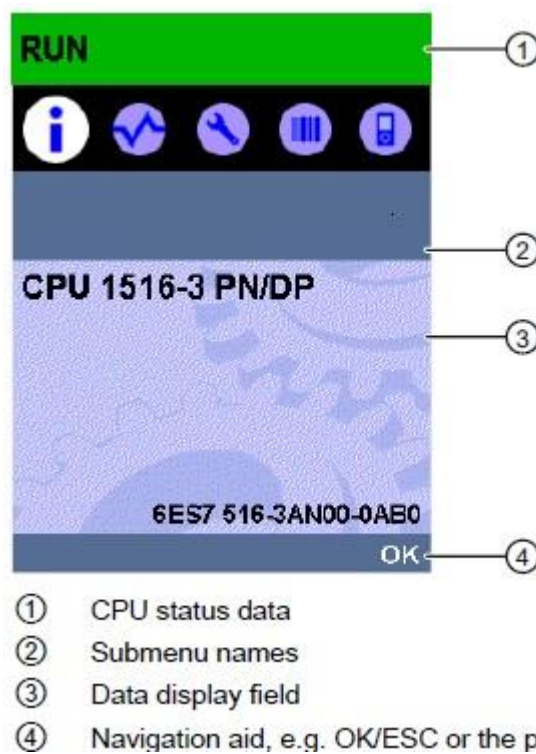
6.2.6 Display of the CPU

The S7-1500 CPU has a front panel with a display and operator buttons. On the display, control and status information can be displayed in different menus, and numerous settings can be made. We navigate through the menus with the operator buttons.






The display of the CPU provides the following functions:

- Six different display languages can be selected.
- Diagnosis indications are shown in plain text.
- The interface settings can be changed locally.
- A password for display operation can be assigned by means of TIA Portal

View of the display of a CPU 1516-3 PN/DP:

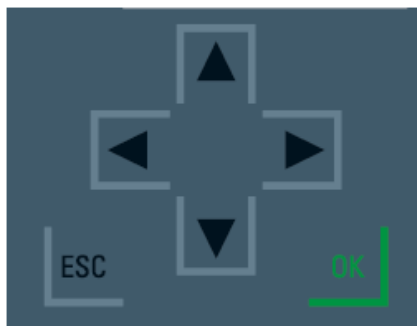


Available submenus of the display:

Main menu items	Meaning	Description
	Overview	The "Overview" menu contains information about the properties of the CPU.
	Diagnostics	The "Diagnostics" menu contains information about diagnostic messages, the diagnostic description, and the indication of interrupts. Additionally there is information about the network properties of each of the CPU's interfaces.
	Settings	In the "Settings" menu, IP addresses of the CPU are assigned, the date, time of day, time zones, operating modes (RUN/STOP) and protection levels are set, a memory reset and reset to factory settings can be performed on the CPU, and the status of firmware updates displayed.
	Modules	The "Modules" menu contains information about the modules that are used in your configuration. The modules can be centrally or peripherally deployed. Peripherally deployed modules are connected to the CPU via PROFINET and/or PROFIBUS. You can set the IP addresses for a CP here.
	Display	In the "Display" menu, settings with respect to the display can be configured, for example language setting, brightness and Energy-saving mode (the Energy-saving darkens the display. The Standby mode switches the display off).

Operator keys of the display

- Four arrow buttons: "up", "down", "left", "right"
- ESC button
- OK button

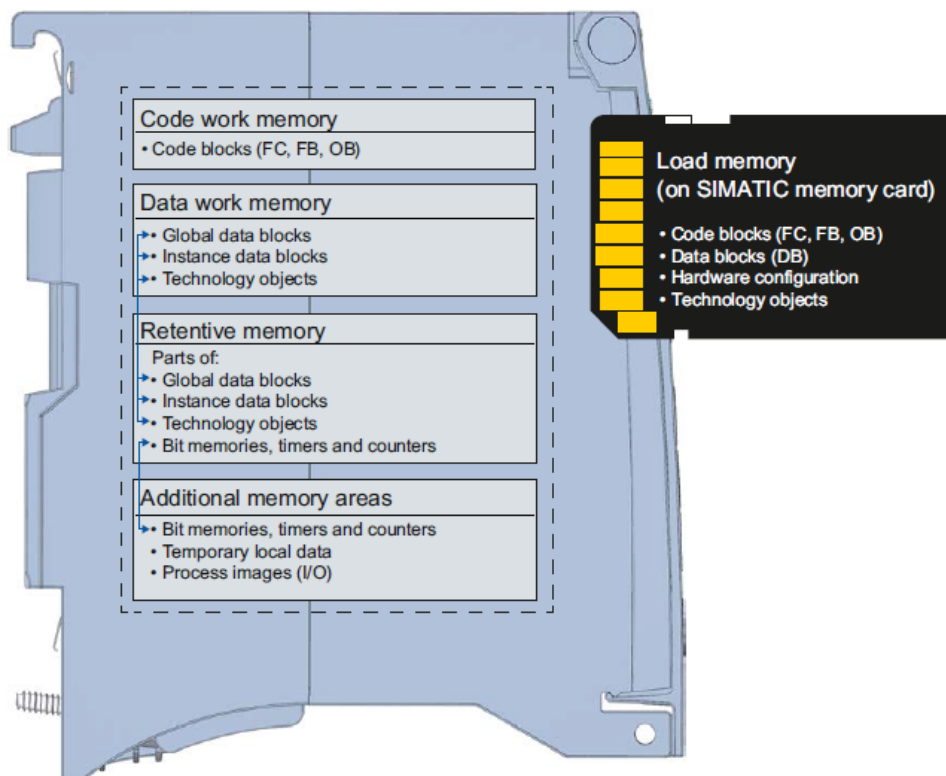


Functions of the "OK" and "ESC" button

- For menu options where inputs can be made:
 - OK → valid access to menu option, confirmation of input and exiting the editing mode
 - ESC → re-establishing the original content (i.e., changes are not saved) and exiting the edit mode
- For menu options where no input can be made:
 - OK → to the next submenu option
 - ESC → back to the previous menu option

6.3 Storage Areas of the CPU 1516-3 PN/DP and the SIMATIC Memory Card

The figure below shows the storage areas of the CPU and the load memory on the SIMATIC memory card. In addition to the load memory, additional data can be loaded with the Windows Explorer to the SIMATIC memory card. This includes, for example, recipes, data logs, backup of projects, additional program documentation.



Load memory

The load memory is a non-volatile memory for code blocks, data blocks, technology objects and for the hardware configuration. When loading these objects to the CPU, they are initially stored in the load memory. This memory is located on the SIMATIC memory card.

Work memory

The work memory is a volatile memory that contains the code and data blocks. The work memory is integrated into the CPU and cannot be expanded. In the case of the S7-1500 CPUs, the work memory is split into two areas:

- **Code work memory:**
The code work memory contains the process-relevant parts of the program code.
- **Data work memory:**
The data work memory contains the process-relevant parts of the data blocks and the technology objects

At the operating mode transitions POWER ON after startup and at STOP after startup, the tags of global data blocks, instance data blocks and technology objects are initialized with their start values; retentive variables receive the actual values saved in the retentive memory.

Retentive memory

The retentive memory is a non-volatile memory for saving certain data if there is a power failure. The variables and operand areas that are defined as retentive are saved in the retentive memory. This data is retained beyond a switch-off or a power failure.

All other program tags are set to their start values in the case of operating mode transitions POWER ON after startup, and at STOP after startup.

The retentive memory content is cleared through the following actions:

- General reset
- Rest to factory settings

Note:

Certain tags of technology objects are also stored in the retentive memory. These are not deleted when a general reset is performed.

7. Sample Task Press Control

For our first program, we are programming a press control.

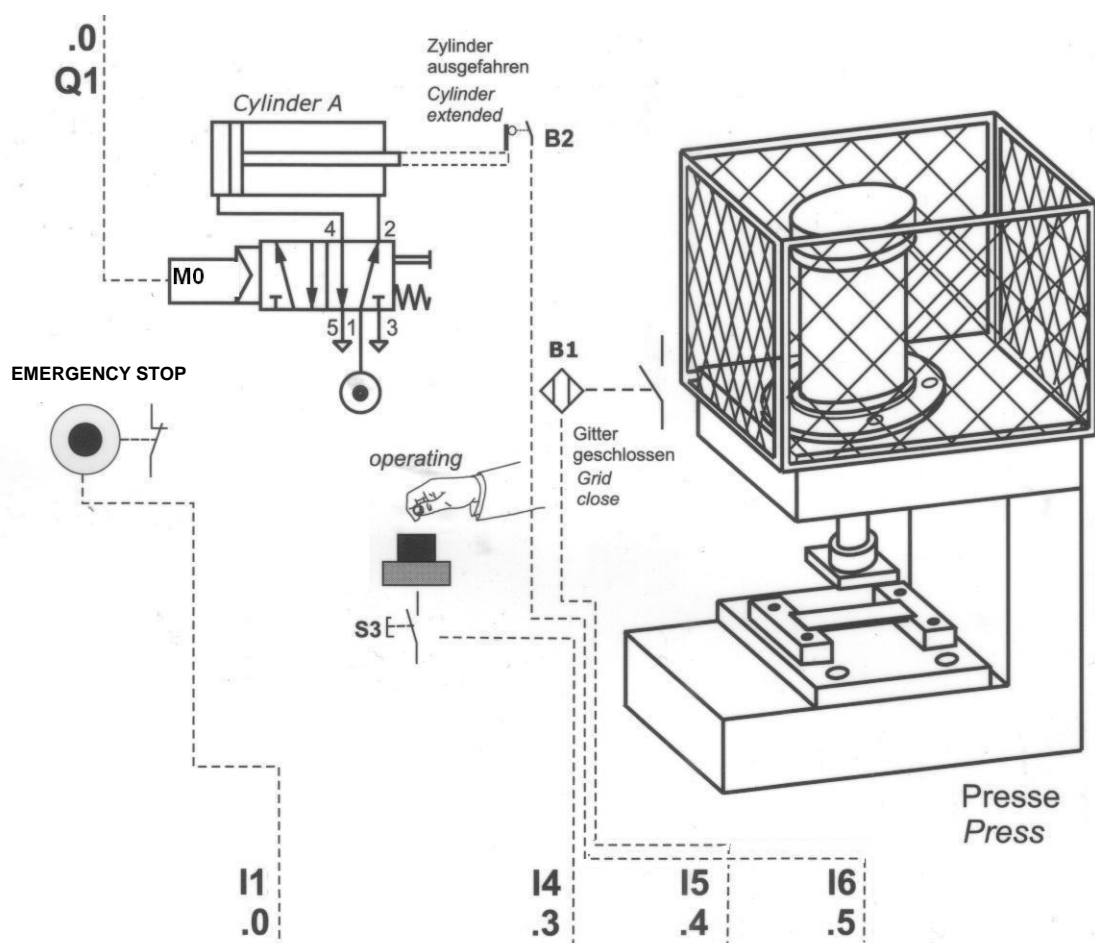
A press with a protection device can only be started with a START button S3 when the protection grid is closed and the EMERGENCY STOP button (NC contact) is not operated. The state Protection grid closed is monitored with a sensor B1.

If this is the case, a 5/2 way valve M0 is set for the press cylinder to press a plastic figure.

The press is to power up again when the start button S3 is operated, the EMERGENCY STOP button (NC contact) is operated or the sensor Protection grid B1 no longer responds.

Assignment list:

Address	Symbol	Comment
%I 0.1	E-STOP	Feedback EMERGENCY STOP (NC)
%I 0.3	S3	Start button (NO)
%I 0.4	B1	Sensor protection grid closed (NO)
%Q 0.0	M0	Extend cylinder A



8. Programming the Press with SIMATIC S7-1500

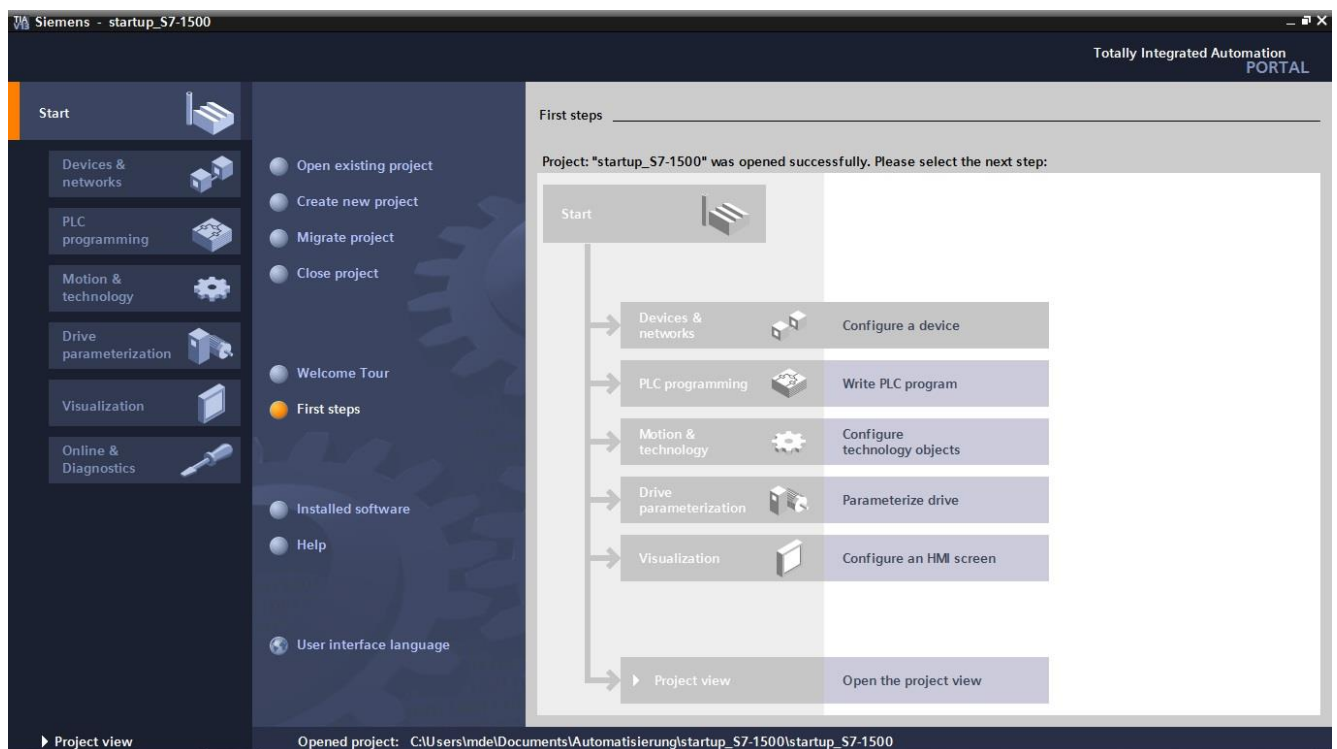
The project is managed and the press is programmed with the software '**Totally Integrated Automation Portal**'.

Here, under a uniform interface, components such as the control system, visualization and networking the automation solution are set up, parameterized and programmed. Convenient online tools are available for error diagnosis.

The software '**Totally Integrated Automation Portal**' has two views: the portal view and the project view.

8.1. Portal View

The portal view offers a task oriented view of the tools used to process the project. Here, you can quickly decide what you want to do and call the tool required for the respective task. If necessary, there is an automatic change to the project view for the selected task. Primarily the entry level and the initial steps are facilitated here.



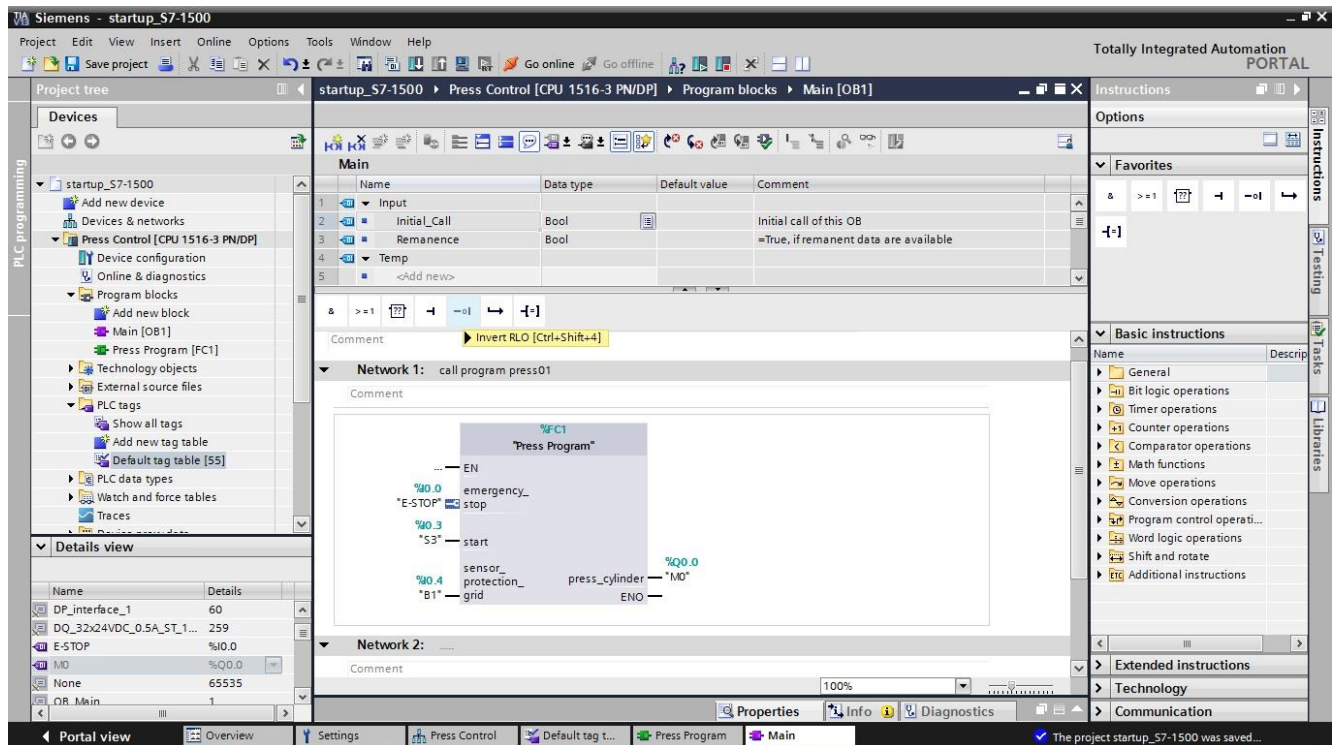
Note:

In the lower left, we can jump from the portal view to the project view.

8.2. Project View

The project view is a structured view to all constituent parts of the project. By default, the following is provided: on top a menu bar with the tool bars; on the left the project tree with all constituent parts of a project, and on the right the 'Task Cards' with instructions and libraries, for example.

If in the project tree an element (for example, the organization block OB1) is selected, it is displayed in the center and can be processed there.



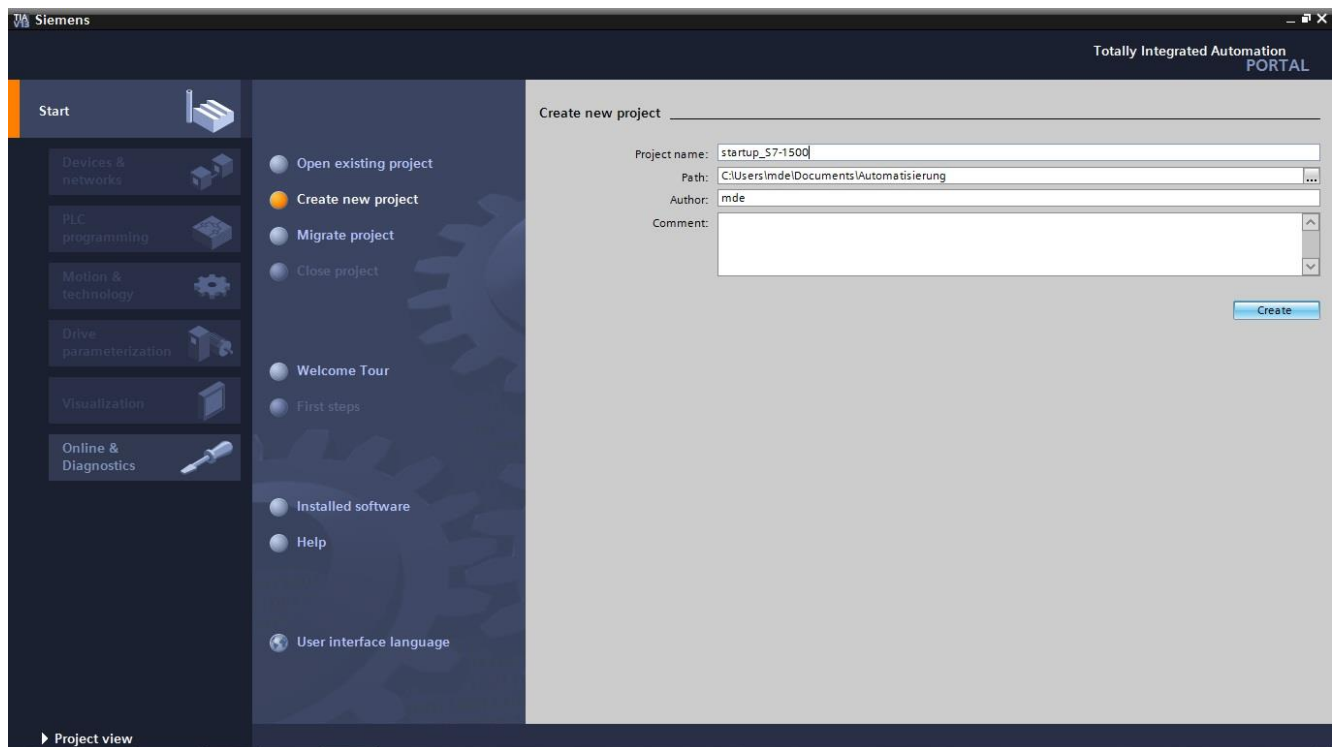
Note: On the lower left, we can jump from the project view to the portal view.

With the steps below, we can set up a project for the SIMATIC S7-1500 and program the solution for the task.

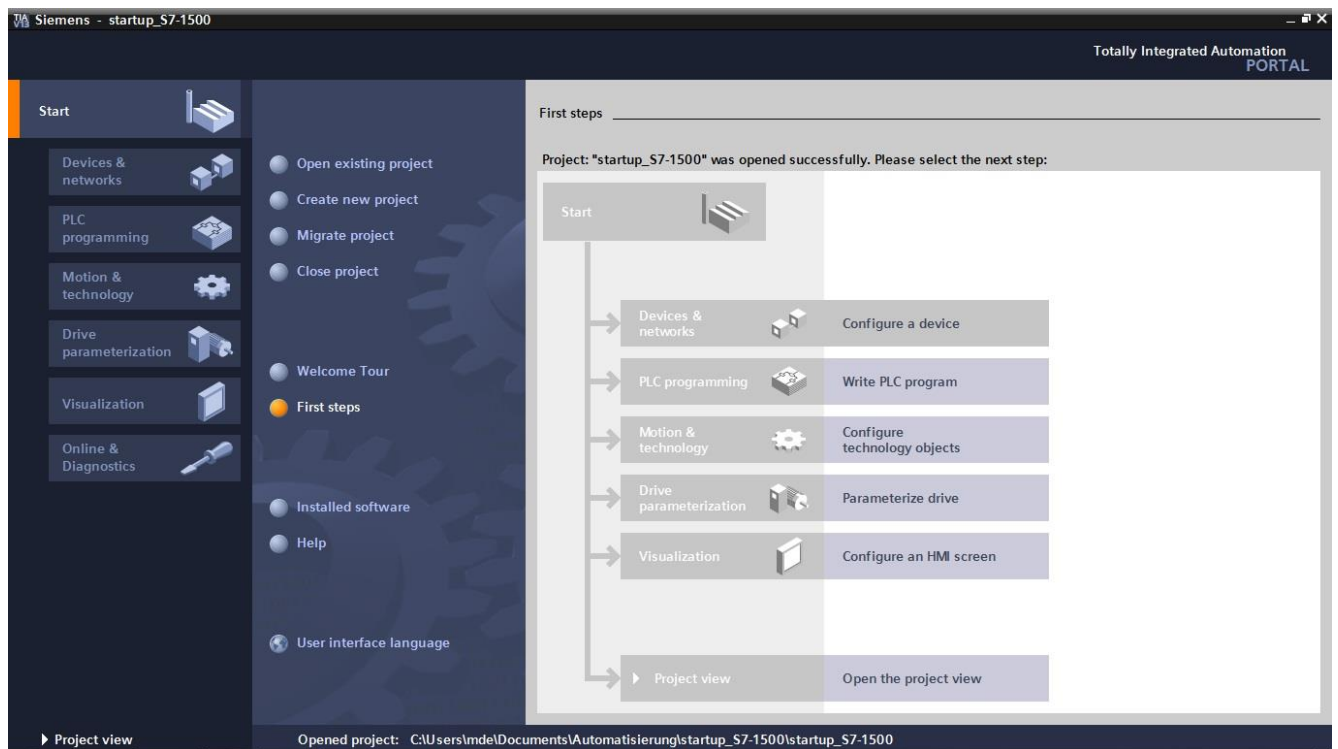
1. The central tool is the **'Totally Integrated Automation Portal'**; we are calling it with a double click.
(→ TIA Portal V13)



2. Programs for SIMATIC S7-1500 are managed in projects. We are now setting up such a project in the portal view (→ Create new project → startup_S7-1500 → Create)



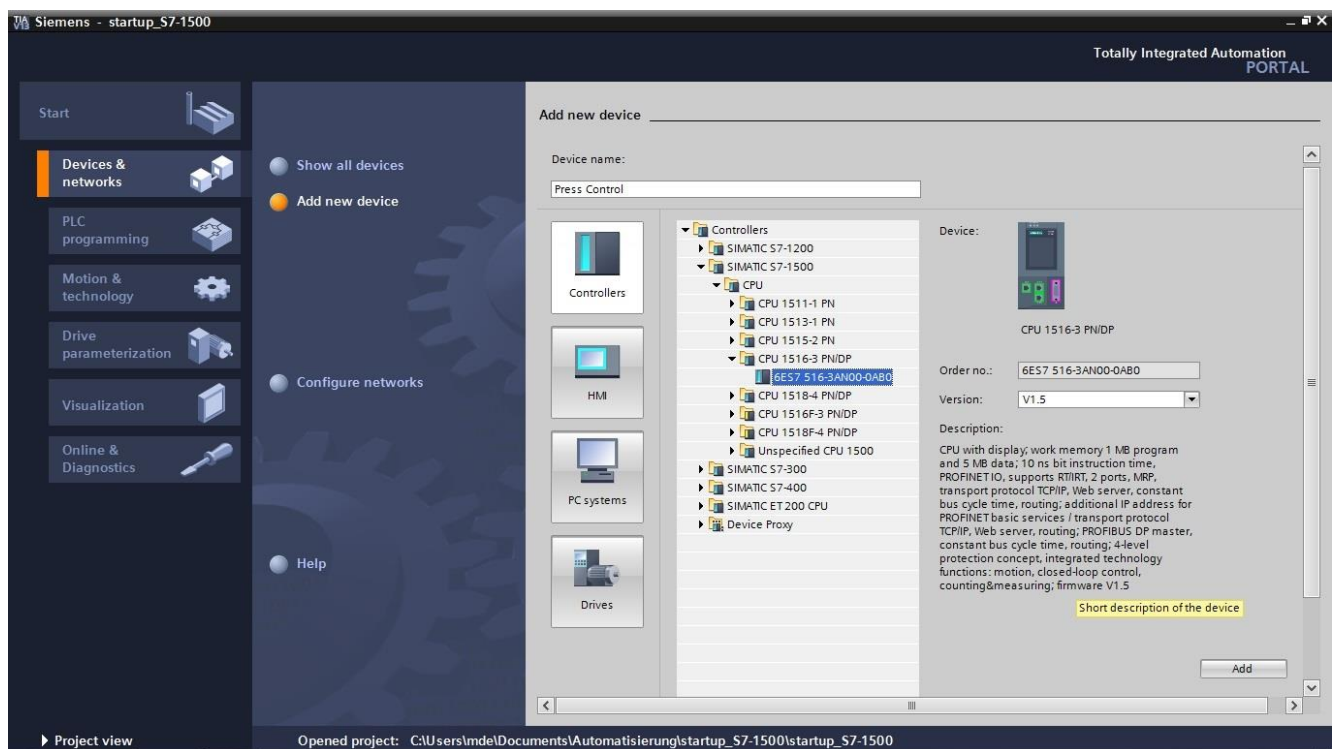
3. Now, '**First steps**' are suggested for the configuration. First, we want to '**Configure a device**'.
(→ First steps → Configure a device)



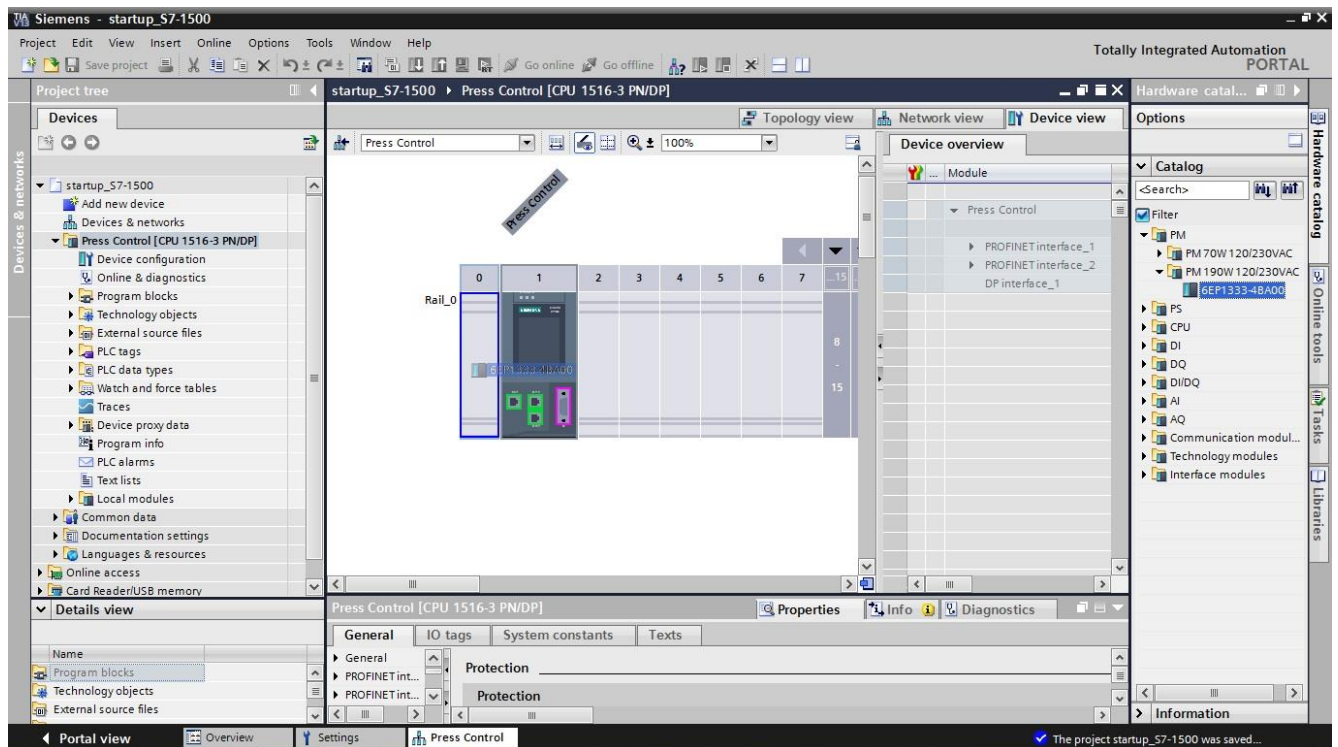
Option 1: Creating the hardware configuration offline

The complete hardware configuration for each controller is stored in the TIA portal. It is important regarding error search and is part of a complete plant documentation. In the initial variant, we are creating this configuration without being connected to the controller. We obtain the information about the modules used from the data imprinted there, or an available order list.

- Below, we 'Add (a) new device' with the 'Device name: Press control'. From the catalog, we select the 'CPU 1516-3 PN/DP' with the matching order number. (→ Add new device → Press control → Controller → SIMATIC S7-1500 → CPU → CPU 1516-3 PN/DP → 6ES7 516-3AN00-0AB0 → V1.5 → Add)



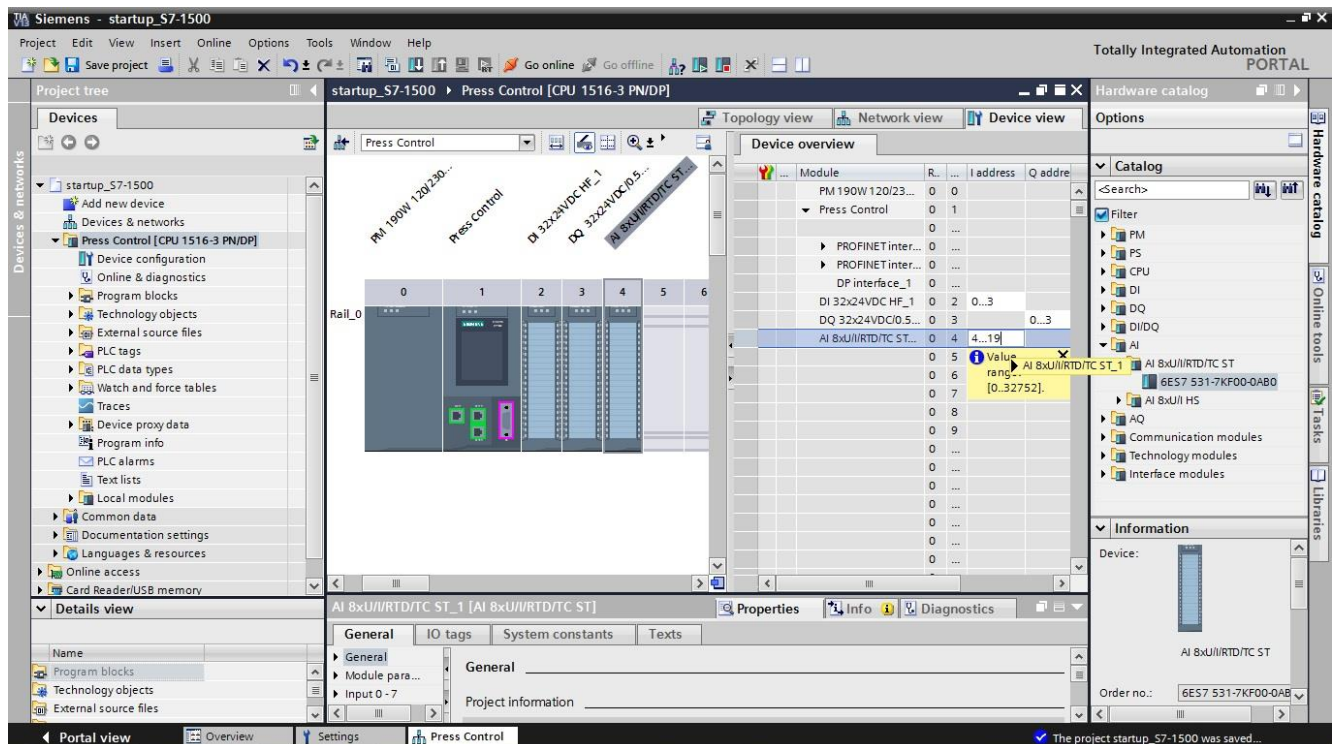
- Now, the software changes automatically to the project view with the opened hardware configuration in the device view. Here, we can add additional modules from the Catalog (on the right!). First, we select the power module PM190W 120/230VAC and drag it to Slot 0 (→ Catalog → PM → PM190W 120/230VAC → 6EP1333-4BA00)



- [illegible]

- [illegible]

8. We now drag signal module AI 8xU/I/RTD/TC ST with 8 analog input channels to Slot 4. (→ Hardware Catalog → AI → AI 8xU/I/RTD/TC ST → 6ES7 531-7KF00-0AB0) In the '**Device overview**' we can set the addresses of the analog channels. Here, the inputs of the signal module have the addresses %IW4 to %IW18. (→ Device overview → AI 8xU/I/RTD/TC ST → 4...19)

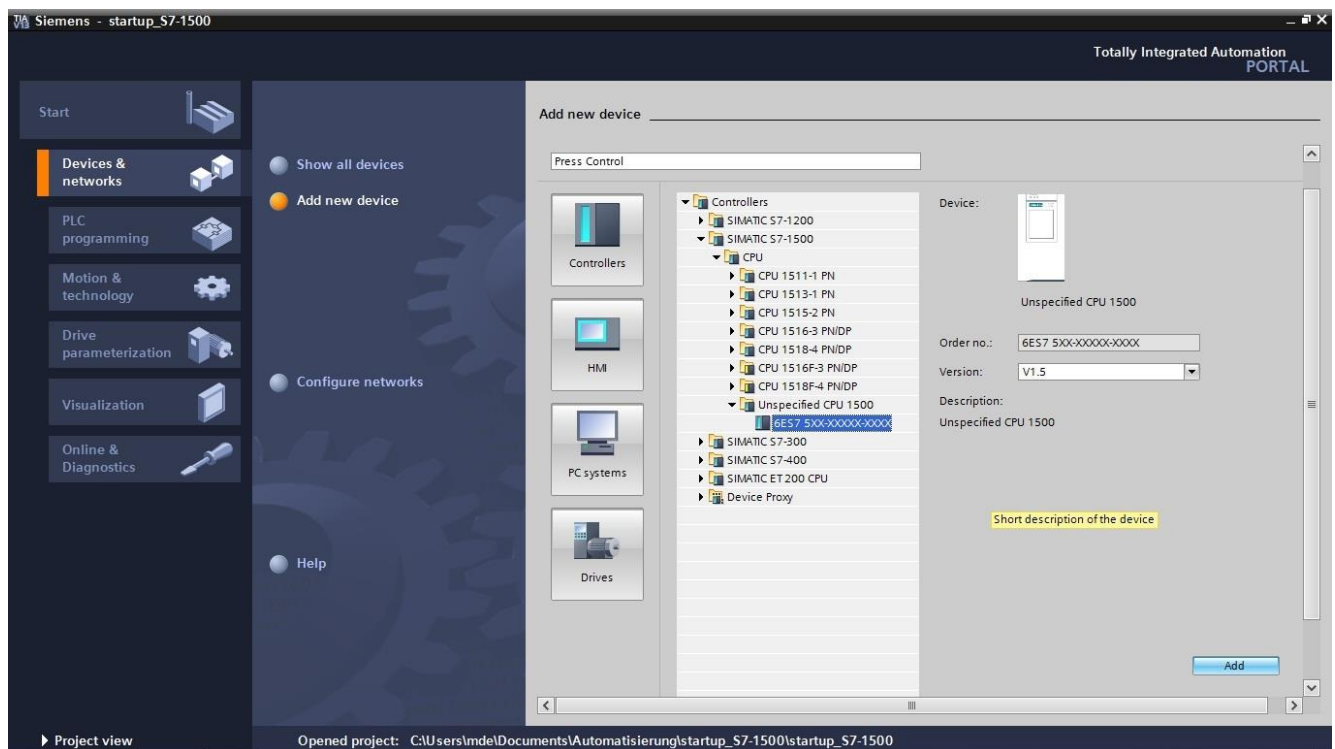


- [illegible]

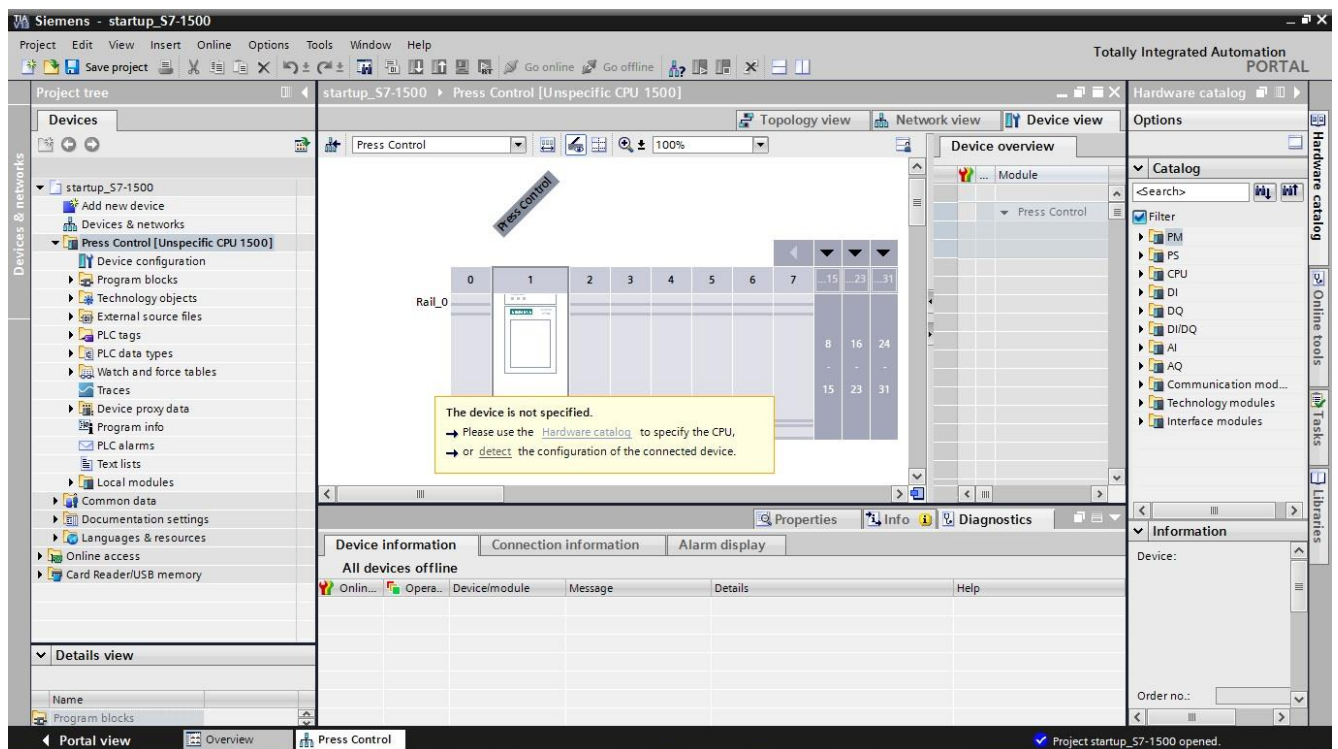
Option 2: Detecting the hardware configuration online

The complete hardware configuration for each controller is stored in the TIA Portal project. It is important regarding error search and is part of a complete plant documentation. In the second variant, we generate the configuration by connecting ourselves to the controller and have the configuration detected online by the TIA Portal.

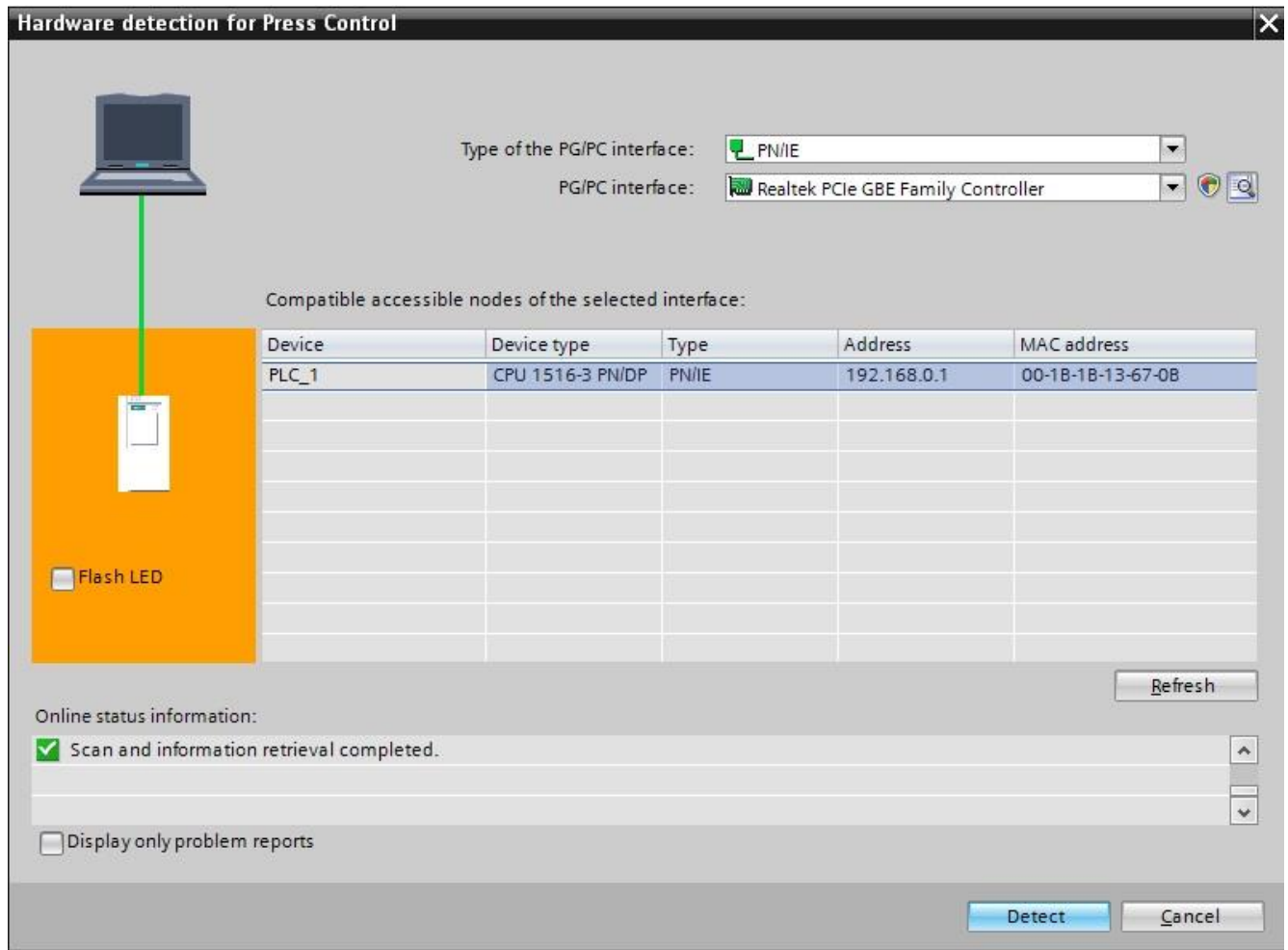
10. First, we **'Add (a) new device'** with the device name **'Press control'**. To this end, we select from the catalog an **'Unspecified CPU 1500'** with the general order number **'6ES7 5XX-XXXX-XXXX'**.
(→ Add new device → Press control → Controller → SIMATIC S7-1500 → CPU → Unspecified CPU 1500 → 6ES7 5XX-XXXX-XXXX → V1.5 → Add)



11. The software now changes automatically to the project view with the opened hardware configuration in the device view. Here, the TIA Portal recommends having the configuration of the connected device '**detect(ed)**'. (→ detect)

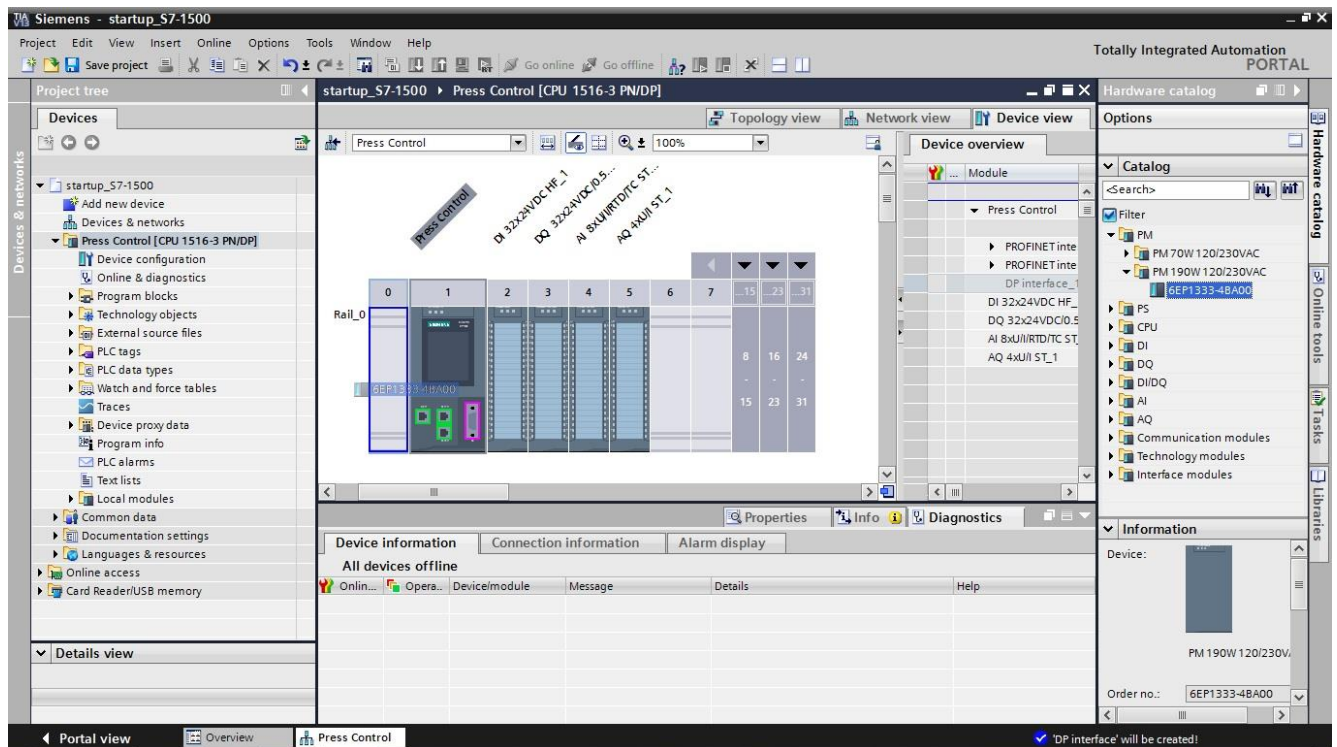


12. In the following dialog, first select '**PN/IE**' as the type of the PG/PC interface, and then the network card that was previously set as PG/PC interface. After '**Refresh**' of the accessible stations, you should now see your '**CPU 1516-3 PN/DP**' with the address 192.168.0.1, and select it as the target device. Click on '**Detect**'. (→ Type of PG/PC interface: PN/IE → PG/PC interface: → Refresh → CPU 1516-3 PN/DP → Detect)



13. Now, all modules in the hardware configuration are displayed in the device view. Only the power module PM190W 120/230VAC has no connection to the backplane and has to be dragged manually from the hardware catalog to Slot 0 for that reason.

(→ Hardware Catalog → PM → PM190W 120/230VAC → 6EP1333-4BA00)



14. In the **'Device overview'** we can set the addresses of the digital and analog channels correspondingly. Here, the inputs of the digital signal module are assigned the addresses %I0.0 to %I3.7, the outputs of the digital signal module the addresses %O0.0 to %O3.7. The analog inputs are assigned the addresses %IW4 to %IW18 and the analog outputs the addresses %OW4 to %OW10 (→ Device overview → 0... 3 → 0... 3 → 4... 19 → 4...11)

The screenshot shows the Siemens TIA Portal interface for a 'Press Control' project. The 'Device overview' window is active, displaying a rack and slot configuration. The modules are as follows:

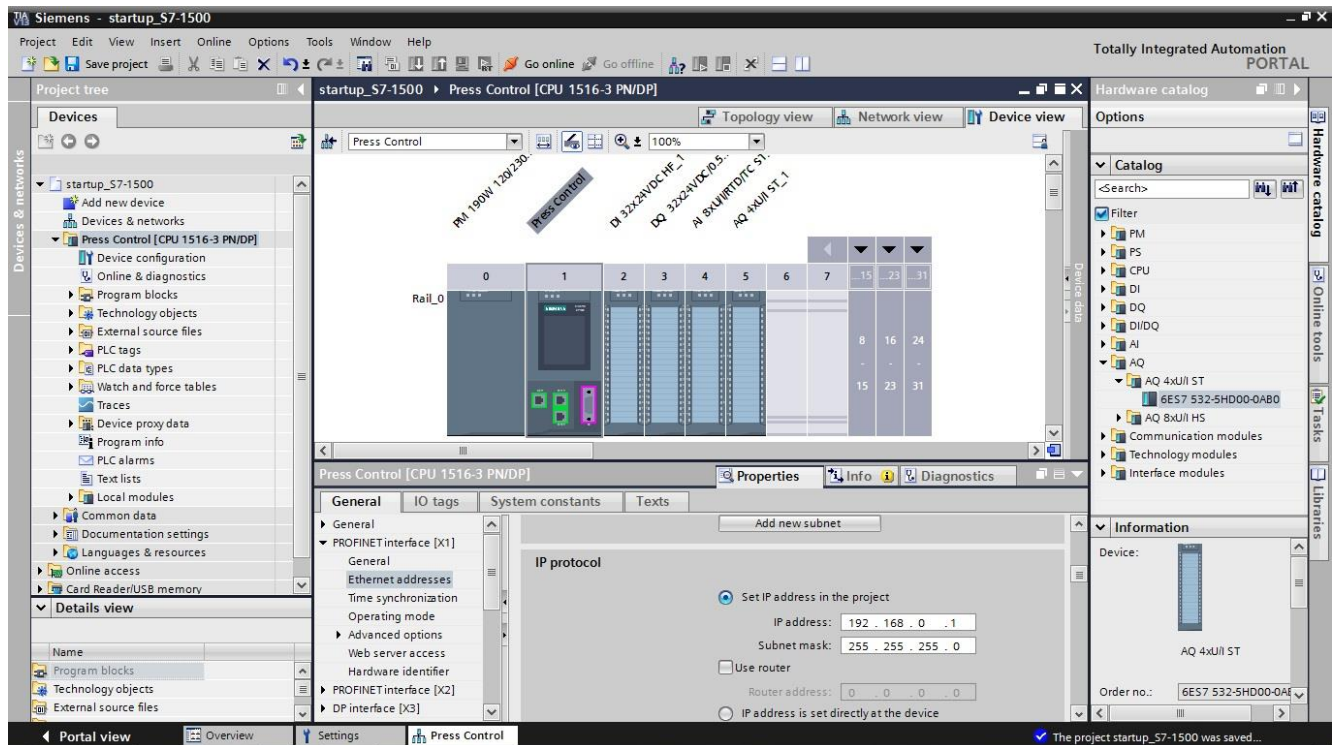
Module	Rack	Slot	I address	Q address
PM 190W 120/230...	0	0		
Press Control	0	1		
PROFINET interf...	0	1 X1		
PROFINET interf...	0	1 X2		
DP interface_1	0	1 X3		
DI 32x24VDC HF_1	0	2	0...3	0...3
DQ 32x24VDC/0.5...	0	3		
AI 8x12bit DTC ST_1	0	4	4...19	
AQ 4x12bit ST_1	0	5		4...11
	0	6		
	0	7		
	0	8		
	0	9		
	0	10		
	0	11		

A tooltip for the AQ module shows a value range of [0...32760]. The 'Device information' section at the bottom shows 'All devices offline'.

15. For the software to later access the correct CPU, the IP address and the subnet mask have to be set.

(→ Press control → Properties → General → PROFINET interface [X1] → Ethernet addresses → Set IP address in the project → IP address: 192.168.0.1 → Subnet mask: 255.255.255.0)

(refer also to: Setting the Programming Interface)



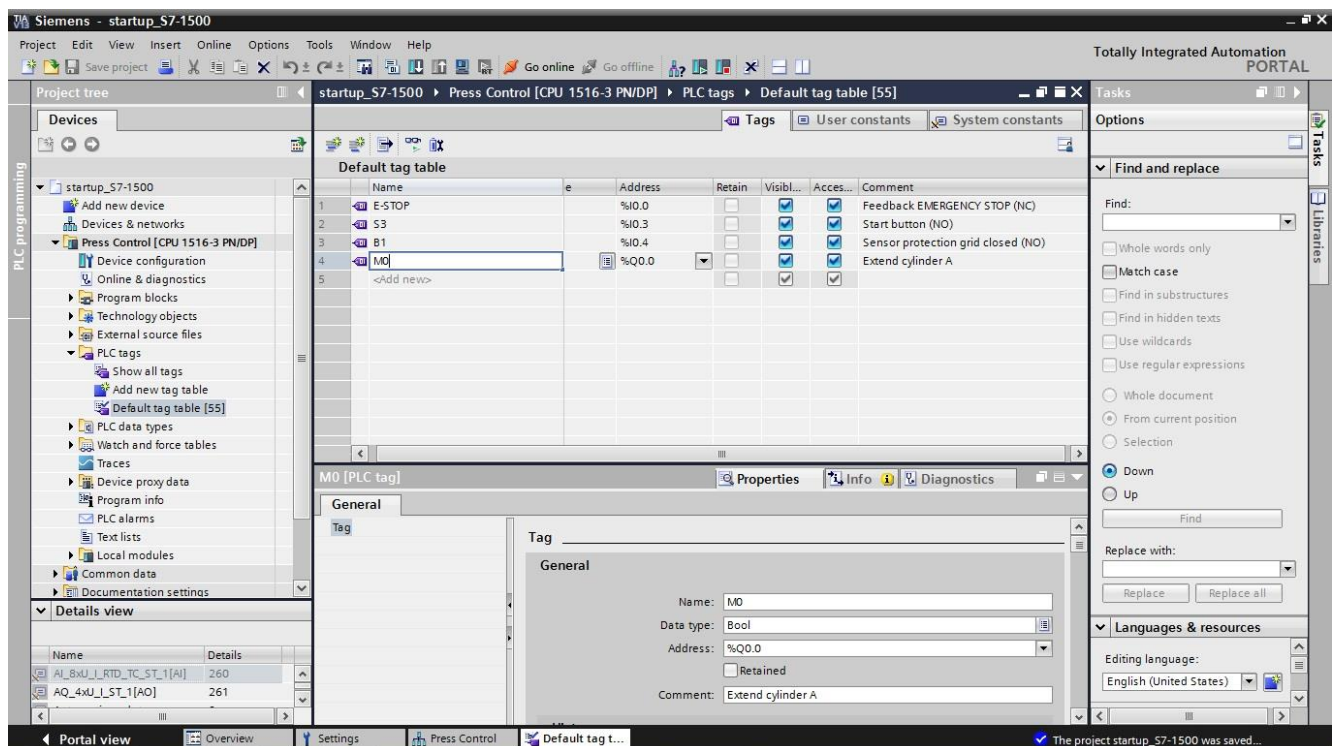
Note:

In this example, we connect the PROFINET interface [X1] with one of the 2 ports.

- Since in modern programming we don't program with absolute addresses but with tags, here we have to specify the **global PLC tags**.

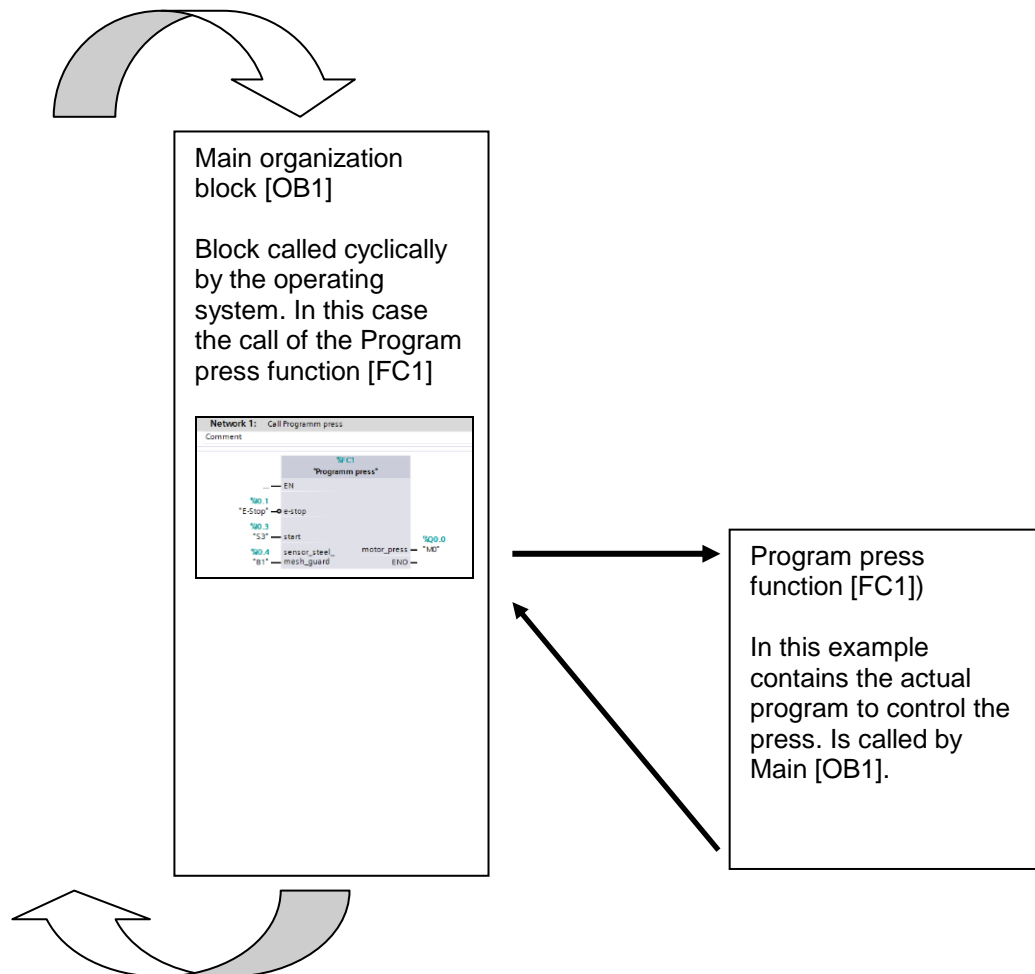
These global PLC tags are descriptive names with a comment for those inputs and outputs that are used in the program. Later, during programming, the global PLC tags can be accessed by means of this name. These global tags can be used in the entire program in all blocks.

To this end, select in the program tree the '**Press Control [CPU 1516-3 PN/DP]**' and the '**PLC tags**'. Then open the '**Default tag table**' with a double click and enter the names for the inputs and outputs as shown below. (→ Press Control[CPU 1516-3 PN/DP] → PLC tags → Default tag table)

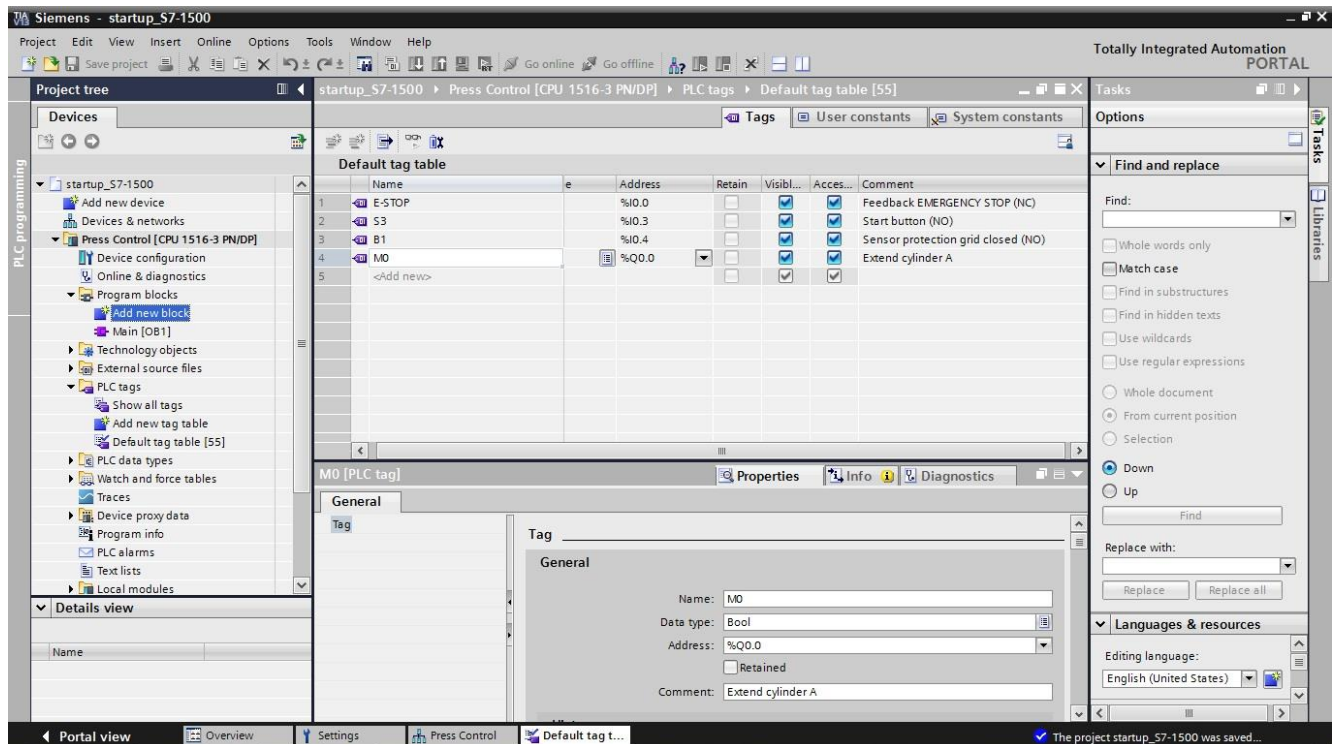


17. In this manner, the program sequence is written to blocks. The organization block Main [OB1] exists by default. OB1 represents the interface to the CPU's operating system. OB1 is called automatically and processed cyclically by the operating system. From this organization block, additional blocks can be called in turn for structured programming; for example, the function Press Program [FC1]. The purpose is: breaking down a task into partial tasks that can then be solved and tested in their functionality more simply.

Program structure of the example:

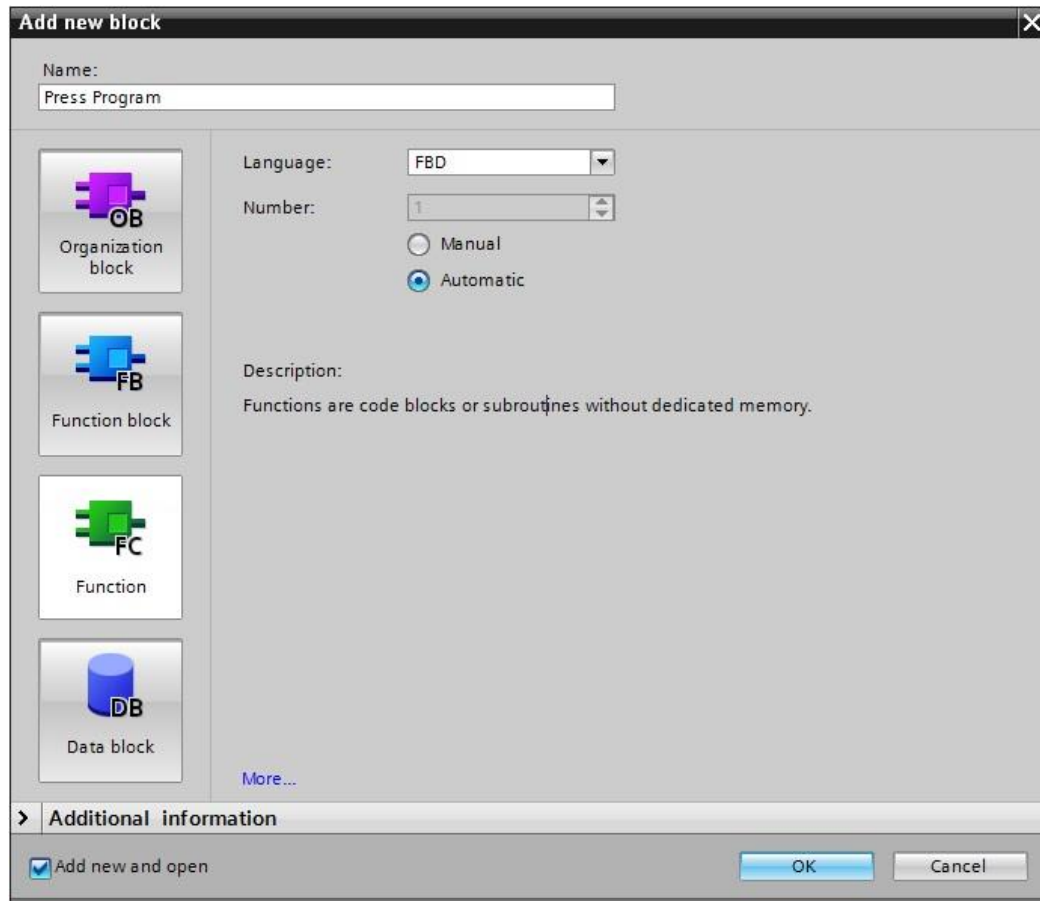


18. To create the function Press Program [FC1] select in the program tree '**Press Control**[CPU 1516-3 PN/DP]' and '**Program blocks**'. Then, double click on '**Add new block**'.
(→ Press Control[CPU 1516-3 PN/DP] → Program blocks → Add new block)



19. From the selection, select '**Function (FC)**' and assign the name '**Press Program**'. As programming language we specify '**FBD**'. Numbering is performed automatically. Since this FC1 is called later under its symbolic name anyhow, the number is not that important any longer. Accept the inputs with '**OK**'.

(→ Function (FC) → Press Program → FBD → OK)



20. The block '**Press Program[FC1]**' is opened automatically. Before the program can be written, the block interface has to be declared. When the interface is declared, the local tags -known only in this block- are specified.

The tags are divided into two groups:

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

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

- Local data that is used for saving intermediate results.

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

21. When the local tags are declared, the following tags are needed for our example.

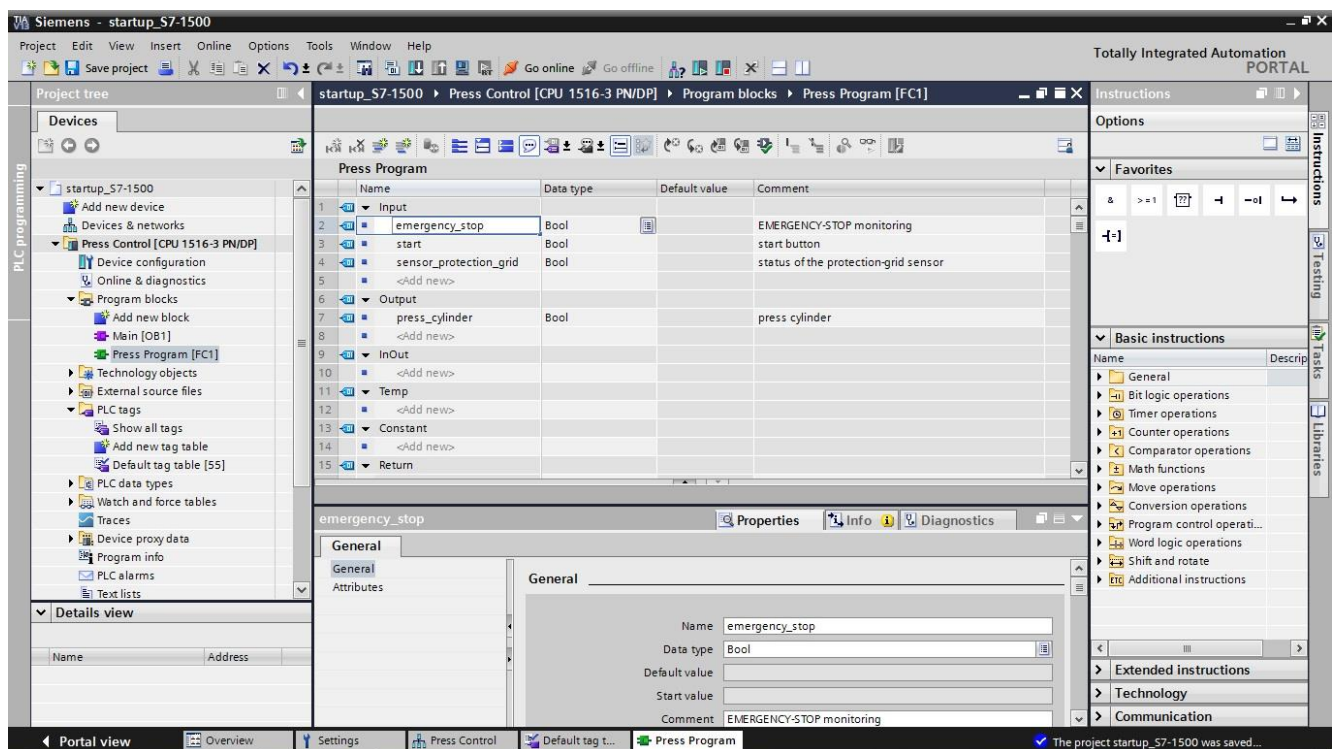
Input:

emergency_stop The EMERGENCY STOP monitoring is entered here
 start The start button is entered here
 sensor_protection_grid The status of the protection-grid sensor is entered here

Output:


press_cylinder The status for the press cylinder output is written here


All tags are of the type 'Bool'; i.e., binary tags that only have the state '0' (false) or '1' (true).
 All local tags should be provided with a sufficient commentary for better understanding.




Note:

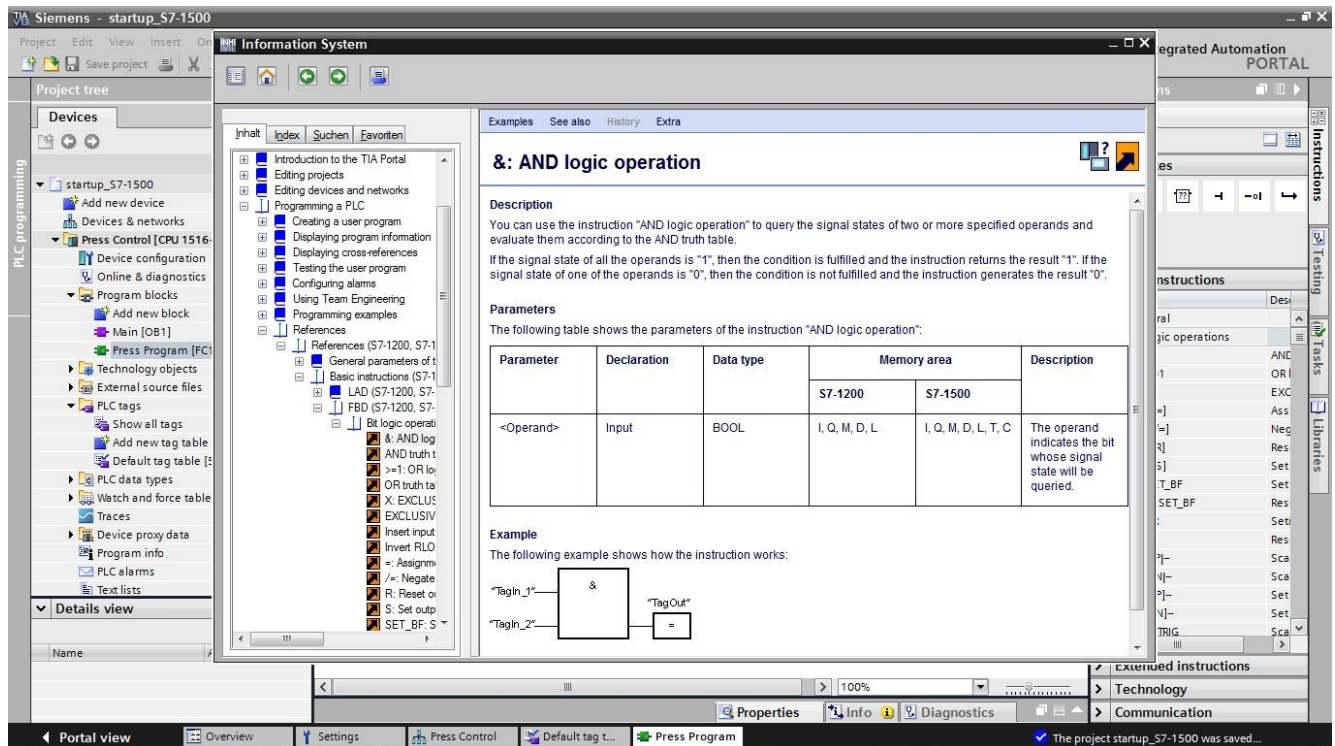
To prevent mistaking these tags with PLC tags, local tags are written here in the lower case.

22. After we declared the local tags, we can start programming. For the sake of clarity, we program in networks. We can add a new network by clicking on the  **'Add network'** symbol. Just like the block itself, each network should be documented at least in the title line. If a longer text is needed for the description, the **'Comment'** field can be used also.

For creating our solution, we need an **'&'** for an AND logic operation. It is located under **'Basic instructions'** in the folder **'Bit logic operations'**. By pointing the mouse to an object such as  detail information about this object is provided.

(→ Basic instructions → Bit logic operations → )

23. For online help to be displayed in a window regarding this object, click on the blue colored text in the brief description of the symbol . (→&AND logic operation)



&: AND logic operation

Description
You can use the instruction "AND logic operation" to query the signal states of two or more specified operands and evaluate them according to the AND truth table.
If the signal state of all the operands is "1", then the condition is fulfilled and the instruction returns the result "1". If the signal state of one of the operands is "0", then the condition is not fulfilled and the instruction generates the result "0".

Parameters
The following table shows the parameters of the instruction "AND logic operation":


Parameter	Declaration	Data type	Memory area		Description
			S7-1200	S7-1500	
<Operand>	Input	BOOL	I, Q, M, D, L	I, Q, M, D, L, T, C	The operand indicates the bit whose signal state will be queried.



Example
The following example shows how the instruction works:

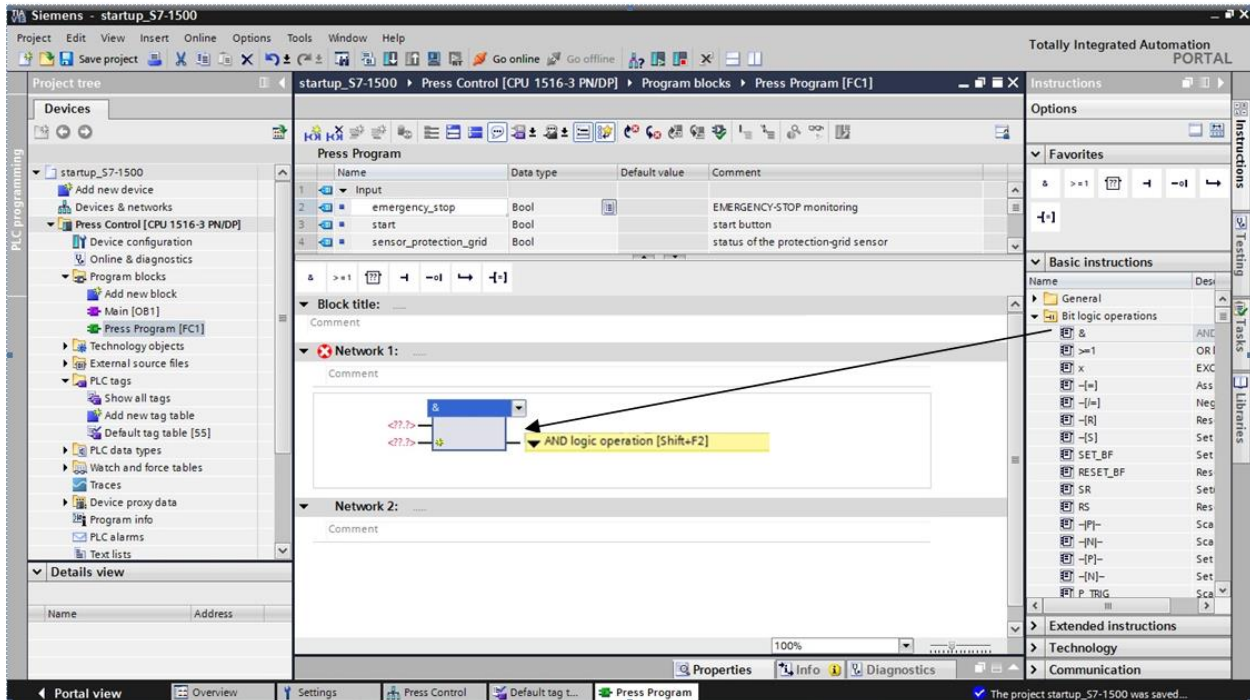
```

graph LR
    TagIn_1["TagIn_1"] --> AND["&"]
    TagIn_2["TagIn_2"] --> AND
    AND --> TagOut["TagOut"]
  
```

Note:

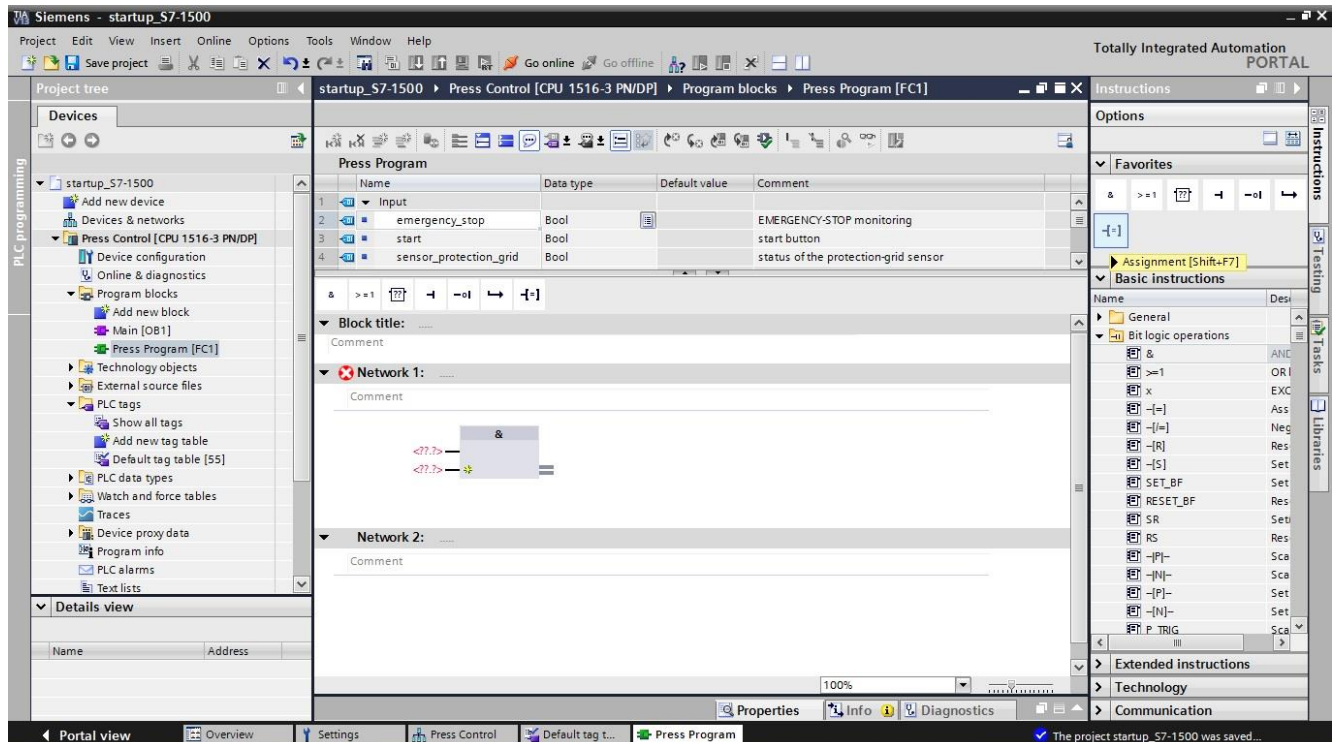
At this point, inform yourself in detail through Online Help regarding the function and wiring of  AND logic operation.




24. With the mouse, drag  to below the commentary in Network 1. (→ )

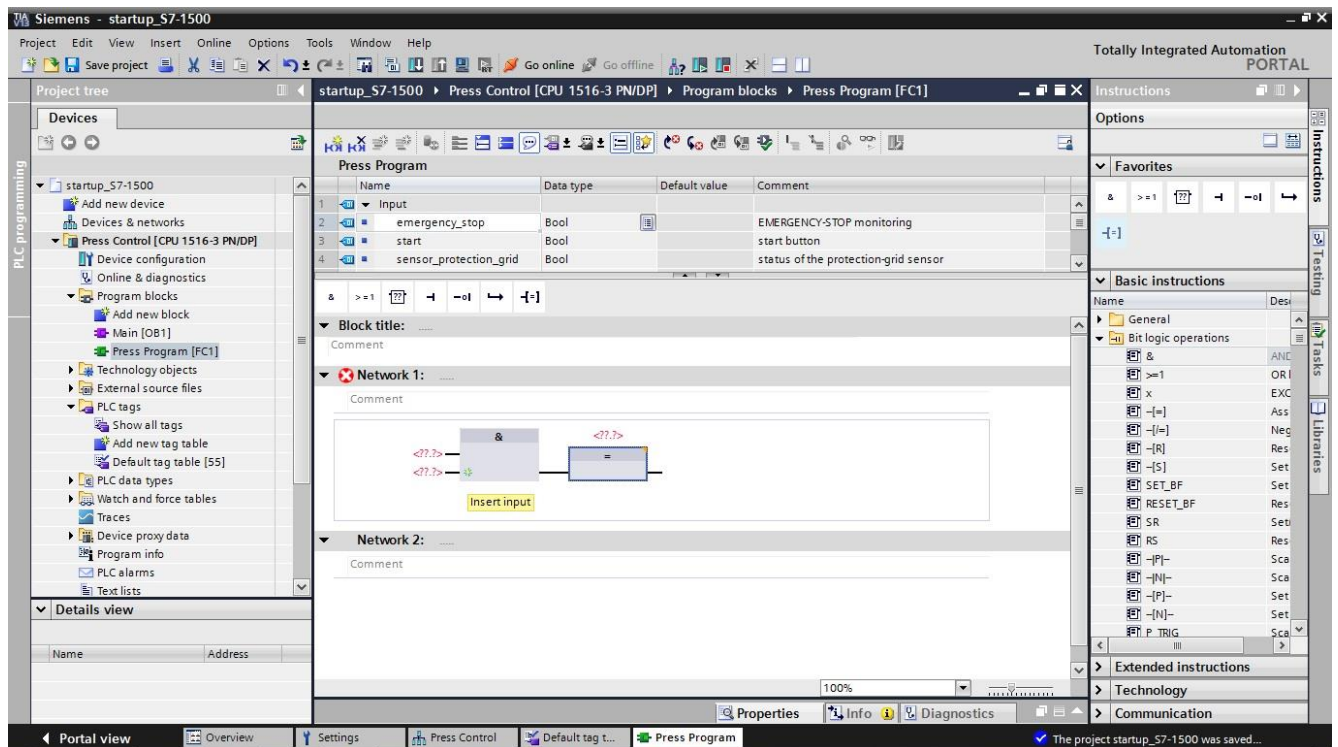


The screenshot displays the Siemens SIMATIC Manager interface for the 'Press Program [FC1]'. The main workspace shows a ladder logic network with an AND operation. A yellow callout box highlights the AND operation with the text 'AND logic operation [Shift+F2]'. The right-hand pane shows the 'Basic instructions' list, where the AND operation is selected. The bottom status bar indicates 'The project startup_S7-1500 was saved...'.

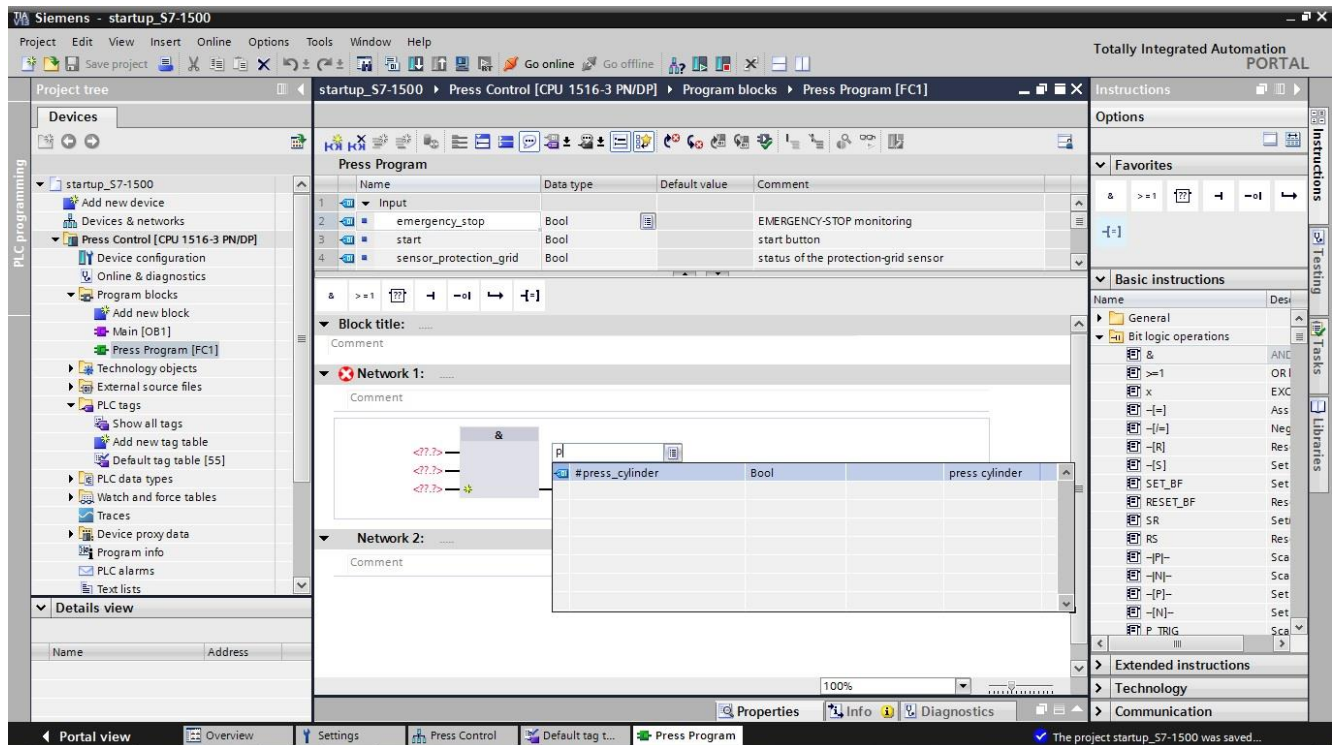
25. Next, we highlight the output of the AND block  and double click on the  Assignment in the Favorites. (→  right input → Favorites →  Assignment)



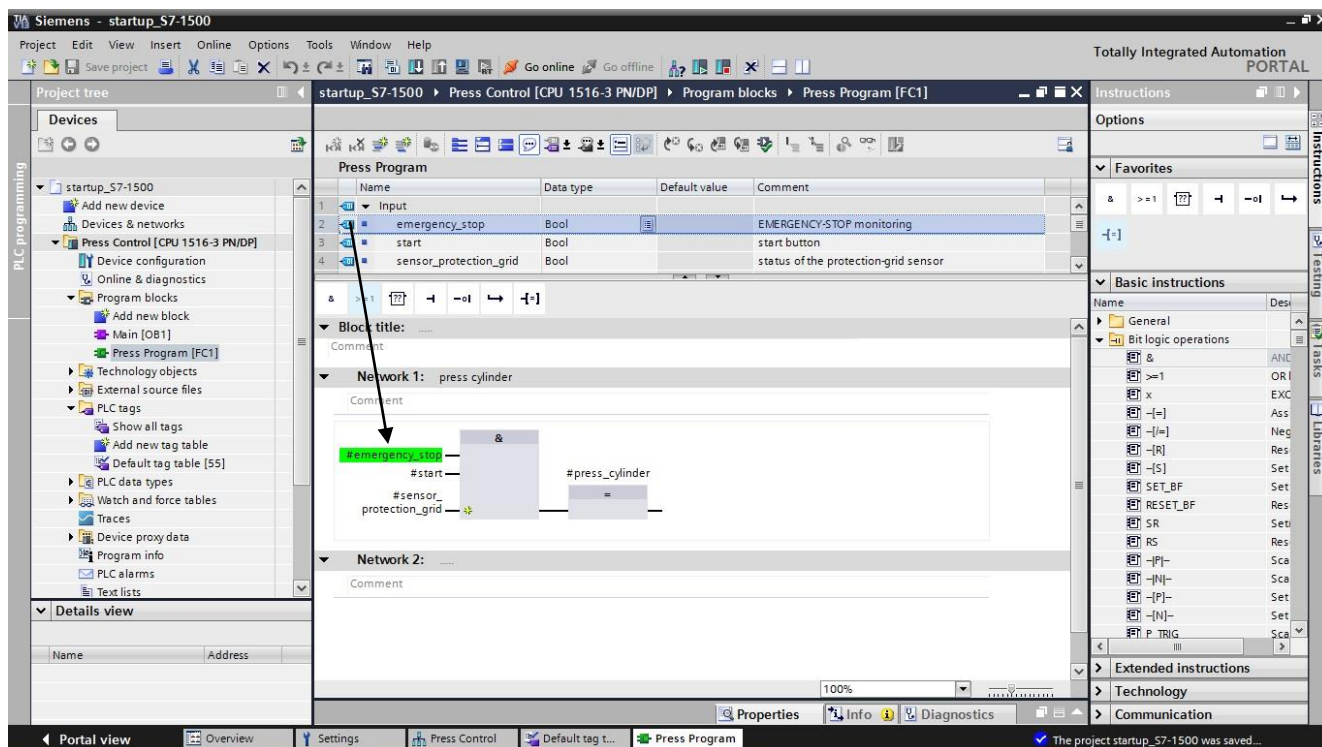
26. An additional input at the AND block can be inserted either with Drag&Drop from the Favorites  or by clicking on the symbol  for insert input - at the lower left of the AND block. (→ )




27. Now we enter the local tags. It is sufficient to enter the first letter of the local tags in the fields at the instructions. Then we select the desired tag from a list. Local tags are always marked with the symbol '#' in front of the name. (→ #press_cylinder)

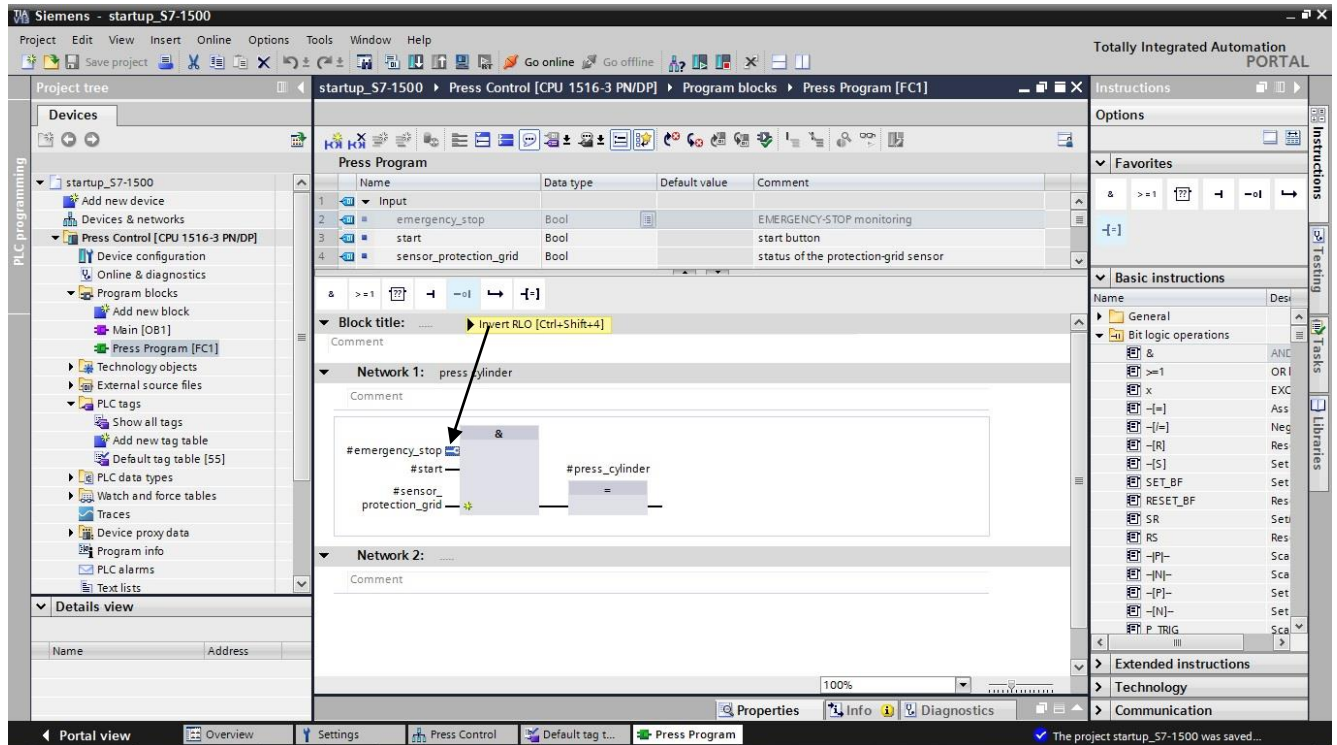


28. We then simply drag the other local tags from the interface to the corresponding input.
 (→ #emergency_stop → #start → #sensor_protection_grid)



29. If an input is to be inverted, just drag the symbol  Negation from the 'Favorites' to the input.

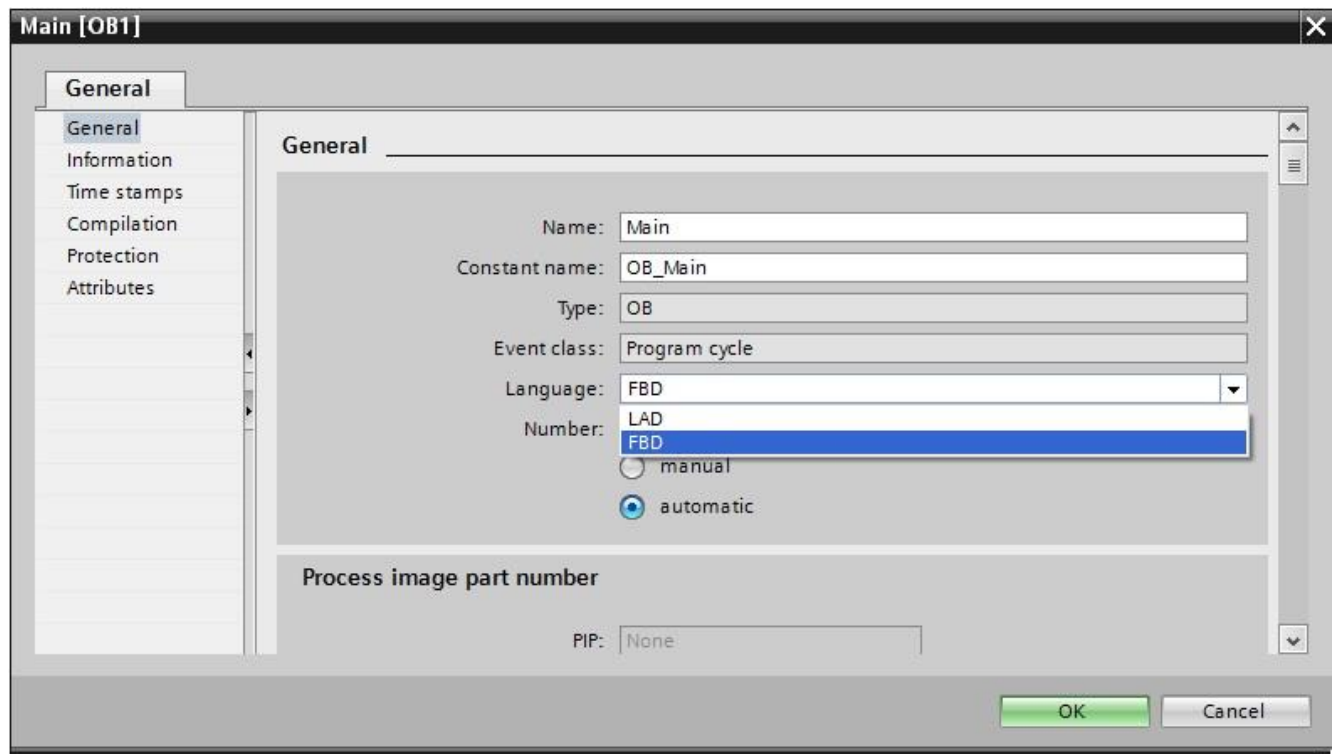
Here, polling the local tag #emergency_stop has to be inverted. (→ Favorites → )



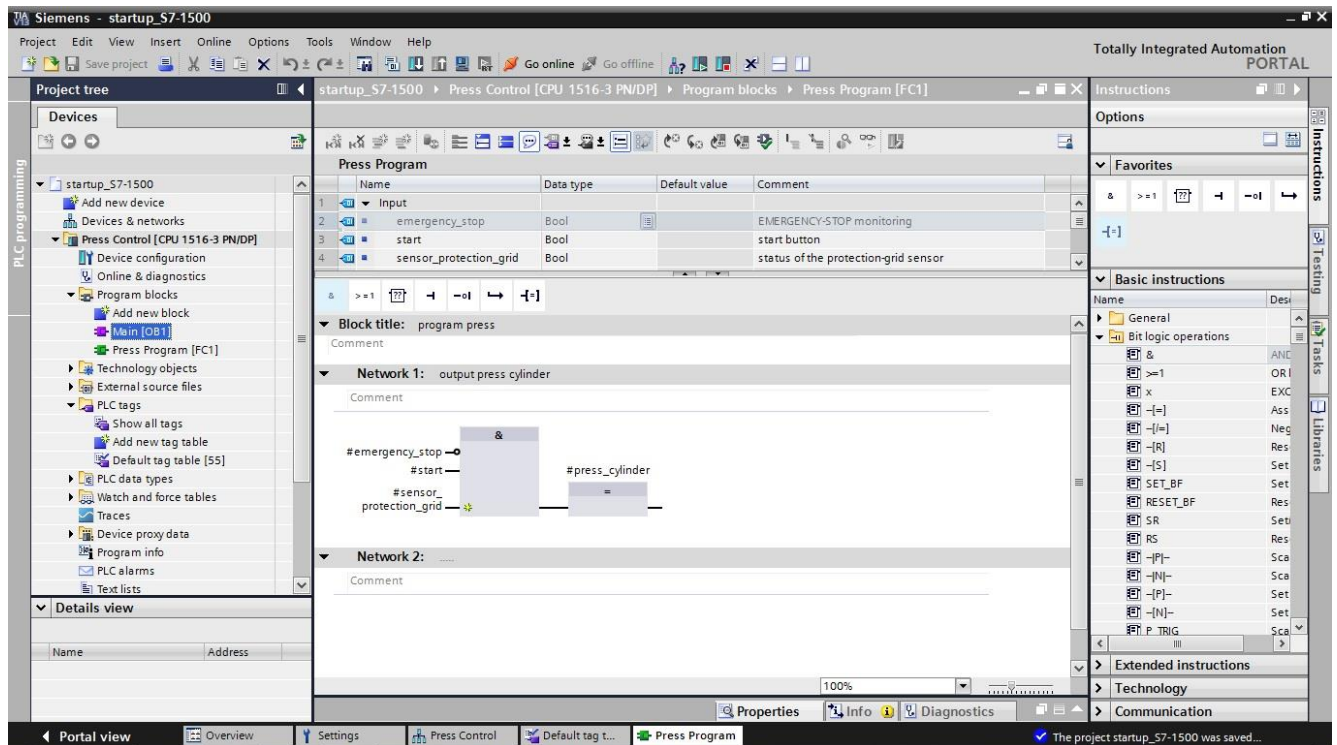
The screenshot shows the Siemens SIMATIC Manager software interface for the 'startup_S7-1500' project. The main workspace displays the 'Press Program' ladder logic. The 'Favorites' panel on the right shows the negation symbol being dragged to the #emergency_stop input in Network 1. The 'Basic instructions' panel shows the 'General' section with the 'Bit logic operations' sub-section. The 'Extended instructions' panel shows the 'Technology' section. The 'Communication' panel shows the 'Communication' section. The 'Properties' panel shows the 'Properties' section. The 'Info' panel shows the 'Info' section. The 'Diagnostics' panel shows the 'Diagnostics' section. The 'Diagnostics' panel shows the 'Diagnostics' section. The 'Diagnostics' panel shows the 'Diagnostics' section.

-
- The screenshot displays the Siemens SIMATIC Manager interface for the 'startups_S7-1500' project. The main window shows the 'Press Program [FC1]' in the 'Press Control [CPU 1516-3 PN/DP]' project. The 'Project tree' on the left lists the project structure, including 'Devices', 'Program blocks', and 'PLC tags'. The 'Main' block is selected, and a context menu is open, showing options like 'Open', 'Cut', 'Copy', 'Paste', 'Delete', 'Rename', 'Compile', 'Download to device', 'Go online', 'Go offline', 'Generate source from blocks', 'Cross-reference information', 'Cross-references', 'Call structure', 'Assignment list', 'Switch programming language', 'Know-how protection', 'Print...', 'Print preview...', and 'Properties...'. The ladder logic editor shows a network with an AND gate connecting 'emergency_stop' and 'start' to 'press_cylinder'. The right-hand pane contains 'Instructions' and 'Options' tabs, with a list of basic instructions like '>=1', '&', 'x', '[-]', '[=]', '[R]', '[S]', 'SET_BF', 'RESET_BF', 'SR', 'RS', '[-P]', '[-N]', '[-P]', '[-N]', and 'P_TRIG'.

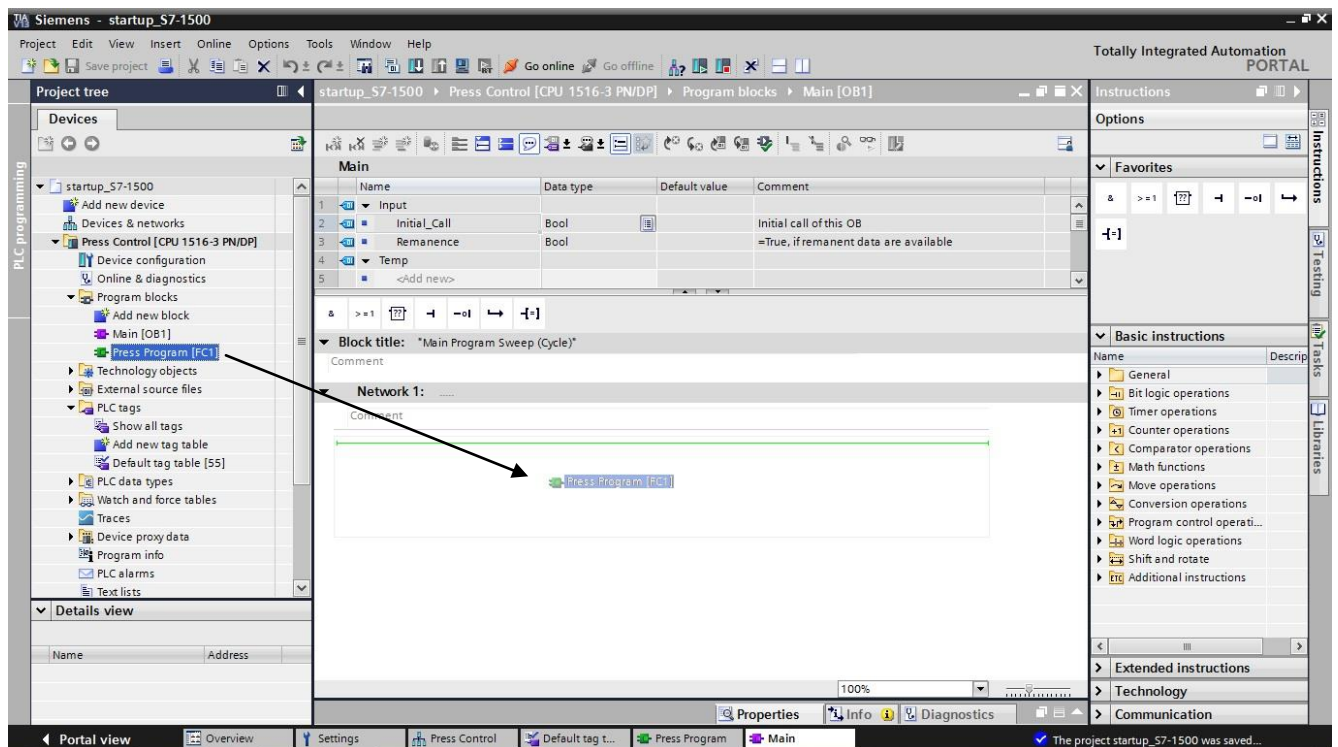
31. In the Properties, select the programming 'Language' 'FBD'. (→ FBD → OK)



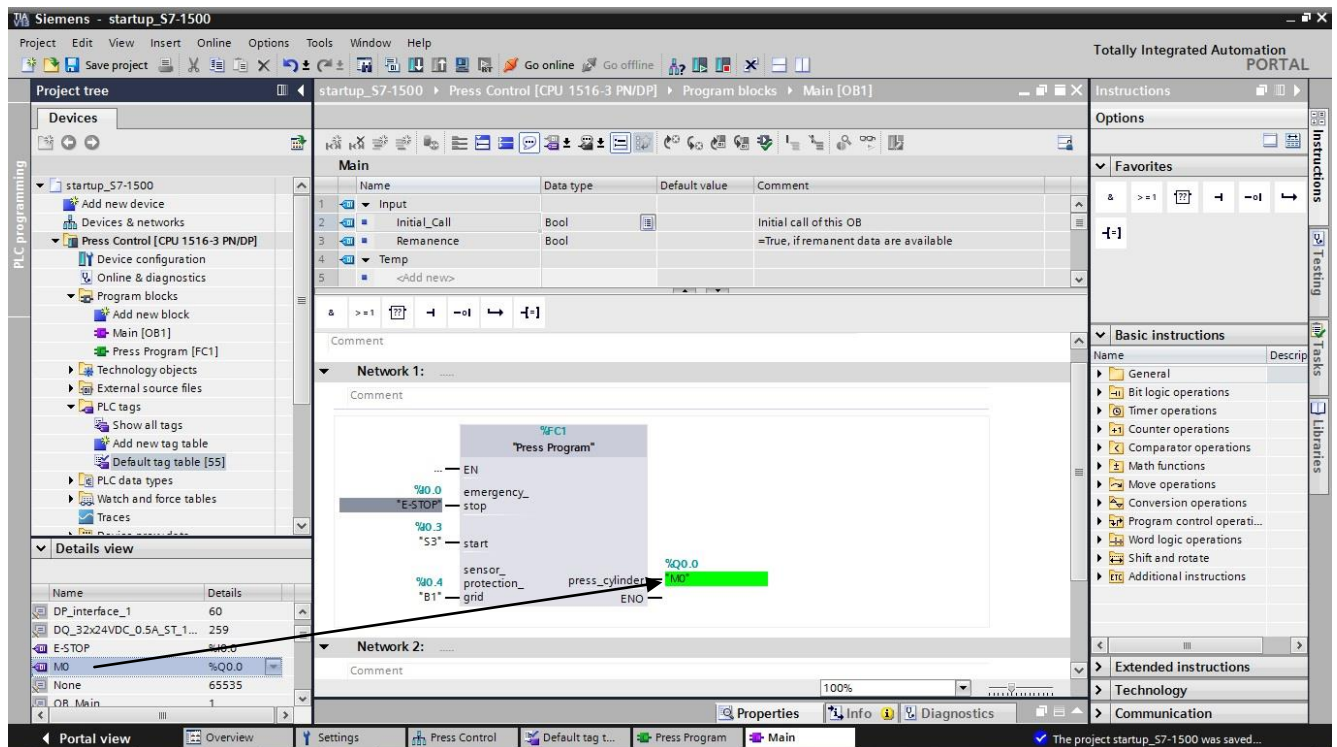
32. As mentioned previously, the block “Press Program” has to be called from the program block Main[OB1]. Otherwise, the block would not be processed at all. Open this block with a double click on **‘Main[OB1]’**. (→ Main[OB1])



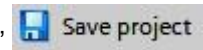
33. Now, just drag the block **"Press Program"** to Network 1 of the Main[OB1]. Don't forget to document the networks also in the Main[OB1]. (→ Press Program [FC1])



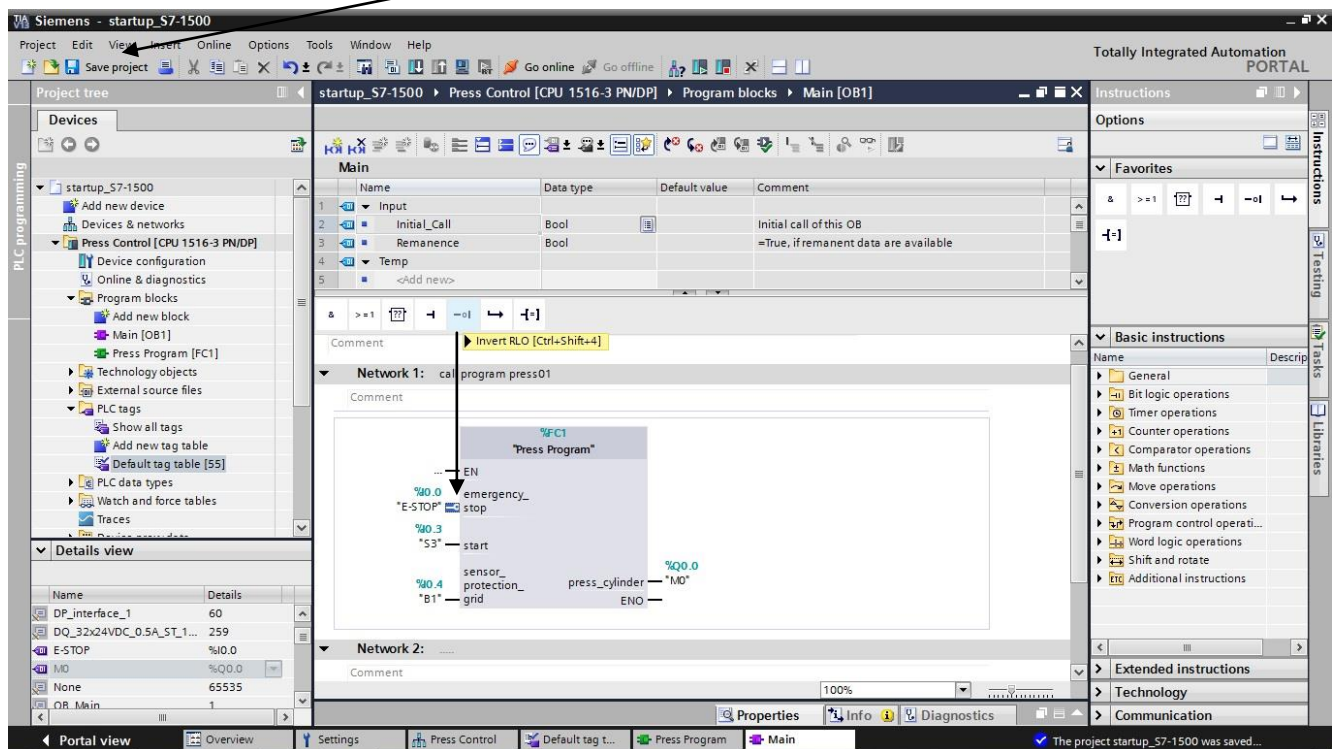
34. Next, the interface parameters of the block “Press Program” have to be connected to global PLC tags. It is sufficient to select the default tag table. From the detail view, we can now drag the desired operand to the block’s connection. (→“E_STOP → "S3" → "B1" → "M0”)





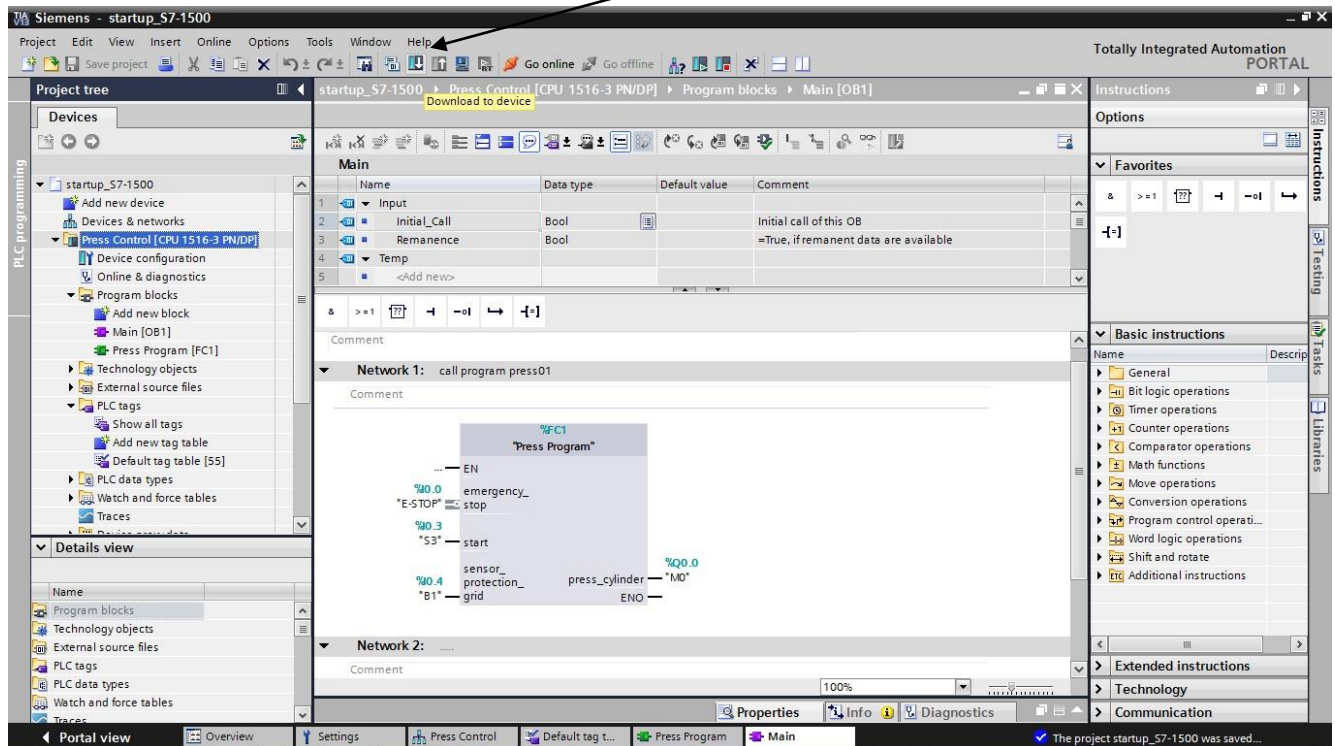
35. The PLC tag “E_STOP” is polled negated. Operating the button ‘Save project’ saves the project.



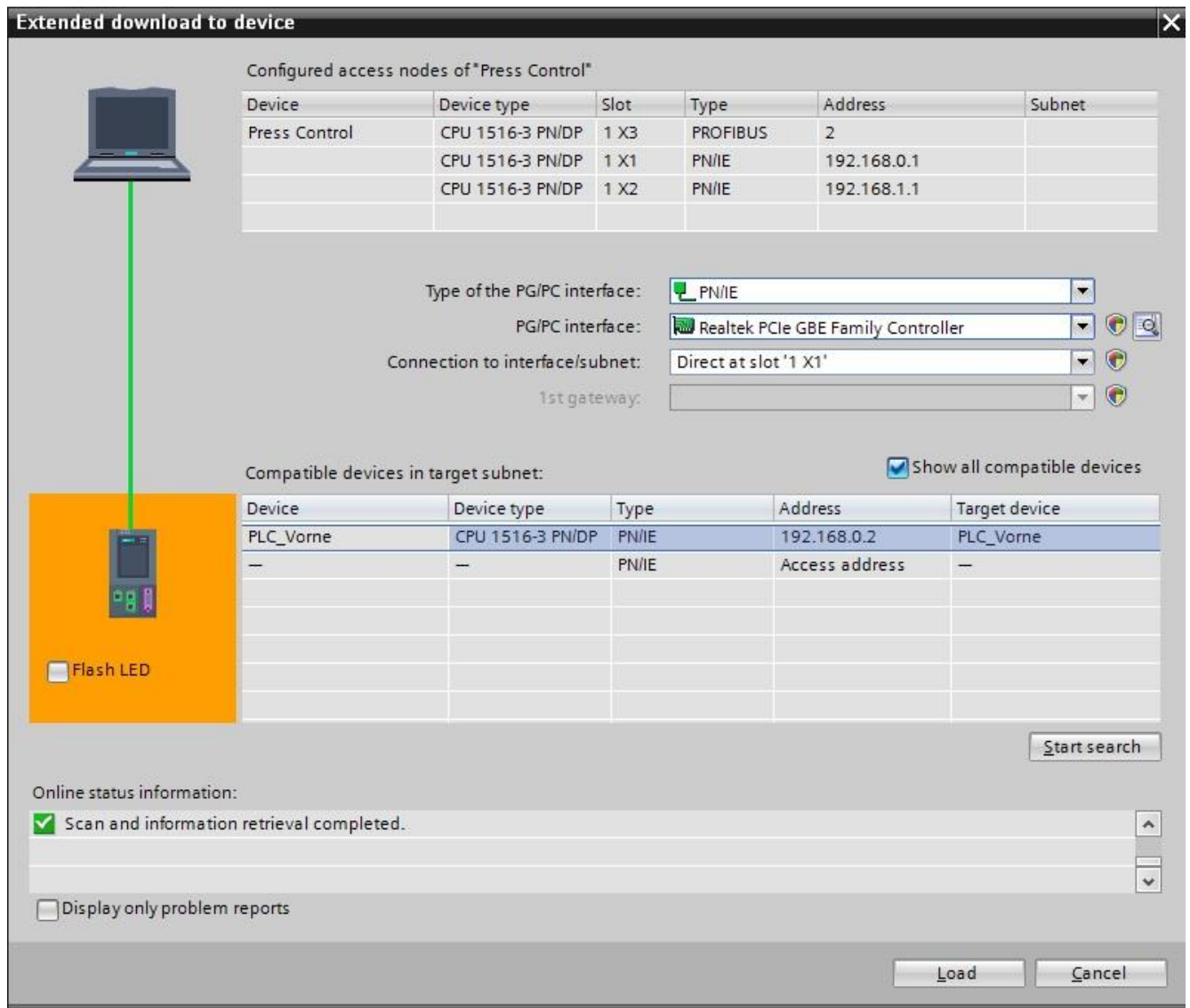
(“E_STOP” →  → Save project )



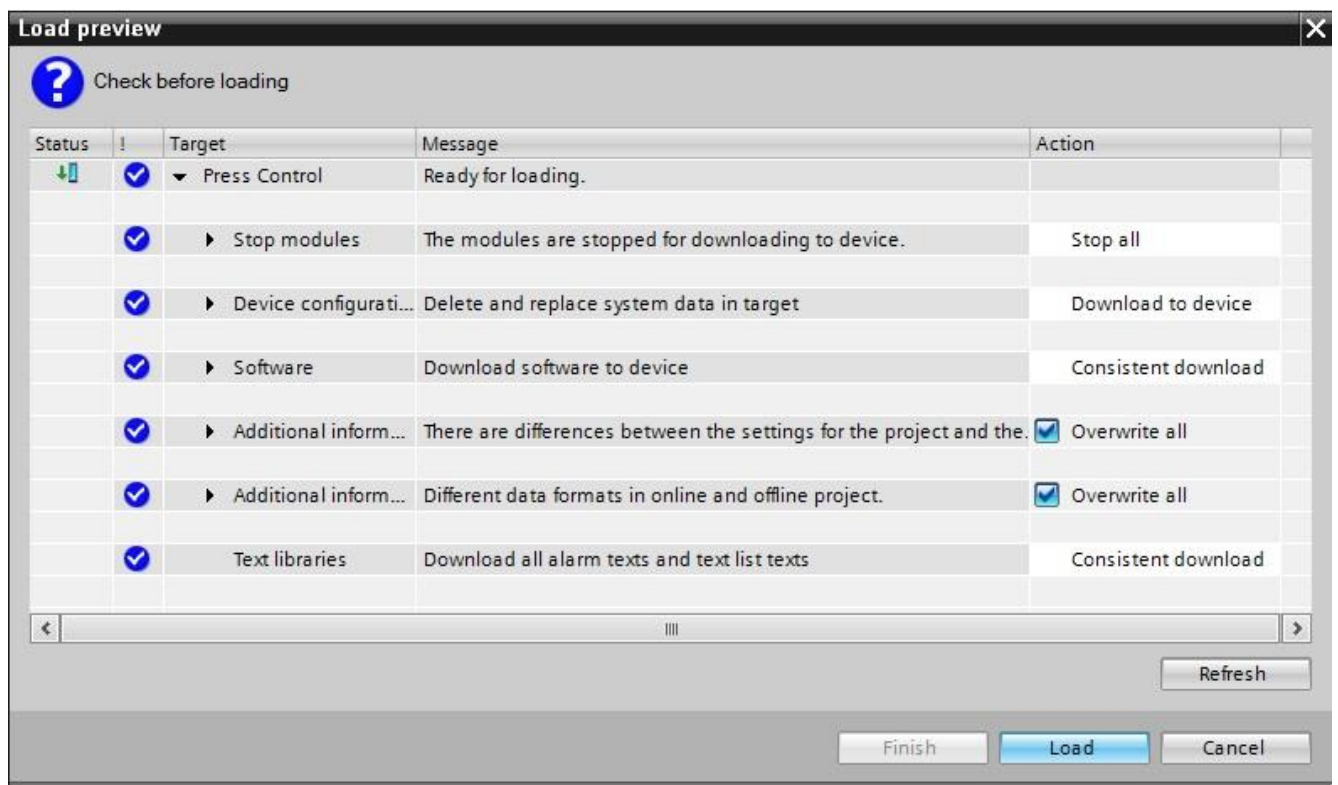
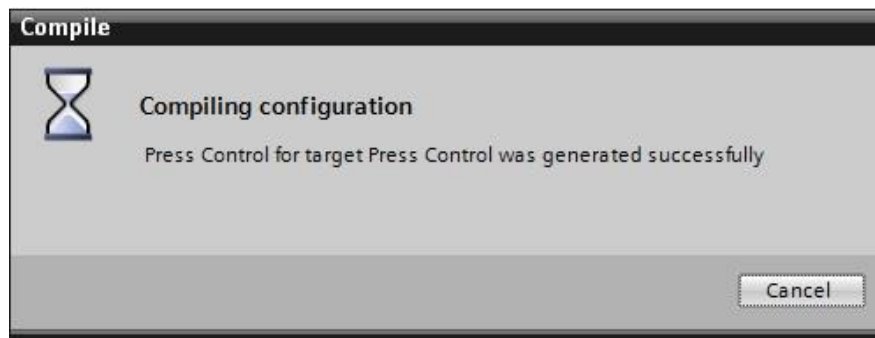
36. To download the entire program to the CPU, first highlight the folder **'Press Control'** and then click on the symbol  Download to device. (→ Press Control → )



37. In the following dialog, select '**PN/IE**' as PG/PC interface type, then the network card that was set previously as PG/PC interface, and '**X1**' as the connection of the CPU to the subnetwork. After '**Start Search**' of the accessible stations, you should see your '**CPU 1516-3 PN/DP**' with the address 192.168.0.1 and select it as the target device. Now, click on '**Load**'. (→ Type of PG/PC interface: PN/IE → PG/PC interface: → Connection to subnet: Direct at slot '1 X1' → Start Search → CPU 1516-3 PN/DP → Load)



38. The configuration is now compiled automatically. Prior to loading, an overview for checking the steps to be performed is displayed once more. Start them with '**Load**'. (→ Load)





- Load results**

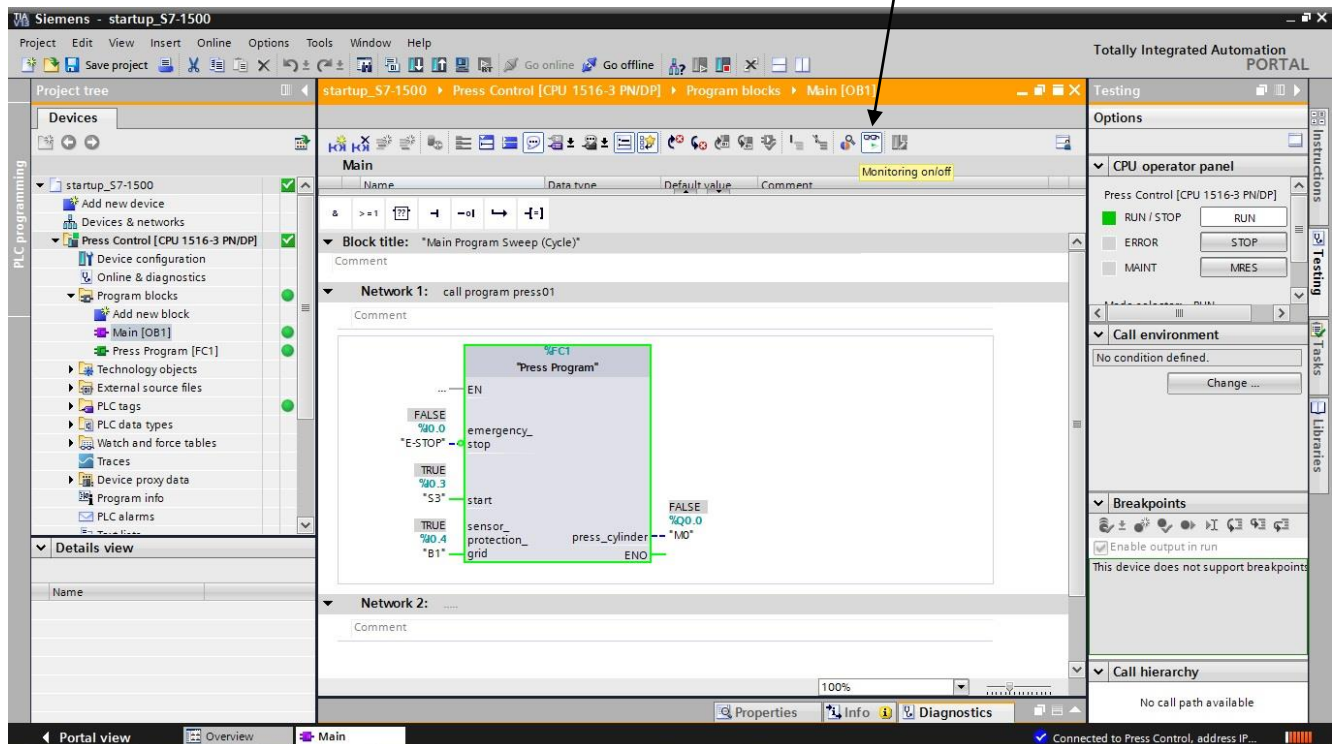
? Status and actions after downloading to device

Status	Target	Message	Action
↓	✓ ⚠ Press Control	Downloading to device completed without error.	
⚠	▶ Start modules	Start modules after downloading to device.	<input checked="" type="checkbox"/> Start all

< ||| >

Finish Load Cancel

40. By clicking on the symbol  Monitoring on/off, you can view the status of the input and output tags at the block "Press Program" during program testing. (→ )



With a right mouse click, the function "Press Program" can be opened and monitored.

