University Grenoble Alpes
PHITEM

# Getting started with TIA PortalV15.1

Autor : KAMMAH Oumaima

Supervised by Asean Factori 4.0 team : -Prof. JM THIRIET

-Prof. E NOVAKOV

-K THAROT

(G-ICS) – 18.03.2022, Siemens, Asean-Factori

# Contents

# List of Figures

# List of Tables

# 1 Objectives

- Getting started with the work environment (PLC+emulator+software)
- Practice By programming some basic examples with LADDER.
- Be prepared to handle a control system project and to analyse the communication between PLC and IHM/SCADA

# 2 TIA Selection Tool

The TIA Selection Tool allows us to select, configure and control devices for Totally Integrated Automation It is the successor to the SIMATIC Selection Tool and combines the well-known automation technology configurators in a single tool. With the TIA Selection Tool, we can generate a complete order list from our product selection or product configuration.

### PLC Simulation in TIA Portal

In this guide manual we will show you how to use PLC Simulation in TIA Portal with PLC SIM. We'll be using S7-1500 PLC in TIA Portal for simulation.

Siemens offers **S7-PLCSIM** Software along with TIA Portal that is used to create a virtual PLC using which we can test and simulate our programs.
After creating your first program in ladder logic or FBD in TIA Portal, you can run the PLCSIM software and start simulation.
The main objective of S7-PLCSIM is to contribute to the error finding and validation of a single of a single PLC program without the need for hardware. Otherwise we can load our program directly into the physical PLC and use a remote control interface **(GICS Tester)** implemented on the "configuration" machine (10.10.4.x) to simulate.

# 3 Architecture

-CPU
-Network Interface

**PLC (Programmable Logic Controller)** The programmable logic controller (PLC) uses the S7 program to monitor and control a machine or process. In doing so, the S7 program scans the I/O modules via input addresses (%I) and addresses their output addresses (%Q). The system is programmed with the STEP 7 Professional V15 software.

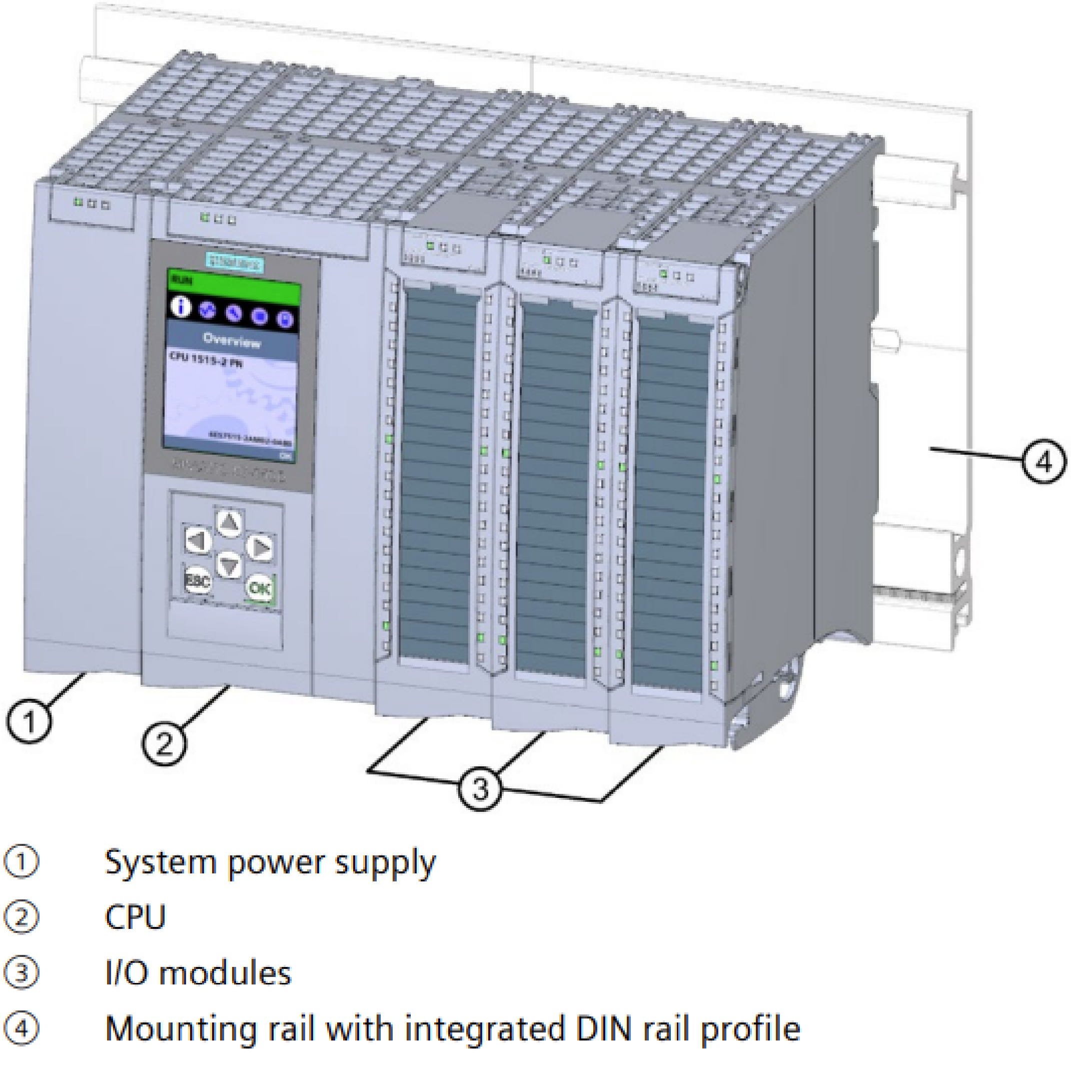


Figure 2: PLC architecture including: Power Unit; Central Unit+ Integrated PROFIBUS and PROFINET interfaces ; Digital Input/Digital Output; Analog Input/Analog Output

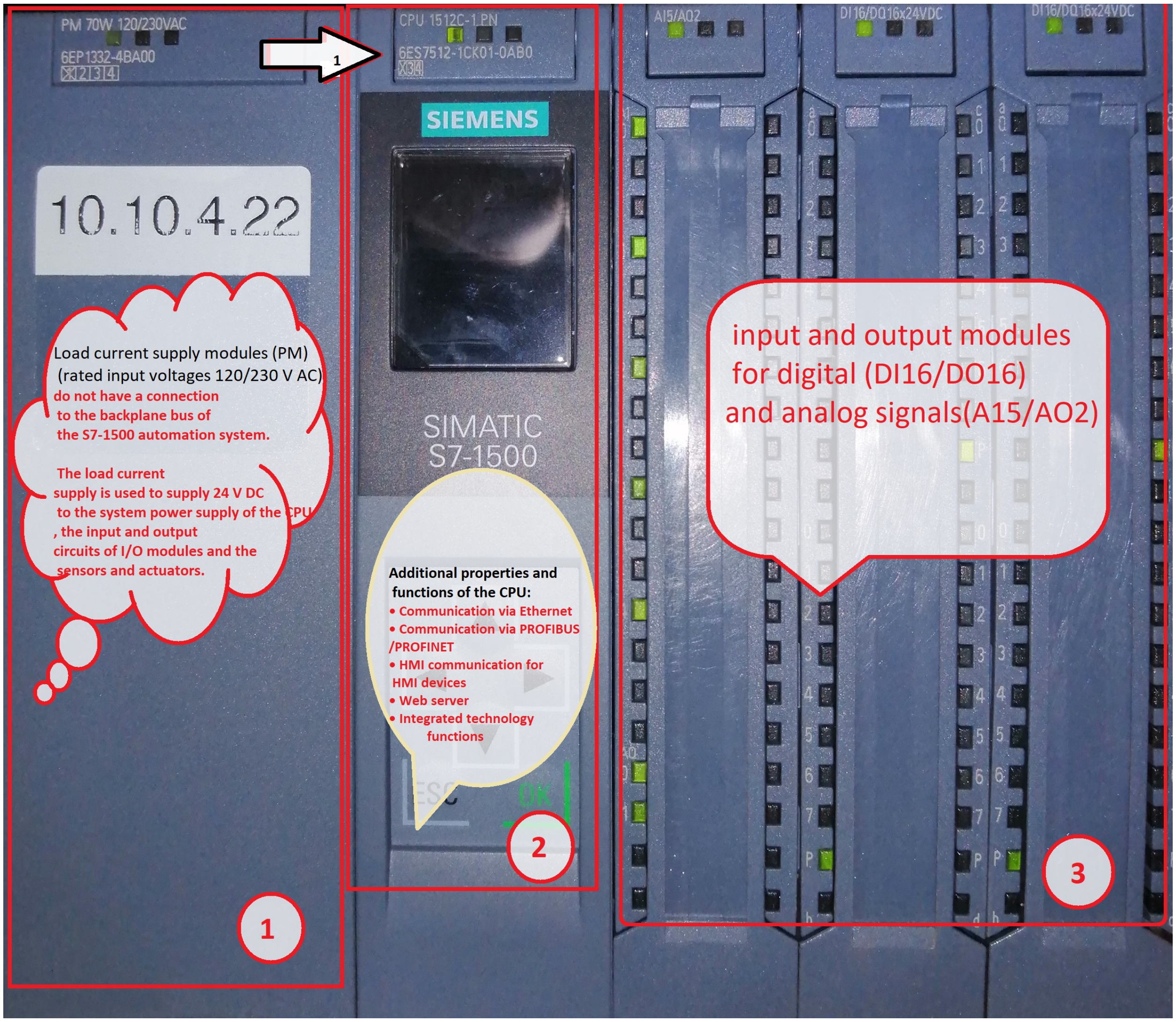

PM 70W 120/230VAC
6EP1332-4BA00
10.10.4.22
1
CPU 1512C-1 PN
6ES7512-1CK01-0AB0
SIEMENS
SIMATIC S7-1500
AI5/AQ2
DI16/DQ16x24VDC
DI16/DQ16x24VDC
Load current supply modules (PM) (rated input voltages 120/230 V AC) do not have a connection to the backplane bus of the S7-1500 automation system.
The load current supply is used to supply 24 V DC to the system power supply of the CPU , the input and output circuits of I/O modules and the sensors and actuators.
Additional properties and functions of the CPU: • Communication via Ethernet • Communication via PROFIBUS /PROFINET • HMI communication for HMI devices • Web server • Integrated technology functions
input and output modules for digital (DI16/DO16) and analog signals(A15/AO2)
1
2
3

Figure 3: The S7-1500 controller consists of a power supply **(1)**,a CPU with integrated display**(2)**,and input and output modules for digital and analog signals**(3)**

# 4 The Emulation card

There are some connections between the architecture and I/O with the emulation card.

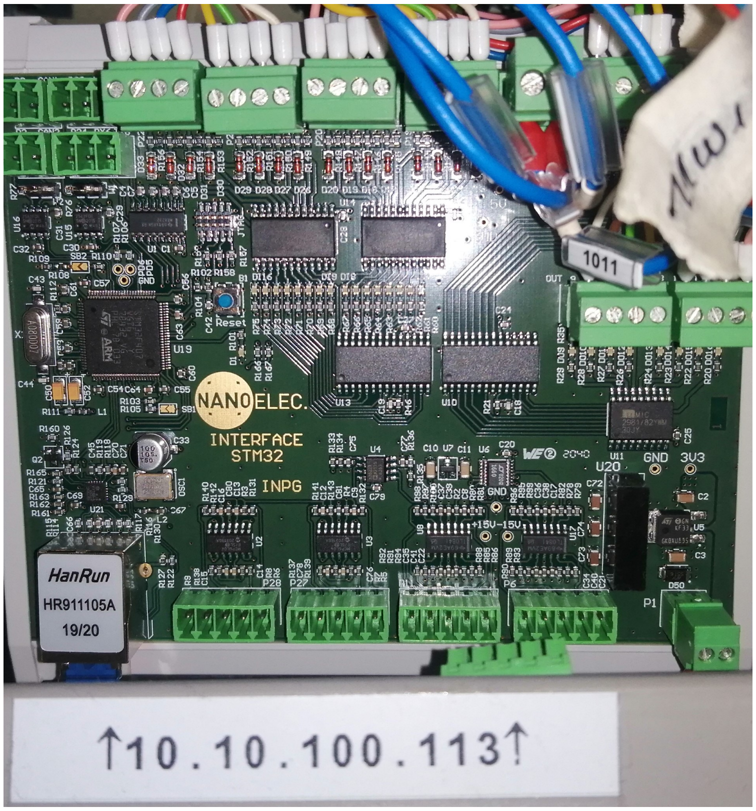


Figure 4: Emulation card

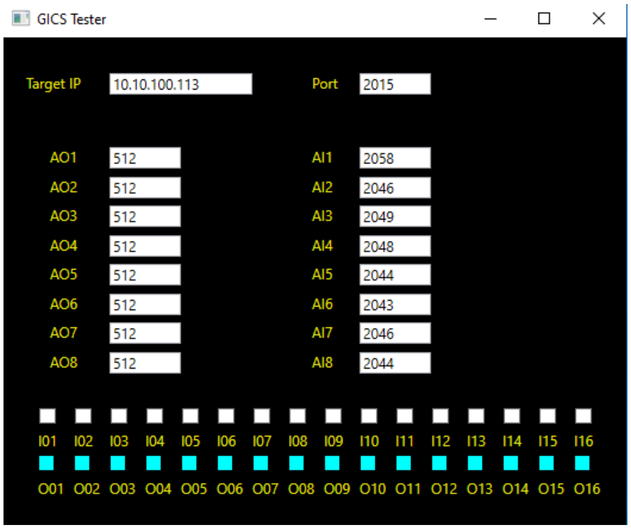
## 5 GICS-Tester

**GICS-Tester** is an interface allowing to interact remotely with an Emulation-card.
The Emulation card (IP 10.10.100.113 for instance) is a card allowing to emulate a physical process, and the inputs-outputs (both analog and digital) are physically connected to a PLC (ex 10.10.4.21).
So the PLC receives physical signals coming from the Emulation cards, uses these signals for the PLC programme, then sent physically its outputs back to the Emulation cards. When using GICS tester, we just control manually the inputs, which are sent from the computer (ex: 10.10.3.4) to the emulation cards through the network (port 2015 is used).
For other applications, it is possible to embed a program on the Emulation card and use another program for the interface with it.

As table 1 , we can check the IP address from the emulation card and the siemens's PLC inside the platform G-ICS(Industrial Control systems Sandbox)

| Emulation Card | Siemens's PLC |
|---|---|
| 10.10.100.114 | 10.10.4.21 |
| 10.10.100.113 | 10.10.4.22 |
| 10.10.100.115 | 10.10.4.23 |
| 10.10.100.116 | 10.10.4.24 |

Table 1: **Corresponding between the emulation card and Siemens's PLC inside GICS platform**



GICS Tester

Target IP: 10.10.100.113
Port: 2015

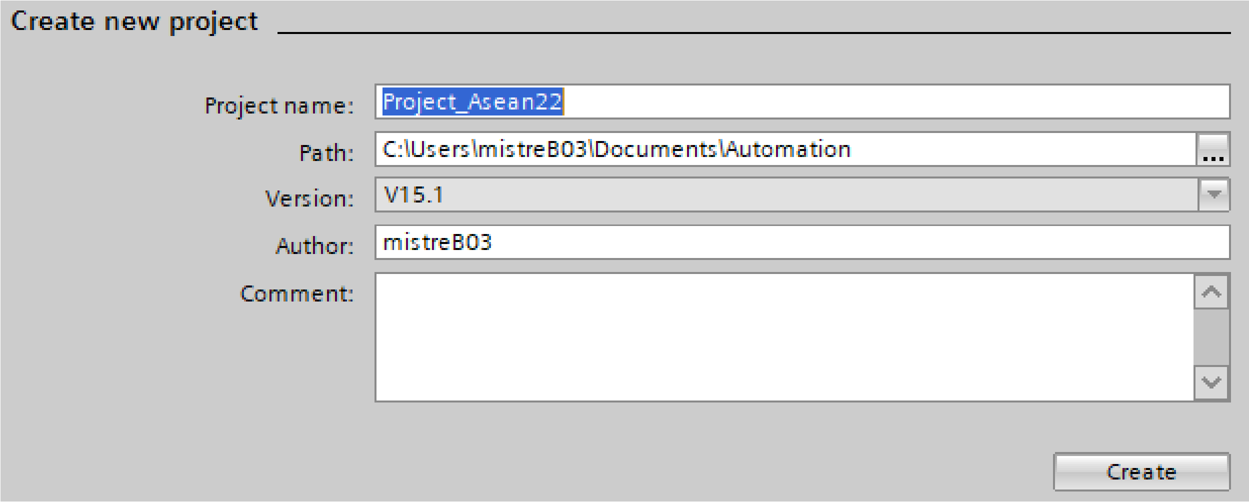
| | | | |
|---|---|---|---|
| AO1 | 512 | AI1 | 2058 |
| AO2 | 512 | AI2 | 2046 |
| AO3 | 512 | AI3 | 2049 |
| AO4 | 512 | AI4 | 2048 |
| AO5 | 512 | AI5 | 2044 |
| AO6 | 512 | AI6 | 2043 |
| AO7 | 512 | AI7 | 2046 |
| AO8 | 512 | AI8 | 2044 |

I01 I02 I03 I04 I05 I06 I07 I08 I09 I10 I11 I12 I13 I14 I15 I16
O01 O02 O03 O04 O05 O06 O07 O08 O09 O10 O11 O12 O13 O14 O15 O16

Figure 5: Step 1 :GICS Tester interface

## 6 Getting Started to create a project

→ Double click on TIA PortalV15 icon →Double click on create a new project



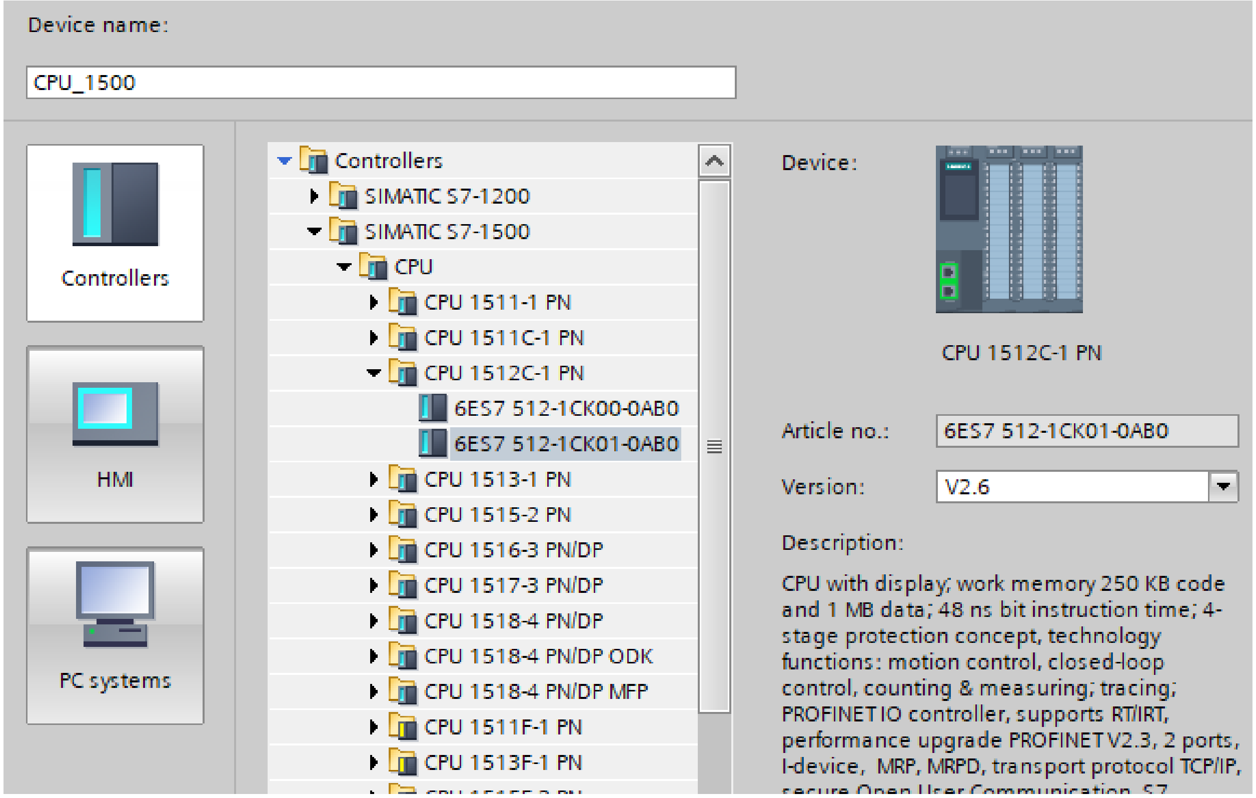
Create new project

Project name: Project_Asean22
Path: C:\Users\mistreB03\Documents\Automation
Version: V15.1
Author: mistreB03
Comment:

Figure 6: Step 1 :Create project

**Read the hardware of the SIMATIC S7-1500**

→ In the Start" portal, select → "First steps" → "Devices  Networks" → "Configure a device".
→ Switch to the "Add new device" menu.
→ Define a device name → Controllers → SIMATIC S7-1500 → CPU (specified in physical CPU unit figure 2) 6ES75XX-XXXXX-XXXX → Version V2.6


Device name: CPU_1500

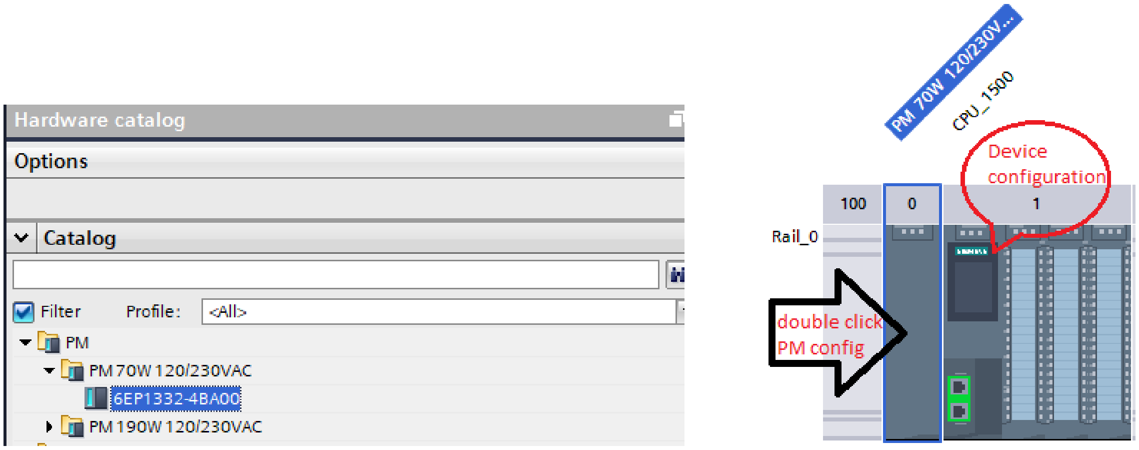
Article no.: 6ES7 512-1CK01-0AB0

Version: V2.6

Description:
CPU with display; work memory 250 KB code and 1 MB data; 48 ns bit instruction time; 4-stage protection concept, technology functions: motion control, closed-loop control, counting & measuring; tracing; PROFINET IO controller, supports RT/IRT, performance upgrade PROFINET V2.3, 2 ports, I-device, MRP, MRPD, transport protocol TCP/IP, secure Open User Communication, S7

Figure 7: Step 2 :Insert a new device and create the CPU with a double-click SIMATIC S7-1500 then CPU then 6ES7 512-1CK01-0AB0 then Add

The power module is not automatically detected and must be manually added from the physical PLC.
→ Insert the power module into slot 0→ In Hardware catalog double click on PM → Physical reference of Power Unit (figure2).



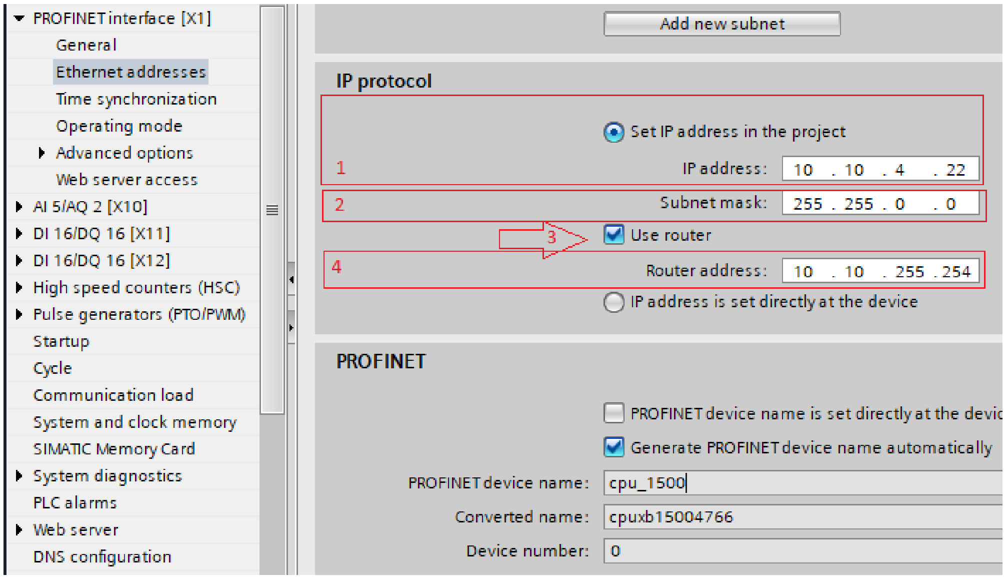
(a) Insert power Module(PM 70W 120:230V)

(b)

Figure 8: Step 3 :Hardware catalog

# 7 Network Configuration

We are working on the 10.10.0.0/16 network which interconnects the following elements:
• ”Configuration” computers (10.10.4.X)
• APIs
• ”Process simulators” cards (10.10.100.X)
• HMIs
→ Double click on CPU_1500[CPU 1512C-1 PN] → Select Device configuration or Double click on black rectangle on (figure 6 label (b)).→ PROFINET interface[X1] → Ethernet addresses then configure IP protocol



IP protocol

● Set IP address in the project

IP address: 10 . 10 . 4 . 22

Subnet mask: 255 . 255 . 0 . 0

☑ Use router

Router address: 10 . 10 . 255 . 254

○ IP address is set directly at the device

PROFINET

□ PROFINET device name is set directly at the devic

☑ Generate PROFINET device name automatically

PROFINET device name: cpu_1500

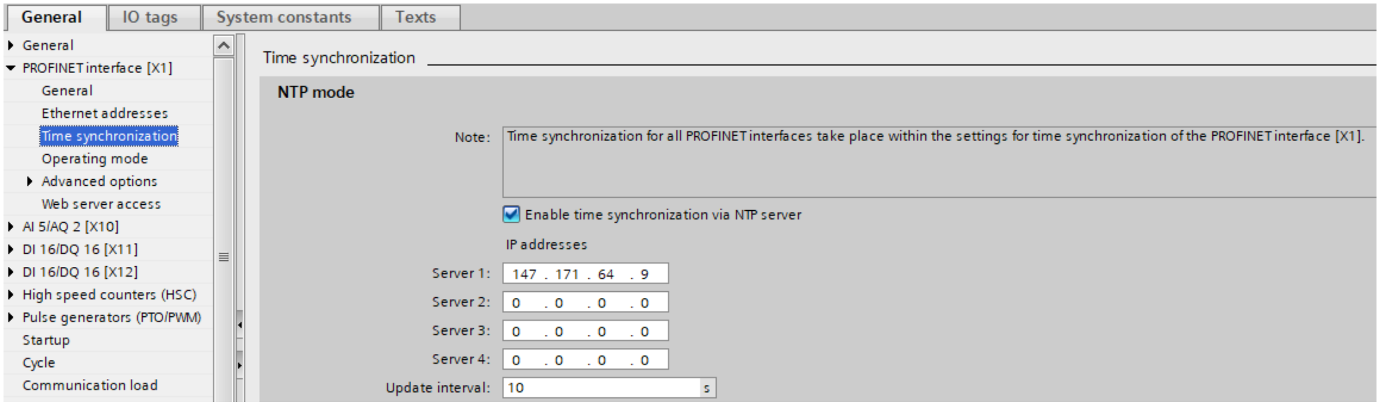
Converted name: cpuxb15004766

Device number: 0

Figure 9: Step 4 : Set network

In order to have a good synchronization in the architecture created, we need to add the reference clock of Grenoble INP which has the IP address 147.171.64.9.
To properly add this clock,we set web access since normally the API is protected against unauthorized access and this leads to not to accept web access. However, the Grenoble INP clock is in the network and it must be accessed to synchronize our system.(figure 8)

→ Time synchronization **(Figure 10)**→ Enable time synchronization via NTP server → Set IP adresse for server 1.



Time synchronization

NTP mode

Note: Time synchronization for all PROFINET interfaces take place within the settings for time synchronization of the PROFINET interface [X1].

☑ Enable time synchronization via NTP server

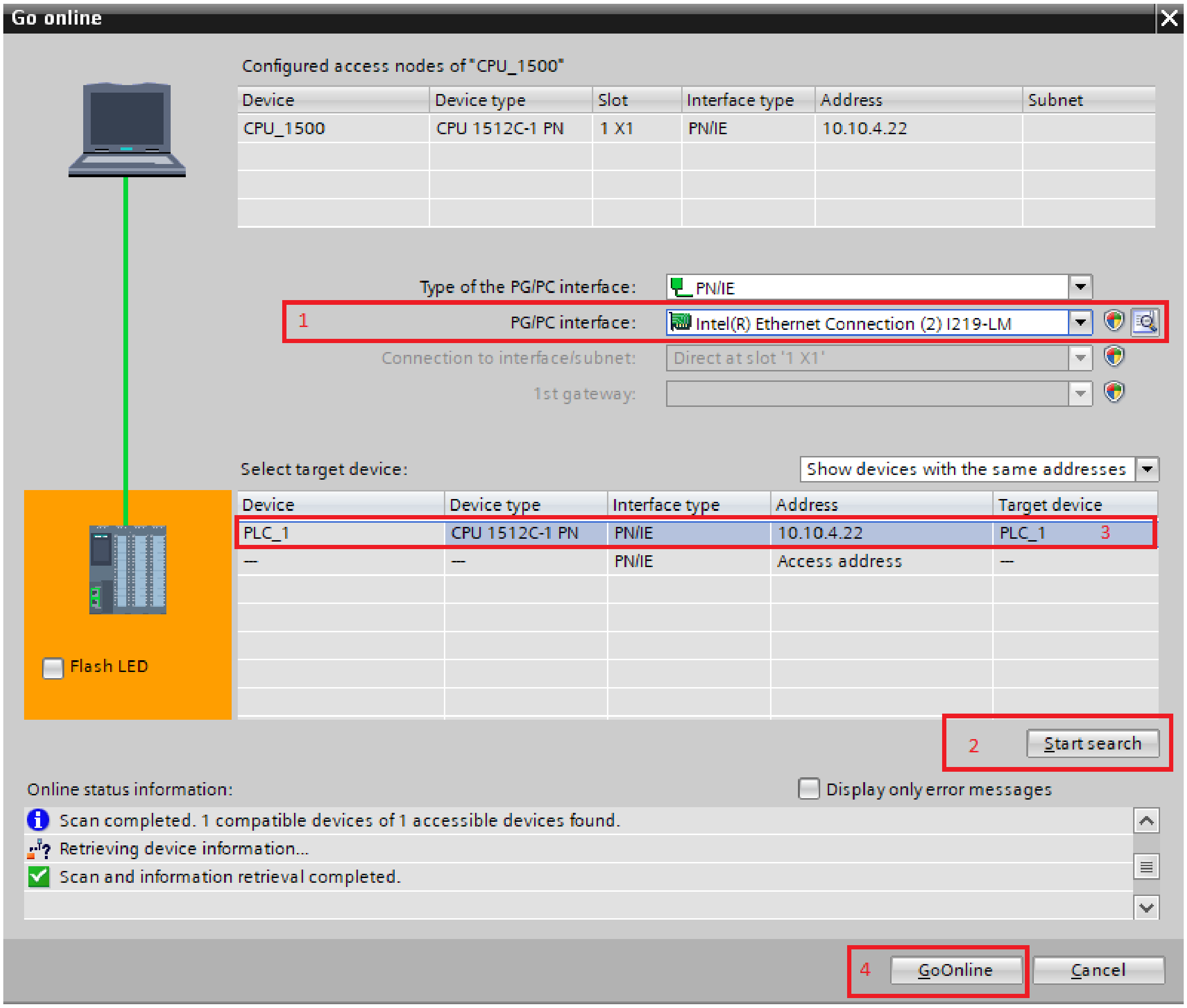
IP addresses

Server 1: 147 . 171 . 64 . 9
Server 2: 0 . 0 . 0 . 0
Server 3: 0 . 0 . 0 . 0
Server 4: 0 . 0 . 0 . 0
Update interval: 10 s

Figure 10: Step 5 : Time Synchronization

### Download the hardware configuration to the device

→To download your entire CPU, select the "CPU_1500 [CPU1516F-3 PN/DP]" folder → click the "Download to device" icon.(Figure 13 Blue).
→The manager for configuring the connection properties (extended download) opens.



Configured access nodes of "CPU_1500"

| Device | Device type | Slot | Interface type | Address | Subnet |
|---|---|---|---|---|---|
| CPU_1500 | CPU 1512C-1 PN | 1 X1 | PN/IE | 10.10.4.22 | |

Type of the PG/PC interface: PN/IE

PG/PC interface: Intel(R) Ethernet Connection (2) I219-LM

Connection to interface/subnet: Direct at slot '1 X1'

1st gateway:

Select target device: Show devices with the same addresses

| Device | Device type | Interface type | Address | Target device |
|---|---|---|---|---|
| PLC_1 | CPU 1512C-1 PN | PN/IE | 10.10.4.22 | PLC_1 |
| — | — | PN/IE | Access address | — |

□ Flash LED

Online status information:

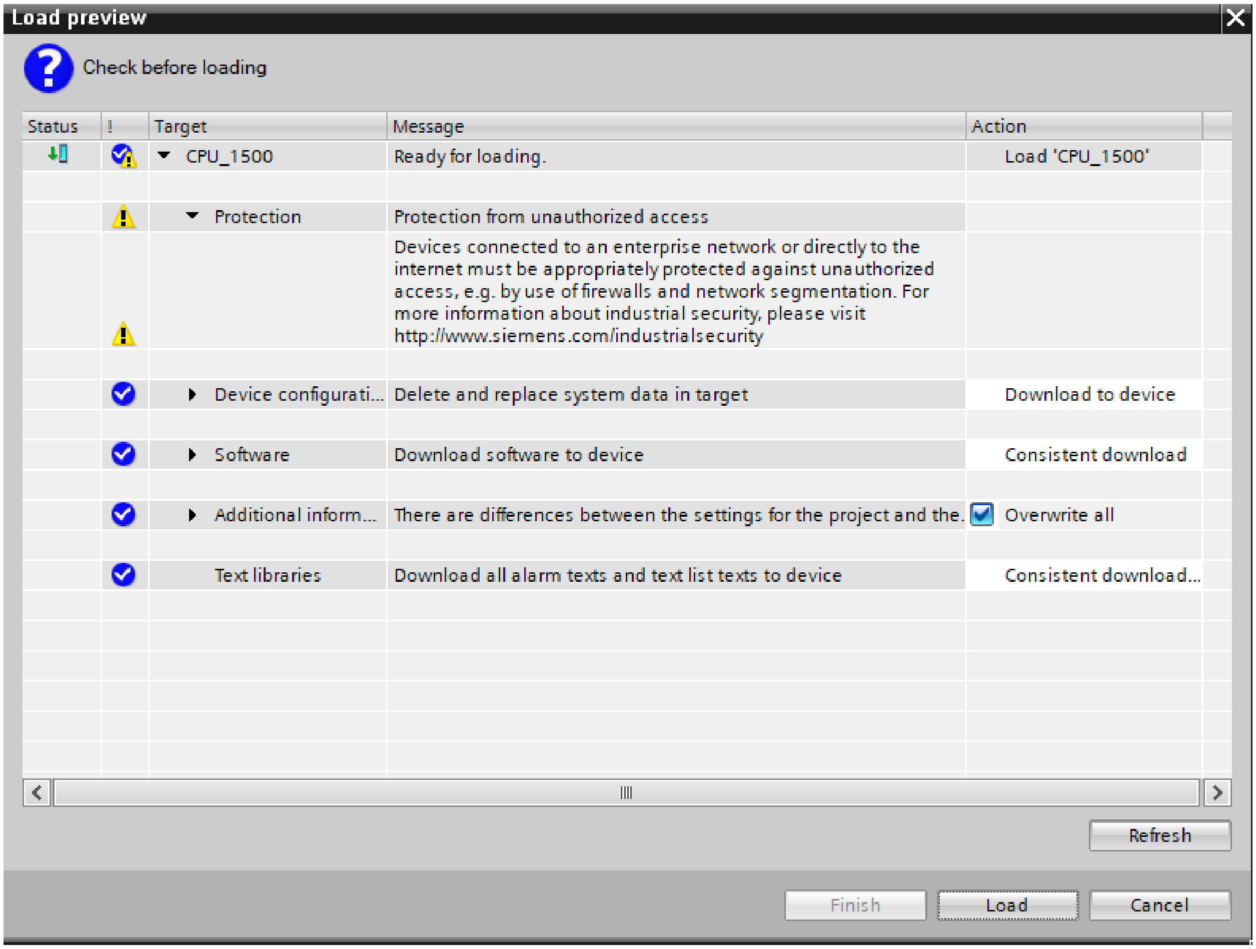
□ Display only error messages

Scan completed. 1 compatible devices of 1 accessible devices found.
Retrieving device information...
Scan and information retrieval completed.

Figure 11: Step 6 Hardware detection for CPU_1500

**Step 1:** Select the type of your PG/PC interface. (Type of the PG/PC interface.Then you select the network adapter you want to use to establish an Ethernet connection to the PLC. (PG/PC interface: Intel(R) Ethernet Connection I219-LM)
**Step 2:** The search for devices in the network must be started by clicking the button.
**Step 3:** All accessible nodes are found and listed. If you have selected the correct CPU, the corresponding CPU and all the connected modules will be detected when you click "GoOnline" **(Step 4)**.

→You first obtain a preview → Overwrite all should be selected→ Double click on load.



Load preview

Check before loading

| Status | ! | Target | Message | Action |
|---|---|---|---|---|
|  | ⚠ | CPU_1500 | Ready for loading. | Load 'CPU_1500' |
|  | ⚠ | Protection | Protection from unauthorized access |  |
|  | ⚠ |  | Devices connected to an enterprise network or directly to the internet must be appropriately protected against unauthorized access, e.g. by use of firewalls and network segmentation. For more information about industrial security, please visit http://www.siemens.com/industrialsecurity |  |
|  | ✓ | Device configurati... | Delete and replace system data in target | Download to device |
|  | ✓ | Software | Download software to device | Consistent download |
|  | ✓ | Additional inform... | There are differences between the settings for the project and the. | ☑ Overwrite all |
|  | ✓ | Text libraries | Download all alarm texts and text list texts to device | Consistent download... |

Figure 12: Step 7: Load preview

**Save and compile the hardware configuration** → Before you compile the configuration, you should save your project by clicking "Save project" (figure 11 yellow). To compile your CPU with the device configuration, first select the "CPU_1500 [CPU1516F-3 PN/DP]" folder and click the → "Compile" icon(figure 11 red).

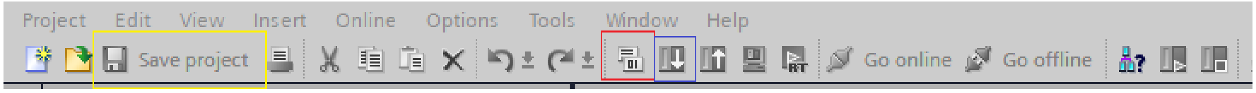
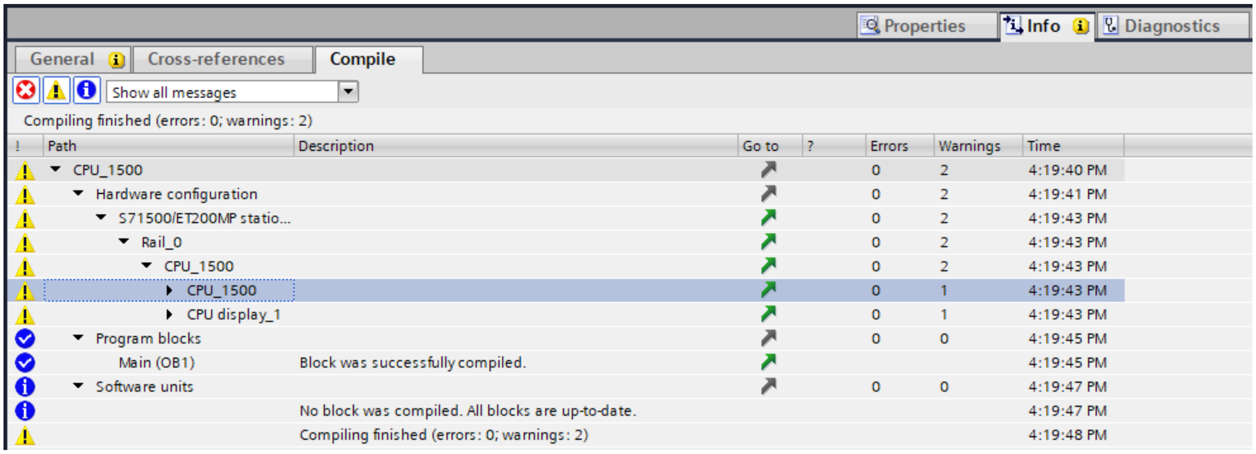


Figure 13: Save(yellow)→ Compile(Red)→Download(Blue)

**Note:** "Save project" should be used repeatedly when working on a project since this does not happen automatically.

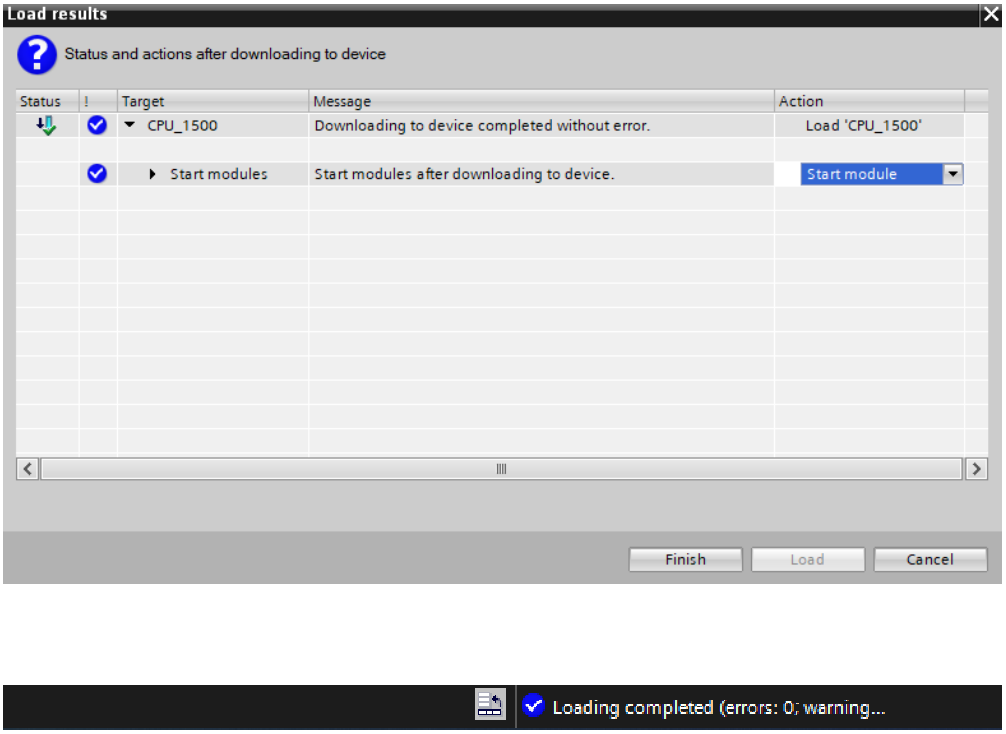


Compiling finished (errors: 0; warnings: 2)

| ! | Path | Description | Go to | ? | Errors | Warnings | Time |
|---|---|---|---|---|---|---|---|
| ⚠ | CPU_1500 | | ↗ | | 0 | 2 | 4:19:40 PM |
| ⚠ | Hardware configuration | | ↗ | | 0 | 2 | 4:19:41 PM |
| ⚠ | S71500/ET200MP statio... | | ↗ | | 0 | 2 | 4:19:43 PM |
| ⚠ | Rail_0 | | ↗ | | 0 | 2 | 4:19:43 PM |
| ⚠ | CPU_1500 | | ↗ | | 0 | 2 | 4:19:43 PM |
| ⚠ | CPU_1500 | | ↗ | | 0 | 1 | 4:19:43 PM |
| ⚠ | CPU display_1 | | ↗ | | 0 | 1 | 4:19:43 PM |
| ✓ | Program blocks | | ↗ | | 0 | 0 | 4:19:45 PM |
| ✓ | Main (OB1) | Block was successfully compiled. | ↗ | | | | 4:19:45 PM |
| ⓘ | Software units | | ↗ | | 0 | 0 | 4:19:47 PM |
| ⓘ | | No block was compiled. All blocks are up-to-date. | | | | | 4:19:47 PM |
| ⚠ | | Compiling finished (errors: 0; warnings: 2) | | | | | 4:19:48 PM |

Figure 14: Compilation done correctly

Compilation done correctly no error we can ignore the two warning appeared related to absence of security
Then click on Download to device icon (figure 11 Blue) → After a successful download, the project view will open again automatically.
A loading report appears in the information field under ”General”. This can be helpful when troubleshooting an unsuccessful download.



Status and actions after downloading to device

| Status | ! | Target | Message | Action |
|---|---|---|---|---|
| ⇩ | ✓ | CPU_1500 | Downloading to device completed without error. | Load 'CPU_1500' |
| | ✓ | Start modules | Start modules after downloading to device. | Start module |

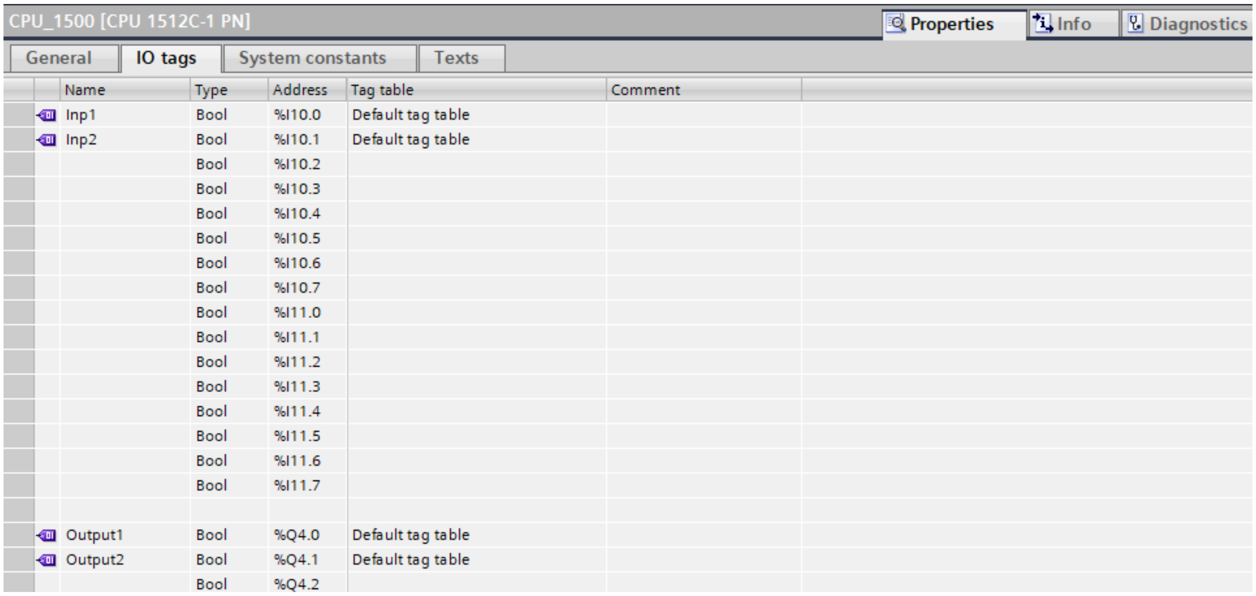
Loading completed (errors: 0; warning...

Figure 16: Load results

# 8 I/O Configuration

Programming should be in a physical because the rack is here in the lab In real industry, they will put in the box and configure from the program here;

→ Devices → Project → CPU_1500[CPU 1512C-1 PN] → Double click on Device configuration→ Click on IO tags→ Insert your Input/Output type bool



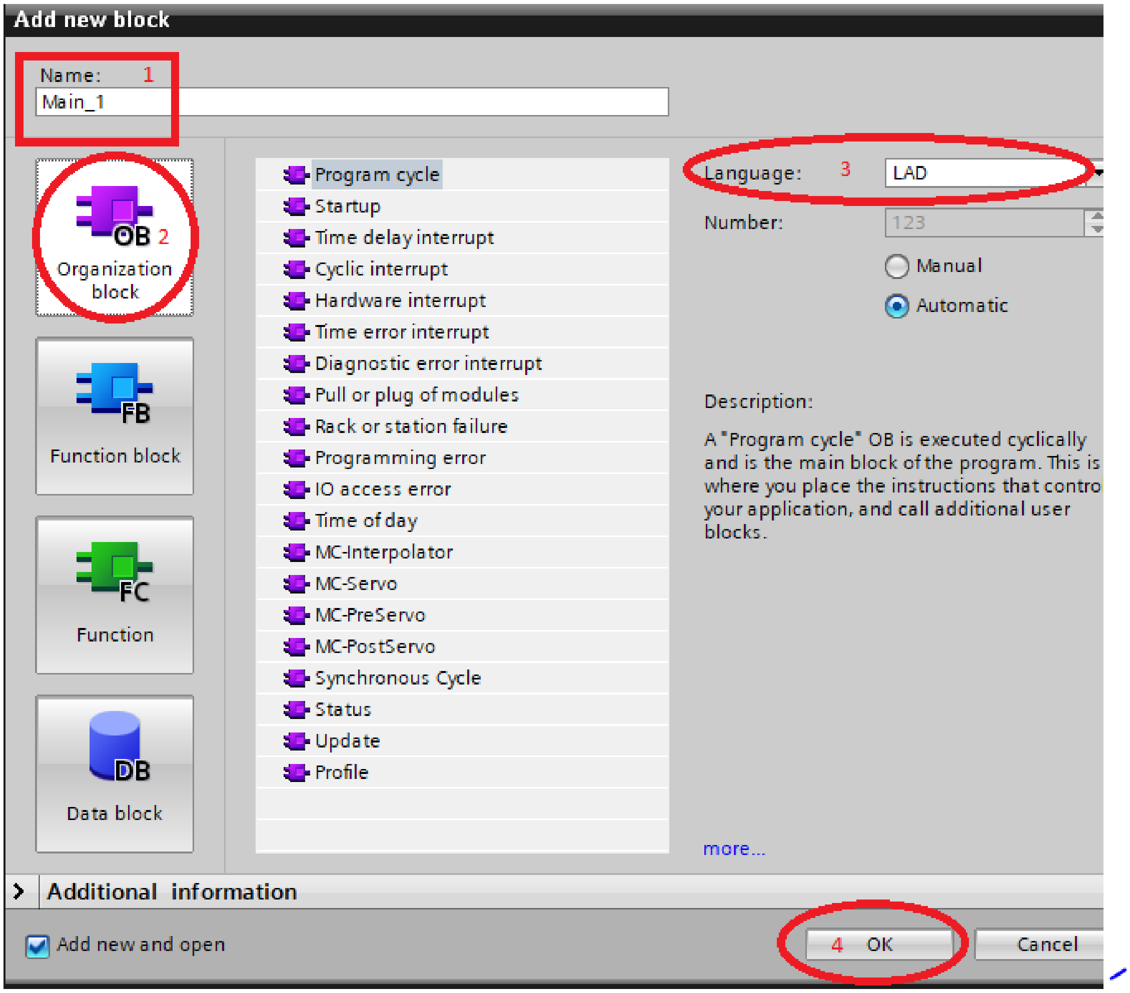
| Name | Type | Address | Tag table | Comment |
|---|---|---|---|---|
| Inp1 | Bool | %I10.0 | Default tag table | |
| Inp2 | Bool | %I10.1 | Default tag table | |
| | Bool | %I10.2 | | |
| | Bool | %I10.3 | | |
| | Bool | %I10.4 | | |
| | Bool | %I10.5 | | |
| | Bool | %I10.6 | | |
| | Bool | %I10.7 | | |
| | Bool | %I11.0 | | |
| | Bool | %I11.1 | | |
| | Bool | %I11.2 | | |
| | Bool | %I11.3 | | |
| | Bool | %I11.4 | | |
| | Bool | %I11.5 | | |
| | Bool | %I11.6 | | |
| | Bool | %I11.7 | | |
| | | | | |
| Output1 | Bool | %Q4.0 | Default tag table | |
| Output2 | Bool | %Q4.1 | Default tag table | |
| | Bool | %Q4.2 | | |

Figure 17: Create Input/Output

# 9 Programming Blocks

PLCs can be programmed in various ways. According to IEC 61131-3 standard, 5 programming languages are included we will give some examples later But in this portion of the lab we will show you how to create your first program LADDER using network.

A network is like an electrical wire, where you can place components such as contacts and assignments.
→To access the networks, drop down the menus from your PLC in the Project Tree, until you see →Program Blocks→Add new block.



Name: Main_1

Language: LAD

Number: 123

○ Manual

● Automatic

Description:

A "Program cycle" OB is executed cyclically and is the main block of the program. This is where you place the instructions that contro your application, and call additional user blocks.

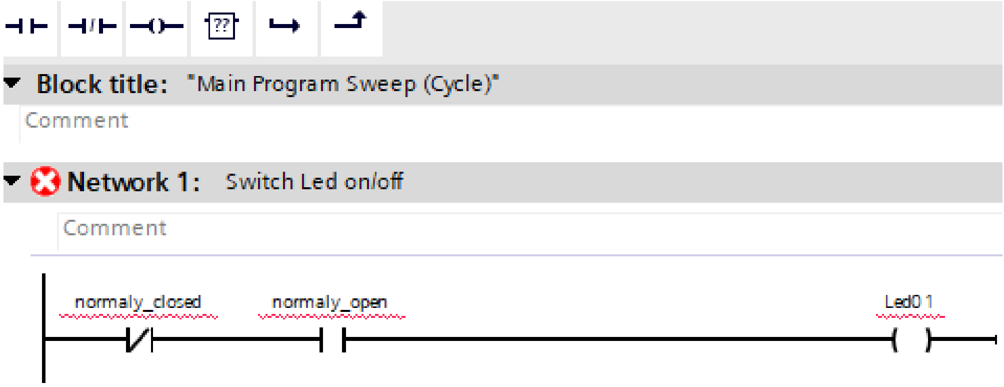
☑ Add new and open

Figure 18: New block

→ Network 1 is already made for you.→On the top of the workspace you find basic blocks(Contacts..The coil is necessary to complete the line and assign the output).

## 9.1 First Example: Programming switch on/switch OFF Led in PLC

→Begin with grabbing the normally closed contact (N/C) anc normaly open(N/O) from the favorite's bar at the top and name its.
→Grab the coil to complete the circuit Drag it on to the network line and name it "Led01".



Block title: "Main Program Sweep (Cycle)"

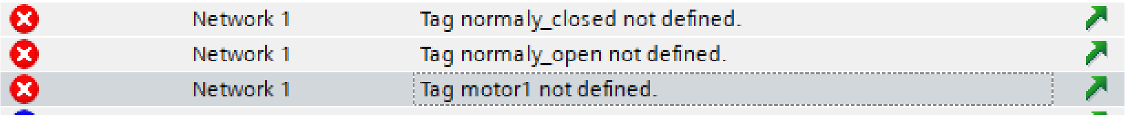
Comment

Network 1: Switch Led on/off

Comment

Figure 19: First system example

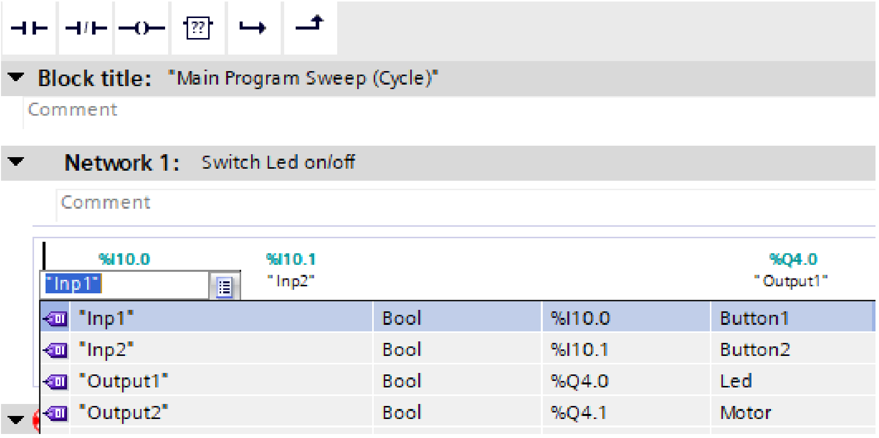
This Input/Output need to be defined otherwise we get this errors after saving and compiling:



| | | |
|---|---|---|
| ✖ | Network 1 | Tag normaly_closed not defined. |
| ✖ | Network 1 | Tag normaly_open not defined. |
| ✖ | Network 1 | Tag motor1 not defined. |

Figure 20: Errors of compilation

→Defining Inputs/Output from PLC tags (See I/O Configuration)



Block title: "Main Program Sweep (Cycle)"

Comment

Network 1: Switch Led on/off

Comment

| | | | |
|---|---|---|---|
| "Inp1" | Bool | %I10.0 | Button1 |
| "Inp2" | Bool | %I10.1 | Button2 |
| "Output1" | Bool | %Q4.0 | Led |
| "Output2" | Bool | %Q4.1 | Motor |

Figure 21: Defining from PLC tags

**Each time after creating the program ladder→Save→Compile →Download to PLC**

→ Now To go online with the controller press

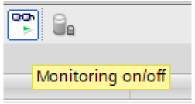
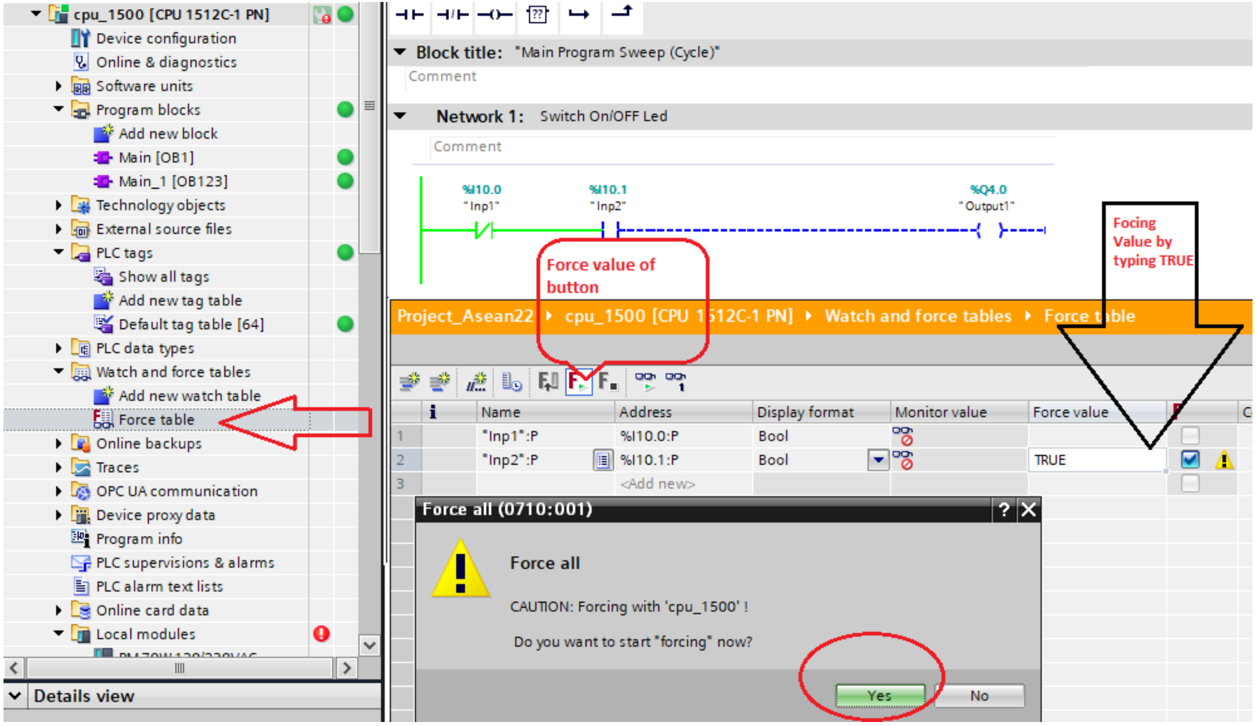


Figure 22: Monitoring On/OFF Icon

### Forcing a value

Because we are connected to the Real-World Inputs, which are designed to have push buttons connected to them. We can not turn them on because you are far away being remotely connected. We will use a command called force



Force value of button

Focing Value by typing TRUE

| | i | Name | Address | Display format | Monitor value | Force value | F |
|---|---|---|---|---|---|---|---|
| 1 | | "Inp1":P | %I10.0:P | Bool | | | □ |
| 2 | | "Inp2":P | %I10.1:P | Bool | | TRUE | ☑ |
| 3 | | | <Add new> | | | | □ |

Force all (0710:001)

Force all

CAUTION: Forcing with 'cpu_1500' !

Do you want to start "forcing" now?

Yes No

Figure 23: Force Table

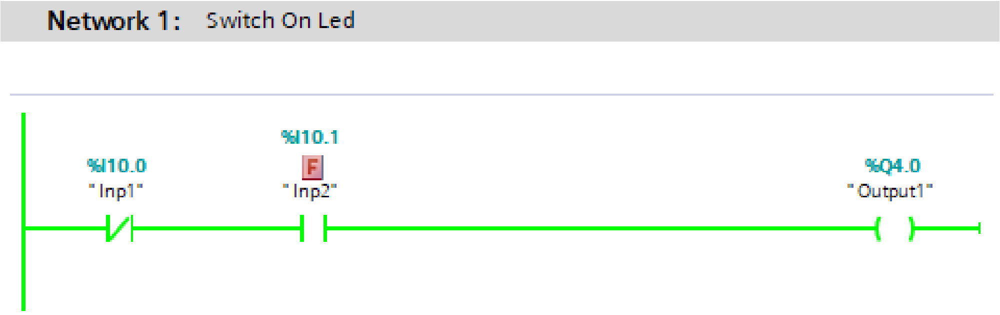
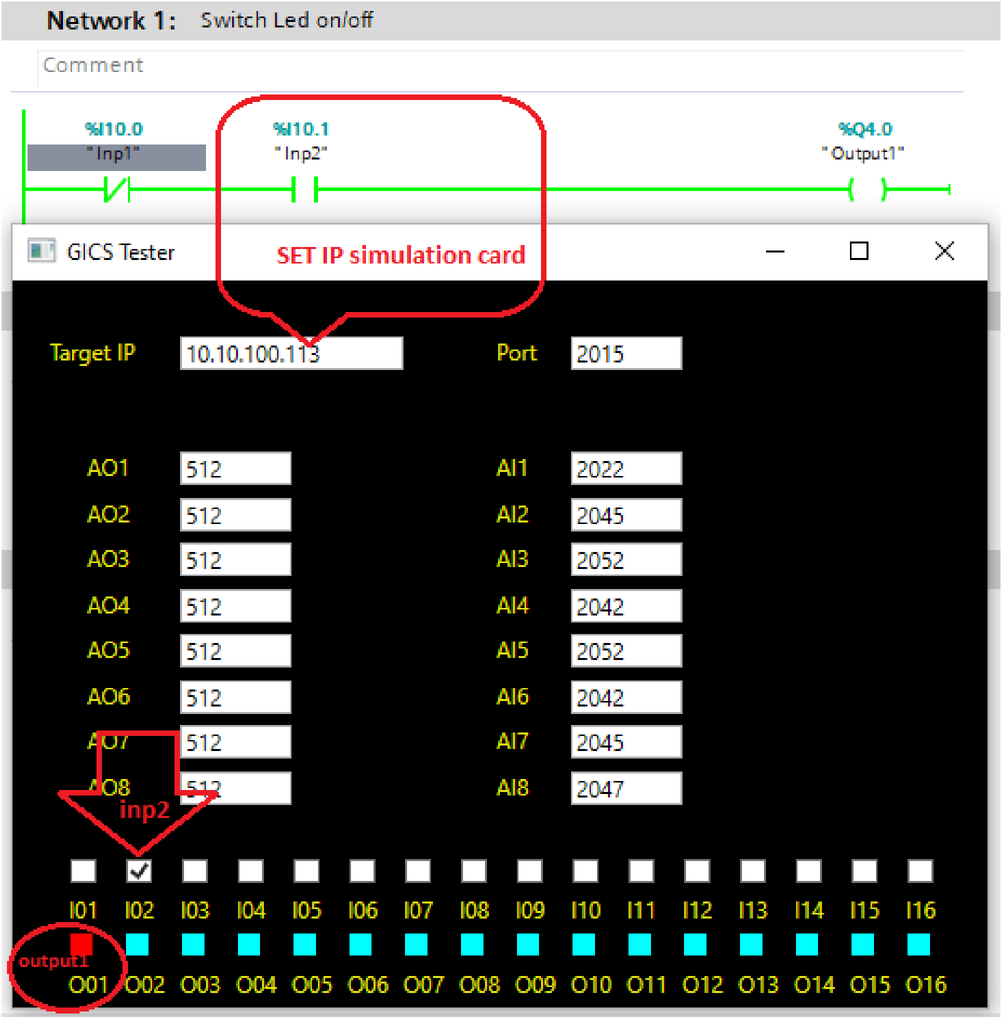


Figure 24: Example of switch on led by forcing value TRUE

### Enabling inputs by using GICS Tester and downoald to physical PLC: A remote control interface (GICS Tester) implemented on the ”configuration” machine (10.10.4.x)



Target IP 10.10.100.113 Port 2015

| | | | |
|---|---|---|---|
| AO1 | 512 | AI1 | 2022 |
| AO2 | 512 | AI2 | 2045 |
| AO3 | 512 | AI3 | 2052 |
| AO4 | 512 | AI4 | 2042 |
| AO5 | 512 | AI5 | 2052 |
| AO6 | 512 | AI6 | 2042 |
| AO7 | 512 | AI7 | 2045 |
| AO8 | 512 | AI8 | 2047 |

| I01 | I02 | I03 | I04 | I05 | I06 | I07 | I08 | I09 | I10 | I11 | I12 | I13 | I14 | I15 | I16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| □ | ☑ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| 🟥 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 |
| O01 | O02 | O03 | O04 | O05 | O06 | O07 | O08 | O09 | O10 | O11 | O12 | O13 | O14 | O15 | O16 |

Figure 25: Control with GICS Tester

**For the same example we will create a virtual PLC to test and simulate our program to get used to PLCSIM tool in case no pysical PLC is available**

→In the main OB environment we start by running PLCSIM using start simulation icon

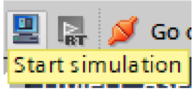
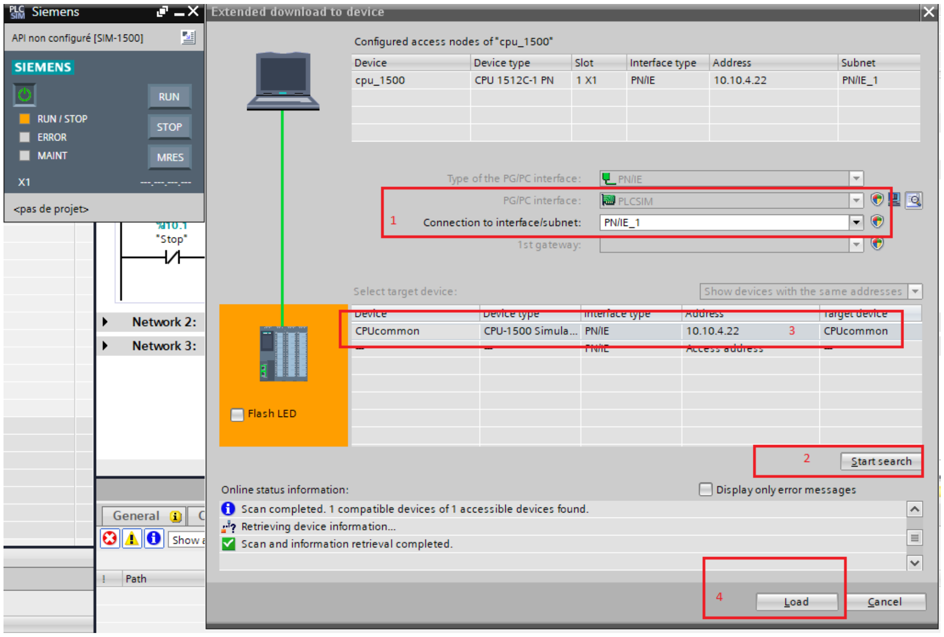


Figure 26: Start simulator Icon



Extended download to device

Configured access nodes of "cpu_1500"

| Device | Device type | Slot | Interface type | Address | Subnet |
|---|---|---|---|---|---|
| cpu_1500 | CPU 1512C-1 PN | 1 X1 | PN/IE | 10.10.4.22 | PN/IE_1 |

Type of the PG/PC interface: PN/IE

PG/PC interface: PLCSIM

Connection to interface/subnet: PN/IE_1

1st gateway:

Select target device: Show devices with the same addresses

| Device | Device type | Interface type | Address | Target device |
|---|---|---|---|---|
| CPUcommon | CPU-1500 Simula... | PN/IE | 10.10.4.22 | CPUcommon |
| — | — | PN/IE | Access address | — |

Online status information: □ Display only error messages

Scan completed. 1 compatible devices of 1 accessible devices found.
Retrieving device information...
Scan and information retrieval completed.

Figure 27: Start device search

→ Follow previous steps → Load previous window open → Run CPU by start module → Finish → Then back to simulator and click on the button as follows to create a table to force the conditions of our interrupters;

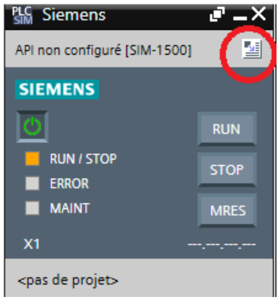
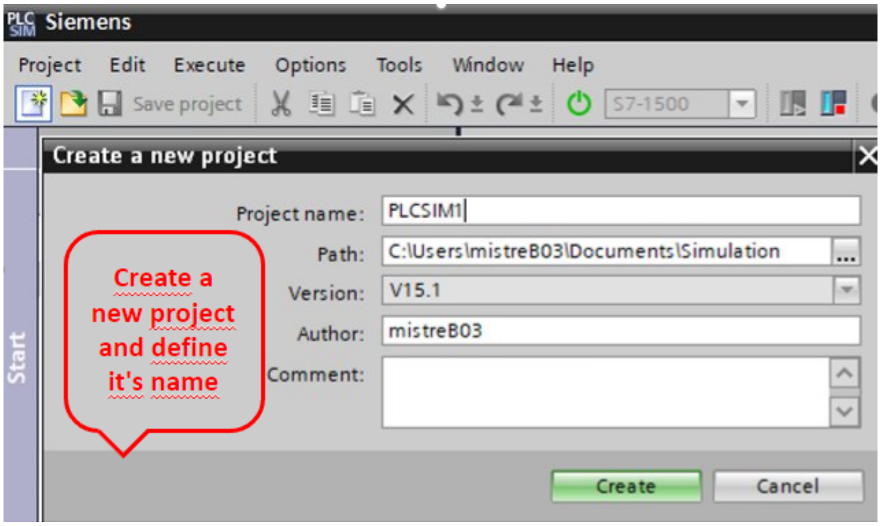


Figure 28: To create a force conditions table

→ Creating a new object in the open window in order to foce valus in the simulation table.



Project name: PLCSIM1

Path: C:\Users\mistreB03\Documents\Simulation

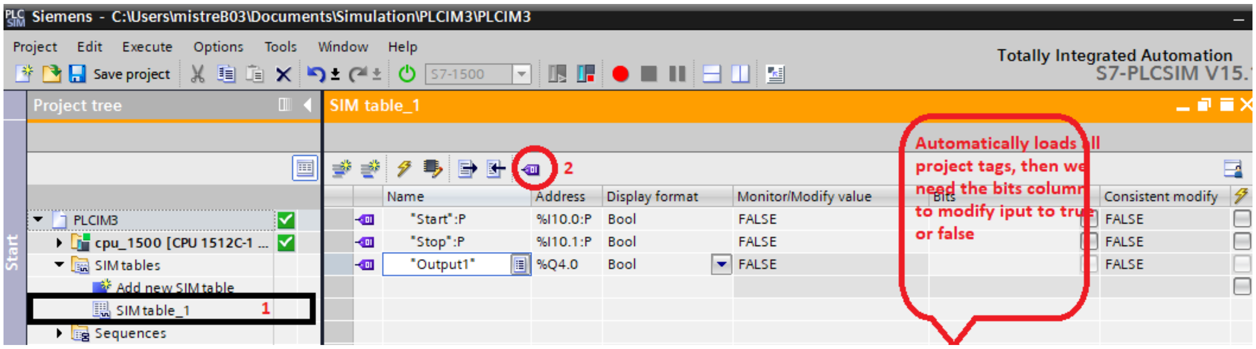
Version: V15.1

Author: mistreB03

Comment:

Create a new project and define it's name

Figure 29: Create new project in the open window

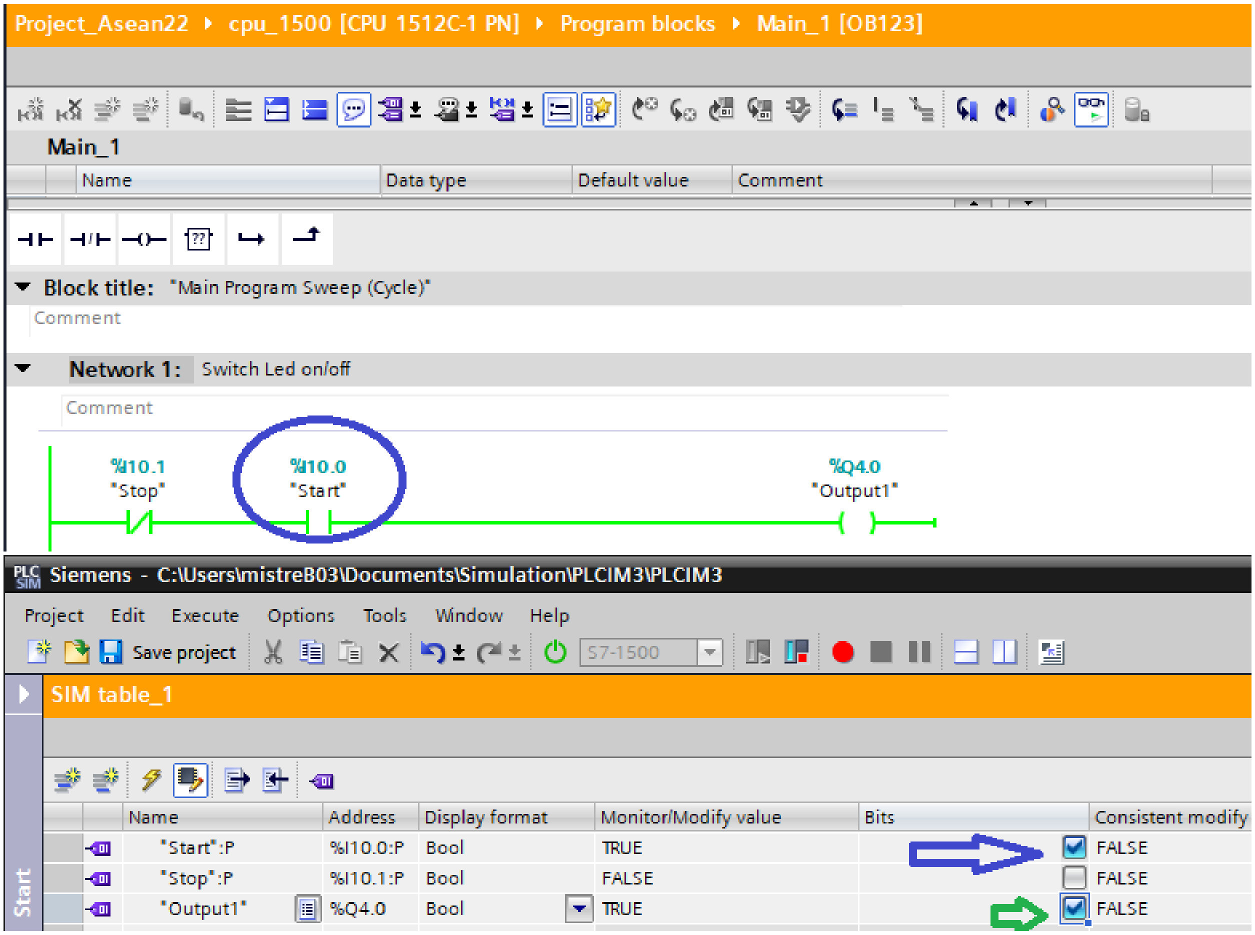


| Name | Address | Display format | Monitor/Modify value | Bits | Consistent modify |
|---|---|---|---|---|---|
| "Start":P | %I10.0:P | Bool | FALSE | ☐ | FALSE |
| "Stop":P | %I10.1:P | Bool | FALSE | ☐ | FALSE |
| "Output1" | %Q4.0 | Bool | FALSE | ☐ | FALSE |

Automatically loads all project tags, then we need the bits column to modify iput to true or false

Figure 30: loading automatically all projet tags

→ Now we back to click on **Monitoring button** to show real time conditions of our switches and output (Condition true green /False blue)
→ Then we force start to true and we observe the activation of output as fpllows;


Project_Asean22 ▸ cpu_1500 [CPU 1512C-1 PN] ▸ Program blocks ▸ Main_1 [OB123]
Main_1
Name
Data type
Default value
Comment
Block title: "Main Program Sweep (Cycle)"
Comment
Network 1: Switch Led on/off
Comment
%I10.1
"Stop"
%I10.0
"Start"
%Q4.0
"Output1"
Siemens - C:\Users\mistreB03\Documents\Simulation\PLCIM3\PLCIM3
Project
Edit
Execute
Options
Tools
Window
Help
Save project
S7-1500
SIM table_1
Start

| Name | Address | Display format | Monitor/Modify value | Bits | Consistent modify |
|---|---|---|---|---|---|
| "Start":P | %I10.0:P | Bool | TRUE | | ☑ FALSE |
| "Stop":P | %I10.1:P | Bool | FALSE | | □ FALSE |
| "Output1" | %Q4.0 | Bool | TRUE | | ☑ FALSE |

Figure 31: Simulation results after forcing values

## 9.2 Second Example: Simple Start/Stop Ladder Logic Relay

→Ladder Diagram (The abbreviation is LAD): Based on the graphical representation of Relay Ladder Logic.
In this ladder logic example, there are two inputs:

**"Start button"** or PLC input **I0.0**.
**"Stop button"** or PLC input **I0.1**.

The **start** button will **activate** the relay, or ladder logic **relay M10.0**. When the start button is released, the relay will **still be activated**, because of **the latch** in ladder rung 2. This **latching will be broken** when the **stop** button is activated.

**Note:**You should use normally closed as stop button, to avoid dangerous situations under failure.

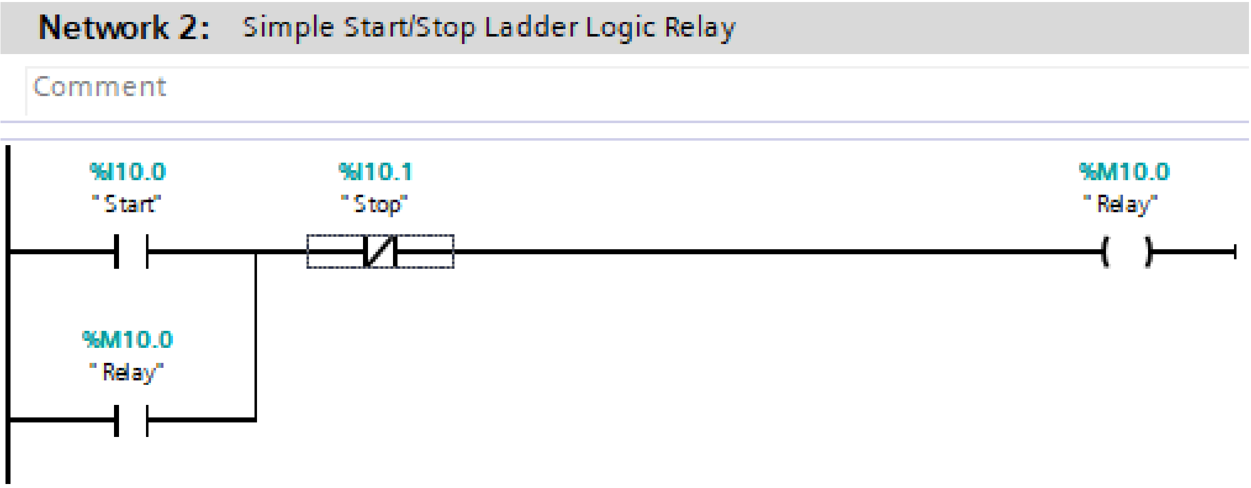
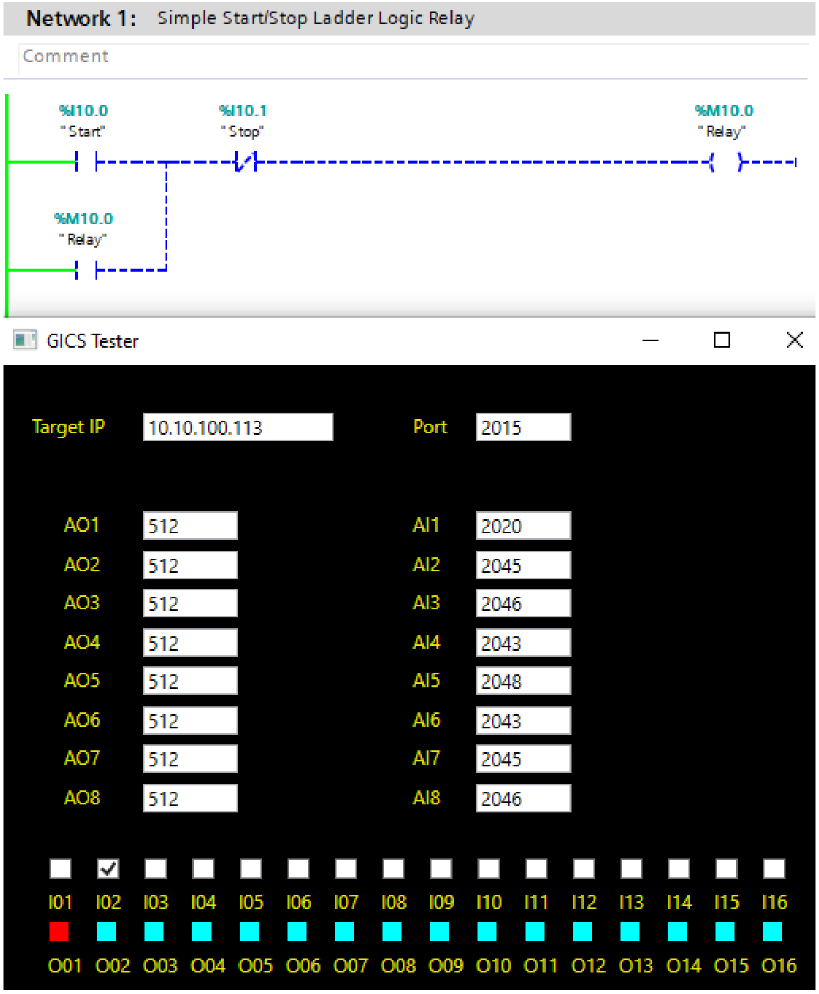


Figure 32: Example Simple Start/Stop Ladder Logic Relay



Target IP 10.10.100.113

Port 2015

| | | | |
|---|---|---|---|
| AO1 | 512 | AI1 | 2020 |
| AO2 | 512 | AI2 | 2045 |
| AO3 | 512 | AI3 | 2046 |
| AO4 | 512 | AI4 | 2043 |
| AO5 | 512 | AI5 | 2048 |
| AO6 | 512 | AI6 | 2043 |
| AO7 | 512 | AI7 | 2045 |
| AO8 | 512 | AI8 | 2046 |

| I01 | I02 | I03 | I04 | I05 | I06 | I07 | I08 | I09 | I10 | I11 | I12 | I13 | I14 | I15 | I16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| □ | ☑ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ | □ |

| O01 | O02 | O03 | O04 | O05 | O06 | O07 | O08 | O09 | O10 | O11 | O12 | O13 | O14 | O15 | O16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 🟥 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 |

Figure 33: Observe result

## 9.3 Examples To do:AND/OR/SET_RESET/TIMER

To practice more here some basic examples with LADDER:
→ Must use instructions to creat your program.
→ Ensure to create all input and Inputs/Output needed or create its directly by typing the address desired on top of your instruction.

**About Flip-Flop**You can monitor the state of a Flip-Flop with various inputs(Reset Dominate Flip-Flop for example : The output goes high with a rising edge on the set input and remains high even when the set input goes low until a rising edge is detected on the reset input.)

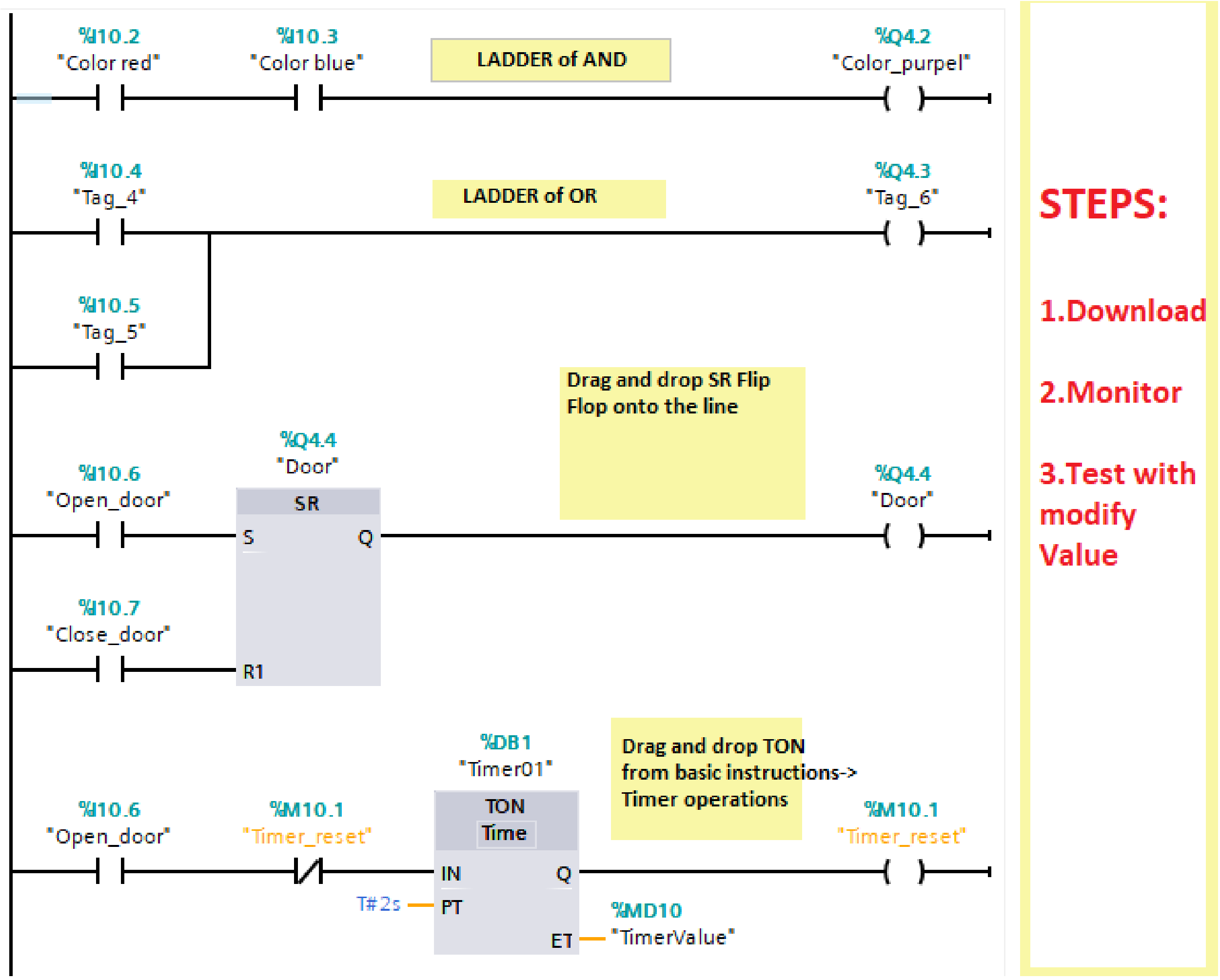
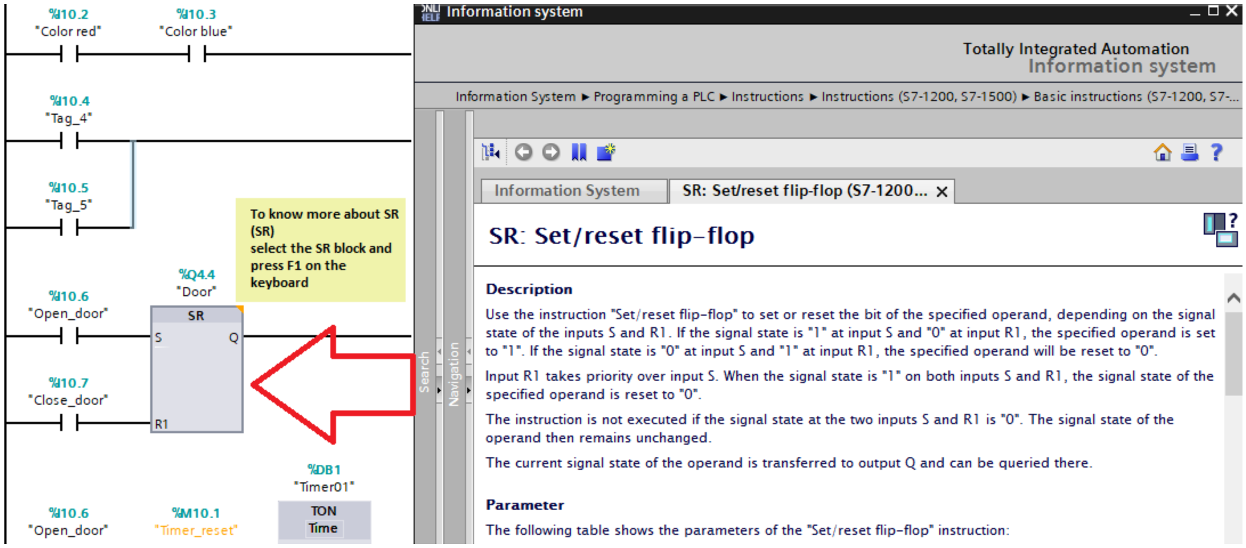


Figure 34: Examples of Ladder programs on the same network



To know more about SR (SR) select the SR block and press F1 on the keyboard

SR: Set/reset flip-flop

**Description**

Use the instruction "Set/reset flip-flop" to set or reset the bit of the specified operand, depending on the signal state of the inputs S and R1. If the signal state is "1" at input S and "0" at input R1, the specified operand is set to "1". If the signal state is "0" at input S and "1" at input R1, the specified operand will be reset to "0".

Input R1 takes priority over input S. When the signal state is "1" on both inputs S and R1, the signal state of the specified operand is reset to "0".

The instruction is not executed if the signal state at the two inputs S and R1 is "0". The signal state of the operand then remains unchanged.

The current signal state of the operand is transferred to output Q and can be queried there.

**Parameter**

The following table shows the parameters of the "Set/reset flip-flop" instruction:

Figure 35: Additional information

# 10 References

For more details :
https://www.plcacademy.com/ladder-logic-examples/#star-delta
https://cache.industry.siemens.com/dl/files/384/86140384 /att_1028674/v1/
s71500_et200mp_manual_collection_en-US.pdf