# PLC Lab 1: Introduction to PLC Systems & Software

## Objectives

* Recognise and understand the functions of components that works with a PLC in a typical automated system (i.e power supply, breakers, terminal blocks, relays, network switches, HMI, PLC modules, etc)
* Learn ladder programming basics using industrial PLC software TIA Portal
* Download and run program on actual PLC Hardware and also Simulator
* Apply PLC ladder programming for simple logic and sequential tasks
* Utilise SFC to plan for program sequence

## Tasks

* Task 1: Recognise and understand the components that works with a PLC
* Task 2: Identify the address mapped to each external device
* Task 3: Install the Software and configure license
* Task 4: Using the PLC Template program
* Task 5: Write a simple ladder diagram, define PLC tags
* Task 6: Knowing the PLC Tag Table
* Task 7: Compile program and resolve errors
* Task 8A: Downloading to real PLC Hardware and test it
* Task 8B: Downloading to a PLC Simulator called PLCSIM
* Task 9: Online Monitoring, Modifying Variables, Force Table & Watch Table
* Task 10: Logical operations in PLC ladder
* Task 11: Creating a latch using internal relay
* Task 12: Programming a conveyor system
* Task 13: Using Sequential Function Chart as a flowchart
* Task 14: Application: Conveyor sequence

## Equipment Required

* PLC training kit with control panel.
* Laptop with internet access, and TIA Portal V15.1 and PLCSIM installed.

## Reference and Self Study materials

* Lab1 playlist (Same as corner QR code) <https://www.youtube.com/playlist?list=PLo5IISMe0m5N7Cd4PMbVd2GiP5bE3j1IE>
* <https://www.electricaltechnology.org/2018/12/what-is-relay-different-types-of-relays-its-operation-applications.html>

## Task 1: Recognise and understand the components that works with a PLC

Below is the typical block diagram of the components in a PLC system

PLC

CPU

PLC

Input  
Module

PLC

Output  
Module

Power Supply

Programming Unit  
(Laptop/PC)

HMI/ User Panel/ Touch Panel

Input Devices

Output Devices

1. *Watch the video* [*https://youtu.be/ml02GB025Ek*](https://youtu.be/ml02GB025Ek). Label the parts as follows

|  |  |  |  |
| --- | --- | --- | --- |
| PLC CPU | HMI/Touch panel | Relays | Terminal Blocks |
| PLC I/O modules | DC power supply | Remote I/O module | Network Switch |

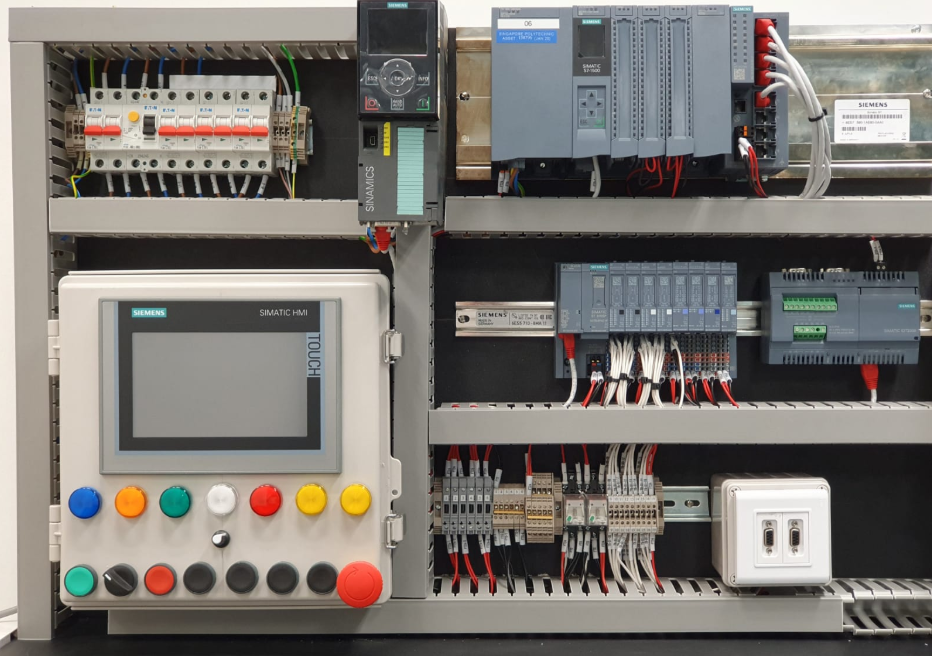
Motor Driver (VFD)

a

b

Circuit Breakers

c



f

Port for conveyor kit

User Panel

h

e

d

IOT module

g

## Task 2: Identify the address mapped to each external device

1. Input and outputs devices (Buttons, Relays) are hardwired to specific PLC I/O points. They have addresses (i.e. %IX.X and %QX.X). On the other hand, each device (CPU, HMI, Drives) also have IP addresses and Profinet Device Names. These identification are very important.

Address for User Panel’s Buttons and Indicators



Write down the address for the Selector Switch

ESTOP

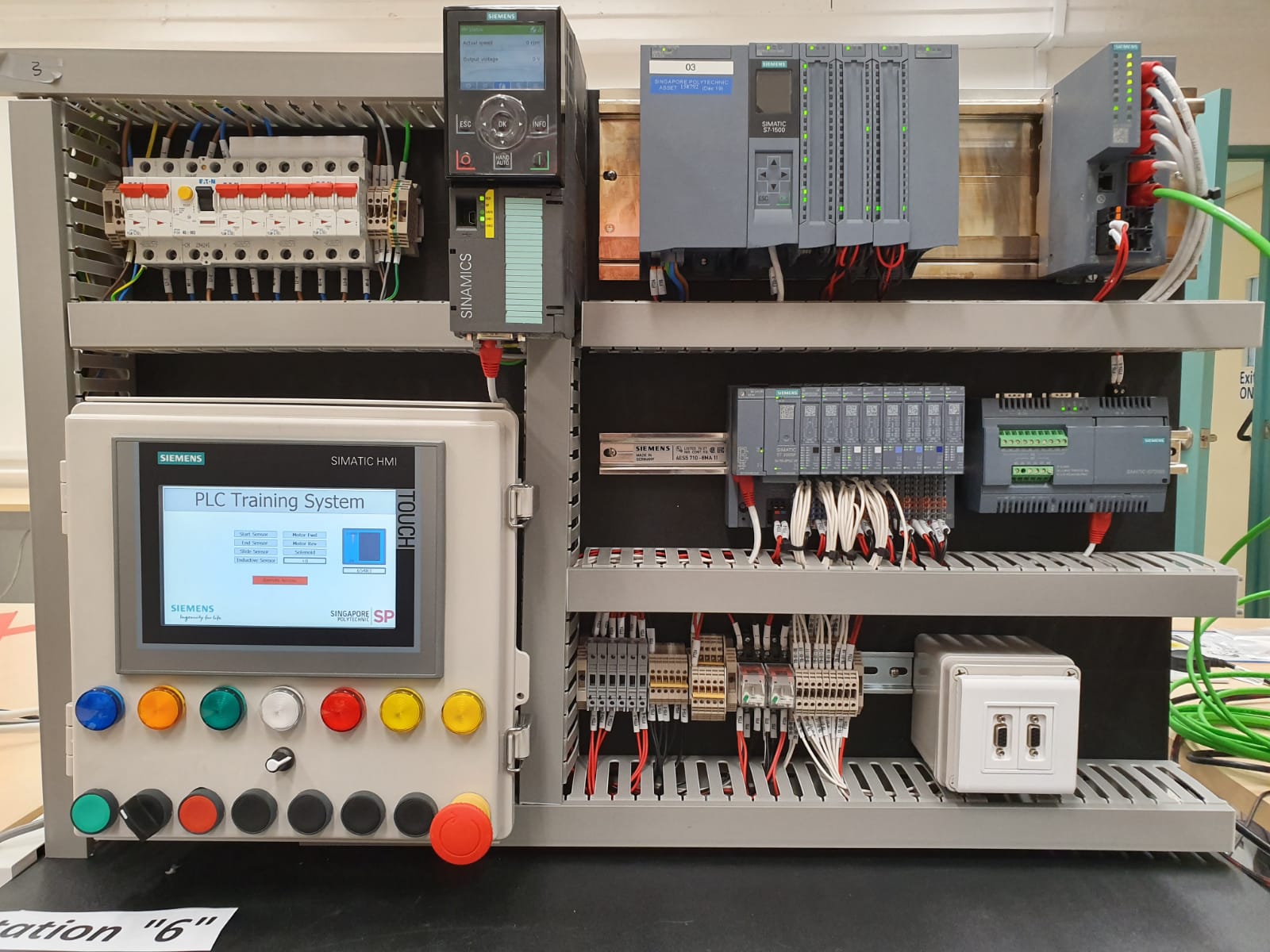
%Q128.6 .5 .4 .3 .2 .1 .0  
 LED1 LED2 LED3 LED4 LED5 LED6 LED7

Red LED (LED5)

PB1 Selector PB2 PB3 PB4 PB5 PB6 ESTOP  
%I126.7 .6 .5 .4 .3 .2 .1 .0

Potentiometer %IW128

IP address and Profinet Device Names



Drive\_2  
192.168.0.7

Switch\_1  
192.168.0.2

PLC\_1  
192.168.0.1

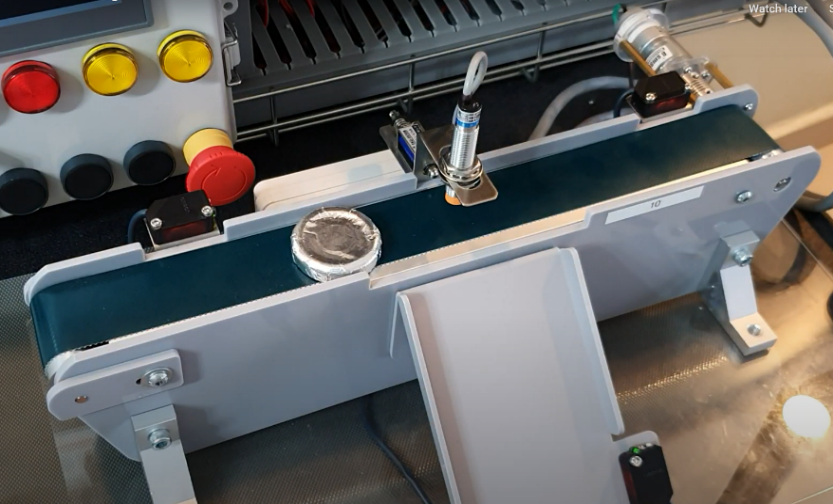
IO device\_1 **192.168.0.4**

HMI\_1  
192.168.0.5

Address for Conveyor System

Sensor\_Inductive  
%I127.3

**Motor\_FWD %Q129.0  
Motor\_REV %Q129.1**



**Solenoid  
%Q129.2**

Sensor\_End  
%I127.2

Sensor\_Start  
%I127.1

Sensor\_Slide  
%I127.0

## Task 3: Install the Software and configure license

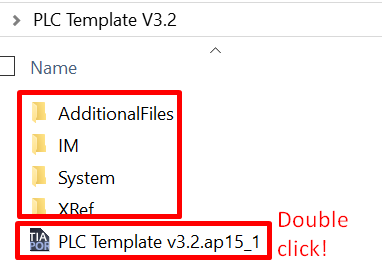
1. The PLC software to install on your personal laptop is

* Siemens TIA Portal V15.1
* Siemens PLCSIM V15

Follow the instructions given on Blackboard/Polymall to install. Also follow the instructions to configure the license, if not, it will expire in 21 days. Only students and staff can use the license.

## Task 4: Using the PLC Template program

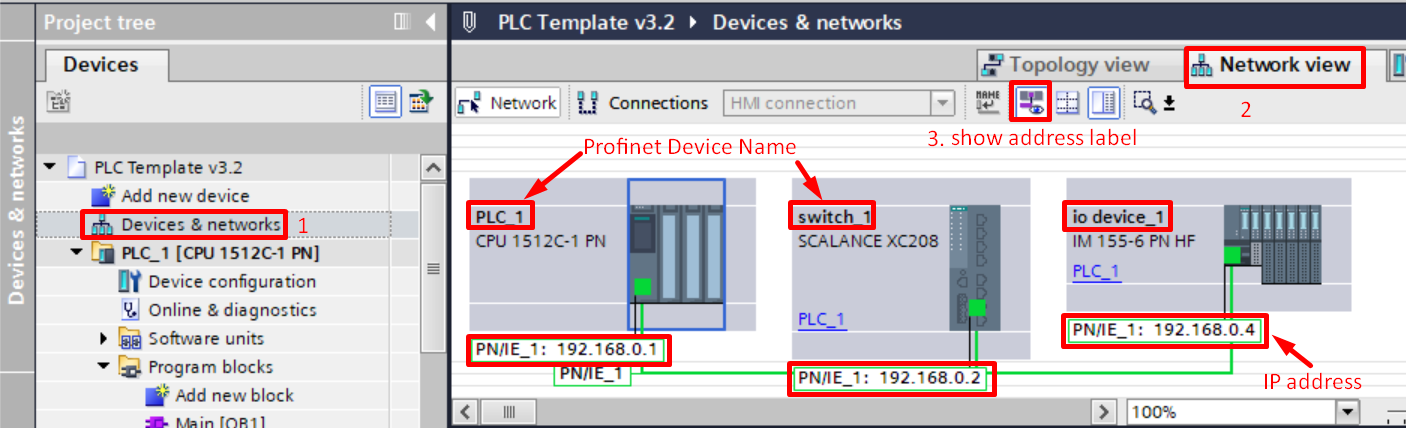
The template provides a quick way to get started with hardware already pre-added and saved. In subsequent lab, you will also learn to create a this from scratch.



1. In D drive, unzip the PLC Template.ZIP file (Right click 🡪 extract all). Open the folder and run the PLC Template.ap15\_1. Take note the project consist of every subfolder in the list, not just that file. At bottom left of the screen, click on Project view

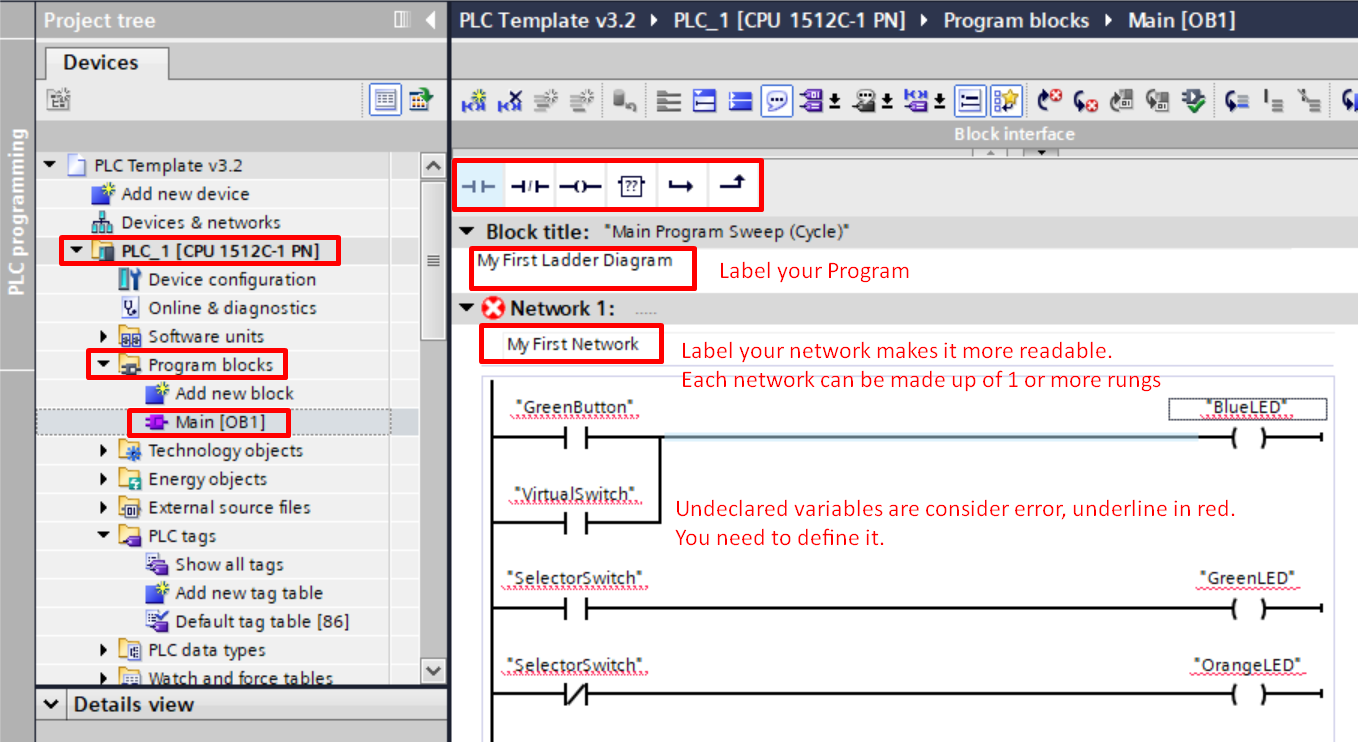


1. Check IP address and Profinet Device Names are correct. Ensure your laptop IP address is in the same network (i.e. 192.168.0.X) in order for them to communicate. You can see that the PLC, Network Switch and Remote IO are already added and linked in a network called PN/IE 1 (Profinet/Industrial Ethernet)



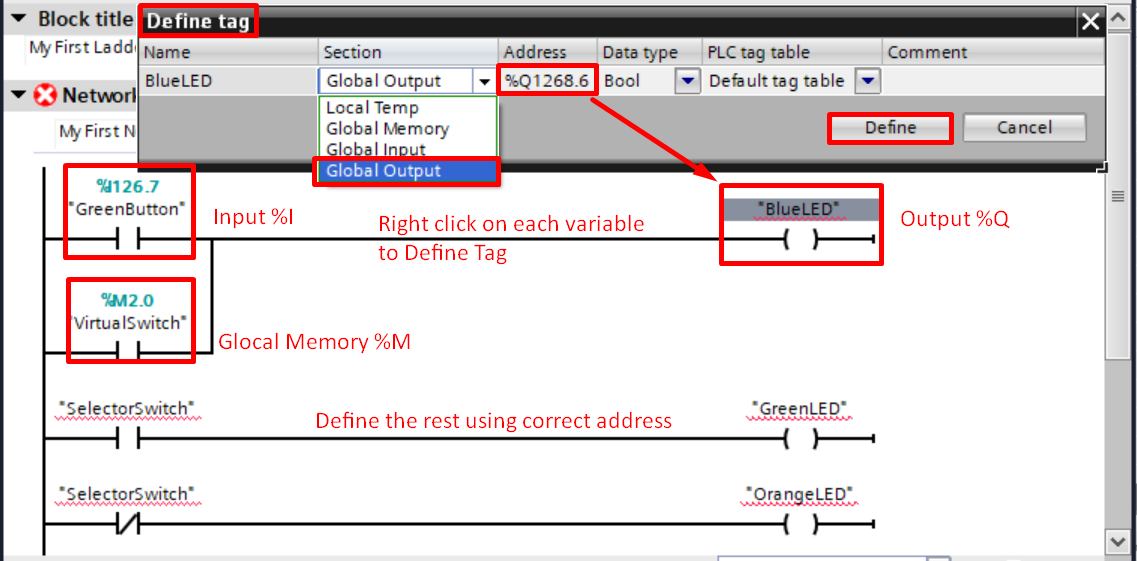
## Task 5: Write a simple ladder diagram, define PLC tags.

1. Follow Go to PLC> Program Blocks> OB1 to start programming. If you write all your programs in OB1, it is called Linear Programming. You can also create blocks and call them from OB1.
2. Use contacts/coils/branches to program ladder. This reference video is not exactly the same, but it will guide you if you are lost. <https://youtu.be/RGQh1g65iYU>

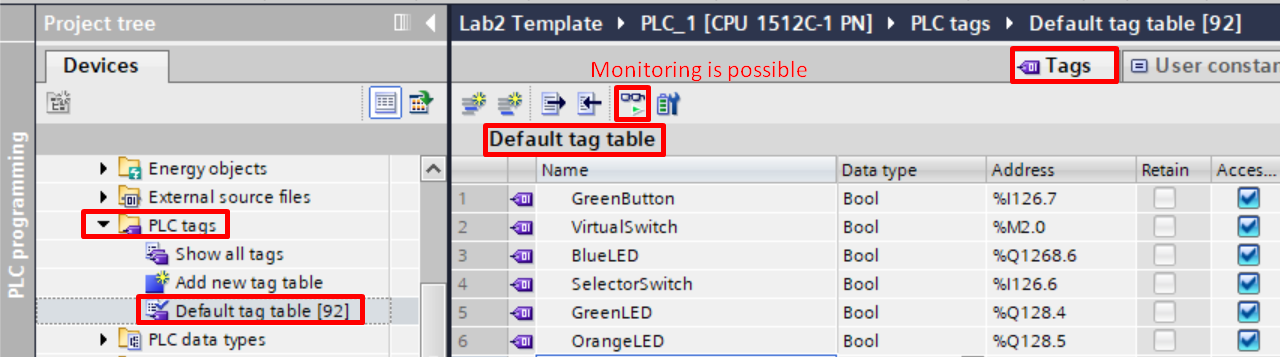


**Notice both are same variable, but one is N.O, the other N.C**

1. Right click on each variable to “Define Tag”, choose address type %M, %I, %Q. Refer to the address map introduced earlier.



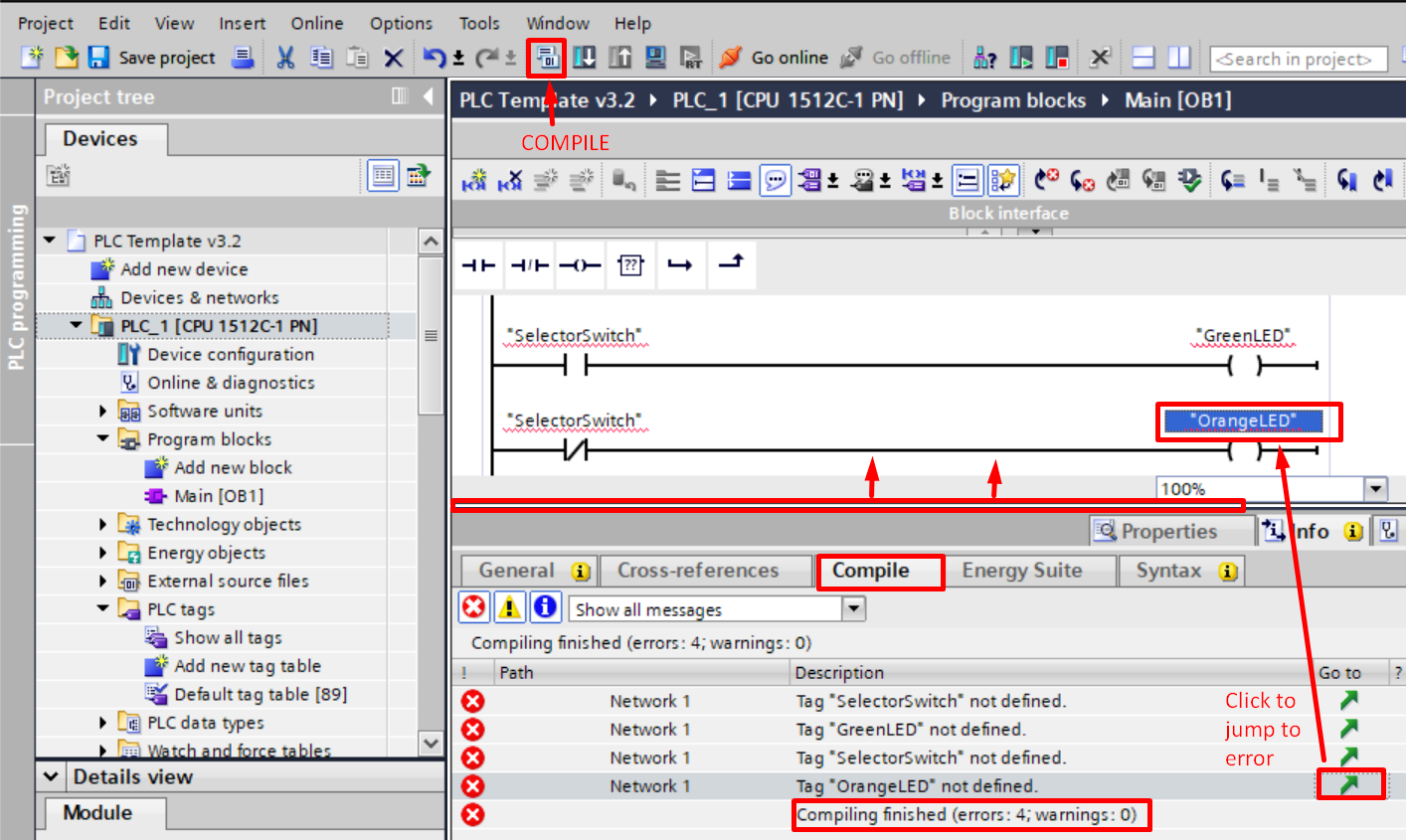
## Task 6: Knowing the PLC Tag Table

1. PLC Tag Table – is very important. Defined variables called "tags" can be viewed and modified from the PLC Tags> Default Tag Table. Double click on the tagname or address to change them as desired. 
2. No two variables can share the same address. Try for yourself to create this error. What do you observe? .

|  |
| --- |
| *Observation:* |

## Task 7: Compile program and resolve errors

1. Compile the program. Errors will show in the compile window which may be hidden. Drag the window up to reveal it. Click on the “Go to” arrow will jump to the error. All errors must be resolved before downloading is permitted.

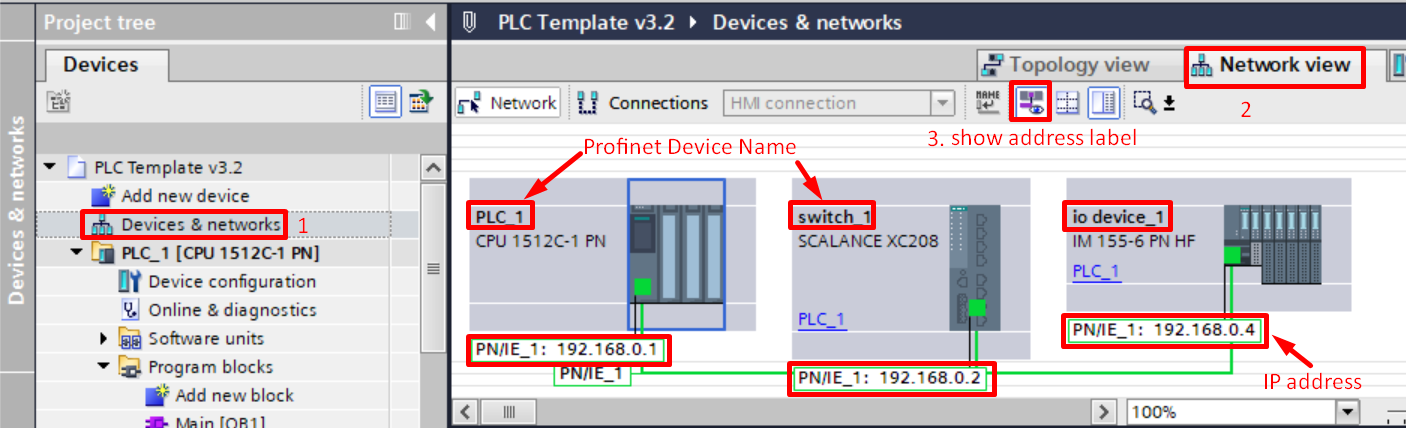


1. Try it. Introduce the following errors and compile it
   1. Misspell a variable name,
   2. Delete an output coil
   3. Start a new branch but leave it unconnected

Check the error message window to see how the system can help you troubleshoot.   
 Resolve the errors and get it to work again.

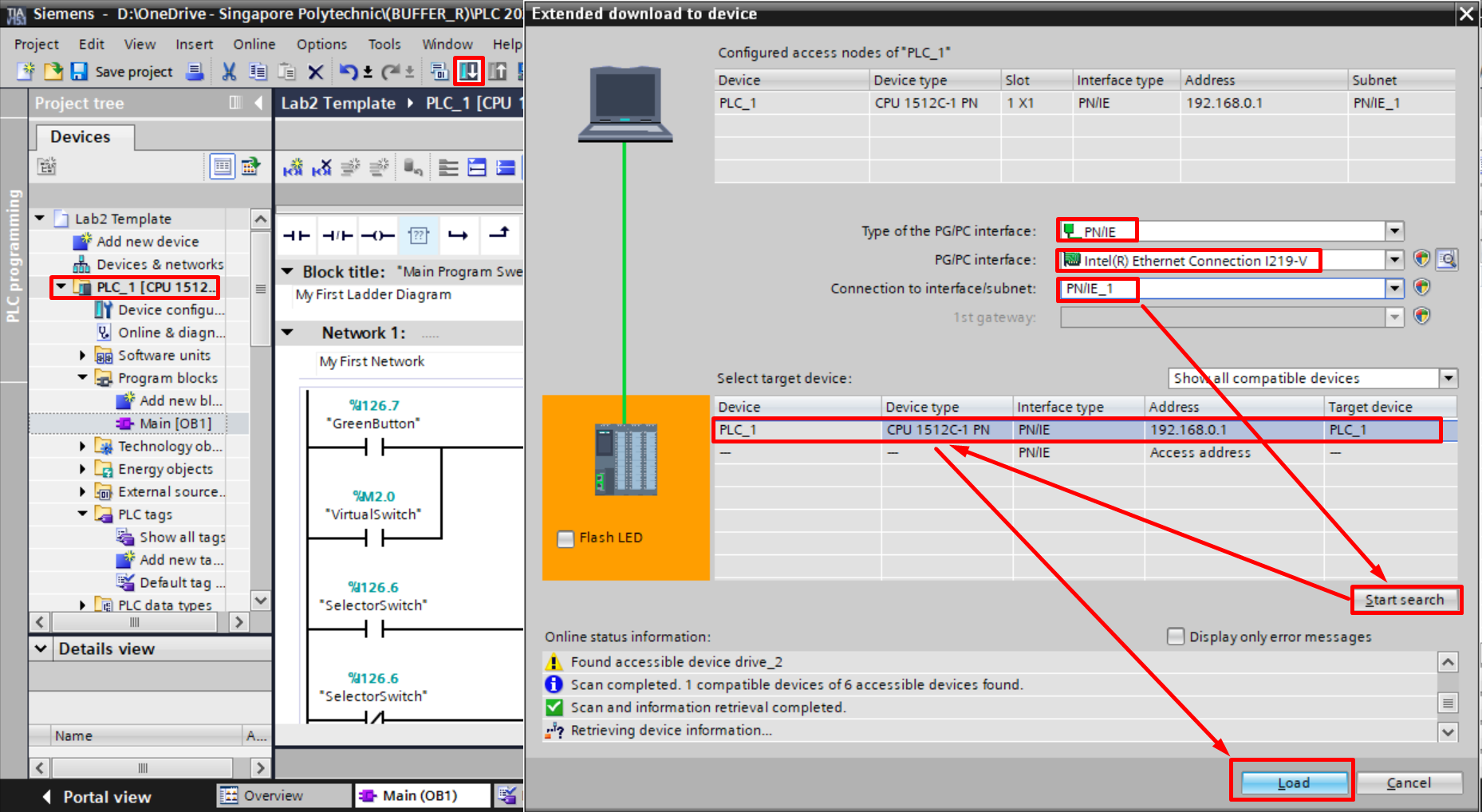
## Task 8A: Downloading to real PLC Hardware and test it

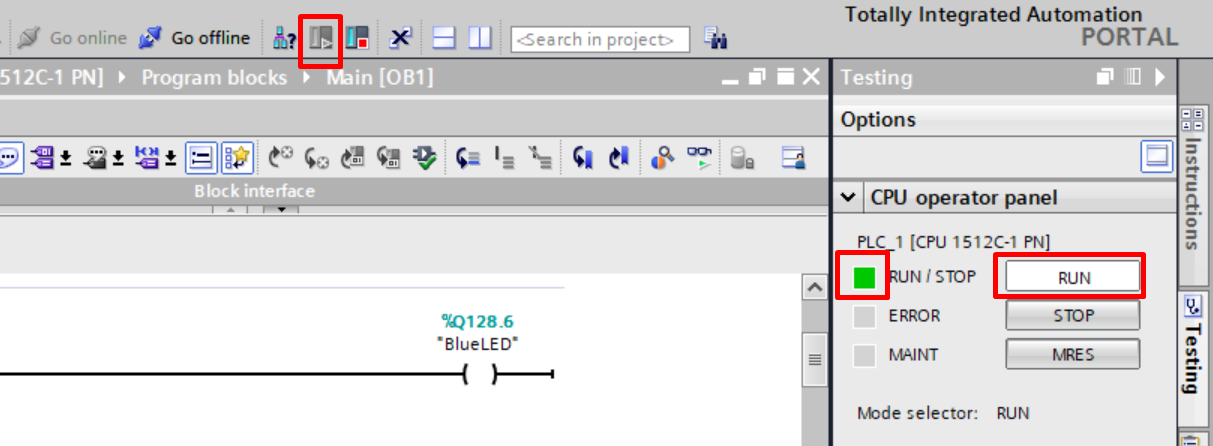
1. If you do not have access to the PLC Hardware at this point in time, you can still continue here. Do know that in the next section, you can still download using a simulator, PLCSIM.
2. Check the Devices are present and the **IP address** and **Profinet Device Names** are correct. Ensure that your laptop IP address is in the **same network** as the PLC (i.e. 192.168.0.X) in order for them to communicate. You can see that the 3 devices are linked up in a network call \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



1. First click on PLC 🡪 click Download icon🡪 set as below 🡪 click Start Search 🡪 select the device 🡪 click Load.

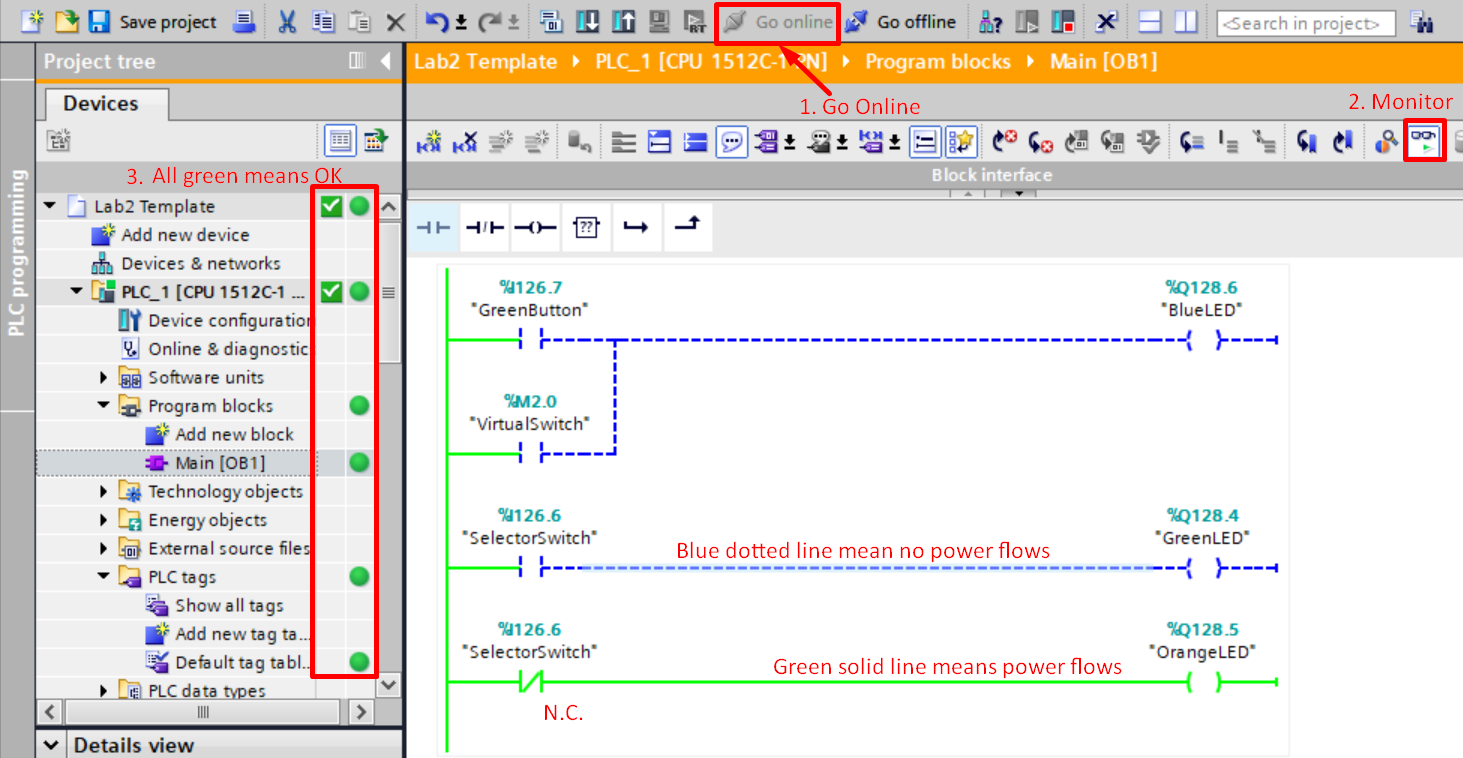
When prompted, click “Continue without synchronisation 🡪 Load



1. PLC status of LED (Run/Green) (Stop/Orange)  
   **PLC Must be in RUN mode in order for the program to execute**. On the **left panel**, click on **PLC\_1(CPU\_1512C),** then click on **“Online”** and **“Run”** icon.   
   You can also run it from right column > Testing > CPU operator Panel,   
   Or directly on the physical PLC itself.
2. Test the GreenButton and SelectorSwitch on the user panel to check the program logic.

## Task 9: Online Monitoring, Modifying Variables, Force Table & Watch Table

Applies to both real hardware and also simulation. One thing we like about the PLC is how easy it is to monitor the flow of power on screen. We also get to see the status of variables.

1. Click **“Go Online**”. Status should be all green. If it’s not, there may be errors. Click on the **goggles to start monitoring**. Flow of power is visible. Solid Green mean power flows in the line. Dotted Blue line means no power.
2. **Modifying Variables.** Right click VirtualSwitch🡪 Modify🡪**Modify to 1**. Global Memory %M status can be "modify" but Input %I and Output %Q cannot be modified in this way. 

4. Flow of power is visible only when Online + Monitoring

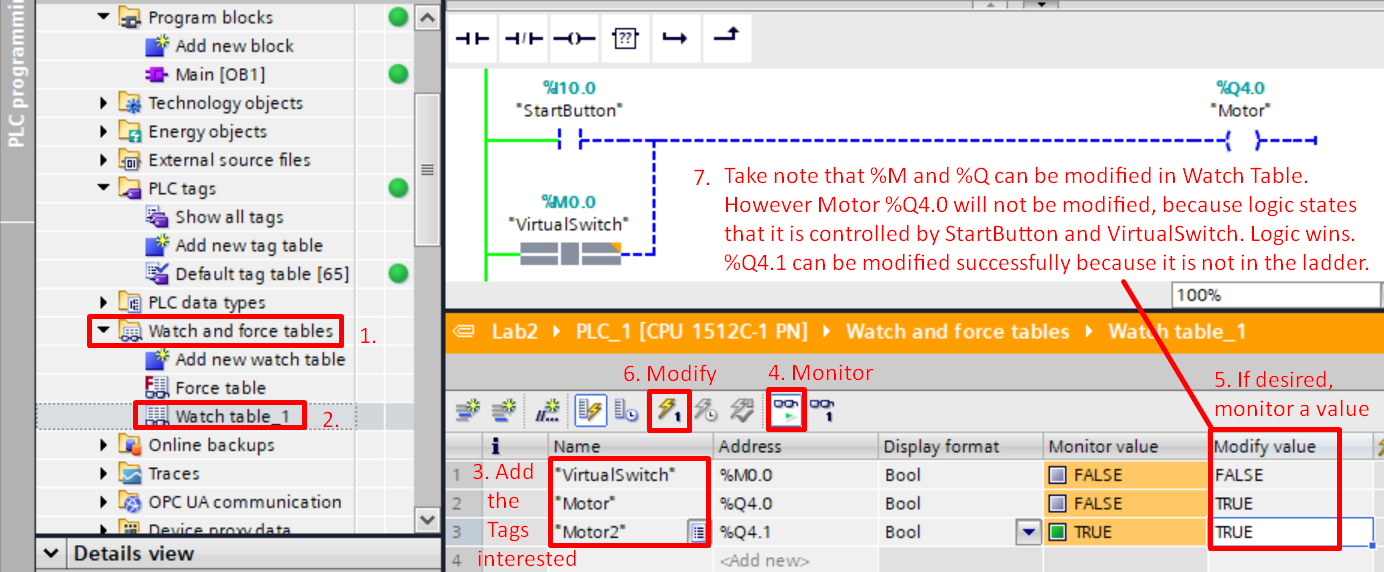
5. Right Click on VirtualSwitch 🡪 Modify 🡪 Modify to 1

1. **Force Tables** – can be used insist **I/O** to take on a certain desired value. (It **overrides** the ladder logic controls, **beware**). Take note, PLC error light turns on when a variable is forced.



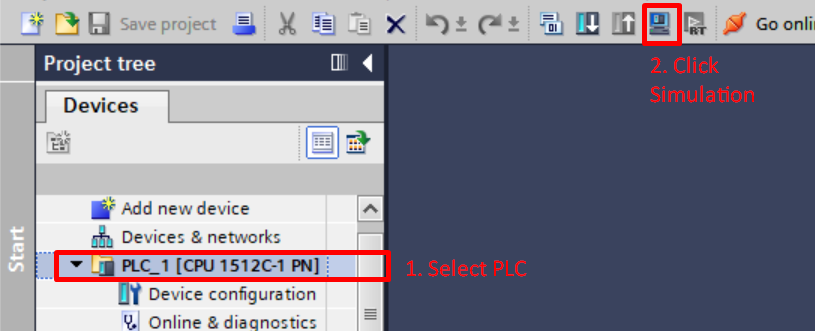
GreenButton

1. **Watch Table** – can be used to **monitor** interested variables. Possible to **modify** %**M,** %**Q,** but cannot override the ladder logic. Ladder logic wins.



## Task 8B: Downloading to a PLC Simulator called PLCSIM

1. Working on PLCSIM gives some flexibility, since no PLC hardware is needed. Click on PLC, then **Click Simulation icon** to run PLCSIM. If prompted, click ok to allow “Enable Support Simulation”.

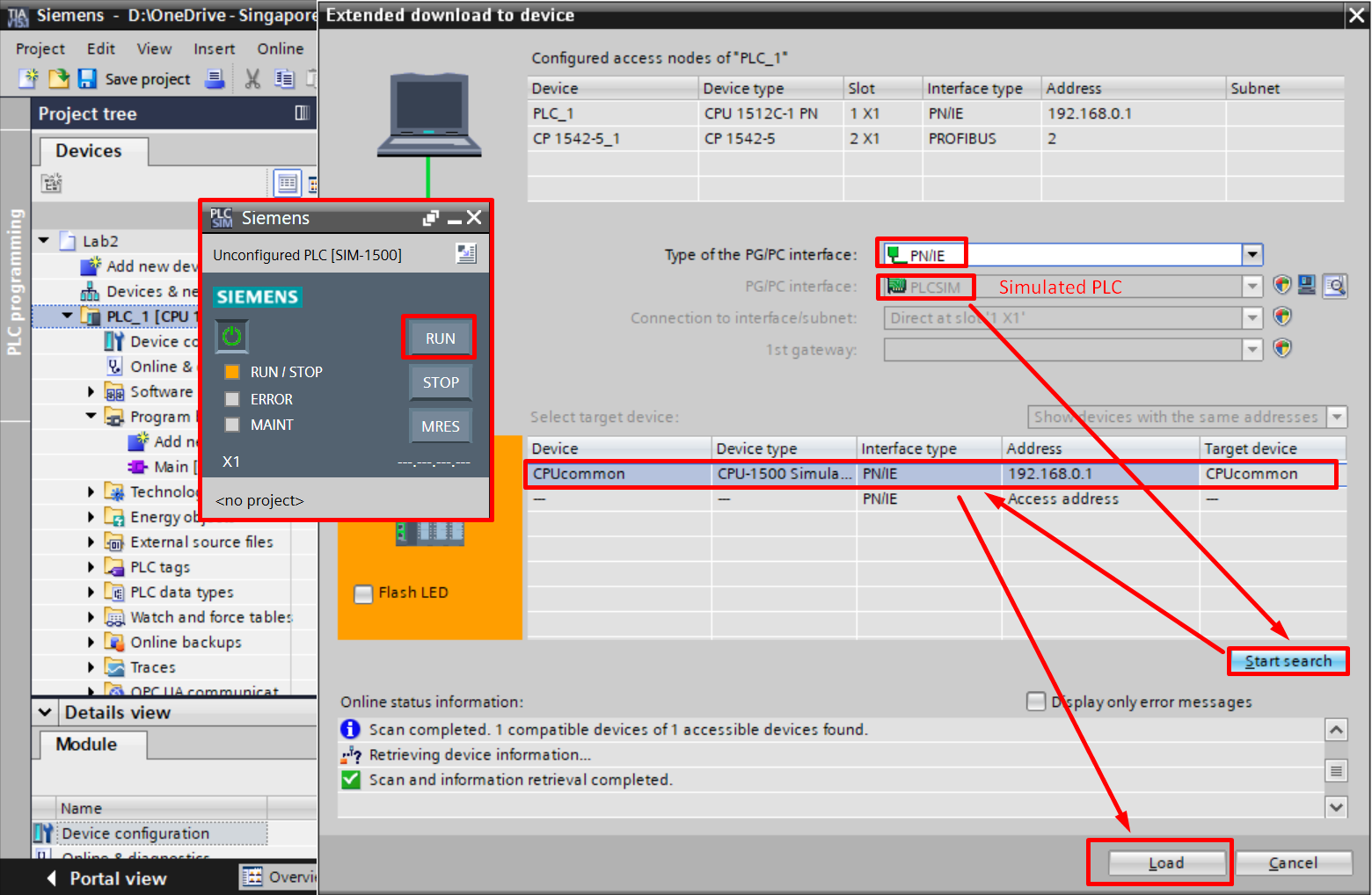


1. Select PLC

2. Click

Simulation

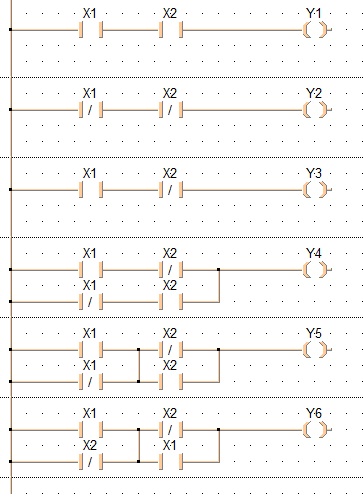
1. The PLCSIM window pops up, wait till the orange RUN/STOP LED appears steady. This means the Simulated PLC has finish booting and is in STOP mode. Select interface as “PN/IE”, click on “Start Search”, Select the PLC that appears, click Load.



1. PLC status of LED (Run/Green) (Stop/Orange)  
   **PLC Must be in RUN mode in order for the program to execute. Ensure. Click RUN**
2. You are now running in PLCSIM. All interfaces works accordingly and you should not notice any difference, other than the fact that your hardware is not responding because they are not connected.
3. If you have not done it**, Backtrack to Task 9:** Online Monitoring, Modifying Variables, Force Table & Watch Table; which is applicable for both real hardware and simulation.

## Task 10: Logical operations in PLC ladder

1. We can make use of ladder to create logic operation such as AND, OR, XOR, etc.   
   Program the ladder logic below and complete the truth table.



**Task**: complete the truth table from your completed program

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| X1 | X2 | Y1 | Y2 | Y3 | Y4 | Y5 | Y6 |
| 0 | 0 |  |  |  |  |  |  |
| 0 | 1 |  |  |  |  |  |  |
| 1 | 0 |  |  |  |  |  |  |
| 1 | 1 |  |  |  |  |  |  |

## Task 11: Creating a latch using internal relay

Green Lamp

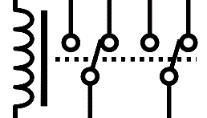
Start

Stop

24VDC

24VDC

Relay



1. This is typically how an electro-mechanical 2PDT relay is wired up to create a latch that will hold on to itself after the N.O pushbutton has been depress momentarily. A PLC can easily do this without a real relay. We call it internal relay. (using global memory %M, we can create unlimited relay coils and contacts) If you need a revision on relays, go to <https://www.electricaltechnology.org/2018/12/what-is-relay-different-types-of-relays-its-operation-applications.html>
2. Converting the circuit to ladder diagram and **draw it below**. Enter the ladder into TIA portal and test it. IF you are lost <https://images.app.goo.gl/8tGbxwuvg4FY8eHE7>
3. (After you are successful. Think how you can add a Red Lamp such that it is always opposite in logic to the Green Lamp.)

Relay   
Coil

Stop

Start

N.O. Contact

Green  
Lamp

N.O. Contact

1. Fill in the PLC tag table below

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Address | Data Type | |
| Start |  |  | |
| Stop |  |  | |
| Relay |  |  |
| Green Lamp |  |  | |
| Red Lamp |  |  | |

1. Test the program. Describe how the latch function

|  |
| --- |
|  |
|  |
|  |

## Task 12: Programming a conveyor system

Sensor\_Inductive  
%I127.3

**Motor\_FWD %Q129.0  
Motor\_REV %Q129.1**



**Solenoid  
%Q129.2**

Sensor\_Start  
%I127.1

Sensor\_End  
%I127.2

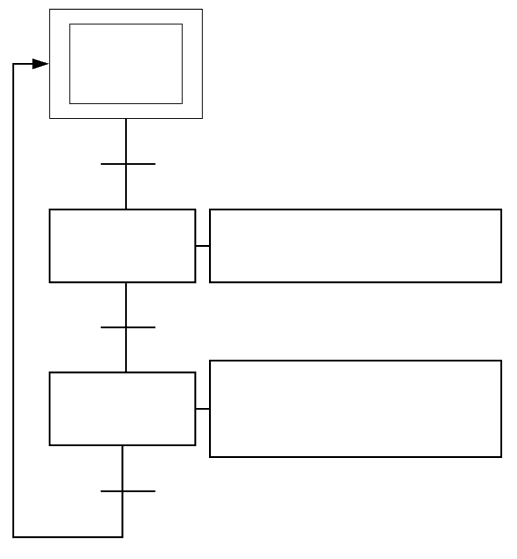
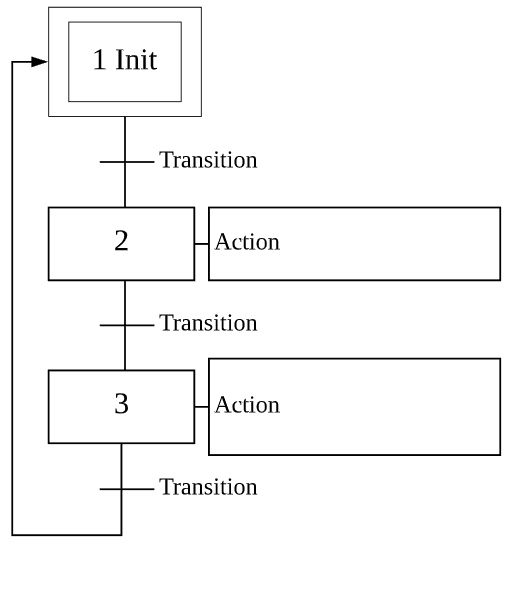
Sensor\_Slide  
%I127.0



1. Connect up the conveyor system. <https://youtu.be/g6gCH38n-nA>  
   See the automation sequence here. <https://youtu.be/Ic5PcAIeUF8>  
   Write a program such that when a part is place in front of “Sensor\_Start”, the part will travel forward on the conveyor. When the part reaches “Sensor\_End” the conveyor will stop.   
   Test and show it to your instructor.
2. When you are successful. Add a START Button so that the operation only begins a part is present and also the Start Button is pressed. Add a STOP button so that the operation can be terminated anytime.

## Task 13: Using Sequential Function Chart as a flowchart

SFC is powerful tool and in fact also an IEC 61131-3 programming language. But, for now, we will use it like a flow chart to help us understand our program. Based on Task12, fill in the SFC with help and explanation from your instructor.



When STOP\_BUTTON is pressed, all outputs will turn off. Sequence will return to initial state.

## Task 14: Application 2

Re-Program the conveyor system such that the work piece moves back to the starting point after reaching “Sensor\_End”. Demonstrate to the instructor

*The End*