1501324 Sequence Control and PLC

Course Description:

Basic sequence control; Programmable Logic Controllers (PLCs); Programming of PLC systems; Ladder program development; Sequential control facilities; Advanced features of PLCs; PLC Communications and automation*; Plant emulation; Process monitoring and control*.

Learning outcome:

1. Students are able to implement PLC systems.
2. Students are able to simulate PLC systems.
3. Students are able to work on PLC applications.

Lecturer:

Assoc. Prof. Punnarumol Temdee, Ph.D.
Asst. Prof. Roungsan Chaisricharoen, Ph.D.
Asst. Prof. Santichai Wicha, Ph.D.
Lect. Chayapol Kamyod, Ph.D.

Credit: 3(2-2)
Lecture: 30 Hours (6 hours of modified content)
Lab: 30 Hours (6 hours of modified content)

Assessments:

Attendance	10%
HW/CW	20%
Midterm	25%
Final	25%
Project	20%

Lecture (seminar):

Content	Hours
PLC HW and SW	4
Paradigm of PLC logics	4
PLC programming	4
PLC communications	4
Ladder programs	4
Advanced PLC	4
Simulation and emulation*	2
Process monitoring*	2
Process control*	2

(*modified in the framework of an Erasmus + project: Asean Factori 4.0 Across South East Asian Nations: From Automation and Control Training to the Overall Roll-out of Industry 4.0 609854-EPP-1-2019-1-FR-EPPKA2-CBHE-JP)

Lab (internship):

Content	Hours
PLC configuration	4
PLC wiring	4
PLC digital I/O	4
PLC analog I/O	4
PLC simulation	4
PLC communications	4
Process monitoring via HMI*	2
Process control via HMI*	2
Emulation card*	2

(*modified in the framework of an Erasmus + project: Asean Factori 4.0 Across South East Asian Nations: From Automation and Control Training to the Overall Roll-out of Industry 4.0 609854-EPP-1-2019-1-FR-EPPKA2-CBHE-JP)

1501324 Sequence Control and PLC

Program: Bachelor program in Computer Engineering

Credit: 3(2-2) Lecture: 30 Hours Lab: 30 Hours

2nd Semester, Academic Year: 2023

Assoc. Prof. Punnarumol Temdee, Ph.D.

Asst. Prof. Roungsan Chaisricharoen, Ph.D.

Asst. Prof. Santichai Wicha, Ph.D.

Lect. Chayapol Kamyod, Ph.D.

Co-funded by the Erasmus+ Programme of the European Union

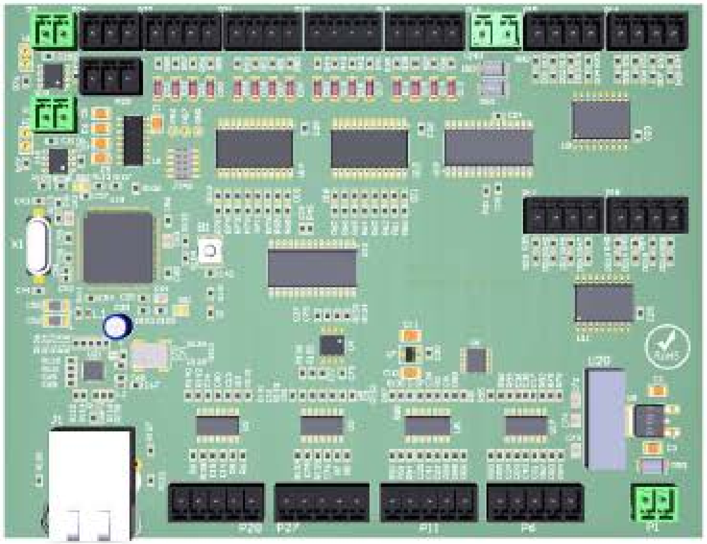
This course has been modified in the framework of an Erasmus + project: Asean Factori 4.0 Across South East Asian Nations: From Automation and Control Training to the Overall Roll-out of Industry 4.0

609854-EPP-1-2019-1-FR-EPPKA2-CBHE-JP

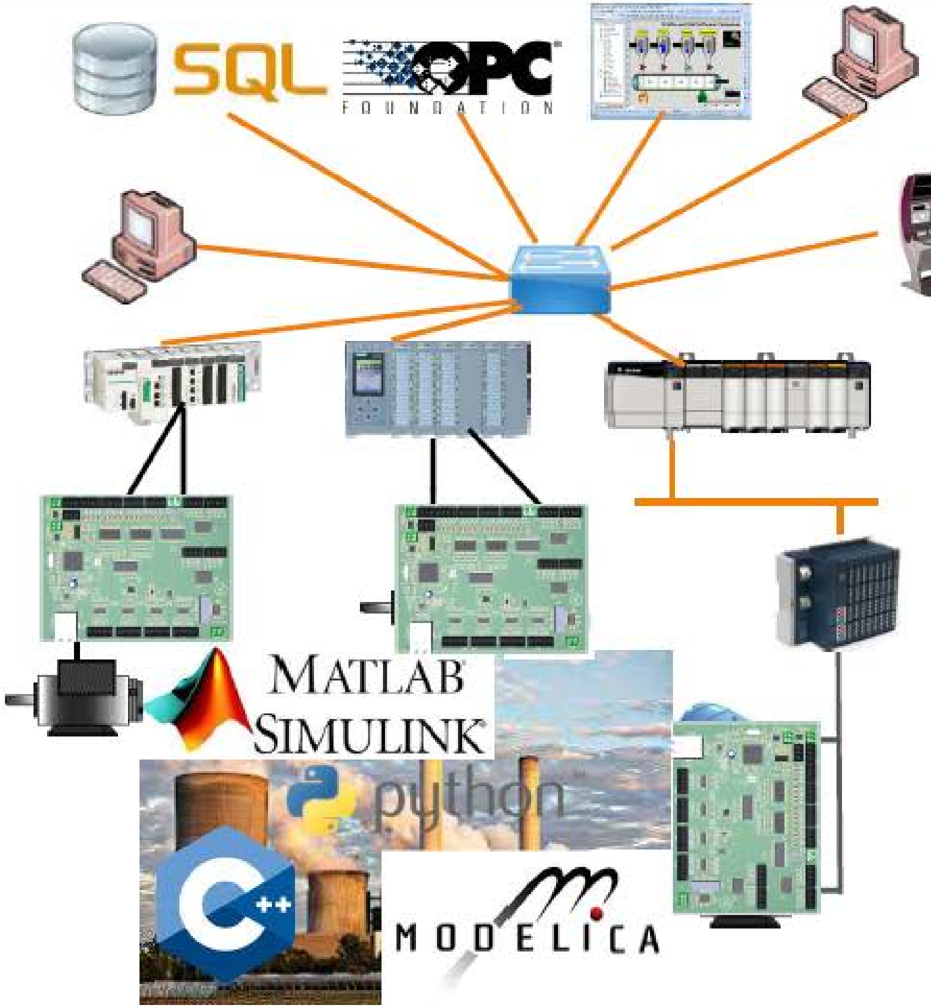
Lecture 01: Simulation and emulation

HARDWARE IN THE LOOP SYSTEM

- **Home made electronic interface card**

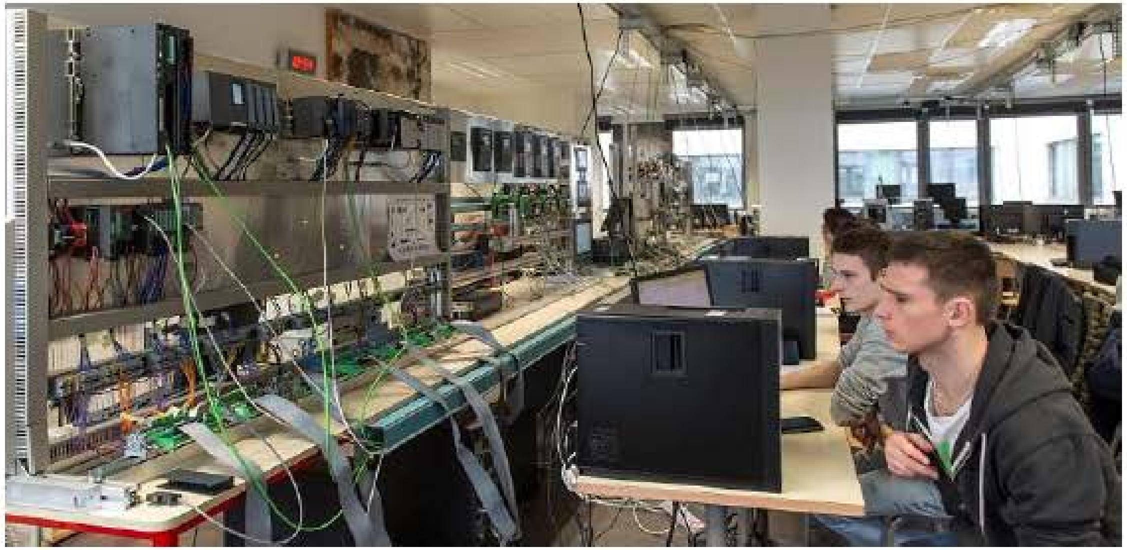


- **24 sensors and 24 actuators**
  - 16 digital inputs / 16 digital outputs
  - 8 analog inputs / 8 analog outputs
- **Less than 500€**
- **Reasonable timing performance (< 10 ms response time)**
- **Easily chain (Ethernet addressing)**



TWO MAIN APPLICATION FIELDS

Industrial automation: PLC, SCADA et OPC

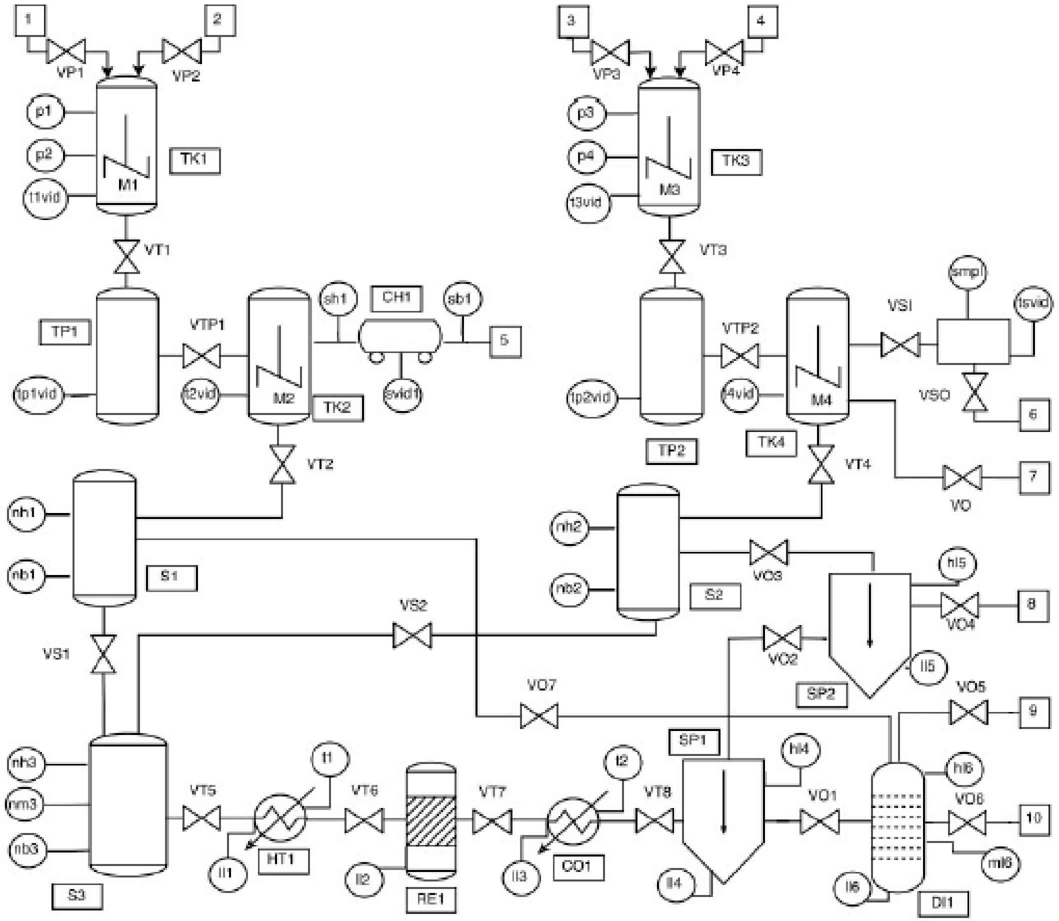


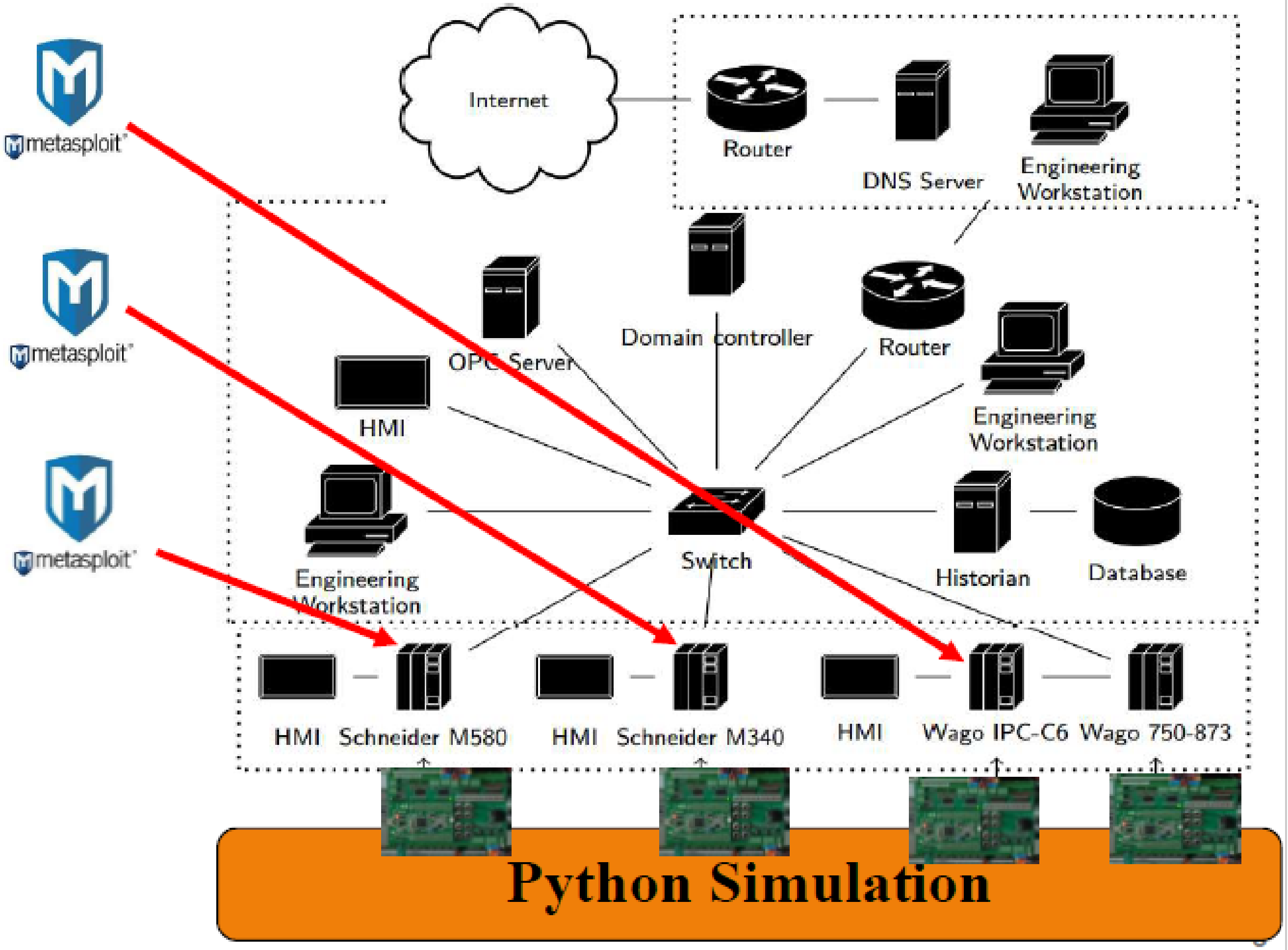
Smart-grid communication

- **IEC 61850**

A SIMPLE APPLICATION EXAMPLE (~70 I/O)

- Tennessee Eastman Chemical Company (O. Koucham PhD benchmark)

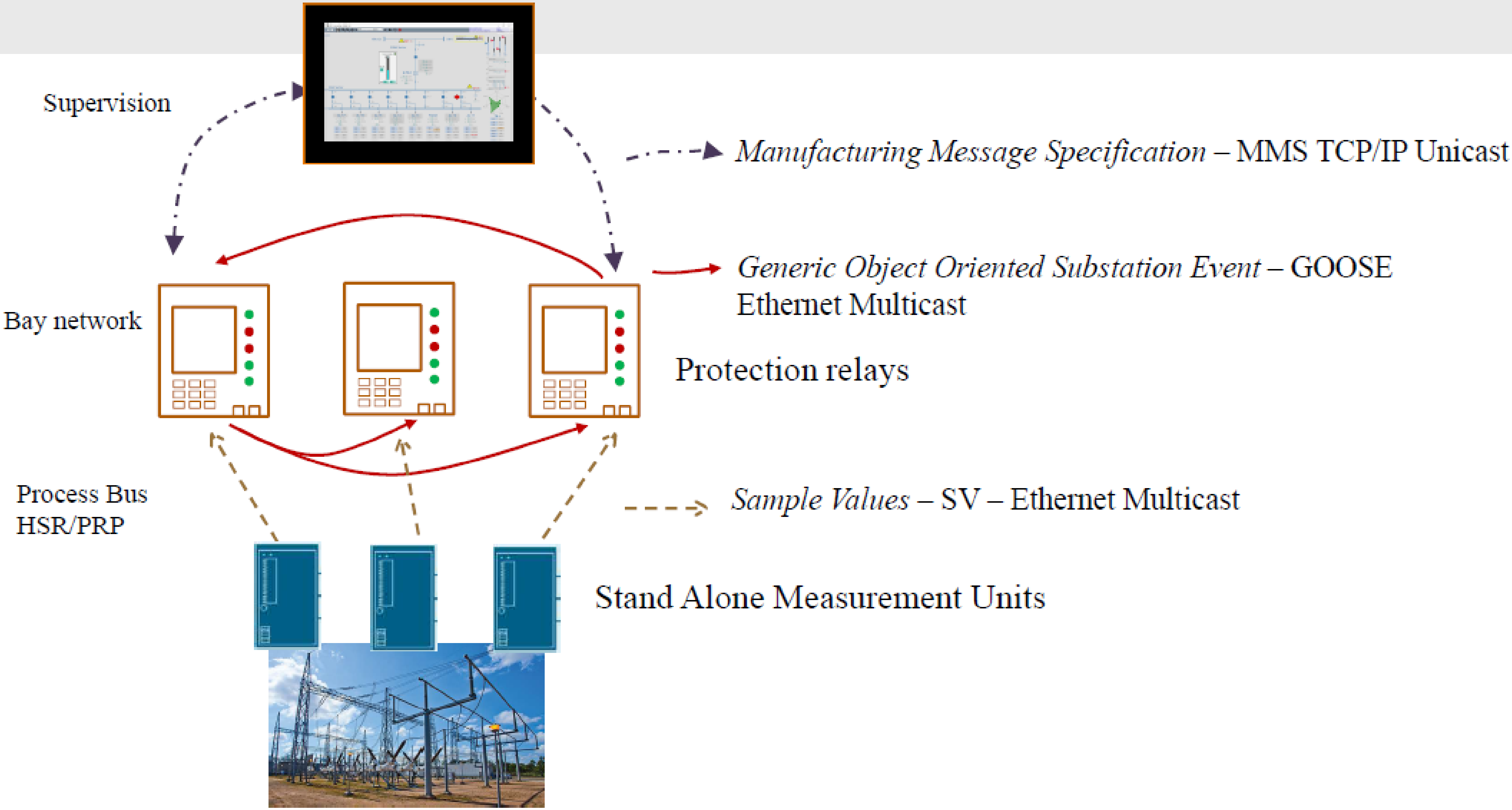




NANOELEC.

Cybersecurity Institute
Univ. Grenoble Alpes

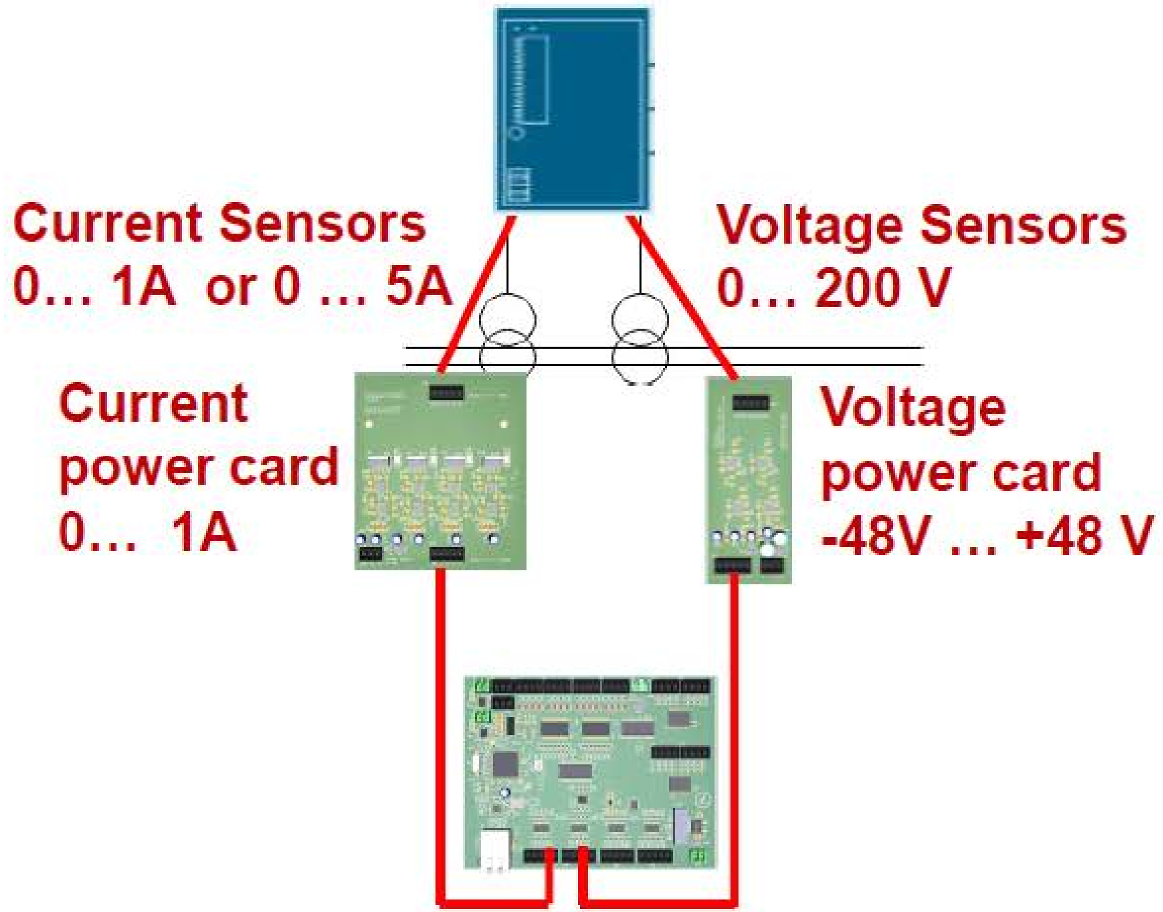
61850 SMART GRID APPLICATIONS



NANOELEC.

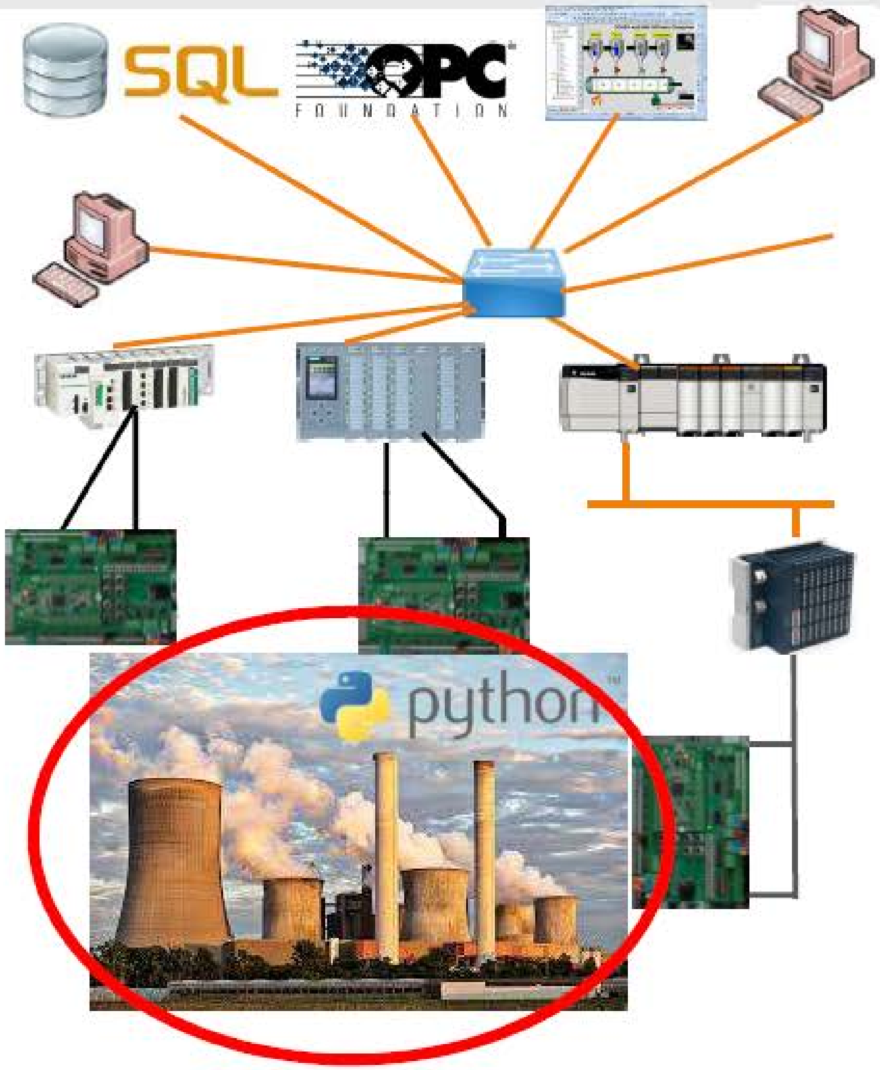
Cybersecurity Institute
Univ. Grenoble Alpes

61850 SMART GRID APPLICATIONS



- **Cybersecurity applications (M. Kabir-Querrec PhD benchmark)**
  - **False event injection (GOOSE attack)**
  - **False measures injection (SMV attack)**
  - **Resilient architectures**
    - *C&ESAR 2015*

INDUSTRIAL PROCESS SIMULATOR



NEEDS FOR PROCESS SIMULATOR

- **Genericity**
  - Ability to model any industrial process (e.g., electricity distribution, chemical factory)
  - Not focused only on a single application type
  - Easily customizable
- **Fine grain modeling**
  - Not our priority at the moment /!\
  - We focus on basic functionalities (e.g., “opening a valve” and not on the intrinsic properties of the valve), yet finer modeling can be required.
- **Ability to communicate /!**
  - With interface cards (thus with PLC, SCADA, etc)
  - In TCP/RTU Modbus
  - With data bases, API, etc
  - With GPIOS
- **Ability to run on various systems**
  - Raspberry PI, computer, etc

EXISTING PROCESS SIMULATORS

- **A real-time (time stepped simulation is needed)**
  - **Process dynamics simulated at real-life time not computer time**

- **Tested and validated simulators :**
  - **Modelica (with real-time) toolbox**
    - Native UDP communication implementation
  - **Factory I/O**
    - Engine I/O library -> interface provided
  - **Home I/O**
    - Engine I/O library -> interface provided
  - **Matlab/Simulink**
    - Native UDP communication implementation

PROGRAM AND CONFIGURATION

- Built on Atollic TrueSTUDIO® for STM32 V9.2.0

- One can configure
  - Mode (PLC or protection relay)
  - Network configuration
  - Background task
- Mode (gics.c)
- `static int config_mode = GICS_API; // configure for PLC`
- `static int config_mode = GICS_IED; // configure for protection relay`

NETWORK CONFIGURATION

- By default a card is idenfyied by a RACK_ID and a CARD_ID

- Ethernet config (ethernetif.c)
- `MACAddr[0] = 0x02;`
- `MACAddr[1] = 0x61;`
- `MACAddr[2] = 0xc5;`
- `MACAddr[3] = RACK_ID;`
- `MACAddr[4] = 0x00;`
- `MACAddr[5] = CARD_ID;`
  - ▶ **You can use any values instead default ones but respect the rules :**
    - Each card into the same network has a different MACAddr
    - First bit on byte 0 has to be 0 (otherwise it is a broadcast address).

IP CONFIGURATION

- File lwip.c
- IP_ADDRESS[0] = 10;
- IP_ADDRESS[1] = 10;
- IP_ADDRESS[2] = 100;
- IP_ADDRESS[3] = RACK_ID*16+GICS_ID;
- NETMASK_ADDRESS[0] = 255;
- NETMASK_ADDRESS[1] = 255;
- NETMASK_ADDRESS[2] = 0;
- NETMASK_ADDRESS[3] = 0;
- GATEWAY_ADDRESS[0] = 10;
- GATEWAY_ADDRESS[1] = 10;
- GATEWAY_ADDRESS[2] = 255;
- GATEWAY_ADDRESS[3] = 254;

CARD COMMUNICATION PROTOCOL

- **Simple requests derived from Modbus protocol**
- **Frame data defined in gics.h**
  - typedef struct GICSTransaction {
  -   unsigned char function;
  -   unsigned char magic;   always GICS_MAGIC 0xd0
  -   unsigned short length;
  -   unsigned short data[255];
  - } GICSTransaction;
- **Elementary Functions**
  - #define GICS_READ 0x01
  - #define GICS_WRITE 0x02
  - #define GICS_DISCRETE 0x04
  - #define GICS_ANALOG 0x08
  - #define GICS_DA 0x10

CARD COMMUNICATION PROTOCOL

- **Request functions**
  - Combination of READ/Write and data type
  - Read Analog/digital = GICS_READ + GICS_DA = 0x11
  - Write Analog/digital = GICS_WRITE + GICS_DA = 0x12
- **Anwsers**
  - Write requests are not answered
  - Read answers have a magic number 0xd1
- **Default UDP port = 2015**
- **A Wireshark dissector exists**

DATA EXCHANGE

- Write request (D/A)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.20.200	10.10.100.50	G-ICS	64	Request Write DigitalAnalog
2	0.099809	10.10.20.200	10.10.100.50	G-ICS	46	Request Read DigitalAnalog
3	0.101450	10.10.100.50	10.10.20.200	G-ICS	64	Answer Read DigitalAnalog

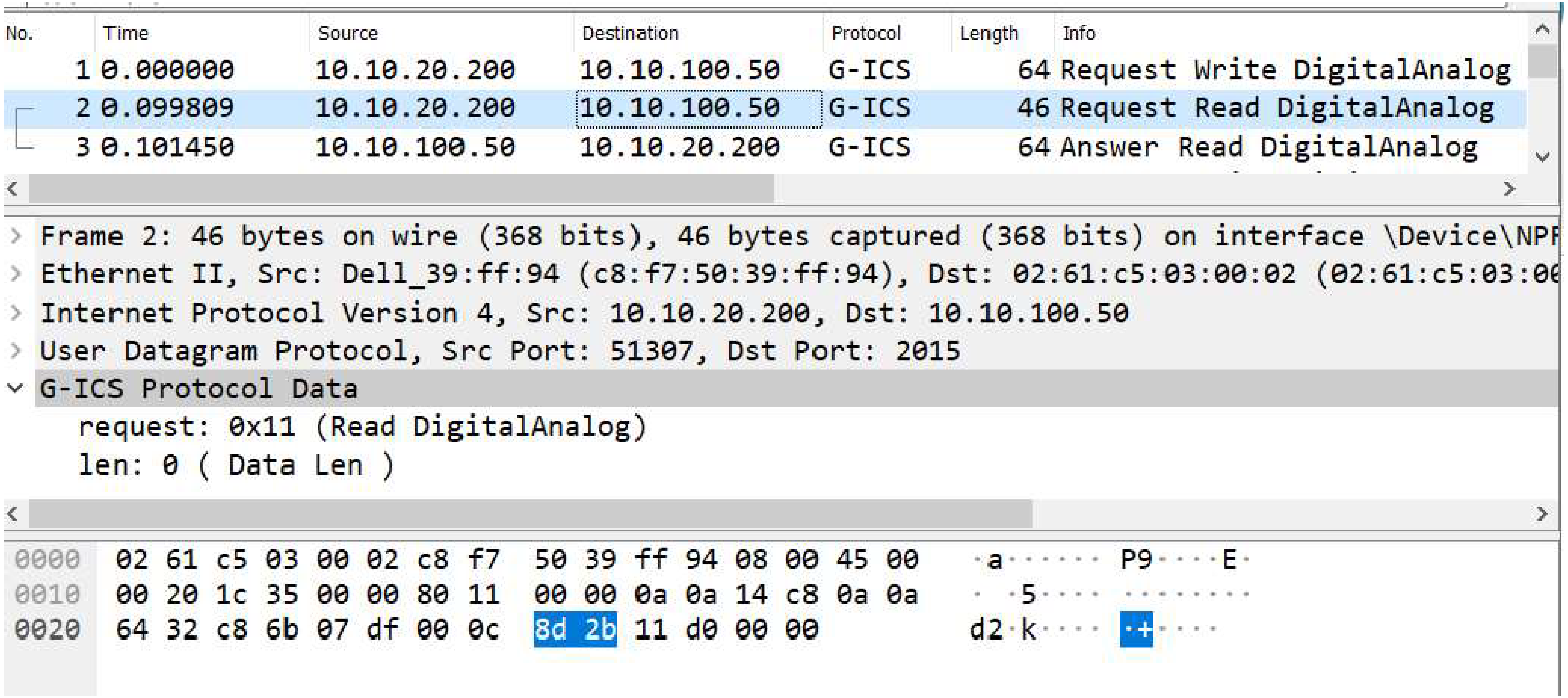
```
> Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface \Device\NPF
> Ethernet II, Src: Dell_39:ff:94 (c8:f7:50:39:ff:94), Dst: 02:61:c5:03:00:02 (02:61:c5:03:00
> Internet Protocol Version 4, Src: 10.10.20.200, Dst: 10.10.100.50
> User Datagram Protocol, Src Port: 51306, Dst Port: 2015
v G-ICS Protocol Data
    request: 0x12 (Write DigitalAnalog)
    len: 18 ( Data Len )
v Analog Inputs to PLC
    Analog In1:512
    Analog In2:512
    Analog In3:512
    Analog In4:512
    Analog In5:512
    Analog In6:512
    Analog In7:512
    Analog In8:512
  Digital Inputs to PLC:0000 0001 0000 0000

0000  02 61 c5 03 00 02 c8 f7  50 39 ff 94 08 00 45 00   ·a······ P9····E·
0010  00 32 1c 34 00 00 80 11  00 00 0a 0a 14 c8 0a 0a   ·2·4···· ········
0020  64 32 c8 6a 07 df 00 1e  8d 3d 12 d0 00 12 02 00   d2·j···· ·=······
0030  02 00 02 00 02 00 02 00  02 00 02 00 02 00 01 00   ········ ········
```

G-ICS Protocol (g-ics), 22 byte(s) | Paquets: 24 · Affichés: 24 (100.0%) | Profile: Default

Cybersecurity Institute
Univ. Grenoble Alpes

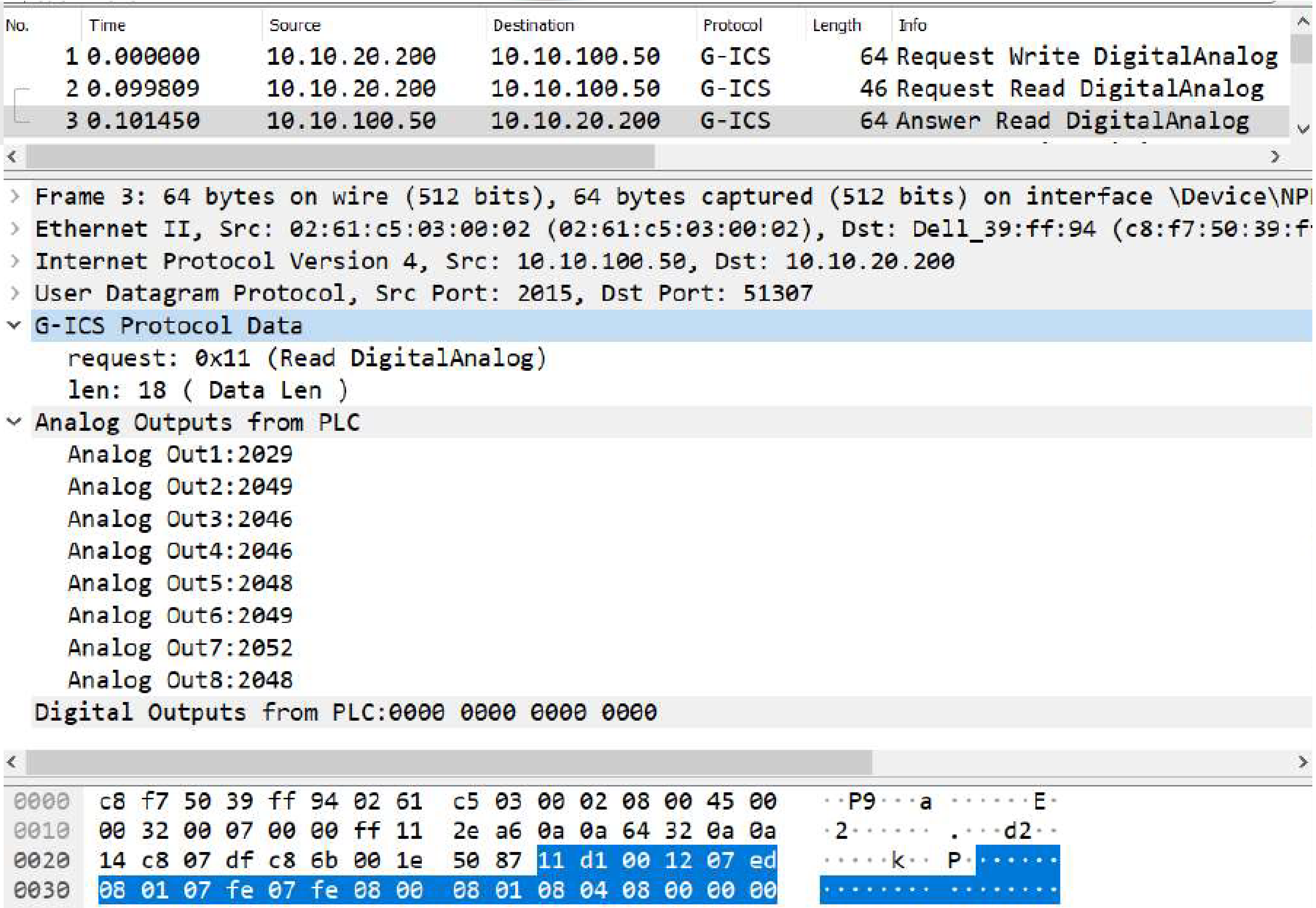
DATA REQUEST

- **Read D/A request**



DATA REQUEST

- **Read D/A Answer**



PROGRAM MAIN LOOP (GICS.C)

- Request decoding and DI/O
  - gics_udp_server_receive_callback
- Interrupt DA/C updater
  - Update_DAC_Handler
- Analog I/O : directly handled via memory transfer
- Protection relay signal adapter (sinusoïdal signals generation)
  - gics_update_DAC_vars

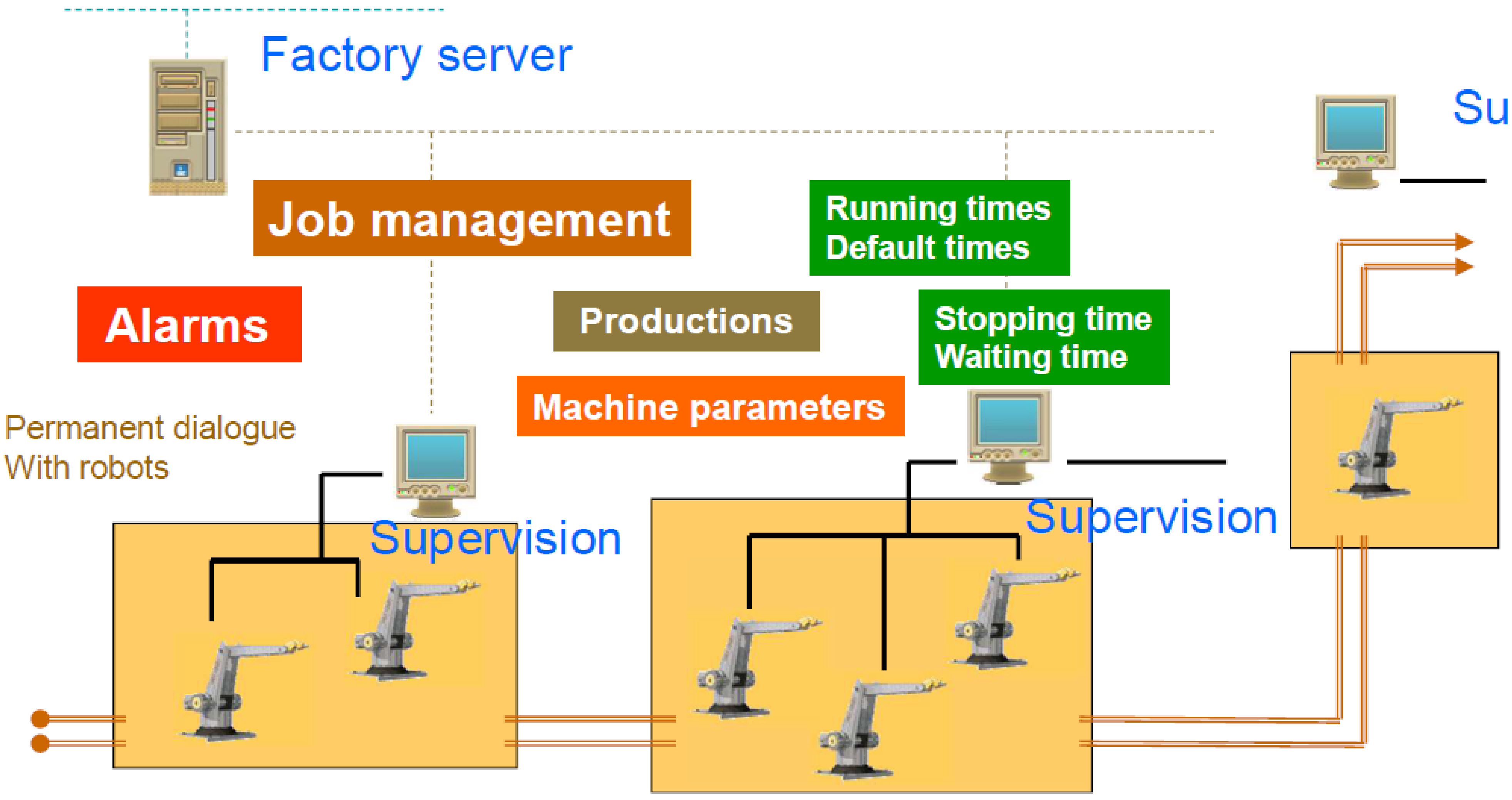
- A/D and D/A conversion rules
- DAC are 10 bits precision. Therefore DAC inputs are 0…1023 for a +/- 10V output
- ADC are 12 bits precision. For à +/-10V input, the output will vary between 0 and 4095
- Note that a small bias is to be expected

Classwork

- Synthesize a UDP frame to
  - Request data from GICS card
  - Write the set of boolean {1,0,1,0,1,0,1,0} to the output no. 1 – 8 of the GICS card

Lecture 02: Process monitoring

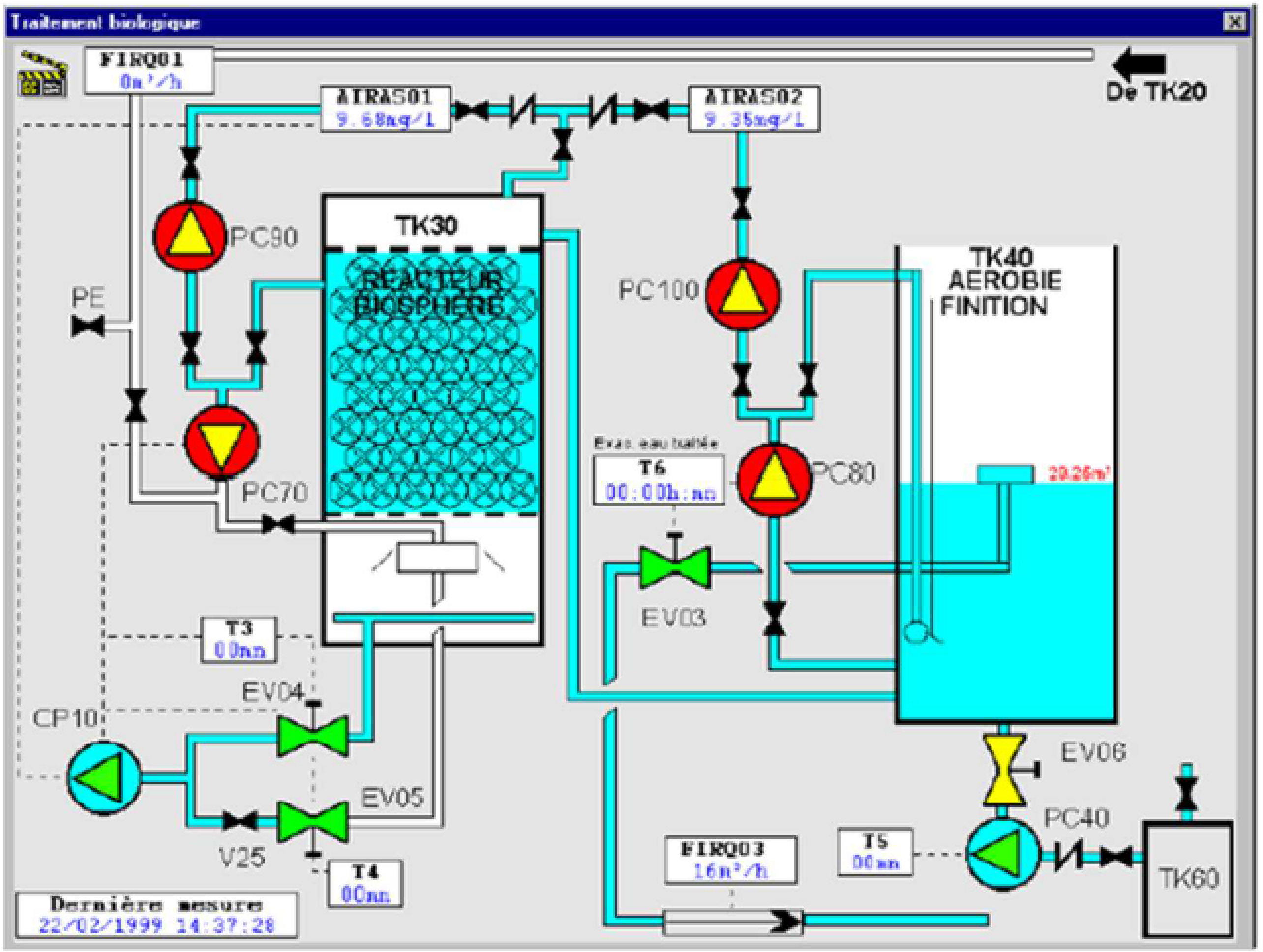
Supervision



Supervisors
Collect and process information from the robots
Basis of the factory's information system

Supervision functions

Synoptic: essential function of the supervision, provides a synthetic, dynamic and instantaneous representation of all the means of production of the unit

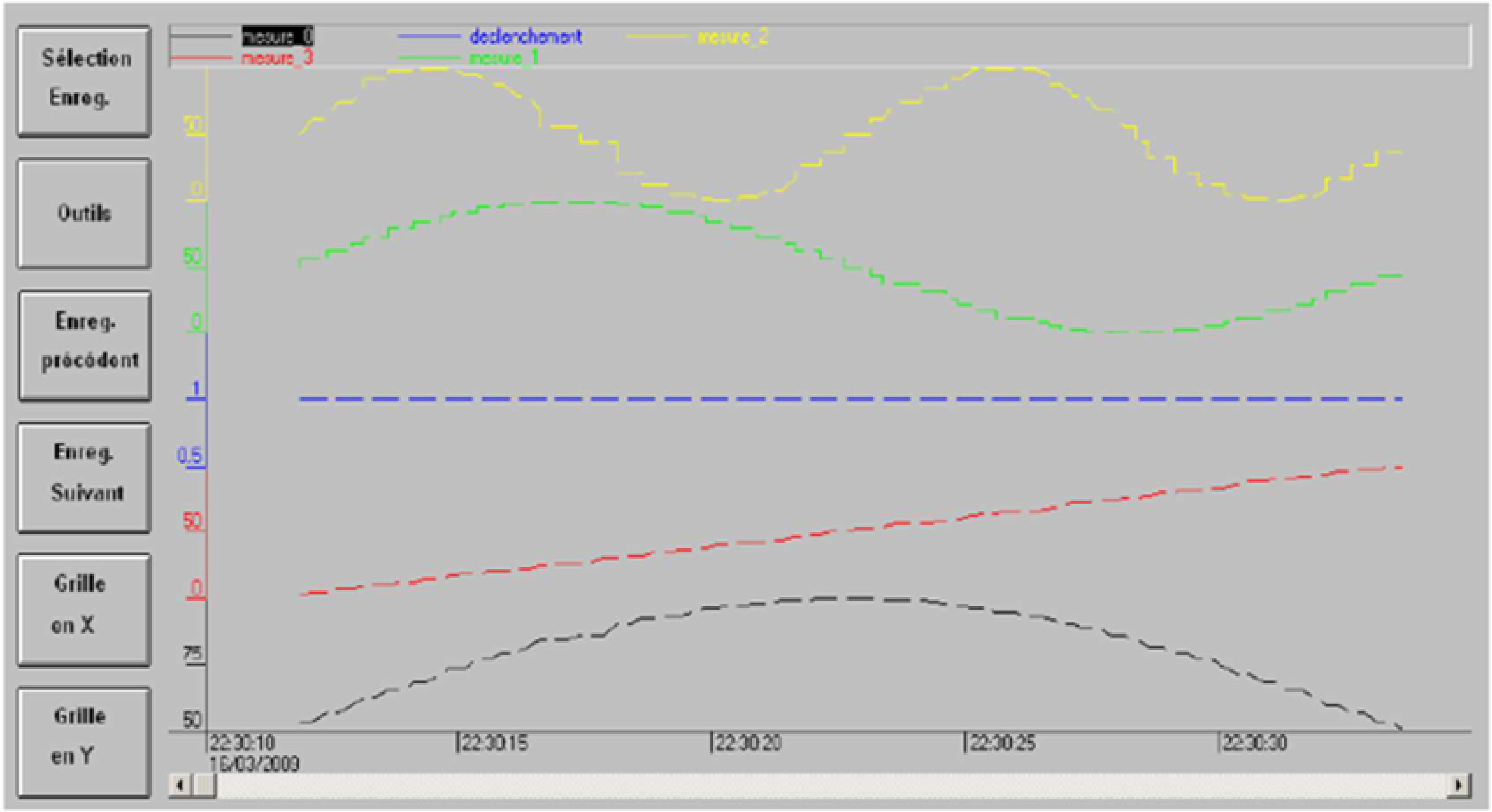


Supervision functions

Curves:

- gives a graphical representation of different process data

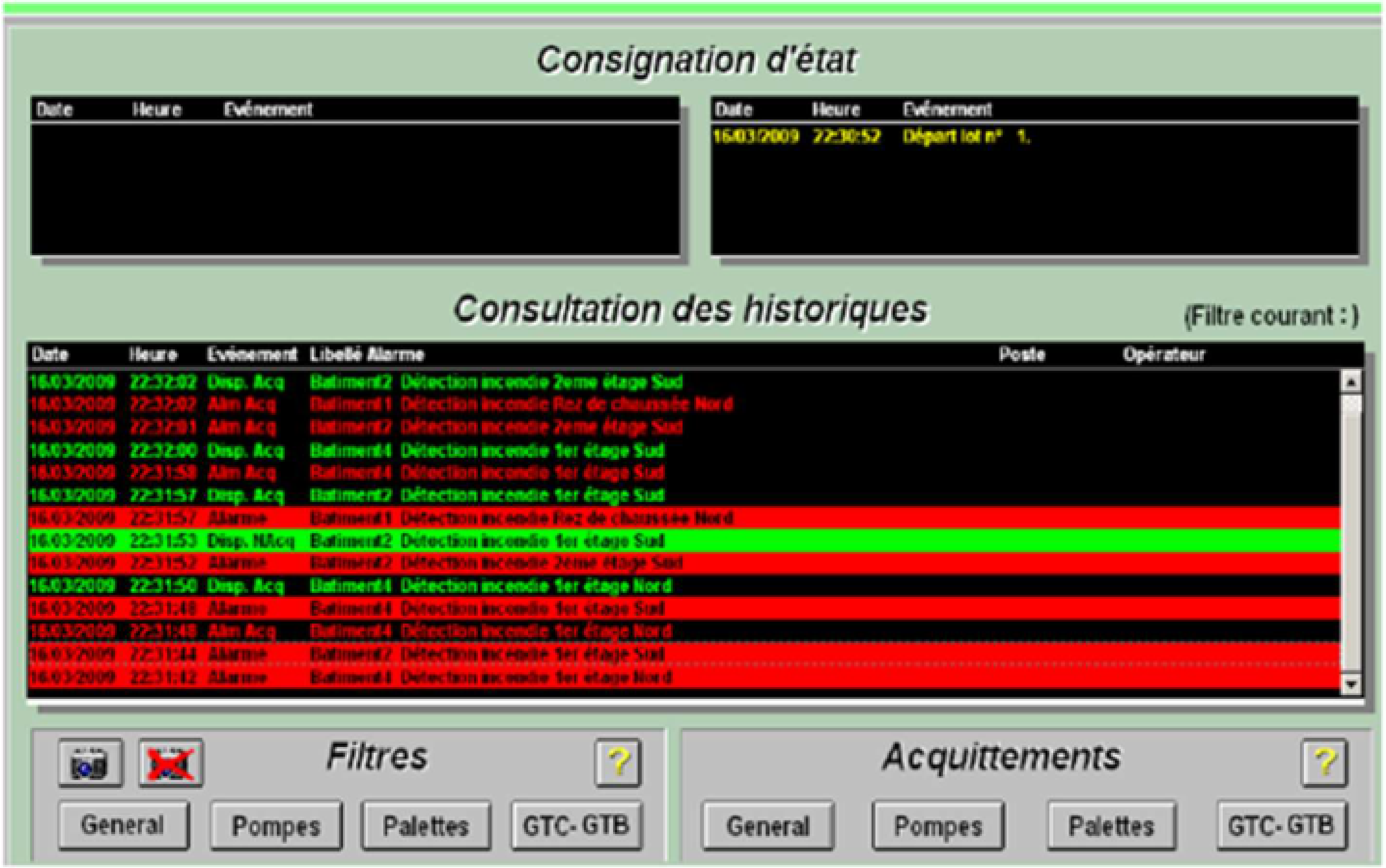
- gives the tools to analyze the historical variables



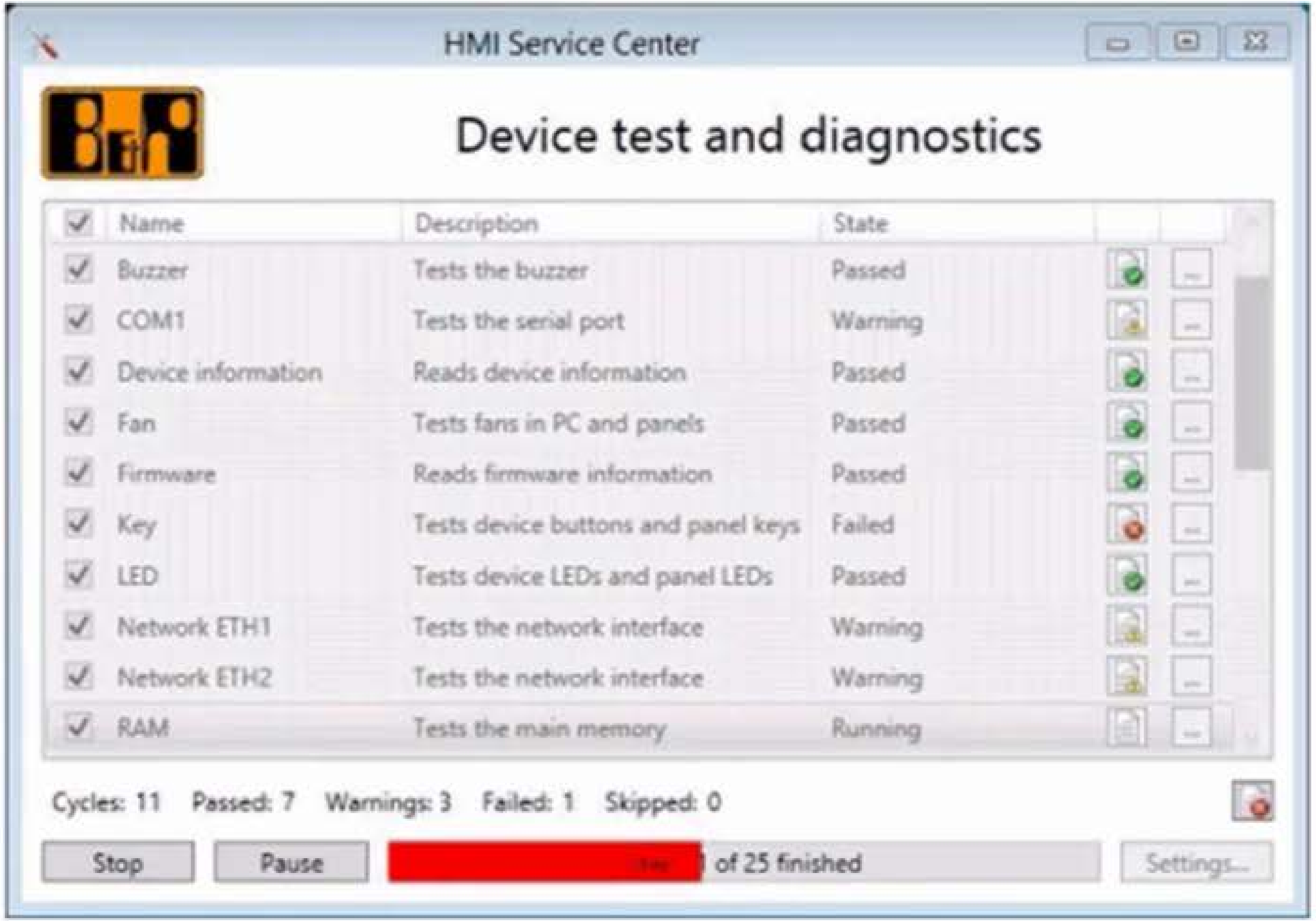
Supervision functions

Alarms

- Calculates in real time the conditions for triggering alarms
- Displays all alarms according to priority rules
- gives management tools
- ensures the recording of all the steps of the alarm processing



Supervision functions



FACTORI 4.0 Erasmus +

Alarms

UGA Université Grenoble Alpes

Circumscribe the cause of the feared event (cause of the incident)
Limit the impact of the event, protect (consequences)
Be able to assess the system after the incident: repair, reconfigure (total and partial redundancies)
Reconstruct, recover the system: time required for it to be operational again, what happens and what are the recovery steps? (Activity Return Plan)

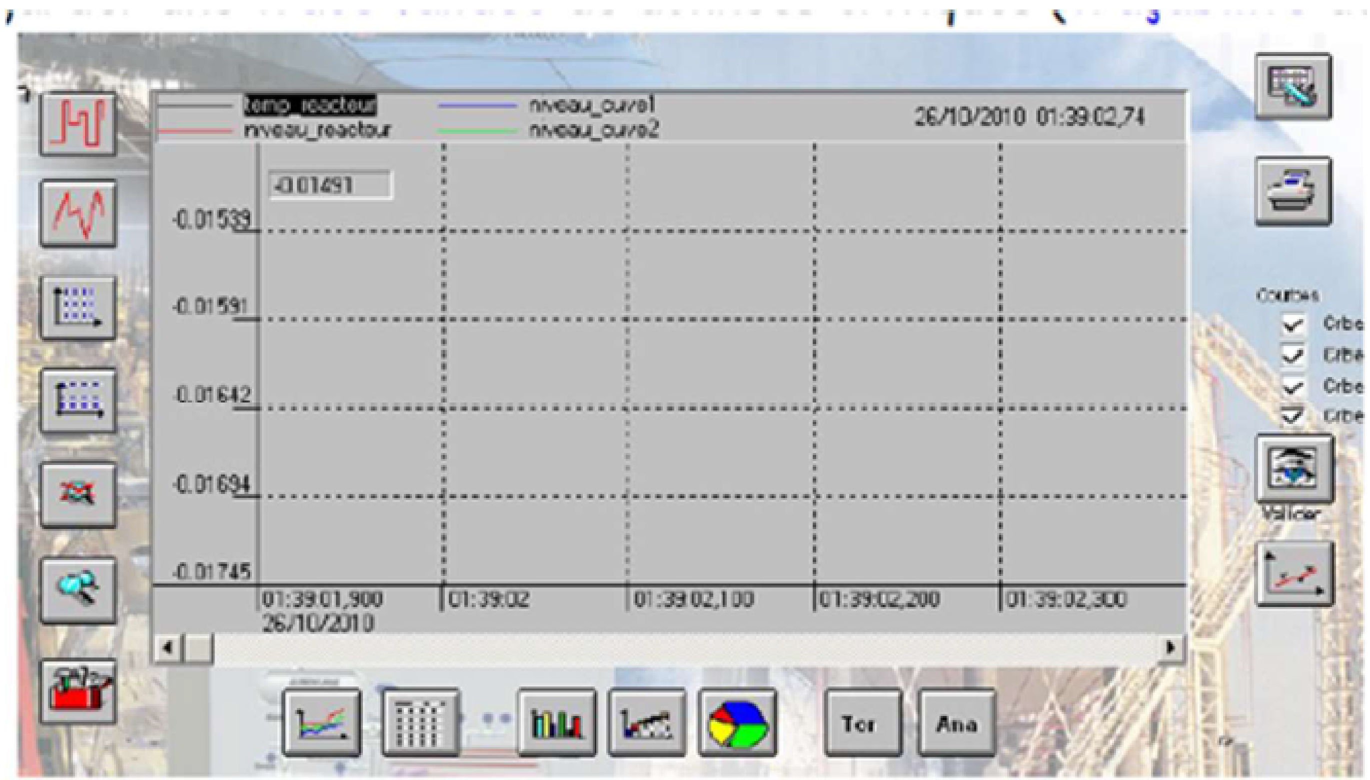
Other related aspects: robustness, resilience (ability to maintain the system as well as possible in a situation of "attacks")

Alarms detection

- TP (true positive) corresponds to correctly identified alarms
- FP (false positive) corresponds to authentic behavior identified as faulty
- TN (True Negative) corresponds to the correct rejection of authentic behavior
- FN (False Negative) corresponds to undetected failures
- Two metrics are used to evaluate the performance of alarm detection
  - True Positive Rate TPR=TP/(TP+FN)
    => 1 if no False Negative
  - False Positive Rate FPR=FP/(FP+TN)
    => 0 if no False Positive

Supervision functions

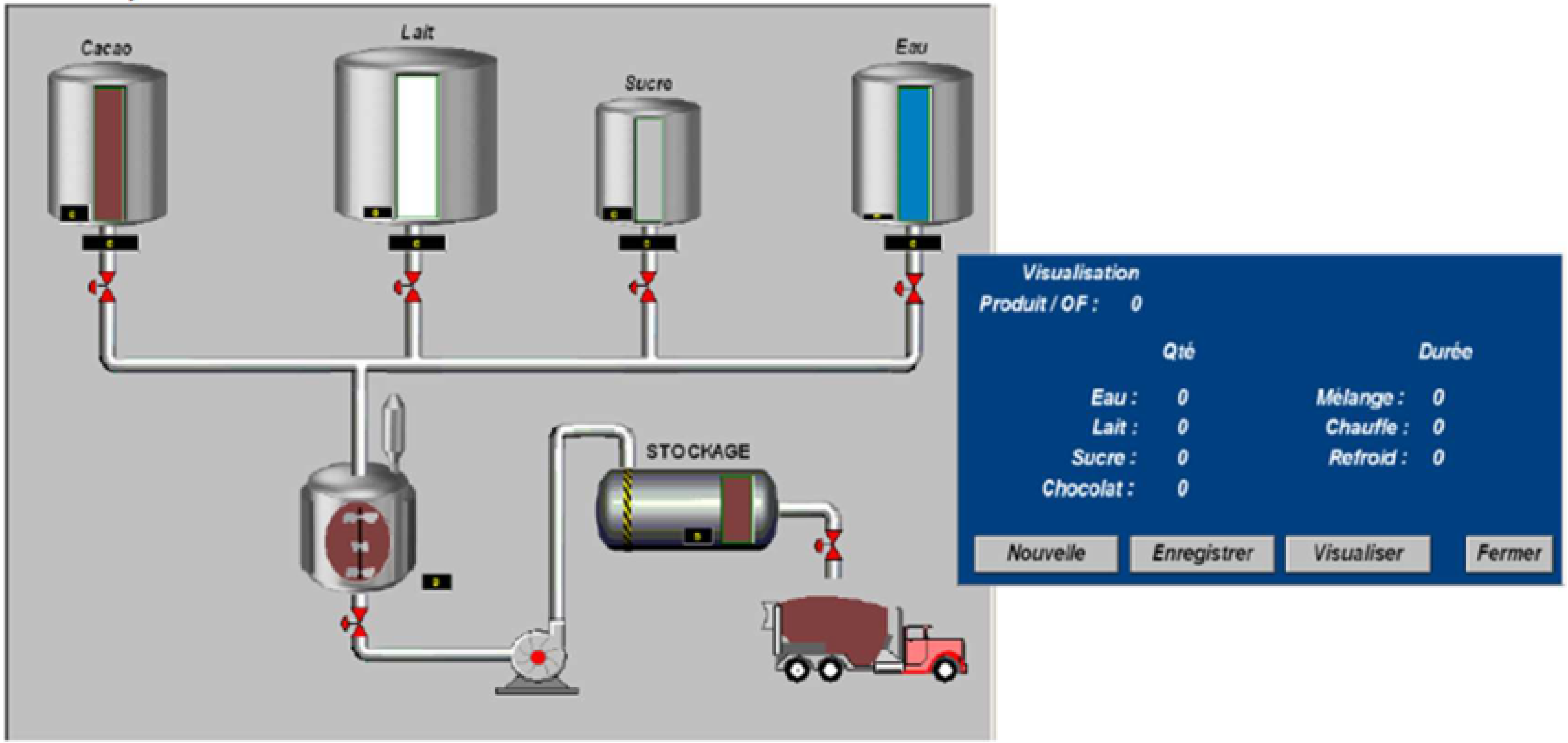
Historicization of the process:
- Allows the saving of time-stamped events (selective archiving)
- provides search tools in the archived years
- provides the possibility to run the synoptic again with archived data (replay function)
- allows to keep a validated trace of critical data (traceability of production data)



Supervision functions

Management of production lines and recipes:

- Provides a tool for managing production batches
- Manages the parameters of the machines for each batch (recipes)

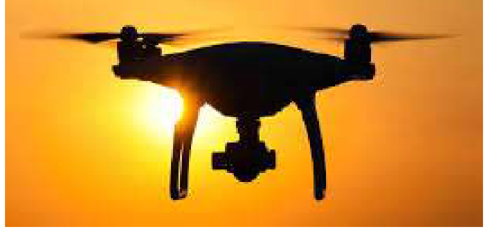


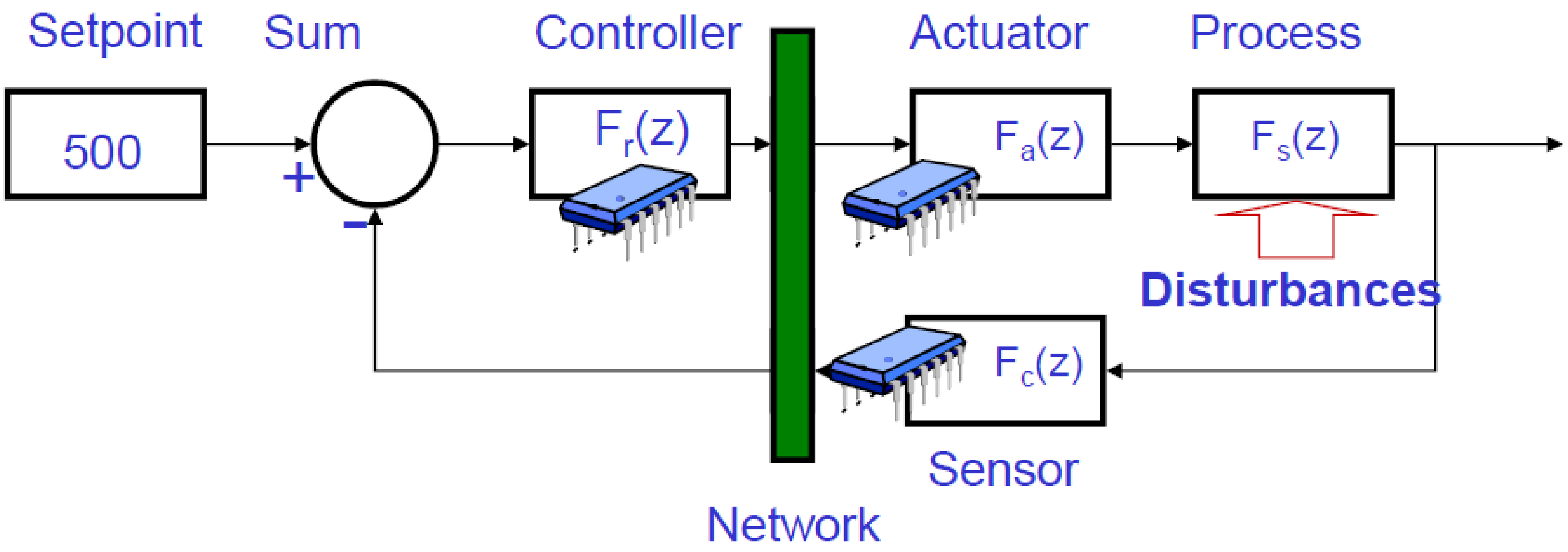
Classwork

- Select and study a manufacturing process and then design and list its monitoring parameters

Lecture 03: Process control

Networked Control Systems (NCS)





1. Continuous/sampled Components
2. Discrete events-components
3. Network influence
    1. Delays
    2. Jig
    3. Packet loss

→ Hybrid System

→ Time Delay system

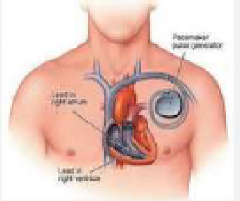
Quality of Service and Quality of Control for Safer Networked Control Systems (SafeNCS)

Communication networks are more and more used in control-based applications with real-time and/or critical constraints.

Communication and Control aspects need so to be seen from a global point of view. Communication (networks) is shared between various applications, and some aspects such as wireless communication and mobility needs to be taken into account in the design of SafeNCSs. Two examples of SafeNCS can be:

A drone with very strict real-time constraints => control-oriented,

A pacemaker which is remote-supervised from time to time, the infrastructure should protect strongly the integrity of supervision => security-oriented



Cyber-security of industrial systems is now a crucial issue => impact of cyber-security to the safety of networks

In both cases, focusing on control and/or on security aims at guaranteeing **safety**

Quality of Service and Quality of Control for Safer Networked Control Systems (SafeNCS)

Quality of service aims at guaranteeing the best communication aspects, focusing mainly on:

- security aspects: to protect the communication, in order to protect confidentiality of exchanges, integrity of data and control, authentication of actors of the SafeNCS.
- availability of the network, for the considered control application, by allowing the network to control the distribution of throughput as a function of the requirements of the applications (priorities of applications). For that, we can study the network protocols and mechanisms as well as the infrastructure.

Quality of control deals with the need of "automatic control" from the point of view of control, diagnosis, supervision...

- stability which means to guaranty the controllability of the system, despite the potential unavailability of the network
- performance which should be the best as possible in a varying environment, taking account of minimal levels of security, stability and safety.

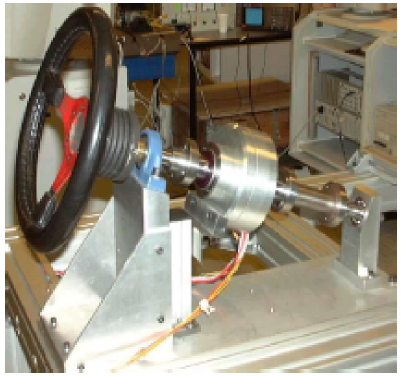
The presentation will present the problems, propose some approaches and results, and orientations concerning the study of Safe Networked Control Systems

Synthesis on the concepts

1. Dependability : Confidence in the system to ensure its mission without risk (or with a risk management)
   => Co-design approach (Network QoS ⇔ System QoC)
2. Functional safety: part of the overall safety that depends on a system or equipment operating correctly in response to its inputs [IEC]
3. Cyber-security: Cyber security is the protection of systems, networks and data in cyberspace [www.itgovernance.co.uk]
4. Networked Control Systems: Control System closed through a network
5. Complex systems, infrastructure, distributed systems
6. Embedded system, autonomous system, connected objects, IoT
7. ICS : Industrial Control Systems
8. Cyber-physical systems (CPS): Marrying physicality and computation [persyval-lab.org]
9. Our interest: To analyse CPS from the point of view of the potential impact of the system in the physical world (dependability point of view) due to a cyber-attack (attack in the digital world) and define the ways to protect it

Safety/Dependability level (RAMS) of a networked based system, wired networks

Drive shaft

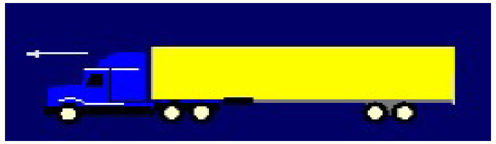


X by wire,
steering by wire

- Steering by wire
  - Probability that the vehicle doesn’t turn, when it is requested
  - Probability that the vehicle turns unexpectedly
- Difficult evaluation
  - Network more complex than a set of point-to-point links
  - Network more complex than a delayed system
  - Network-system Interaction

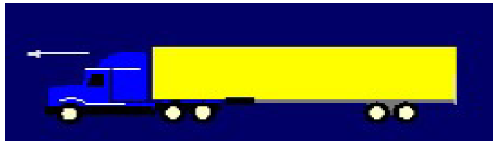
Dependability of networked-based systems (Wireless Network)

1st vehicle (controlled)



X by wire, brake by wire

2nd vehicle (following the 1st one)



Automated driving (virtual train)

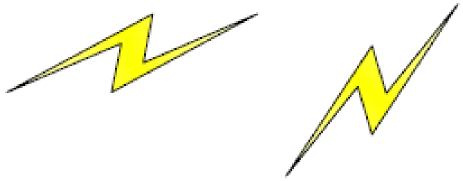
- **Braking Function**
  - First vehicle
    - Probability that the vehicle does not brake when it is asked for
    - Probability the vehicle brakes without any request
  - Second vehicle
    - Probability that it receives a braking information from the 1st vehicle, if everything is correct for the first vehicle
    - ...
- **Existing system**

  Verification model (formal approach, Monte-Carlo simulations)
- **Non existing system**

  Design model: « ideal » model + dependability constraints

Embedded system
(Embedded wired network + Remote wireless communication) with strong dynamics

Drone-helicopter



Definition of the mission

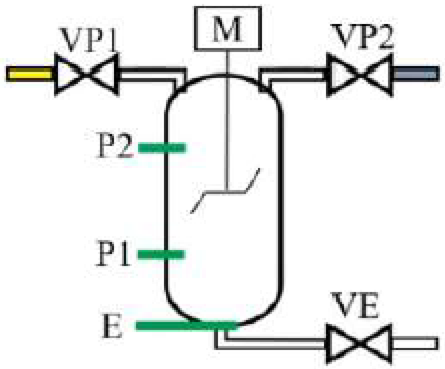
- Weak dynamics (normal displacement straight ahead)
- Strong dynamics (ex : slaloms between trees)
- Disturbed communication environment (e.m. disturbances, trees…)

Quality of service of the network

- High in critical situation
- High in strong dynamic situations (if remote-control)
- Lower other time

Example: control specification

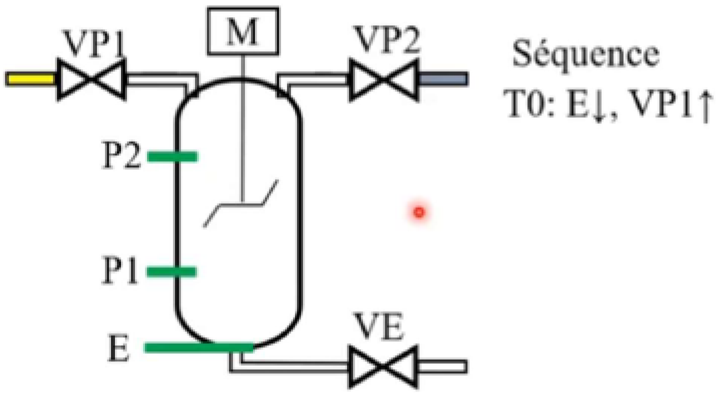
- Sequential behaviour: finite state machine, determitist automata, graphs with states and transitions
- Outputs are function of inputs and internal state of the automaton



- Process: mixture (blending) between two products in a tank containing a blender
- Mixture between Product 1 (yellow) and product 2 (blue)
- The blender is actuated by an engine (M)
- At the end the mixed product (green) will be evacuated through the valve VE

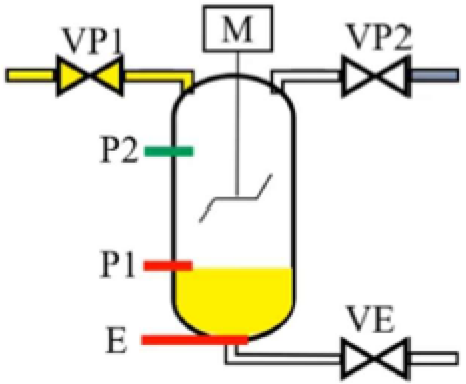
Example: control specification Transition 0

- Time T0
- When E switches from 1 to 0 (descending front), which means the tank is empty, this begins a new sequence



Example: control specification Transition 0 and 1

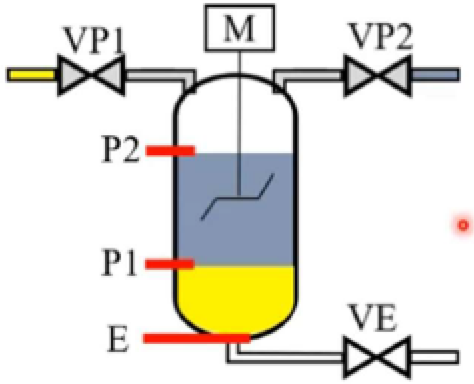
- Transition 0 : Opening of VP1 valve to add yellow product in the tank
- Once the P1 sensor (level) is reached => Transition 1: VP1 is closed, and VP2 (blue product) is open to add blue product



Séquence
T0: E↓, VP1↑
T1: P1↑, VP1↓, VP2↑

Example: control specification Transition 2

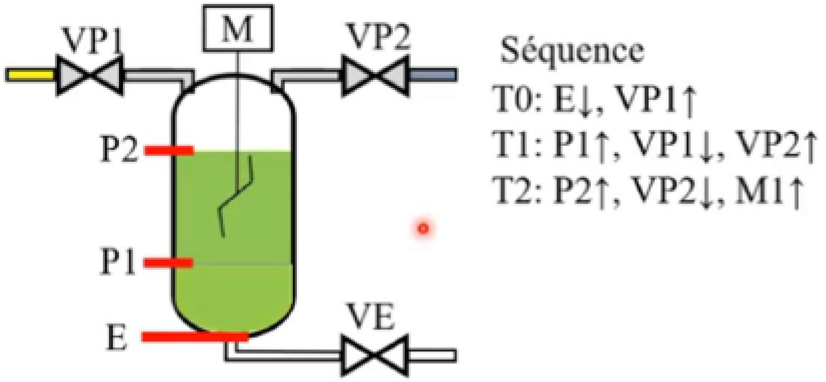
- Once the P2 sensor (level) is reached => Transition 2: VP2 is closed, and M (blender) is switches on



Séquence
T0: E↓, VP1↑
T1: P1↑, VP1↓, VP2↑
T2: P2↑, VP2↓, M1↑

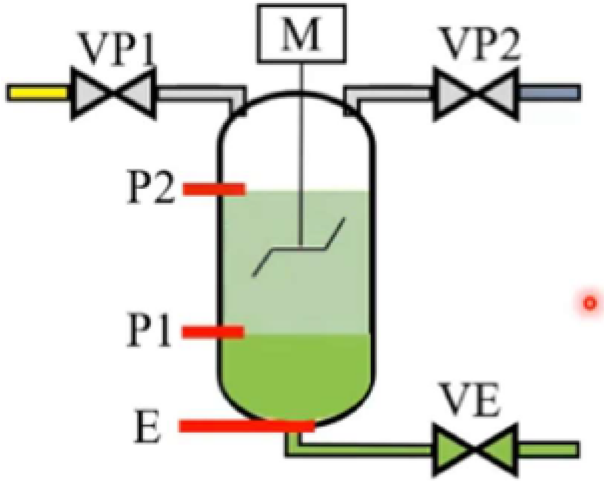
Example: control specification Transition 2

- The mixing (blending) operation will last a certain time. We use a timeout for that.



Example: control specification Transition 3

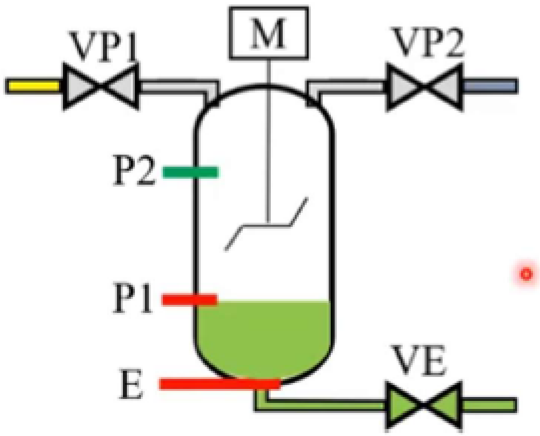
- At the end of the mixing, we stop the engine, and open VE, which is the evacuation valve.



Séquence
T0: E↓, VP1↑
T1: P1↑, VP1↓, VP2↑
T2: P2↑, VP2↓, M1↑
T3: M1↓, VE↑

Example: control specification Transition 4

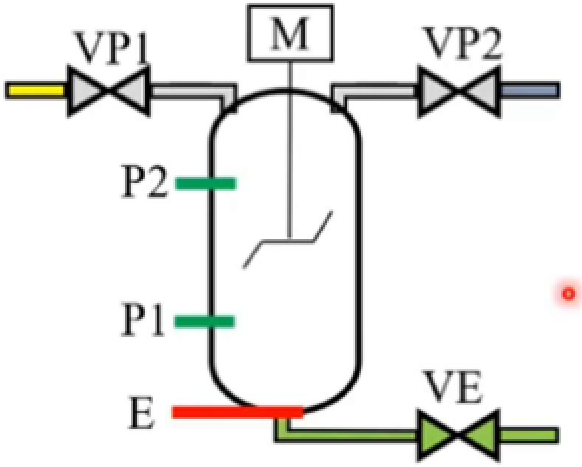
- The tank is emtying, P2 is deactivated



Séquence
T0: E↓, VP1↑
T1: P1↑, VP1↓, VP2↑
T2: P2↑, VP2↓, M1↑
T3: M1↓, VE↑
T4: P2↓

Example: control specification Transition 5

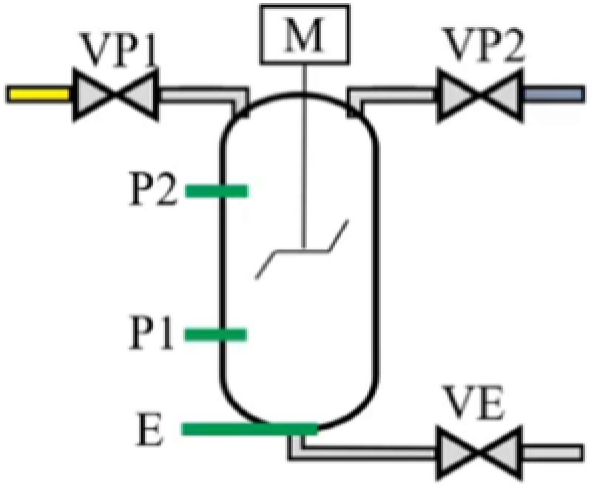
- Then P1 is deactivated



Séquence
T0: E↓, VP1↑
T1: P1↑, VP1↓, VP2↑
T2: P2↑, VP2↓, M1↑
T3: M1↓, VE↑
T4: P2↓
T5: P1↓

Example: control specification Transition 6

- VE is closed, the cycle is over
- We can begin again



Séquence
T0: E↓, VP1↑
T1: P1↑, VP1↓, VP2↑
T2: P2↑, VP2↓, M1↑
T3: M1↓, VE↑
T4: P2↓
T5: P1↓
T6: VE↓

Classwork

- Implement the ladder diagram of the mixture problem

1501324 Sequence Control and PLC

Program: Bachelor program in Computer Engineering

Credit: 3(2-2) Lecture: 30 Hours Lab: 30 Hours

FACTORI 4.0 Erasmus +

Co-funded by the Erasmus+ Programme of the European Union

2nd Semester, Academic Year: 2023

Assoc. Prof. Punnarumol Temdee, Ph.D.

Asst. Prof. Roungsan Chaisricharoen, Ph.D.

Asst. Prof. Santichai Wicha, Ph.D.

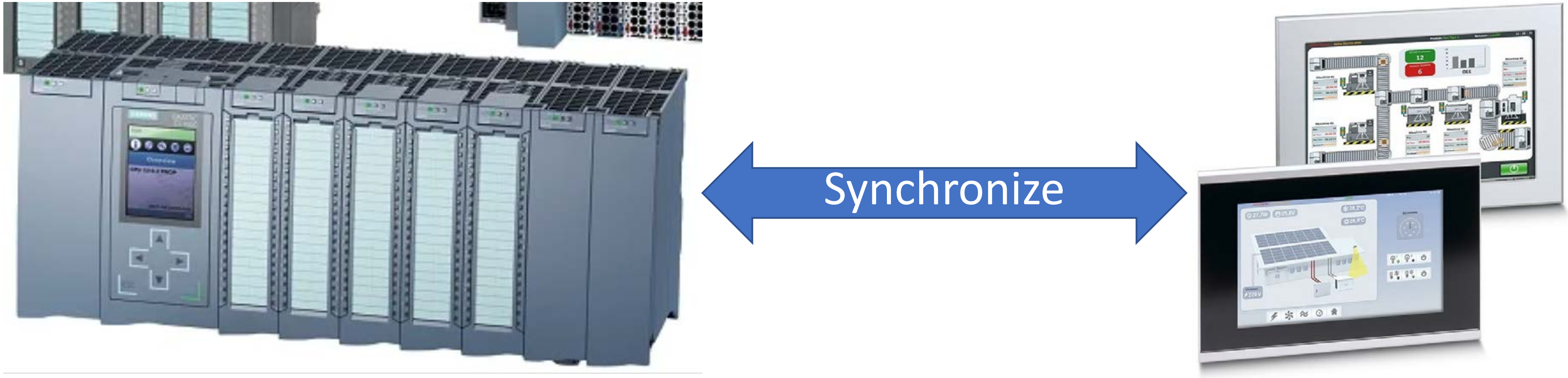
Lect. Chayapol Kamyod, Ph.D.

This course has been modified in the framework of an Erasmus + project: Asean Factori 4.0 Across South East Asian Nations: From Automation and Control Training to the Overall Roll-out of Industry 4.0

609854-EPP-1-2019-1-FR-EPPKA2-CBHE-JP

Lab 01: Process monitoring via HMI

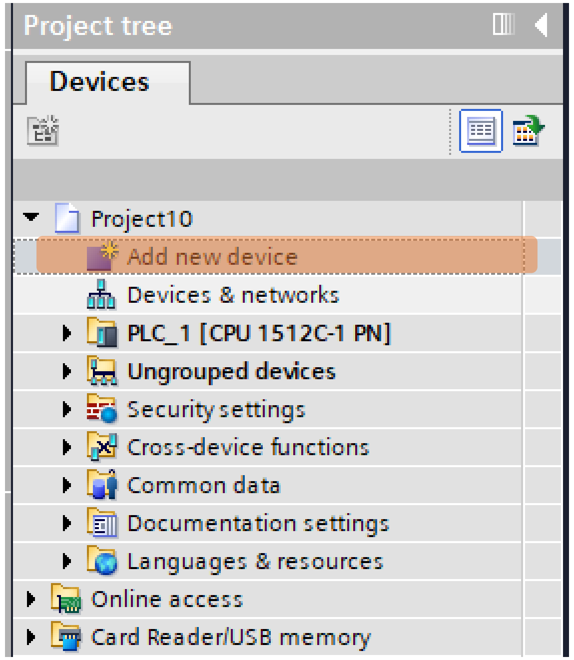
Human-Machine Interface

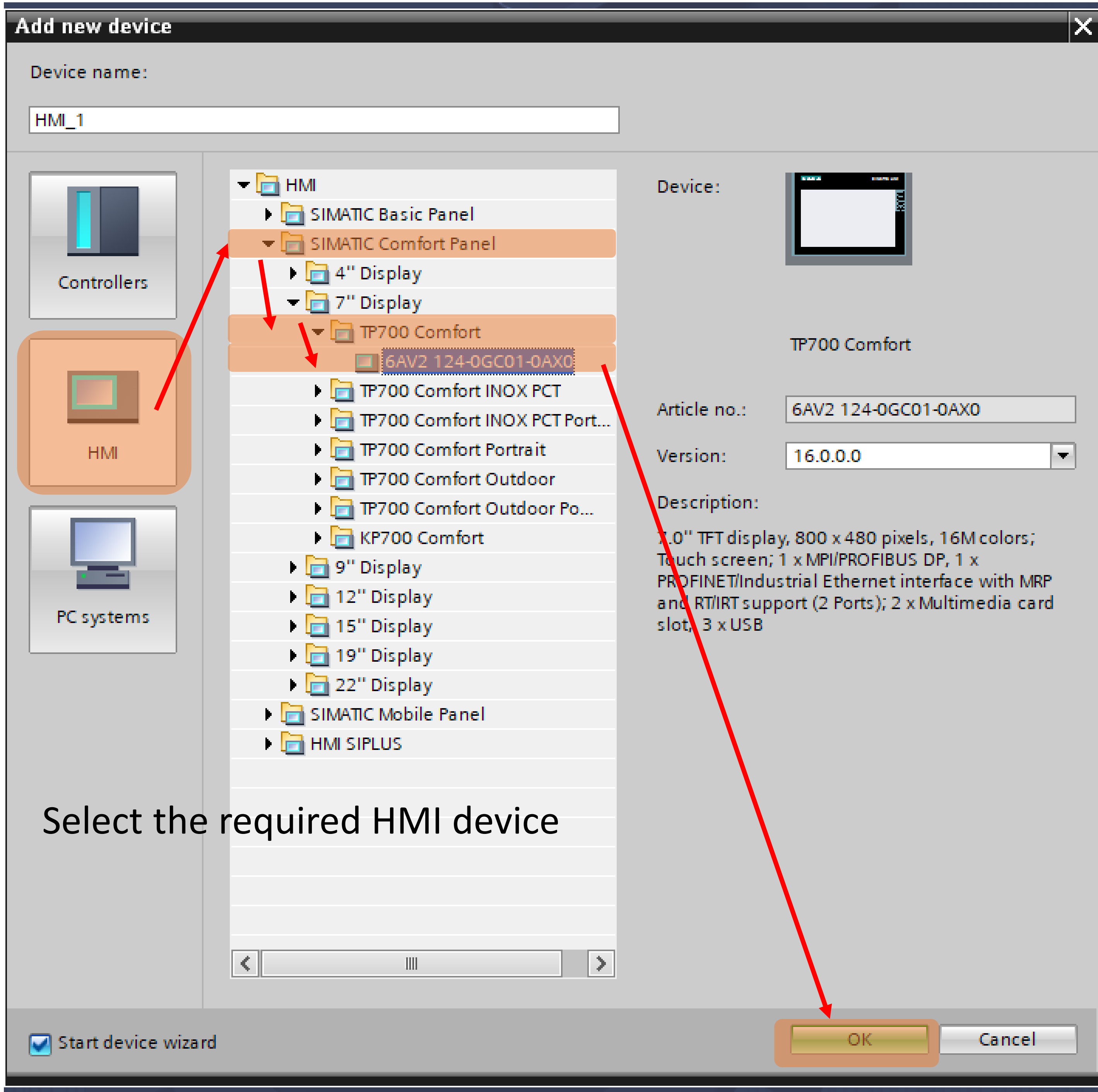


HMI provides a method of displaying information and obtaining inputs, modeling the control system as a whole. HMIs do not typically provide any way to modify the logic program

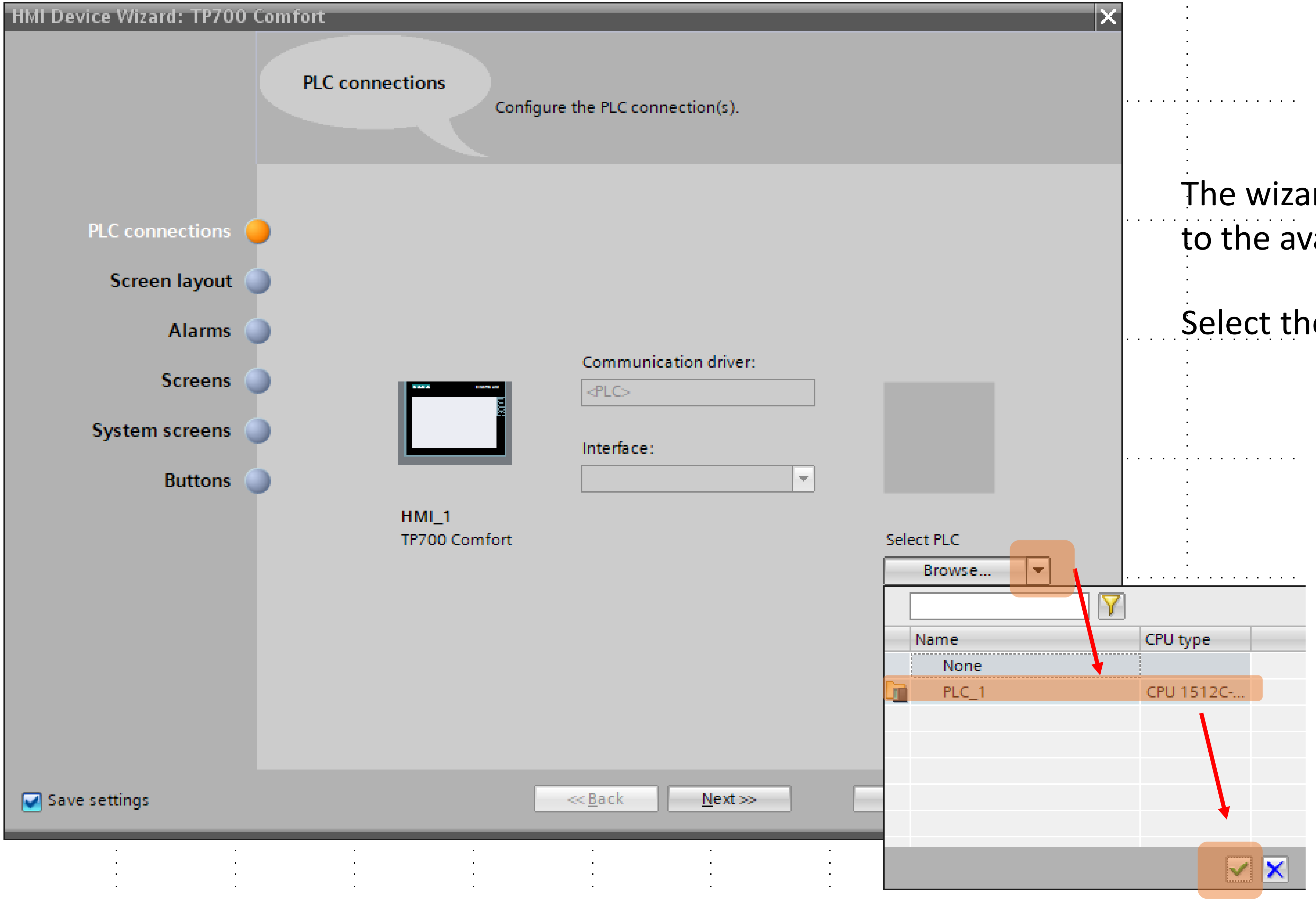
Add an HMI to the project

Double click the “Add new device” command



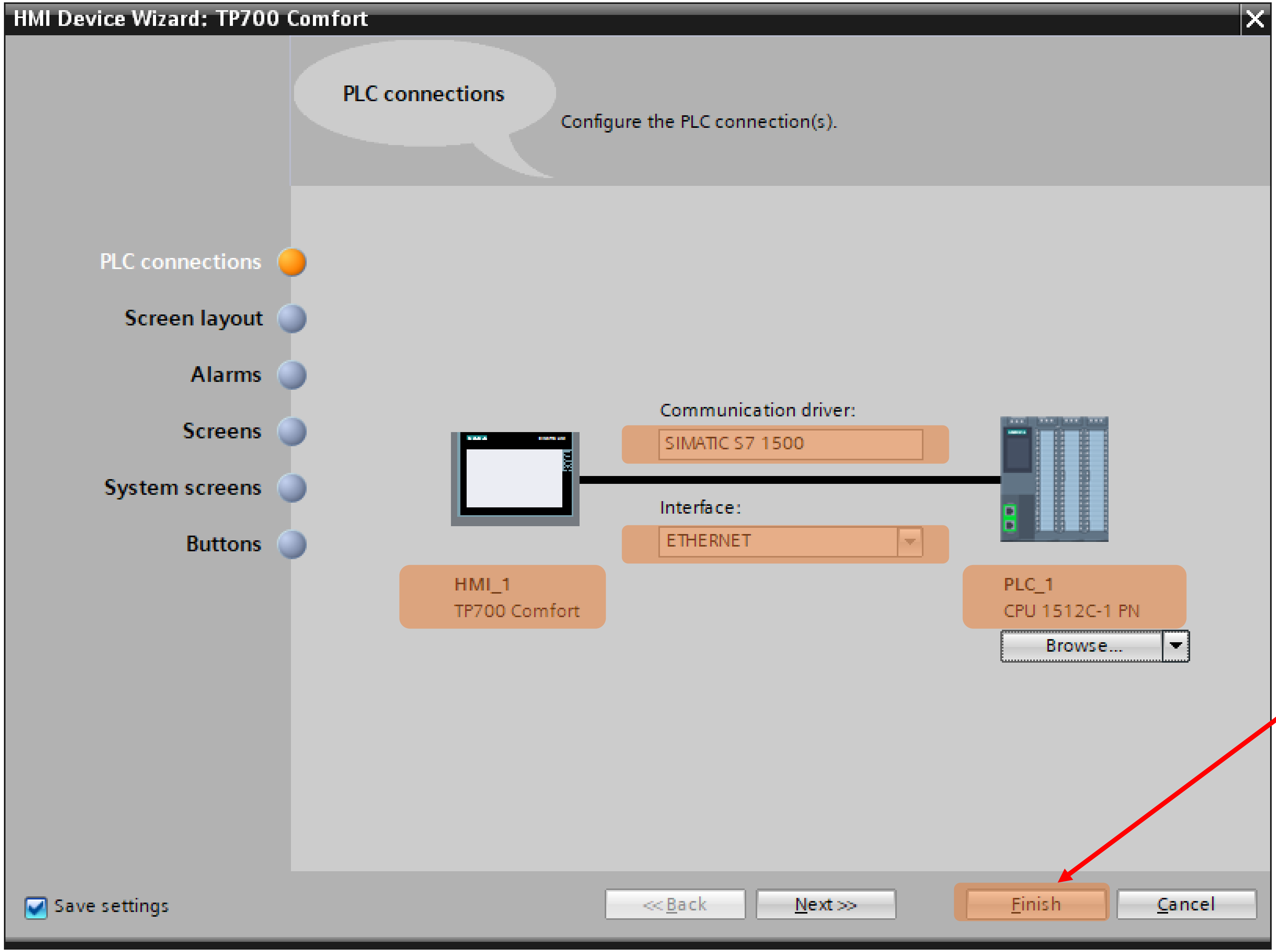


Select the required HMI device



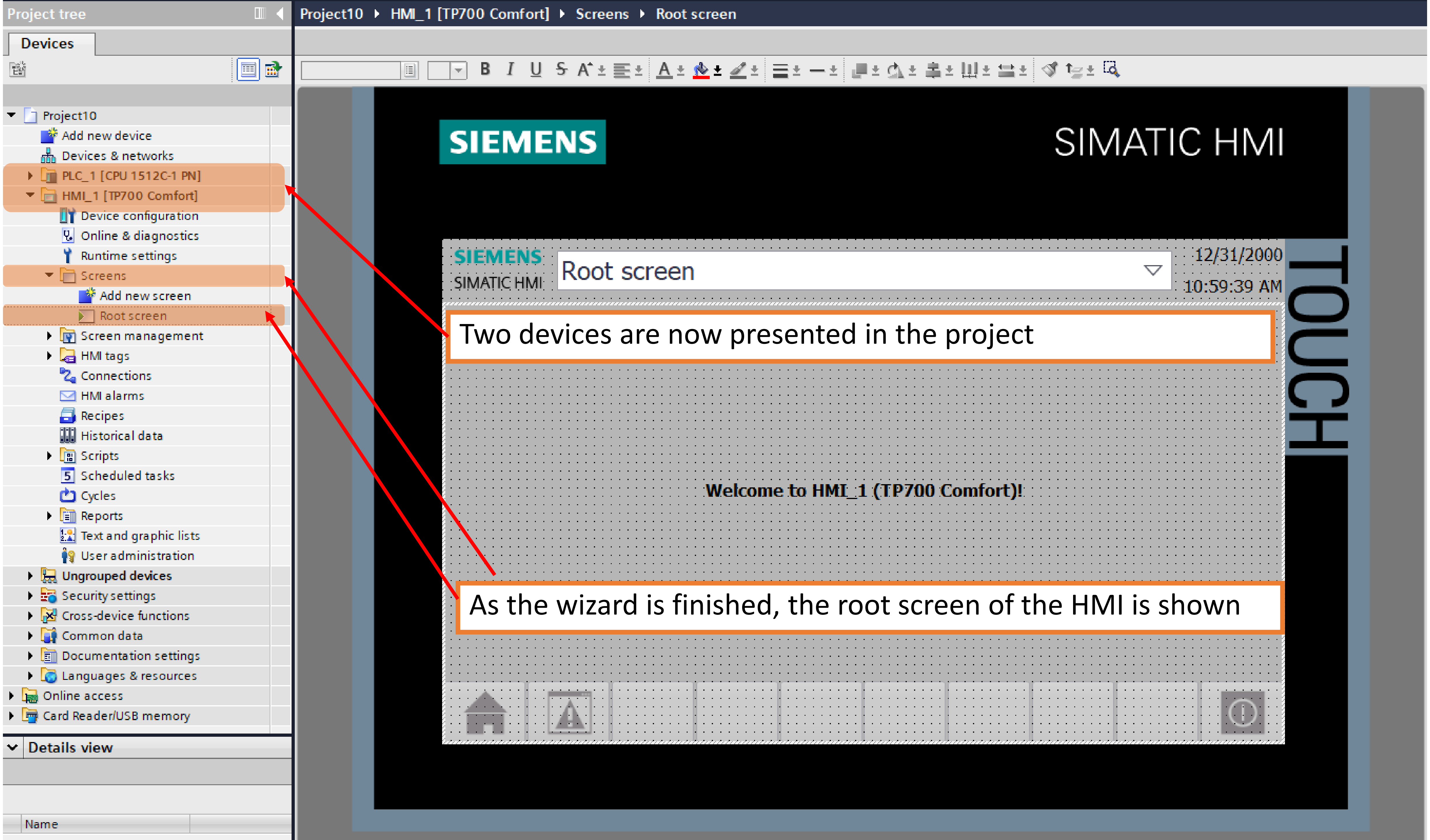
The wizard that helps connect to the available PLC is popped

Select the right PLC

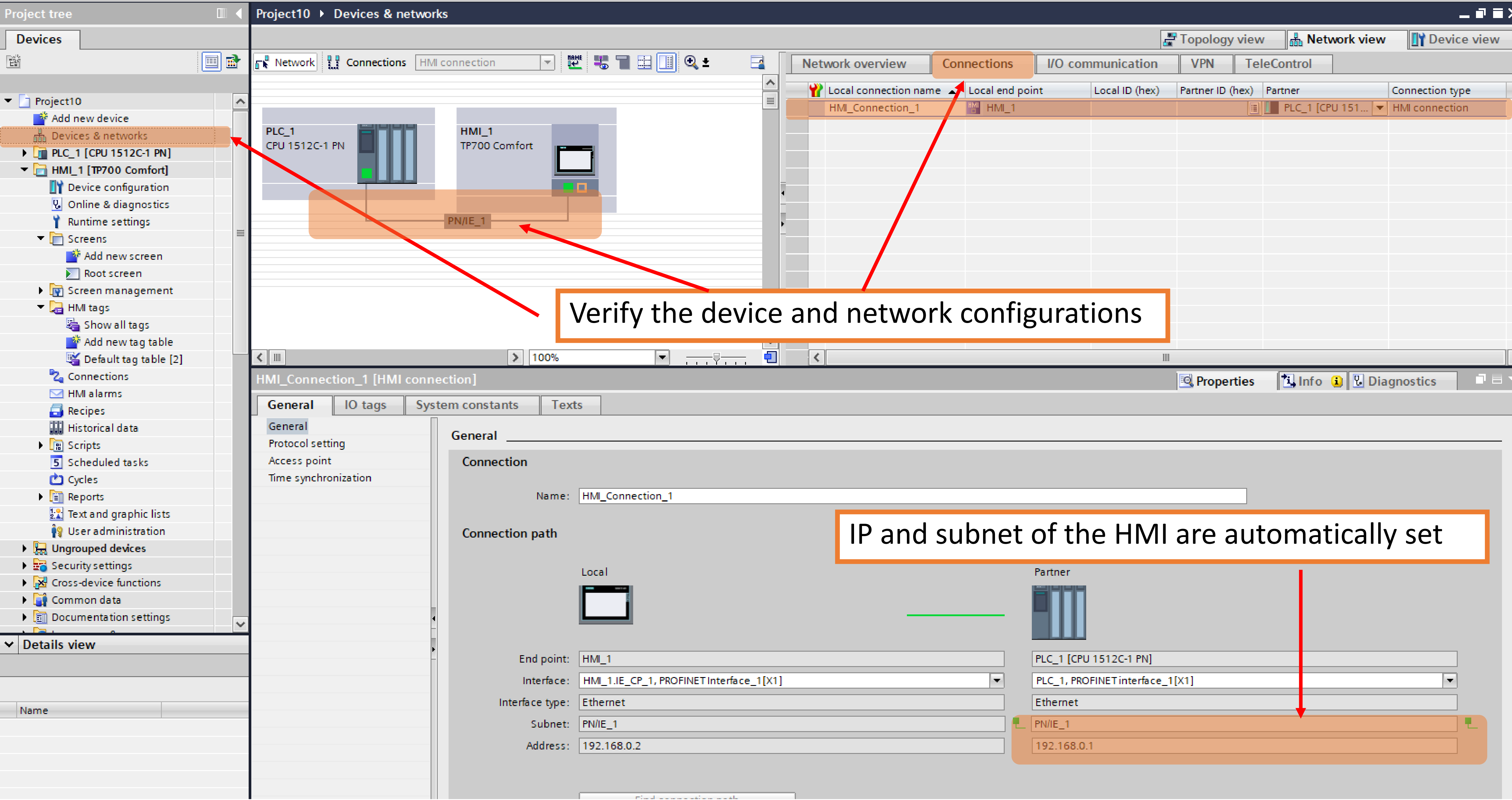


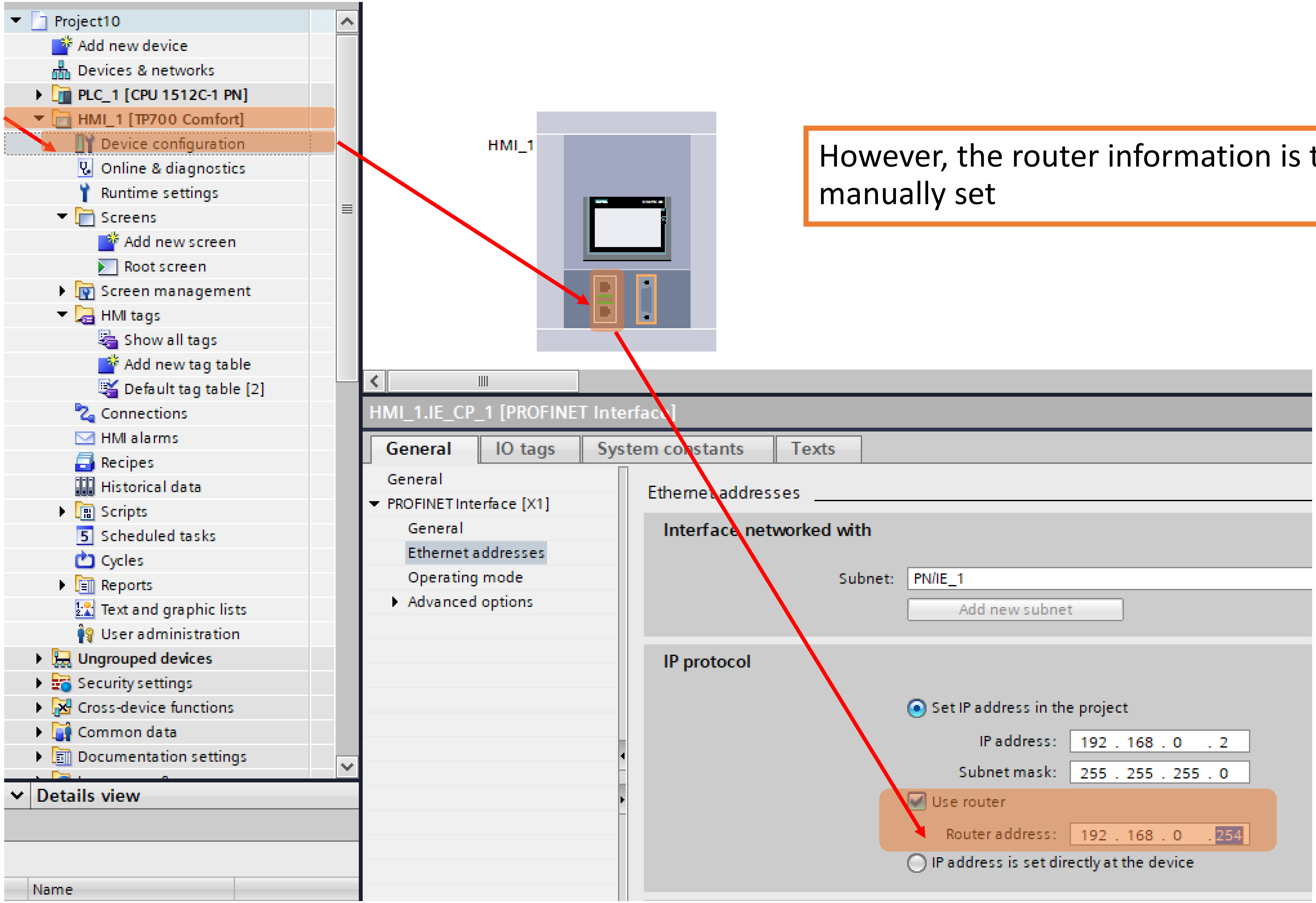
Check the connection's info

Finish the wizard as other steps are not necessary

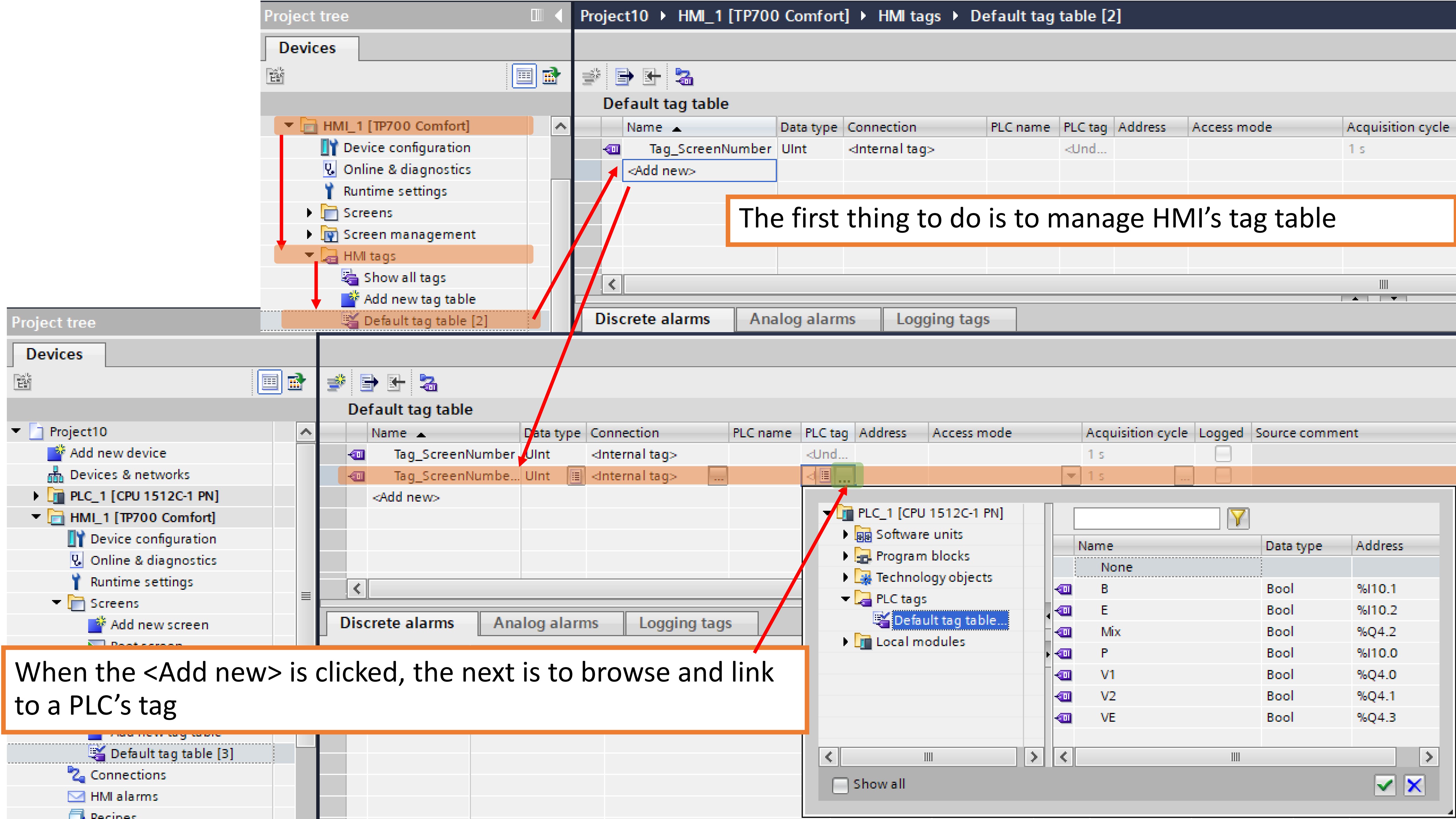


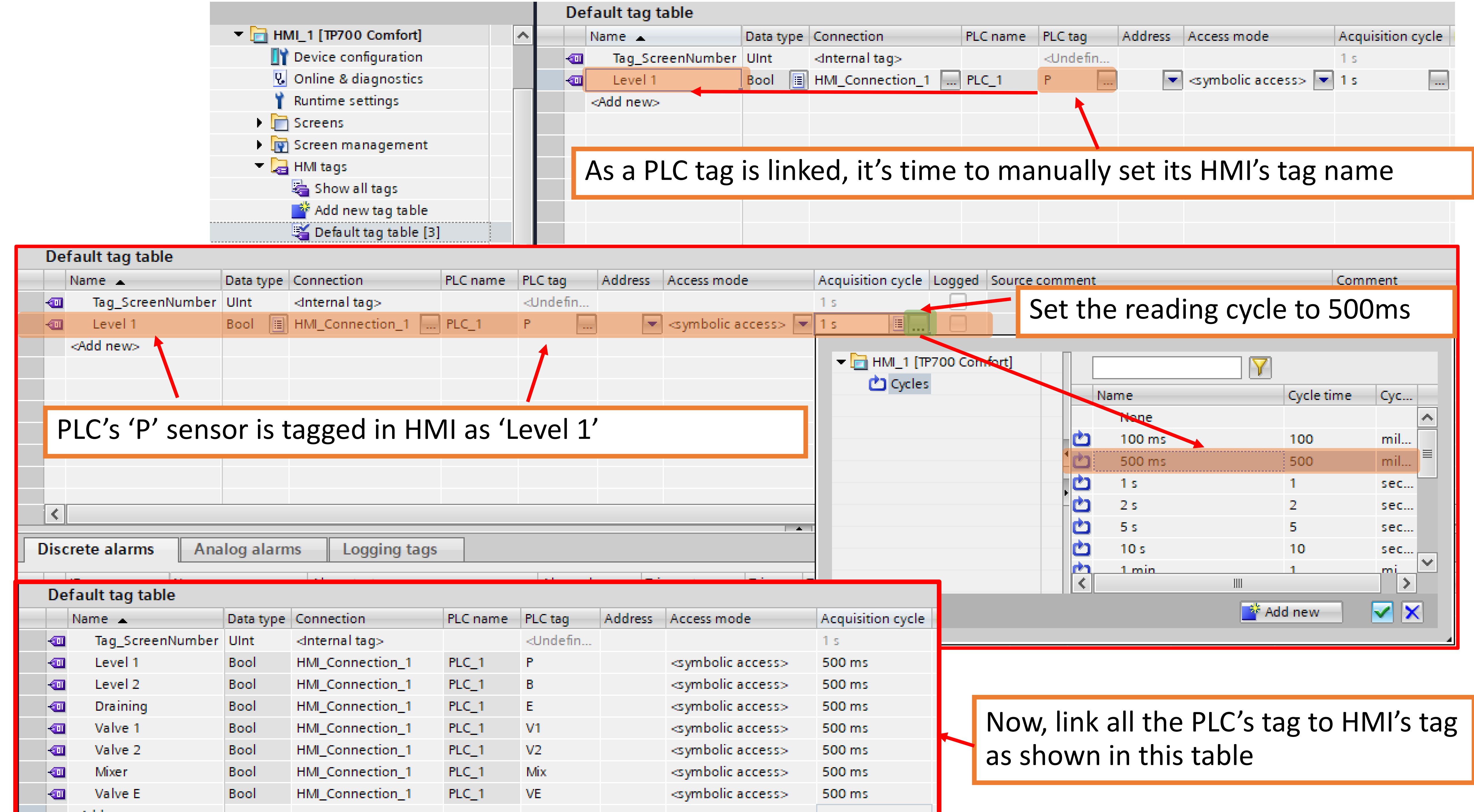
Project tree
Project10 ▸ HMI_1 [TP700 Comfort] ▸ Screens ▸ Root screen
Devices
Project10
Add new device
Devices & networks
PLC_1 [CPU 1512C-1 PN]
HMI_1 [TP700 Comfort]
Device configuration
Online & diagnostics
Runtime settings
Screens
Add new screen
Root screen
Screen management
HMI tags
Connections
HMI alarms
Recipes
Historical data
Scripts
Scheduled tasks
Cycles
Reports
Text and graphic lists
User administration
Ungrouped devices
Security settings
Cross-device functions
Common data
Documentation settings
Languages & resources
Online access
Card Reader/USB memory
Details view
Name
SIEMENS
SIMATIC HMI
TOUCH
SIEMENS
SIMATIC HMI
Root screen
12/31/2000
10:59:39 AM
Two devices are now presented in the project
Welcome to HMI_1 (TP700 Comfort)!
As the wizard is finished, the root screen of the HMI is shown


Verify the device and network configurations
IP and subnet of the HMI are automatically set
Project tree
Devices
Project10
Add new device
Devices & networks
PLC_1 [CPU 1512C-1 PN]
HMI_1 [TP700 Comfort]
Device configuration
Online & diagnostics
Runtime settings
Screens
Add new screen
Root screen
Screen management
HMI tags
Show all tags
Add new tag table
Default tag table [2]
Connections
HMI alarms
Recipes
Historical data
Scripts
Scheduled tasks
Cycles
Reports
Text and graphic lists
User administration
Ungrouped devices
Security settings
Cross-device functions
Common data
Documentation settings
Details view
Name
Project10 ▸ Devices & networks
Topology view
Network view
Device view
Network
Connections
HMI connection
Network overview
Connections
I/O communication
VPN
TeleControl
Local connection name
Local end point
Local ID (hex)
Partner ID (hex)
Partner
Connection type
HMI_Connection_1
HMI_1
PLC_1 [CPU 151...
HMI connection
PLC_1
CPU 1512C-1 PN
HMI_1
TP700 Comfort
PN/IE_1
100%
HMI_Connection_1 [HMI connection]
Properties
Info
Diagnostics
General
IO tags
System constants
Texts
General
Protocol setting
Access point
Time synchronization
General
Connection
Name: HMI_Connection_1
Connection path
Local
Partner
End point: HMI_1
PLC_1 [CPU 1512C-1 PN]
Interface: HMI_1.IE_CP_1, PROFINET Interface_1[X1]
PLC_1, PROFINET interface_1[X1]
Interface type: Ethernet
Ethernet
Subnet: PN/IE_1
PN/IE_1
Address: 192.168.0.2
192.168.0.1



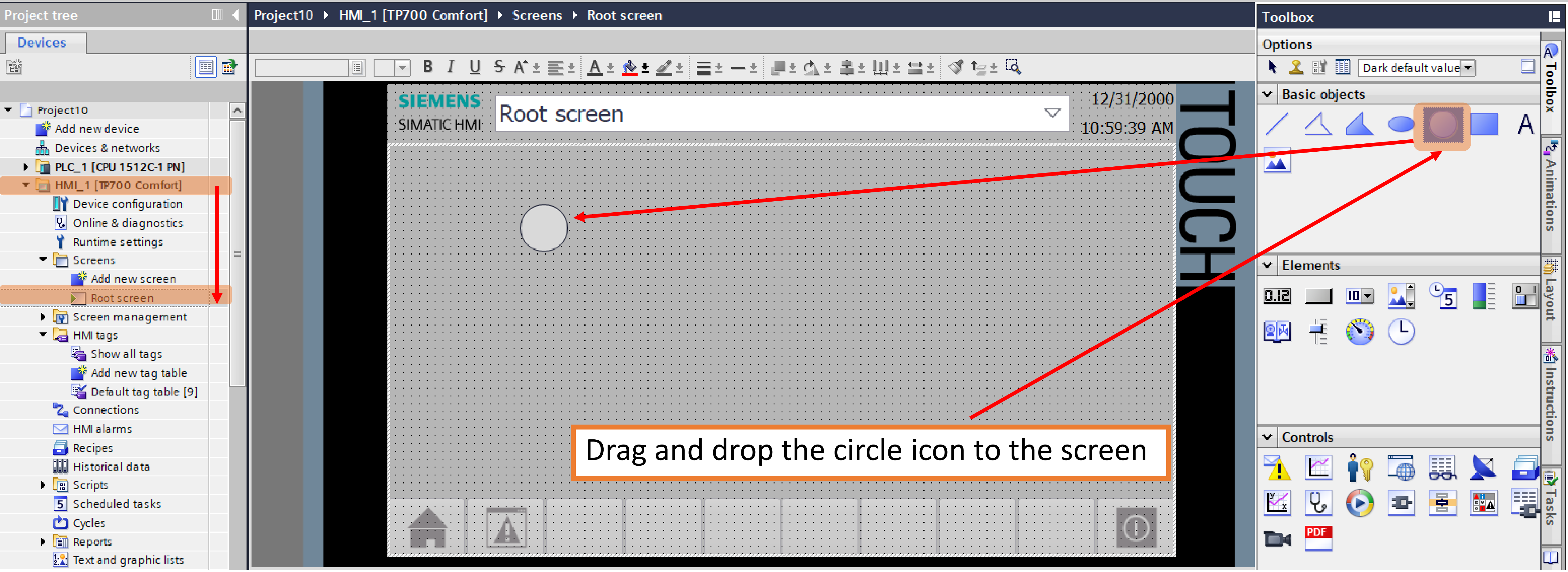
However, the router information is to be manually set


The first thing to do is to manage HMI's tag table
When the <Add new> is clicked, the next is to browse and link to a PLC's tag
Project10 ▸ HMI_1 [TP700 Comfort] ▸ HMI tags ▸ Default tag table [2]
Project tree
Devices
HMI_1 [TP700 Comfort]
Device configuration
Online & diagnostics
Runtime settings
Screens
Screen management
HMI tags
Show all tags
Add new tag table
Default tag table [2]
Default tag table
Name
Data type
Connection
PLC name
PLC tag
Address
Access mode
Acquisition cycle
Logged
Source comment
Tag_ScreenNumber
UInt
<Internal tag>
<Add new>
Discrete alarms
Analog alarms
Logging tags
Project10
Add new device
Devices & networks
PLC_1 [CPU 1512C-1 PN]
Add new screen
Default tag table [3]
Connections
HMI alarms
Software units
Program blocks
Technology objects
PLC tags
Default tag table...
Local modules
None
B Bool %I10.1
E Bool %I10.2
Mix Bool %Q4.2
P Bool %I10.0
V1 Bool %Q4.0
V2 Bool %Q4.1
VE Bool %Q4.3
Show all

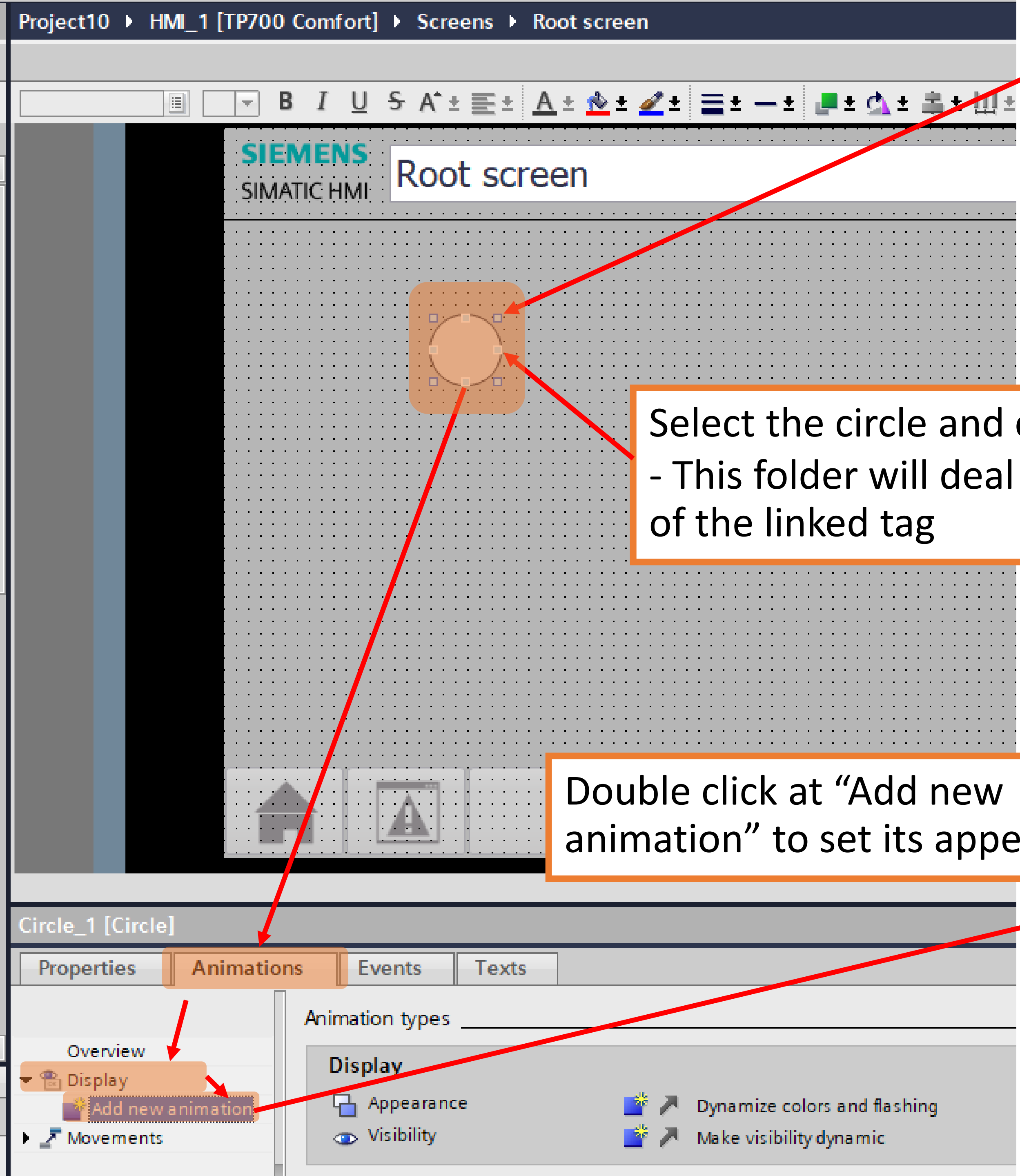


HMI_1 [TP700 Comfort]
Device configuration
Online & diagnostics
Runtime settings
Screens
Screen management
HMI tags
Show all tags
Add new tag table
Default tag table [3]
Default tag table
Name | Data type | Connection | PLC name | PLC tag | Address | Access mode | Acquisition cycle
Tag_ScreenNumber | UInt | <Internal tag> | | <Undefin... | | | 1 s
Level 1 | Bool | HMI_Connection_1 | PLC_1 | P | | <symbolic access> | 1 s
<Add new>
As a PLC tag is linked, it's time to manually set its HMI's tag name
Default tag table
Name | Data type | Connection | PLC name | PLC tag | Address | Access mode | Acquisition cycle | Logged | Source comment | Comment
Tag_ScreenNumber | UInt | <Internal tag> | | <Undefin... | | | 1 s
Level 1 | Bool | HMI_Connection_1 | PLC_1 | P | | <symbolic access> | 1 s
<Add new>
Set the reading cycle to 500ms
HMI_1 [TP700 Comfort]
Cycles
Name | Cycle time | Cyc...
None
100 ms | 100 | mil...
500 ms | 500 | mil...
1 s | 1 | sec...
2 s | 2 | sec...
5 s | 5 | sec...
10 s | 10 | sec...
1 min | 1 | mi...
Add new
PLC's 'P' sensor is tagged in HMI as 'Level 1'
Discrete alarms
Analog alarms
Logging tags
Default tag table
Name | Data type | Connection | PLC name | PLC tag | Address | Access mode | Acquisition cycle
Tag_ScreenNumber | UInt | <Internal tag> | | <Undefin... | | | 1 s
Level 1 | Bool | HMI_Connection_1 | PLC_1 | P | | <symbolic access> | 500 ms
Level 2 | Bool | HMI_Connection_1 | PLC_1 | B | | <symbolic access> | 500 ms
Draining | Bool | HMI_Connection_1 | PLC_1 | E | | <symbolic access> | 500 ms
Valve 1 | Bool | HMI_Connection_1 | PLC_1 | V1 | | <symbolic access> | 500 ms
Valve 2 | Bool | HMI_Connection_1 | PLC_1 | V2 | | <symbolic access> | 500 ms
Mixer | Bool | HMI_Connection_1 | PLC_1 | Mix | | <symbolic access> | 500 ms
Valve E | Bool | HMI_Connection_1 | PLC_1 | VE | | <symbolic access> | 500 ms
Now, link all the PLC's tag to HMI's tag as shown in this table

Now, it's time to work with the screen. As the current sample is not complicate, the root screen is only used.



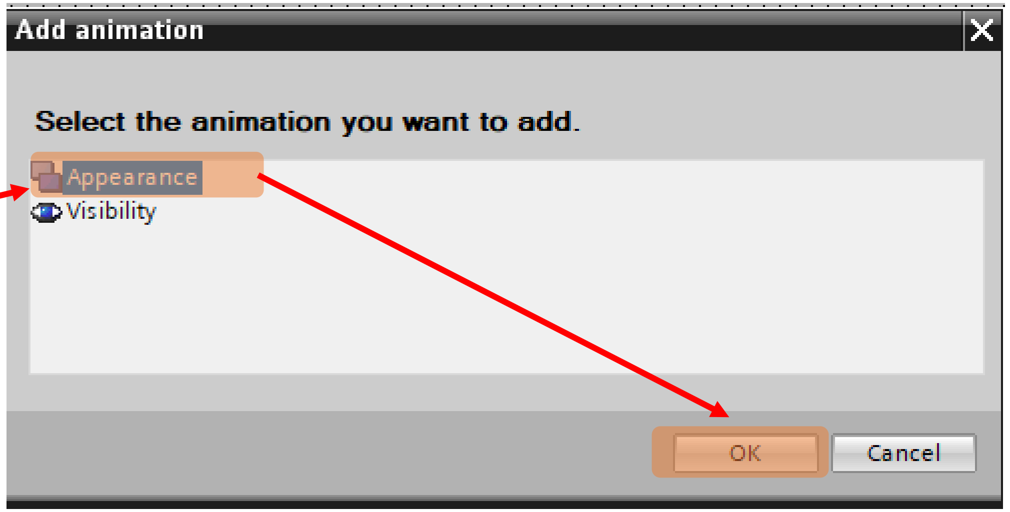
This circle is to be linked to a HMI tag to display its status

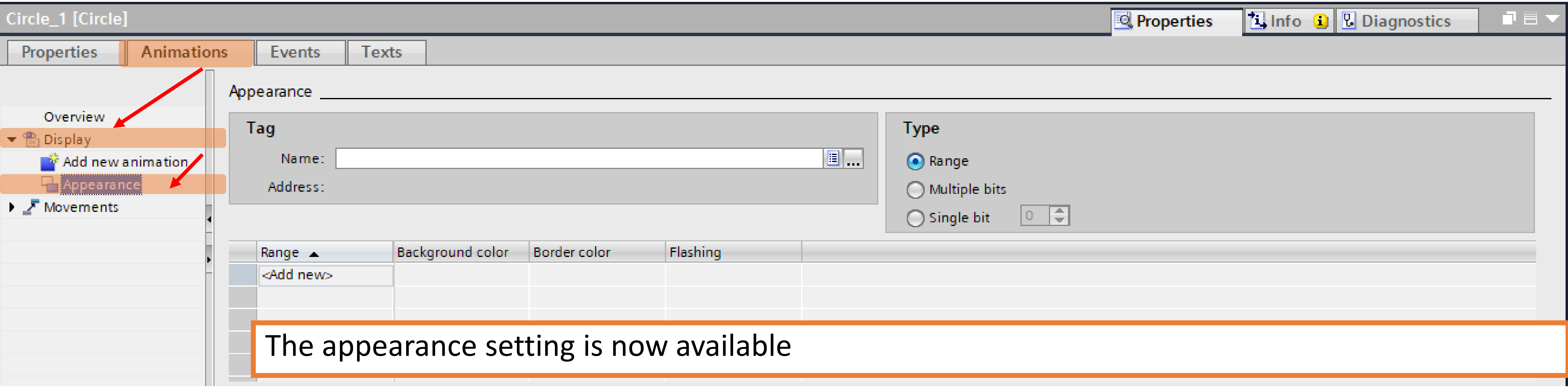


Select the circle and check its 'Animations' folder.

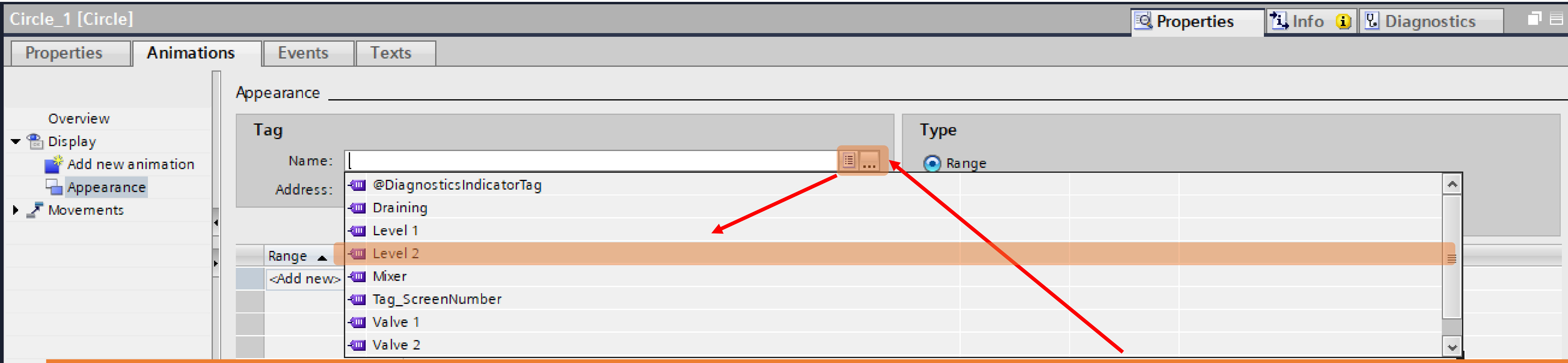
- This folder will deal with its change of appearance according to the changing of the linked tag

Double click at "Add new animation" to set its appearance

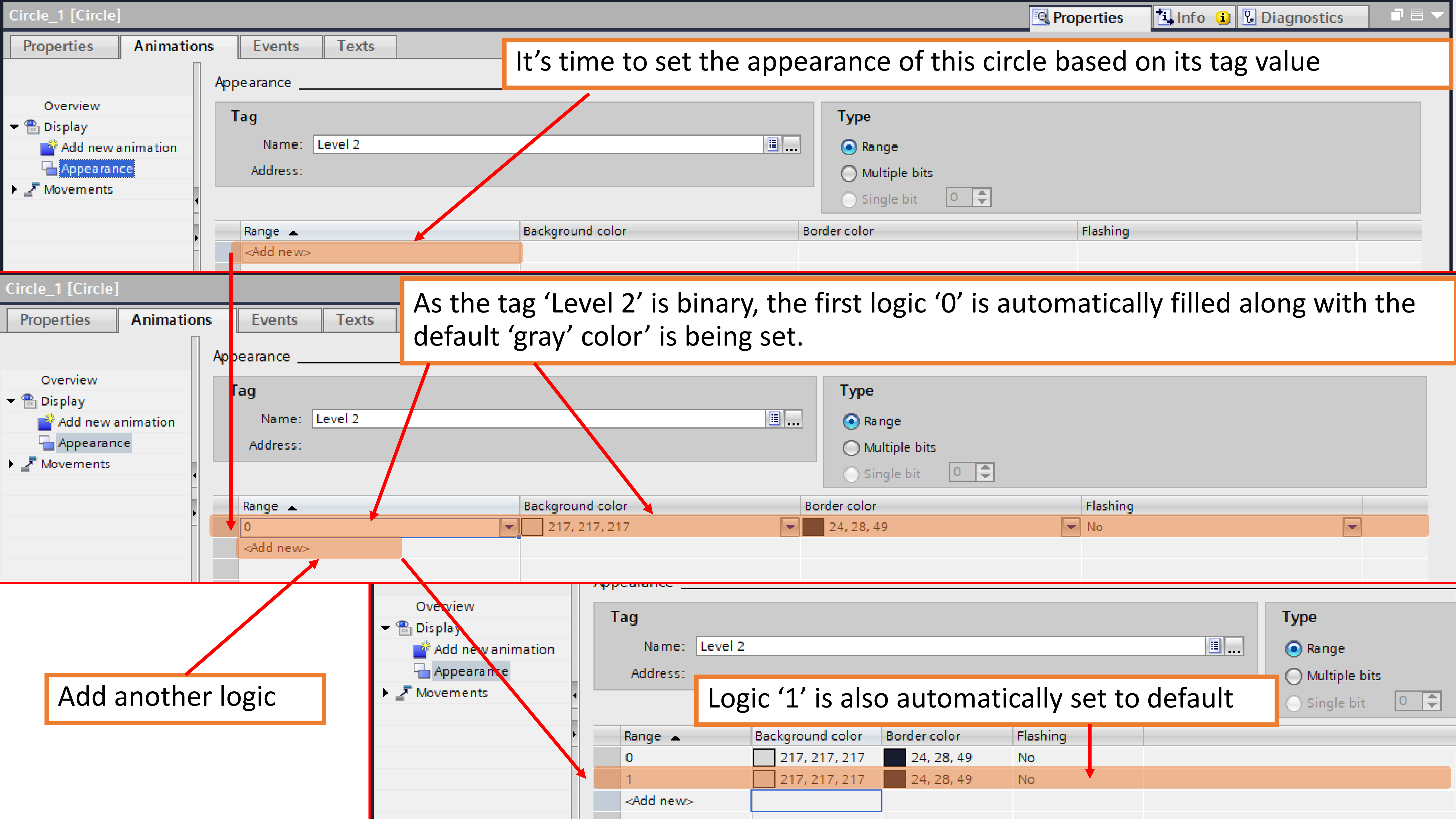




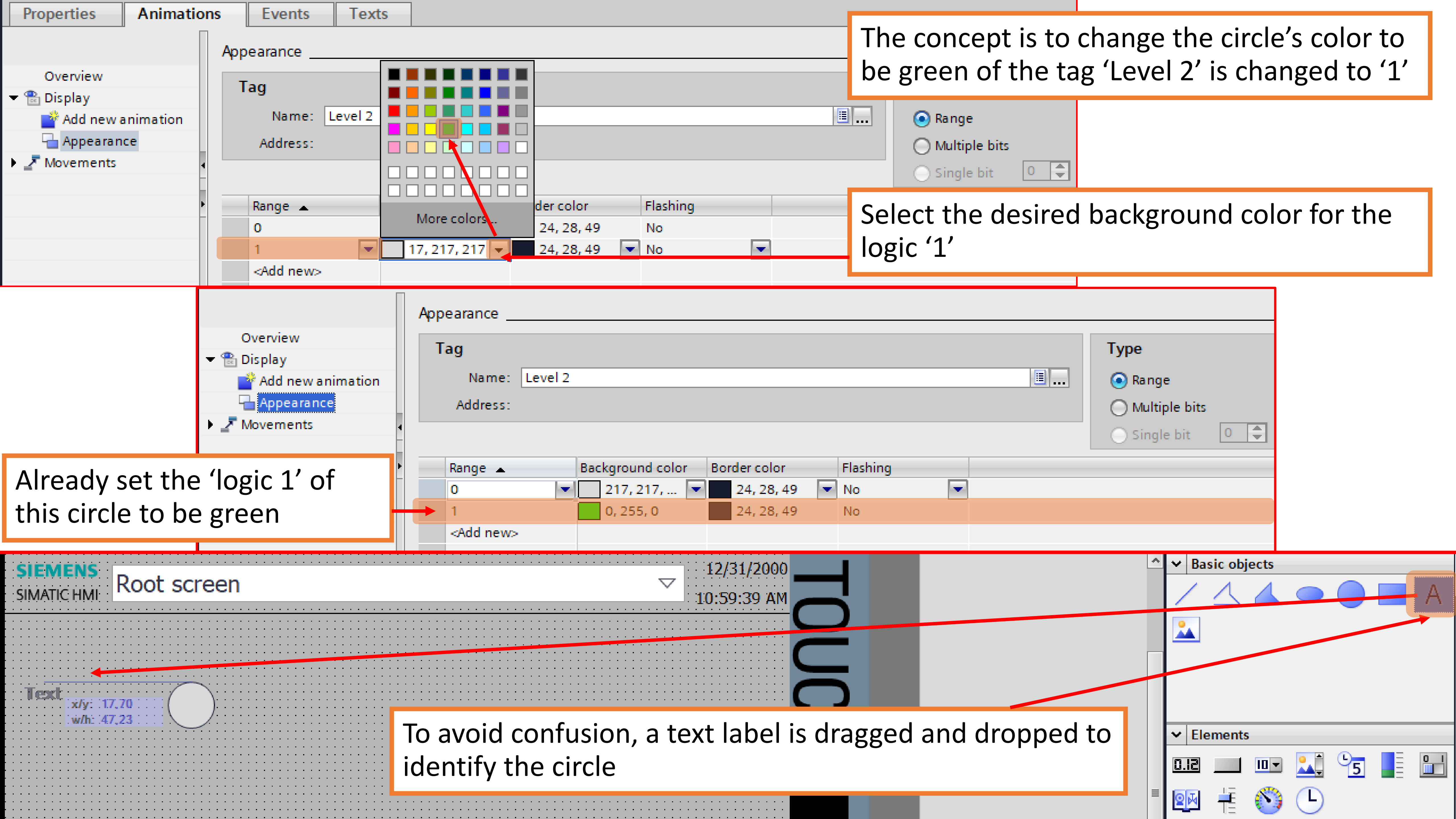
The appearance setting is now available



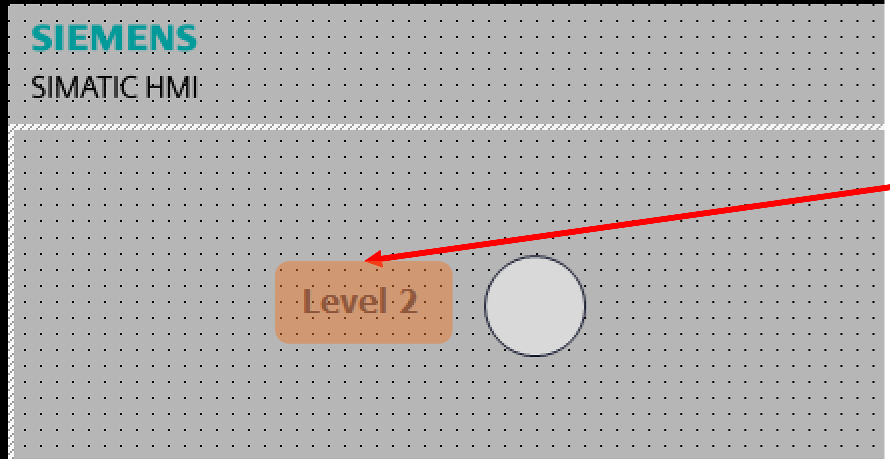
Browse and select the desired HMI's tag. In this case, select the 'Level 2' tag.



Circle_1 [Circle]
Properties
Info
Diagnostics
Properties
Animations
Events
Texts
It's time to set the appearance of this circle based on its tag value
Appearance
Overview
Display
Add new animation
Appearance
Movements
Tag
Name: Level 2
Address:
Type
Range
Multiple bits
Single bit
0
Range
Background color
Border color
Flashing
<Add new>
Circle_1 [Circle]
As the tag 'Level 2' is binary, the first logic '0' is automatically filled along with the default 'gray' color' is being set.
0
217, 217, 217
24, 28, 49
No
<Add new>
Add another logic
Logic '1' is also automatically set to default
1
217, 217, 217
24, 28, 49
No



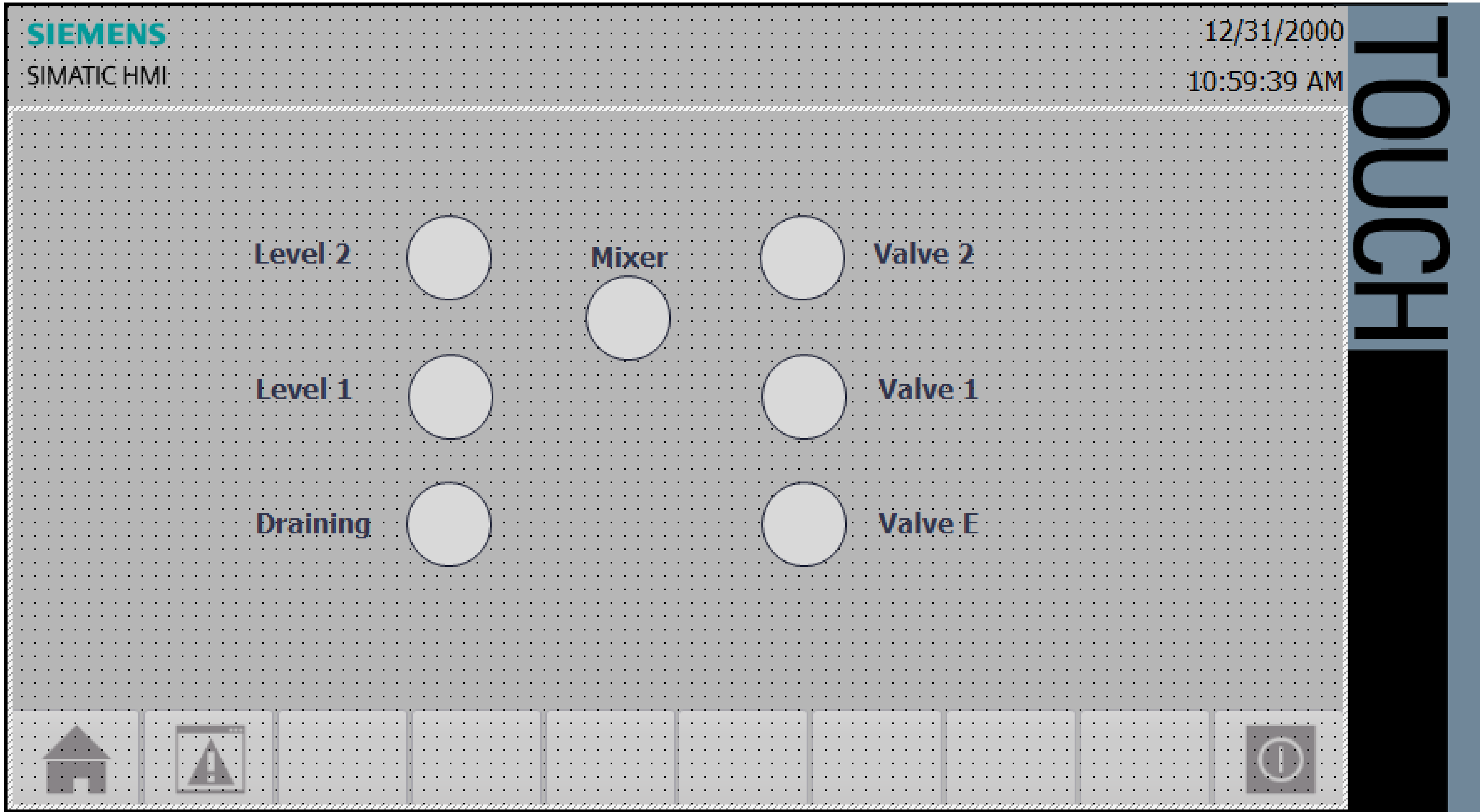
The concept is to change the circle's color to be green of the tag 'Level 2' is changed to '1'
Select the desired background color for the logic '1'
Already set the 'logic 1' of this circle to be green
To avoid confusion, a text label is dragged and dropped to identify the circle
Properties
Animations
Events
Texts
Appearance
Overview
Display
Add new animation
Movements
Tag
Name:
Level 2
Address:
Type
Range
Multiple bits
Single bit
More colors...
Background color
Border color
Flashing
0
1
217, 217, ...
17, 217, 217
24, 28, 49
0, 255, 0
No
<Add new>
SIEMENS
SIMATIC HMI
Root screen
12/31/2000
10:59:39 AM
Text
x/y: 17,70
w/h: 47,23
Basic objects
Elements



Manually edit the text to ‘Label 2’ which is matching its tag

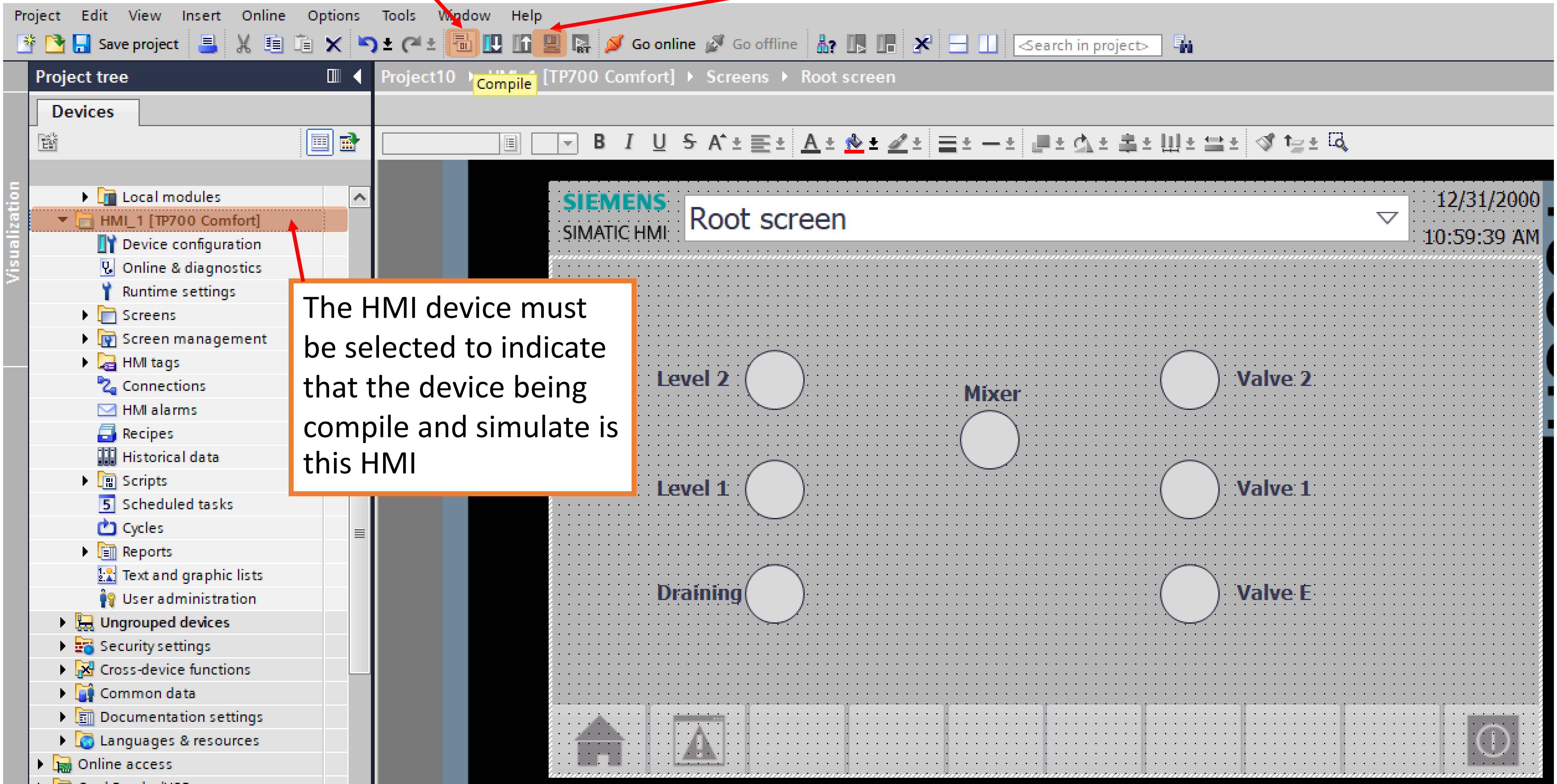
Now, repletely add circles and labels to show all available HMI’s tags

Don’t forget to link each added circle to its correlated tag



As everything is ready, it's time to compile the HMI

If there is no error in compilation, click this button to simulate the HMI

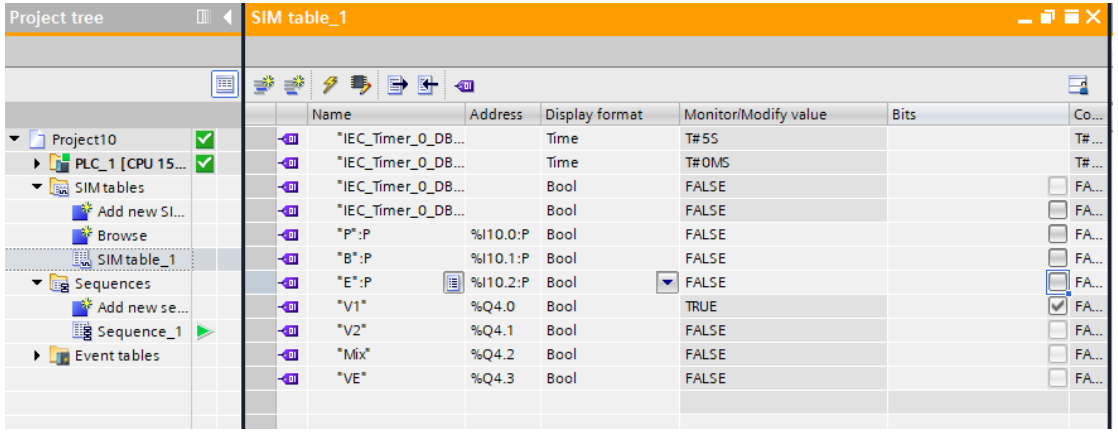


The HMI device must be selected to indicate that the device being compile and simulate is this HMI

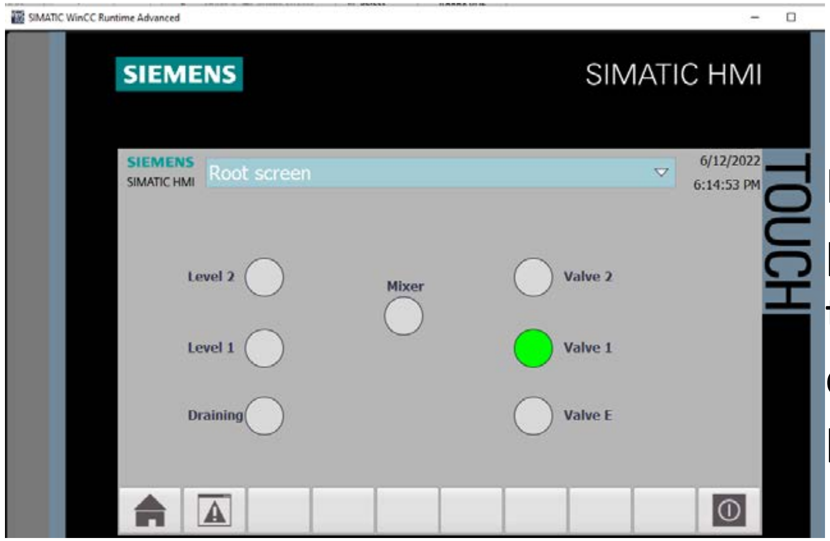
Simulating an HMI

- HMI simulation is an independent session that sync with the targeted session of PLC simulation
- Actually, it doesn't matter which one is running first
  - Personally, running a PLC simulation first is recommended
- In this case, run the PLC simulation first, and make sure the operation is correct
- Then, run the HMI simulator, they will automatically synchronized each others

PLC Simulator



Automatically synchronized as set in the TIA Portal



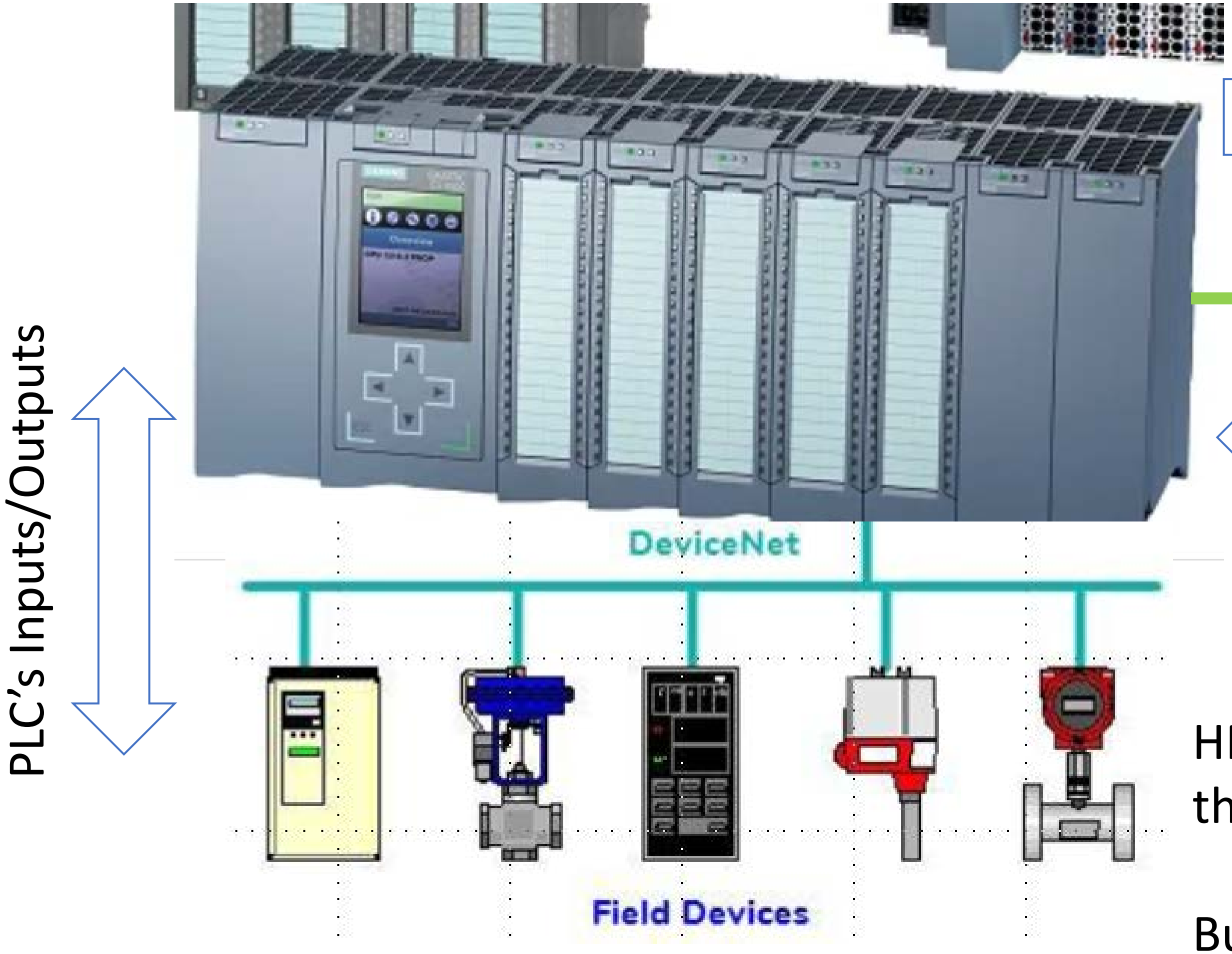
HMI Simulator programmed to display the on going process

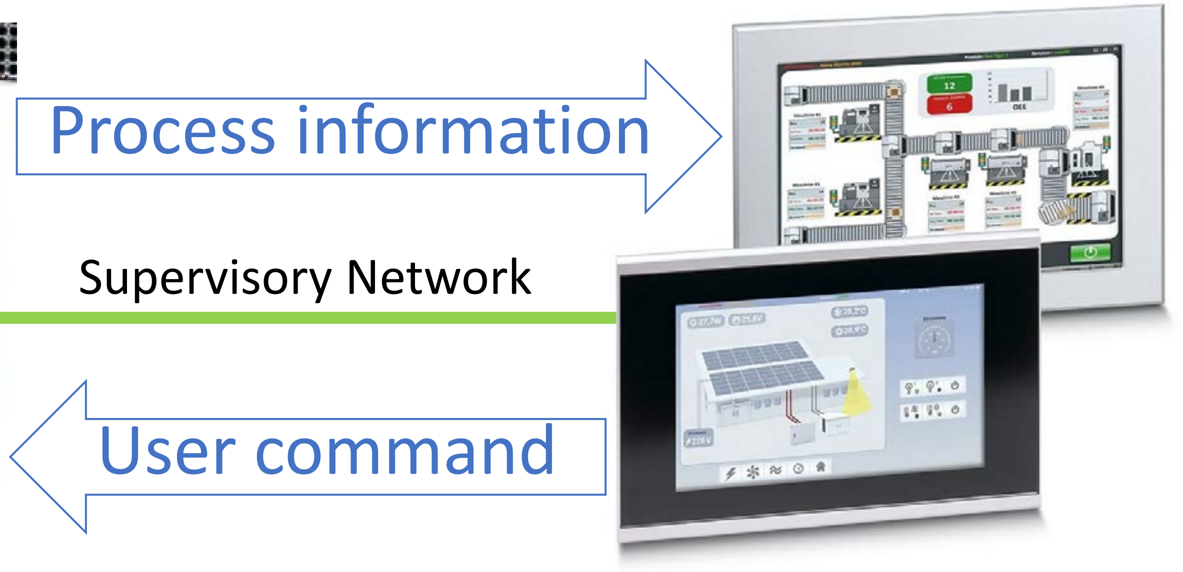
Exercise

- Modify the problem to contain another mixer, and monitoring its status in the HMI

Lab 02: Process control via HMI

Process control via HMI



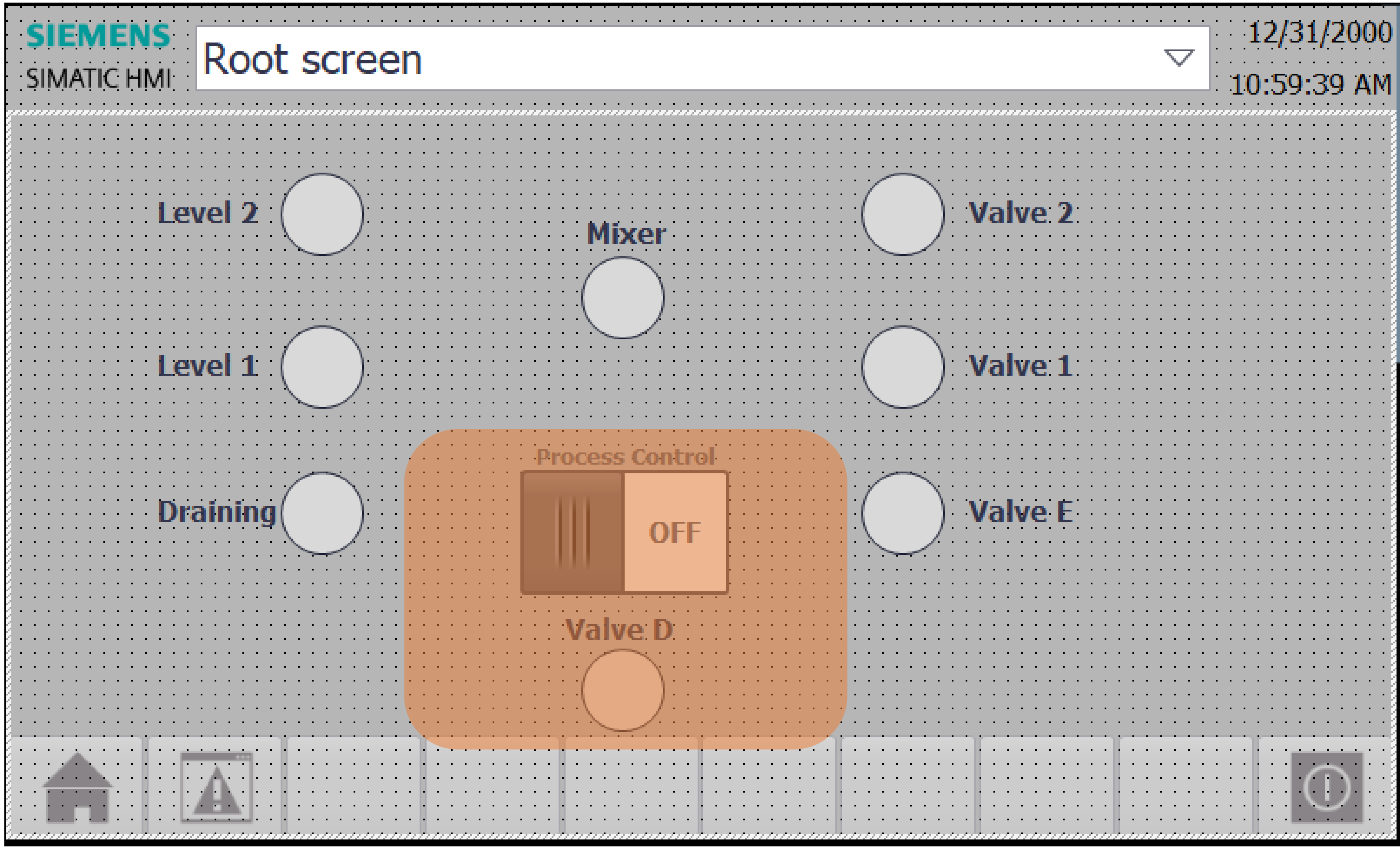


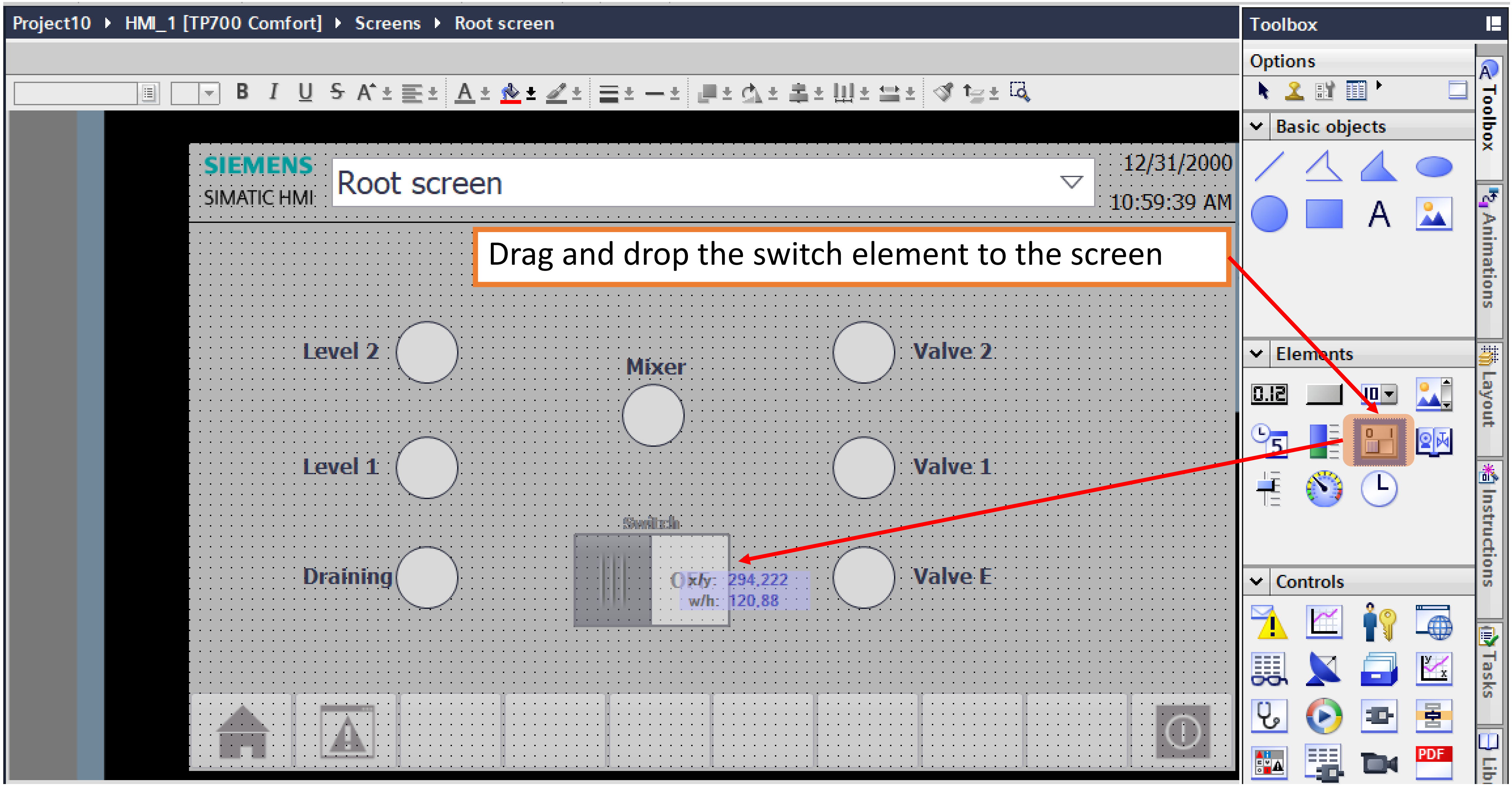
HMI are never meant to interfere with the PLC inputs as they must represent the real situation

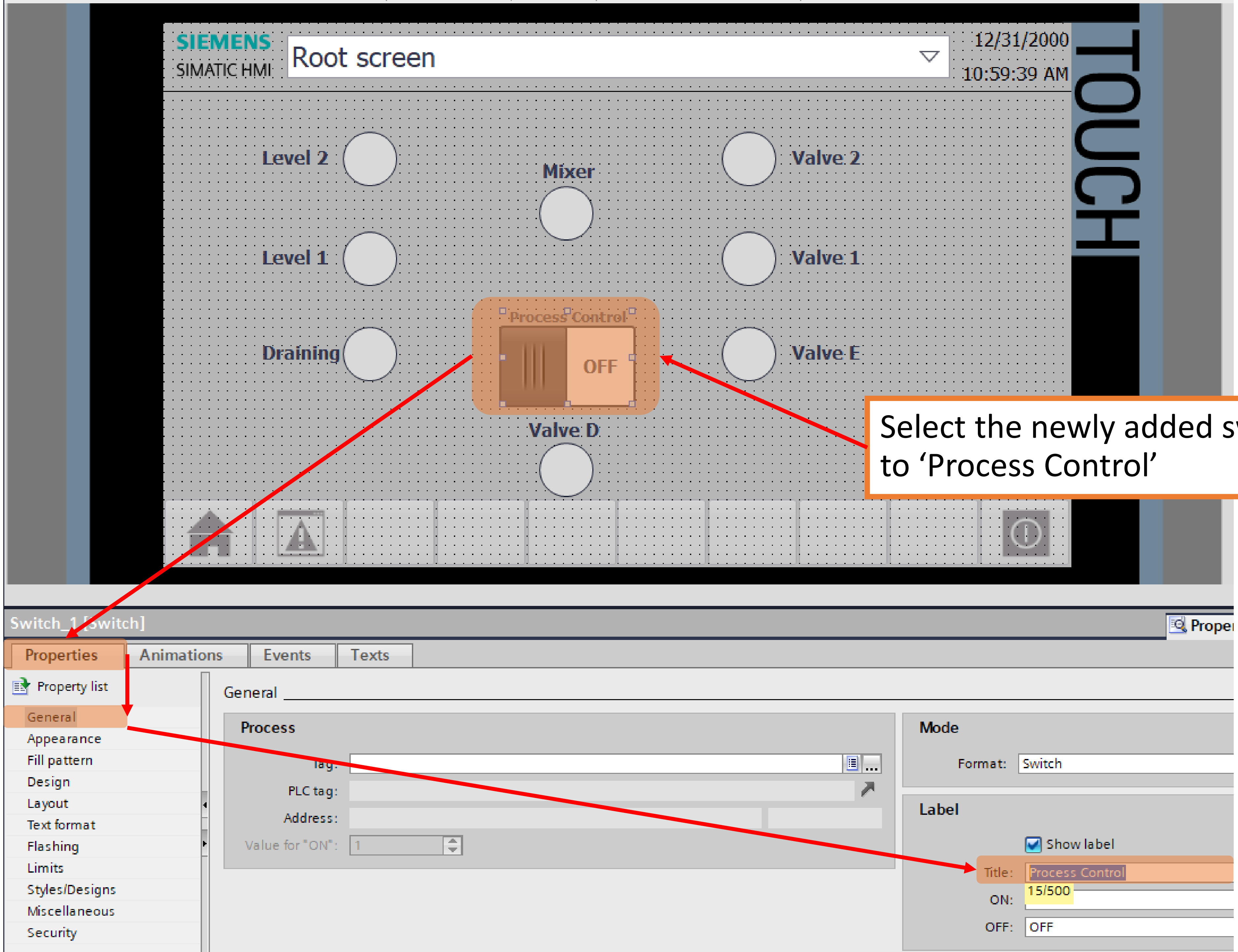
But, the output of the PLC can be manipulated by the HMI in order to override or suspense the normal process if there are necessary factors

Start/Stop the process from HMI

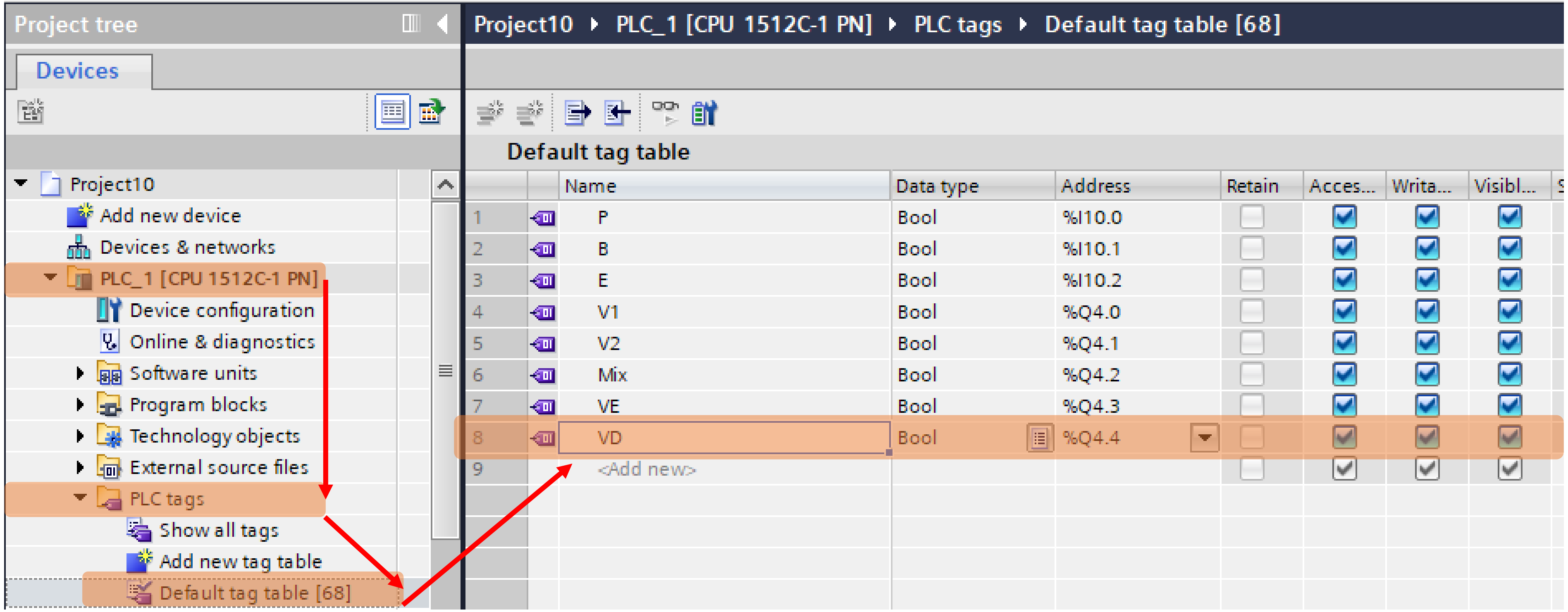
- Add a control switch to start and stop the whole process
- V1, V2, Mixer and VE are off as the switch is off
- VD or ditching valve is added to ditch the remaining liquid in the tank as the process is forced to stop
  - VD is on as long as the process is forced to stop




Project10 ▸ HMI_1 [TP700 Comfort] ▸ Screens ▸ Root screen
Toolbox
Options
Basic objects
Elements
Controls
SIEMENS
SIMATIC HMI
Root screen
12/31/2000
10:59:39 AM
Drag and drop the switch element to the screen
Level 2
Mixer
Valve 2
Level 1
Valve 1
Draining
Valve E
Switch
x/y: 294,222
w/h: 120,88
Animations
Layout
Instructions
Tasks
PDF



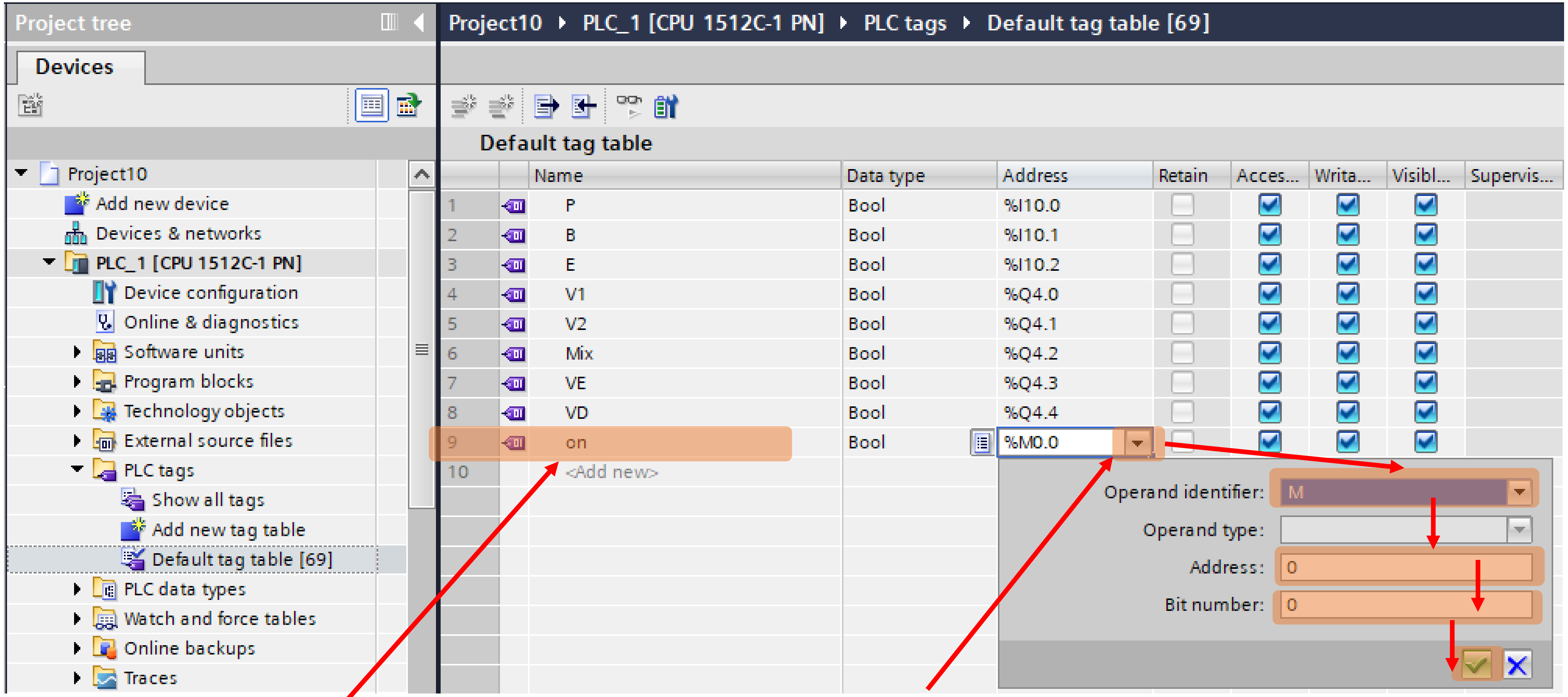
Select the newly added switch and change its title to 'Process Control'



With no HMI tags for the added components, return back to the PLC

- Add new valve 'VD' and set its address

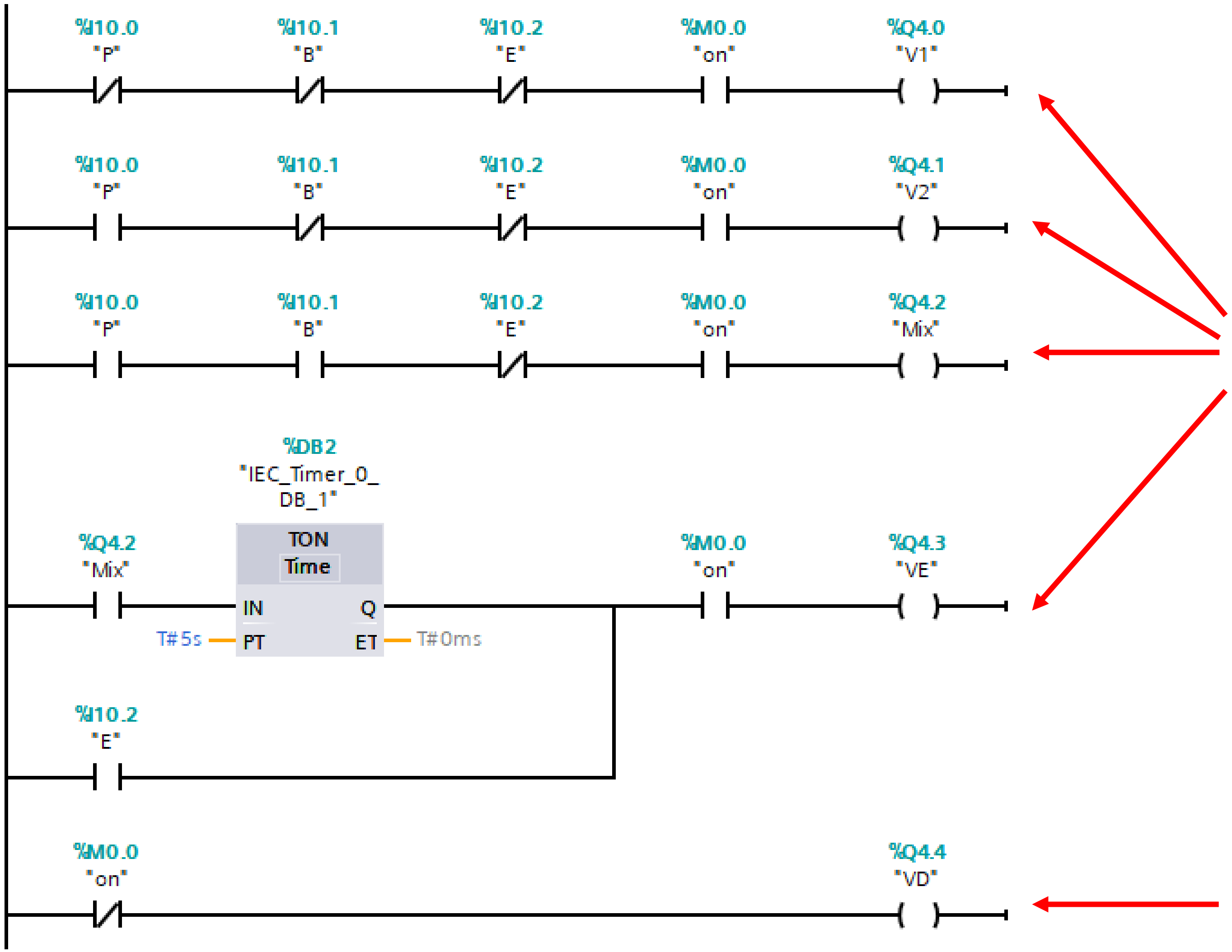
The user command from the HMI is not input and output of the PLC.
- So, its tag is not I (input) or Q (output)



A tag named 'on' is added to set the status of the process to be on or not.

- As it is to be received from the HMI, it is not input or output.
- So, just put it in PLC's memory (M)
- Its value is to be controlled by the HMI

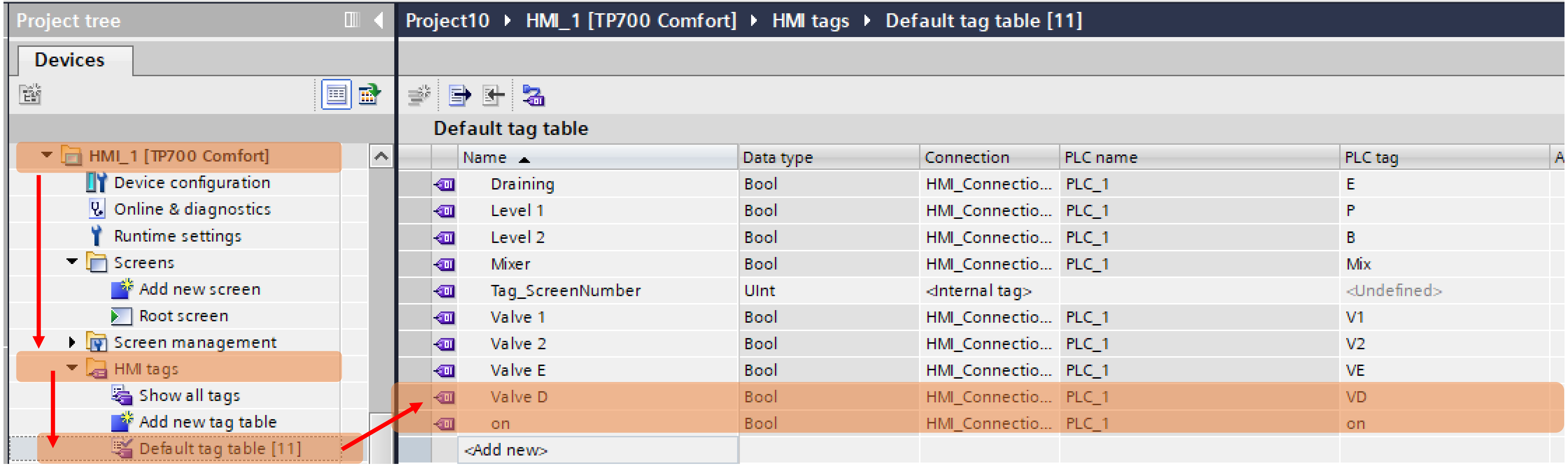
Put the control variable into the ladder



V1, V2, Mixer and VE are off as the switch is off

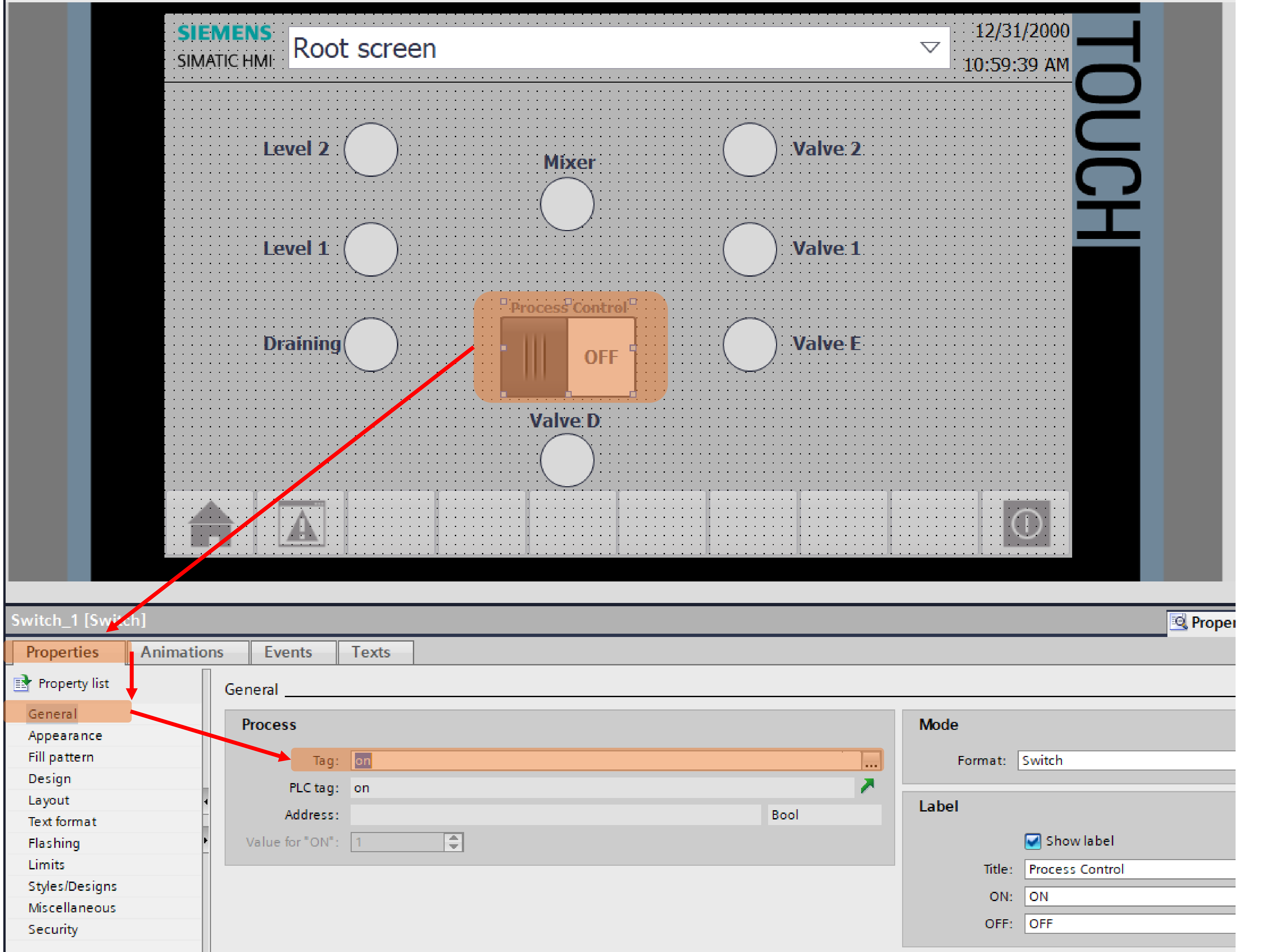
VD is on as long as the process is forced to stop

Config the HMI to control the tag



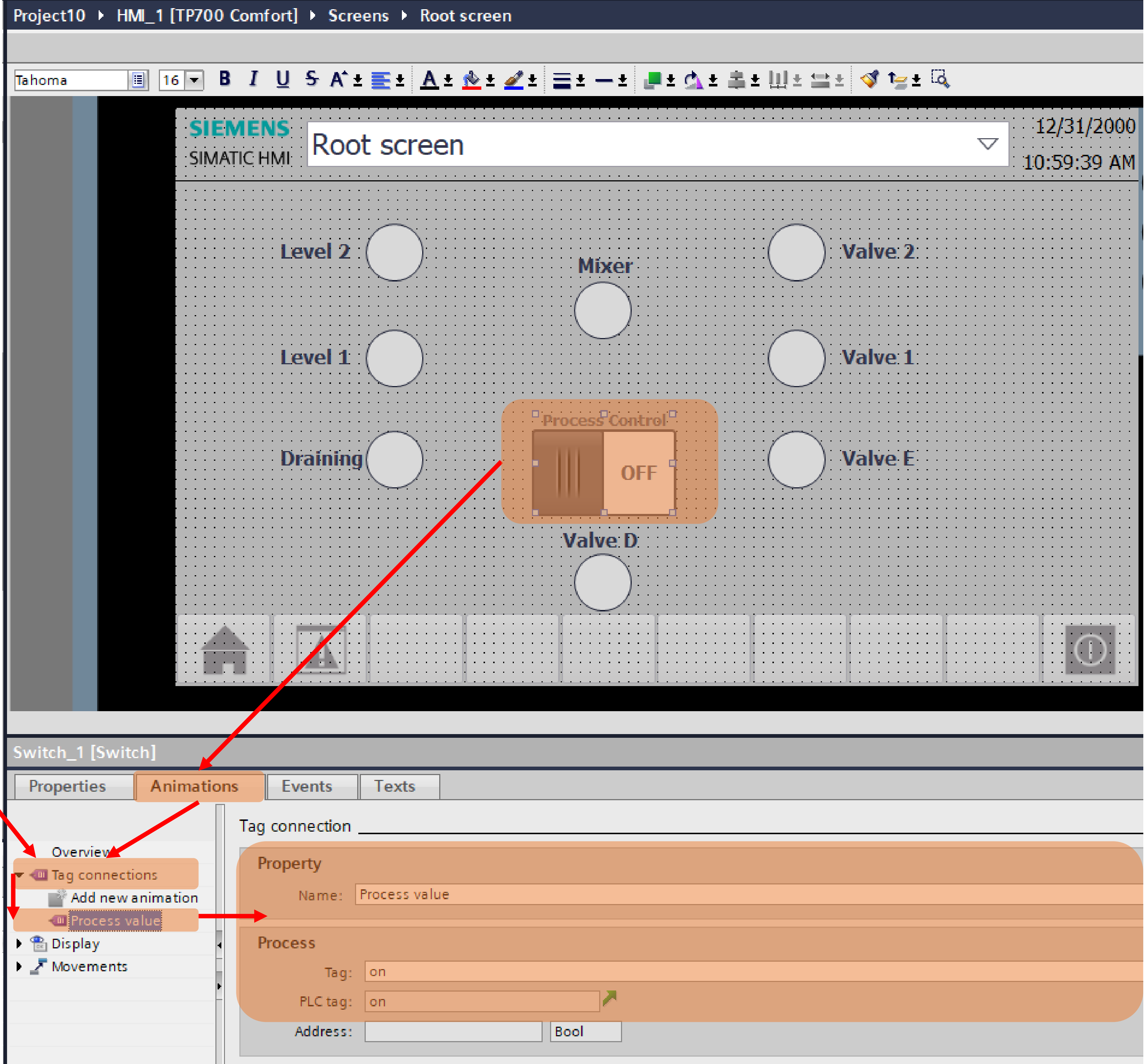
First, update the HMI tag table with two new tags from the PLC
Then, link these new tags to the screen components

Link the ‘on’ tag to the switch

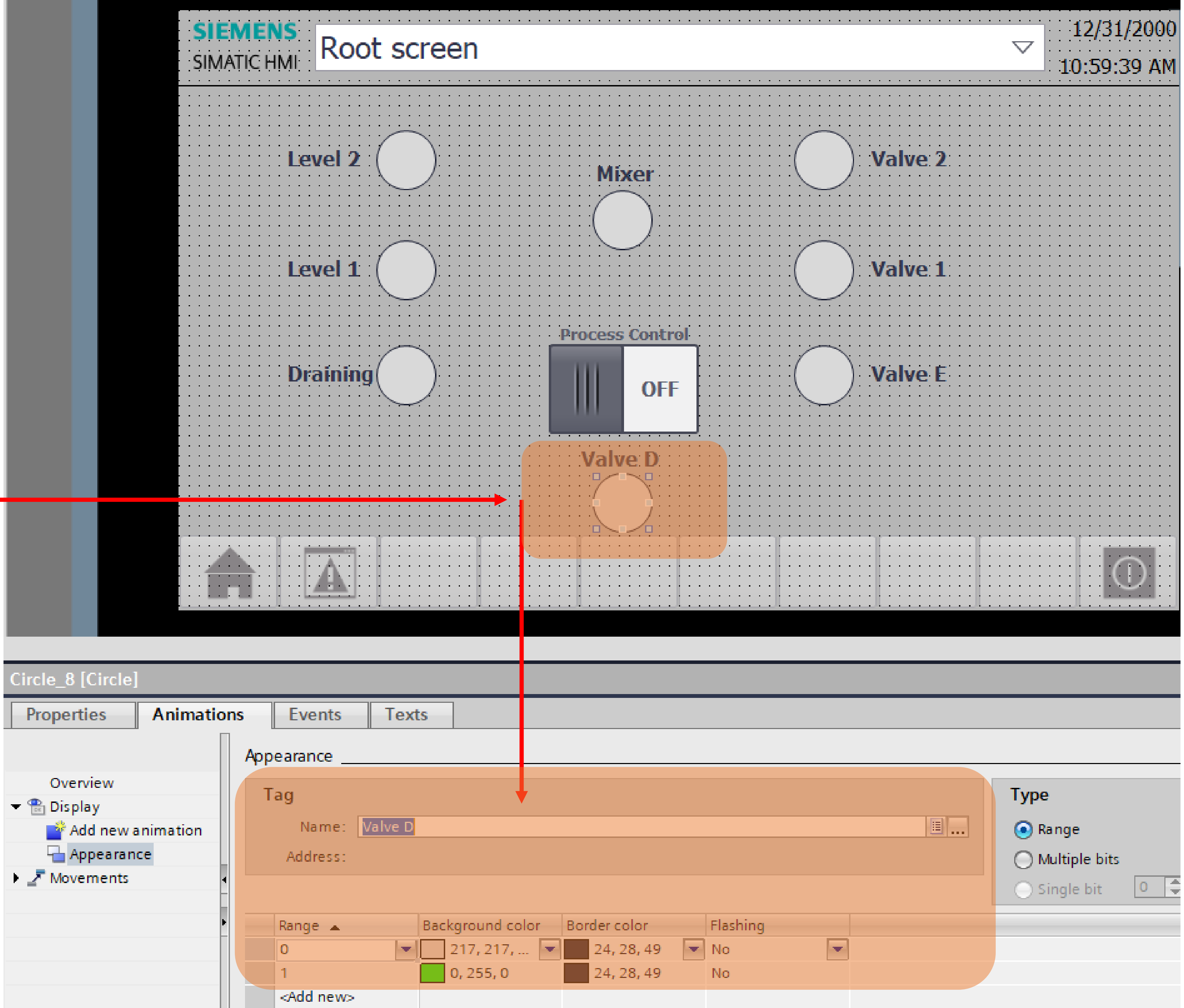


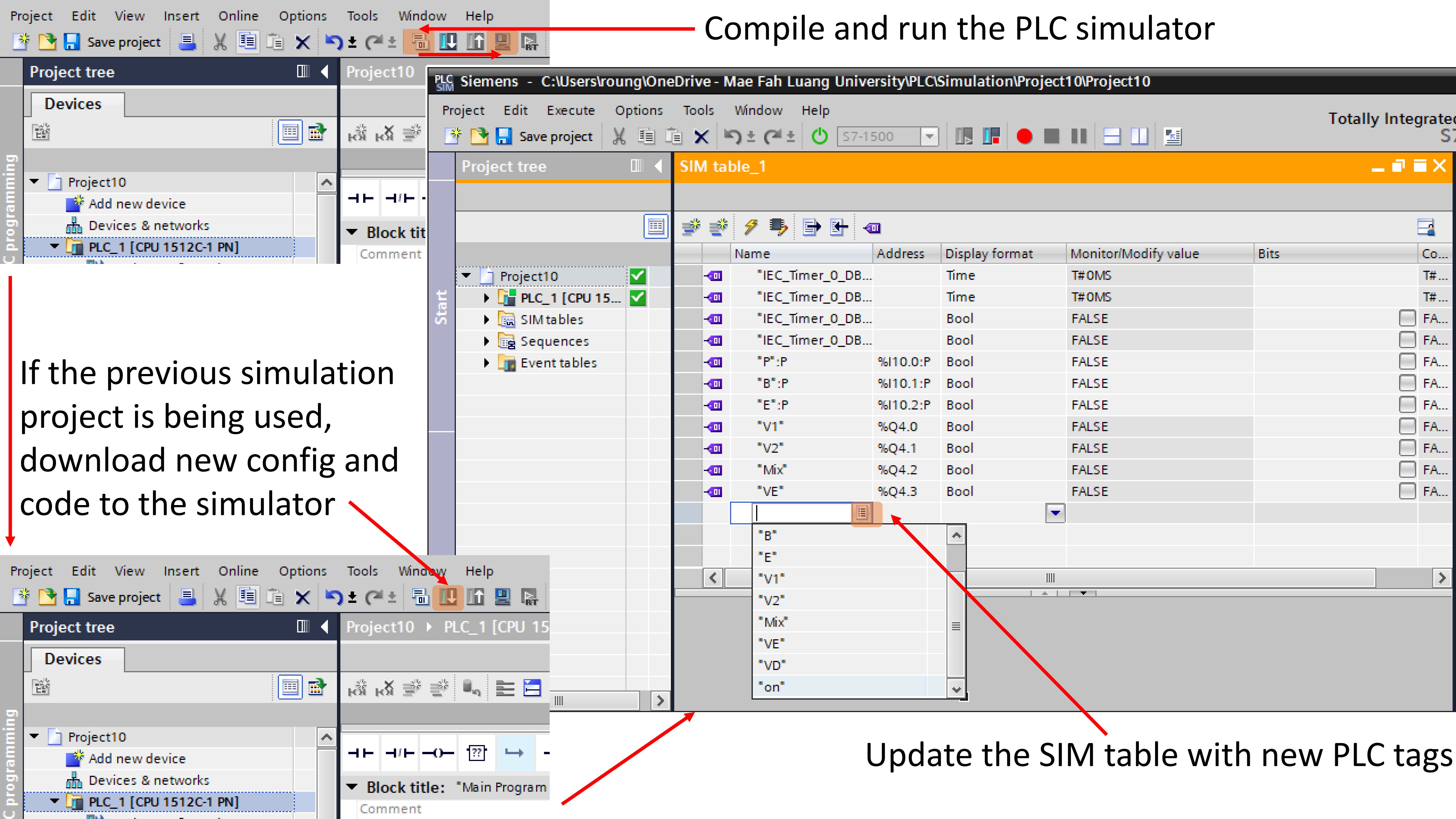
As it is a switch, the tag connections is enable by default

Config its property and tag if they're not automatically set by previous setting



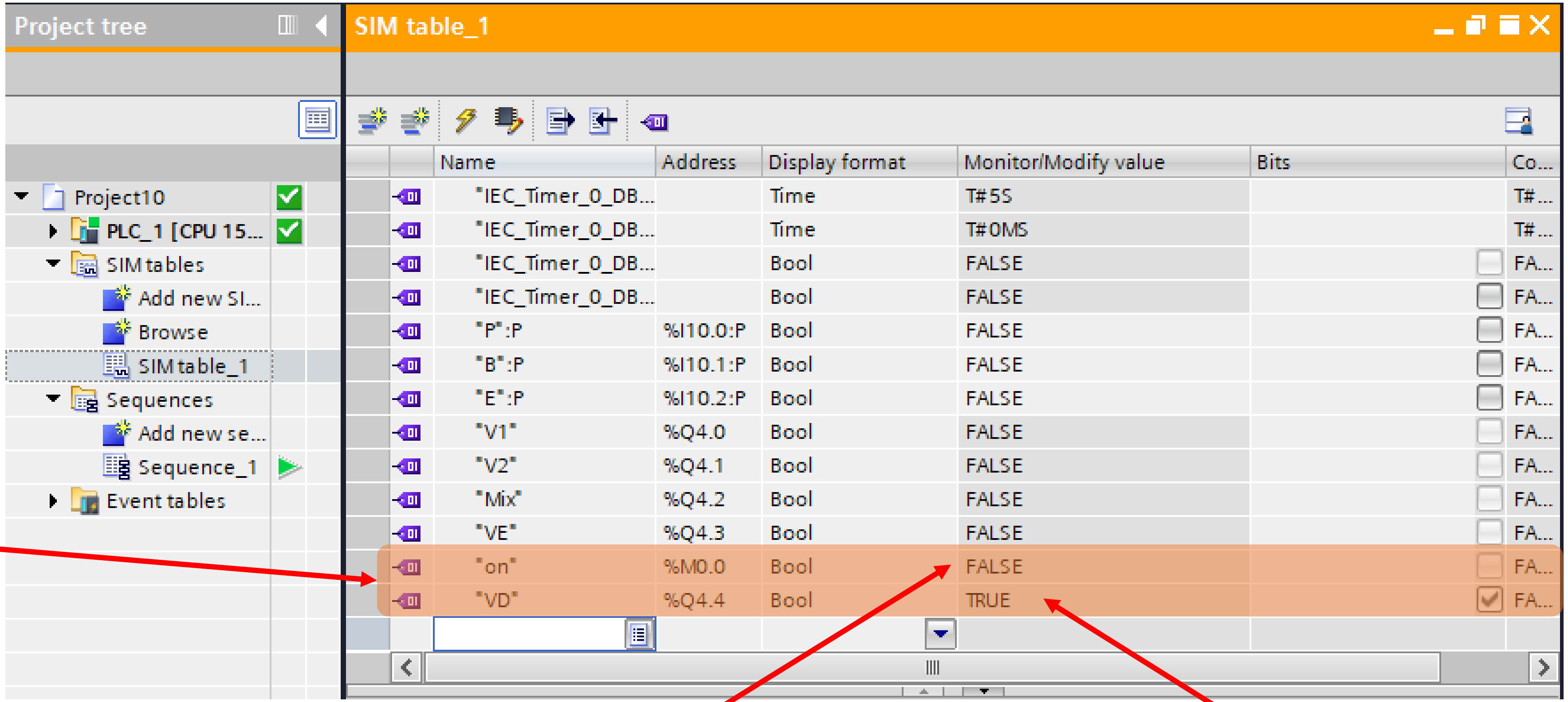
Update Valve D



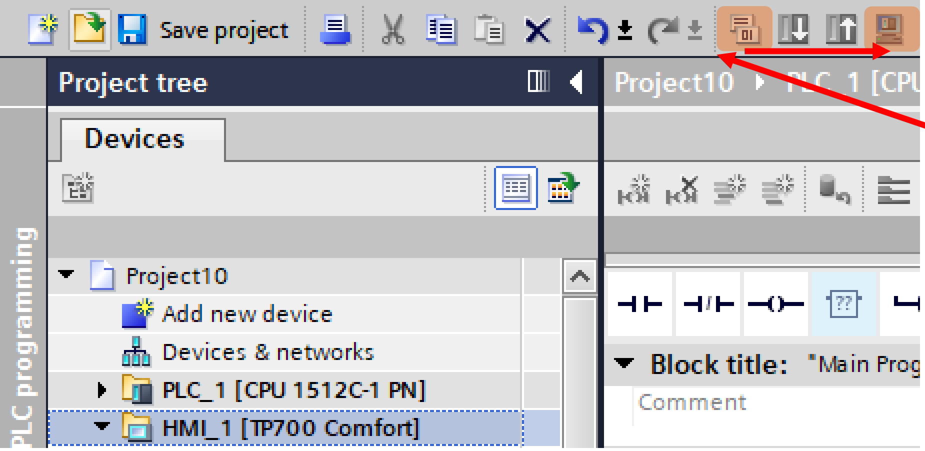


Compile and run the PLC simulator
If the previous simulation project is being used, download new config and code to the simulator
Update the SIM table with new PLC tags

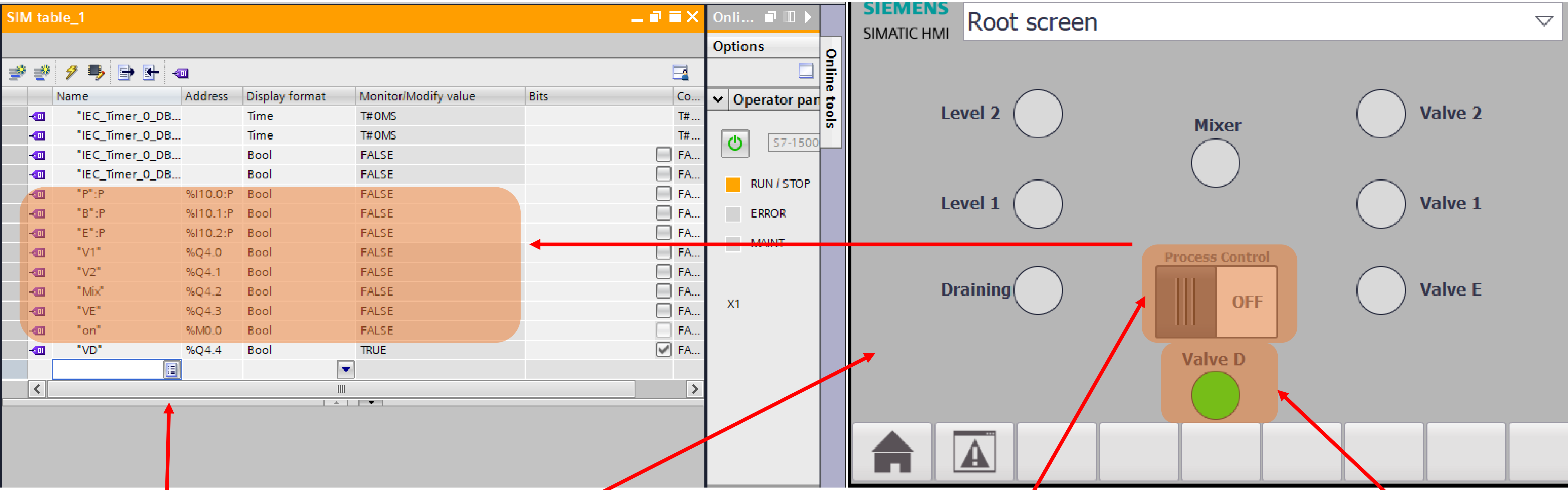
Add new tag to the SIM table: ‘on’ and ‘VD’



The tag ‘on’ is initially off, all other tags but ‘VD’ are off



Now, compile and run the HMI simulator



The SIM table of PLC and the screen of HMI are being compared

As the process is initially off, any device but ‘Valve D’ is off

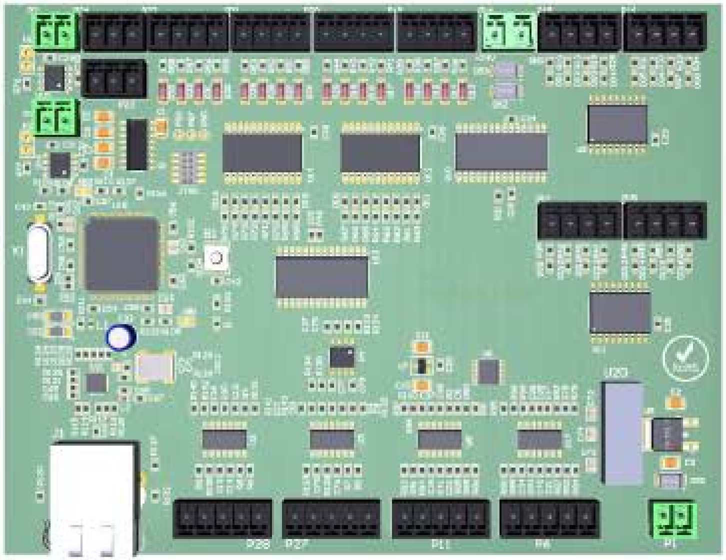
Exercise

- Try to put the “Pause” button to just pause the process without flooding the mixed liquid

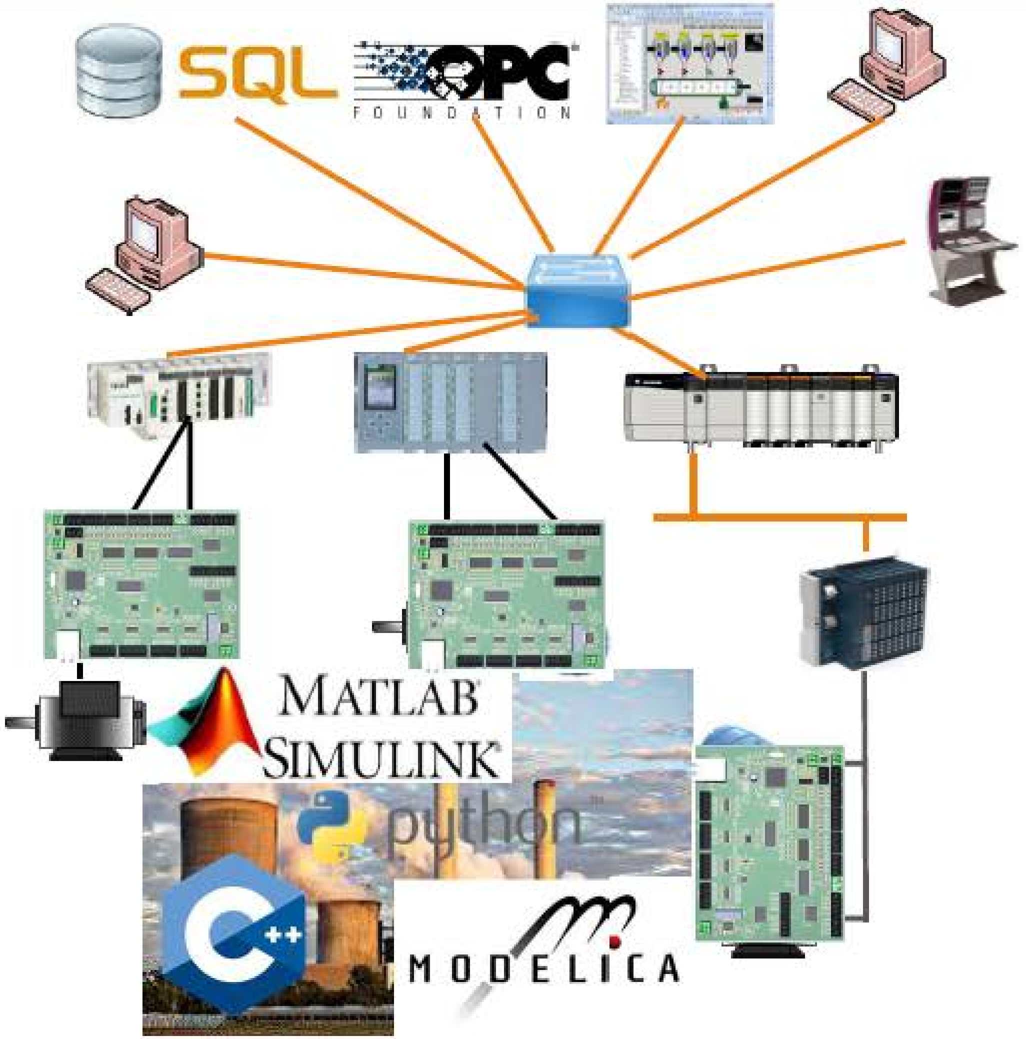
Lab 03: Emulation card

HARDWARE IN THE LOOP SYSTEM

- **Home made electronic interface card**



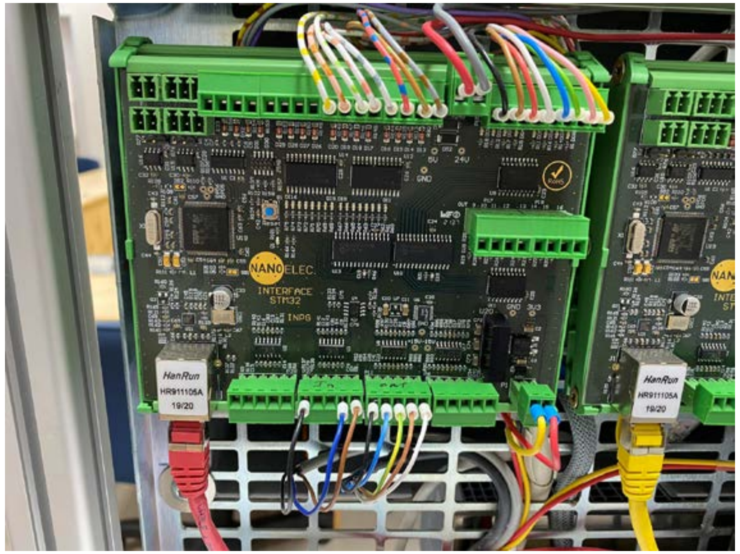
- **24 sensors and 24 actuators**
  - 16 digital inputs / 16 digital outputs
  - 8 analog inputs / 8 analog outputs
- **Less than 500€**
- **Reasonable timing performance (< 10 ms response time)**
- **Easily chain (Ethernet addressing)**



Configuration

Output → Input

Input ← Output

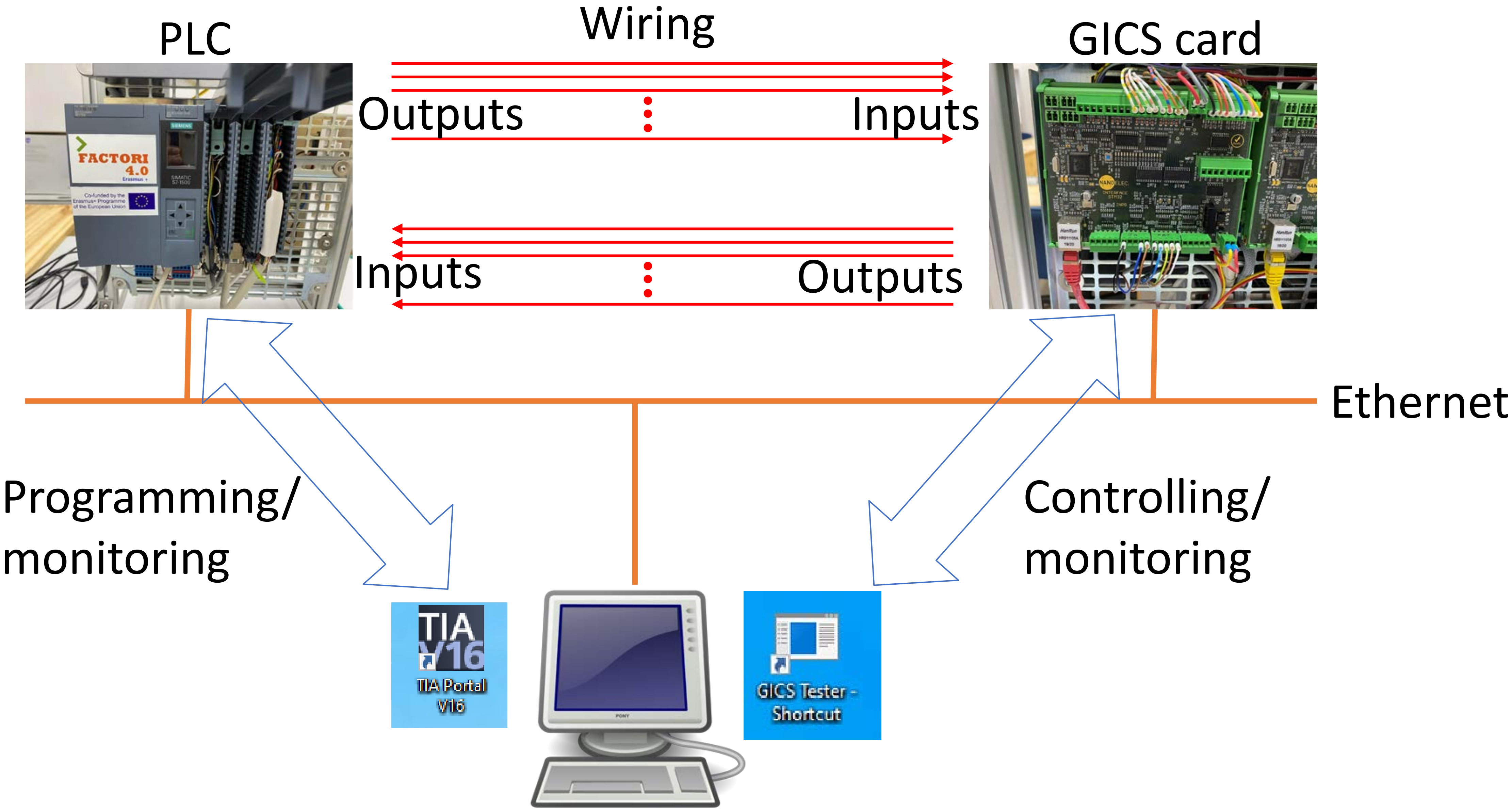


PLC

GICS card

PLC I/O are programmed in TIA Portal

GICS's I/O are manipulated in a special program called "GICS Tester".

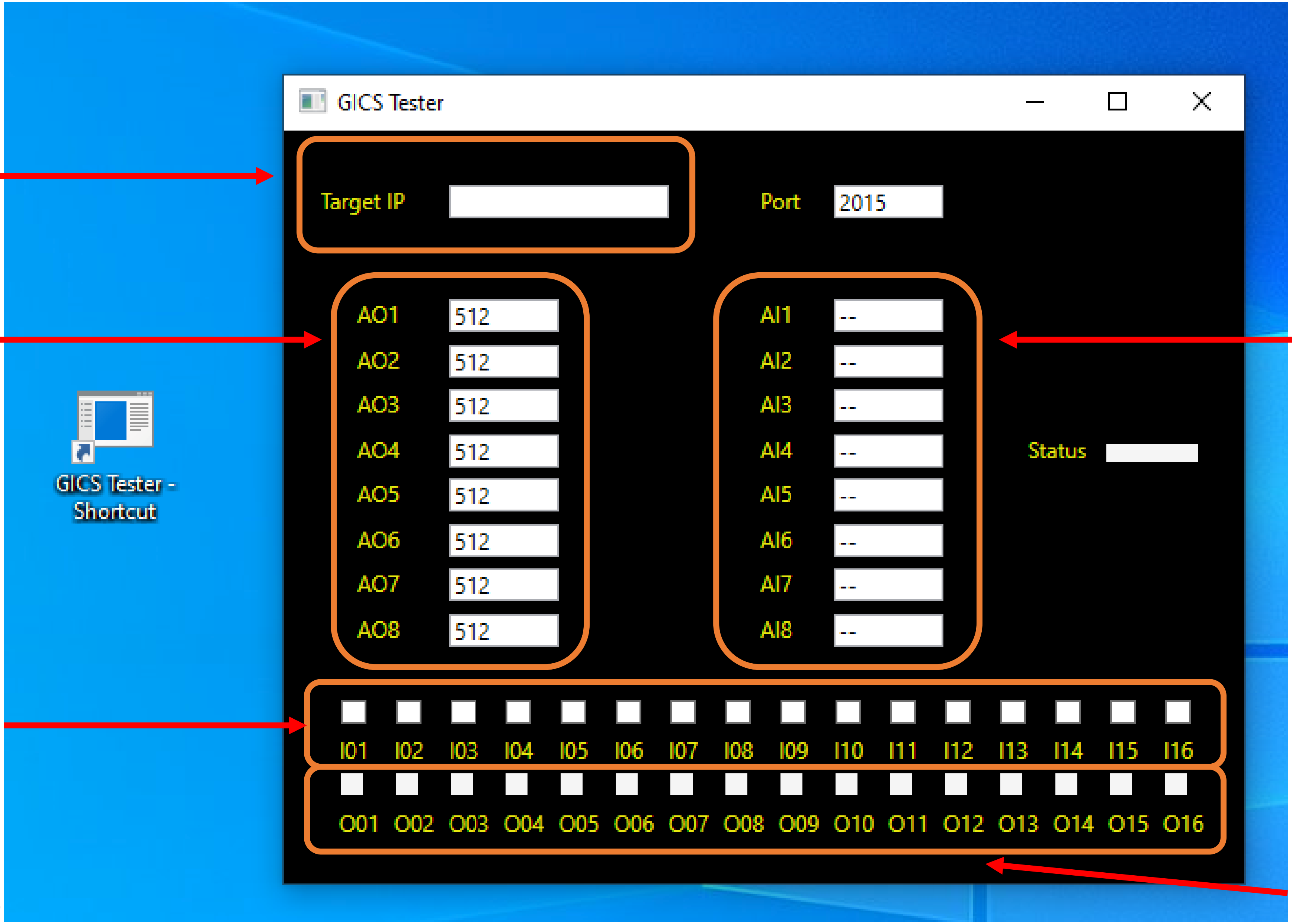


PLC
Wiring
GICS card
Outputs
Inputs
Inputs
Outputs
Ethernet
Programming/
monitoring
Controlling/
monitoring
TIA V16
TIA Portal V16
GICS Tester - Shortcut

Card's IP address

Card's analog outputs or PLC's analog inputs

Card's digital outputs or PLC's digital inputs



Card's analog inputs or PLC's analog outputs

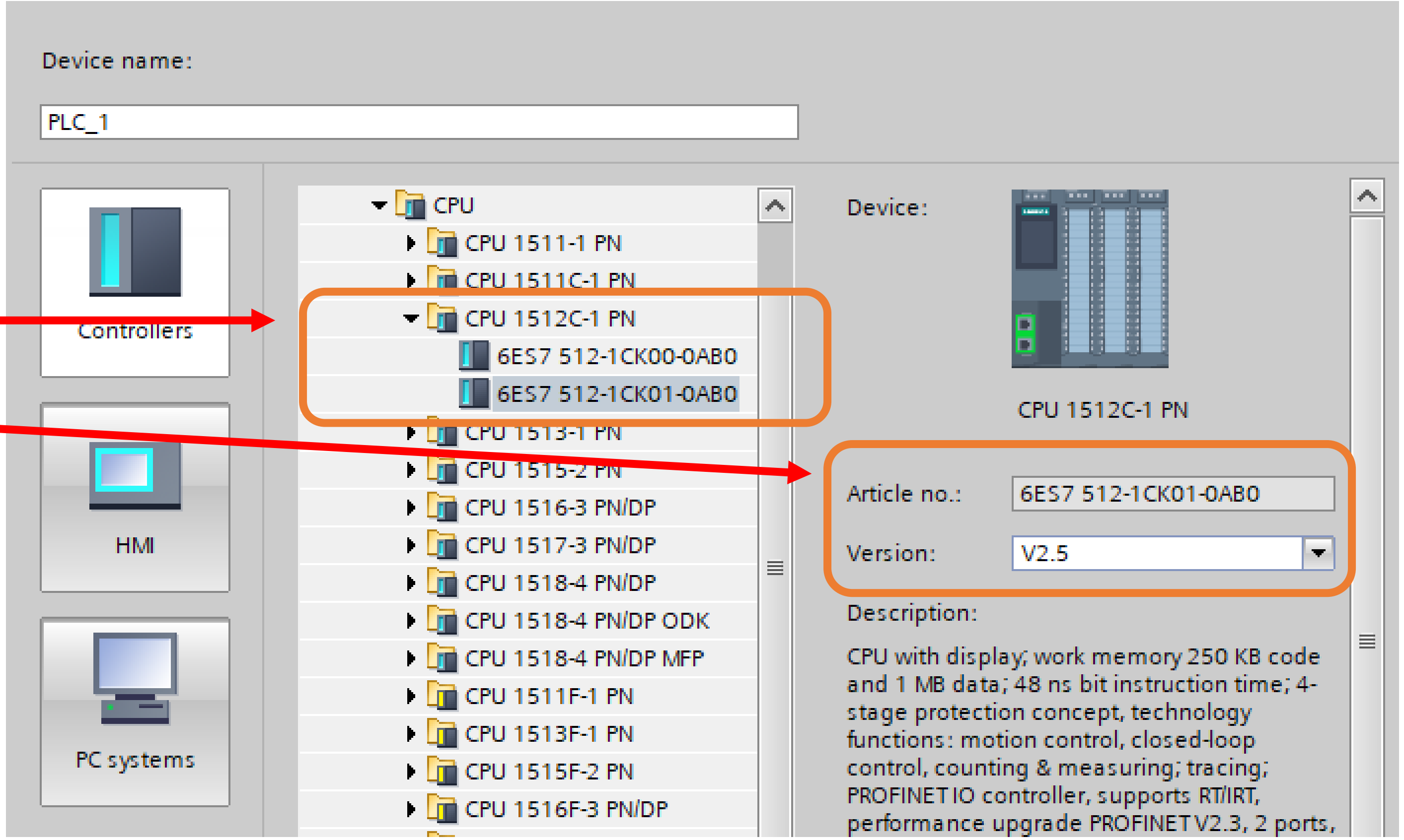
Card's digital inputs or PLC's digital outputs

Problem: Simple inputs and outputs

- Input 1 on → Output 1 on
- Input 2 on → Output 2 on
- Input 3 on → Output 3 on
- Input 4 on → Output 4 on
- Input 1 off → Output 1 off
- Input 2 off → Output 2 off
- Input 3 off → Output 3 off
- Input 4 off → Output 4 off

Configuration

- Create a project with PLC of firmware version 2.5

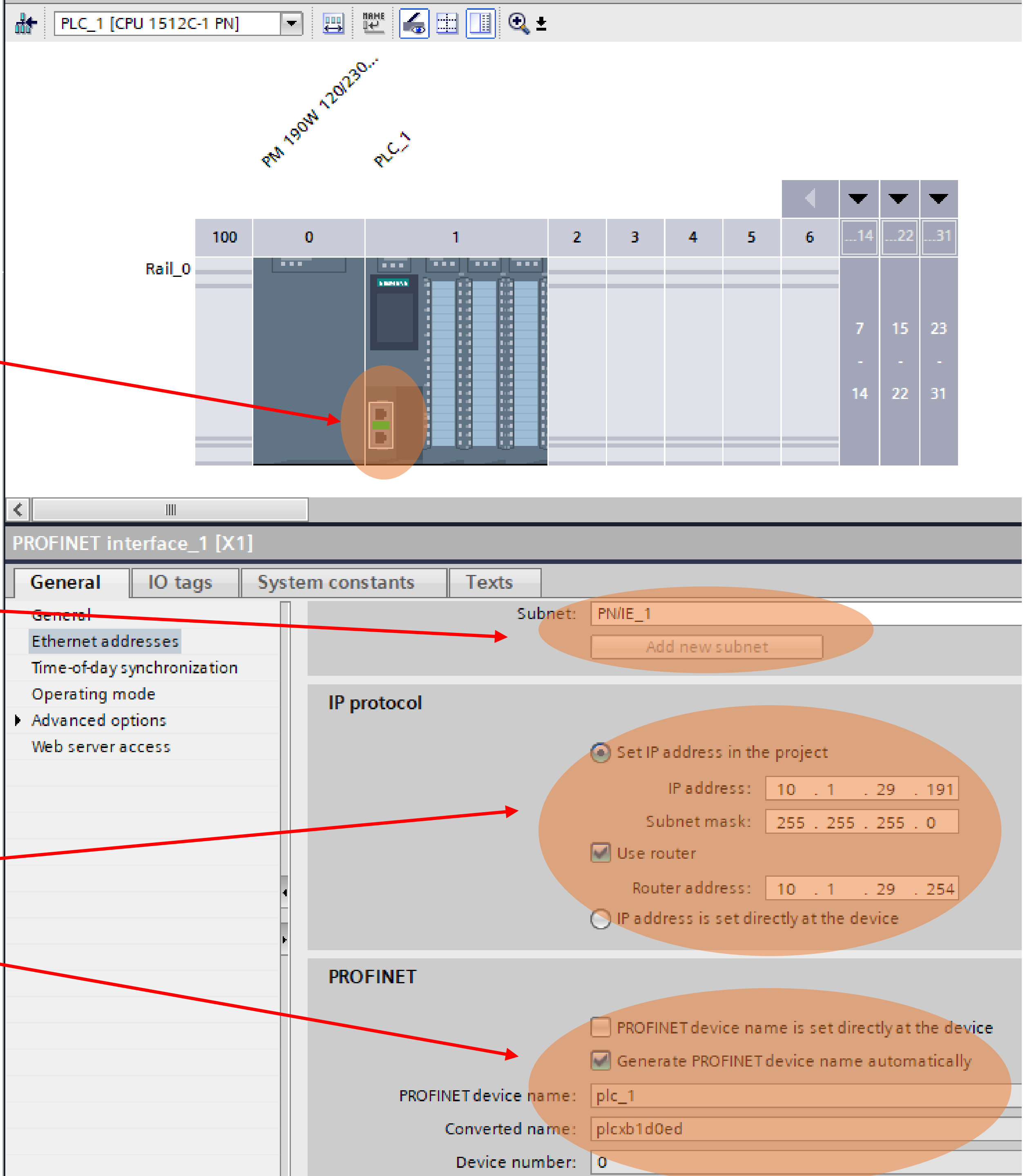


Network configuration

Select Ethernet symbol

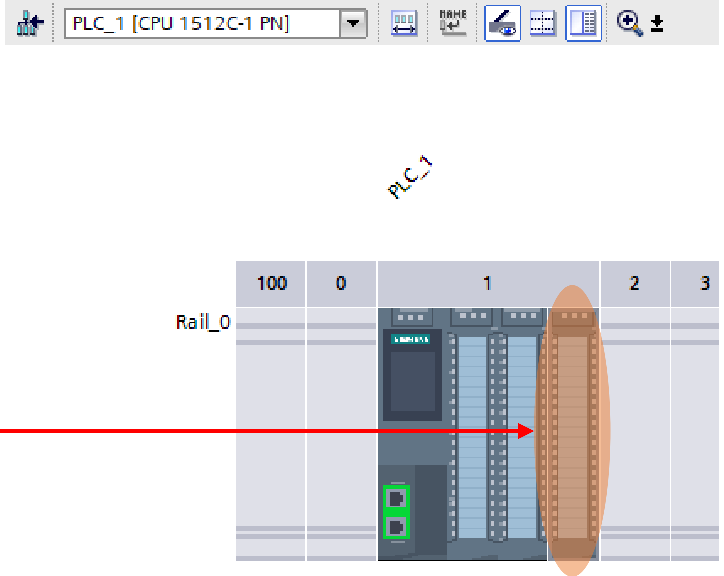
Add new subnet

Set IP and Profinet

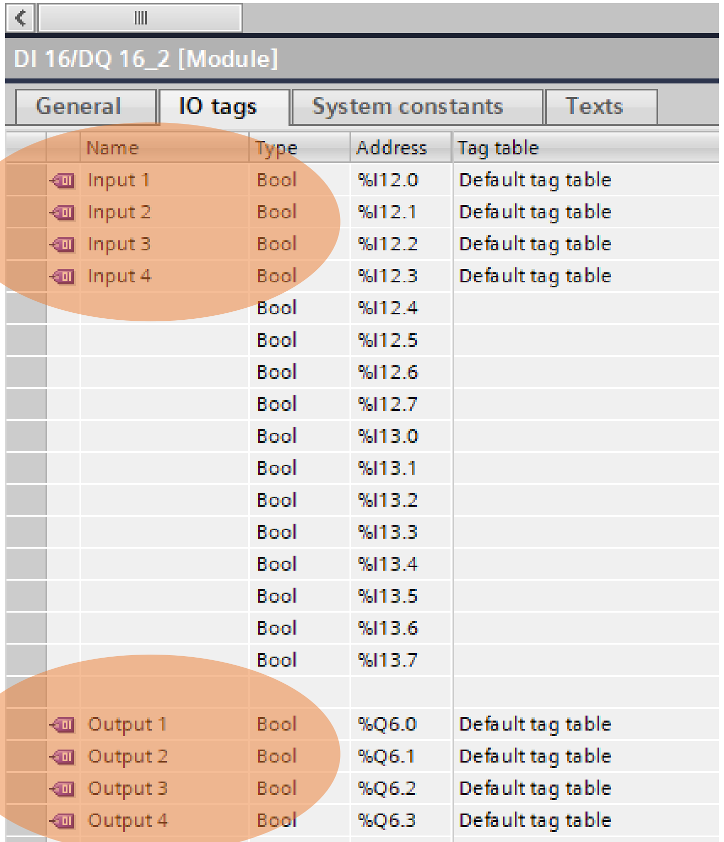


PLC tags

Output of the emulator card is mapped to this module

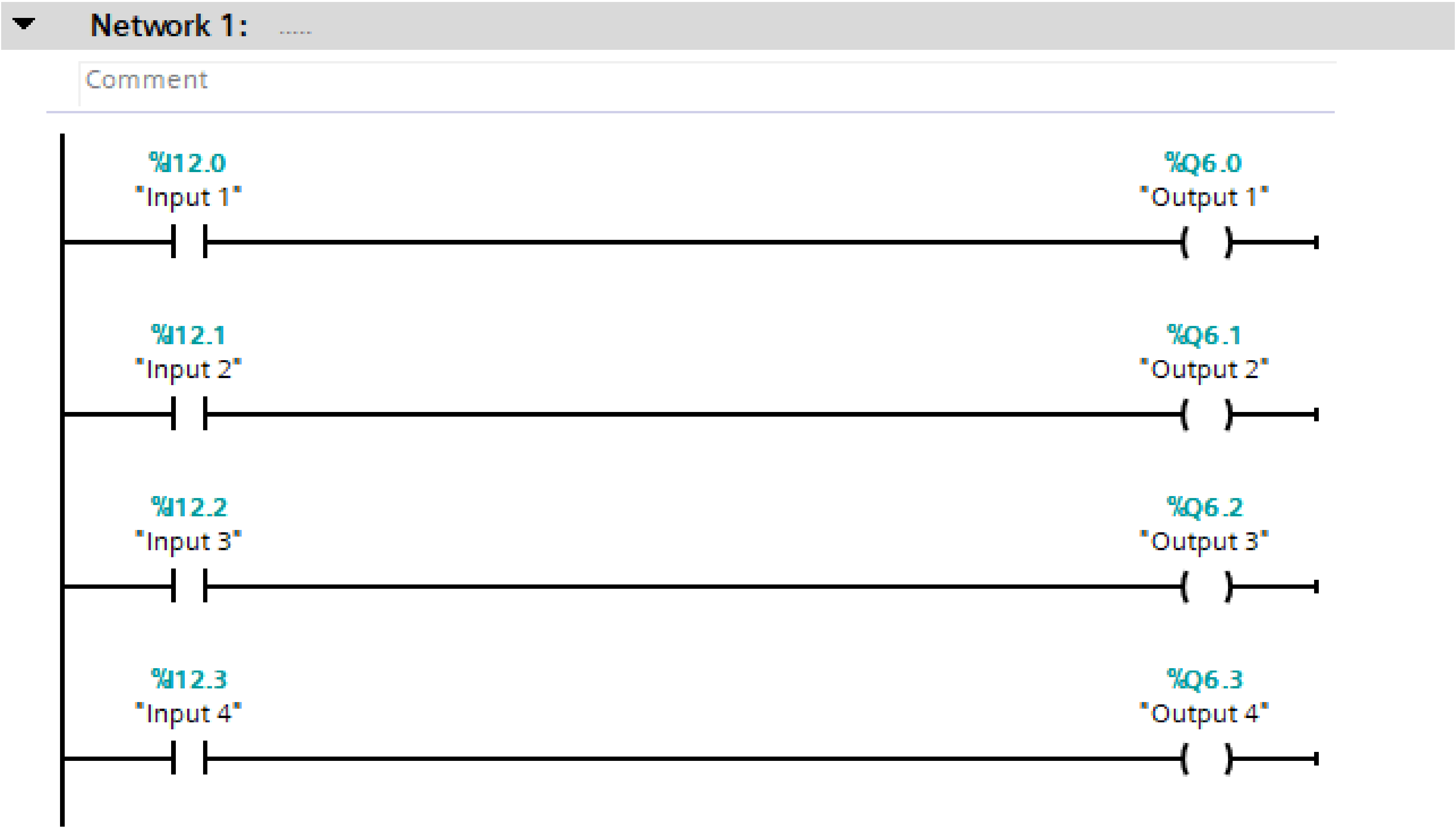


Set inputs and outputs



Name	Type	Address	Tag table
Input 1	Bool	%I12.0	Default tag table
Input 2	Bool	%I12.1	Default tag table
Input 3	Bool	%I12.2	Default tag table
Input 4	Bool	%I12.3	Default tag table
	Bool	%I12.4	
	Bool	%I12.5	
	Bool	%I12.6	
	Bool	%I12.7	
	Bool	%I13.0	
	Bool	%I13.1	
	Bool	%I13.2	
	Bool	%I13.3	
	Bool	%I13.4	
	Bool	%I13.5	
	Bool	%I13.6	
	Bool	%I13.7	
Output 1	Bool	%Q6.0	Default tag table
Output 2	Bool	%Q6.1	Default tag table
Output 3	Bool	%Q6.2	Default tag table
Output 4	Bool	%Q6.3	Default tag table

Programming the PLC



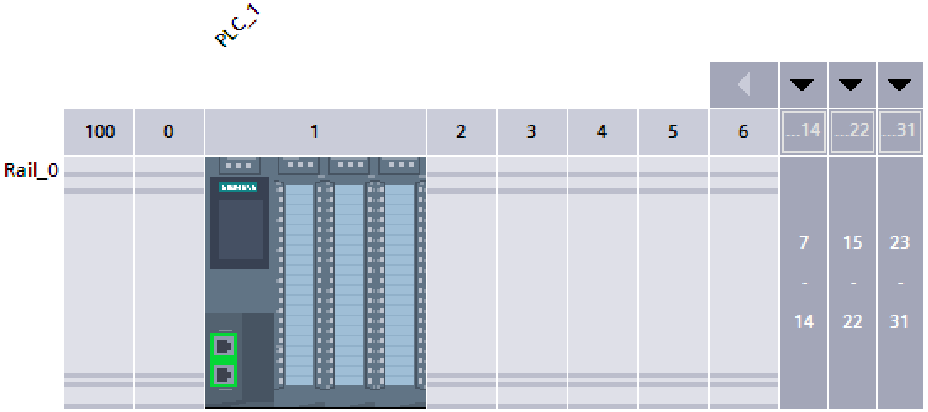
Compile and load config/code to PLC

In practical, at this point, the code and config are to be loaded to the destination PLC.

TIA Portal run in a PC

Compiled codes and configs

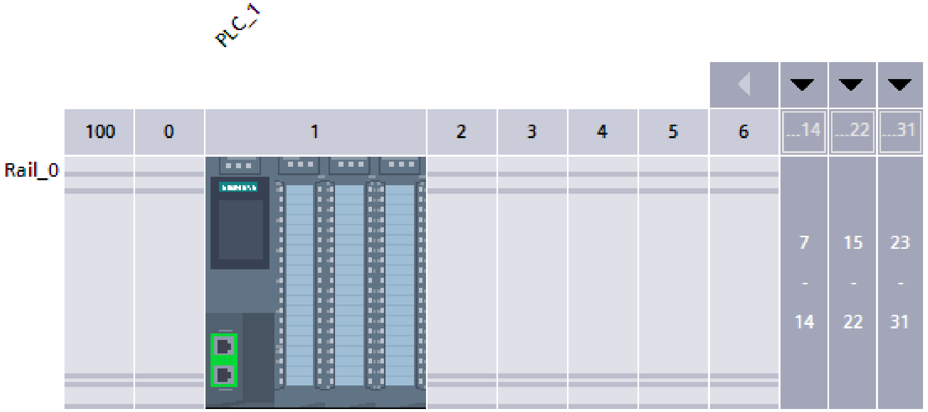
Download



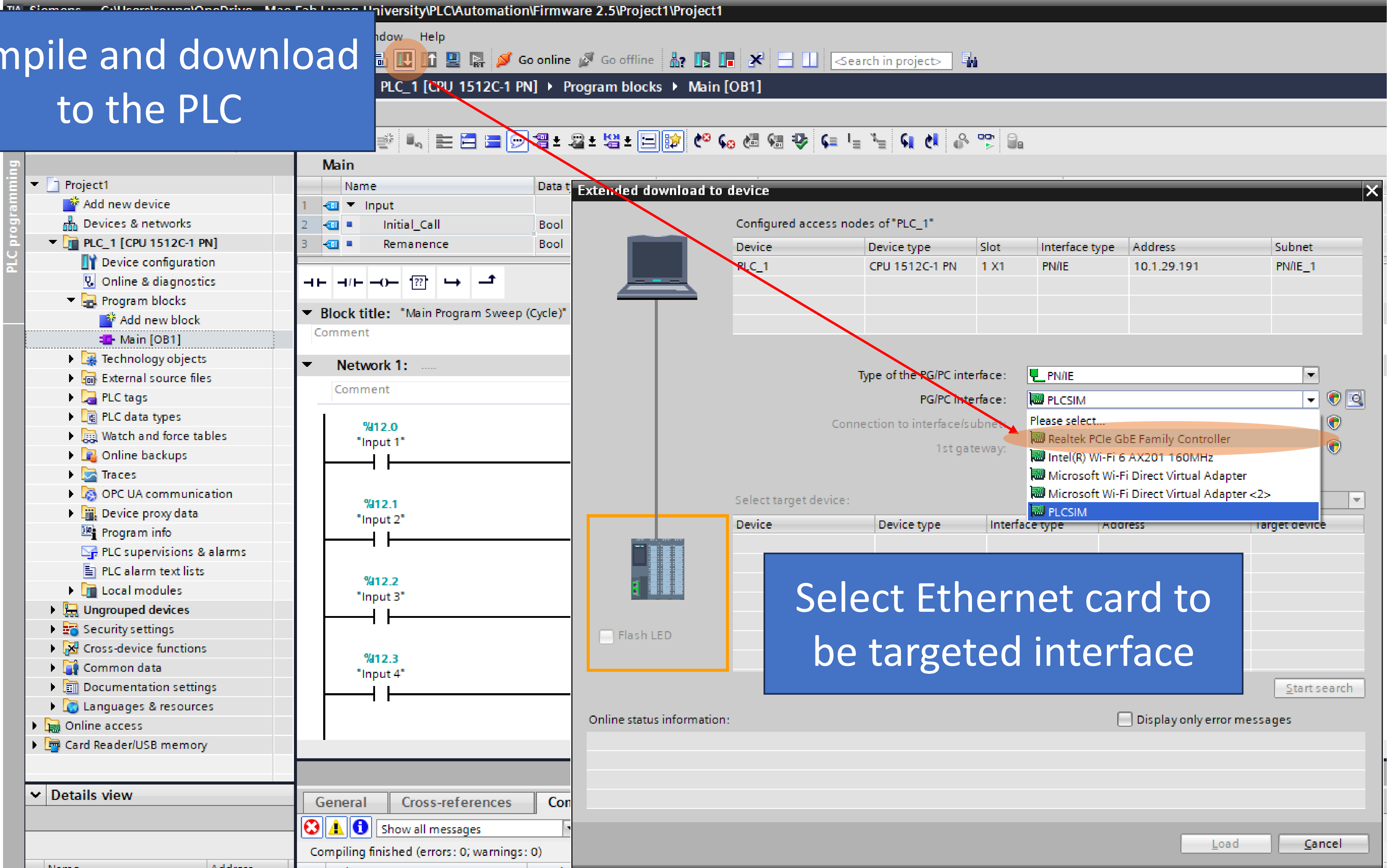
If there is no error, the TIA Portal and the targeted PLC are svnchronize. The PLC will run automatically but the Portal can monitor and debug.

TIA Portal run in a PC

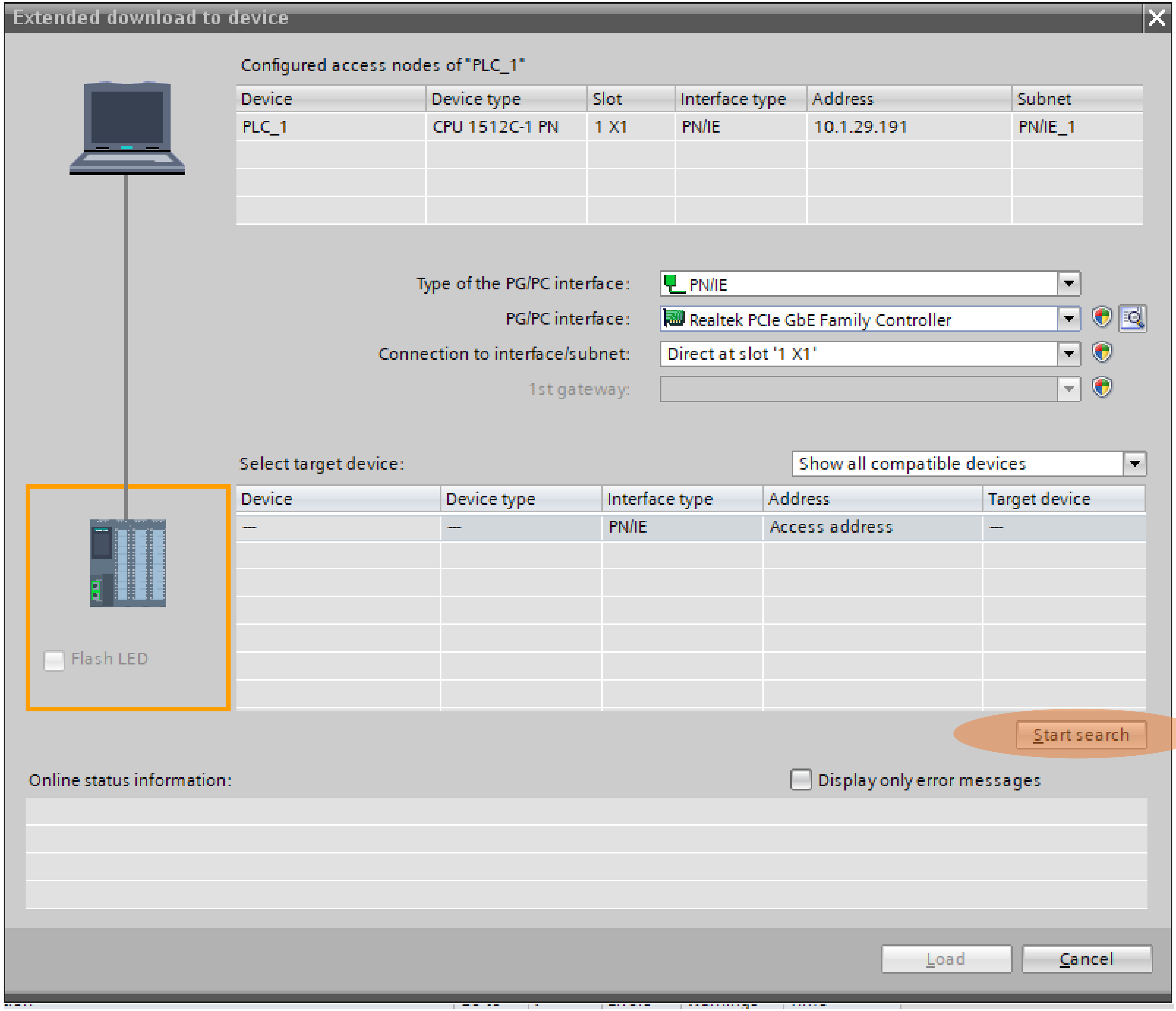
Synchronize



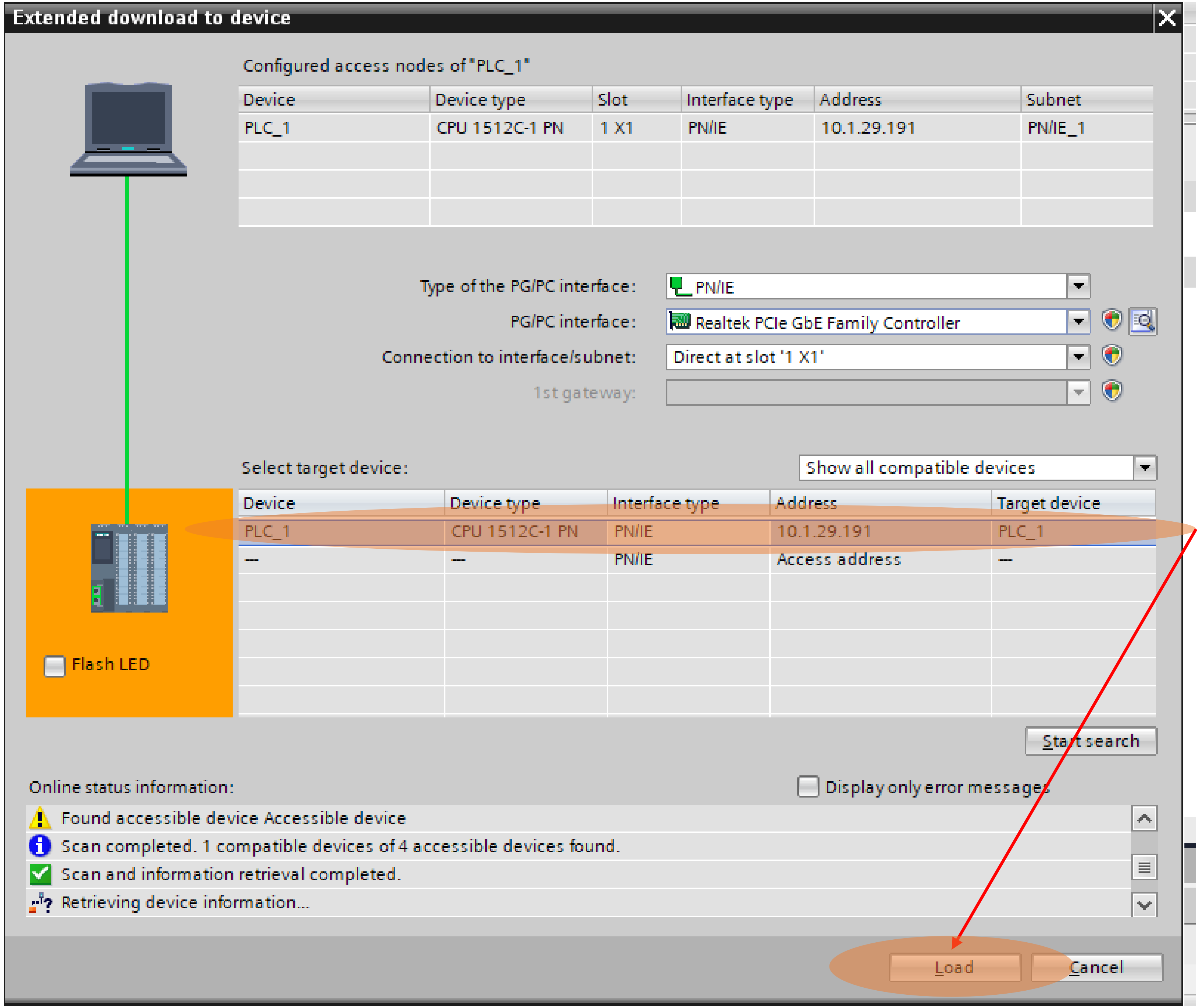
Compile and download to the PLC



Select Ethernet card to be targeted interface

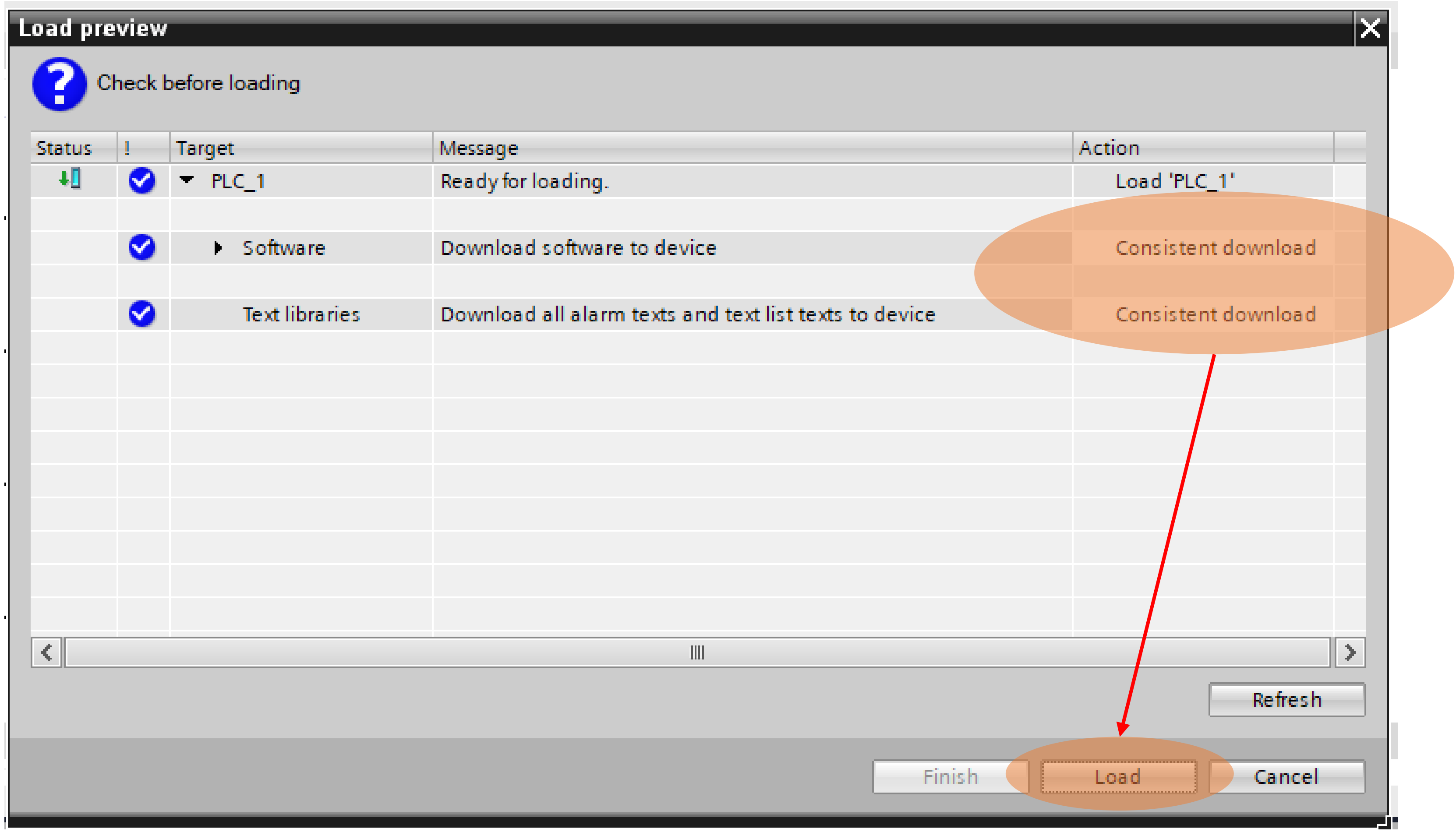


Search for the PLC in the subnet



Found the targeted PLC and select it

Load the config and code to the PLC

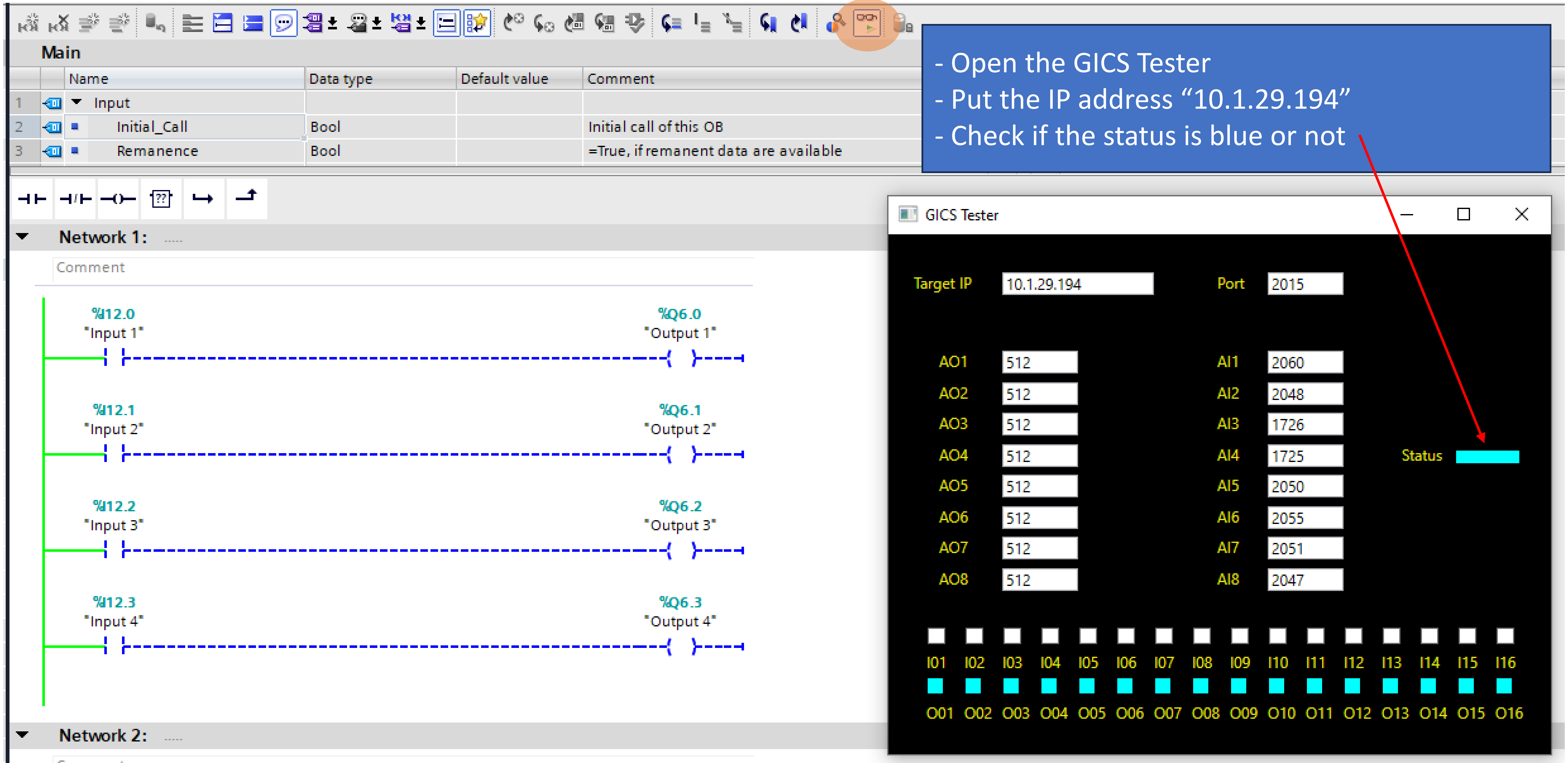


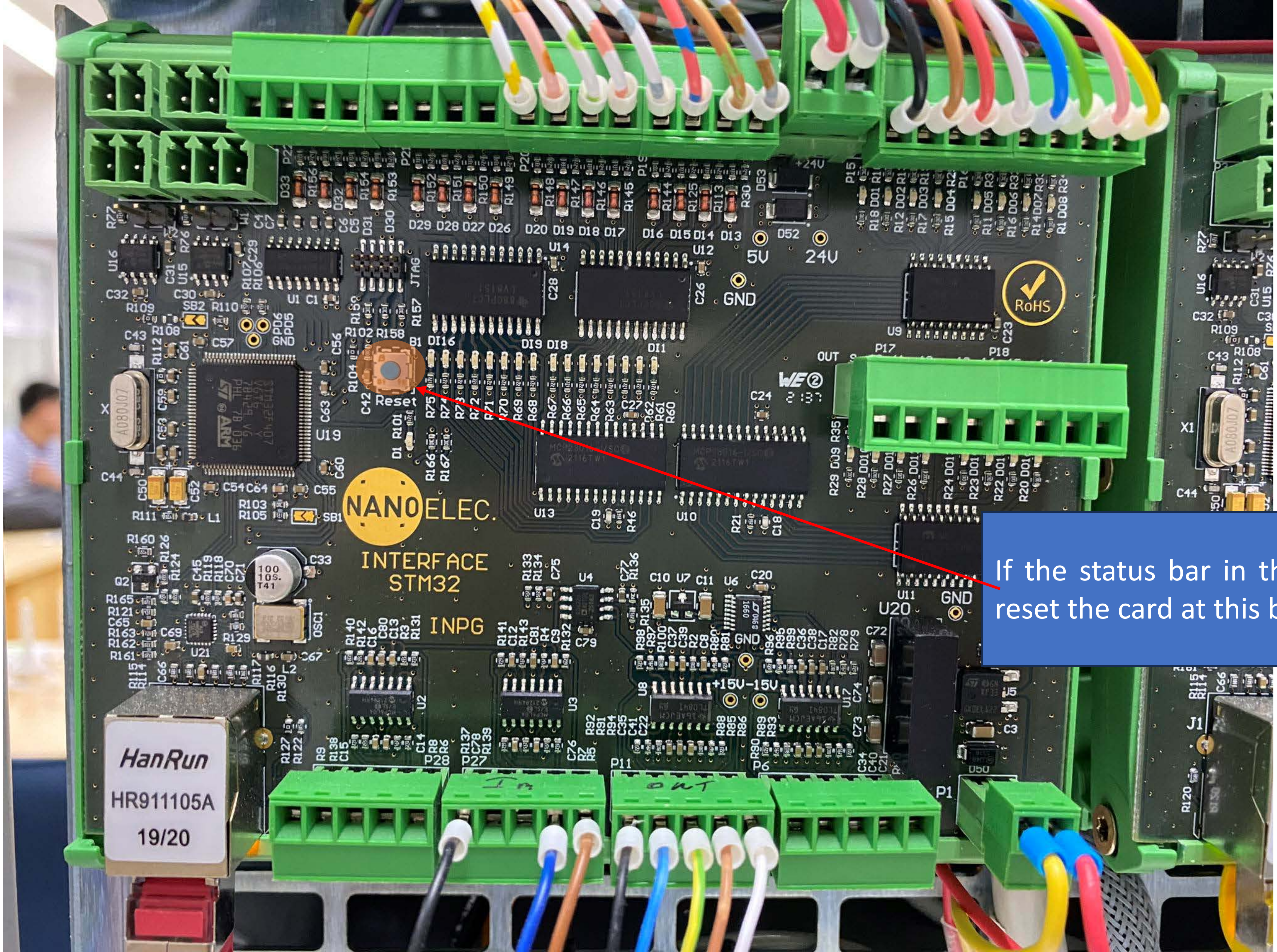
Check the loading option

Load the config and code to the PLC

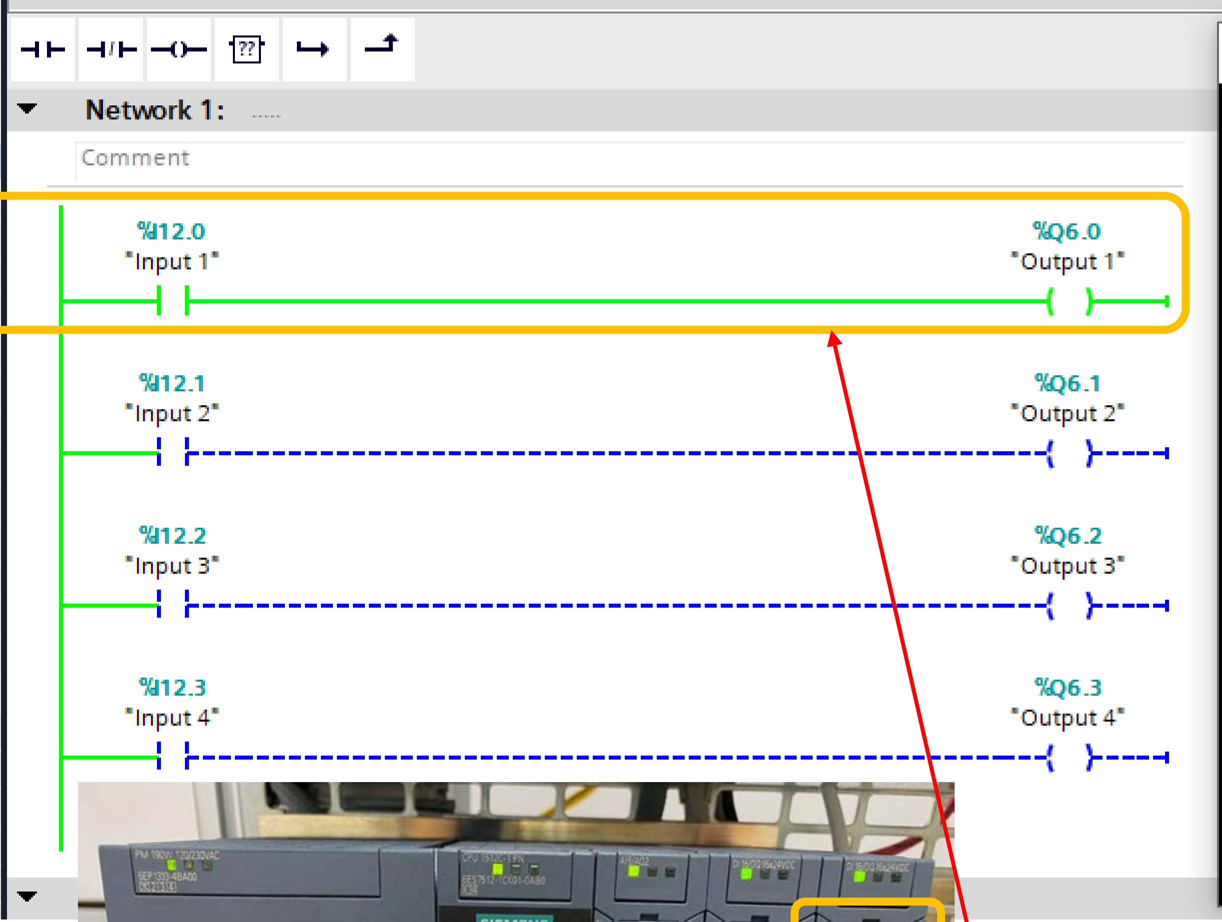
If the download success, make the TIA portal online with the PLC

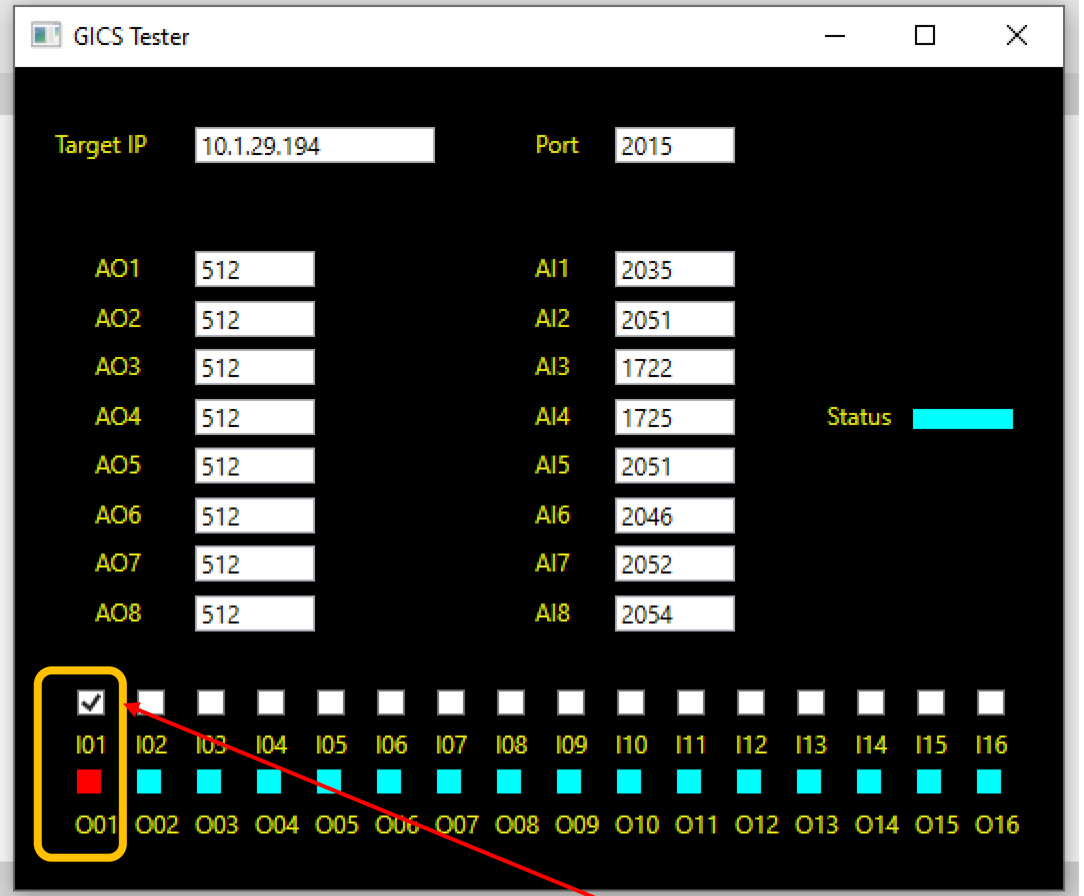
- Open the GICS Tester
- Put the IP address “10.1.29.194”
- Check if the status is blue or not



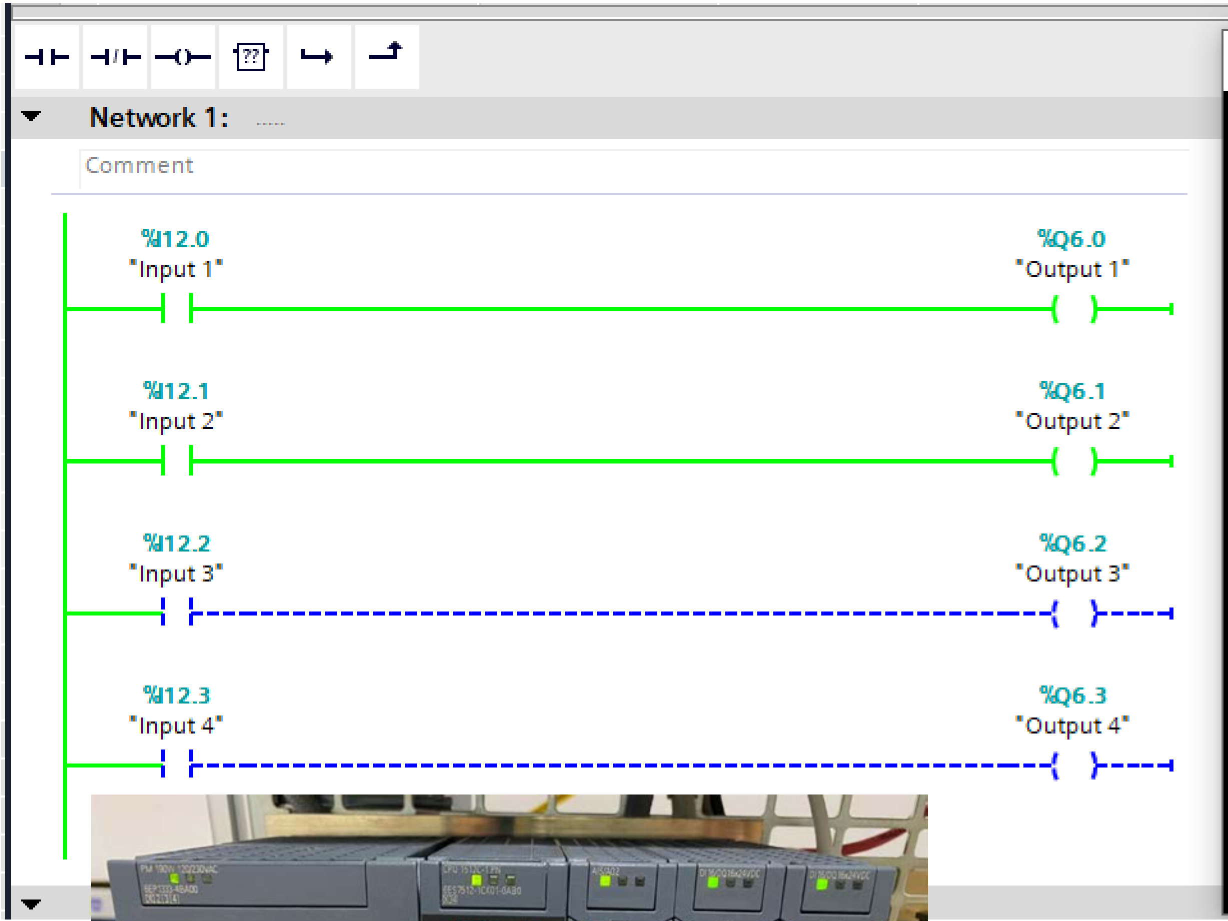


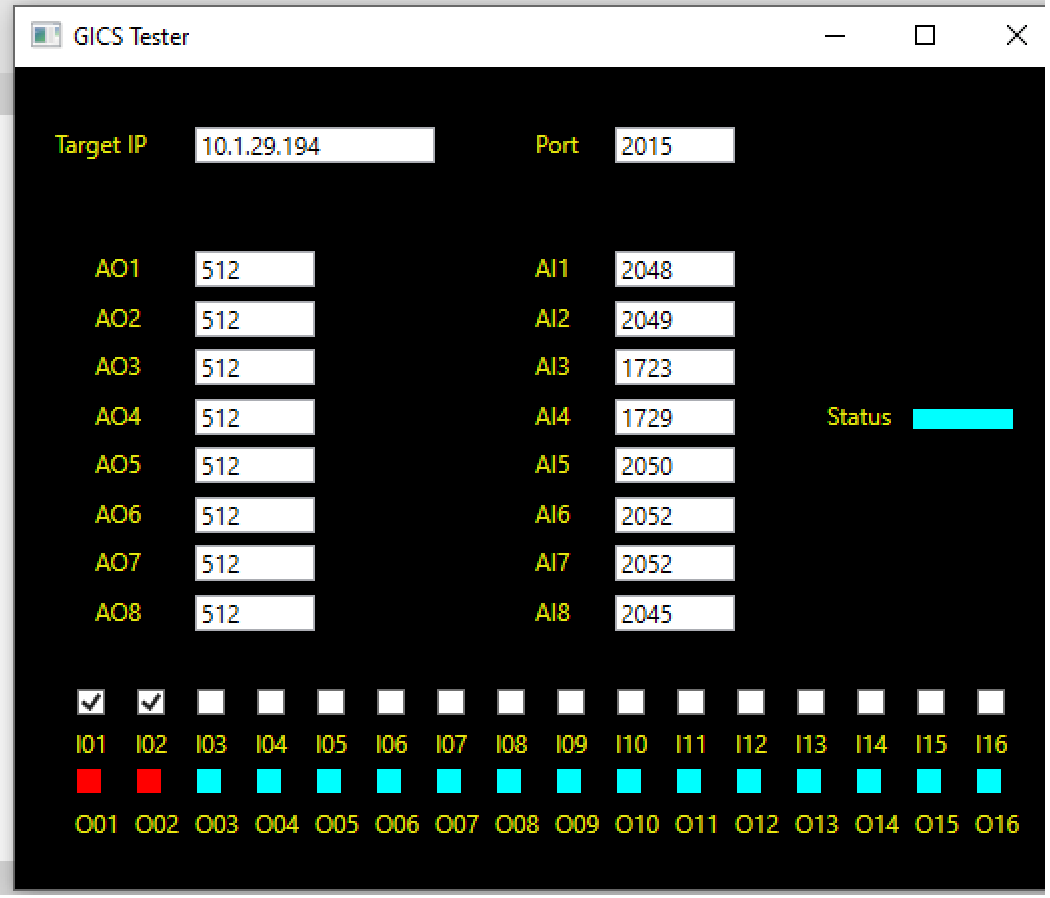
If the status bar in the GICS Tester is not blue, reset the card at this button.



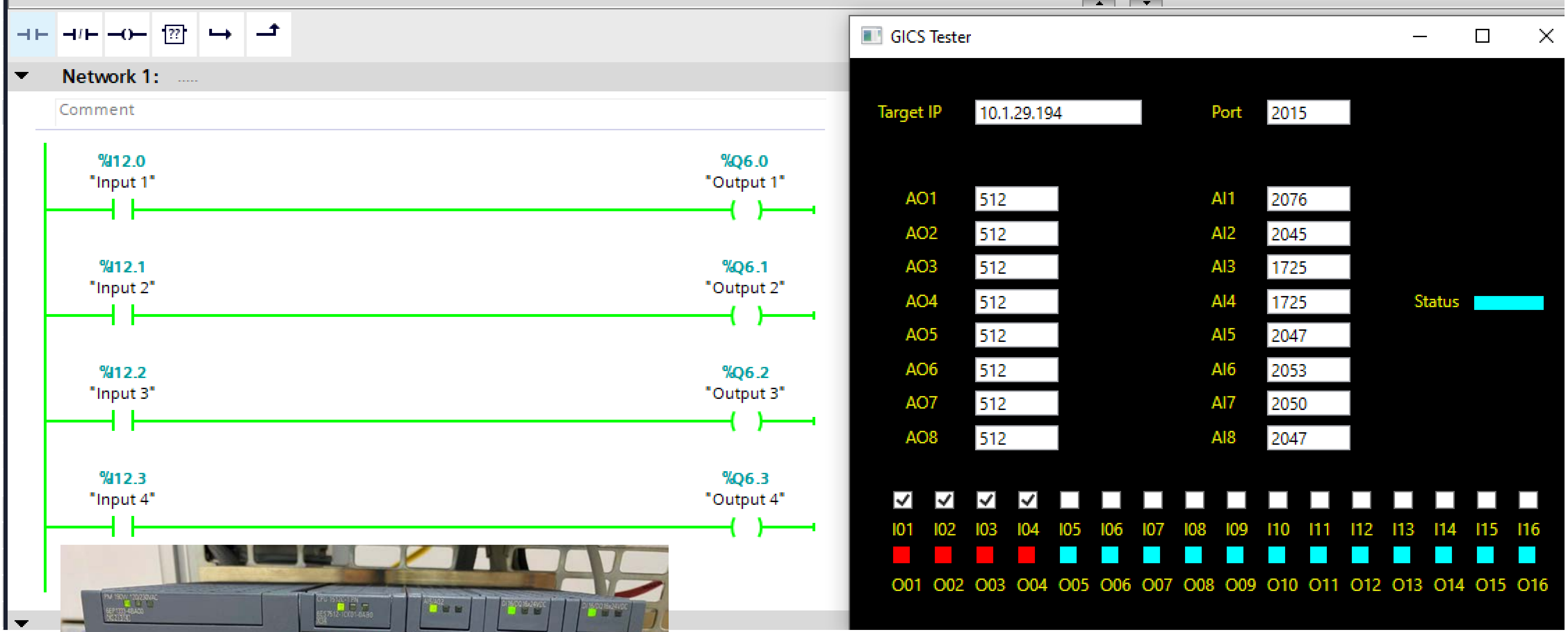


- Click to turn on “I01” of the card
- PLC’s “Input 1” will turn on which will also turn on PLC’s “Output 1”
- The card’s “O01” will turn on
- I/O lights on the PLC module are lit

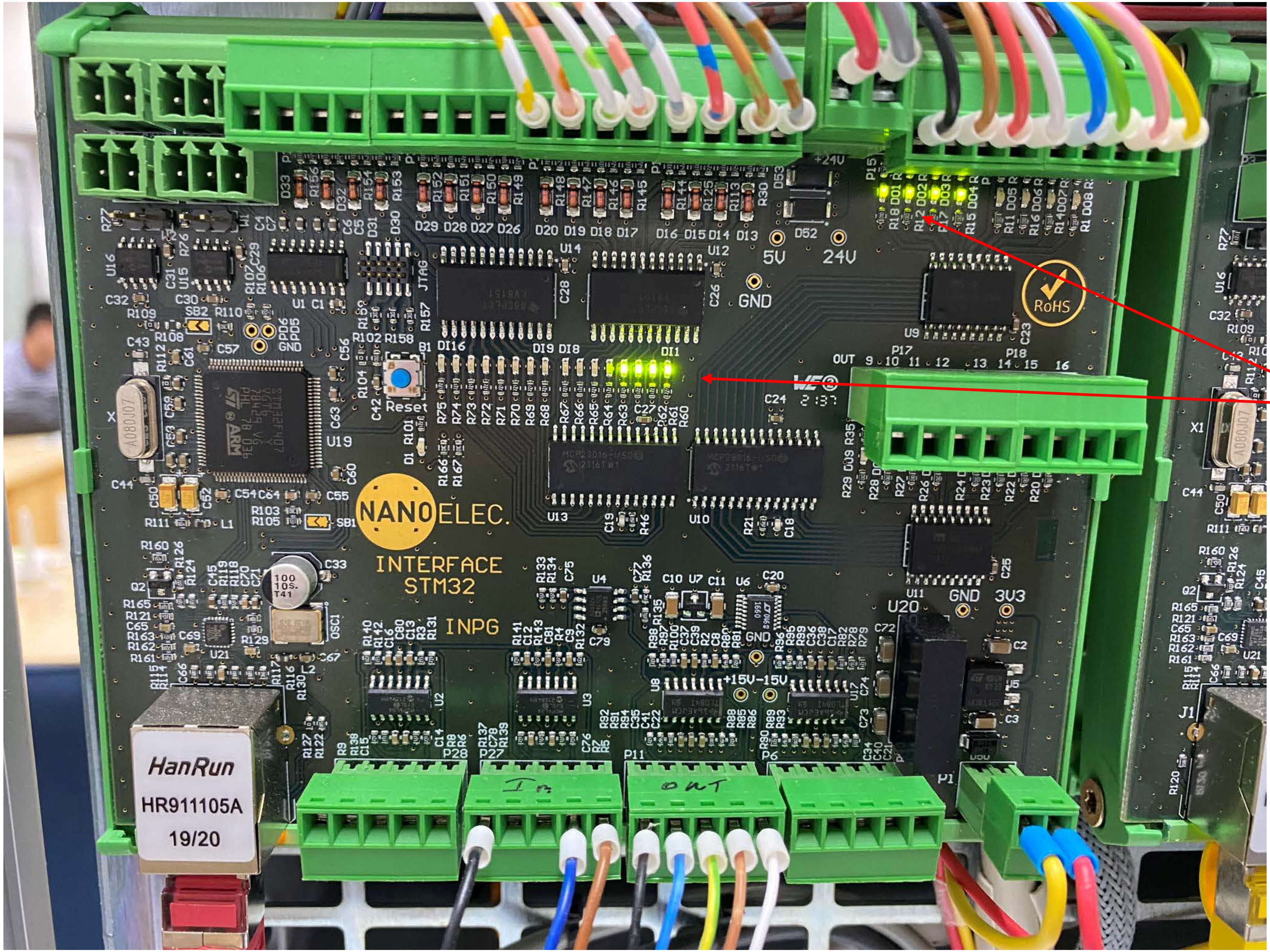




- Turn on two inputs to the PLC: “I01” and “I02” on the card
- Two inputs and two outputs should be turned on



- Turn on all inputs to the PLC: “I01”, “I02”, “I03”, and “I04” on the card
- Four inputs and four outputs should be turned on



There are light indicators of I/O on the card

Exercise

- Create a project based on the mixing problem shown on the right figure
- Map I/O to the emulation card and emulate the system