Industrial communication labs (Siemens)

1. Introduction

Modern complex Industrial Control System (ICS), which includes SCADA and Distributed Control Systems (DCS), heavily relies on the use of a communication system tailored for the real-time and reliability requests of distributed applications. In contrast with general communication systems (like Ethernet/TCP/IP networks and other Internet related technologies) industrial communication systems are part of the controller and are able to guarantee that real-time constrains of the application will be met. Despite the fact that some of the protocols are Ethernet embedded (like GOOSE and ProfinetI/O) or even TCP/IP transported (like ModbusTCP and S7) the deployment of the TCP/IP stack requires some modifications in the Internet protocols (like sockets reuse, never-closing TCP connection, Ethernet Vlan frames tagging and priority, etc) that brings the TCP/IP stack close to a deterministic behavior. Therefore, for a control engineer, the practical knowledge of industrial communication protocols like behavior of the data flows, relationship between PLC programs and network connections, bandwidth use and optimization is an important requirement in order to be able to deploy an industrial control system. The purpose of these labs is to let you acquire a minimal knowledge in the industrial communication field.

Lab objectives

1) Program an application using Grafcet

2) Understand the communication between PLC and HMI/SCADA

2) Implement and observe communication flows

3) Analyze the flows, optimize the communication

2. General organization of the lab, work environment

The lab is organized in two workshops corresponding to the study of one major SCADA protocol each: S7 (Siemens) and Modbus/TCP (Schneider). Several PLCs are available for S7 and for Modbus/TCP, together with development computers, a SCADA software (TIA Portal for S7 and PC Vue for Modbus/TCP) and HMIs. Each student will work alone on one machine.

2.1 Network:

The 10.10.0.0/16 network interconnects different elements:

- The PLCs.

- The "configuration" computers (10.10.4.x). -

- The "process simulator" cards (10.10.100.x, visible on the PLC User Manual of Siemens).

- The Human Machine Interfaces (HMI).

These elements are interconnected via two CISCO switches.

Control architecture and supervision architecture

You have to understand and specify the global functioning of the architecture and the control loop. This architecture consists of:

Element of the architecture for control Interfaces	Interfaces
• A PLC to receive the command	- Analog and digital inputs/outputs - Network card - Special case: 10.10.4.1: Remote I/O via the left ET200S interface via Profibus (the one without IP address...)
• A "physical process" represented by a simulation card (via address 10.10.100.x) embedded SMT32.	-Analog and digital inputs/outputs - Network card
• A remote-control interface (GICS Tester) implemented on the "configuration" machine (10.10.4.x)	- Network card

QUESTION: Draw the control loop with the controller, the actuators, the sensors, the action and specify the interfaces used and the links/modes of "connections" (ex: network(x), direct link I/O...) Beyond the control loop, the architecture is also used for the configuration.

2.2 Identification of your working space

The first aspect will be to identify your computer (4.X), your PLC, your HMI, your simulation card, and their relative IP addresses. These addresses should be kept (not changed). Describe your architecture, the interconnections between elements, the IP addresses...

2.3 Getting to know your workspace

On each computer a local account is opened for the students (ex MISTREA01, you must ask the teacher what is your login and password on your machine, and note it down).

The useful software is:

- Software to configure your PLC: TIA Portal for Siemens

- A remote control interface GICS Tester

QUESTION: What is the use of the GICS Tester control interface?

2.4 Configuration of the GICS Tester remote control interface

On the GICS Tester control interface (the "configuration" computer), the IP address of the simulation card you are using must be configured as "target IP": 10.10.100.x, on port 2015. We will not use the analog inputs and outputs (the digital values are between 0 and 4095 for voltages between -10 and +10 V).

From now, you can test the outputs by forcing them through the GICS Tester control interface and by observing their variations on the physical interfaces of the I/O cards.

DO NOT TOUCH THE PHYSICAL WIRING.

If necessary, use a table to match the I/O numbers on the physical interface with the numbers on the GICS Tester control interface.

2.5 The simulation card

For practical (available space) reasons, it is not possible to have 12 plants to be controlled by the 12 PLCs. Therefore, the plant is simulated by a “simulation card” which is actually an embedded system built around a microprocessor and allowing programming for simulation of plants... However, the PLC receives “true signals” on the I/O card interacting with the “simulation card”. Figure 2 displays the lab configuration (it is called a Hardware in the Loop simulation).

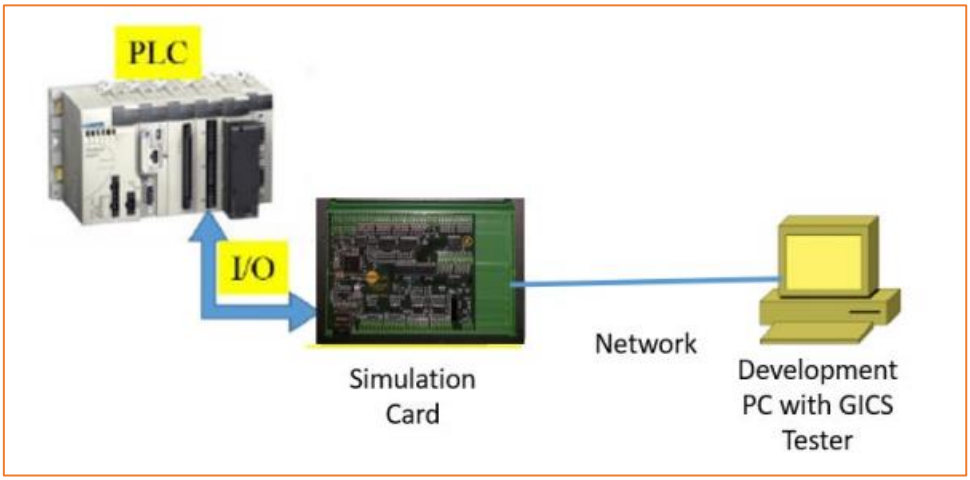


Figure 2. Hardware in the loop bench

Control Problem:

The control specification concerns a system based on a tank to mix-up two products with engine (M) (one yellow product1 (VP1) and one blue product2(VP2)) into a green product will be evacuated through the valve (VE). All the specifications are detailed in the table 1 as follows;

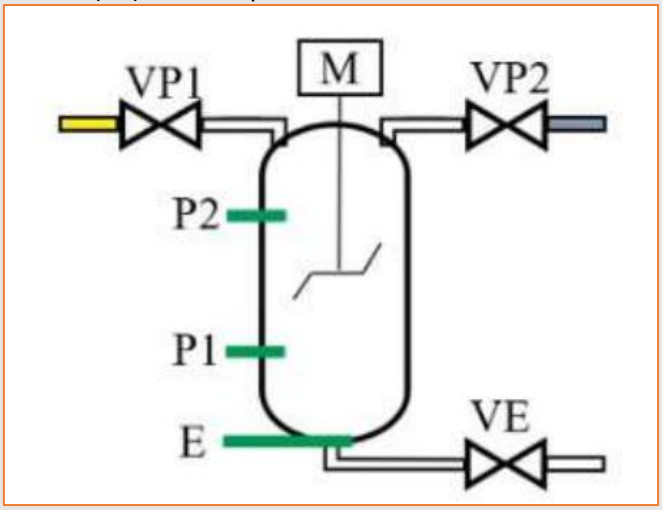


Figure 3. Sequential Systems

Specifications:

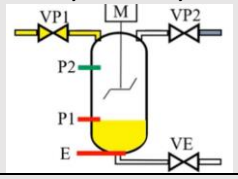
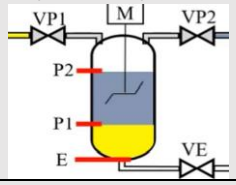
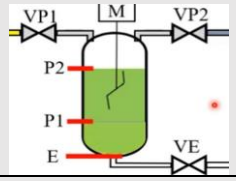
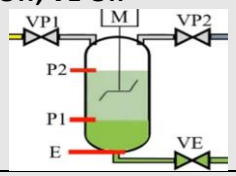
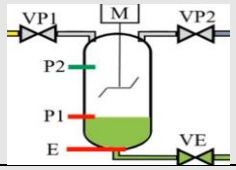
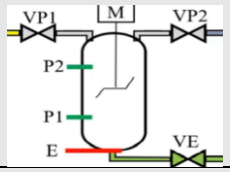
T0: E OFF, VP1 ON	tank is empty (not E) open valve yellow
T1: P1 On, VP1 Off, VP2 On 	P1 sensor (level) is reached VP1 closed, VP2 open valve blue product
T2: P2 On, VP2 off, M On 	P2 sensor (level) is reached, VP2 valve closed, activation of the engine M for mixing.
T3: Timer for 10 s 	The mixing (blending) operation will last a certain time. We use a timeout for that.
T4: M Off, VE On 	At the end of the mixing, we stop the engine, and open VE, which is the evacuation valve.
T5: P2 Off 	The tank is emptying, P2 is deactivated
T6: P1 Off 	Then P1 is deactivated

Table 1 Specifications of the system problem

3. Work to do

You'll have to:

A. Write a control program for the PLC (to control the tank according to the example given up in table 1).

B. Set-up the communication between the SCADA and the PLC.

C. Check the communication flows using Wireshark.

Configuration of the PLC and I/O

In the first part of the lab, you have to build a simple SCADA system. The system schematics are presented in Figure 4.

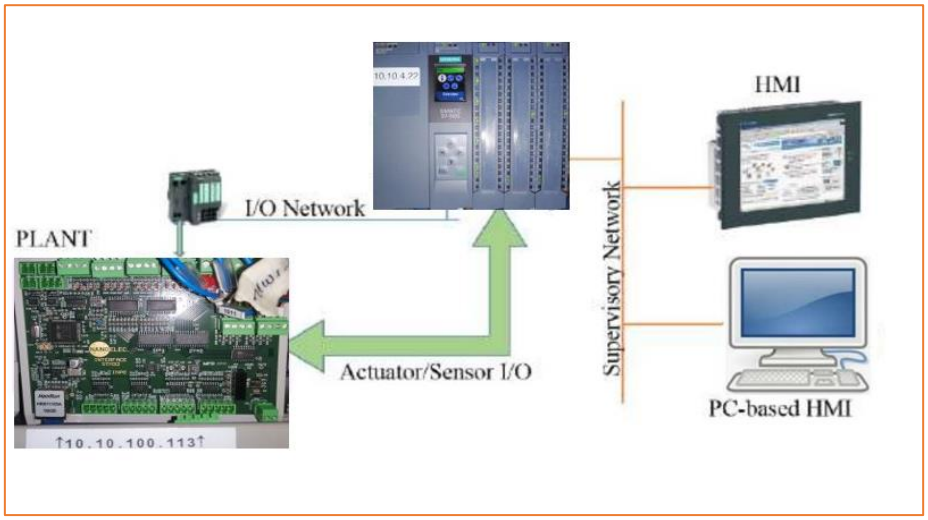


Figure 4. The simple SCADA system

The PLC will interact with the plant either by direct data exchange with the sensor and actuators connected to the I/O cards or through a remote I/O unit accessible via a dedicated network. The process data (including inputs, outputs and device status) is collected by the SCADA system and displayed by a field HMI or a PC-based software.

a. Project configuration of your PLC

On the "configuration" computer, you will configure your PLC (depending on the PLC and the environment, these operations can be more or less automated...), the objective is that you do it to see and understand the different steps. It is an operation where you have to be rigorous and careful, choose the right elements and configure them well, because then the corresponding drivers will be sent to the PLC... and must correspond to the precise elements of the physical architecture of the PLC. You have to configure everything including the units you will not use (for example the analog I/O) because the software and hardware configurations have to match.

With the help of PLC User Manual of Siemens (page 6/7):

Launch "TIAPortal"

• Create a new project

• " Configure a device " (if needed, you may have to " Add new device ", it will depend on your PLC and CPU, some PLCs are more or less " integrated ") (page 6).

Compare your PLC with the other PLCs in the room, is yours more integrated (integrating CPU, power module, I/O) or is it more made up of grouped elements)?

• On some PLCs, you will need to explicitly add the power module (page 7).

Note: Depending on the PLCs, the power supply and I/Os can be internal or external, there can be remote I/Os that will be joined for example via a field network: Modbus, CAN, ...

• On some it will be necessary to add analog and/or digital input and/or output modules (AI, AQ, DI, DQ...)

The important point here is that the architecture must be configured in accordance with the actual physical architecture of the platform components. The software architecture will then be "compiled" and sent to the real PLC.

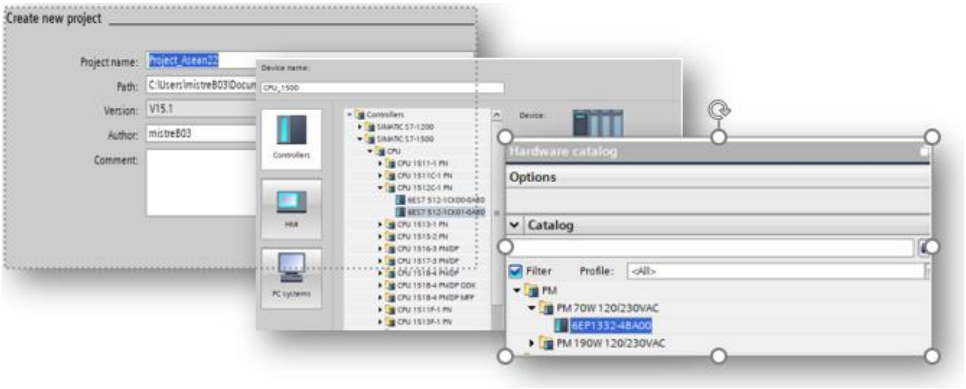


Figure 5. Follow's steps page 6 (Create project+ Configure a device + Add power module)

b. Configuration of the PLC network interface, clock synchronization

The PLC will use at least one network interface. PLCs usually have one or more IP network interfaces (RJ45) and a Profibus network interface on Siemens PLCs (RS 485).

Carefully consider which network interface(s) will be used (Note: for the 10.10.4.1, the Profibus connection must be configured with the ET200S remote interface).

The IP address will be that of the PLC.

- For that "add new subnet". It will be necessary to configure the exit gateway to have access to Internet, that will allow to connect on the network ("use router").

Check that your "configuration" machine has access to the network (by testing an external site for example).

- To have a good synchronization between all the APIs in an architecture, it can be interesting to have a reference clock (NTP, Network Time Protocol). We can use the clock of Grenoble INP (search on internet), check that the address is 147.171.64.9.

We will finish the connection of the network interface by allowing the web access (webserver access => enable).

QUESTION: Why give the web access?

- The next step consists in configuring the network interfaces (external network, possibly network to the remote I/O) as well as the I/O. Before configuring the I/O, check on the physical architecture of the PLC which I/O are actually wired and usable.

You can follow the steps in PLC User Manual of Siemens (Page 8/9/10)

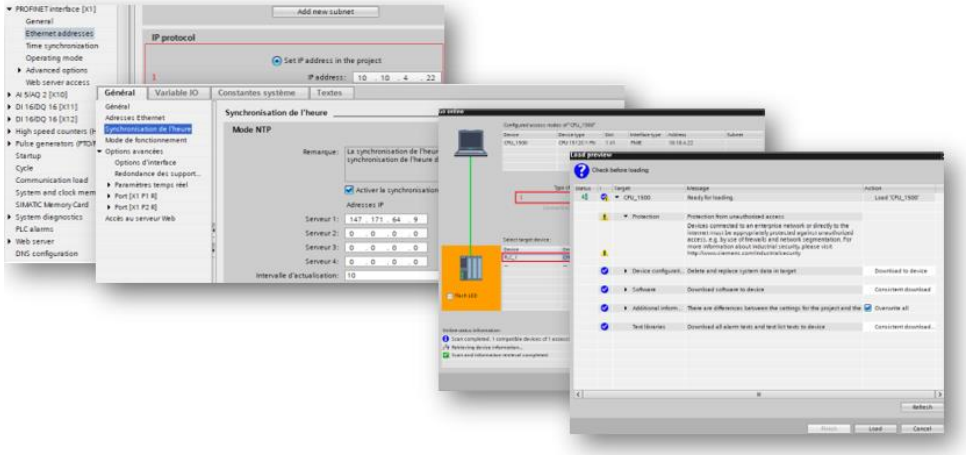


Figure 6. Steps: Set network + Time synchronization+ Hardware detection Device+ Load preview

C. Compiling and sending the configuration to the PLC's CPU

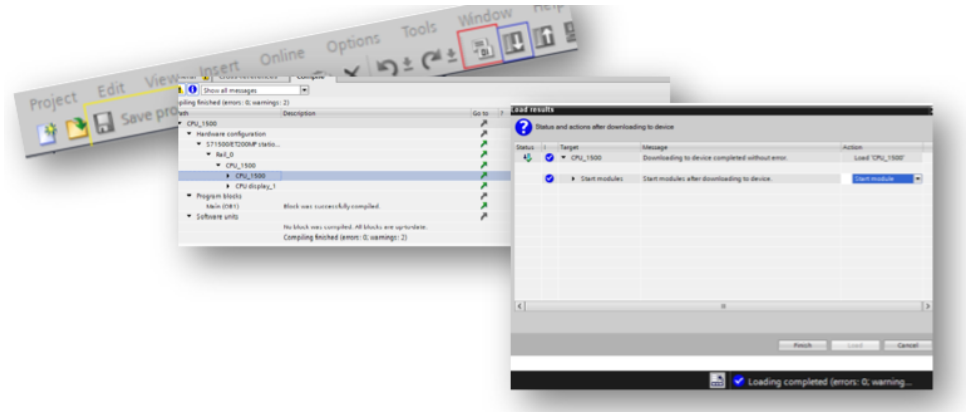


Figure 8. Save and compiling steps in PLC User manual of Siemens(page11/12)

- ❖ It is necessary to compile your project (right click on the name of the PLC), it is necessary to compile the hardware when making a new PLC configuration and the software when developing new programs.

☺ When the compilation is correct, the program must be sent to the PLC's CPU ("Download to device"). Then we will have to launch a search ("start search") to find an API compatible with the architecture that we have configured.

QUESTION: At compilation we may have a security "warning" because we have given access to the web server, why this warning?

✓ Debugging and execution mode

"Go online" on Siemens allows you to go into debugging mode (mode that allows you to see what is actually happening on the PLC itself during the execution of programs).

d. Configuration of variables

In "Siemens", variables are accessible via the "General" menu. The managed variables are of four types:

• The physical outputs (%Q or %A in German)

• The physical inputs (%I or %E in German)

• The variables stored in memory (%M)

• System variables (%S)

Sometimes the variables will be on 16 bits (%IW), W for Word (16 bits word).

In this part, we will have to configure the six inputs and outputs of our tank system...

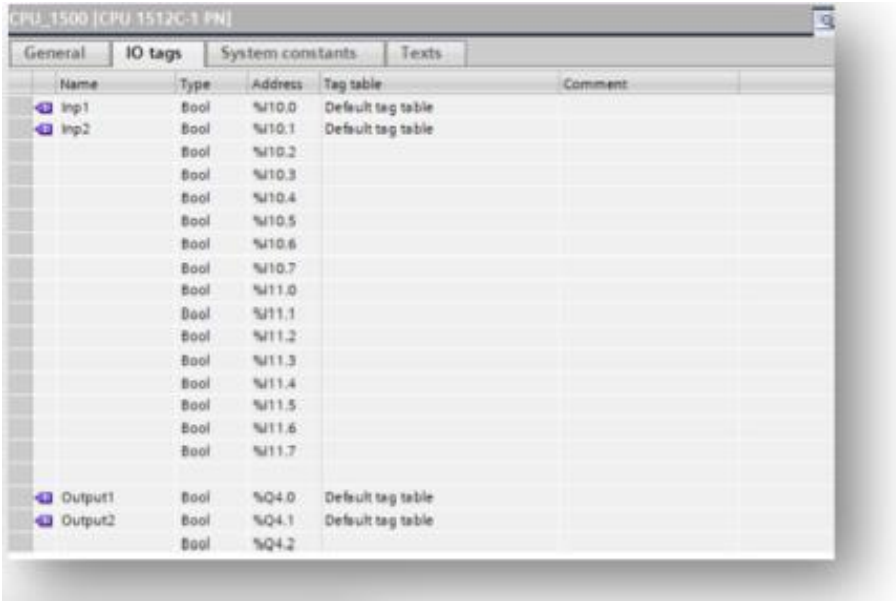


Figure 7. I/O Configuration in PLC User Manual of Siemens (Page 12)

e. Writing a program, compiling, sending, executing

We can program in ladder the control problem respecting the specifications of the subject.

You can follow the steps in PLC User Manual of Siemens (Page 13/14/15)

QUESTION: Explain in your report your understanding of the specifications, make technological choices (normally open valves, normally closed valves...) then explain your program. N.B.: If you are late, do not waste too much time in the session, make simple programs and test them on the machine.

Then you need to perform the following actions:

• compilation,

• send to the CPU ("download to device", possibly if needed choose "continue without synchronization"),

• On SIEMENS: do " start CPU " (in the icons at the very top of the screen) " load ", " monitor";

on " Go Online ", you should be able to see what is happening in the PLC as it is indicated in the PLC user manual of siemens.

f. SCADA interfacing

Generally, the HMI will be an industrial touch screen (it can be also a computer-based HMI) using the industrial supervisory control software PCVue for instance. Your HMI has to display the values of the two-level sensors, the actuators controls, the error bit and error code from the PLC and a control allowing for error acknowledgement.

Details on PLC programming, digital I/O mapping and HMI interfacing depend on the hardware available for your lab.

See with the teacher which HMI you will use (local or remote and so the way to interact with it).

(1) HMI creation

(Follow steps in Lab Siemens (Following3.) IHM Implementation)

QUESTIONS: On the platform, the elements are interconnected via two CISCO switches.

How does a switch work? What is its role?

The architecture described in Appendix A proposes a hub with a monitoring link to which all the supervision machines (10.10.3.x) are connected.

How does a hub work? What is the role of a hub in general?

What is the role of the hub in this diagram?

To get out on the internet, this network has a gateway, not visible on the general diagram. This gateway uses the IP address generally used for gateways.

What is this address?

j. Conclusion

At the end of the lab, you shall be able to:

- Understand a practical SCADA communication architecture

- Understand the rationale of some industrial protocols (Modbus/TCP and S7 for instance)

- Identify practically data flows in an industrial network using a network sniffer

- Find the relationship between functional data flows (what your project needs) and physical data flows (what the network really carries)

APPENDIX A. G-ICS network architecture

Only the devices used for the lab are included into the diagram

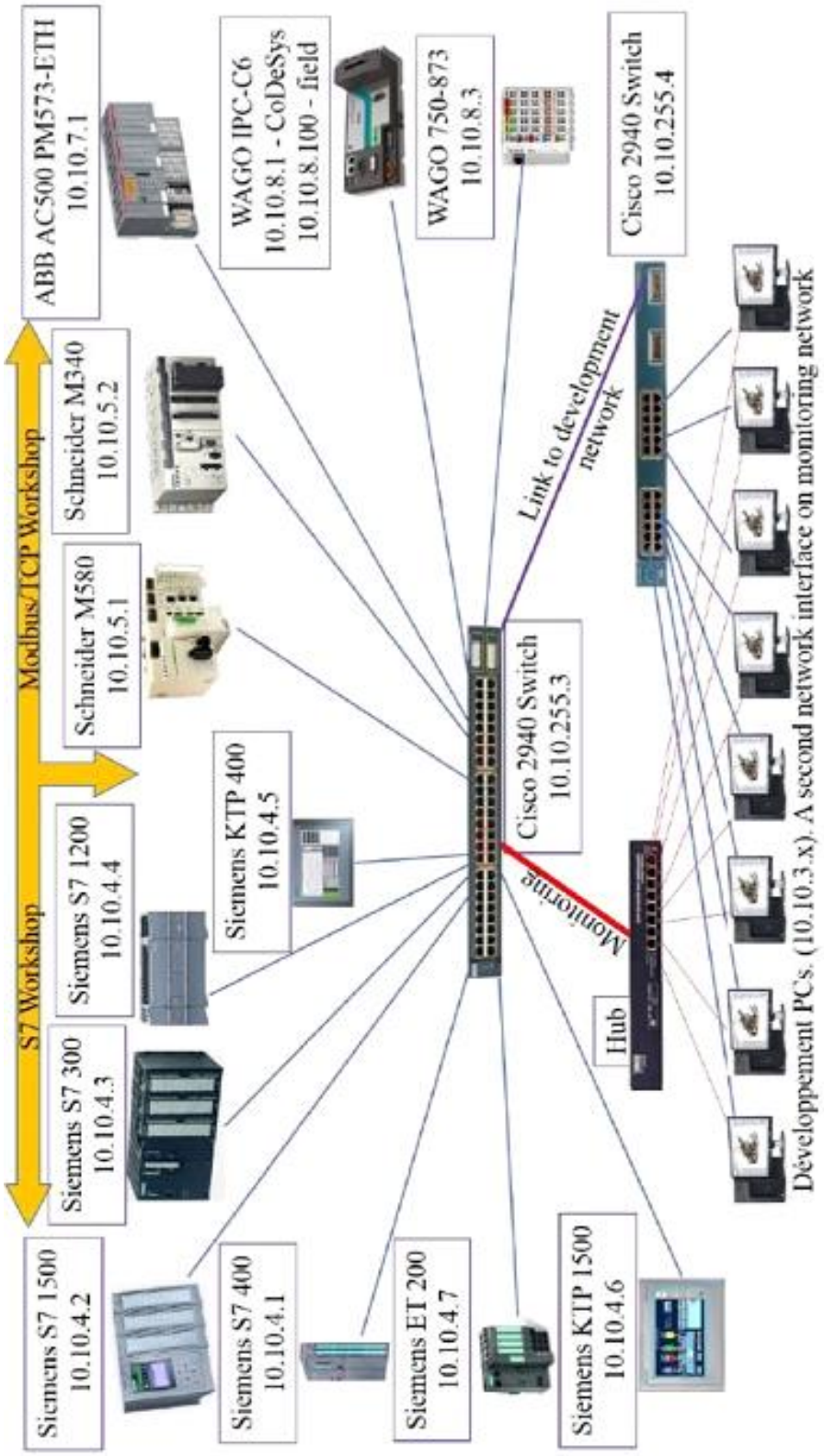


ABB AC500 PM573-ETH
10.10.7.1
WAGO IPC-C6
10.10.8.1 - CoDeSys
10.10.8.100 - field
WAGO 750-873
10.10.8.3
Cisco 2940 Switch
10.10.255.4
Modbus/TCP Workshop
Schneider M340
10.10.5.2
Schneider M580
10.10.5.1
Link to development network
Siemens KTP 400
10.10.4.5
Siemens S7 1200
10.10.4.4
Cisco 2940 Switch
10.10.255.3
Monitoring
S7 Workshop
Siemens S7 300
10.10.4.3
Hub
Siemens S7 1500
10.10.4.2
Siemens S7 400
10.10.4.1
Siemens ET 200
10.10.4.7
Siemens KTP 1500
10.10.4.6
Développement PCs. (10.10.3.x). A second network interface on monitoring network