

Industrial communication labs (Siemens)

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 constraints of the application will be met. Despite the fact that some of the protocols are Ethernet embedded (like GOOSE and Profinet I/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) Understand the communication between PLC and HMI/SCADA
- 2) Implement and observe communication flows
- 3) Analyze the flows, optimize the communication

1. 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.

1.1 Réseau : Le réseau 10.10.0.0/16 interconnecte différents éléments :

- les API ,
- les ordinateurs de « configuration » (10.10.3.x),
- les cartes « simulateurs de processus » (10.10.100.x, non visible sur le schéma général de l'Annexe A).
- les interfaces Homme-Machine (HMI).

Ces éléments sont interconnectés via deux switches CISCO.

Architecture de commande et architecture de supervision

Bien comprendre et préciser le fonctionnement global de l'architecture et de la boucle de commande. Cette architecture est constituée de :

Elément de l'architecture par groupe pour la commande	Interfaces
Un API destiné à recevoir la commande	- Entrées/Sorties analogiques et numériques - Carte réseau - Cas particulier : 10.10.4.1 : E/S déportées par l'interface ET200S de gauche via Profibus (celui sans adresse IP...)
Un « processus physique » représenté par une carte de simulation (via l'adresse 10.10.100.x) embarquée SMT32.	- Entrées/Sorties analogiques et numériques - Carte réseau
Une interface de commande à distance (GICS Tester) implémentée sur la machine de « configuration » (10.10.3.x)	- Carte réseau

QUESTION : Dessiner la boucle de commande avec le contrôleur, les actionneurs, les mesures, l'action de demande de vidange (client_request) et préciser les interfaces utilisées et les liens/modes de « connexions » (ex : réseau(x), lien direct E/S...)

Au-delà de la boucle de commande, l'architecture sert également à la configuration.

1.2 The first aspect will be to identify your computer (3.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...

1.3 Prise en main de votre espace de travail

Sur chaque ordinateur un compte local est ouvert pour les étudiants (ex MISTREA01, vous devez demander à l'enseignant quel est votre login et mot de passe sur votre machine, et bien le noter). Les logiciels utiles sont :

- Les logiciels pour configurer votre API : TIA Portal pour Siemens et Unity Pro XL pour Schneider
- Une interface de commande à distance GICS Tester

QUESTION : Quelle est l'utilité de l'interface de contrôle GICS Tester ?

1.4 Configuration de l'interface de commande à distance GICS Tester

Sur l'interface de contrôle GICS Tester (ordinateur de « configuration »), il faut configurer l'adresse IP de la carte de simulation que vous utilisez « target IP » : 10.10.100.x, sur le port 2015. Nous n'utiliserons pas les entrées et sorties analogiques (les valeurs numériques sont comprises entre 0 et 4095 pour des tensions comprises entre -10 et +10 V).

A partir de là, vous pouvez tester les sorties en les forçant par l'interface de contrôle GICS Tester et en observant leurs variations sur les interfaces physiques des cartes E/S.

NE PAS TOUCHER LES CABLAGES PHYSIQUES.

Si besoin, faites la correspondance via un tableau entre les numéros des E/S sur l'interface physique et les numéros sur l'interface de contrôle GICS Tester (décalage éventuel).

2. Configuration of the PLC

In the first part of the lab you have to build a simple SCADA system. The system schematics are presented in Figure 1.

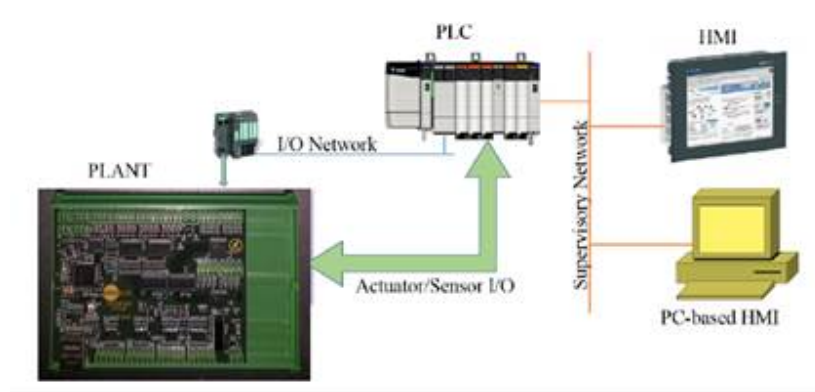


Figure 1. The simple SCADA system

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.

Work to do.

This job does not involve communication between various PLCs, but involves just your own PLC.

You'll have to:

- A. Write a very simple control program for the PLC (the objective of the lab is communication, not PLC programming).
- B. Set-up the communication between the SCADA and the PLC
- C. Check the communication flows using a network sniffer

2.1 Projet de configuration de votre API

Sur l'ordinateur de « configuration », vous allez configurer votre API (selon les API et les environnements, ces opérations peuvent être plus ou moins automatisées ...), l'objectif est que vous le fassiez pour bien voir et comprendre les différentes étapes. C'est une opération où il faut être rigoureux et précautionneux, choisir les bons éléments et bien les configurer, car ensuite les drivers correspondants seront envoyés à l'automate... et doivent correspondre aux éléments précis de l'architecture physique de l'automate. Il faut tout configurer y compris les unités que vous n'utiliserez pas (par exemple les E/S analogiques) car les configurations logicielle et matérielle doivent être conformes.

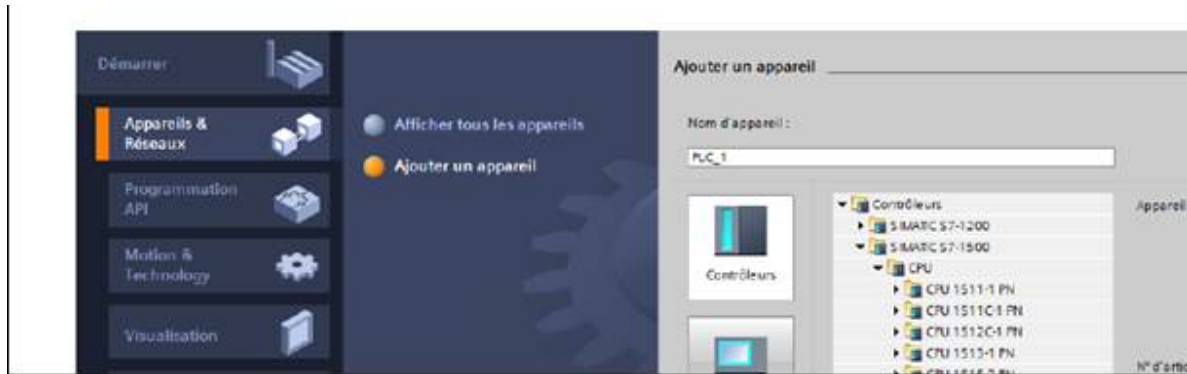
Pour les **API SIEMENS** : lancer « TIAPortal »

- Créez un nouveau projet
- « Configure a device » (si besoin, vous pourrez être amené à « Add new device », cela dépendra de votre automate et du CPU, certains automates sont plus ou moins « intégrés »)

Comparez votre automate avec les autres automates de la salle, le vôtre est-il plutôt intégré (intégrant CPU, module d'alimentation, E/S) ou est-il plutôt constitué d'éléments regroupés ?

- Sur certains automates, il faudra ajouter explicitement le module d'alimentation (Power module)
- Sur certains il faudra ajouter des modules d'entrées et/ou sorties analogiques et/ou numériques (AI, AQ, DI, DQ...)

Configuration de l'architecture physique de l'automate



La première étape consiste à créer un projet. Il faudra ensuite en fonction de l'automate choisir les différents éléments constitutifs de l'architecture physique :

- automate proprement dit,
- alimentation,
- interfaces d'entrées-sorties (I/O) analogiques et numériques,
- interfaces d'entrées-sorties déportées (I/O déportées), réseau de communication vers les I/O déportées.

En fonction des automates, l'alimentation et des I/O peuvent être internes ou externes, il peut y avoir des I/O déportées qui seront jointes par exemple via un réseau de terrain : Modbus, CAN, ...

Le point important ici est que l'on doit configurer une architecture conforme à l'architecture physique réelle des composants de la plate-forme. L'architecture logicielle sera ensuite « compilée » puis envoyée sur l'automate réel.

Une fois le projet ouvert, on peut « configure a device », puis « add new device » : il faut bien prendre la référence précise de l'automate utilisé, et des différents éléments physiques (alimentation, I/O, I/O déportées) sur les éléments eux-mêmes.

2.2 Configuration de l'interface réseau de l'automate, synchronisation d'horloge

L'automate utilisera au moins une interface réseau. Les automates ont en général une ou plusieurs interfaces réseau IP (RJ45) et une interface réseau Profibus sur les API Siemens (RS 485).

Bien regarder quelle(s) interface(s) réseau on va utiliser (Nota : pour le 10.10.4.1, il faudra configurer la connexion Profibus avec l'interface déportée ET200S).

L'adresse IP sera celle de l'API.

Pour cela « add new subnet ». Il faudra configurer la passerelle de sortie pour avoir accès à internet, cela permettra de se connecter sur le réseau (« use router »).

Vérifier par ailleurs que votre machine de « configuration » a bien accès au réseau (en testant un site externe par exemple).

Pour avoir une bonne synchronisation entre tous les API dans une architecture, il peut être

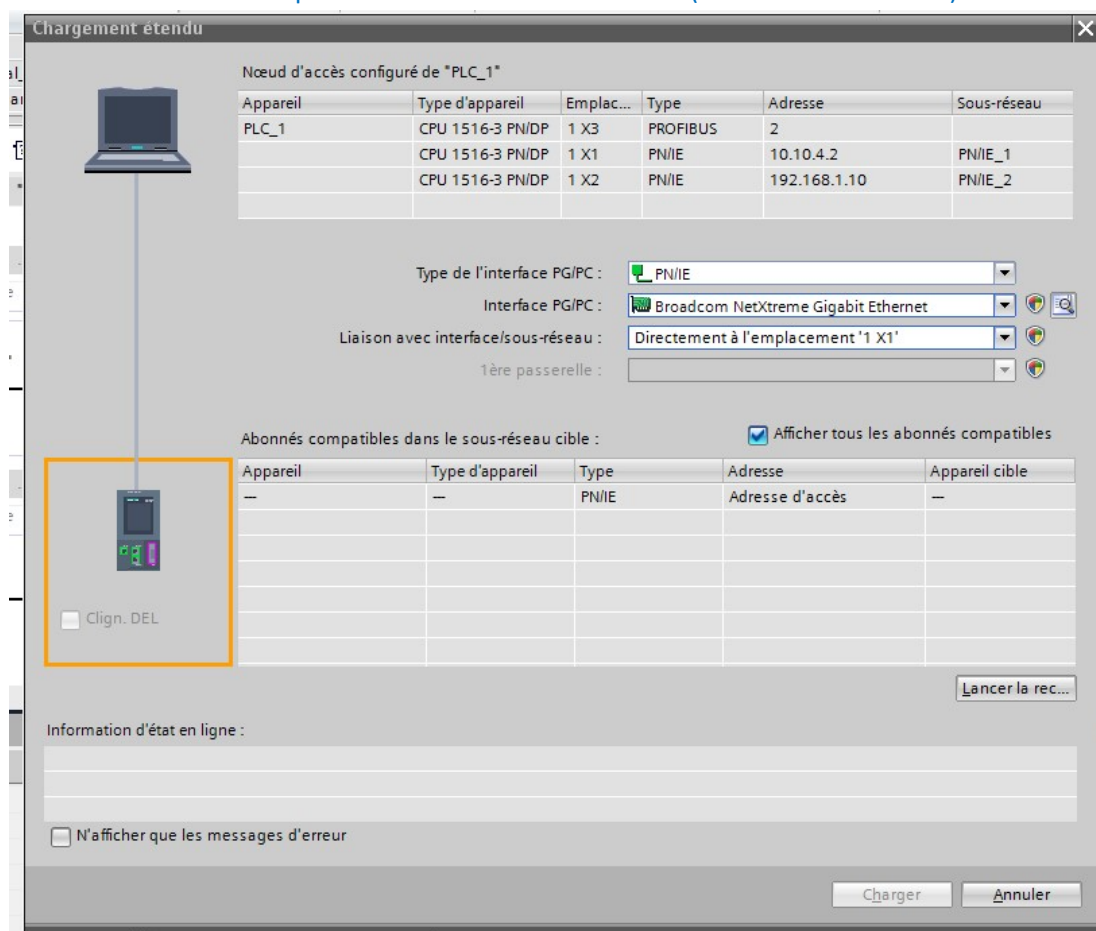
intéressant d'avoir une horloge de référence (protocole NTP, *Network Time Protocol*). On peut utiliser l'horloge de Grenoble INP (recherche sur internet), vérifiez que l'adresse est 147.171.64.9.

Nous terminerons la connexion de l'interface réseau en permettant l'accès web (webserver access => enable).

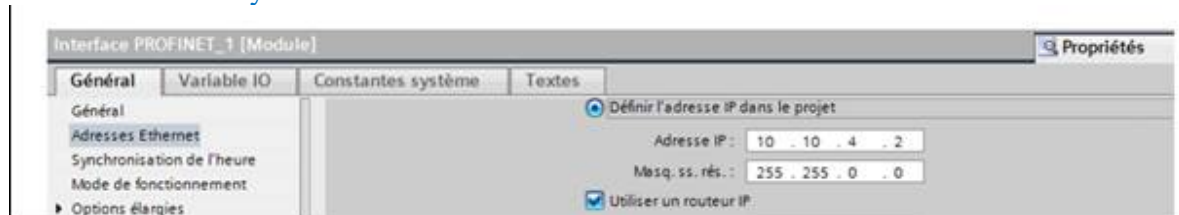
QUESTION : Pourquoi donner l'accès web ?

L'étape suivante consiste à configurer les interfaces réseaux (réseau externe, éventuellement réseau vers les I/O déportées) ainsi que les I/O. Avant de configurer les I/O, vérifier sur l'architecture physique de l'automate quelles sont les I/O effectivement câblées et utilisables.

- Il faut communiquer via la carte réseau Broadcom (voir dessin ci-dessous)



Il faut ensuite envoyer dans l'automate.



2.3 Configuration des variables

En « Siemens », les variables sont accessibles via le menu « General ». Les variables gérées sont de quatre types :

- Les sorties physiques (%Q ou %A en Allemand)
- Les entrées physiques (%I ou %E en Allemand)
- Les variables stockées en mémoire (%M)
- Les variables systèmes (%S)

Parfois les variables seront sur 16 bits (%IW), W pour Word (mot de 16 bits).

Dans cette partie, il faudra configurer les cinq entrées et sorties de notre système de cuve...

2.4 Compilation et envoi de la configuration à la CPU de l'automate

Il faut compiler votre projet (click droit sur le nom de l'automate), il faut compiler le hardware lorsque l'on fait une nouvelle configuration d'API et le software lorsqu'on développe de nouveaux programmes.

Lorsque la compilation est correcte, il faut envoyer le programme dans la CPU de l'automate (« Download to device »). Il faudra alors lancer une recherche (« start search ») pour trouver un API compatible avec l'architecture que nous avons configurée.

QUESTION : A la compilation nous pourrions avoir un « warning » de sécurité car nous avons donné l'accès au serveur web, pourquoi ce warning ?

2.5 Mode debugging et exécution

« Go online » sur Siemens permet de se mettre en mode débogage (mode qui permet de voir ce qui se passe réellement sur l'API lui-même pendant l'exécution des programmes).

2.6 Ecriture d'un programme, compilation, envoi, exécution

On peut programmer en ladder par exemple.

Pour programmer, sur Siemens, dans TIAPortal, dans « program blocks », faire « add new block ».

Faites votre programme respectant les spécifications du sujet (voir partie 3 page suivante).

QUESTION : Expliciter dans votre compte-rendu votre compréhension des spécifications, faire des choix technologiques (vannes normalement ouvertes, vannes normalement fermées...) puis expliquer votre programme. **N.B.** : Si vous êtes en retard, ne perdez pas trop de temps en séance, faites des programmes simples et testez-les sur la machine.

Il faut ensuite effectuer les actions suivantes :

- compilation,
- envoi à la CPU (« download to device », éventuellement si besoin choisir « continue without synchronisation »),
- Sur SIEMENS : Faire « start CPU » (dans les icônes tout en haut de l'écran) « load », « monitor » ; sur « Go Online », (en haut de l'écran), on doit pouvoir voir ce qui se passe dans l'automate.

3. The control problem.

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).

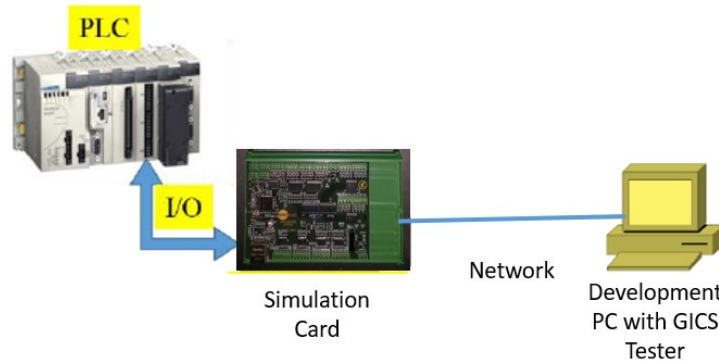


Figure 2. Hardware in the loop bench

The (very simple) control problem that you have to solve is the following. Consider a fluid tank (Figure 3) equipped with two level digital sensors (minimal and maximal levels) and two actuators (the two valves). There is also an external signal (“client request” for fluid supply). Your program has then three digital inputs and two digital outputs.

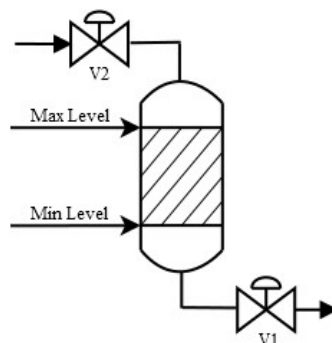


Figure 3. A simple tank

Control specification.

Your main objective is to keep the fluid level between the min and max levels. You’ll also serve the client requests for the fluid supply. The control logic specification is:

- A. If the tank level is between min and max and a client request is raised open the valve V1 as long as the client request is active.
- B. If the tank level reaches level min (that will necessary happen when V1 is open) close V1 and open V2 until the tank level reaches level max. When the tank level reaches max close V2.

OPTIONAL: Supplementary diagnostic tasks: if level min and max are simultaneously activated (sensors failure) or level min is active raise an error (set an error bit in the PLC memory) write the error code (0x01) in a memory word and wait for an external error acknowledgement before going back to the normal state.

4. 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).

HMI creation.

S7 communications setup

For the S7 communication you have to choose “Siemens” and “IP-ISO-S7” for the “Network” creation. Choose whatever name you want for the network and let the other values to default.

For the “Node” configuration choose “S7-300/400” for “Equipment Type” and add the IP number of your equipment into “Network address”. Let the other parameters to default.

Finally, into the “Frame” configuration you have to choose the type of variables you want to retrieve (bit/byte/word/etc) and the variables addresses into the PLC memory. Whatever the type of your variables is, the S7 driver from PCVue is able to transfer only WORDs. It means that, even if you declare your variable as a bit (%M0.0 for example), PCVue will still retrieve at least a WORD (16 bits) per frame (for the example, the bit you want will be the bit 0 of %MW0).

Then you have the choice between six types of variables addresses:

- 1) DBW: Data Block Words. Data Blocks are Step 7 data structures of free type. This choice let you to retrieve a PLC structured variable WORD by WORD. It is Siemens preferred data access because it does not rely on memory addresses, but only on variables names.
- 2) MW: memory words. Access to %MW addresses
- 3) EW: input memory addresses. Direct access to PLC inputs (E states for “Ein” meaning Input in German). Depending on PLC model and configuration, EW addresses access may be protected.
- 4) AW: output memory addresses. Direct access to PLC outputs (A states for “Aus” meaning Output in German). Depending on PLC model and configuration, AW addresses access may be protected.
- 5) ZW: counters memory addresses. Direct access to PLC counters (Z states for “Zähler” meaning Counter in German). Do not concern this lab.
- 6) Information: four bytes containing PLC CPU status information.

The choice of the type and the address of the variable depends on you and on your programming. The value of an input (for example) may be copied in a DB or a memory location. Then the access to the value of the input is possible by any of the three first methods (DBW, MW or EW).

5. Communication analysis.

Using Wireshark software on your computer you’ll intercept the traffic between the PLC and the SCADA. You’re asked to identify the flows (data exchanges) corresponding to every logical flow (variable exchange) in your system. Caution: when you intercept the traffic you’ll receive ALL the traffic from the industrial network of the lab, meaning also the traffic from the other groups. You’ll need to filter the packets you’ll receive.

QUESTIONS : Sur la plate-forme, les éléments sont interconnectés via deux switches CISCO.

Comment fonctionne un switch ? Quel est son rôle ?

L'architecture décrite Annexe A propose un hub avec un lien de monitoring sur lequel sont reliées toutes les machines de supervision (10.10.3.x).

Comment fonctionne un hub ? Quel est le rôle d'un hub de manière générale ?

Quel est le rôle du hub dans ce schéma ?

Pour sortir sur internet, ce réseau dispose d'une passerelle (gateway), non visible sur le schéma général. Cette passerelle utilise l'adresse IP généralement utilisée pour les passerelles.

Quelle est donc cette adresse ?

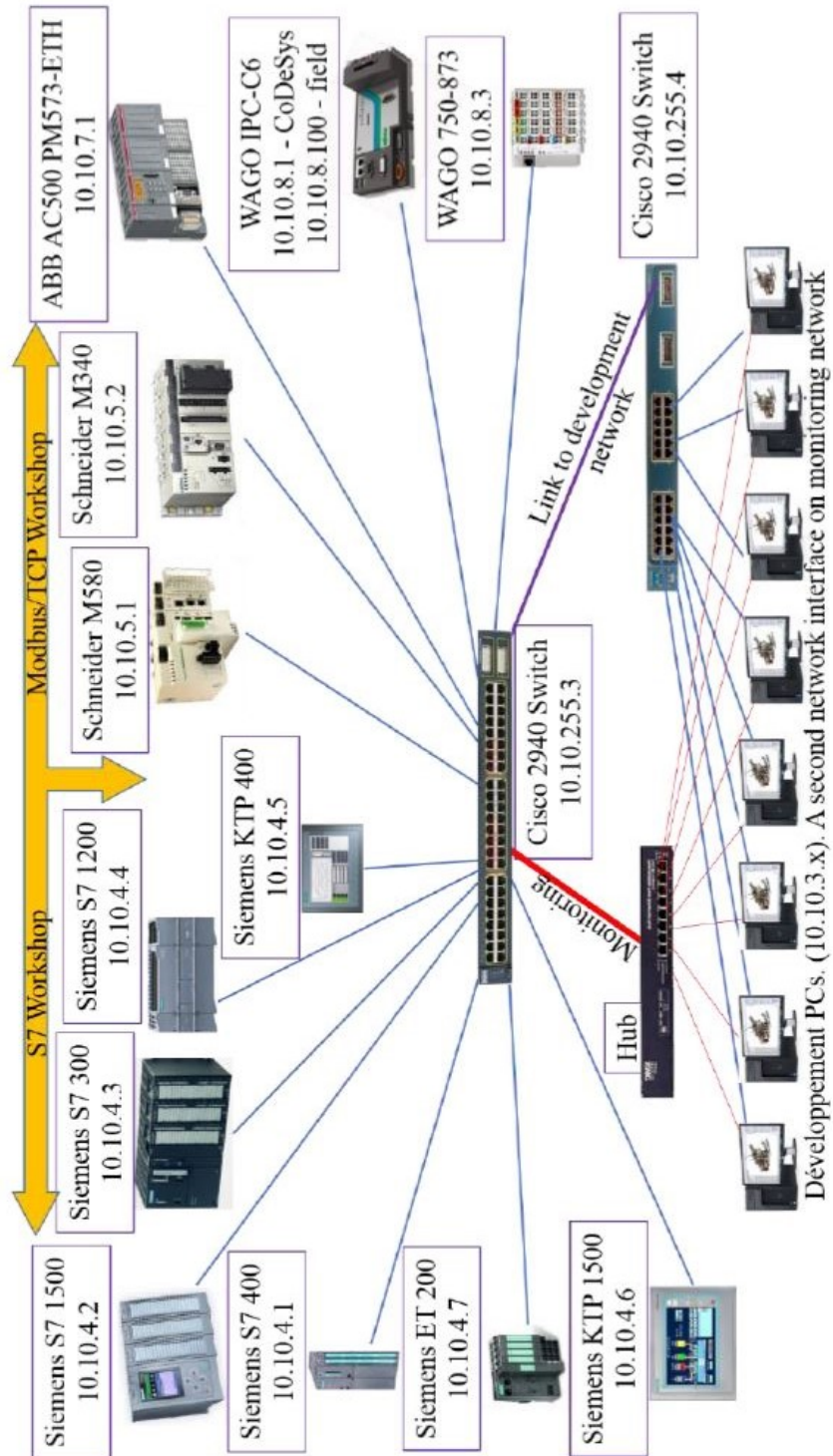
6. Conclusion

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



APPENDIX B.

ASPECT WIRESHARK

Voici un exemple de ce que l'on peut observer grâce à Wireshark, sur l'adresse 10.10.4.3

Capturing from Connexion au réseau local 2 [Wireshark 1.12.3 (v1.12.3-0-gbb3e9a0 from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: ip.addr==10.10.4.3 Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
63154	3930.027850	10.10.3.5	10.10.4.3	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
63155	3930.029501	10.10.4.3	10.10.3.5	S7COMM	109	ROSCTR:[Userdata] Function:[Push] -> [Prograi
63156	3930.029725	10.10.3.5	10.10.4.3	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
63157	3930.031469	10.10.4.3	10.10.3.5	S7COMM	103	ROSCTR:[Userdata] Function:[Push] -> [Prograi
63158	3930.031693	10.10.3.5	10.10.4.3	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
63164	3930.094352	10.10.4.3	10.10.3.5	TCP	60	102->57187 [ACK] Seq=664824 Ack=361325 win=40
63165	3930.132271	10.10.3.5	10.10.4.3	S7COMM	113	ROSCTR:[Userdata] Function:[Request] -> [Proi
63166	3930.135504	10.10.4.3	10.10.3.5	S7COMM	87	ROSCTR:[Userdata] Function:[Response] -> [Pri
63167	3930.135729	10.10.3.5	10.10.4.3	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
63168	3930.136290	10.10.3.5	10.10.4.3	S7COMM	115	ROSCTR:[Userdata] Function:[Request] -> [Proi
63169	3930.139381	10.10.4.3	10.10.3.5	S7COMM	127	ROSCTR:[Userdata] Function:[Push] -> [Prograi
63170	3930.139605	10.10.3.5	10.10.4.3	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
63171	3930.143503	10.10.4.3	10.10.3.5	S7COMM	87	ROSCTR:[Userdata] Function:[Response] -> [Pri
63172	3930.143727	10.10.3.5	10.10.4.3	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
63173	3930.145468	10.10.4.3	10.10.3.5	S7COMM	109	ROSCTR:[Userdata] Function:[Push] -> [Prograi
63174	3930.145691	10.10.3.5	10.10.4.3	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
63175	3930.146469	10.10.4.3	10.10.3.5	S7COMM	103	ROSCTR:[Userdata] Function:[Push] -> [Prograi
63176	3930.146645	10.10.3.5	10.10.4.3	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
63180	3930.194365	10.10.4.3	10.10.3.5	TCP	60	102->57187 [ACK] Seq=665067 Ack=361480 win=40
63183	3930.257323	10.10.3.5	10.10.4.3	S7COMM	115	ROSCTR:[Userdata] Function:[Request] -> [Proi
63184	3930.260482	10.10.4.3	10.10.3.5	S7COMM	87	ROSCTR:[Userdata] Function:[Response] -> [Pri
63185	3930.260705	10.10.3.5	10.10.4.3	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
63186	3930.262506	10.10.4.3	10.10.3.5	S7COMM	109	ROSCTR:[Userdata] Function:[Push] -> [Prograi
63187	3930.262733	10.10.3.5	10.10.4.3	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
63188	3930.264468	10.10.4.3	10.10.3.5	S7COMM	103	ROSCTR:[Userdata] Function:[Push] -> [Prograi
63189	3930.264692	10.10.3.5	10.10.4.3	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]
63191	3930.294408	10.10.4.3	10.10.3.5	TCP	60	102->57187 [ACK] Seq=665204 Ack=361562 win=40
63197	3930.366341	10.10.3.5	10.10.4.3	S7COMM	115	ROSCTR:[Userdata] Function:[Request] -> [Proi
63198	3930.370519	10.10.4.3	10.10.3.5	S7COMM	87	ROSCTR:[Userdata] Function:[Response] -> [Pri
63199	3930.370744	10.10.3.5	10.10.4.3	COTP	61	DT TPDU (0) [COTP fragment, 0 bytes]

Frame 63120: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0

Ethernet II, Src: Siemens_le:f7:8a (28:63:36:1e:f7:8a), Dst: Broadcom_4d:4e:ae (00:0a:f7:4d:4e:ae)

Internet Protocol Version 4, Src: 10.10.4.3 (10.10.4.3), Dst: 10.10.3.5 (10.10.3.5)

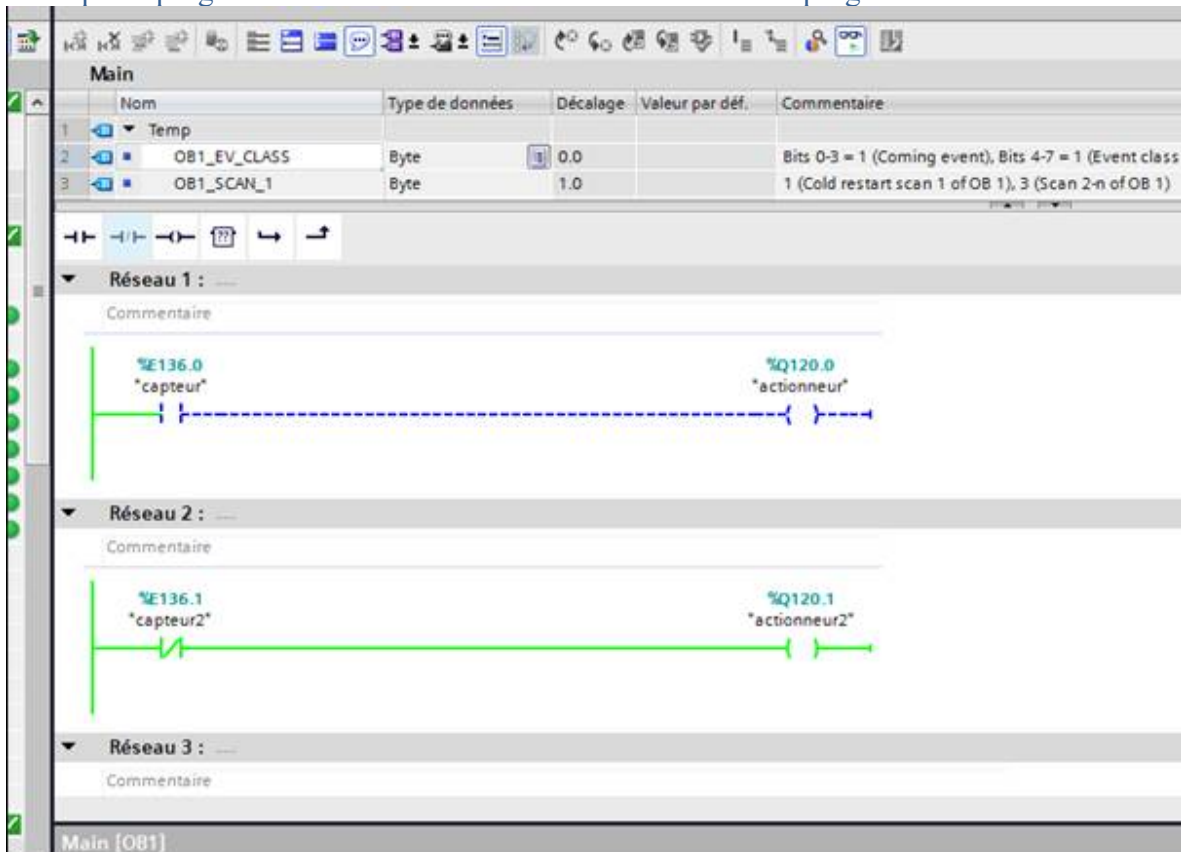
Transmission Control Protocol, Src Port: 102 (102), Dst Port: 57185 (57185), Seq: 15818, Ack: 18183, Len: 27

APPENDIX C. Documentation technique Réseau & IHM (SIEMENS)

Mise en route du programme (pour visibilité sur le PC de programmation)

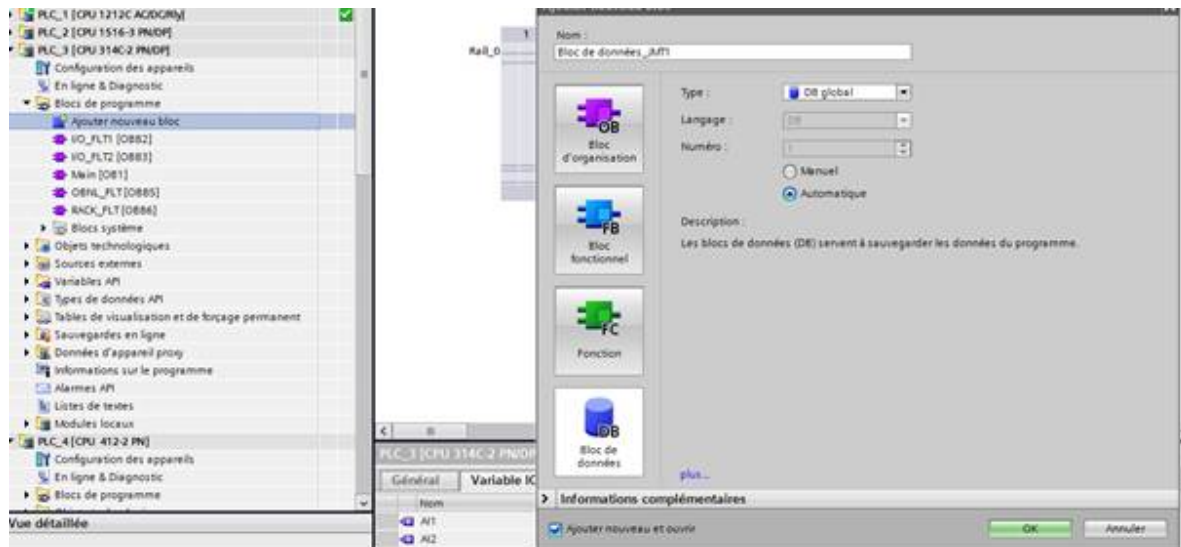


Il faut cliquer sur la lunette ci-contre
Pour que le programme soit en mode visible sur l'ordinateur de programmation

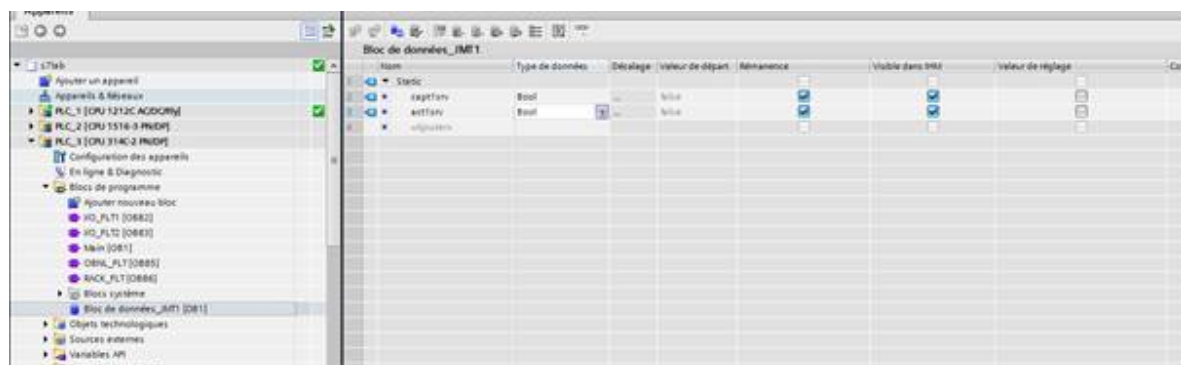


IHM (Siemens)

- Pour l'automate Siemens 1200, utiliser l'interface IHM « physique » 10.10.4.5
- Pour les automates Siemens 1500, utiliser les interfaces IHM « physiques » 10.10.4.6 et 10.10.4.7
- Insérez le modèle de l'IHM choisi dans TIA Portal.



Ajouter Nouveau Bloc puis Bloc de Données



Dans le bloc de données il faut rendre capt1srv et act1srv visibles sur l'IHM