

Asean-factori

3.1 Networks- complement S7-Modbus

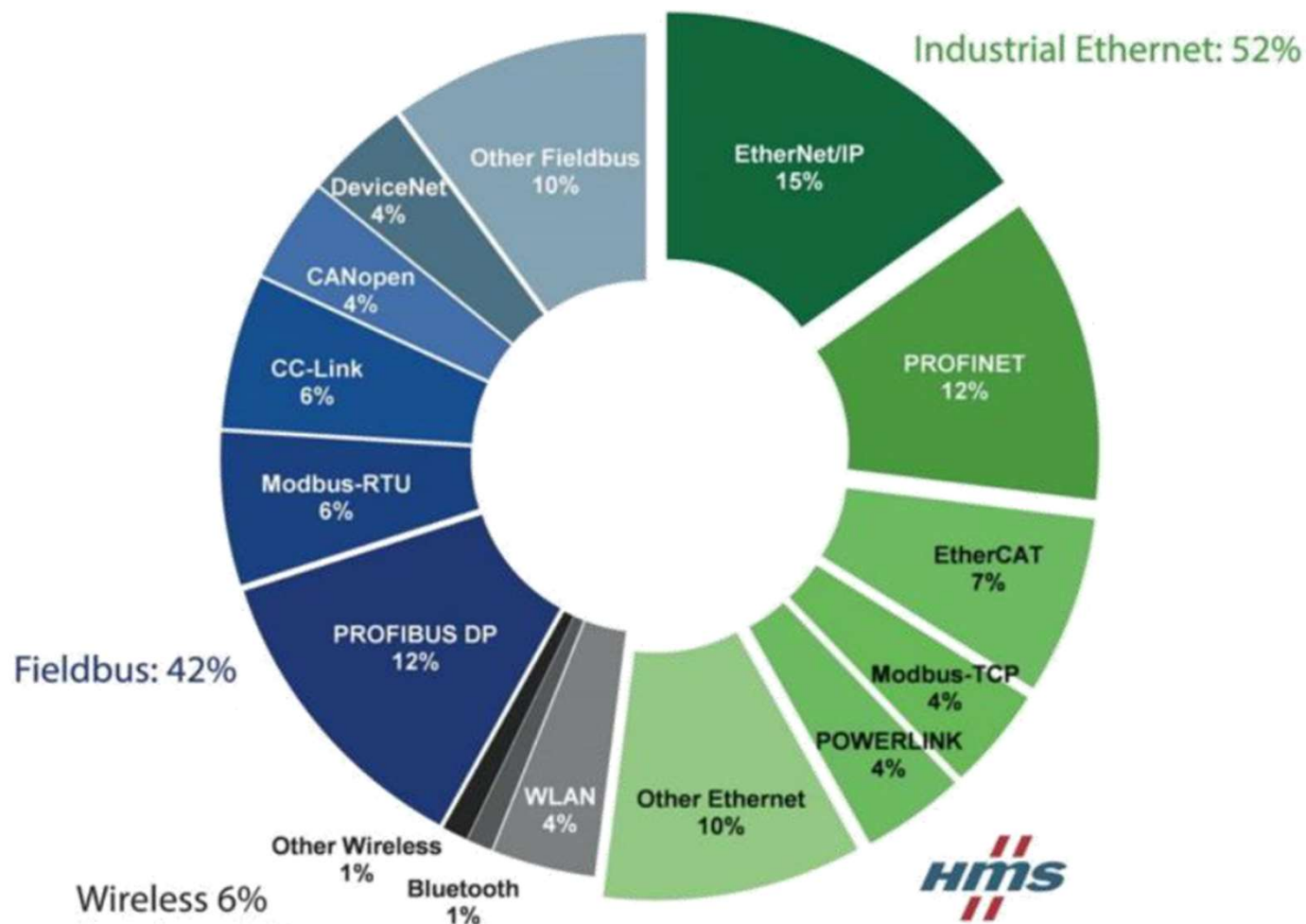
Jean-Marc THIRIET

jean-marc.thiriet@univ-grenoble-alpes.fr
<http://www.gipsa-lab.grenoble-inp.fr/~jean-marc.thiriet/asean/asean.html>



<http://www.gipsa-lab.grenoble-inp.fr/~jean-marc.thiriet/asean/asean.html>

Distribution Fieldbus Networks vs. Industrial Ethernet (2018, from Stormshield)



S7 Protocol

S7 Protocol

INTRODUCTION

SIEMENS

- S7 ou S7comm
- Protocole utilisé par Siemens depuis 1994 avec la famille de produits Simatic S7
- Utilisé par les automates de la famille S7-300/400



S7-300 S7-400 S7-1200 S7-1500

STORMSHIELD

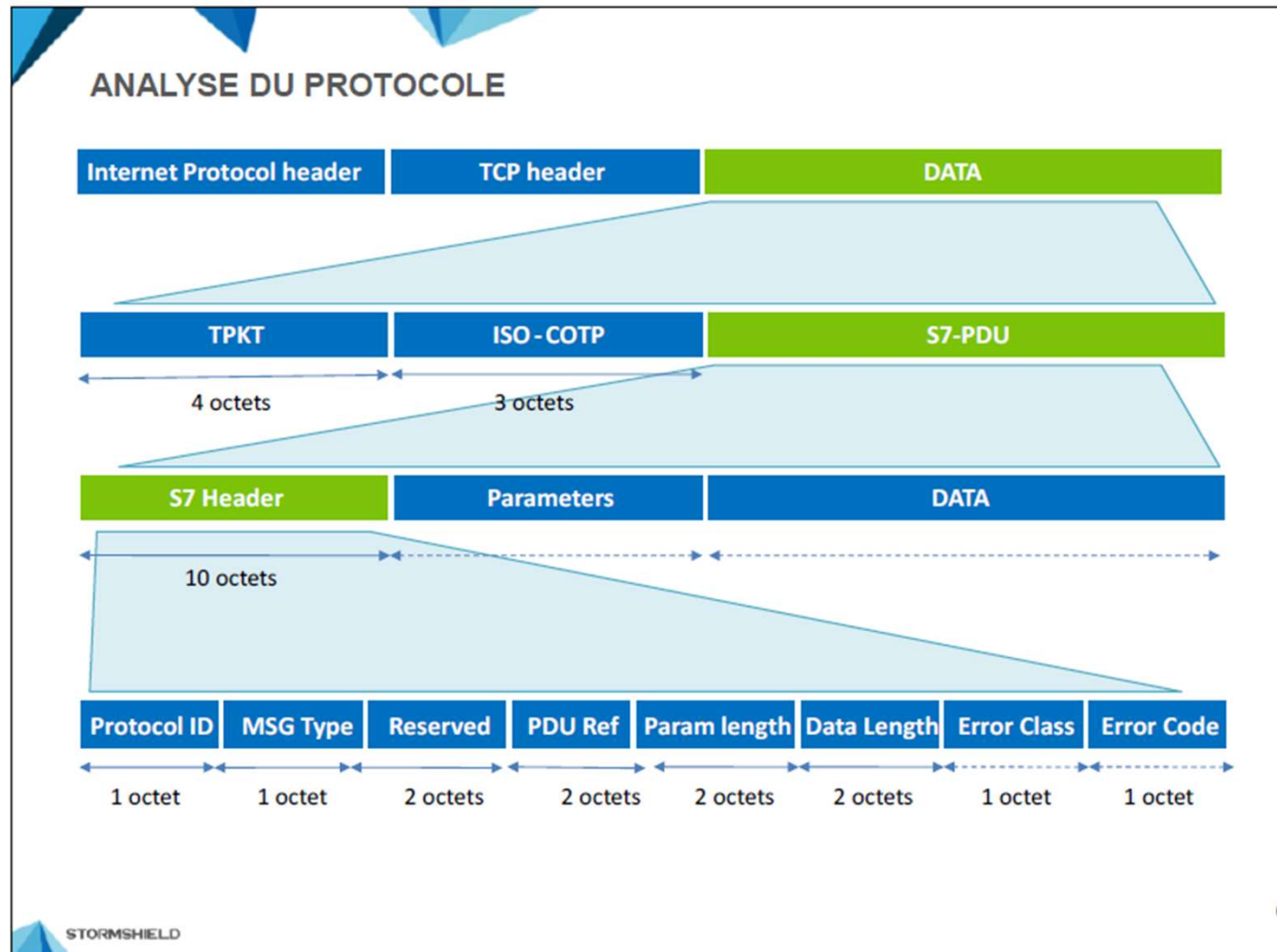
3

- Between PLCs
- Between PLCs and SCADA or HMI

INTRODUCTION		
	Couches OSI	Protocoles
7	Application	S7
6	Présentation	S7
5	Session	S7
4	Transport	ISO-on-TCP (RFC 1006) : • COTP • TPKT • TCP port 102
3	Réseau	IP
2	Liaison	Ethernet
1	Physique	Ethernet

- TCP/IP implementation of the S7 protocol is based on the ISO-COTP transport service.
- COTP (Connection Oriented Transport Protocol) is an ISO protocol, connected and oriented protocol.
- TPKT allows to emulate the services of the ISO layers above TCP. These services are necessary for the proper functioning of COTP

Protocol analysis



Header S7

Protocol ID : Constante, toujours égal à 0x32

MSG Type : Type de message général :

- *0x01-Job Request* : requête envoyée par le maître (lecture, écriture,...).
- *0x02-Ack* : accusé de réception simple envoyé sans champ de données.
- *0x03-Ack-Data* : accusé de réception avec champ de données, contient la réponse à une demande de travail.
- *0x07-Userdata* : extension du protocole d'origine, le champ de paramètre contient alors le type de requête. Ce type est utilisé pour la programmation, le débogage, etc.

Reserved : Toujours avec la valeur 0x0000, probablement ignoré.

PDU Ref. : Généré par le maître, incrémenté à chaque nouvelle transmission, utilisé pour lier les réponses à leur requêtes.

Param Length : Longueur du champ de paramètre.

Data Length : Longueur du champ de données.

ANALYSE DU PROTOCOLE

- **ID:** toujours 0x32
- **MSG Type:**
 - 0x01 Job
 - 0x02 Ack (acquittement normal, sans donnée)
 - 0x04 Ack Data (acquittement avec données, par exemple suite à une requête de lecture)
 - 0x07 User Data (extension du protocole original, utilisé pour programmer des périphériques)
- Dans le cas d'un MSG Type Job, on trouvera dans les paramètres le champ Job Request décrivant ce que le destinataire doit faire.

Code	Job request
0X00	CPU Services
0XF0	Setup communication
0X04	Read Variable
0X05	Write variable
0X1A	Request download
0X1B	Download Block
0X1C	Download ended
0X1D	Start Upload
0X1E	Upload
0X1F	End Upload
0X28	PLC Control
0X29	PLC Stop

ANALYSE DU PROTOCOLE

The screenshot displays the Wireshark interface with a capture of S7COMM traffic. The packet list shows several S7COMM packets between 192.168.1.10 and 192.168.1.40. The packet details pane shows the structure of an S7COMM packet, including the header and the function code field. The function codes management panel is open, showing a list of predefined operations and their corresponding function codes.

Code	Function	Action
0	JOB	
1	USERDATA Group 0 : Mode transition	
2	USERDATA Group 1 : Programmer command	
3	USERDATA Group 2 : Cyclic data	
4	USERDATA Group 3 : Blocks functions	
5	USERDATA Group 4 : CPU functions	
6	USERDATA Group 5 : Security	
7	USERDATA Group 6 : Programmable Block Communication (Send/receive)	
8	USERDATA Group 7 : Time functions	

8

Modbus

MODBUS

- Le protocole **MODBUS** est une structure de messagerie créée par **MODICON** en **1979** pour connecter des automates à des outils de programmation.
- Ce protocole est de nos jours largement utilisé pour établir des communications de type **maître/client** vers **esclaves/serveurs** entre **équipements intelligents**.



Variantes de MODBUS

	Il existe 3 variantes de MODBUS:	
1)	MODBUS liaison série: sur liaison série fonctionnant de 1200 à 56 kbits/s avec une méthode d'accès maître/esclave .	

- 2) **MODBUS + :** est un bus fonctionnant à **1 Mbit/s** basé sur une méthode d'accès par **anneau à jeton**.
- 3) **Ethernet TCP/IP MODBUS:** utilise le protocole **TCP/IP** et **Ethernet 10 Mbit/s** ou **100 Mbits/s** pour transporter une trame MODBUS.

MODBUS liaisons série

- **RS232:**

- Liaison point à point par connecteur **SUB-D 25 broches**.
- Distance **< 15 mètres**, débit **< 20 kbits/sec**.

- **RS422A:**

- Bus multipoint **full duplex** (bidirectionnel simultané) sur **4 fils**.
- Distance maxi **1200 mètres** à **100 kbits/sec**.
- **2 fils** en émission, **2 fils** en réception.

- **RS485:**

- Bus multipoint **half duplex** (bi directionnel alterné) sur **2 fils**.
- Mêmes caractéristiques que **RS422A** mais sur **2 fils**.

MODBUS série et modèle ISO (3 layers 1, 2, 7)

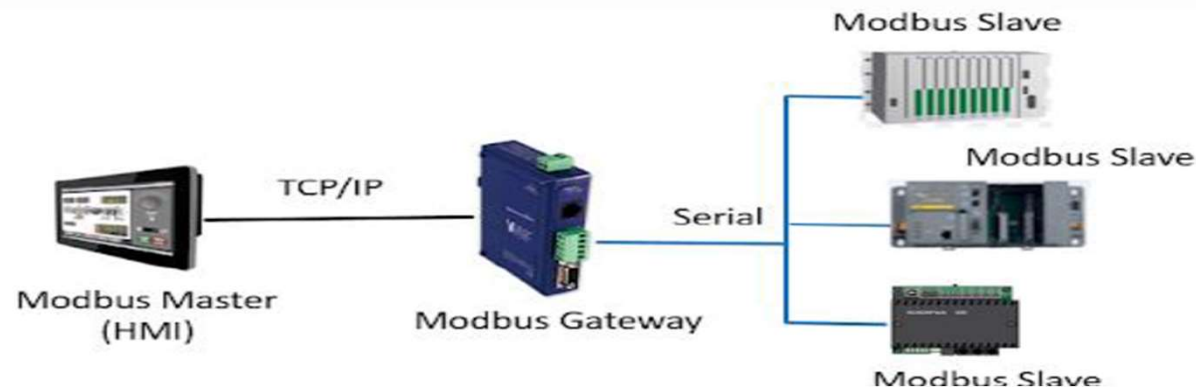
7	Application	<i>Modbus</i>
2	Link	<i>Master / Slave</i>
1	Physic	<i>RS485</i>

MODBUS + et modèle ISO (3 layers 1, 2, 7)

7	Application	<i>Modbus</i>
2	Link	<i>Token Ring</i>
1	Physic	<i>RS485</i>

Functioning

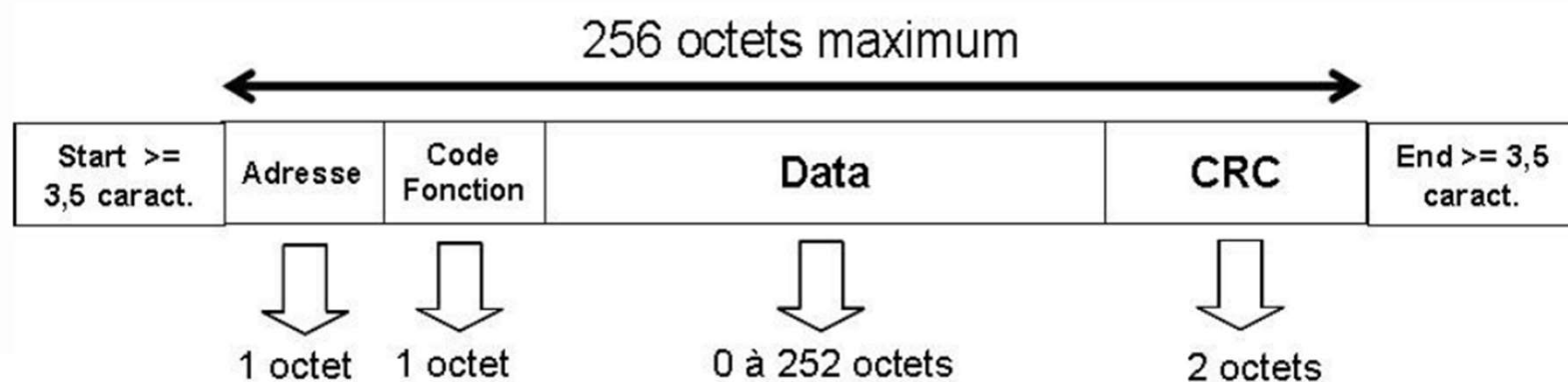
- MODBUS fonctionne en mode **maitre/esclave**.
- un maitre va interroger un ou plusieurs dispositifs fonctionnant comme esclaves.
- Le maitre peut communiquer avec un nombre allant jusqu'à 247 esclaves.



Frame structure

Chaque trame est constituée de:

- **L'indentifiant de l'esclave** codé sur 8 bits.
- **Le code de fonction** (lecture, ecriture ...): codé sur 8 bits.
- **Les données**: elles sont codées sur 253 octets.
- **Le CRC**: codé sur 16 bits pour s'assurer du bon envoi des données.



Ethernet TCP/IP MODBUS

- Quand ces protocoles sont associés à Modbus, une information additionnelle est affectée à la donnée par TCP (appelée « MBAP header »). Ensuite IP la place dans un paquet (ou datagramme) et la transmet.
- TCP doit établir une connexion avant de transmettre la donnée. Le maître (client) établit alors une connexion avec l'esclave (serveur). Une fois cette connexion est établie, le serveur répond aux demandes du client, jusqu'à fermeture de la connexion par le client

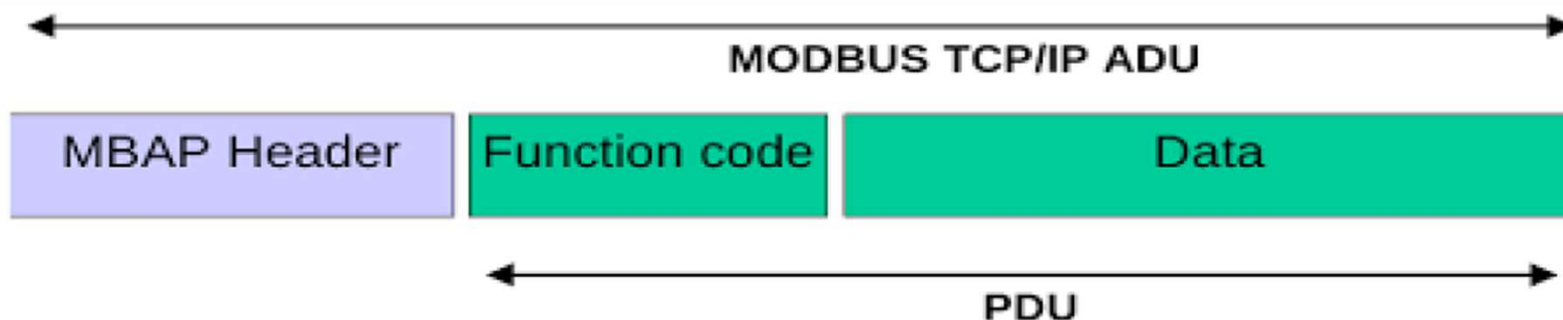
Ethernet TCP/IP MODBUS et modèle ISO (7 layers, from 1 to 7)

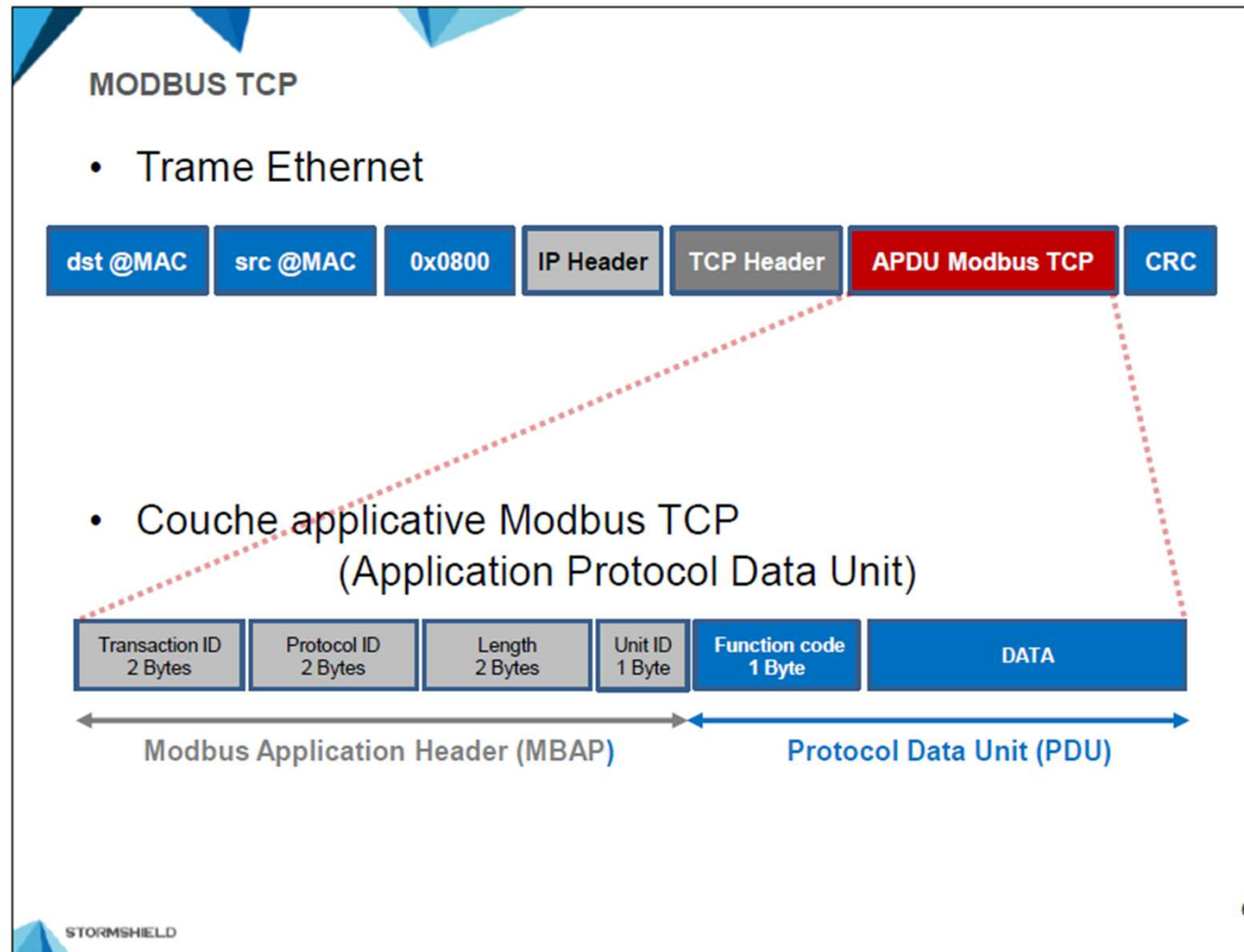
7	Application		Modbus
6	Présentation		
5	Session		
4	Transport		TCP
3	Réseau		IP
2	Liaison	<i>CSMA / CD</i>	
1	Physique	<i>ETHERNET</i>	

Le mode TCP



- Une extension du protocole MODBUS récemment développée permettant le fonctionnement sur des réseaux TCP/IP.
- L'envoi des données débute avec un identifiant de transaction du protocole à 2 octets.
- L'absence de l'octet CRC vue que la couche TCP effectue déjà l'opération de recherche des erreurs.





MODBUS TCP

- Le modèle de données MODBUS contient quatre types de base (blocs de données) :
 - Discrete Inputs (bits d'entrées)
 - Coil (bits de sorties, bits internes)
 - Input Registers (mots d'entrées, mots diagnostic)
 - Holding Registers (mots de sorties, mémoire interne, registres, compteurs, ...)

Data type	Format	Access
Discrete Inputs	Single bit	Read only
Coil	Single bit	Read / Write
Inputs Registers	16-bit word	Read only
Holding Registers	16-bit word	Read / Write



7

Exemples d'utilisation :

- **Discrete input** : lire une valeur qui peut être ON ou OFF (interrupteur à 0 ou 1),
- **Coil** : passer un interrupteur de la valeur OFF à la valeur ON,
- **Input registers** : lire la valeur (entière) de la température d'un thermomètre,
- **Holding registers** : définir la température d'une cuve à 200°C

MODBUS TCP

- Code de fonction (avec vue sur firewall SNS)

Function code	Size	Description
01	1	Read coil
02	1	Read discrete inputs
05	1	Write single coil
15	1	Write multiple coils
03	16	Read holding registers
04	16	Read input registers
06	16	Write single registers
16	16	Write multiple registers
23	16	Read/Write multiple registers

MODBUS function codes management

PUBLIC OPERATIONS

Searching... X Modify write operations Modify all operations

Code	Function	Action	Type
1	Read Coils	Scan	Reading
2	Read Discrete Inputs	Scan	Reading
3	Read Holding Registers	Scan	Reading
4	Read Input Register	Scan	Reading
5	Write Single Coil	Scan	Writing
6	Write Single Register	Scan	Writing
7	Read Exception Status	Block	Reading
8	Diagnostic	Block	Reading
11	Get Com Event Counter	Block	Reading
12	Get Com Event Log	Block	Reading
15	Write Multiple Coils	Scan	Writing
16	Write Multiple Registers	Scan	Writing
17	Report Slave ID	Block	Reading
20	Read File Record	Scan	Reading
21	Write File Record	Scan	Writing
22	Mask Write Register	Scan	Writing

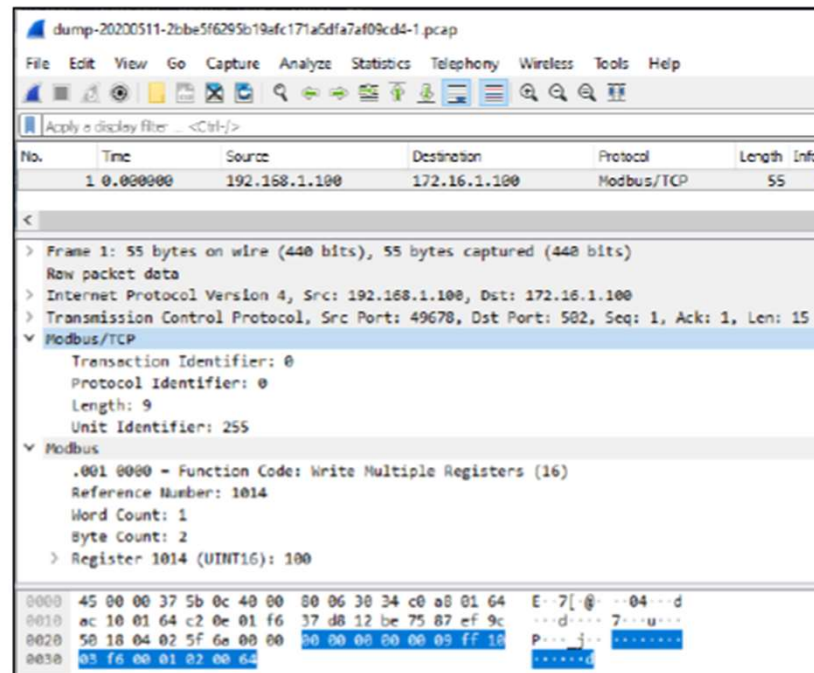
5 blocked on 19

8

Certains codes sont manquants (9,10,13,14, etc.), ils sont réservés.

MODBUS TCP

- Wireshark décode complètement Modbus TCP



10

STORMSHIELD

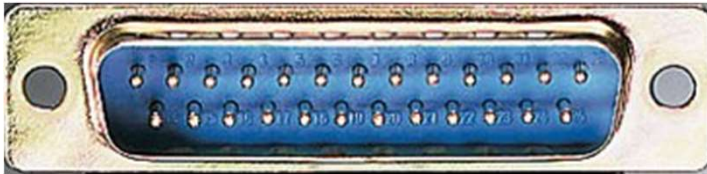
Ethernet: Pourquoi?

- Coût réduit, transmission possible par câble RJ45, fibre optique ou sans fil. Débit allant jusqu'à 100Mbits/s. Transmission du type Full Duplex.
- Pour l'intelligence embarquée, Ethernet rend possible l'intégration de services TCP/IP pour la configuration, le diagnostic, les alertes...
- Lien simplifié vers IHM, MES, ERP, SCADA...
- Echanges entre équipements de modèles différents.
- Transmission de l'alimentation via le câble.

Couche physique

- **Médium:**
Typiquement une paire torsadée blindée.
- **Topologie:**
bus avec dérivations et terminaisons de fin de ligne.
- **Longueur maximale:**
1.3 km sans répéteur.
- **Débit:**
9600 ou 19200 bits/s (56 kbits/s sur certains produits).
- **Nombre maxi d'équipements:**
 $32 = 1 \text{ maître} + 31 \text{ esclaves}$.

Connectiques



DB-25 ou DB-15

(série RS422/485/232)



RJ 45

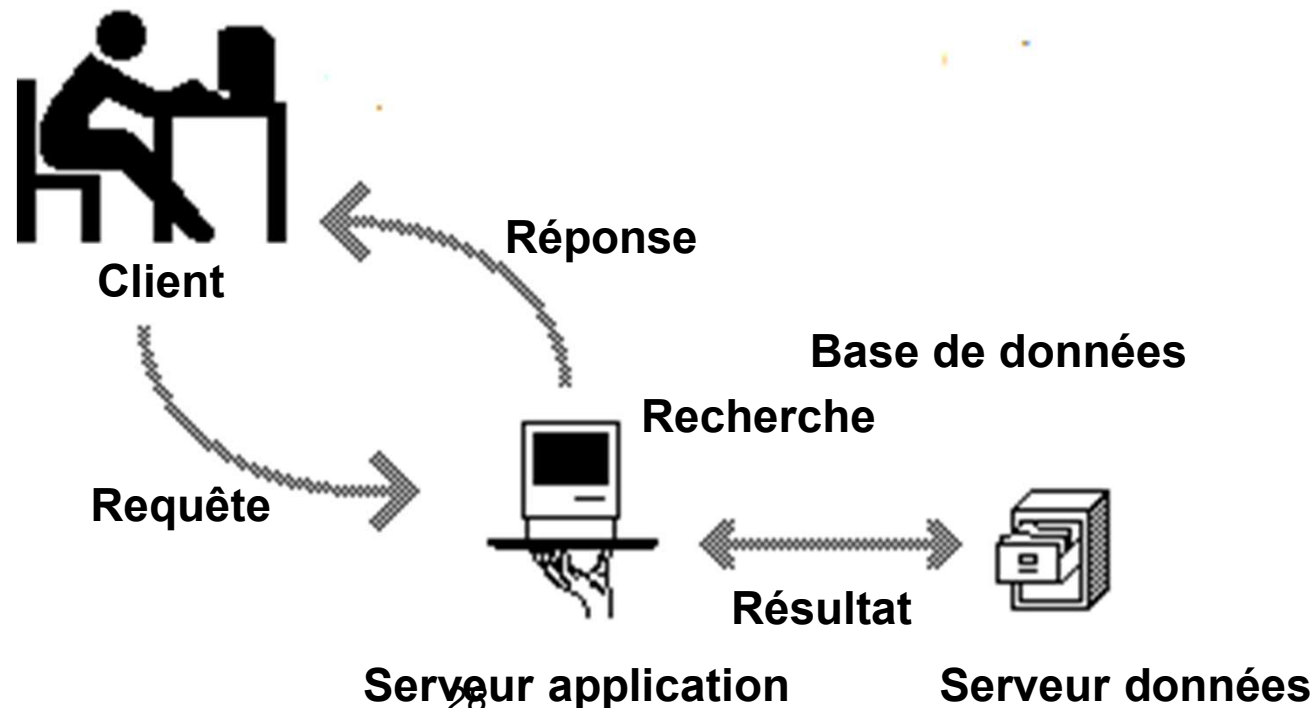
(Ethernet)

Couche de liaison : Caractéristiques générales

- **Méthode d'accès au médium: Maître / esclave.**
- **Méthode de transmission: Client / serveur.**
 - Le maître est client, l'esclave est serveur.
 - L'échange de données nécessite un programme applicatif.
- **Taille maxi des données utiles: 128 mots (256 octets).**

Couche de liaison : Méthode client-serveur

- Le **CLIENT** est une entité demandant un service sur le réseau (une requête). Une programmation des requêtes est nécessaire. Tous les clients sont **actifs**.
- Le **SERVEUR** est l'entité qui répond à la demande d'un client. Le serveur est complètement **passif**.



Couche de liaison : Types de trames

La trame MODBUS existe en 2 versions :

- **Mode ASCII:**

Chaque octet de la trame est transmis sous la forme de 2 caractères ASCII.

- **Mode RTU (Remote Terminal Unit/Unité Terminale Distant):**

Chaque octet de la trame est transmis sous la forme de 2 caractères hexadécimaux.

- Le principal avantage du mode RTU est qu'il transmet plus rapidement les informations.

Exemples de produits MODBUS



Répartiteur MODBUS

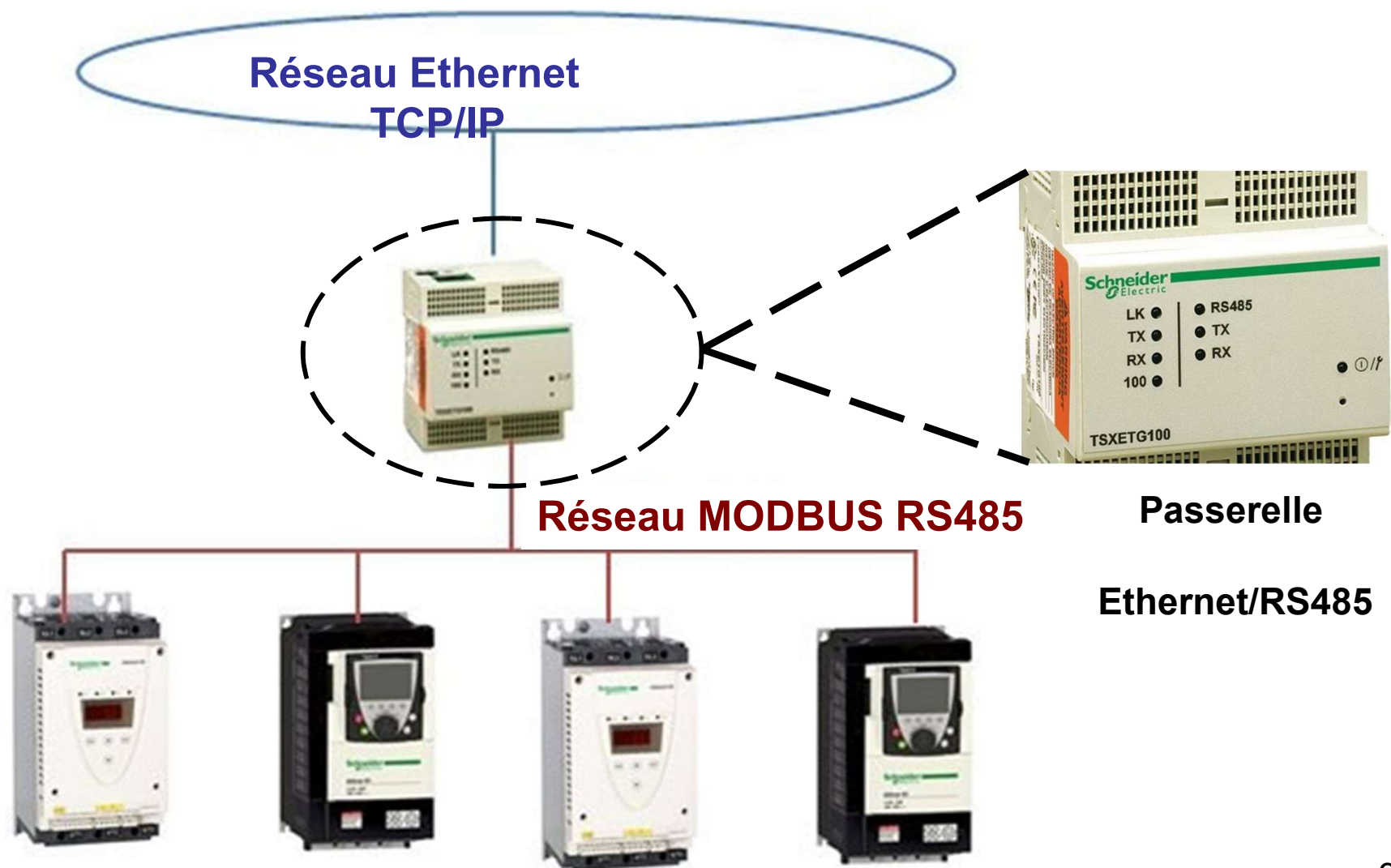


Coupleur MODBUS
(RS485/Ethernet)

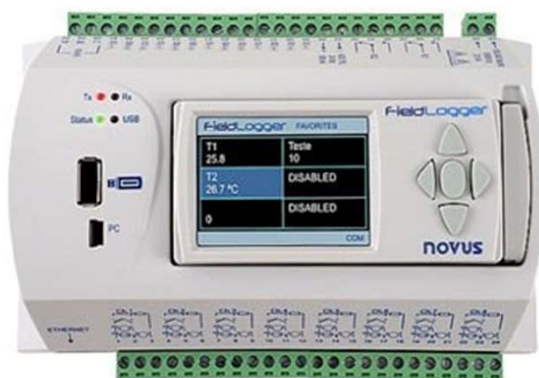


Modules d'E/S

Exemples de produits MODBUS



Exemples de produits MODBUS



Centrale de mesure

(8 entrées analogiques
configurables en
thermocouples, Pt100,
Pt1000, courant, tension)



Instruments de mesure

(Power meter/Power
monitoring)



Codeur absolu

**Ethernet TCP/IP
Modbus**

Exemples de produits MODBUS



**Interface Homme-Machine HMI
(Magelis)**



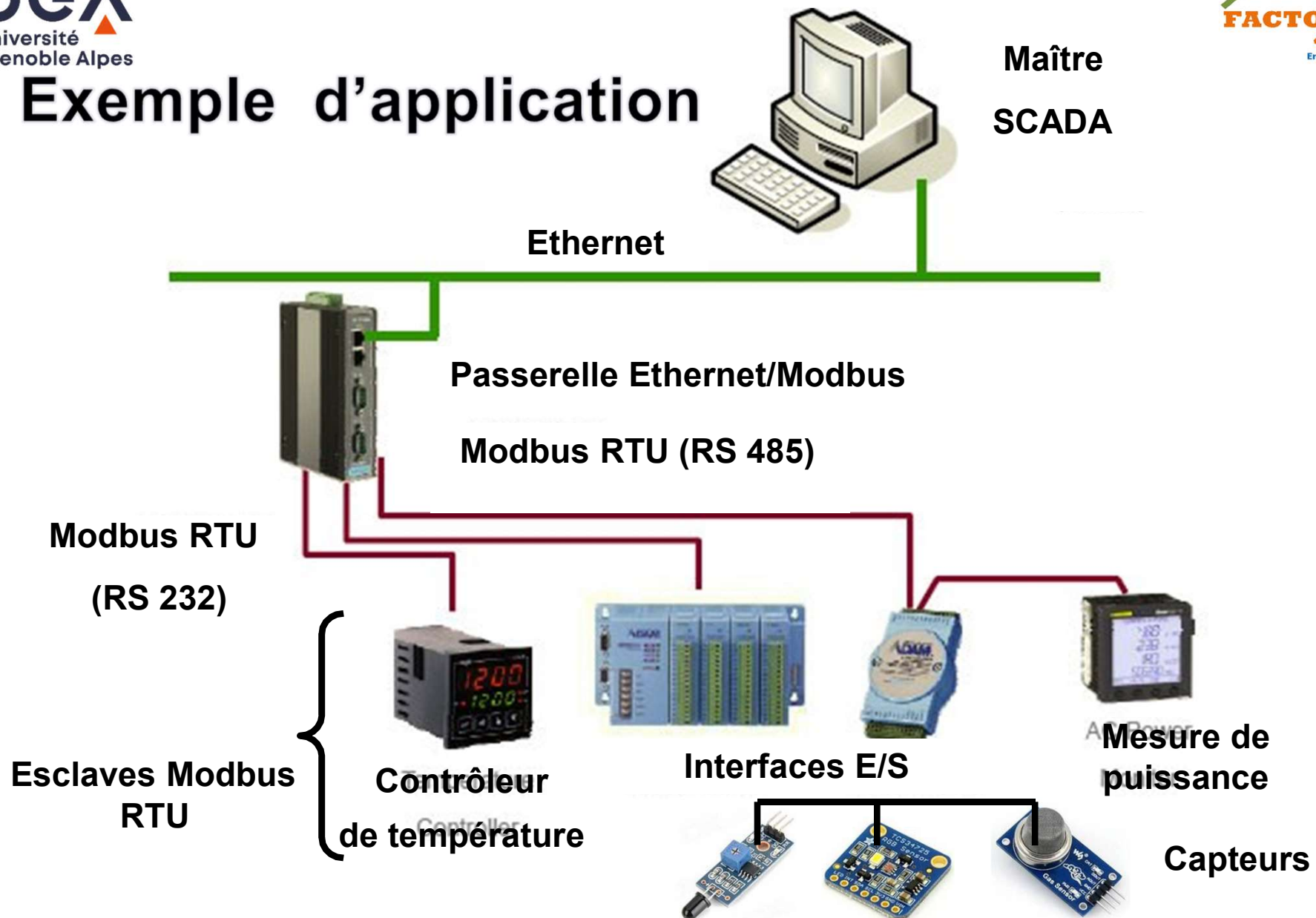
HMI

Modbus-RTU



Instrument de mesure

Exemple d'application



Références



https://www.virtual-serial-port.org/fr/articles/modbus-rtu-guide/?fbclid=IwAR3KE1yYh3nnke7FQe2zqtEY7YOsBdkXZ49xc4_a7Mzqw9qR_KRFHis_q_c#1

https://www.automation-sense.com/medias/files/guide-du-modbus-pour-les-nuls-extrait.pdf?fbclid=IwAR3dQ-_2kU2snc5wRKXGvW0wdGvDcQcP9JMz9LZJQuXjAkXp7nfCSMh8R4w

https://www.bb-elec.com/Learning-Center/All-White-Papers/Modbus/The-Answer-to-the-14-Most-Frequently-Asked-Modbus.aspx?fbclid=IwAR157ZWBb_wycTFNmaexEvbLfO3mFEpjWlpLz7P4a_-s1U-IKH9rmhOxMO8

https://fr.wikipedia.org/wiki/Modbus?fbclid=IwAR1sQcKXszXU8v9UK5nCTOi6hIfaST7Oma5KIiK6UwKOHbhrYC8_Gjcl7To