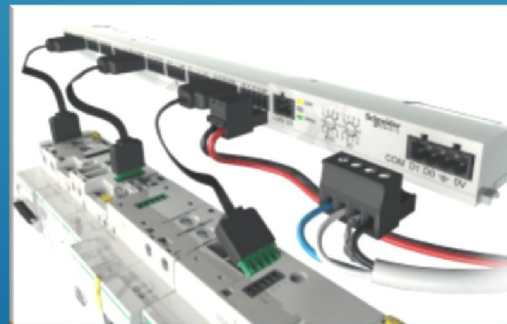
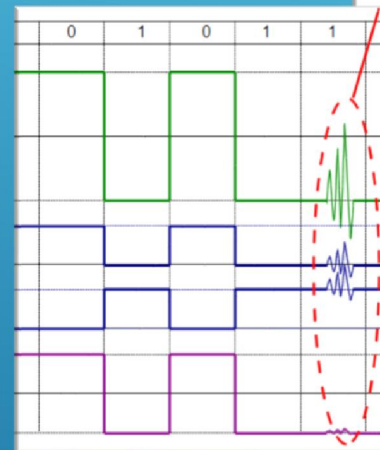


# LE PROTOCOLE



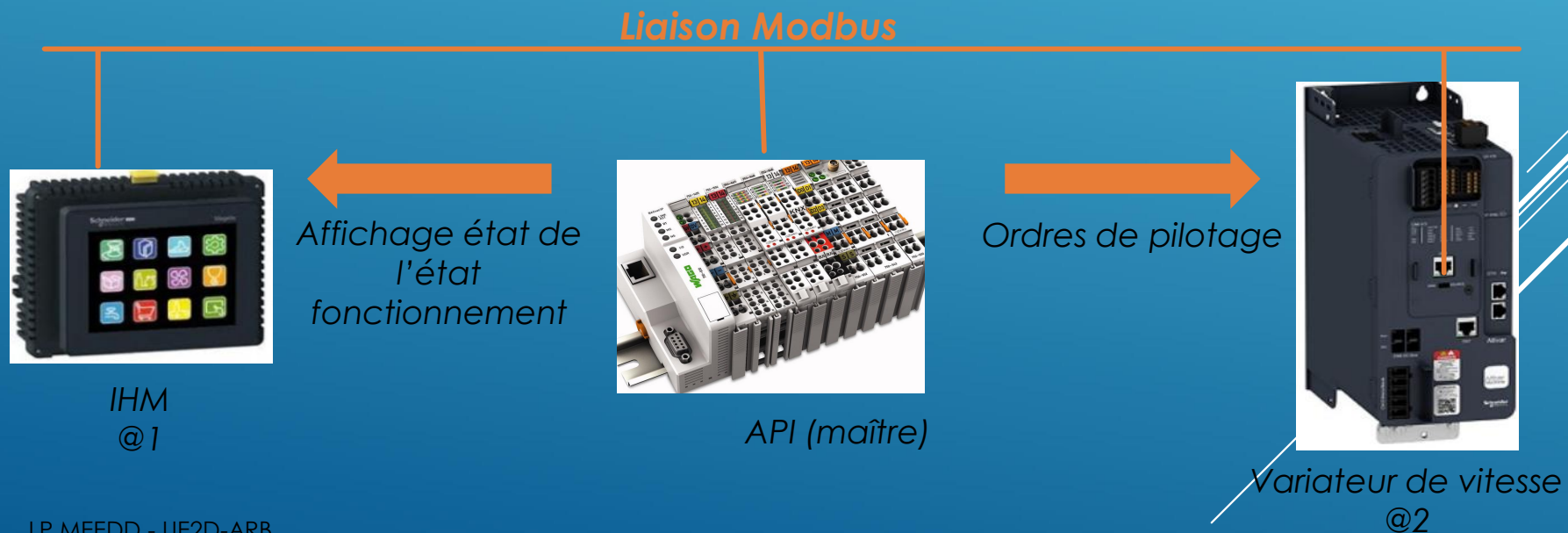
# PLAN :

- ▶ **1 - Présentation**
  - ▶ 1.1 – Objectifs
  - ▶ 1.2 – Exemple d'application
  - ▶ 1.3 – Types de données
  - ▶ 1.4 – Principales fonctions
- ▶ **2 – Supports de Communication**
  - ▶ 2.1 – RS232
  - ▶ 2.2 – RS485
  - ▶ 2.3 – Ethernet
  - ▶ 2.4 - Synthèse
- ▶ **3 – Transactions Modbus**
  - ▶ 3.1 principe des échanges
  - ▶ 3.2 Trames Modbus RTU
  - ▶ 3.3 Trames Modbus TCP
- ▶ **Annexes**
  - ▶ Principales fonctions Modbus
  - ▶ Extraits de documentations

# 1 - PRÉSENTATION

## ► 1-1 – Objectifs

- Groupe Schneider / Merlin Gerin / Télémécanique (Modicon à l'époque) à l'origine du projet
- Système de communication basé sur une liaison série
- Faire communiquer entre eux différents équipements d'automatisme (automates, variateurs, IHM...)
- Echanger des données sous formes de Bits et de Mots
- Exemple :



## □ 1-2 – Exemple d'application

- L'architecture suivante permet la supervision d'une installation électrique destinée à la ventilation d'un local :
  
- 4 variateurs de vitesse Schneider ATV61 sont utilisés pour piloter les moteurs d'entraînement des ventilateurs
  
- Un coupleur\* Modbus Wago 750-312 muni d'une carte 8 entrées TOR (750-430) est utilisé pour relevé l'état des disjoncteurs de l'installation. Les contacts auxiliaires des disjoncteurs sont reliés comme suit :
  - Q1 (extraction 1) : DI1
  - Q2 (extraction 2) : DI2
  - Q3 (insufflation 1) : DI3
  - Q4 (insufflation 2) : DI4
  - Q0 (général) : DI5 (DI6, DI7, DI8 : N.C)
  
- Un compteur d'énergie communicant Schneider IEM3150 effectue différents relevés (puissance instantanée, énergie consommée, courants, tensions...)



**Poste de supervision**  
PC + carte réseau  
Logiciel de supervision  
utilisant Modbus (TCP)  
@IP : 192.168.0.100



Réseau Ethernet



**Extraction 1**  
Variateur  
ATV61  
@ Modbus : 1



**Passerelle MB**  
Ethernet/RS485  
@IP :  
192.168.0.101



**Extraction 2**  
Variateur  
ATV61  
@ Modbus : 4



**Compteur  
Energie**  
iem3150  
@ Modbus : 5



Liaison RS485



**Insufflation 1**  
Variateur  
ATV61  
@ Modbus : 2



**Insufflation 2**  
Variateur  
ATV61  
@ Modbus : 3



**Surveillance disjoncteurs**  
Coupleur 750-312  
Carte 8<sup>E</sup> TOR  
@ Modbus : 6

## ► 1-3 – Types de données

► 4 types de données sont définis par le protocole:

	Type d'objet	Accès	Exemples
Discret Inputs	Bit	R	<ul style="list-style-type: none"><li>- entrées TOR</li><li>- Fin de course</li><li>- Contact auxiliaire de disjoncteur</li></ul>
Coils	Bit	R/W	<ul style="list-style-type: none"><li>- Sorties TOR</li><li>- Bit interne</li><li>- RAZ d'un compteur d'énergie</li></ul>
Input Registers	Mot	R	<ul style="list-style-type: none"><li>- Entrées analogiques</li><li>- Lecture d'un capteur</li></ul>
Holding Registers	Mot	R/W	<ul style="list-style-type: none"><li>- Sorties analogiques</li><li>- Variable d'un programme (ex: temporisation, opérande d'un calcul...)</li><li>- Valeur de paramétrage d'un équipement (ex : consigne de vitesse d'un variateur...)</li></ul>

✓ Les données d'un équipement sont identifiées par des adresses codées sur 16 bits.

✓ Exemple (compteur d'énergie Schneider iem3150) :

Register Address	Action (R/W/WC)	Size	Type	Units	Description
<b>Current</b>					
3000	R	2	Float32	A	I1: phase 1 current
3002	R	2	Float32	A	I2: phase 2 current
3004	R	2	Float32	A	I3: phase 3 current
3010	R	2	Float32	A	Current Avg
<b>Voltage</b>					
3020	R	2	Float32	V	Voltage L1–L2
3022	R	2	Float32	V	Voltage L2–L3
3024	R	2	Float32	V	Voltage L3–L1
3026	R	2	Float32	V	Voltage L–L Avg
3028	R	2	Float32	V	Voltage L1–N
3030	R	2	Float32	V	Voltage L2–N
3032	R	2	Float32	V	Voltage L3–N
3036	R	2	Float32	V	Voltage L–N Avg
<b>Power</b>					
3054	R	2	Float32	kW	Active Power Phase 1
3056	R	2	Float32	kW	Active Power Phase 2
3058	R	2	Float32	kW	Active Power Phase 3
3060	R	2	Float32	kW	Total Active Power
3068	R	2	Float32	kVAR	Total Reactive Power Not applicable for iEM3150 / iEM3250 / iEM3350
3076	R	2	Float32	kVA	Total Apparent Power Not applicable for iEM3150 / iEM3250 / iEM3350
<b>Power Factor</b>					
3084	R	2	Float32	–	Total Power Factor: -2 < PF < -1 = Quad 2, active power negative, capacitive -1 < PF < 0 = Quad 3, active power negative, inductive 0 < PF < 1 = Quad 1, active power positive, inductive 1 < PF < 2 = Quad 4, active power positive, capacitive
<b>Frequency</b>					
3110	R	2	Float32	Hz	Frequency

## ► 1-4 – Principales Fonctions

Les fonctions Modbus permettent à des équipements d'échanger ces différents types de données :

Code Fonction	Description
1	Lecture de bits (coils)
2	Lecture de bits d'entrée (discret inputs)
3	Lecture de mots (holding registers)
4	Lecture de mots d'entrée (input registers)
5	Ecriture de 1 bit
6	Ecriture de 1 mot
15	Ecriture de n bits
16	Ecriture de n mots

### **Exemples (installation ventilation du §1.2) :**

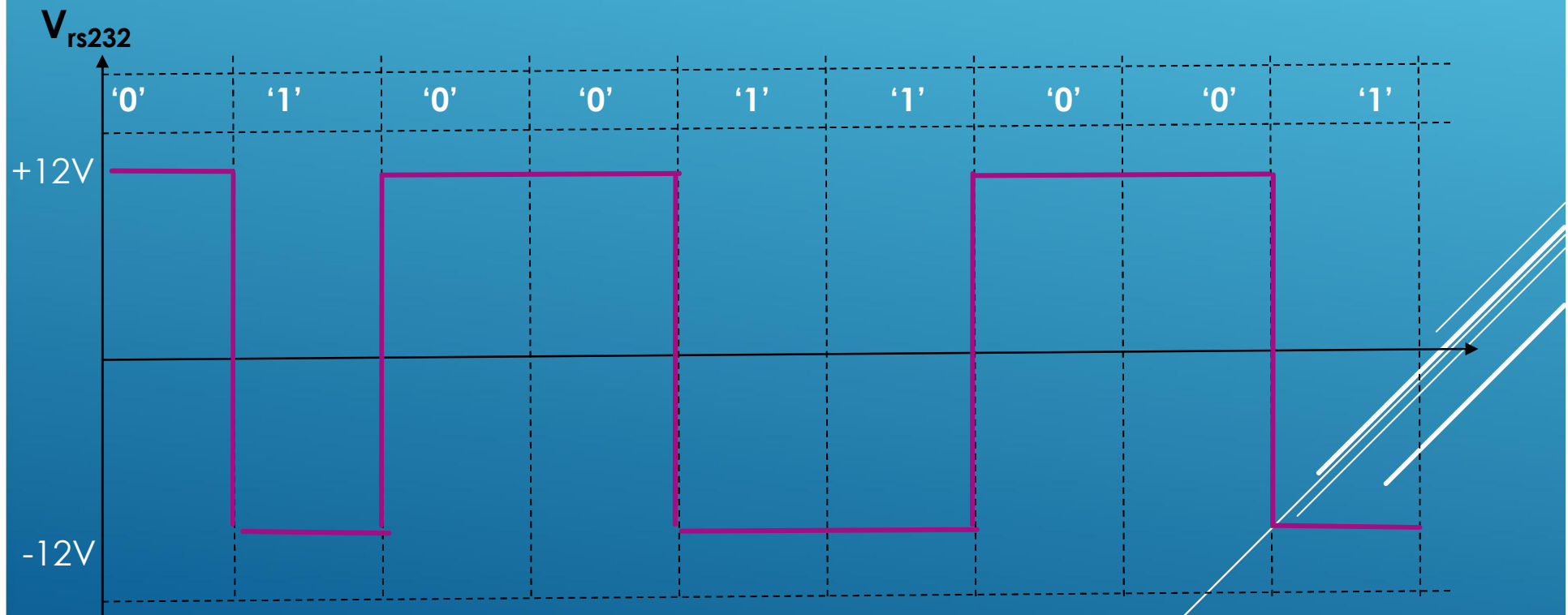
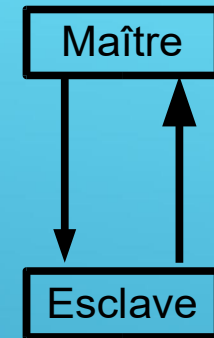
- Fixer la vitesse de rotation d'un ventilateur : fonction 6
- Relever les mesures du compteur d'énergie : fonction 3 ou 4
- Afficher l'état des disjoncteurs : Fonction 1



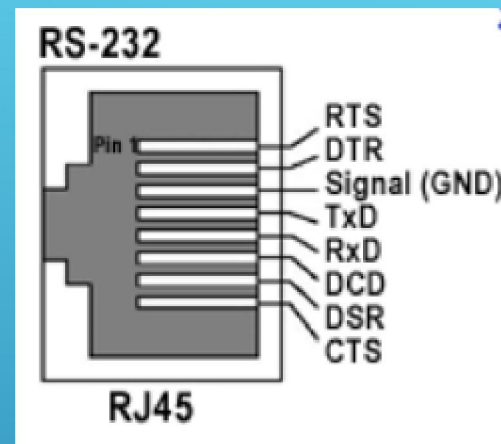
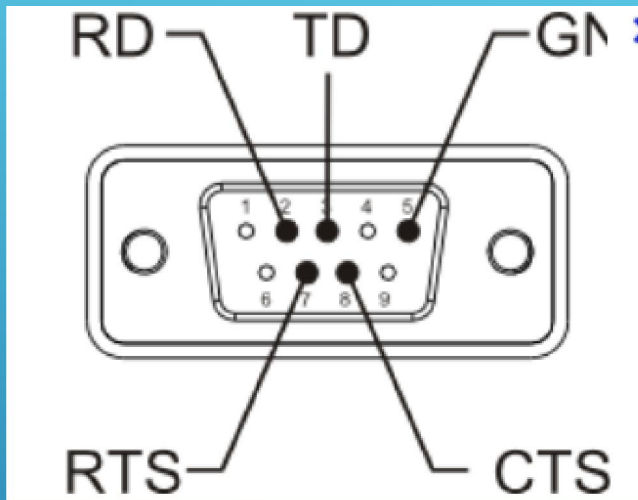
# 2 – SUPPORTS DE COMMUNICATION

## ► 2.1 - RS 232

- ✓ Liaison Point à Point
- ✓ 3 Fils minimums : Emission (Tx) – Réception (Rx) – Masse/référence commune (Gnd)
- ✓ Niveaux de tension +12V ('0') / -12V ('1')
- ✓  $D < 30m$

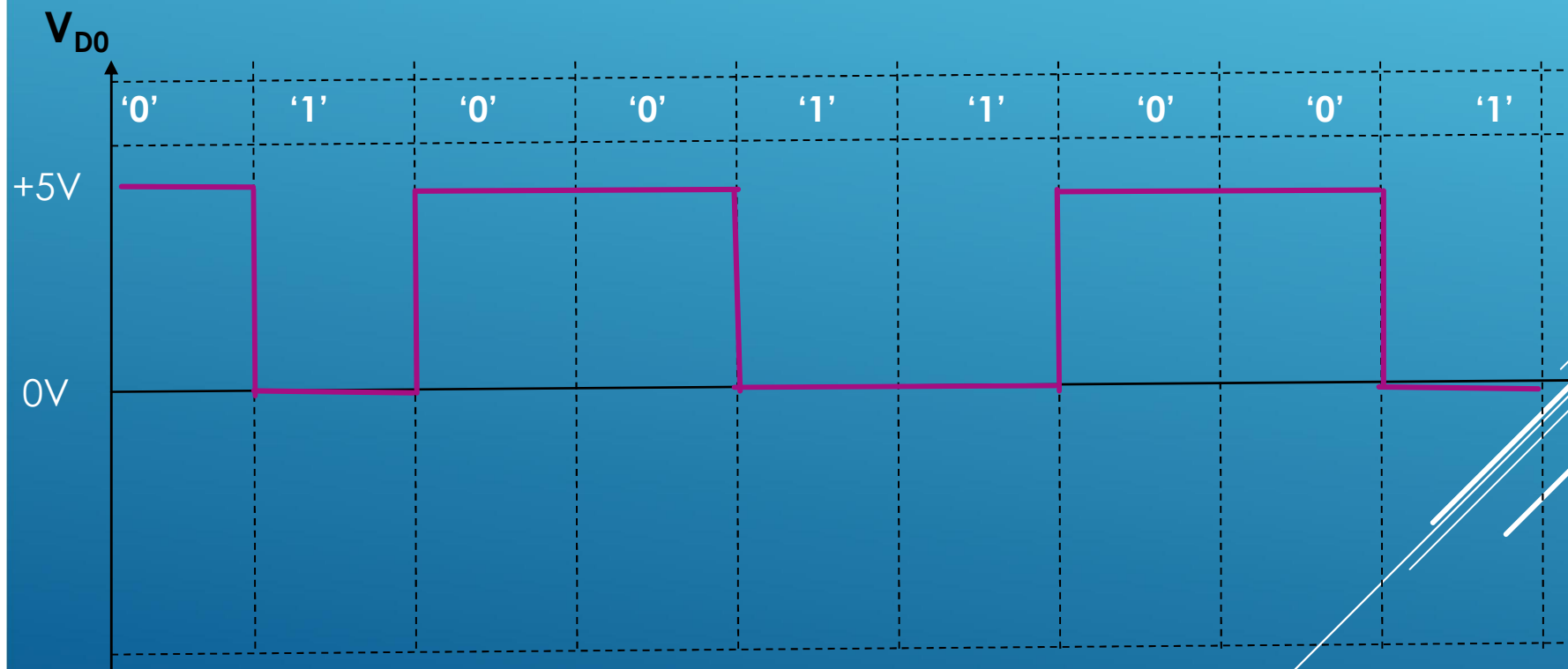
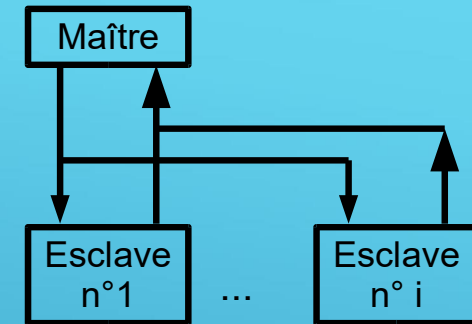


► Connectique :

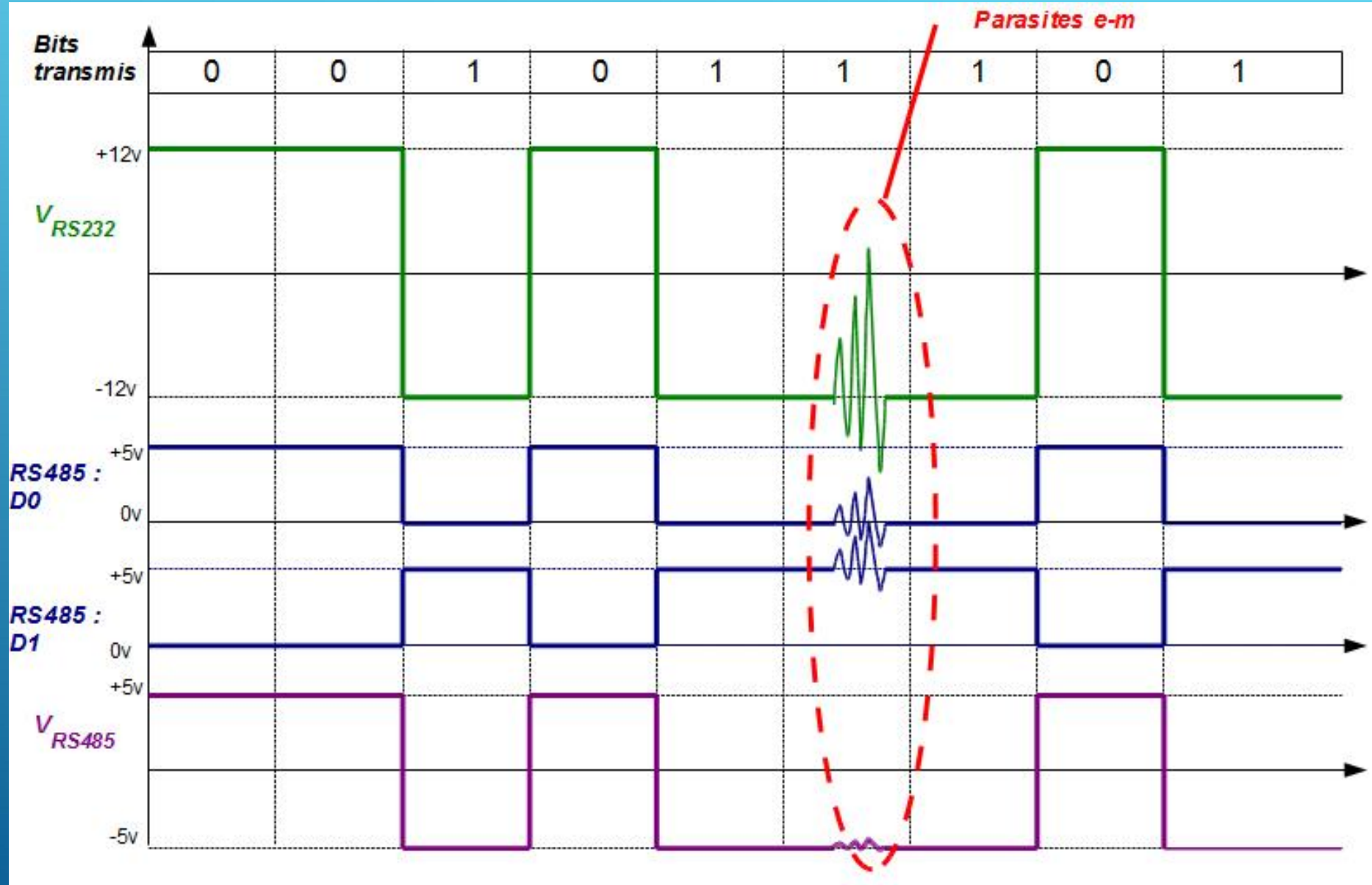


## ► 2.2 - RS 485

- ✓ Liaison Multipoints (jusqu'à 63 esclaves en Modbus)
- ✓ 2 Fils minimums : D0 – D1
- ✓ Transmission différentielle (Niveaux de tension 0V/5V)
- ✓ D < 120m (1000m avec répéteurs)



# Comparaison RS232 / RS 485 :



- ▶ Dans tous les cas (RS232 et RS485), on doit définir les paramètres de la communication considérée, et vérifier que le paramétrage soit identique sur toutes les stations :
  - ▶ **Débit binaire (« Baudrate »)** : Egalement appelé « vitesse de transmission », il s'exprime en Bauds (Bds) et correspond au nombre de bits par seconde.

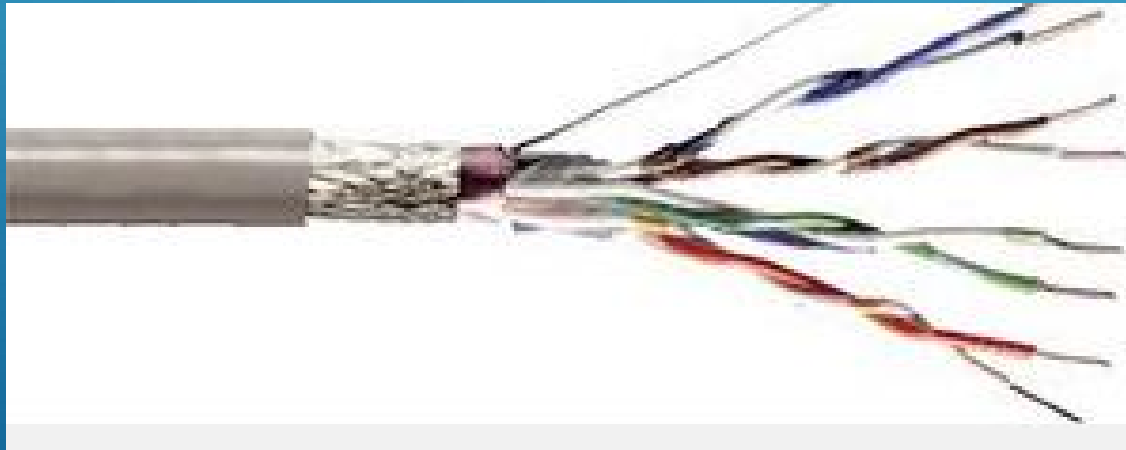
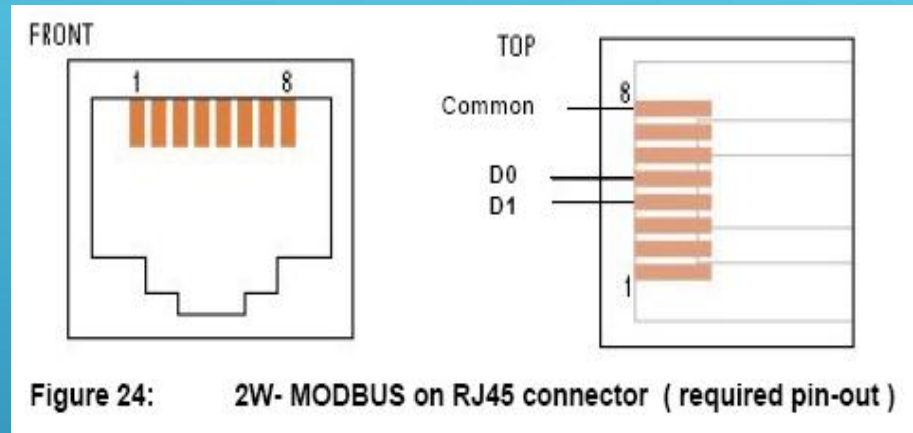
Par défaut, le débit est de 19200 Bds en Modbus. Autres débits usuels : 1200 Bds, 2400 Bds, 4800 Bds, 9600 Bds.

- ▶ **Parité (« Parity »)** : Bit de contrôle insérer à la fin de chaque octet transmis vérifiant, selon le paramétrage choisi, que le nombre de bits à '1' dans l'octet est paire (« even ») / impaire (« odd »).

Le contrôle de parité peut aussi être désactivé (« none »)

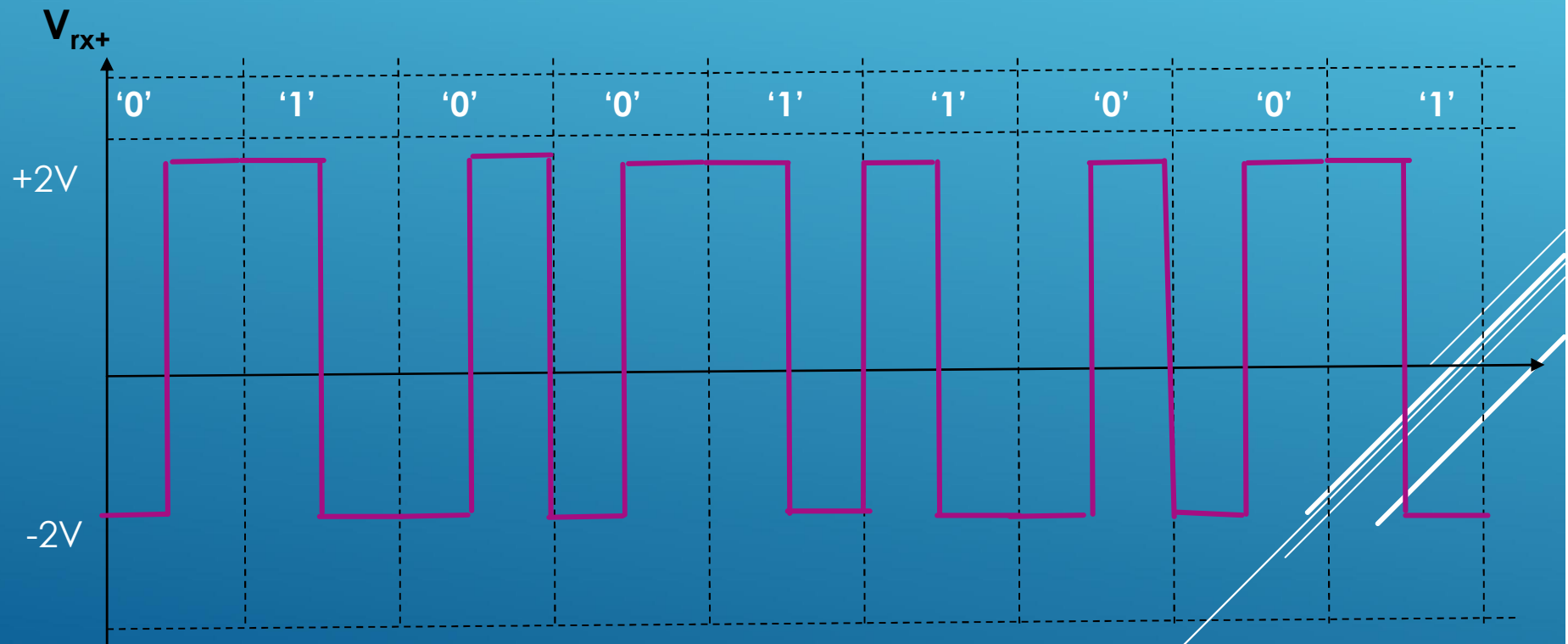
- ▶ **Bit(s) de stop** : Un bit de Start à '0' est inséré au début de chaque octet à transmettre. L'utilisateur peut également paramétrer 1 ou 2 bits de STOP (à '1') à la fin de celui-ci.
- ▶ **Contrôle de flux** : Matériel (utilisation des signaux RTS/CTS), logiciel ou aucun.

► Connectique :

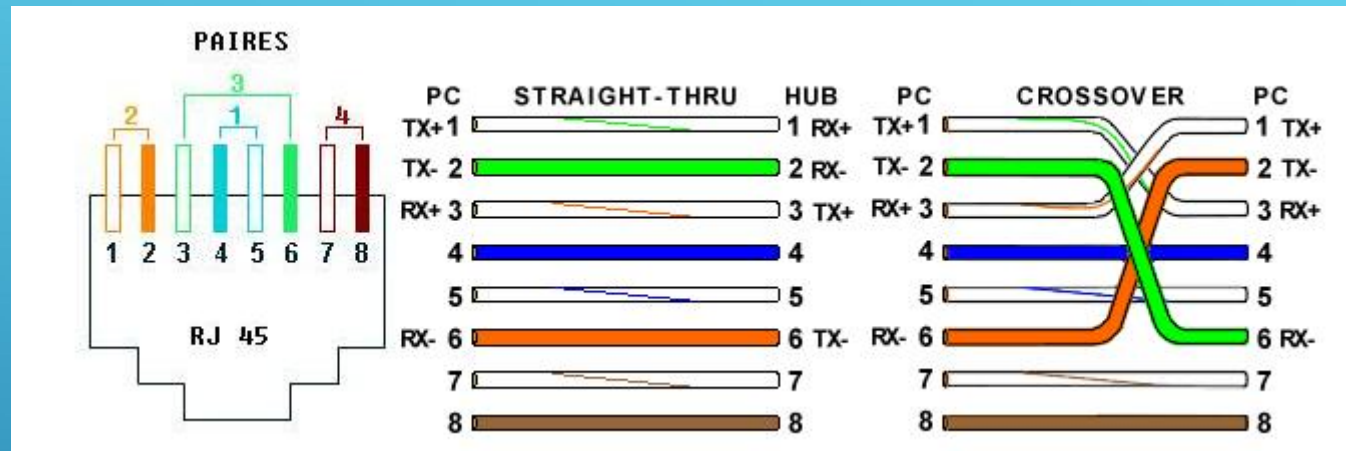


## ► 2.3 - Ethernet

- ✓ Liaison Multipoints
- ✓ 4 Fils minimums (2 paires Rx/Tx)
- ✓ Transmission différentielle (niveaux +/-2V) + codage manchester
- ✓ D < 100m



► Connectique (câble rj45 100 base T):



► **2.4 - Synthèse**

- ✓ **RS232** : Limité à 2 équipements et aux courtes distances.
- ✓ **RS485** : Solution industrielle, bonne immunité aux parasites em, choix pour les connexions dans le TGBT.
- ✓ **Ethernet** : Lien vers la supervision



# 3 – TRANSACTIONS MODBUS

## ▶ 3-1 – Principe des échanges

- ▶ Un échange Modbus est également appelé transaction
- ▶ Une transaction est composée d'un couple Requête/Réponse
- ▶ Un réseau Modbus est composé d'un maître et d'un ou plusieurs esclave(s)
- ▶ Le maître est le seul à pouvoir émettre des requêtes
- ▶ Les esclaves répondent au maître
- ▶ Le protocole Modbus définit la composition des trames de requêtes et de réponse

## ► 3-2 – Trames Modbus RTU

- La variante RTU de Modbus est la forme la plus utilisée sur lignes séries (RS485 et RS232)
- Une frame Modbus RTU comporte les champs suivants :
  - *Rq les trames sont généralement présentées en format hexadécimal*

Adresse	Code Fonction	Données	CRC
1 octet	1 octet	N octets	2 octets

- Adresse : n° d'esclave Modbus
- Code fonction : définit le type d'échange (cf § 1.4)
- Données : contient les données relatives à la fonction. Pour une lecture : adresse et nombre de données à lire. Pour une écriture : adresse, nombre de valeurs et valeurs à écrire.
- CRC : champ de contrôle

## Transaction sans erreur :

Maître

Adresse	Code Fonction	Données	CRC
1 octet	1 octet	N octets	2 octets

Esclave

Adresse	Code Fonction	Données	CRC
Idem requête	Idem requête	N octets	2 octets

Maître

## Transaction avec erreur :

Maître

Adresse	Code Fonction	Données	CRC
1 octet	1 octet	N octets	2 octets

Esclave

Adresse	Code Fonction D'exception	Code d'exception	CRC
Idem requête	Code + 0x80	1 octets 1 : $\epsilon$ sur code fct 2 : $\epsilon$ sur adresse 3 : $\epsilon$ sur données 4 : autre $\epsilon$	2 octets

Maître

- Lecture de l'état des disjoncteurs pour affichage sur le superviseur :

@ Modbus	Code Fct	Adr 1 <sup>er</sup> bit	Nb de bits	CRC
06	02	00 00	00 05	XX XX

- Réponse (seul Q3 est ouvert) :

@ Modbus	Code Fct	Nb octets	État des bits	CRC
06	02	01	0	XX XX

xxx1 1011<sub>2</sub> → 1B<sub>16</sub>  
 Q0 Q4 Q3 Q2 Q1

- Affichage des vitesses de rotation des 4 ventilateurs sur le superviseur :

@ Modbus	Code Fct	Adr 1 <sup>er</sup> mot	Nb de mots	CRC
01	03	01 91	00 01	XX XX

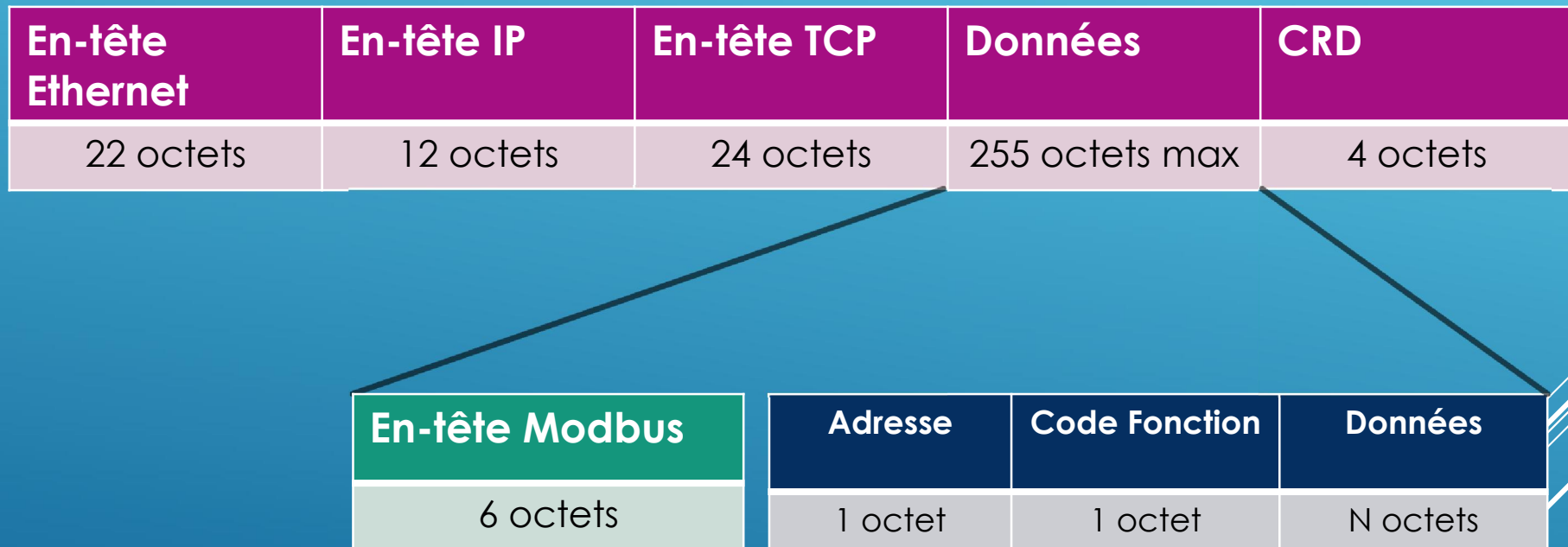
Idem pr les 3 autres ventilateurs  
 avec @Modbus=2,3 et 4

401<sub>10</sub> → 191<sub>16</sub>

- Réponses (Ext1=5,5 Hz, Ext2=15 Hz, Ins1=7Hz, Ins2=22,5Hz)
- Commande du ventilateur d'extraction 2 (Consigne de fréquence = 20 Hz)
- Réponse :

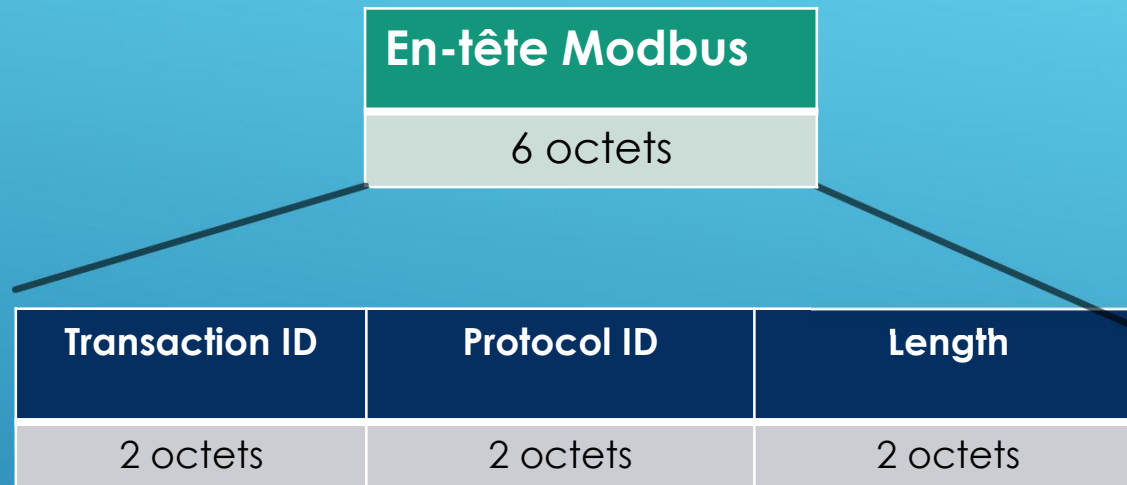
### ▶ 3-3 – Trames Modbus TCP

- ▶ Le protocole Modbus TCP permet l'encapsulation de trames Modbus dans les trames Ethernet :



- ▶ Le Port n°502 est utilisé pour les échanges Modbus sur TCP.

- ▶ Modbus rajoute une en-tête composée des champs suivants :



- ✓ Transaction ID :
- ✓ Protocol ID :
- ✓ Length :

- Lecture des courants pour affichage sur le superviseur:

- Lecture des tensions pour affichage sur le superviseur:



- Lecture des puissances affichage sur le superviseur:

# ANNEXES

## **- PRINCIPALES FONCTIONS MODBUS -**

- Fonction 01
- Fonction 02
- Fonction 03
- Fonction 04
- Fonction 05
- Fonction 15
- Fonction 16

## **- DOCUMENTATIONS TECHNIQUES -**

- Variateur ATV61
- Coupleur Modbus Wago 750-31x
- Borne 8 E TOR Wago 750-430

## Fonction 0x01 : Lecture de n bits de sorties.

Cette fonction permet de lire 1 à 2000 bits consécutifs dans le segments mémoire du serveur correspondants aux sorties discrètes (« coils »).

La trame de requête contient les champs suivants :

<b>Code Fonction</b>	1 Octet	0x01
<b>Adresse de départ</b>	2 Ocets	De 0x0000 à 0xFFFF
<b>Nombre de bits à lire :</b>	2 Octets	De 1 à 2000 (0x7D0)

La trame de réponse, si aucune erreur ne survient :

<u>Champ :</u>	<u>Taille :</u>	<u>Valeur :</u>
Code Fonction :	1 octet	0x01
Nombre d'octets de la trame de réponse :	1 octets	N*
Etat des bits lus :	N octets	Etat des bits lus

\* Rq :  $N = (nb\_de\_bits) / 8$  si  $nb\_de\_bits$  est un multiple de 8  
 $N = (nb\_de\_bits) / 8 + 1$  dans le cas contraire

## Fonction 0x02 : Lecture de n bits d'entrées.

Cette fonction permet de lire 1 à 2000 bits consécutifs dans le segments mémoire du serveur correspondants aux entrées discrètes (« discrete inputs »).

La **trame de requête** contient les champs suivants :

Code Fonction	1 octet	0x02
Adresse du 1° bit à lire	2 octets	0x00 à 0xFFFF
Nombre de bits à lire	2 octets	0x00 à 0x2000

La **trame de réponse**, si aucune erreur ne survient :

Code Fonction	1 octet	0x02
Nombre d'octets de du champ suivant	1 octets	N
État des bits lus	N octets	Valeurs lues

## Fonction 0x03 : Lecture de n registres.

Cette fonction permet de lire 1 à 125 mots (16 bits) consécutifs dans le segments mémoire du serveur correspondants aux registres accessibles en lecture/écriture (« Holding register »).

La trame de requête contient les champs suivants :

<u>Champ :</u>	<u>Taille :</u>	<u>Valeur :</u>
Code Fonction :	1 octet	0x03
Adresse du 1° registre à lire :	2 octets	0 à 0xFFFF
Nombre de registres à lire (n) :	2 octets	1 à 125 (0x7D)

La trame de réponse, si aucune erreur ne survient :

<u>Champ :</u>	<u>Taille :</u>	<u>Valeur :</u>
Code Fonction :	1 octet	0x03
Nombre d'octets du champ suivant :	1 octet	2 x n
Mots lus :	n x 2 octets	Valeurs lues

## Fonction 0x04 : Lecture de n mots d'entrée.

Cette fonction permet de lire 1 à 125 mots (16 bits) consécutifs dans le segments mémoire du serveur correspondants aux registres accessibles en lecture seule (« Input register »).

La trame de requête contient les champs suivants :

<u>Champ :</u>	<u>Taille :</u>	<u>Valeur :</u>
Code Fonction :	1 octet	0x04
Adresse de départ :	2 octets	0x0000 à 0xFFFF
Nombre de registres à lire (n) :	2 octets	1 à 125 (0x7D)

La trame de réponse, si aucune erreur ne survient :

<u>Champ :</u>	<u>Taille :</u>	<u>Valeur :</u>
Code Fonction :	1 octet	0x04
Nombre d'octets de la trame de réponse :	1 octet	2 x n
Valeurs de registres lus :	nx2 octets	valeurs

## Fonction 0x05 : Ecriture d'un bit de sortie.

Cette fonction permet de forcer une sortie à '1' ou à '0' sur l'équipement distant. Le sous-champ « valeur » du champ « données » contient 2 octets : la valeur 0xFF00 permet le forçage à '1', la valeur 0x0000 le forçage à '0'; toutes les autres valeurs sont interdites. La réponse à cette requête est un écho de la requête.

La **trame de requête** contient les champs suivants :

<u>Champ :</u>	<u>Taille :</u>	<u>Valeur :</u>
Code Fonction :	1 octet	0x05
Adresse du bit à écrire :	2 octets	0x0000 à 0xFFFF
Valeur à écrire ('0' ou '1') :	2 octets	<b>0x0000 ou 0xFF00</b>

*Si aucune erreur ne survient, la trame de réponse est identique à la trame de requête (écho)*

La **trame de réponse**, si une erreur survient :

<u>Champ :</u>	<u>Taille :</u>	<u>Valeur :</u>
Code d'erreur :	1 octet	0x85
Code d'exception :	1 octet	01, 02, 03, ou 04

## Fonction 0x06 : Ecriture d'un registre de sortie.

Cette fonction permet l'écriture d'une variable sur un mot accessible en lecture/écriture de l'équipement distant. La réponse à cette requête est un écho de la requête.

La **trame de requête** contient les champs suivants :

<u>Champ :</u>	<u>Taille :</u>	<u>Valeur :</u>
Code Fonction :	1 octet	0x06
Adresse du registre à écrire :	2 octets	0x0000 à 0xFFFF
Valeur à écrire (big endian) :	2 octets	<b>0x0000 à 0xFFFF</b>

*Si aucune erreur ne survient, la trame de réponse est identique à la trame de requête (écho)*

La **trame de réponse**, si une erreur survient :

<u>Champ :</u>	<u>Taille :</u>	<u>Valeur :</u>
Code d'erreur :	1 octet	0x86
Code d'exception :	1 octet	01, 02, 03, ou 04

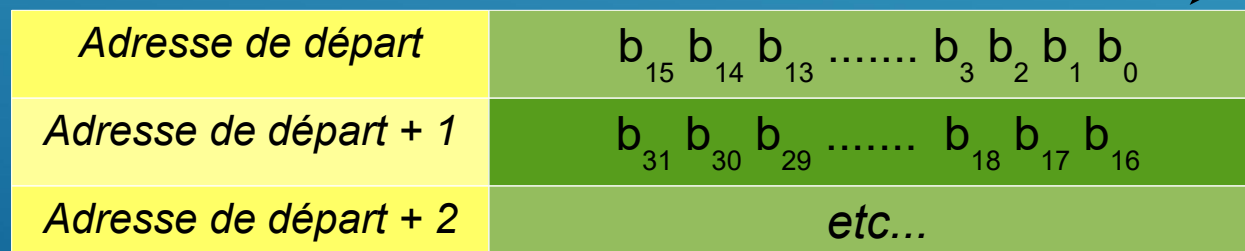


## Fonction 0x0F : Ecriture de n bits de sortie.

Cette fonction permet d'écrire 1 à 1968 bits consécutifs d'un équipement distant. La réponse à cette requête renvoie le nombre de bits écrits ainsi que l'adresse de départ en écho.

La trame de requête contient les champs suivants :

<u>Champ :</u>	<u>Taille :</u>	<u>Valeur :</u>
Code Fonction :	1 octet	0x0F
Adresse de départ :	2 octets	0x0000 à 0xFFFF
Nombre de bits à écrire :	2 octets	<b>0x0001 à 0x07B0</b>
Nombre d'octets du champ suivant	1 octet	0 à 255
Valeurs à écrire	N octets	.....



## Fonction 0x10 : Ecriture de n registres.

Cette fonction permet d'écrire 1 à 123 registres consécutifs d'un équipement distant. La réponse à cette requête renvoie le nombre de registres écrits ainsi que l'adresse de départ.

La **trame de requête** contient les champs suivants :

<u>Champ :</u>	<u>Taille :</u>	<u>Valeur :</u>
Code Fonction :	1 octet	0x10
Adresse du 1° mot à écrire :	2 octets	0x0000 à 0xFFFF
Nombre de mots à écrire (n) :	2 octets	0 à 0x7B
Nombre d'octets du champ suivant	1 octets	0 à 255
Valeurs à écrire (big endian) :	n octets	<b>0x0000 à 0xFFFF</b>

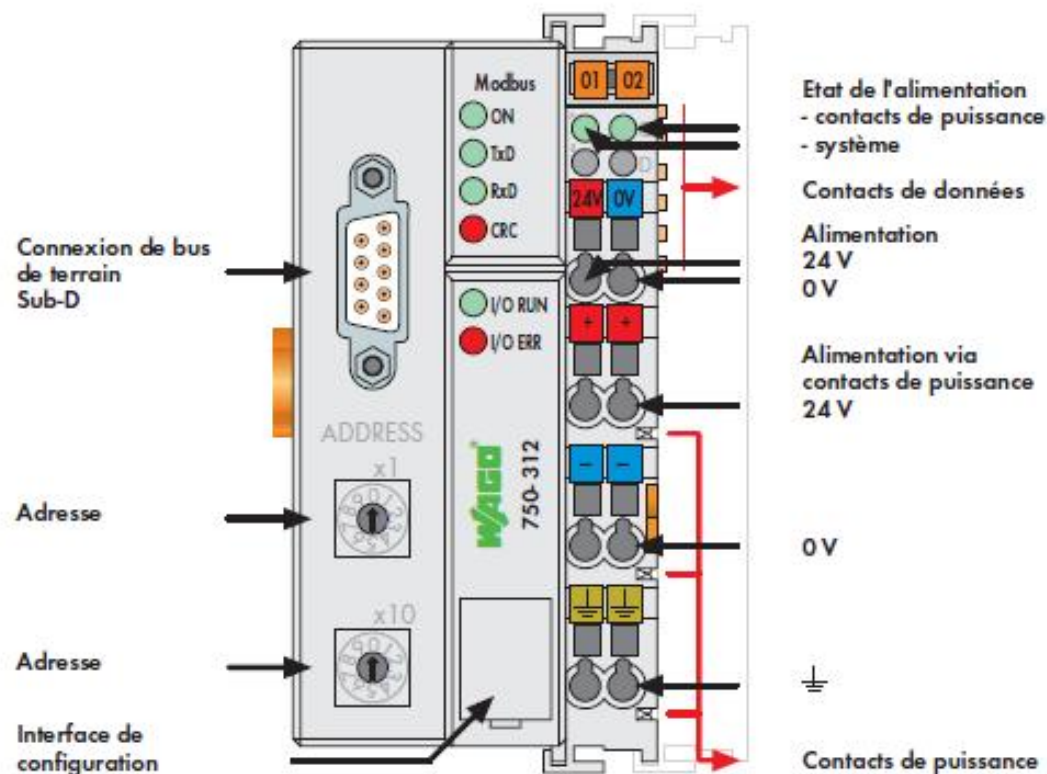
La **trame de réponse (sans erreur)** :

<u>Champ :</u>	<u>Taille :</u>	<u>Valeur :</u>
Code Fonction :	1 octet	0x10
Adresse du 1° mot écrit :	2 octet	0 à 0xFFFF
Nombre de mots écrits	2 octets	n

750-312, 750-314, 750-315, 750-316

## Coupleur de bus de terrain MODBUS

RS-232/-485; 150 (1200) bauds ... 19,2 (115,2) kbauds; signaux digitaux et analogiques



Ce coupleur permet de connecter les modules WAGO-I/O-SYSTEM au réseau MODBUS en tant que station esclave. Il supporte également le protocole ASCII.

Le coupleur reconnaît automatiquement les bornes d'E/S et crée une table image correspondant aux E/S. Le bornier peut être constitué indifféremment de modules analogiques (échange de données par mot-word) et de modules digitaux (échange de données par bit).

Lors de la création automatique de la table image, les E/S analogiques apparaissent en premier dans l'ordre de leur position sur le bornier en partant du coupleur et en allant vers la borne finale de bus. Les bits des signaux d'E/S digitaux sont placés dans le premier bit de l'octet suivant les analogiques, et sont compactés sous la forme d'un octet (byte) dans l'ordre de leur position sur le bornier en partant du coupleur et en allant vers la borne finale de bus. Si le nombre de signaux d'E/S digitaux occupe plus de 8 bits, le coupleur commence automatiquement un nouvel octet. Les bits d'entrées et de sorties se trouvent dans des octets différents, une séparation claire entre les E/S est ainsi maintenue.

Lecture Mots : FC 03 / 04		
DEC	HEX	CEI 61131-3
0	0x0000	%IW0
255	0x00FF	%IW255
256	0x0100	%QW256
511	0x01FF	%QW511
512	0x0200	%QW0
767	0x02FF	%QW255
768	0x0300	%IW256
1023	0x03FF	%IW511
12288	0x3000	%MW0
24575	0x5FFF	%MW12287
24576	0x6000	%IW512
25340	0X62FC	%IW1275
28672	0x7000	%QW512
29346	0X72FC	%QW1275

Ecriture Mots : FC 06 / 16		
DEC	HEX	CEI 61131-3
0	0x0000	%QW0
255	0x00FF	%QW255
256	0x0100	%IW256
511	0x01FF	%IW511
512	0x0200	%QW0
767	0x02FF	%QW255
768	0x0300	%IW256
1023	0x03FF	%IW511
12288	0x3000	%MW0
24575	0x5FFF	%MW12287
24576	0x6000	%QW512
25340	0X62FC	%QW1275
28672	0x7000	%QW512
29346	0X72FC	%QW1275

Lecture Bits : FC 01 / 02		
DEC	HEX	CEI 61131-3
0	0x0000	%IX0.0
255	0x00FF	%IX15.15
256	0x0100	%IX16.0
511	0x01FF	%IX31.15
512	0x0200	%QX0.0
767	0x02FF	%QX15.15
768	0x0300	%QX16.0
1023	0x03FF	%QX31.15
4096	0x1000	%QX256.0
8191	0x1FFF	%QX511.15
8192	0x2000	%IX256.0
12287	0x2FFF	%IX511.15
12288	0x3000	%IX512,0
13815	0X35F7	%IX1275,15
16384	0x4000	%QX512,0
17911	0x45F7	%QX1275,15

Ecriture Bits : FC 05 / 15		
DEC	HEX	CEI 61131-3
0	0x0000	%QX0.0
255	0x00FF	%QX15.15
256	0x0100	%QX16.0
511	0x01FF	%QX31.15
512	0x0200	%QX0.0
767	0x02FF	%QX15.15
768	0x0300	%QX16.0
1023	0x03FF	%QX31.15
4096	0x1000	%IX256.0
8191	0x1FFF	%IX511.15
8192	0x2000	%IX256.0
12287	0x2FFF	%IX511.15
12288	0x3000	%QX512,0
13815	0X35F7	%QX1275,15
16384	0x4000	%QX512,0
17911	0x45F7	%QX1275,15

## Borne 8 E TOR Wago 750-430 :

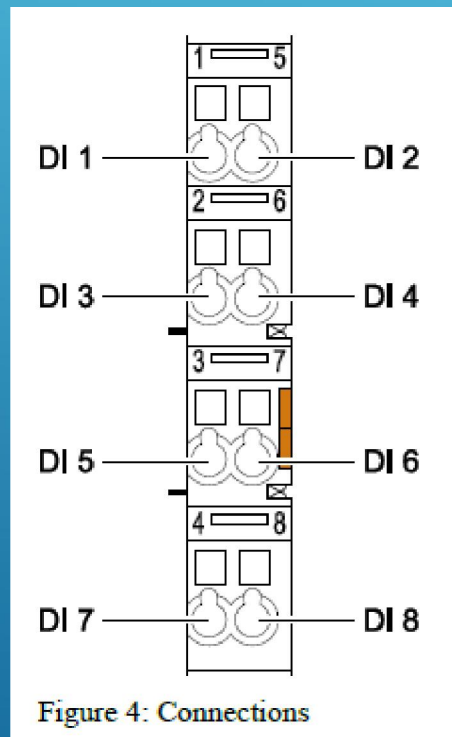


Table 9: Input bits

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DI 8	DI 7	DI 6	DI 5	DI 4	DI 3	DI 2	DI 1
DI 1	Signal state DI 1 – Digital input channel 1						
DI 2	Signal state DI 2 – Digital input channel 2						
DI 3	Signal state DI 3 – Digital input channel 3						
DI 4	Signal state DI 4 – Digital input channel 4						
DI 5	Signal state DI 5 – Digital input channel 5						
DI 6	Signal state DI 6 – Digital input channel 6						
DI 7	Signal state DI 7 – Digital input channel 7						
DI 8	Signal state DI 8 – Digital input channel 8						

# Variateur de Vitesse ATV61 :

Tableau 20 : Paramètres de commande

Mot	Code	Unités	Description	Valeurs ou gamme possibles
W400	CMD	–	Paramètre de commande DRIVECOM. Le paramètre est réinitialisé à la fin du dépassement de temps (time-out) sauf si le bit 14 de CMI (W402) est mis à 1.	<p>Bit 0 = 0 and Bit 15 = 0 : pas prêt            Bit 1 = 1 et Bit 15 = 0 : prêt            Bit 1 = 0 : Retour à l'état «Commande en ligne inhibée»            Bit 1 = 1 : Aucune action</p> <p>Bit 2 = 0 et Bit 15 = 0 : arrêt d'urgence            Bit 2 = 1 : Aucune action</p> <p>Bit 3 = 0 et Bit 15 = 0 : commande d'arrêt DRIVECOM            Bit 3 = 1 et Bit 15 = 0 : commande de marche DRIVECOM</p> <p>Bits 4 à 6 : réservés</p> <p>Bit 7 = 0 : Aucune action            Bit 7 = 1 : Effacement des défauts</p> <p>Bit 8 = 0 et Bit 15 = 1 : Activation de commande via liaison série            Bit 8 = 1 et Bit 15 = 1 : Désactivation de commande via liaison série</p> <p>Bits 9 et 10 : réservés</p> <p>Bit 11 = 0 : Commande de sens normal            Bit 11 = 1 : Commande de sens inverse</p> <p>Bit 12 = 0 : Commande de marche du moteur (RUN)            Bit 12 = 1 : Commande d'arrêt du moteur</p> <p>Bit 13 = 0 : Aucune action            Bit 13 = 1 : Arrêt par commande d'injection cc</p> <p>Bit 14 = 0 : Aucune action            Bit 14 = 1 : Commande d'arrêt rapide</p> <p>Bit 15 = 0 : Registre de commande DRIVECOM            Bit 15 = 1 : Registre de commande VVD</p>
W401	LFR	0,1 Hz	Référence de fréquence en mode de ligne (signée en complément à deux). Le paramètre est réinitialisé à la fin du dépassement de temps (time-out) sauf si le bit 14 de CMI (W402) est mis à 1.	LSP à HSP