

# S7: Modbus

Introduction aux protocoles  
MODBUS et JBus



## Présentation

### **MODBUS est un protocole**

- ***Très fiable à cause du contrôle CRC***
- ***Dialogue hiérarchisé entre un maître et plusieurs esclaves***
- ***Possibilité de multipoints entre Maître et Esclaves***
- ***Liaison RS-485, RS422, RS432, et Ethernet (Modbus over Ethernet)***
- **Topologie bus**
- ***Inventé en 1979 par la société MODICON***

2

### **Historique:**

1979: Création de MODBUS par MODICON (Modular Digital Controller)  
1994: Modicon fusionne avec Schneider (Telemecanique / April / Square D)  
2003: Transfert de compétences Schneider à MODBUS-IDA  
2004: Pré-Standard international IEC62030  
2004: MODBUS/TCP leader mondial (840000 nœuds)  
2005: MODBUS adopté en tant que standard chinois

### **Domaines d'utilisation:**

Manufacturier, Infrastructures, Energie, Bâtiment

### **Applications :**

Echanges automate <-> périphérie décentralisée

Echanges supervision <-> automate



## L'essentiel

### Un système de question/réponse

#### **4 champs (dans la question et aussi dans la réponse)**

- **Adresse de l'esclave (1 octet)**
- **Code fonction (1 octet)**
- **Paramètres (dépend du code fonction)**
- **CRC (2 octets)**

3

Q/A

Q: Le maître a aussi une adresse ?

R: Pas besoin. les esclaves ne s'adressent jamais au maître, ils ne savent que répondre.

Q: Alors l'esclave répond avec sa propre adresse dans la champs Adresse ?

R: Oui.

Q: Pourquoi ?

R: Pour qu'un autre esclave, qui se réveillerait sur le bus au moment de la réponse sans avoir entendu la question, ne prenne par mégarde cette réponse pour une question qui lui serait posée.

Q: Pas de CRLF comme en ASCII ? Lorsque les requêtes sont en binaire, comment sait-on la fin de la requête ?

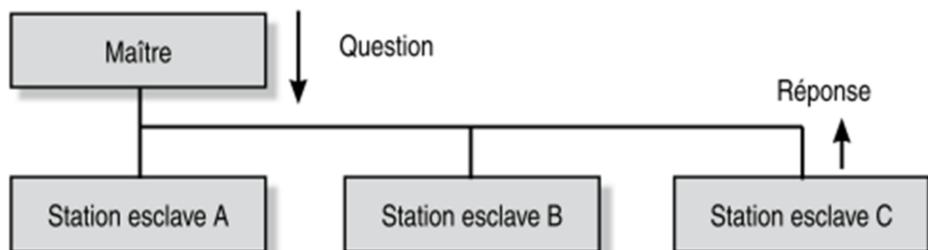
R1: Il y a le silence qui dure un temps équivalent à celui nécessaire pour transmettre 3,5 octets

R2: Si on sait comprendre le code fonction (c'est le cas de l'esclave adressé) on peut aussi décoder pendant la réception.

# Principe du dialogue

## Echange maître vers un esclave

Le maître interroge un esclave sur le réseau et attend de la part de cet esclave une réponse.



4

## Deux types de dialogue :

**Dialogue :** La maître parle à un esclave en particulier et attend sa réponse. C'est le cas général,

**Diffusion :** Le maître parle à l'ensemble des esclaves, sans attente de réponse (en général c'est pour initialiser tout le système)

## Dans tous les cas :

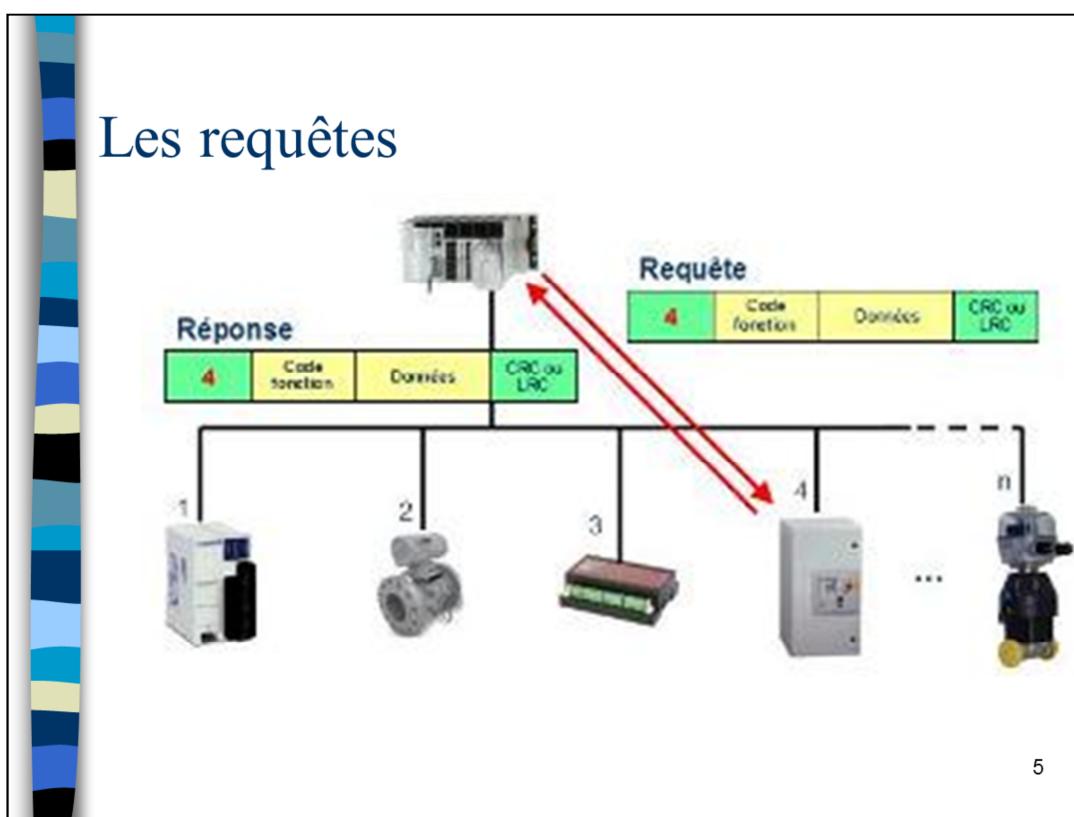
Un seul équipement peut-être en train d'émettre sur le bus (c'est du half duplex).

Pas d'échange simultané dans les deux sens (pas de full duplex).

Aucun esclave ne peut demander un échange au maître.

Le dialogue entre esclaves est impossible.

## Les requêtes



Le maître (client) envoie une requête à destination d'un esclave (serveur).

La requête contient :

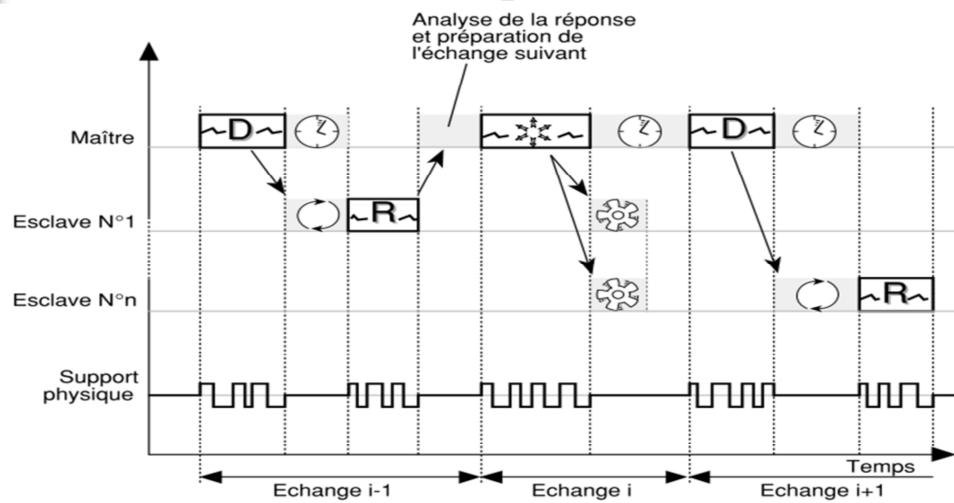
- l'adresse de l'esclave à interroger,
- un code fonction, qui indique le type d'action à exécuter (lecture bit, écriture registre,...),
- la plage de bits/registres concernés,
- les données à écrire dans le cas d'une écriture.

La réponse contient :

- l'adresse de l'esclave qui répond,
- un code fonction, qui indique le type d'action exécutée,
- le nombre d'octets de données compris dans la réponse,
- les données lues dans le cas d'une lecture.

MODBUS ne définit pas de notion de temps de cycle. Les requêtes sont envoyées à la demande du programme, en cas de besoin.

## Un silence sépare chaque requête On travaille en half-duplex

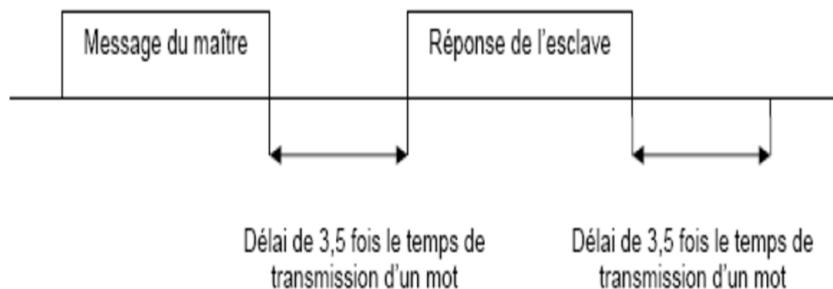


6

Remarque : les durées de DEMANDE, REPONSE, DIFFUSION, ATTENTE, et TRAITEMENT sont liées à la fonction réalisée.



## Les requêtes - Synchronisation

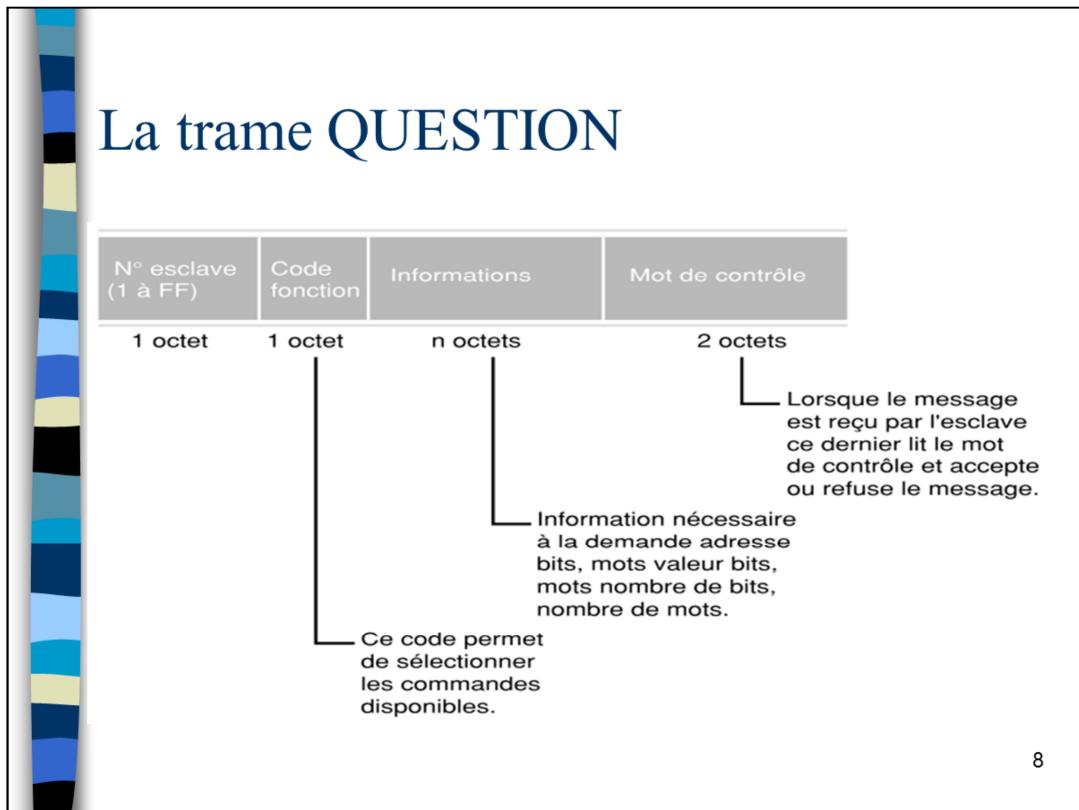


7

Tout caractère reçu après un silence supérieur à 3 caractères est considéré comme  
.....

Avant et après chaque message, il doit y avoir un silence équivalent à 3,5 fois le temps de transmission d'un caractère (soit  $3,5 * (\text{nombre de bits par caractères}) = 3,5 * (1+8+1 \text{ [si 8N1]}) = \text{le temps pour 35 bits}$ ).

L'ensemble du message doit être transmis de manière continue. Si un silence de plus de 1,5 fois le temps de transmission d'un mot intervient en cours de transmission, le destinataire du message considérera que la prochaine information qu'il recevra sera l'adresse du début d'un nouveau message.



8

### Message

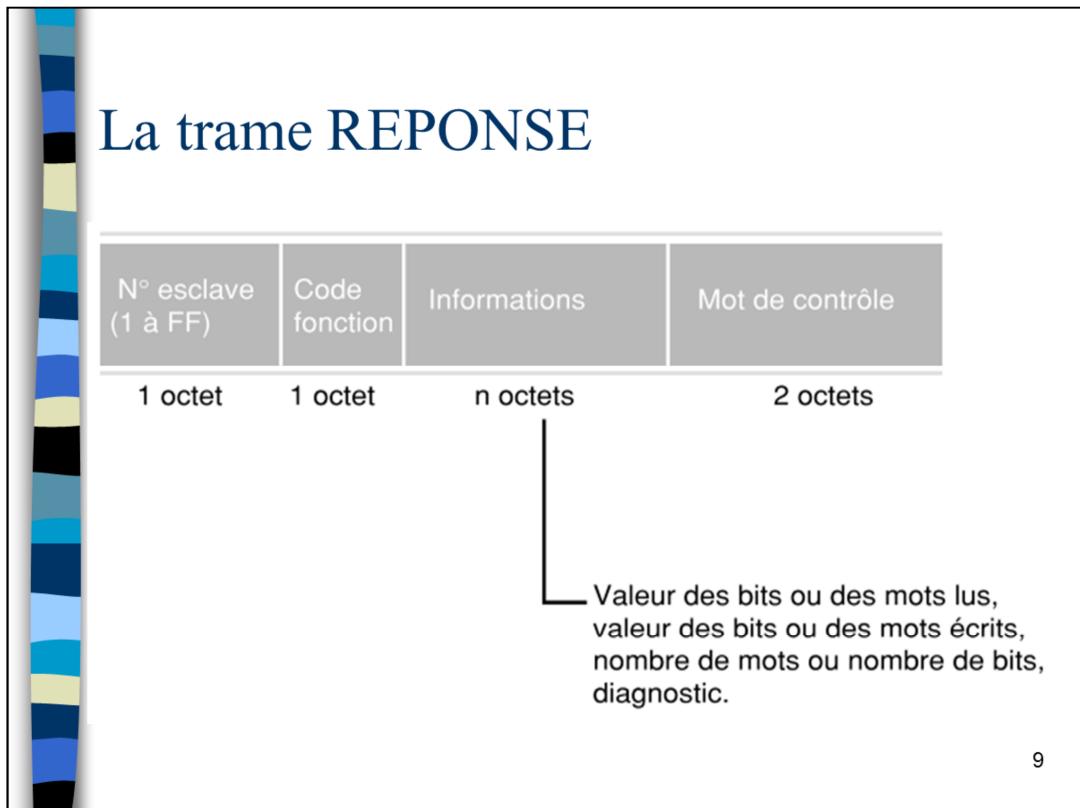
*Adresse de l'esclave : numéro compris entre 1 et 247 codés sur 1 octet.*

*Adresse 0 : diffusion générale*

*Instruction codée sur un octet (ex: 03)*

*Données : composées de plusieurs mots (adresse du premier mot (2 octets), nombre de mots (2 octets)).*

*CRC calculé par l'émetteur avant de le transmettre. Il est comparé à l'arrivée avec celui recalculé par le récepteur.*





## La réponse EXCEPTION

N° esclave (1 à FF)	1		CRC 16
1 octet	1 octet	1 octet	2 octets

PF \* | pf \*

Code d'exception :  
1. - Code fonction inconnu  
2. - Adresse incorrecte  
3. - Donnée incorrecte  
4. - Automate non prêt  
5. - Acquittement  
7. - Non acquittement  
8. - Défaut d'écriture  
9. - Chevauchement de zone

Code fonction reçu et  
bit de poids fort à 1.

10

Le maître émet une demande en indiquant :

- .....
- .....
- \*.....

Il calcule et ajoute le contenu du mot de contrôle (CRC 16).

Lorsque l'esclave reçoit le message de demande, il le range en mémoire, recalcule le CRC et le compare au CRC 16 reçu :

- \* Si le message reçu est incorrect (inégalité des CRC 16), l'esclave ne répond pas.
- \* Si le message reçu est correct mais que l'esclave ne peut le traiter (adresse erronée, donnée incorrecte...), il renvoie une réponse d'exception.



## Exemple de trame

Lecture par le maître des mots W450 à W456 sur l'esclave N°2 (ATV-28)

02	03	01 C2	00 07	XX XX
n° de l'esclave	instruction de lecture de N mots	450 est converti en code hexadécimal	7 mots	Valeur du CRC

Cette requête comporte 8 octets.

Une requête fait au plus 255 octets.

11

Note : un « mot » représente deux octets, soit

..... bits.

Le nombre maximum de mots à lire doit être inférieur à

.....

### Liste des instructions possibles

\$01 : Lecture de n bits de sortie consécutifs

\$02 : lecture de n bits d'entrées consécutifs

\$03 : Lecture de n registres internes consécutifs de 16 bits (ou mots)

\$04 : Lecture de n cartes d'entrées consécutives de 16 bits

\$05 : Ecriture d'un bit à une valeur donnée

....

\$0A : Demande de compte rendu de fonctionnement

...

## Exemples de trames

**Trame émise par le maître** : 04 03 00 02 0001 25 CA

- Adresse esclave : 04
- Code fonction 03 = lecture registre
- N° du registre de début de lecture : MSB : 00 et LSB : 02
- Nombre de registre de lecture : MSB : 00 et LSB : 01
- CRC : 25 CA

**Réponse de l'esclave avec erreur** : 04 83 02 01 31

- Adresse esclave : 04
- Code fonction : lecture avec MSB = 1 : 83
- Code erreur (n° registre) : 02
- CRC : 01 31

**Réponse de l'esclave sans erreur** : 04 03 02 02 58 B8 DE

- Adresse esclave : 04
- Code fonction : lecture registre : 03
- Nombre d'octets données : 02
- Données du registre 0002 : MSB 02 et LSB : 58
- CRC : B8 DE



## JBUS

- Protocole développé par les sociétés Renault et April après Modbus
- Utilisé essentiellement sur des équipements fabriqués par des sociétés françaises
- Très proche de Modbus

13

**Même structure hiérarchisée**

1 maître et plusieurs esclaves

**Même format d'échange**

7 ou 8 bits avec ou sans parité

1 bit de start et un bit de stop

**Même type d'échange**

Half-duplex

**Mêmes types de dialogue**

Diffusion générale (broadcast)

Échange avec un esclave

**Même format de trame**

**Même contrôle de redondance (CRC)**

**Même domaine d'utilisation**



## Différences entre Modbus et JBUS

### Décalage d'adresse

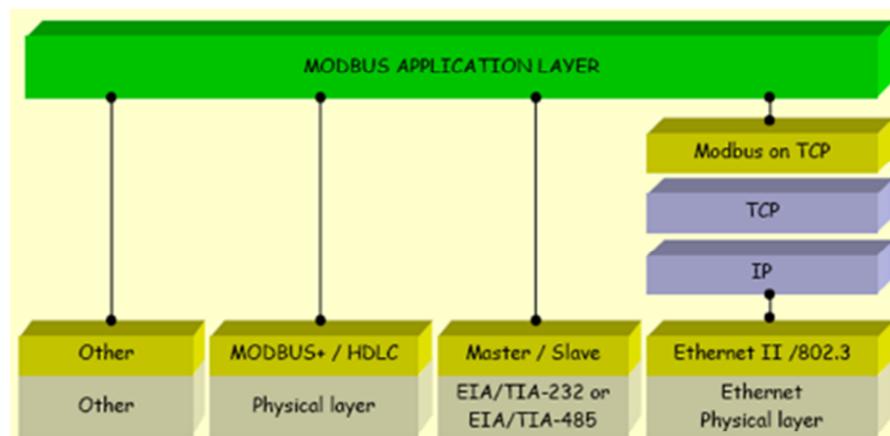
- Modbus : objet n, adresse n-1
- JBUS : objet n, adresse n

### Capacités (Modbus / JBUS)

- Nombre d'esclaves : 247 / 255
- Lire n bits : 1920 / 2000
- Lire n mots : 120 / 125
- Écrire n bits : 1920 / 1968
- Écrire n mots : 120 / 123



# Communication Modbus



15

## Principales caractéristiques techniques MODBUS série:

**Couche physique:** .....

**Câble** : paire torsadée

**Débit**: 9600 baud à 5 Mbaud

7 ou 8 bits, avec ou sans parité, 1 Start et 1 Stop

Échanges de type Half-duplex

**Topologies:** .....

**Méthode d'accès:** .....

**Mode de transmission:**

- RTU (information codée sur 1 octet), ex : valeur 43 se code 0x2B

- ASCII (information codée sur 2 octets au format ASCII) ex : valeur 43 se code 0x32, 0x42

**Equipements**: maître, esclave, passerelle, terminaisons, répéteurs

**Connecteurs recommandés**: Sub-D 9 points, RJ45

## Principales caractéristiques techniques MODBUS/TCP:

**Couche physique+liaison:** .....

**Débit**: 10-100 Mbaud

**Topologies**: étoile, anneau

Couche réseau+transport: TCP/IP

**Méthode d'accès:** .....

**Equipements**: client, serveur, passerelle, switch, hub

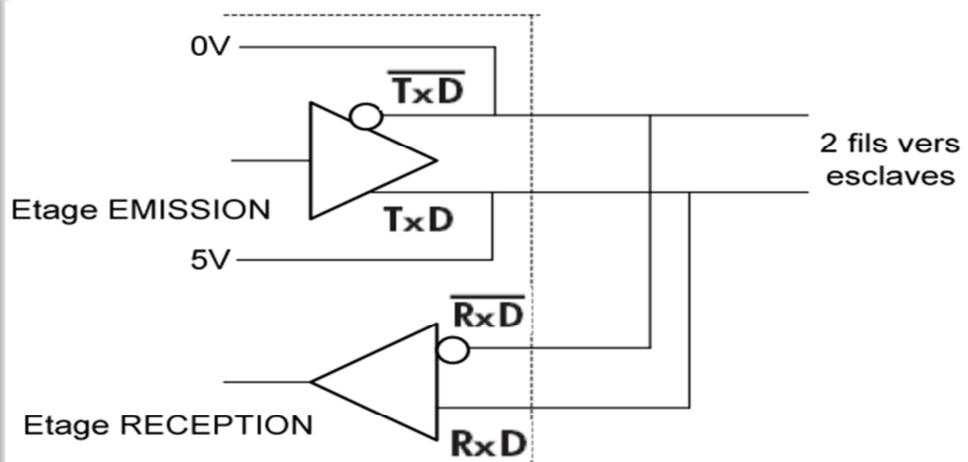
**Connecteurs recommandés**: RJ45 classique ou IP65

L'échange est purement de type maître/esclave avec MODBUS série (une requête doit être suivie d'une réponse, avant de pouvoir envoyer une nouvelle requête).

Avec MODBUS/TCP, les échanges sont de type client/serveur. Un client peut envoyer une requête vers un serveur, sans avoir forcément reçu la réponse à la requête précédente.



## Détail du câblage en RS-485



16

Ce type de liaison permet un débit élevé (maxi 10 mégabits par secondes) sur une distance pouvant aller jusqu'à 1200m.

Le même couple de fil transporte les signaux alternativement du maître vers l'esclave et réciproquement.

Plusieurs esclaves peuvent être interconnectés car

.....

.....