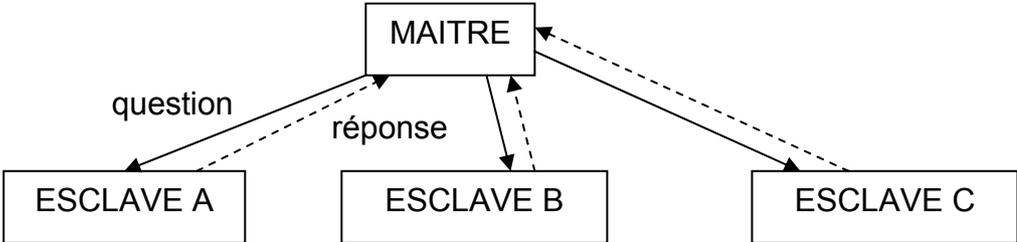


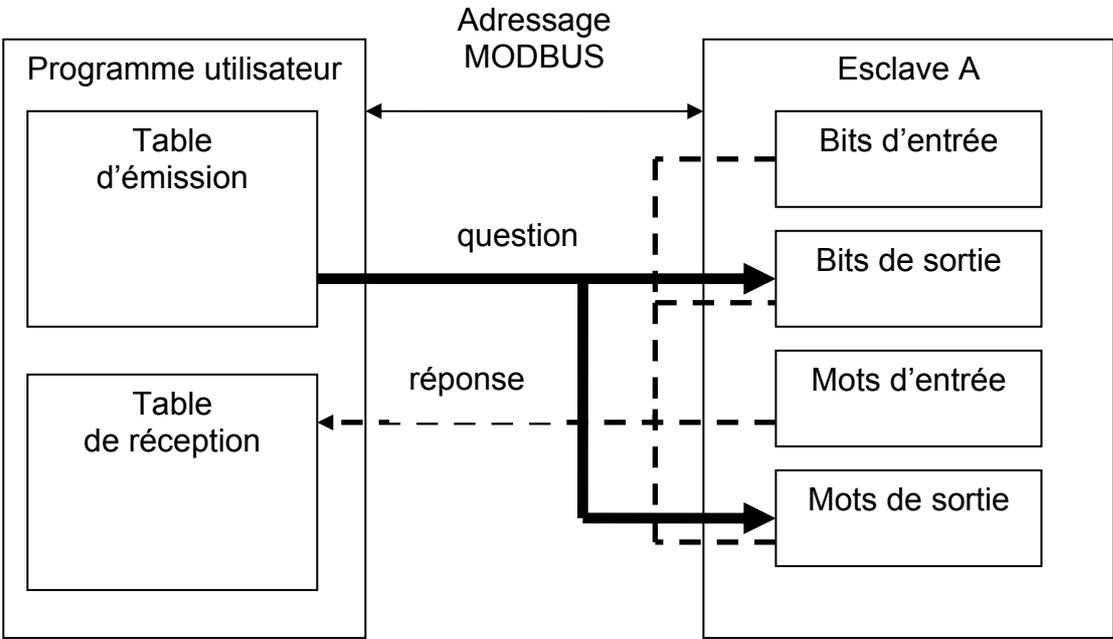
# Protocole MODBUS

Le protocole MODBUS consiste en la définition de trames d'échange.



Le maître envoie une **demande** et attend une **réponse**.  
Deux esclaves ne peuvent dialoguer ensemble.  
Le dialogue maître – esclave peut être schématisé sous une forme successive de liaisons point à point.

## Principe des échanges MODBUS

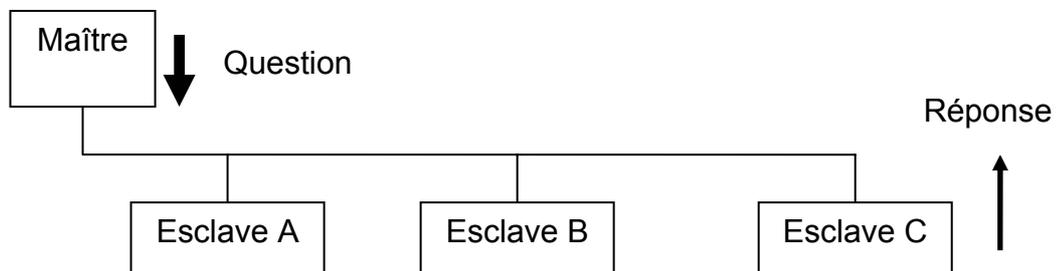


## Adressage

Les abonnés du bus sont identifiés par des adresses attribuées par l'utilisateur. L'adresse de chaque abonné est indépendante de son emplacement physique. Les adresses vont de 1 à 64 et ne doivent pas obligatoirement être attribuées de manière séquentielle. Deux abonnés ne peuvent avoir la même adresse.

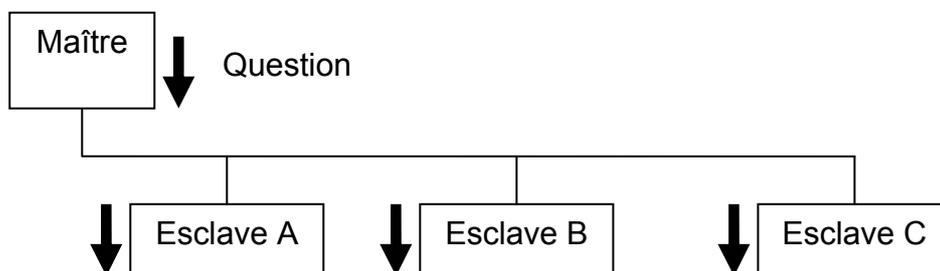
## Echange maître vers 1 esclave

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



## Echange Maître vers tous les esclaves

Le maître diffuse un message à tous les esclaves présents sur le réseau, ceux-ci exécutent l'ordre du message sans émettre une réponse.



## Trame d'échange question/réponse

### La question

Elle contient un code fonction indiquant à l'esclave adressé quel type d'action est demandé. Les données contiennent des informations complémentaires dont l'esclave a besoin pour exécuter cette fonction.

Le champ octets de contrôle permet à l'esclave de s'assurer de l'intégralité du contenu de la question.

|              |               |  |                 |
|--------------|---------------|--|-----------------|
| N° d'esclave | Code fonction | Information spécifique concernant la demande | Mot de contrôle |
| 1 octet      | 1 octet       | n octets                                     | 2 octets        |

### La réponse

|              |               |                |                 |
|--------------|---------------|----------------|-----------------|
| N° d'esclave | Code fonction | Données reçues | Mot de contrôle |
| 1 octet      | 1 octet       | n octets       | 2 octets        |

Si une erreur apparaît, le code fonction est modifié pour indiquer que la réponse est une réponse d'erreur.

Les données contiennent alors un code (code d'exception) permettant de connaître le type d'erreur.

Le champ de contrôle permet au maître de confirmer que le message est valide.

|              |               |                  |                 |
|--------------|---------------|------------------|-----------------|
| N° d'esclave | Code fonction | Code d'exception | Mot de contrôle |
| 1 octet      | 1 octet       | 1 octet          | 2 octets        |

## Format général d'une trame

Deux types de codage peuvent être utilisés pour communiquer sur un réseau Modbus. Tous les équipements présents sur le réseau doivent être configurés selon le même type.

**Type ASCII** : chaque octet composant une trame est codé avec 2 caractères ASCII (2 fois 8 bits).

| START       | Adresse      | Fonction     | Données      | LRC          | END                       |
|-------------|--------------|--------------|--------------|--------------|---------------------------|
| 1 caractère | 2 caractères | 2 caractères | n caractères | 2 caractères | 2 caractères<br>« CR LF » |

LRC : C'est la somme en hexadécimal modulo 256 du contenu de la trame hors délimiteurs, complétée à 2 et transmise en ASCII.

**Type RTU** (Unité terminale distante) : chaque octet composant une trame est codé sur 2 caractères hexadécimaux (2 fois 4 bits).

| START   | Adresse | Fonction | Données  | CRC      | END     |
|---------|---------|----------|----------|----------|---------|
| Silence | 1 octet | 1 octet  | n octets | 2 octets | Silence |

La taille maximale des données est de 256 octets.

Le mode ASCII permet d'avoir des intervalles de plus d'une seconde entre les différents caractères sans que cela ne génère d'erreurs, alors que le mode RTU permet un débit plus élevé pour une même vitesse de transmission.

**L'ensemble des informations contenues dans le message est exprimé en hexadécimal.**

**Le maître** s'adresse à l'esclave dont l'adresse est donnée dans le champ prévu à cet effet. Le code fonction indique à l'esclave le type d'action à réaliser. Exemple : lecture de registre, code de fonction (03)<sub>HEX</sub>, écriture dans un registre, code de fonction (10)<sub>HEX</sub>. Le champ de données est codé sur n mots en hexadécimal de 00 à FF, soit sur n octets. Selon le code fonction, le champ de données contient diverses informations complémentaires permettant à l'esclave de décoder le message (voir l'exemple plus bas). Dans le cas du mode RTU, le champ contrôle d'erreur CRC (Contrôle de Redondance Cyclique) contient une valeur codée sur 16 bits.

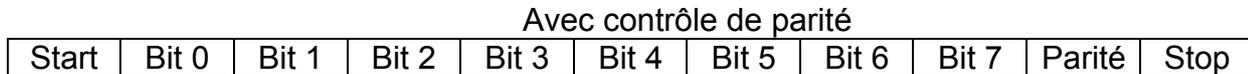
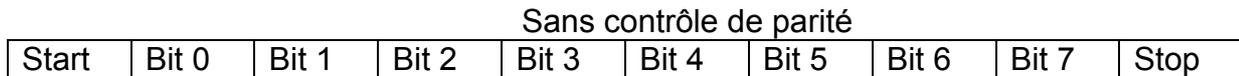
**Nota** : Le contrôle de parité peut dans certains cas être supprimé car d'autres contrôles d'échanges sont mis en œuvre (cas du contrôle CRC encore appelé contrôle par Checksum)

**L'esclave** renvoie sa réponse ; il place sa propre adresse dans le champ adresse afin que le maître puisse l'identifier.

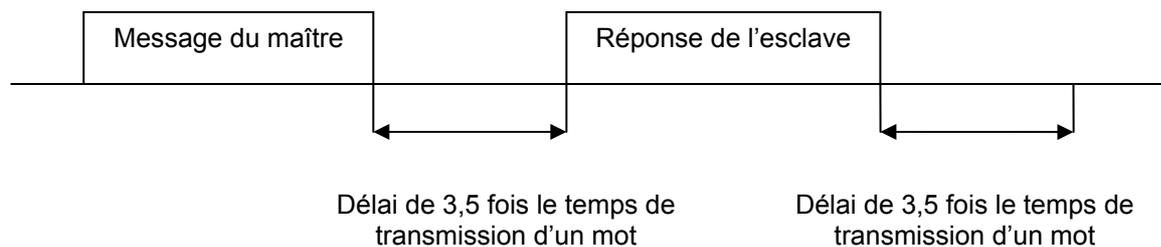
Il utilise ensuite le champ fonction pour indiquer si la réponse contient une erreur. Pour une réponse normale, l'esclave reprend le même code fonction que celui du message envoyé par le maître, sinon il renvoie un code erreur correspondant au code original avec son MSB à 1. Le champ de données contient diverses informations dépendant du code fonction. Le champ contrôle d'erreur contient une valeur codée sur 16 bits. Cette valeur est le résultat d'un CRC (Cyclical Redundancy Check) calculé à partir d'un message.

## Support de transmission

Chaque octet composant un message est transmis en mode RTU de la manière suivante :



Avant et après chaque message, il doit y avoir un silence équivalent à 3,5 fois le temps de transmission d'un mot.



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.

Le protocole MODBUS ne définit que la structure des messages et leur mode d'échange. On peut utiliser n'importe quel support de transmission RS 232, RS 422 ou RS 485, mais la liaison RS 485 est la plus répandue car elle autorise le « multipoints ».

## Exemple d'échange entre un maître et un esclave

**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

## Services supportés par MODBUS

MODBUS offre 19 fonctions différentes. Elles se caractérisent par un code fonction sur un octet (en hexadécimal). Tous les équipements ne supportent pas tous les codes fonction.

| Code  | Nature des fonctions MODBUS                         | TSX 37 |
|-------|---|--------|
| H'01' | Lecture de n bits de sortie consécutifs             | *      |
| H'02' | Lecture de n bits de sortie consécutifs             | *      |
| H'03' | Lecture de n mots de sortie consécutifs             | *      |
| H'04' | Lecture de n mots consécutifs d'entrée              | *      |
| H'05' | Ecriture de 1 bit de sortie                         | *      |
| H'06' | Ecriture de 1 mot de sortie                         | *      |
| H'07' | Lecture du statut d'exception                       |        |
| H'08' | Accès aux compteurs de diagnostic                   |        |
| H'09' | Téléchargement, télé déchargement et mode de marche |        |
| H'0A' | Demande de CR de fonctionnement                     |        |
| H'0B' | Lecture du compteur d'événements                    | *      |
| H'0C' | Lecture des événements de connexion                 | *      |
| H'0D' | Téléchargement, télé déchargement et mode de marche |        |
| H'0E' | Demande de CR de fonctionnement                     |        |
| H'0F' | Ecriture de n bits de sortie                        | *      |
| H'10' | Ecriture de n mots de sortie                        | *      |
| H'11' | Lecture d'identification                            | *      |
| H'12' | Téléchargement, télé déchargement et mode de marche |        |
| H'13' | Reset de l'esclave après erreur non recouverte      |        |