

# Cours ModBus

Thierry Vaira

La Salle Avignon

© v.1.0 - 9 décembre 2018



# Sommaire

## 1 Rappels

- Définition
- Modèle OSI

## 2 ModBus

- Présentation
- Couche Physique
- Couche Liaison
- ModBus TCP

# RLI

Un **réseau local industriel** (RLI) est un système de communication entre plusieurs équipements de type industriel (capteurs, automates, actionneurs, ...) dans une zone géographique limitée (un « terrain »).

On parle aussi de « **bus de terrain** » ou de « **réseau de terrain** ».

Il existerait plus de 2000 bus de terrain différents ! Les technologies les plus répandues sont : **Modbus**, Profibus, Interbus-S, ASI, Lonworks et bus CAN.

📌 Ce type de réseau est basé le plus souvent sur la **restriction du modèle OSI aux couches** : **Physique**, **Liaison** et éventuellement **Application**.

# Modèle OSI (1/3)

Un **réseau local industriel** (RLI) est basé le plus souvent sur la **restriction du modèle OSI à 3 couches** :

- la couche **Application** (qui peut être vide dans de nombreux réseaux)
- la couche **Liaison** qui doit assurer un transport fiable de quantité assez faible de données mais en respectant des contraintes "temps réel" (déterminisme)
- la couche **Physique** qui doit respecter des contraintes fortes liées à l'environnement (température, vibrations, ...)

# Modèle OSI (2/3)

Les raisons concernant l'absence des autres couches sont les suivantes :

- couche 3 (Réseau) : aucun besoin de routage dans les réseaux locaux industriels car les stations sont toutes connectées sur le même réseau physique
- couche 4 (Transport) : les messages sont très courts (contenu dans une seule trame) : pas besoin de segmentation
- couche 5 (Session) : les concepts de session ne sont pas supportés sur les RLI
- couche 6 (Présentation) : toutes les applications donnent le même sens à la définition d'une information : il n'y donc pas besoin de (re)présentation.

# Modèle OSI (3/3)

Couche OSI	Nom de la couche	Rôle de la couche		Format des données
7	Application	Interface entre le réseau et l'utilisateur	Elle assure tous les services qui peuvent simplifier la vie de l'utilisateur et qui ne sont pas assurés par les autres couches	Messages
2	Liaison de données	Méthodes d'accès au support	Aléatoires	CSMA/CD : Détection de collision
			Déterministes	CSMA/CA : Evitement de collision
				Maître Esclave
				Arbitre de bus
1	Physique	Sécurisation des échanges		Anneau à jeton
				En-tête de Trame (question, réponse, ..)
				Bits de Redondance (parité, checksum, CRC)
1	Physique	Codage de l'information	Tension	
		Topologie	Bus	
			Etoile	
			Anneau	
		Support de transmission (média)	Le cuivre	Câble coaxial
				Paire torsadée
			Fibre optique	Monomode
				Multimode
			Autres	Courant porteur
				Radio
				Infrarouge

# Modbus

**Modbus** est un **protocole de communication utilisé pour des réseaux d'automates programmables** (API).

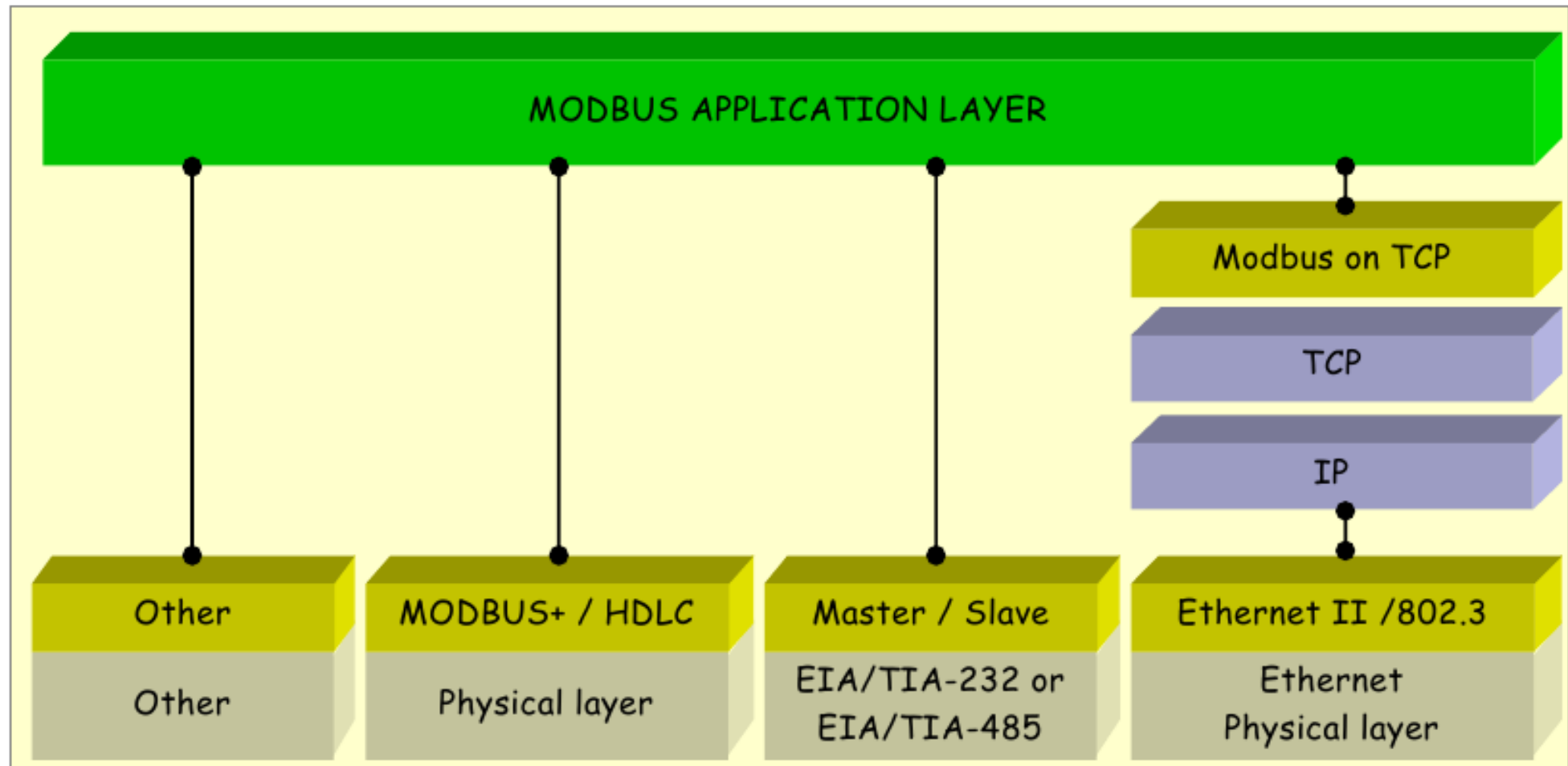
Il fonctionne sur le mode **maître/esclave** pour l'échange des trames.

Le protocole Modbus peut être utilisé :

- directement sur **une liaison série** de type RS-422 ou **RS-485** ou TTY (boucle de courant) avec des débits et des distances variables
- via **TCP/IP avec Ethernet** : on parle alors de Modbus TCP/IP ou **Modbus TCP**
- via **Modbus Plus (ou Modbus+)**. Modbus Plus est un réseau à passage de jetons à 1 Mb/s, pouvant transporter les trames Modbus et d'autres services propre à ce réseau.



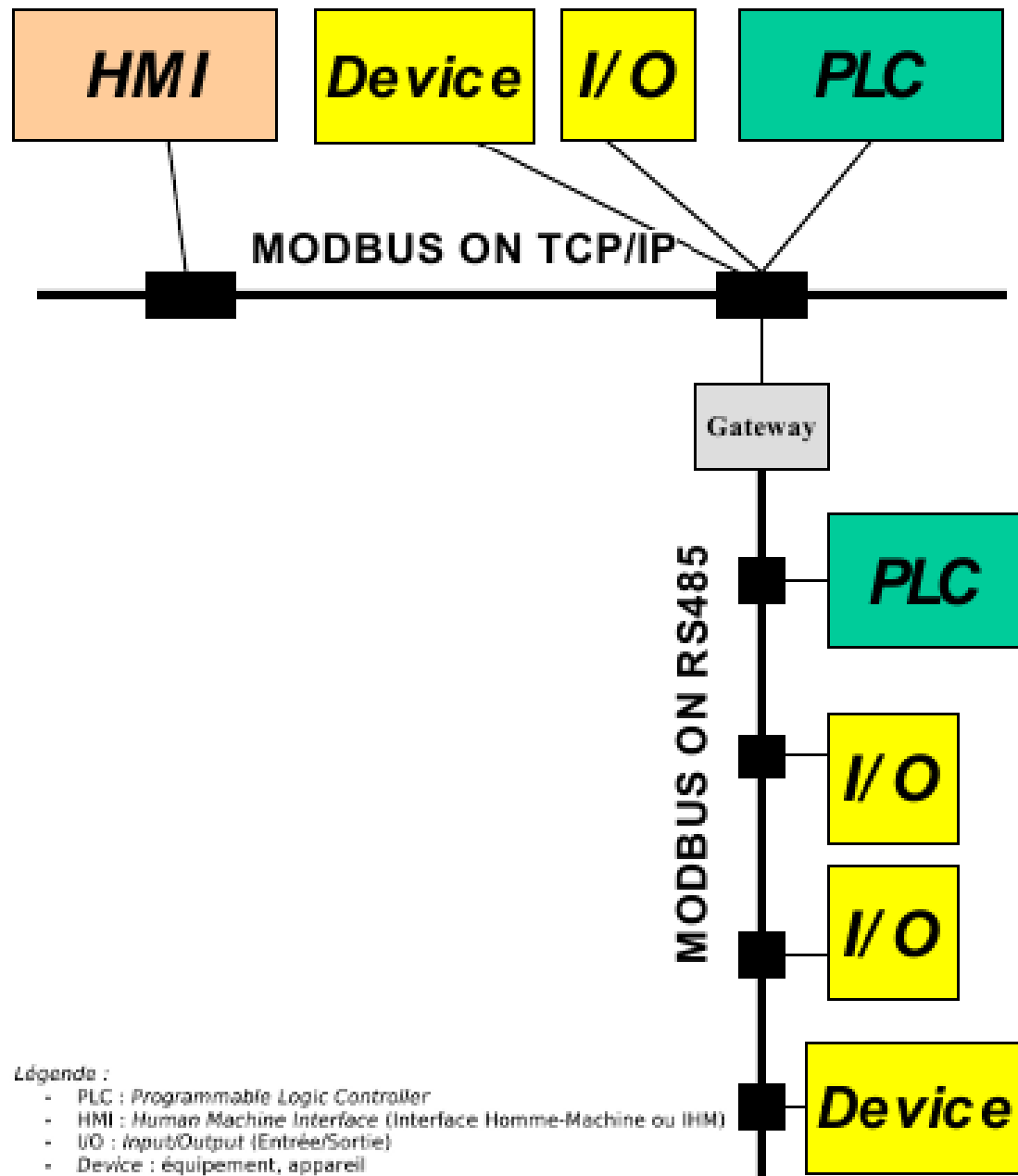
# Les différentes versions de Modbus



Selon des études récentes, Modbus TCP serait le protocole *Ethernet* Industriel le plus utilisé au monde.

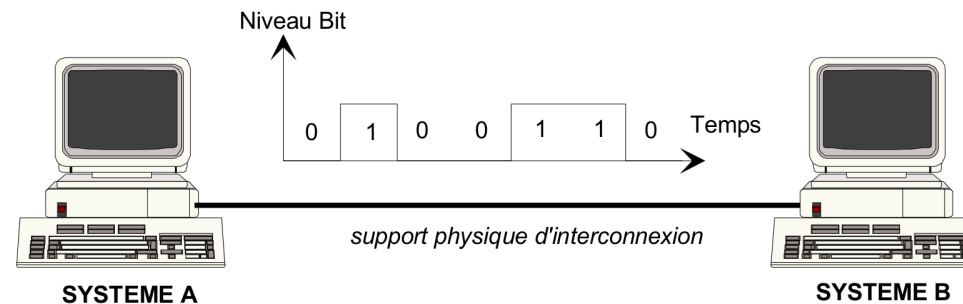


# Exemple de réseaux Modbus



# Mode d'exploitation

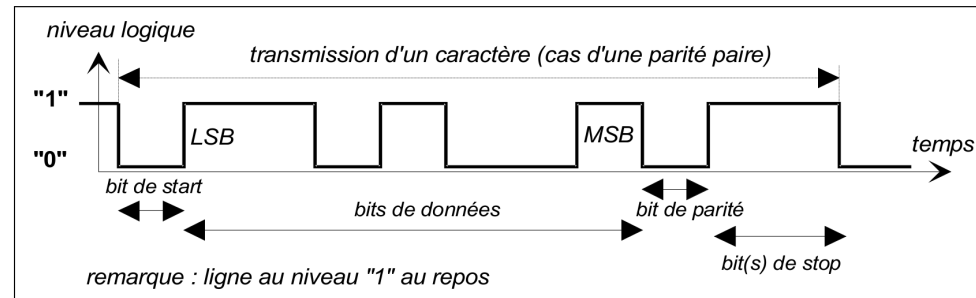
Principe d'une **liaison série** : les **n bits** sont transmis au rythme d'une **horloge** dont la valeur de période permet de définir la notion de **débit** (**vitesse de transmission**) égal au nombre de bits transmis par unité de temps (**bits/s**)



Les différents modes d'exploitation d'une liaison série :

- **simplex** : l'exploitation de la ligne se fait en mode **unidirectionnel**
- **half duplex** : l'exploitation de la ligne se fait en mode **bidirectionnel** mais pas simultanément
- **full duplex** : l'exploitation de la ligne se fait en mode **bidirectionnel simultané** sur le même support physique

# Trame asynchrone



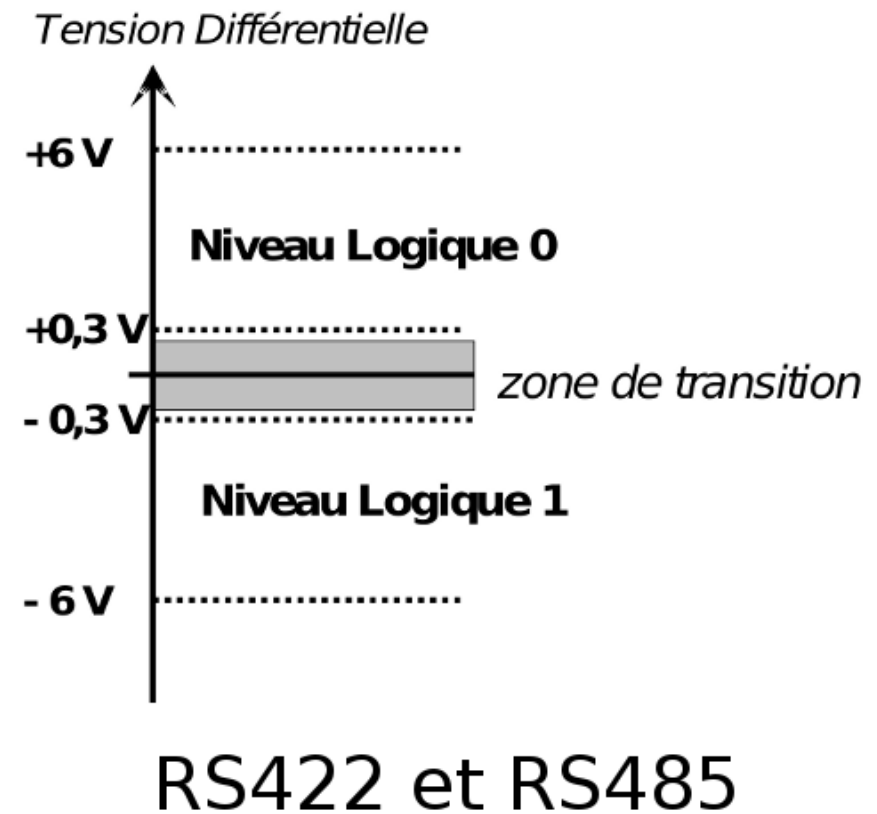
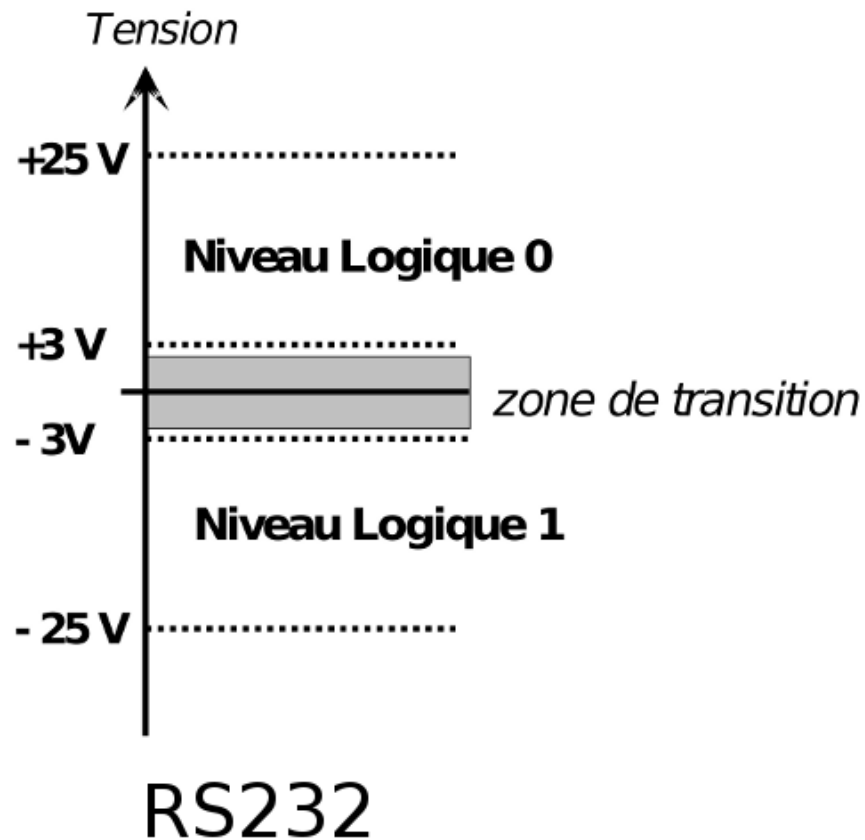
- bit de start : le front descendant indique au récepteur qu'il va devoir se synchroniser
- bits de données : entre 7 et 8 bits
- bit de parité (facultatif) : bit généré lors de l'émission et testé lors de la réception pour valider la conformité de la donnée. Si on utilise une parité paire (*even*) alors le nombre de bits (donnée + parité) doit être pair. Si on utilise une parité impaire (*odd*) alors le nombre de bits (donnée + parité) doit être impair
- bit(s) de stop : durée pour dissocier la fin d'émission du caractère courant du début (bit de start) du caractère suivant (1, 1.5 ou 2 bits de stop)

# Comparaison

- **RS232** (ou V24) : norme électrique caractérisant la transmission d'un signal sur un seul fil référencé par rapport à la masse (liaison point à point).
- **RS422** et **RS485** : norme électrique caractérisant la transmission d'un signal sur un support **différentiel** (liaison multi-point ou bus). Deux fils correspondant à des niveaux complémentaires sont utilisés pour coder l'information.

Spécifications	RS 232	RS 422	RS 485
Type de communication	Unipolaire	Différentiel	Différentiel
Connexions électriques minimales	3 fils Tx, Rx et masse	5 fils Paire Tx, Paire Rx et masse	3 fils Paire Tx/Rx, et masse
Nombre de transmetteurs et récepteurs alloués par la ligne	1 transmetteur 1 récepteur	1 transmetteur 31 récepteurs	32 transmetteurs 32 récepteurs
Longueur maximum de câble	16,5 m	1320 m	1320 m
Débit maximum	64 Kbits/s	10 Mbits/s	10 Mbits/s

# Codage de l'information



# La liaison RS-485

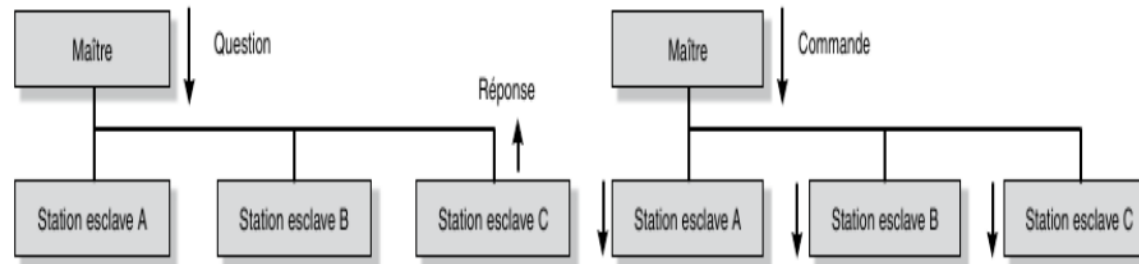
**EIA-485** (souvent appelée **RS-485**) est une **norme qui définit les caractéristiques électriques de la couche physique d'une interface numérique série** utilisée dans de nombreux réseaux industriels (**Modbus**, Profibus, ...).

Ses caractéristiques essentielles sont :

- liaison multi-point permettant d'interconnecter plusieurs dispositifs (jusqu'à 32 émetteurs et 32 récepteurs)
- bus informatique câblé avec 2 fils (en "half duplex") ou 4 fils (en "full duplex")
- distance maximale de l'ordre du kilomètre en mode différentiel (qui permet d'obtenir une meilleur tolérance aux perturbations)
- débit élevé jusqu'à 10Mbits/s

# Le mode maître/esclave

Le **maître** envoie une demande à un **esclave** et attend une réponse de celui-ci.



Les règles de fonctionnement sont les suivantes :

- Les esclaves sont identifiés par une adresse (sur 8 bits soit un octet).
- Aucun esclave ne peut envoyer un message sans une demande préalable du maître.
- Le dialogue entre les esclaves est impossible.
- Le maître peut diffuser un message à tous les esclaves présents sur le réseau (diffusion générale ou *broadcast*). Pour cela, il utilise l'adresse 0.

# Trames Modbus

Les trames sont de 2 types :

- mode RTU (*Remote Terminal Unit*) : les données sont sur 8 bits
- mode ASCII : les données sont codées en ASCII (il faut deux caractères pour représenter un octet, exemple 0x03 sera codé '0' et '3')

## La question

Elle contient un code fonction indiquant à l'esclave adressé le type d'action demandé.

Les données contiennent des informations complémentaires dont l'esclave a besoin pour exécuter cette fonction.

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

## Question :

N° station esclave	Code fonction + bit d'erreur	Information spécifique concernant la demande	Mot de contrôle
1 octet	1 octet	n octets	2 octets

## Réponse :

N° station esclave	Code fonction + bit d'erreur	Données transmises	Mot de contrôle
1 octet	1 octet	n octets	2 octets

## Réponse lors d'une erreur :

N° station esclave	Code fonction + bit d'erreur	Code d'exception	Mot de contrôle
1 octet	1 octet	1 octet	2 octets

## La réponse

Si une erreur apparaît, le code fonction est modifié pour indiquer que la réponse est une réponse d'exception (MSB=0 : pas d'erreur ; MSB=1 : erreur).

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

Code d'exception :

- 01 Fonction illégale (erreur sur le code fonction)
- 02 Erreur sur l'adresse du registre ou du coil
- 08 Erreur de transmission (suite au contrôle du CRC ou du Timing)



# Mot de contrôle

Le **mot de contrôle** d'une trame Modbus est un **code de vérification d'erreur** appelé **contrôle de redondance cyclique sur 16 bits ou CRC16**.

- Le CRC (*Cyclical Redundancy Check*) est calculé par l'émetteur avant d'être transmis.
- Le récepteur calcule aussi un CRC avec la trame reçue et le compare avec le CRC reçu : des valeurs différentes indiqueront une erreur dans la transmission du message.

Le CRC utilisé par Modbus est basé sur un calcul utilisant un OU EXCLUSIF (XOR).

# Codes de fonctions

MODBUS offre 19 fonctions différentes. Tous les équipements ne supportent pas tous les codes fonctions.

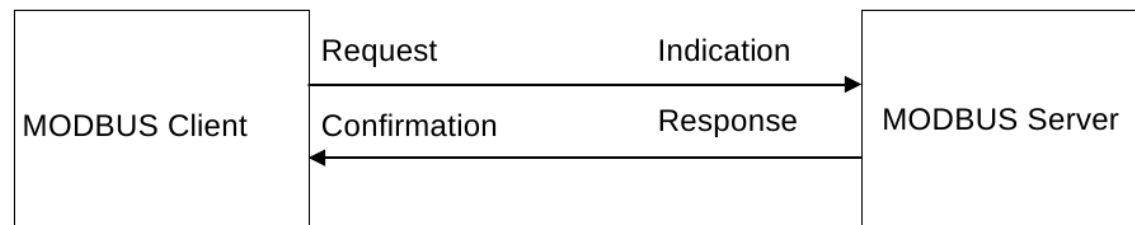
Code	Nature des fonctions MODBUS	TSX 37
H'01'	Lecture de n bits de sortie consécutifs	*
H'02'	Lecture de n bits de sortie consécutifs	*
H'03'	Lecture de n mots de sortie consécutifs	*
H'04'	Lecture de n mots consécutifs d'entrée	*
H'05'	Ecriture de 1 bit de sortie	*
H'06'	Ecriture de 1 mot de sortie	*
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	

# Client/Serveur en Modbus TCP

Évidemment la communication **Modbus TCP** est basée sur l'**architecture client/serveur**. Pour permettre l'établissement des connexions et l'échange de données entre équipements, le processus serveur Modbus TCP "écoute" sur le **port TCP 502**.

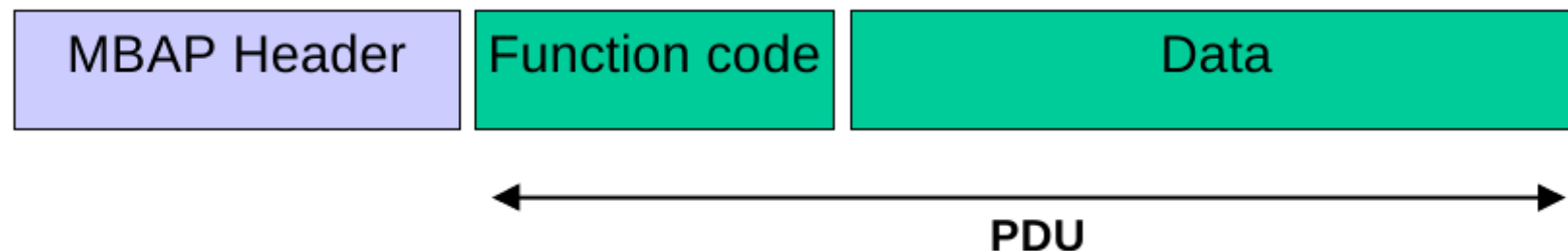
Le fonctionnement de base est le suivant :

- Le module client Modbus construit une requête sur la base des informations transmises par l'application
- Un module serveur Modbus est, quant à lui, chargé de recevoir les requêtes et de mettre en oeuvre des actions (de lecture et d'écriture notamment) afin d'y répondre.



# Protocole Modbus TCP

Le protocole Modbus définit une « unité de données de protocole », ou PDU (*Protocol Data Unit*), indépendante des autres couches de communication. L'encapsulation du protocole Modbus sur TCP/IP introduit un champ supplémentaire le *MBAP Header*.



# MBAP Header

Le contenu du *MBAP Header* est le suivant :

Fields	Length	Description -	Client	Server
Transaction Identifier	2 Bytes	Identification of a MODBUS Request / Response transaction.	Initialized by the client	Recopied by the server from the received request
Protocol Identifier	2 Bytes	0 = MODBUS protocol	Initialized by the client	Recopied by the server from the received request
Length	2 Bytes	Number of following bytes	Initialized by the client ( request)	Initialized by the server ( Response)
Unit Identifier	1 Byte	Identification of a remote slave connected on a serial line or on other buses.	Initialized by the client	Recopied by the server from the received request

# L'adressage et le routage en Modbus TCP

Le champ "*Unit Identifier*" est utilisé pour le routage lorsqu'on s'adresse à un périphérique sur un réseau série Modbus derrière une passerelle (*gateway*) :

- Dans ce cas, le champ "*Unit Identifier*" contient l'adresse esclave de l'appareil distant. Et l'adresse IP identifie la passerelle et non l'esclave.

Sinon le champ "*Unit Identifier*" est inutile et la valeur 0xFF doit être utilisée. C'est le cas pour les équipements sur le réseau TCP/IP qui sont identifiables par leur adresse IP.

# Capture Modbus TCP

```

Source      Destination  Protocol Info
10.98.0.254 10.98.0.3  Modbus/TCP query [ 1 pkt(s)]: trans:1;
unit: 5 func: 16: Write Multiple Registers
Ethernet II, Src: (00:16:d3:64:8e:14), Dst: (00:20:4a:b2:38:6c)
Internet Protocol, Src: (10.98.0.254), Dst: (10.98.0.3)
Transmission Control Protocol, Src Port: (30261), Dst Port: (502),
Seq: 0, Ack: 0, Len: 65
Modbus/TCP
  transaction identifier: 1
  protocol identifier: 0
  length: 59
  unit identifier: 5
  Modbus
    function 16: Write Multiple Registers
    reference number: 0
    word count: 26
    byte count: 52
    Data
      00 20 4a b2 38 6c 00 16 d3 64 8e 14 08 00 45 00 . J.8l...d....E.
      00 69 26 56 40 00 80 06 be 74 0a 62 00 fe 0a 62 .i&V@....t.b...b
      00 03 76 35 01 f6 87 5a 7a 9b 04 2d 9a b8 50 18 ..v5...Zz..-..P.
      ff ff 16 20 00 00 00 01 00 00 00 3b 05 10 00 00 ... .....;....

      -----
      00 1a 34 24 46 31 24 4d 31 24 4c 30 34 49 6e 66 ..4$F1$M1$L04Inf
      -----
      6f 72 6d 61 74 69 6f 6e 20 76 6f 79 61 67 65 75 ormation voyageu
      -----
      72 20 3a 20 6c 69 67 6e 65 20 43 20 65 6e 20 70 r : ligne C en p
      -----
      61 6e 6e 65 24 46 30
      -----
  
```

----- ModbusTCP  
 ---- Modbus

# Encapsulation Modbus TCP

