Exposé : Introduction aux Réseaux Locaux Industriels (RLI)

Sommaire :

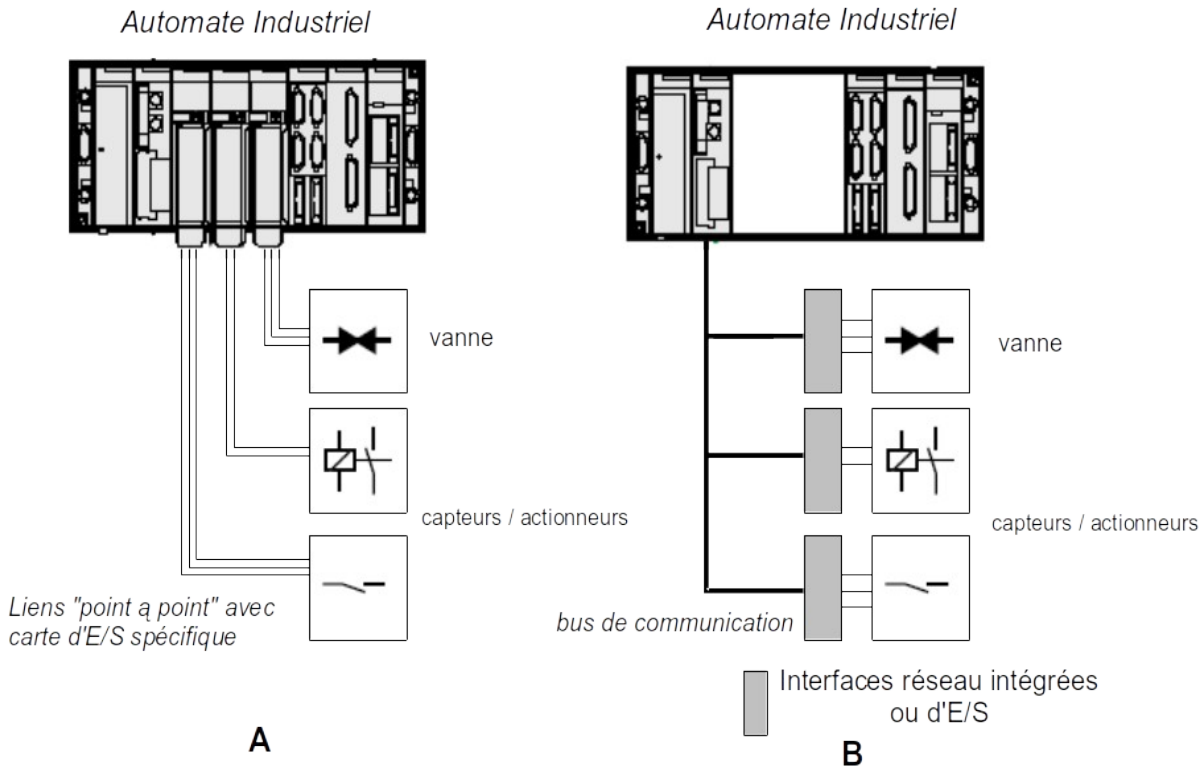
I - Introduction aux RLI

I.1. Définitions

De nombreux termes qualifient les moyens de transmission de l'information en milieu industriel :

- **Réseau de terrain** ;
- **Bus de terrain** ;
- **Réseau local industriel** ;
- **Réseau capteurs / actionneurs**...

Tous tendent à permettre l'échange de données entre **capteurs**, **actionneurs** et **systèmes informatiques** (API, PC industriels, E/S déportées, ...) gérant ces **Entrées/Sorties**. Le câblage traditionnel des Entrées/Sorties (**figure A**) a été peut à peut remplacé par un **bus de communication** (**figure B**) pour délocaliser ces dernières :



I.2. Les RLI dans le modèle OSI de l'ISO

Le **modèle OSI** (**O**pen **S**ystem **I**nterconnection) définit **7 couches** pour décrire toute communication. Ces couches sont **hiérarchisées**, et chaque couche, du niveau le plus bas (**1**) au niveau le plus haut (**7**), est soumise à la couche qui lui est immédiatement supérieure :

Numéro de la couche	Nom de la couche	Rôle de la couche
7	**Application**	Elle définit les mécanismes communs aux applications réparties et la signification des informations échangées.
4	**Transport**	Elle fournit les moyens de transporter l'information entre deux entités situées dans des systèmes différents.
3	**Réseau**	Elle réalise l'acheminement des informations dans le réseau, entre entités non directement reliées.
2	**Liaison de données**	Elle assure le transfert fiable de l'information entre entités physiquement reliées.
1	**Physique**	Elle décrit les interfaces électriques et mécaniques pour l'échange des informations sur le support physique.

Dans le cas des **R**éseaux **L**ocaux **I**ndustriels (**RLI**), seulement 3 couches sont utilisées :
- **Couche 1** : couche « **Physique** » ;
- **Couche 2** : couche « **Liaison de données** » ;
- **Couche 7** : couche « **Application** ».

II - Les différentes couches des RLI

II.1. La couche 1 « Physique »

La couche « **Physique** » a pour mission d'assurer la transmission physique de bits. Elle n'a pas à s'intéresser à la signification des données qui lui sont confiées, les autres couches sont là pour ça !

II.2. La couche 2 « Liaison de données »

La couche « **Liaison de données** » doit permettre de :
- **Sécuriser les échanges** c'est-à-dire palier les éventuelles erreurs de transmission physique. Elle ajoute à chaque donnée émises des champs binaires (parité, checksum, CRC etc.) qui permettront le contrôle à la réception ;
- **Décider des moments d'émission** en respectant les règles en vigueur baptisées « **méthodes d'accès** ».

Il existe deux catégories de « **méthodes d'accès** » différentes :
- Méthodes d'accès **déterministes** : Dans cette catégorie, aucune station ne peut s'exprimer sans avoir reçu l'autorisation préalable d'une station particulière baptisée « **Maître** » ou « **Arbitre de bus** ». Les émissions simultanées baptisées « **collisions** » n'existent pas. Les **principales méthodes d'accès déterministes** sont **Maitre/Esclaves** (Protocole **Modbus**) et **Arbitre de bus** ;

- Méthodes d'accès **aléatoires** : L'objectif de ces méthodes est de permettre à toute station de s'exprimer quand bon lui semble. Chaque station doit intégrer les mécanismes pour **détecter les collisions et s'en affranchir**. Les **principales méthodes d'accès aléatoires** sont **CSMA/CD - Carrier Sense Multiple Access with Collision Detection** (cas d'**Ethernet filaire**) et **CSMA/CA - Carrier Sense Multiple Access with Collision Avoidance** (cas d'**Ethernet sans fil**).

II.3. La couche 7 « Application »

La couche « **Application** » constitue la véritable **interface entre le réseau et l'utilisateur**.

Sa situation frontale la rend par définition totalement incontournable. Cette couche va donc comporter tous les services qui peuvent simplifier la vie de l'utilisateur et qui ne sont pas déjà assurés par les autres couches.

III - Évolutions des RLI

Actuellement l'**interconnexion de capteurs hétérogènes** est réalisée en majorité par les **réseaux de terrain** de type **Modbus**, **CAN**, **Profibus**, **FIP**, **AS-i**, etc....

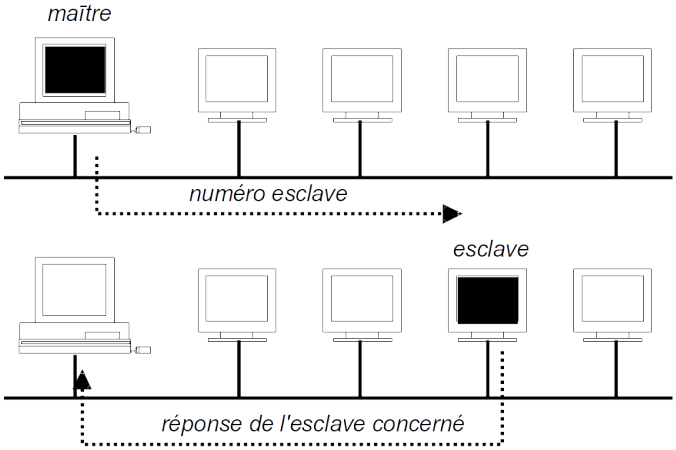
Les composants sont de plus en plus intelligents (ex : modules d'E/S ou vannes directement connectables sur réseau local industriel) et ils intègrent des services de communication de haut niveau.

De plus en plus de composants sont équipés de **services Web** (**http**, ...) et leur configuration peut s'opérer par exemple via un **client http**. Ces nouveaux produits nécessitent donc des réseaux locaux industriels et des réseaux de terrains de plus en plus performants.

IV - Protocole Modbus

IV.1. Présentation de Modbus

Modbus est un protocole de dialogue permettant d'assurer la communication entre deux (ou plusieurs) stations dans un réseau local industriel de type **Maître / Esclaves**.

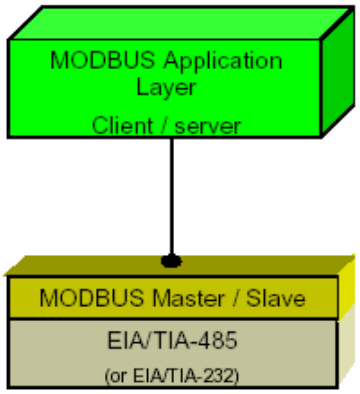


Le protocole **Modbus** est très utilisé pour les communications des automates programmables industriels et des variateurs de vitesse des moteurs électriques.

IV.2. Modbus et le modèle OSI

Comme dans tout les **R**éseaux **L**ocaux **I**ndustriels (**RLI**), seules les **couches 1**, **2** et **7** (**Physique**, **Liaison de données** et **Application**) du modèle **OSI** sont implémentés par **Modbus** :

Layer	ISO/OSI Model	
7	Application	MODBUS Application Protocol
6	Presentation	Empty
5	Session	Empty
4	Transport	Empty
3	Network	Empty
2	Data Link	MODBUS Serial Line Protocol
1	Physical	EIA/TIA-485 (or EIA/TIA-232)



IV.3. La couche « Liaison de données » du protocole MODBUS

Le protocole **Modbus** utilise une **méthode d'accès** de type **Maître / Esclaves**. L'unique **maître** du réseau est à l'initiative de tous les échanges entre **esclaves**.

Deux types de dialogue sont possibles :
- le **maître** parle à **un esclave** et attend sa réponse (**unicast** mode) ;
- le **maître** parle à l'**ensemble des esclaves**, sans attente de réponse : diffusion générale (**broadcast** mode).

Les **échanges** sont donc du type **half-duplex**. Il ne peut y avoir sur la liaison qu'un seul équipement en train d'émettre. Aucun esclave ne peut envoyer un message sans une demande préalable du maître. Le dialogue entre les esclaves est impossible.

Une liaison **série multipoints** de type **RS485** (ou **RS232** dans le cas d'une liaison **point à point**) relie le **maître** et les **esclaves**.

Le **format des échanges** est le suivant :
- 9600 ou 19200 bits / seconde ;
- 8 bits de données, pas de parité ou parité paire, 1 bit de stop.

L'espace d'adressage du protocole **Modbus** comporte **256 adresses** :

0	**De 1 à 247**	**De 248 à 255**
Adresse de Broadcast (diffusion)	**Adresses individuelles des esclaves**	**Réservées**

L'**adresse 0** est réservé pour la **diffusion** (**broadcast**). Tous les esclaves doivent reconnaître l'adresse de **broadcast**.

Il existe deux modes de transmission :
- Mode **RTU** (**R**emote **T**erminal **U**nit)
- Mode **ASCII** (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange)

Comme tous les esclaves **Modbus** supportent le mode **RTU** par défaut et que le mode **ASCII** est en option, on détaillera dans ce qui suit uniquement le mode **RTU**.

Le maître envoie un **message** (une **trame**) constitué de la façon suivante :

Adresse	Fonction	Données	Contrôle
1 octet	*1 octet*	*0 à 252 octets*	*2 octets*

- **Adresse** de l'esclave concerné, pour établir la liaison avec lui. Numéro compris entre **1** et **247** codé sur **1 octet**. Le numéro **0** indique que tous les esclaves sont concernés (**diffusion**) ;
- **Fonction à réaliser** sur l'esclave, codé sur **1 octet** (lecture, écriture, ...) ;
- **Données** impliquées dans la fonction à réaliser. Ce champ être composé de plusieurs mots, par exemple, adresse du premier mot (2 octets), puis le nombre de mots (2 octets) ;
- **Contrôle**, mot de contrôle codé sur 2 octets de type redondance cyclique (**CRC**) calculé sur l'ensemble du message et destiné à assurer l'intégrité de l'échange.

Les **fonctions Modbus** disponibles sont :

Fonction	Code Fonction associé
Lecture de **n bits** de **sortie**	**0x01**
Écriture de **n bits** d'**entrée**	**0x02**
Lecture de **n mots** de **sortie**	**0x03**
Lecture de **n mots** d'**entrée**	**0x04**
Écriture d'**un bit** de **sortie**	**0x05**
Écriture d'**un mot** de **sortie**	**0x06**
Lecture rapide / Diagnostic	**0x07**
Écriture de **n bits** de **sortie**	**0x0F**
Écriture de **n mots** de **sortie**	**0x10**

Exemple de trame Modbus envoyée par le maître :

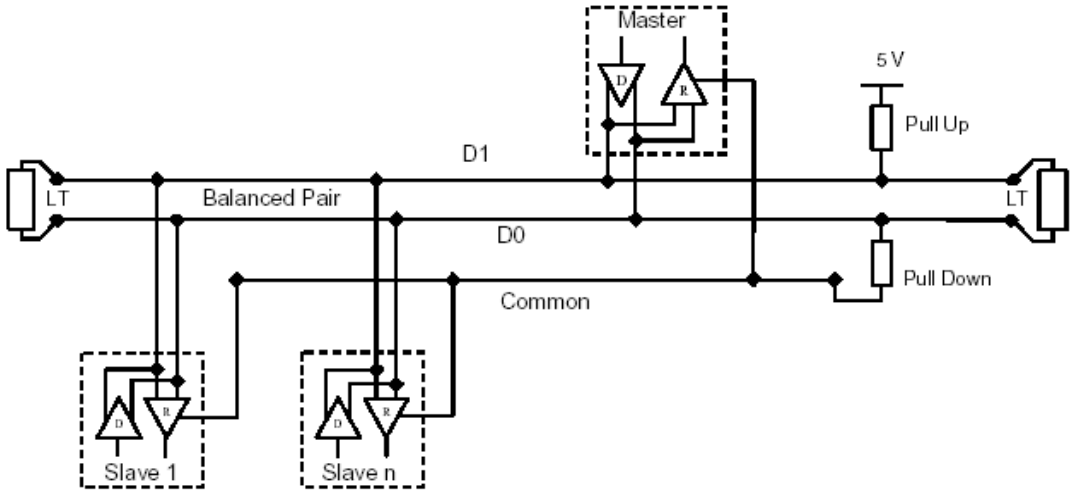
Lecture par le maître des mots **W450** à **W456** sur un variateur de vitesse de type **ATV-28** (esclave n°**2**). Cette requête comporte 8 octets :

02	**03**	**01 C2**	**00 07**	**A4 3B**
n° de l'esclave	instruction de lecture de N mots	450 est converti en code hexadécimal	7 mots à lire	Valeur du CRC

IV.4. La couche « Physique » du protocole Modbus

La **bus** utilisée respecte la norme **RS 485**. Cela permet de réaliser des communications **point à point** et **multipoints** sur **2 fils** (**1 paire**). Pour une communication **point à point**, le bus utilisé peut respecter la norme **RS 232**.

Le **bus** est constitué d'**une paire différentielle** servant à l'émission et à la réception. Un troisième conducteur peut être connecté sur le bus : le **commun** ou la **masse** :



Remarque : Le **bus RS485** permet d'avoir une longue distance de transmission (jusqu'à **1000 m** à **100 Kbps**) et d'un grand nombre de nœuds (**32** émetteurs et **43** récepteurs). Pour les distances importantes, il peut être nécessaire d'ajouter une **résistance terminale** (**120 Ohms**) aux extrémités du **bus** pour empêcher le signal d'être réfléchi.

V - Ethernet Industriel

V.1. Introduction

Les **R**éseaux **L**ocaux **I**ndustriels (**RLI**) permettent de résoudre le problème de l'i**nterconnexion de capteurs hétérogènes**, mais on peut leur reprocher plusieurs **inconvénients** notables :

- la **connexion** de ces réseaux de capteurs à un **système informatique** s'effectue par l'intermédiaire de **cartes d'interface propriétaires** dont le **coût n'est pas toujours avantageux** ;
- leur **mise en œuvre** nécessite la **connaissance approfondie** de la topologie et de l'adressage.

Or on peut remarquer qu'en parallèle le réseau **Ethernet** est devenu un standard de fait pour l'interconnexion des PC **systèmes informatiques** que ce soit en milieu industriel ou même domestique. Les composants d'interface sont devenus bon marché (**cartes Ethernet**), et le protocole **IP** (**I**nternet **P**rotocol) est bien connu et documenté et propose des capacités d'adressage intéressantes. Les protocoles de plus haut niveau s'appuyant sur **IP** sont eux-aussi standardisés (**HTTP**, **FTP**, **SMTP**, **SNMP** ...).

Un **réseau de capteurs** s'appuyant sur **Ethernet/IP** présente donc de **multiples avantages** :

- l'interfaçage capteurs / systèmes informatiques est meilleur marché (jeu de la concurrence) et indépendant d'un fournisseur ;
- les protocoles sont parfaitement normalisés et connus ;
- l'utilisation de protocoles de plus haut niveau permet d'envisager de nouvelles applications;
- il est possible d'utiliser le câblage réseau informatique existant dans un bâtiment industriel ou d'habitation sans changer de type de support.

Dans un **réseau de capteurs** utilisant **Ethernet** sous **IP**, les informations circulent sous forme de **datagrammes**, c'est-à-dire de paquets encapsulant les données à transmettre.

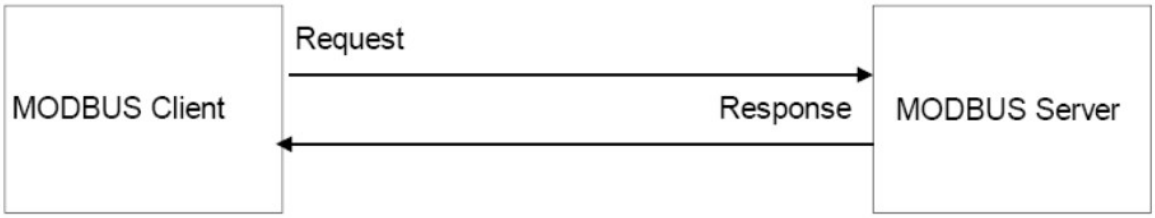
Le protocole de couche « **Application** » utilisé s'appuiera sur un des 2 protocoles de couche « **Transport** » suivants :

- **UDP** (**U**ser **D**atagram **P**rotocol) : Protocole simple de transmission d'informations en un seul datagramme sans acquittement ;
- **TCP** (**T**ransmission **C**ontrol **P**rotocol) : Protocole de transmission d'informations en plusieurs datagrammes avec acquittement par le destinataire.

V.2. Modbus/TCP

Modbus/TCP est la variante « **encapsulée** » dans **TCP/IP** du protocole **Modbus**. **Modbus/TCP** est une spécification ouverte au public basée sur un mode de communication **clients / serveur**. Les temps de réponse varient de 0.2 s à 1 s. Il ne sera donc pas employé pour des applications nécessitant de faibles temps de réponse mais permet de couvrir une majeure partie des besoins de communication industrielle.

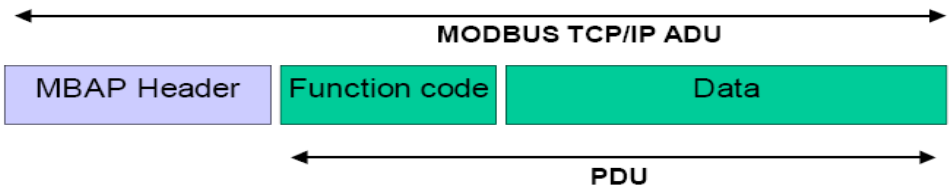
Le modèle **Client-Serveur** est basé sur 2 types de messages :



- **Request** : Message envoyé par le client pour initier une transaction ;
- **Response** : Message de réponse envoyé par le serveur.

Remarque : Le **port TCP** d'écoute réservé au dialogue **Modbus/TCP** est le port **502**.

Le message (niveau application) **Modbus / TCP** (**ADU**) a la structure suivante :



Le **MBAP** (ModBus Application Protocol) **header** a la structure suivante :

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

La partie **PDU** correspond à la **trame Modbus sur liaison série sans le champ adresse** et **sans le champ de contrôle (CRC)** :

fonction	**Données**
1 octet	*0 à 252 octets*

- **fonction** : instruction ou fonction à réaliser sur l'esclave, codé sur 1 octet (lecture, écriture, ...) ;
- **Données** : données impliquées dans la fonction à réaliser. Elle peut être composée de plusieurs mots, par exemple, adresse du premier mot (2 octets), puis le nombre de mots (2 octets).

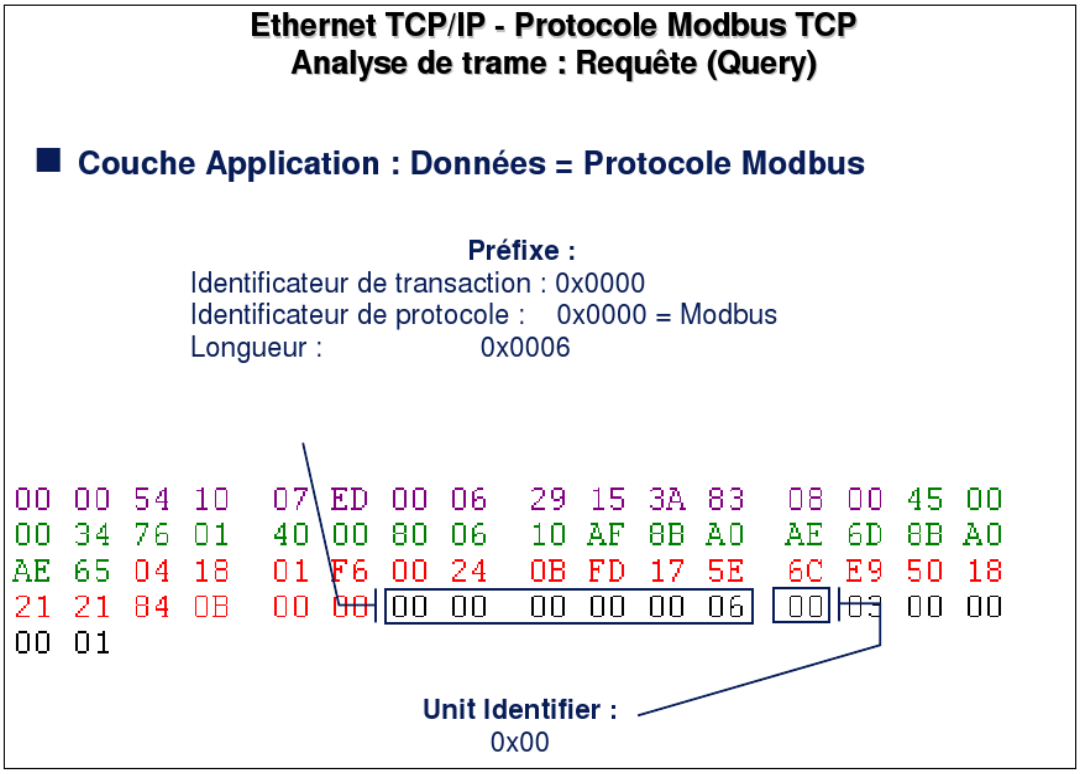
Le serveur peut retourner 2 types de **réponse** :

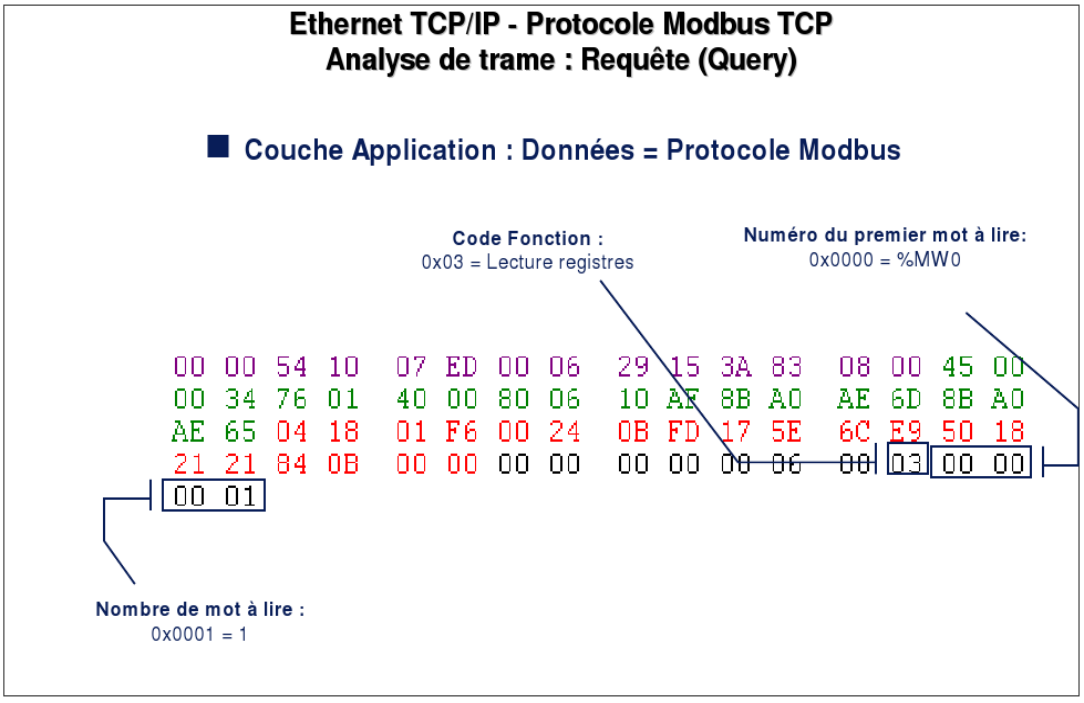
- **Réponse positive** : Dans ce cas, le **code fonction** de la **réponse** est identique au **code fonction** de la **question** ;
- **Réponse d'anomalie** : Le serveur retourne une **réponse d'anomalie** quand il est incapable d'effectuer la demande qui lui est adressée. Le format d'une réponse d'anomalie (**PDU**) est le suivant :

Code réponse	**Code d'erreur (Exception Code)**
Code Fonction + 0x80	**cf. tableau ci-dessous**
1 octet	*1 octet*

Exception Code	MODBUS name	Comments
01	Illegal Function Code	The function code is unknown by the server
02	Illegal Data Address	Dependant on the request
03	Illegal Data Value	Dependant on the request
04	Server Failure	The server failed during the execution
05	Acknowledge	The server accepted the service invocation but the service requires a relatively long time to execute. The server therefore returns only an acknowledgement of the service invocation receipt.
06	Server Busy	The server was unable to accept the MB Request PDU. The client application has the responsibility of deciding if and when to re-send the request.
0A	Gateway problem	Gateway paths not available.
0B	Gateway problem	The targeted device failed to respond. The gateway generates this exception

Exemple de trame requête Modbus / TCP





Pour la **réponse**, le champ indiquant « **numéro du premier mot à lire** » (*sur deux octets*) devient « **nombre d'octets lus** » (*sur un octet*) et celui « **nombre de mots à lire** » (*sur deux octets*) devient « **valeur du mot lu** » (*sur deux octets*).