2.1 Introduction

Le protocole de communication Modbus de Gould Modicon est un réseau de type maître-esclave très répandu au point qu'il n'existe quasiment aucun constructeur de systèmes de pilotage d'automatismes qui ne propose de couplage Modbus. Dans la pratique, Modbus permet d'interconnecter des matériels aussi hétérogènes que des automates programmables industriels, des variateurs de vitesse pour moteurs, des capteurs et des calculateurs. Ce protocole est caractérisé par une forme et un contenu de messages simples et universels.

2.2 Historique

1979: Création de Modbus par MODICON (Modular Digital Controller).
1994: Modicon fusionne avec Schneider (Télémécanique / 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.

2.3 Intégration dans le modèle OSI

Le rapprochement Modbus et modèle OSI s'effectue au niveau des couches physiques, liaison de donnée et application [16].

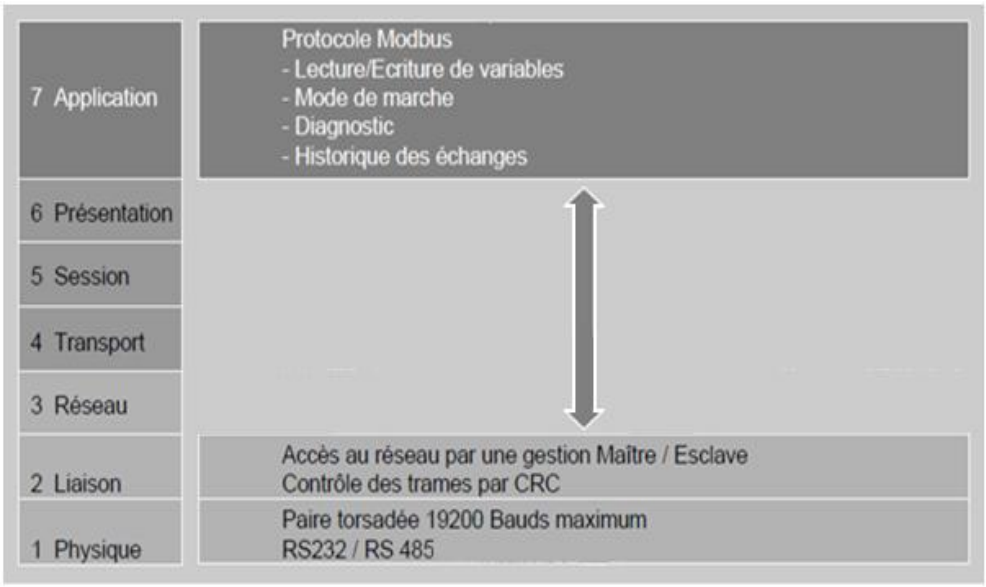

Figure 2.1 *Intégration de Modbus dans le modèle OSI.*

2.4 Architecture du réseau Modbus

Le réseau Modbus est un réseau composé d'un seul maître et d'esclaves, qui communiquent par une liaison série asynchrone [17].

On peut utiliser n'importe quel support de transmission RS 232 ou RS 485, mais la liaison RS 485 est la plus répandue car elle autorise le mode « multipoints ».

2.4.1 La liaison RS 232

La liaison RS 232 est une liaison point à point sur deux fils, permet un début de 20 k bit/s sur une distance inférieure à 15 mètres. Cette liaison sera utilisée pour l'initiation au fonctionnement du réseau [18].

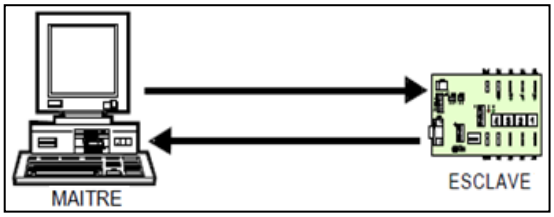

Figure 2.2 *La liaison série type RS232.*

2.4.2 La liaison RS 485

Dans le cas d'un réseau réel, le support de transmission est une liaison multipoint half duplex sous la norme RS485 caractérisée par une bonne immunité aux parasites [18]. Le début est de 100 k bit/s sur une distance maximale de 1200 mètres.

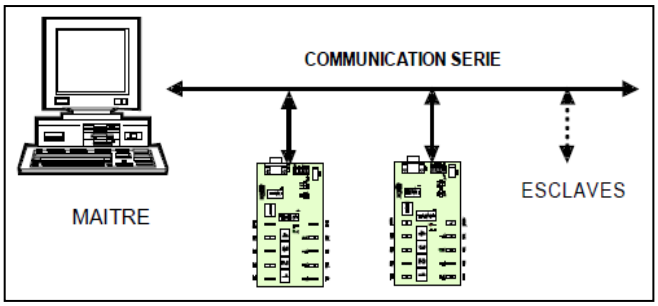

Figure 2.3 *La liaison série type RS485.*

Les caractéristiques générales d'une liaison RS 485 sont:

- Topologies possibles : point à point ou multipoints de type bus.
- Topologie déconseillée : en étoile.
- Distance minimale entre 2 points : 27 cm.
- Longueur maximale d'une dérivation : 1m.
- Vitesse de transmission : 4800/9600/19200 Bauds.
- Débit nécessaire à l'utilisation : optimisation des liaisons en fonction du besoin, maximum 10 Mbits/s.

2.5 Principe de fonctionnement

Les automates utilisent la technique "maître - esclave", dans laquelle un équipement (le maître) a l'initiative des transactions en émettant une demande (requête). L'équipement à qui s'adresse la requête (esclave) émet vers le maître la réponse à sa requête. Le maître peut aussi diffuser un message à l'ensemble des esclaves (requête en diffusion), dans ce cas les esclaves ne répondent pas au maître [17]. Ce type de communication est disponible sur les différents modules de communication au format PCMCIA intégrant une liaison Modbus.

Mécanisme question - réponse : Le maître questionne un esclave de numéro unique sur le réseau, et attend de la part de cet esclave une réponse.

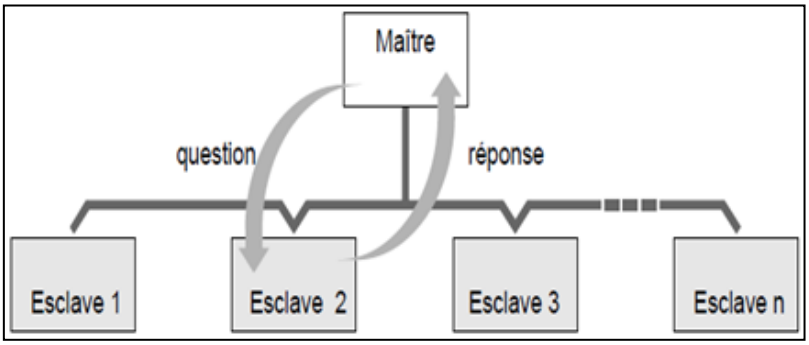

Figure 2.4 *Mécanisme question - réponse.*

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

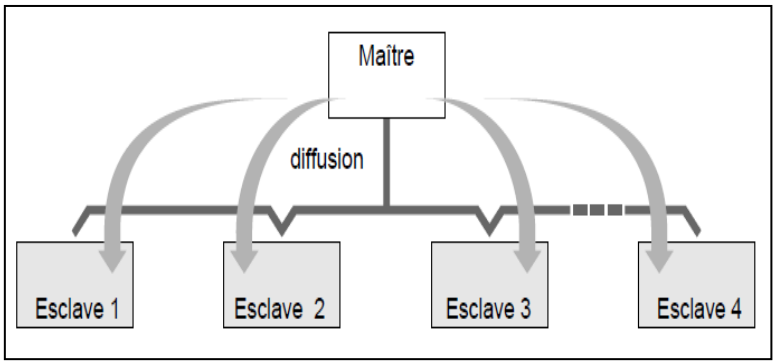

Figure 2.5 *Mécanisme de la diffusion.*

2.5.1 La question

La question 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'intégrité du contenu de la question [19].

2.5.2 La réponse

Lorsqu'un esclave émet une réponse suite à une transaction normale, le code fonction de la réponse est un écho de celui contenu dans la question. Les données sont celles collectées par l'esclave comme par exemple la valeur d'un registre.

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 d'exception permettant de connaître le type d'erreur. Le champ de contrôle permet au maître de confirmer que le message est valide.

Après réception d'une question, une station esclave contrôle la cohérence de la trame. Si un paramètre illégal est détecté (code fonction, adresse, valeur…) ou si la station n'est pas apte à exécuter la demande, elle renvoie une réponse d'exception de la forme ci- dessous [20].

Exemple de codes d'exception d'une esclave

01 : code fonction inconnu.
02 : adresse incorrecte.
03 : valeur incorrecte.
04 : station non prête à exécuter la demande.
05 : acquittement (la station a accepté et est en train de traiter la demande).
06 : la station réalise un traitement et est indisponible.
07 : acquittement négatif.
08 : défaut d'écriture.
09 : chevauchement de zone.

2.5.3 Format d'une trame question - réponse

Question :

Numéro d'esclave	Code fonction	Information spécifiques concernant la demande (adresse, nombre, valeur…)	Mot de contrôle
1 octet	**1 octet**	**n octets**	**2 octets**

Réponse positive :

Numéro d'esclave	Code fonction	Données reçues	Mot de contrôle
1 octet	**1 octet**	**n octets**	**2 octets**

Réponse d'exception :

Numéro d'esclave	Code fonction	Code d'exception	Mot de contrôle
1 octet	**1 octet**	**n octets**	**2 octets**

Cet octet prend la valeur : code fonction + bit poids fort à 1.

2.6 Format général d'une trame

Deux types de codages peuvent être utilisés pour communiquer sur un réseau Modbus. Tous les équipements présents sur le réseau doivent être configurés selon le même type [21].

2.6.1 Type ASCII

En mode ASCII « American Standard Code for Information Interchange », tous les messages commencent par le caractère 'deux points' ":", et se terminent avec deux caractères "CRLF". Les caractères transmis dans les autres champs sont du type hexadécimal 0-9, A-F. Les équipements sur le réseau surveillent continuellement l'arrivée du caractère ":", quand il est arrivé, chaque équipement décode le champ suivant (champ adresse) de façon à connaître l'adresse du destinataire, et ensuite prendre en compte les caractères suivants si l'esclave s'est reconnu. La fin du message sera indiquée par les caractères "CRLF" précédés par les deux caractères de contrôle contenant le LRC.

START	ADRESSE	FONCTION	DONNEES	LRC	END
1 Caractère ":"	2 Caractères	2 Caractères	n Caractères	2 Caractères	2 Caractères "CRLF"

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

- Sans contrôle de parité :

Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Stop	Stop

- Avec contrôle de parité :

Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit7	Parité	Stop

2.6.2 Type RTU « Remote Terminal Unit »

C'est le mode le plus utilisé, il est plus performant que le mode ASCII. En mode RTU, les messages commencent par un intervalle de silence sur le réseau d'au-moins 3,5 caractères. Tous les équipements présents sur le réseau écoutent le bus en permanence, et décodent le premier octet de façon à connaître l'adresse du destinataire, et ainsi prendre en compte les caractères suivants si l'esclave s'est reconnu. Le dernier caractère transmis, un silence d'au moins 3,5 caractères indique la fin du message. Une nouvelle trame peut alors être émise. Les caractères sont du type hexadécimal 0-9, A-F. Les datas contenus dans la trame doivent contenir la totalité du message, et être transmis en continu. L'intégrité du message est indiquée par le contenu du CRC.

START	ADRESSE	FONCTION	DONNEES	CRC	END
Silence	1 octet	1 octet	n octet	2 octets	Silence

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

- Sans contrôle de parité :

Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8	Stop	Stop

- Avec contrôle de parité :

Start	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8	Parité	Stop

Un réseau Modbus RTU se compose d'un "maître" comme un API, et jusqu'à 247 "esclaves" reliés par une liaison multipoint [22].

2.7 Services supportés par Modbus

Modbus offre 19 fonctions différentes. Elles se caractérisent par un code fonction sur un octet (en hexadécimale). Tous les équipements ne supportent pas tous les codes fonction [16].

Les services sont classés en trois catégories :

- L'écriture ou la lecture de mots ou de bits.
- Les fonctions pour le diagnostic des équipements.
- Les fonctions pour gérer les modes de marche d'un équipement.

Tableau 2.1 *Code fonction de Modbus.*

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

NOTE : Les fonctions principales sont écrites en caractères épais.

2.7.1 Les fonctions principales

- **Lecture de n bits de sortie**

Code 01: cette fonction permet d'accéder à des bits de sortie, pouvant être lus ou écrits, définis dans la mémoire d'un esclave.

- **Lecture de n bits d'entrée**

Code 02: cette fonction, identique à la précédente et possédant les mêmes limites, s'adresse aux bits d'entrée seulement en lecture par le maître.

- **Lecture de n mots de sortie**

Code 03: cette fonction permet d'accéder à des mots de sortie, pouvant être lus ou écrits, définis dans la mémoire d'un esclave.

- **Lecture de n bits d'entrée**

Code 04: cette fonction, identique à la précédente et possédant les mêmes limites, s'adresse aux mots d'entrée seulement en lecture par le maître.

- **Ecriture d'un bit de sortie**

Code 05: cette fonction permet le positionnement à 0 ou à 1 d'un bit de sortie défini dans la mémoire d'un esclave.

- **Ecriture d'un mot de sortie**

Code 06: cette fonction, identique à la précédente et possédant les mêmes limites, s'adresse aux bits d'entrée seulement en lecture par le maître.

- **Ecriture de n bits de sortie**

Code 0F: cette fonction permet au maître d'écrire des bits de sortie, pouvant être lus ou écrits, dans la mémoire d'un esclave.

- **Ecriture de n mots de sortie**

Code 10: cette fonction permet au maître d'écrire des mots de sortie, pouvant être lus ou écrits, dans la mémoire d'un esclave.

2.7.2 Les fonctions secondaires

- **Lecture du statut d'exception**

Code 07: cette fonction donne accès à 8 bits de statuts enregistrant certains événements chez un esclave.

- **Diagnostic**

Code 08: cette fonction de diagnostic permet de tester le système de communication entre un maître et un esclave en testant au niveau de l'esclave un certain nombre d'informations

internes. Pour cela un code sous-fonction est inséré dans la trame à la suite du code fonction sur 1 octet [16].

- **Lecture du compteur d'événements**

Code 0B: cette fonction permet de lire 2 mots de 16 bits.

- **Lecture événements de connexion**

Code 0C: cette fonction permet d'accéder aux informations d'un esclave:

- Mot statuts et compteur d'événements.
- Nombre de messages vus sur la ligne et traités par l'esclave.
- Contenu du compteur d'événements de connexion (64 octets maximum).

- **Lecture identification**

Code 11: cette fonction permet de lire un mot de 16 bits contenant les informations de statuts de l'esclave adressé.

2.8 Fonctions gérées par la carte PCMCIA

La carte PCMCIA assure la conversion du protocole Unitelway en protocole Modbus, il est donc possible d'utiliser les fonctions PL7 pour communiquer avec des équipements esclave Modbus. Les fonctions utilisées sont : READ_VAR, WRITE_VAR, SEND_REQ.

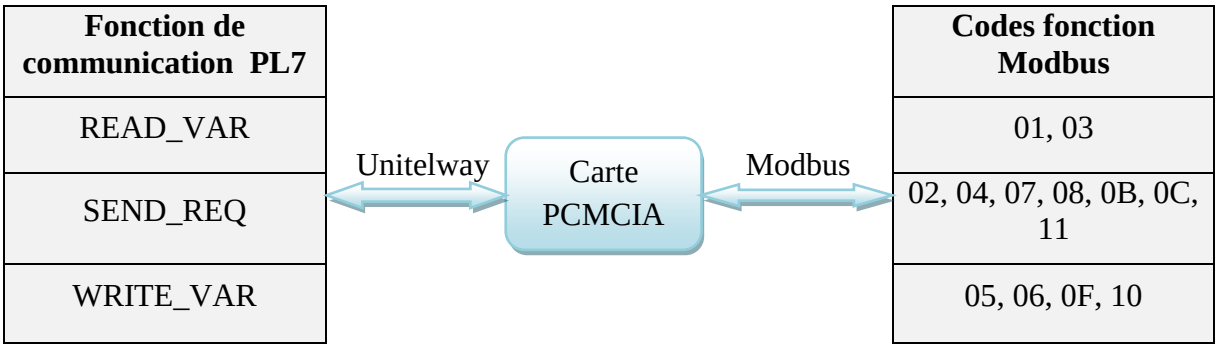

Figure 2.6 *Fonction gérées par la carte PCMCIA.*

2.9 Conclusion

Modbus est un protocole de transmission des messages placé au niveau sept de modèle OSI (couche application), cela fournit une communication "maître - esclave" entre les dispositifs reliés sur différents types des réseaux.

Depuis 1979, Modbus continue à permettre à des millions de dispositifs d'automatisation de communiquer. Aujourd'hui, le soutien de la structure simple et élégante de Modbus continue à se développer.

Modbus est employé pour surveiller et programmer des dispositifs, communiquer les dispositifs intelligents avec des sondes et des instruments et pour surveiller des dispositifs de champ en utilisant des interfaces homme machine. Modbus est également un protocole idéal pour des applications de type RTU où la communication sans fil est exigée.