

1. Introduction

Un **réseau local industriel de terrain** est utilisé dans une usine ou tout système de production pour connecter diverses machines afin d'assurer la commande, la surveillance, la supervision, la conduite, la maintenance, le suivi de produit, la gestion, en un mot, l'exploitation de l'installation de production.

2. Réseau de terrain Modbus

21. Définition

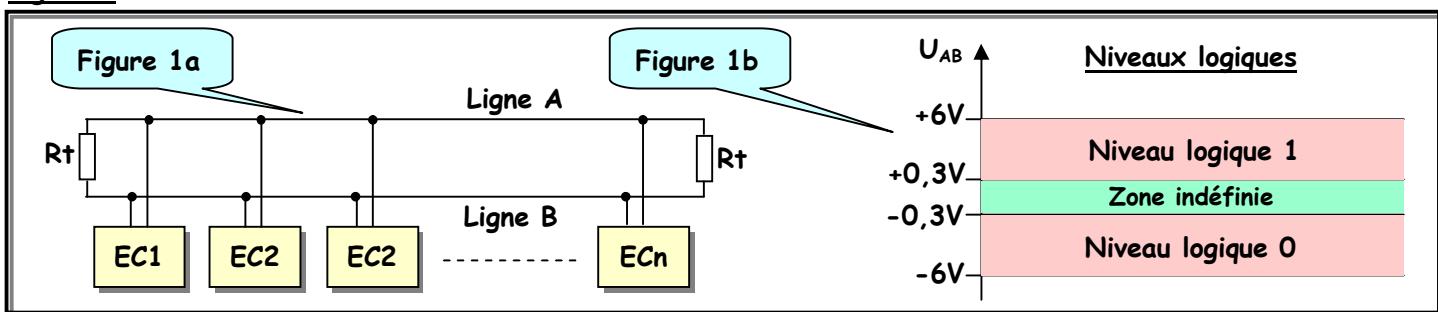
Un **réseau de terrain Modbus** permet d'interconnecter, sur la même ligne ou bus, des équipements d'automatismes industriels en particulier les automates programmables industriels (API), les variateurs de vitesse des moteurs électriques et les commandes numériques.

22. Topologie du réseau

Le réseau de terrain Modbus possède une configuration **multipoints**, principalement à base de la liaison série asynchrone de la norme **RS485**. Sa topologie est donc en **Bus** et utilise la connectique **RS485** comme support physique de transmission.

23. Caractéristiques générales de la norme RS485

Figure 1



Pour relier les différents éléments communicants du Bus (EC1, EC2, ..., ECn), la norme RS485 utilise une ligne différentielle ligne A et ligne B en produisant deux signaux complémentaires sur les deux lignes. Cette technique permet d'immuniser la ligne de transmission contre les parasites ou bruits industriels et d'augmenter la longueur maximale de la ligne estimée à 1200m.

Le câble de la ligne doit être bouchonné en ses deux extrémités par une résistance de terminaison $Rt(120\Omega)$. Elle boucle la ligne sur son impédance caractéristique et minimise le bruit pour une meilleure transmission (figure 1a).

La figure 1b décrit l'équivalence entre les niveaux logiques et les limites de la tension différentielle U_{AB} .

Remarque :

Les circuits logiques à l'origine de la transmission sont compatibles **TTL/CMOS**, il faut alors des circuits drivers ou interfaces d'adaptation à la norme RS485, pour convertir une tension compatible **TTL/CMOS** en une tension différentielle U_{AB} et inversement. On cite à titre d'exemple le circuit **SN75176**.

Le nombre maximal des éléments communicants sur le bus est limité à 32.

La vitesse de transmission des informations en **Bauds** : 4800/9600/19200.

Un **baud** correspond à un nombre d'états électriques par seconde.

Le débit maximal de la ligne atteint **10 Mbits/seconde**.

Le **débit**, exprimé en bits par seconde, est le nombre d'informations transmises chaque seconde sur le réseau.

Le mode de communication sur le bus est le **half duplex** : Il ne peut y avoir qu'un seul élément qui émet sur le bus, le reste des éléments reçoit (il est à l'écoute).

24. Protocole Modbus

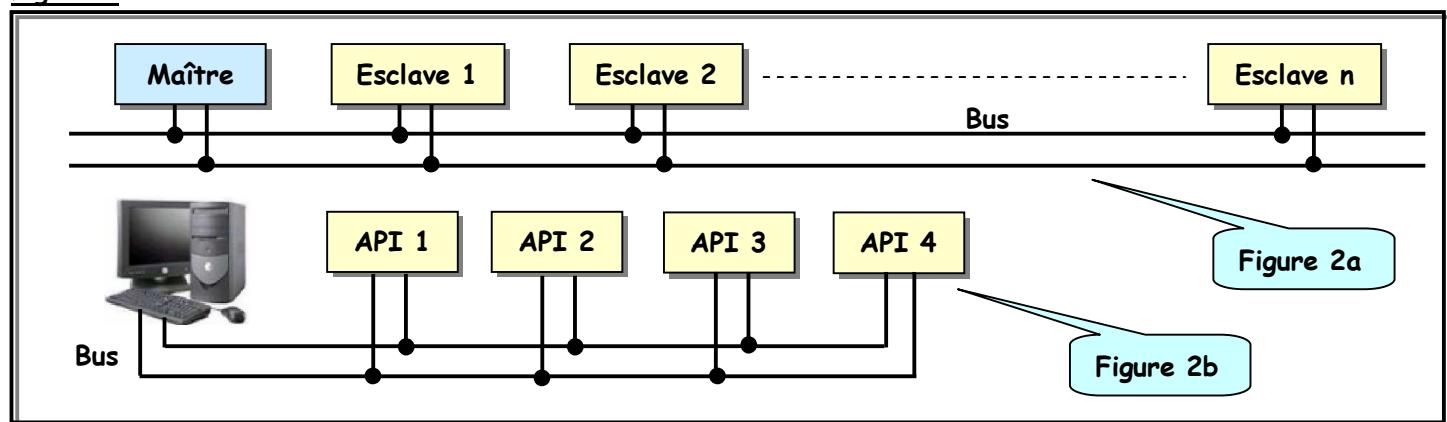
241. Notion de protocole Modbus

Un protocole est un ensemble de règles de communication et une description des mécanismes permettant la gestion des paquets d'informations et leur transition du réseau à l'application.

Le protocole **Modbus**, créé par la société **Modicon**, est un protocole de dialogue basé sur une structure hiérarchisée entre une station **maître**(master) et plusieurs stations **esclaves**(slaves) comme l'indique la figure 2a. Dans cette configuration il est possible de connecter **31 esclaves** dont chacun est repéré sur le bus par une adresse(allant de 1 à 255) unique et indépendante de son emplacement physique.

La figure 2b donne la structure d'un réseau Modbus constitué d'un maître(PC de supervision) et 4 esclaves(**API**).

Figure 2



242. Principe du protocole Modbus

Le maître envoie une **demande** ou **question** à un esclave et attend sa **réponse**(figure 3a).

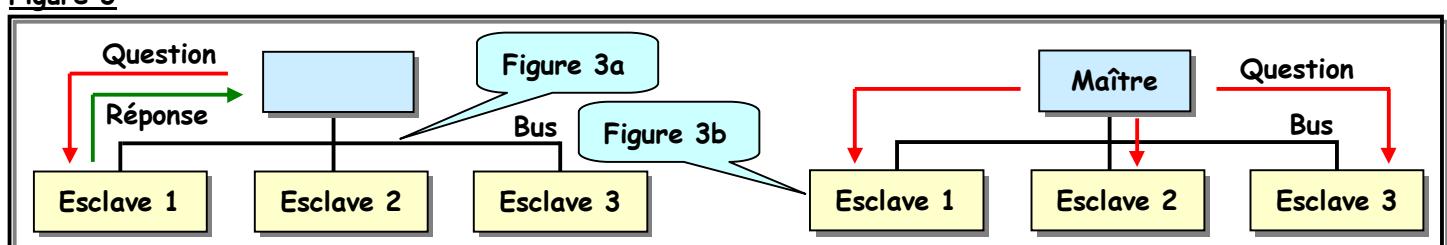
Le maître **diffuse** un **message** ou **question** à tous les esclaves présents sur le réseau, ceux-ci exécutent l'ordre du message sans émettre une réponse(figure 3b).

Deux esclaves ne peuvent dialoguer ensemble.

Chaque demande du maître ou réponse d'esclave est un ensemble d'octets applé **trame** ou **message**.

Le protocole Modbus définit donc la structure des messages et leur mode d'échange du maître vers l'esclave.

Figure 3



243. Format de la trame

L'échange maître/esclave s'effectue par l'envoi de trames Modbus dont le format de base est celui de la figure 4.

L'adresse correspond à l'adresse de la station esclave

destinataire de la requête(adresse 0 pour le mode diffusion).

La requête ou code fonction détermine le type de commande.

Le champ de données contient l'ensemble des paramètres et informations liés à la requête.

Le contrôle de redondance cyclique **CRC16** permet à la station destinatrice de vérifier l'intégrité de chaque trame.

A chaque réception d'une trame, la station adressée envoie une trame de réponse, dont le format est identique à celui de la trame émise par la station maître avec, selon le type de commande, un champ de données plus ou moins important.

Remarque : Les trames de demande et de réponse ont une taille maximale de **256 octets**.

Figure 4



244. Fonctions ou requêtes Modbus

Le protocole Modbus comporte des fonctions principales pour l'échange de données(objets) et des fonctions secondaires pour le contrôle des stations secondaires et des échanges. La liste des principales fonctions Modbus est présentée dans le tableau de la figure 5.

Figure 5

Code Fonction Hexa	Description
01	Lecture de N bits de sortie
02	Lecture de N bits d'entrée
03	Lecture de N mots de sortie
04	Lecture de N mots d'entrée
05	Ecriture d'un bit de sortie
06	Ecriture d'un mot de sortie
07	Lecture de status d'exception
08	Diagnostic
0B	Lecture du compteur d'évènements
0C	Lecture évènements connexion
0F	Ecriture de N bits de sortie
10	Ecriture de N mots de sortie

Les échanges de données entre stations maître/esclave sont représentés par quatre types d'objets :

- Bits d'entrée(lecture).
- Bits de sortie(lecture/écriture).
- Mots d'entrée(lecture).
- Mots de sortie(lecture/écriture).

Chaque station esclave définit ainsi des zones de bits et de mots(tables d'octets), auxquelles sont associées des adresses Modbus.

Dans chaque trame de lecture(bits ou mots) la zone de données contient l'adresse du premier bit ou mot à lire et le nombre de bits ou mots à lire.

Pour chaque trame d'écriture, la zone de données contient l'adresse du premier bit ou mot à écrire, le nombre de bits ou mots à écrire, ainsi que les valeurs des bits ou mots à écrire.

Dans le cas de fonctions secondaires, le contenu des trames de réponse permet d'obtenir différentes informations de chaque station esclave : diagnostic(état de la station), nombre de messages traités, nombre de messages reçus erronés (CRC16), nombre de messages reçus incorrects, nombre de messages reçus (total), etc ...

245. Principe d'échange Maître/Esclave ou Question/Réponse

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. Le champ de données est codé sur n octets. Selon le code fonction, le champ de données contient diverses informations complémentaires permettant à l'esclave de décoder le message.

Le maître calcule, selon un algorithme précis, la valeur du champ de contrôle d'erreur CRC16.

Lorsque l'esclave reçoit le message de demande, il le range en mémoire, calcule la valeur du CRC16 émis et la compare à celle du CRC16 reçu :

- Si le message reçu est incorrect (inégalité des CRC16), 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 ou réponse d'erreur.

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 fonction différent pour indiquer que la réponse est une réponse d'erreur.

Le champ de données contient diverses informations dépendant du code fonction pour une réponse normale ou le code d'exception, permettant de connaître le type d'erreur, pour une réponse d'exception.

Le champ de contrôle CRC16, contient une valeur calculée à partir du message, permet au maître de confirmer que le message est valide.

Echange Question/Réponse relatif aux fonctions 3 et 4

Cet échange correspond à la lecture de N mots de sortie ou lecture de N mots d'entrée(voir figure 6).

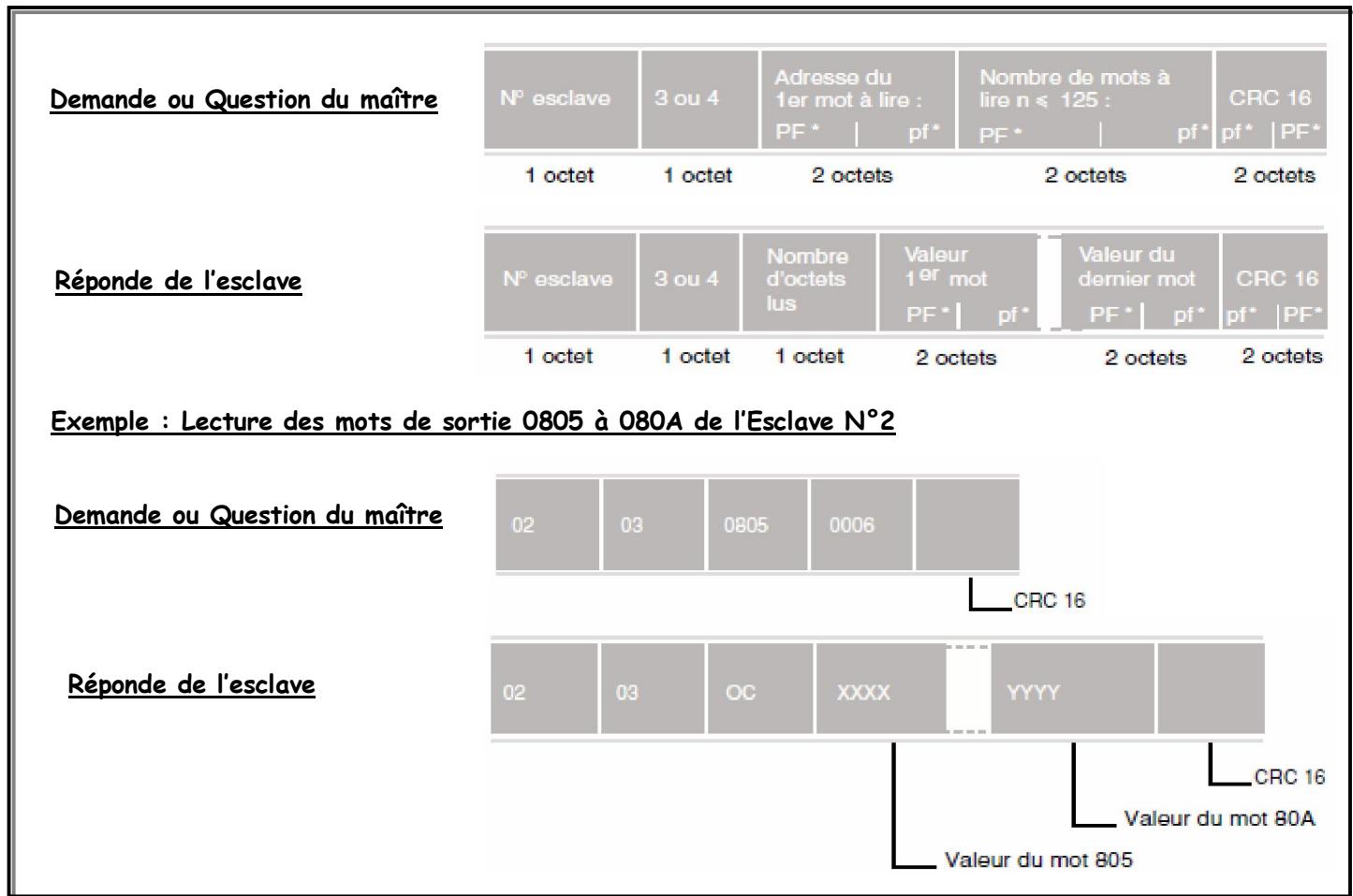
Le nombre de mots à lire doit être inférieur à 125.

Remarque :

Le mot représente ici 2 octets.

PF et pf désignent respectivement le poids fort et le poids faible.

Figure 6



Echange Question/Réponse relatif à la fonction 10

Cet échange correspond à l'écriture de N mots de sortie(voir figure 7).

Remarque : Si le numéro d'esclave est égal à 0, tous les esclaves exécutent l'écriture sans émettre de réponse.

246. Modes de transmission

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).

Type RTU(Unité terminale distante)

Chaque octet composant une trame est codé sur 2 caractères hexadécimaux (2 fois 4 bits).

Remarque : Ces deux modes de transmission peuvent être utilisés dans le protocole Modbus, mais sont incompatibles entre eux. Environ 95% des modules communicants sur Modbus utilisent le mode RTU.

Le format de la trame en mode RTU est donné à la figure 8a dans lequel le silence est équivalent à un délai d'au moins égal à 3,5 fois le temps de transmission d'un mot ou caractère.

La figure 8b illustre une vue d'ensemble de 3 trames séparées par un délai intertrame.

Remarque : En mode RTU, la trame de message entière doit être transmise en tant que flux de caractères continu, car des temps silencieux supérieurs à 1,5 fois délai de caractère entre 2 caractères seront interprétés par l'équipement de réception comme une trame incomplète. Le récepteur éliminera cette trame.

247. Support de transmission

Chaque octet composant un message est transmis en mode RTU, avec ou sans parité, au format de la figure 8c.

Figure 7

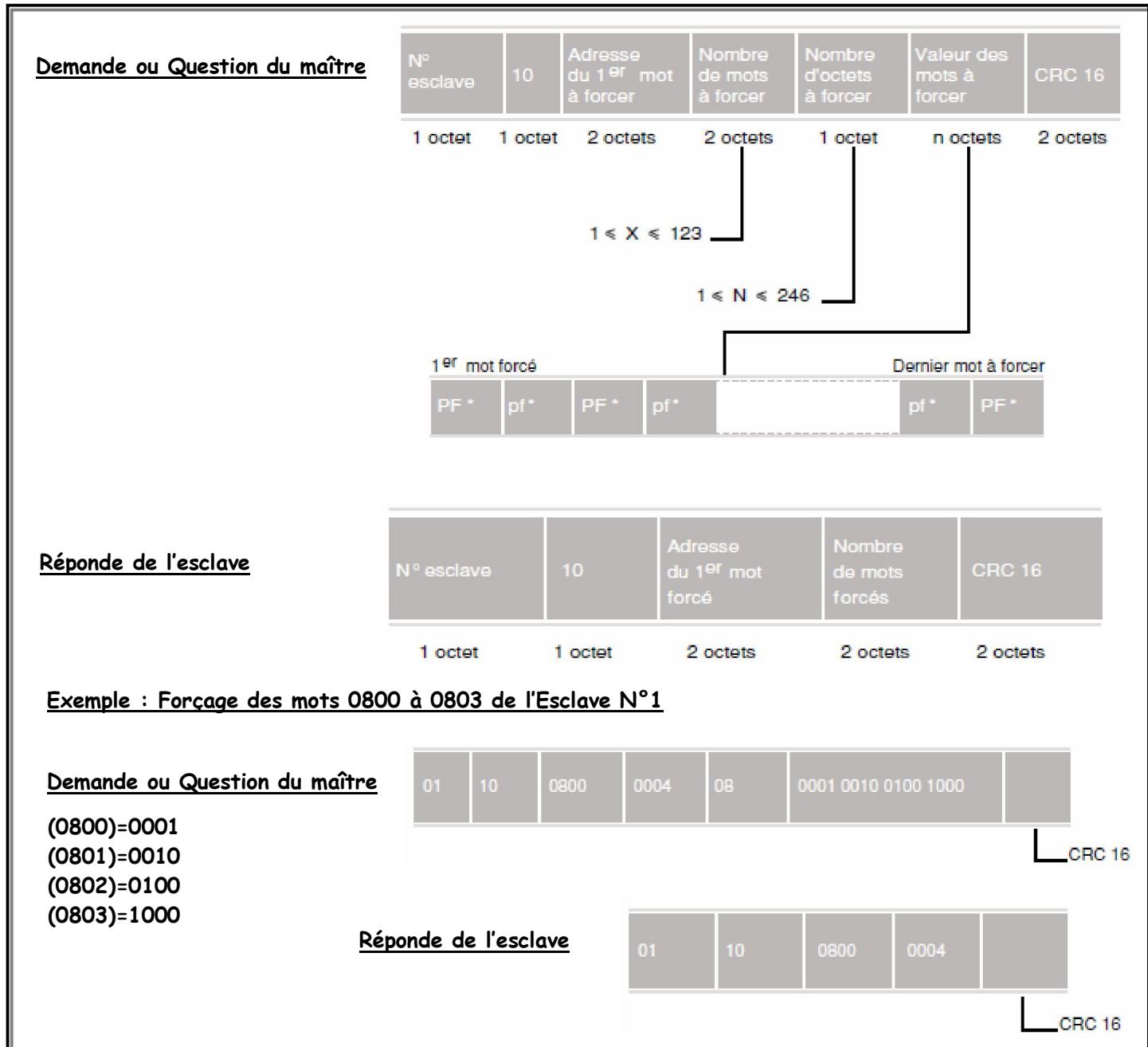
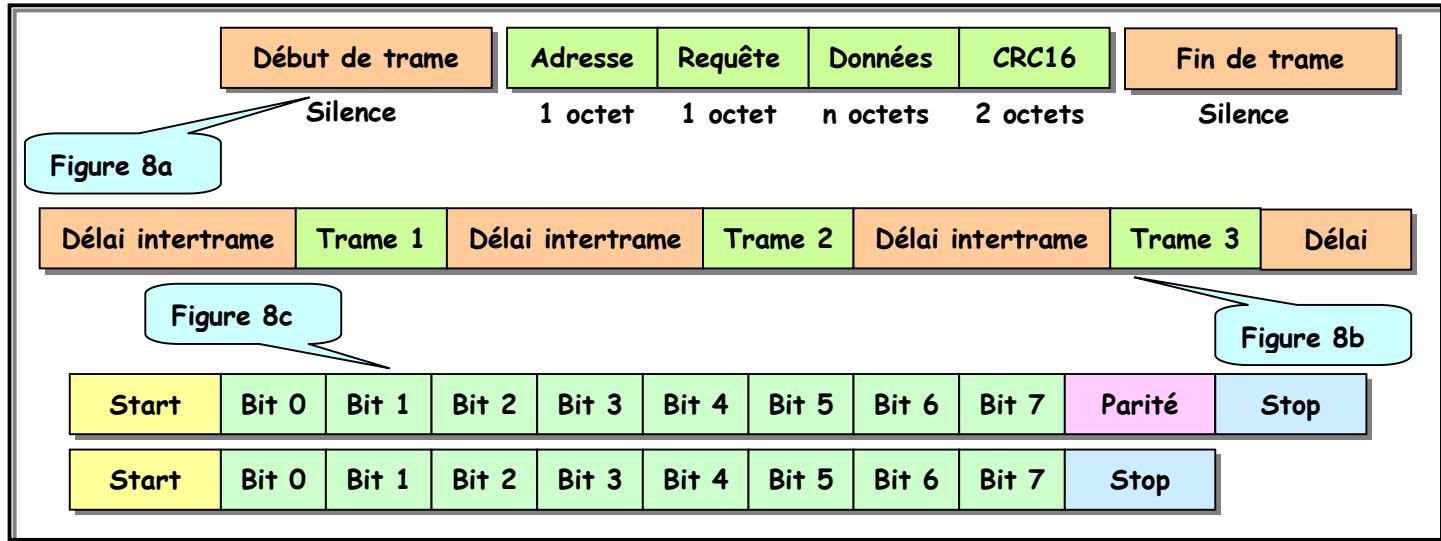


Figure 8



3. Réseau de capteurs/actionneurs ASI

31. Définition

Le terme **ASI** signifie **Actuator Sensor Interface** (interface pour capteurs et actionneurs). C'est un réseau de terrain utilisable pour interconnecter, sur le même câble, des capteurs, des interfaces de dialogue Homme/Machine et des actionneurs. C'est un standard industriel qui permet de mettre en liaison les capteurs et actionneurs, en général, avec un automate programmable industriel(API).

Les capteurs sont principalement TOR mais récemment il est possible d'utiliser des capteurs de type analogique.

32. Topologie du réseau ASI

Le principe du réseau de terrain **ASI** s'appuie sur l'échange **maître/esclave** qui comprend un maître (PC, API, Contrôleur ...) chargé d'assurer la gestion des états des capteurs/actionneurs(esclaves) et de les transmettre à l'automatisme. Sa topologie est libre(étoile, ligne ou bus, arbre et anneau) mais la plus répandue est en **Bus**.

33. Caractéristique du réseau ASI

- Le nombre d'esclaves maximum sur le bus est limité à **31** esclaves pour un adressage standard et **62** esclaves pour un adressage étendu.
- La longueur maximale de bus ne peut excéder **100** mètres sans répéteur et **300** mètres avec répéteur.
- le nombre maximum des entrées/sorties géré par le bus varie suivant la version du réseau ASI : pour la version 1, avec un adressage standard des esclaves, on a **124** entrées et **124** sorties et pour la version 2, avec un adressage étendu des esclaves, on a **248** entrées et **186** sorties.
- Le temps maximal du cycle est de 5ms pour l'adressage standard et de 10ms pour l'adressage étendu.

34. Éléments constitutifs d'un réseau ASI

341. Câble ASI

Le câblage du bus de transmission s'effectue avec un câble spécifique ASI. C'est une liaison bifilaire qui transporte à la fois les données de communication et l'alimentation de 30V pour des équipements connectés.

La section des fils peut être de 0,75 mm², 1,5 mm² ou 2,5 mm², suivant le courant consommé par les équipements. On distingue un câble plat ASI à détrompage de couleur jaune et un câble rond standard.

Remarque : Il existe aussi un câble noir qui sert à acheminer une alimentation auxiliaire 24V dans le cas d'utilisation des composants grands consommateurs de courant.

342. Alimentation ASI

Le bloc d'alimentation ASI sert principalement à fournir une alimentation de **30V** aux périphériques du système via le câble ASI.

Remarque : Lors de l'extension d'un réseau ASI, on utilise un bloc d'alimentation ASI supplémentaire pour chaque segment de câble ajouté. Mais dans le cas des composants grands consommateurs de puissance on utilise une alimentation auxiliaire standard qui fournie **24V**.

343. Puce ASIC

Le concept d'ASI repose non seulement sur le type de câble mais également un circuit intégré spécifique appelé **ASIC(Application Specific Integrated Circuit)**.

Ce circuit spécifique est intégré directement soit dans le capteur ou l'actionneur ou soit dans une interface bus capteur/actionneur.

Il gère toutes les fonctions du capteur ou de l'actionneur et informe notamment le maître sur l'état de la commutation (actionné ou non).

Chacun de ces circuits peut échanger via le câble, entre le maître et l'esclave, suivant le type de capteur, d'actionneur (ou d'interface bus d'entrées/sorties), des données d'entrées et/ou des données de sorties sur 4 bits. 4 bits supplémentaires de paramètres permettent à l'ASIC de contrôler des fonctions spécifiques de l'esclave.

Le bus ASI rend ainsi possible la communication avec 2 catégories principales de capteurs/actionneurs :

- Capteurs ou actionneurs **communicants**. Directement connectés au bus, c'est dans le capteur ou l'actionneur qu'est installé le circuit ASIC.

- Capteurs ou actionneurs traditionnels **non communicants**. Ce sont des capteurs ou actionneurs classiques, (exemples : capteur photoélectrique, inductif, capacitif... lampe, relais, électrovanne...) non dédiés habituellement au bus ASI. On les connecte au bus par l'intermédiaire d'interfaces ASI (répartiteurs dits actifs dans lesquels se trouve l'ASIC) pour leur permettre de dialoguer via le bus avec le maître (API par exemple).

344. Coupleur

Un coupleur est le maître chargé d'assurer l'interface entre le bus et l'organe de commande (PC, API, passerelle...). Un coupleur ASI version 1 peut gérer au total 124 entrées et 124 sorties (soit au total 31 capteurs/actionneurs communicants ou jusqu'à 124 capteurs et 124 actionneurs traditionnels).

345. Répartiteur actif

C'est une interface qui permet de raccorder, à l'aide d'un connecteur, des capteurs/actionneurs traditionnels non communicants. C'est un **module esclave** d'interface ASI qui comporte le circuit ASIC et qui fournit la communication directe avec le maître. Ce module utilise une seule adresse et il peut être raccordé avec **8** capteurs/actionneurs au maximum tels que les capteurs simples TOR (capteur inductif, optique, capacitif, fin de course, etc...) et les actionneurs (lampe de signalisation, relais, électrovanne, démarreurs moteurs, etc...).

346. Répartiteur passif

C'est une interface qui permet de raccorder, à l'aide d'un connecteur, des capteurs/actionneurs communicants. C'est un **module esclave** d'interface ASI qui permet de raccorder des capteurs/actionneurs intelligents.

347. Interface bus/entrées-sorties

C'est une interface qui permet de raccorder par bornes à vis tous types de capteurs/actionneurs traditionnels non communicants.

348. Module intelligent

C'est un capteur ou actionneur intelligent, dit aussi **asifié**, qui comporte lui-même un circuit ASIC, il est donc raccordable directement sur le bus ASI et il communique directement avec le maître. Ce type de module intelligent utilise une seule adresse.

349. Interface de dialogue Homme/Machine asifié

Elle constitue un outil de dialogue parfaitement adapté à un échange d'information entre opérateur et machine. On cite en particulier à titre d'exemple : des boîtes à boutons, des claviers, des colonnes lumineuses, etc...

350. Accessoires de raccordement et dérivation

Le raccordement au bus ASI peut se faire grâce à des **tés** de connexion prévus pour des raccordements sur câble plat ASI ou des **dérivations** câble plat/câble rond.

35. Architecture d'un réseau ASI

La figure 9 montre un exemple d'un réseau ASI.

Figure 9

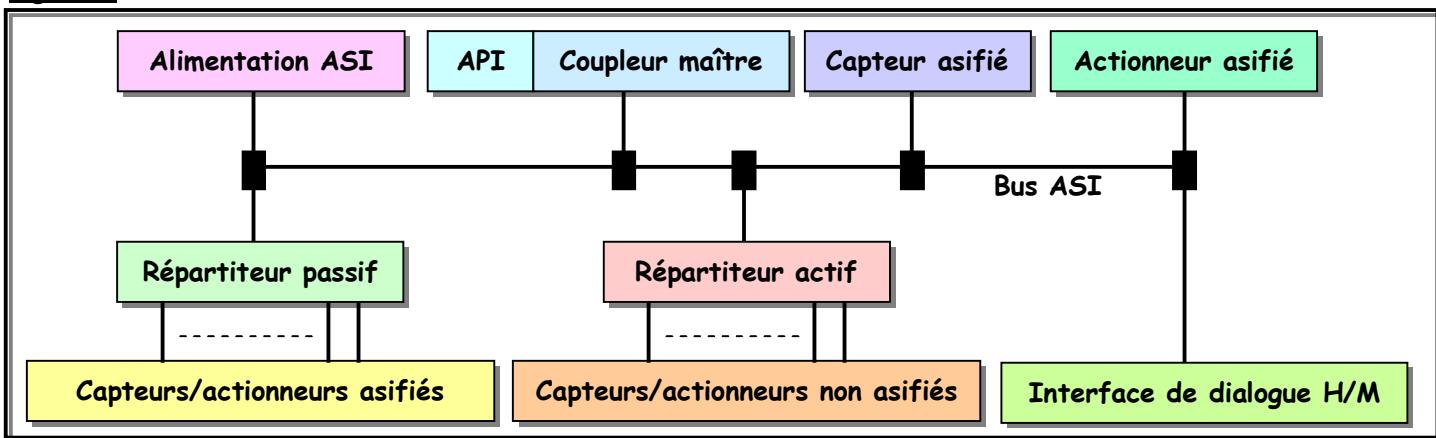
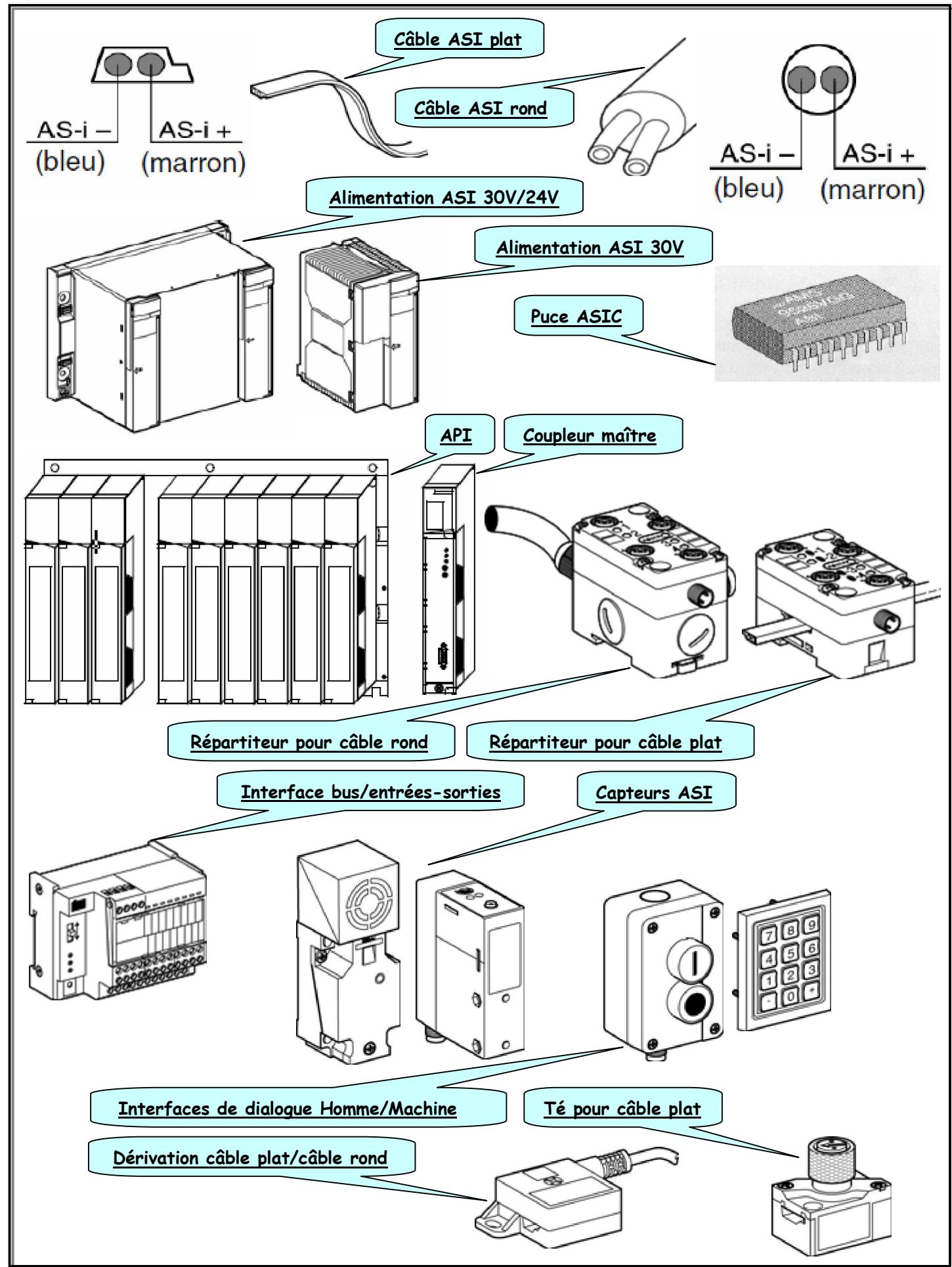


Figure 10



36. Protocole du réseau ASI

361. Principe du protocole

Le principe de communication est basé sur un protocole à un seul maître. Le maître du bus interroge les esclaves à tour de rôle, ceux-ci répondent en envoyant les données requises. Quand tous les esclaves ont été interrogés, le cycle se répète et continue indéfiniment. Le temps de cycle dépend du nombre d'esclaves du bus.

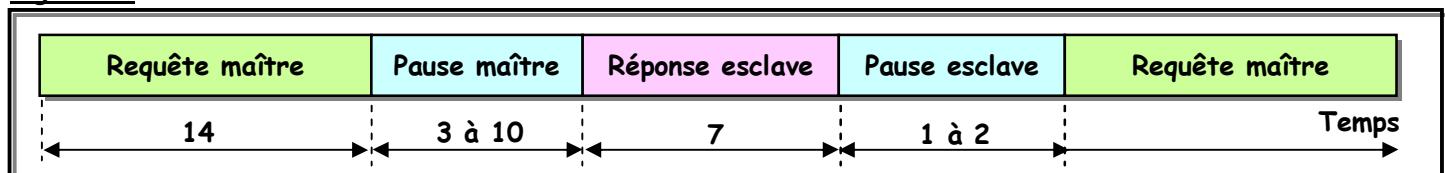
362. Transaction ASI

Une transaction ASI(opération d'échange maître/esclave) comporte les éléments de la figure 11.

Si on suppose que la durée d'un bit est de $6\mu s$, une transaction nécessite un temps global de $6*26\mu s$ soit $156\mu s$.

Le temps nécessaire pour un cycle correspondant à 31 esclaves est estimé à $31*156\mu s$ soit 4,836ms(environ 5ms).

Figure 11



363. Trames ASI

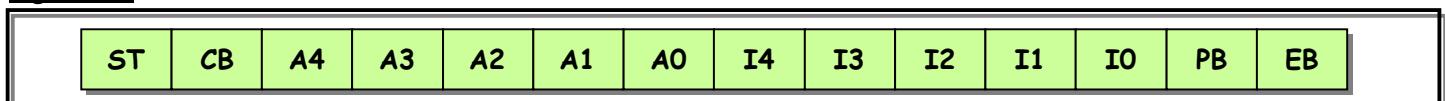
Le maître ASI utilise un télégramme ou trame pour communiquer avec les esclaves raccordés.

On distingue deux types de trames : trame du maître et trame de l'esclave.

Trame du maître

La trame correspondante aux requêtes du maître est constituée des informations de la figure 12.

Figure 12



- **ST** : bit de début $\Rightarrow ST=0$: début.

- **CB** : bit de contrôle

$\Rightarrow CB=0$: échange de paramètre, données et adresses.

$\Rightarrow CB=1$: Commande (reset, suppression d'adresse...).

- **A4...A0** : 5 bits réservés aux adresses des esclaves.

\Rightarrow Les esclaves peuvent avoir des adresses de 1 à 31 pour un adressage standard.

\Rightarrow Les esclaves peuvent avoir des adresses de 1A à 31A et de 1B à 31B pour un adressage étendu.

\Rightarrow L'adresse 0 est réservé à la fonction adressage automatique.

- **I4...IO** : 5 bits réservés aux informations.

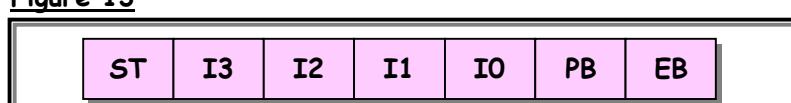
- **PB** : bit de contrôle de parité paire(détection d'erreur).

- **EB** : bit de Fin $\Rightarrow EB=1$: Fin.

Trame de l'esclave

La trame correspondante aux réponses de l'esclave contient les informations de la figure 13.

Figure 13



- **ST** : bit de début $\Rightarrow ST=0$: début.

- **I3...IO** : 4 bits réservés aux informations retournées au maître.

- **PB** : bit de contrôle de parité paire.

- **EB** : bit de Fin $\Rightarrow EB=1$: Fin.

364. Description des requêtes maître/esclave

Lors d'une transaction maître/esclave on peut avoir différentes requêtes selon la demande du maître à l'esclave.

Les requêtes possibles du protocole ASI sont :

- Echange de données.
- Ecriture et lecture de paramètres.
- Attribution d'adresse.
- Suppression d'adresse.
- Réinitialisation d'esclave.
- Lecture de la configuration des entrées/sorties.
- Lecture de l'identificateur de l'esclave.
- Lecture de l'état de l'esclave.
- Lecture et remise à zéro des bits d'état.

La requête la plus fréquemment utilisée est la **requête d'échange de données**. Elle ne peut être utilisée qu'avec les esclaves ayant une adresse non nulle. Cette requête est utilisée par le maître pour échanger des données opérationnelles avec les esclaves :

- Pour demander l'état des entrées d'un périphérique d'entrée tel qu'un capteur (dans ce cas, l'esclave répond en donnant son état).
- Pour modifier l'état des sorties d'un périphérique de sortie tel qu'un actionneur (dans ce cas, la requête contient des données).

Pour un échange de données la requête du maître aura la forme de la figure 14.

Figure 14

0	0	A4	A3	A2	A1	A0	0	D3	D2	D1	D0	PB	1
---	---	----	----	----	----	----	---	----	----	----	----	----	---

- Le bit **I4** est positionné à 0 pour un échange de données.

- **D3...D0** : informations à échanger. Elles correspondent aux sorties de l'esclave.

La réponse de l'esclave suite à une requête d'échange de données aura la forme de la figure 15.

Figure 15

0	D3	D2	D1	D0	PB	1	
---	----	----	----	----	----	---	--

- **D3...D0** : informations de l'esclave. Elles correspondent aux entrées de l'esclave.