

Réseaux Industriels

ISTIA – Ei4

Année 2011/2012

1. Les réseaux industriels

Dans une entreprise, il peut arriver fréquemment que l'automate, les actionneurs et les capteurs ne soient pas situés au même endroit mais à des distances importantes les uns des autres. L'utilisation d'un réseau industriel permet donc de faire communiquer plusieurs automates, chacun reliés à une partie des capteurs/actionneurs.

La gestion d'une ligne de production est souvent présentée de la manière suivante :

- 1) capteurs / actionneurs
- 2) contrôle / commande (automates)
- 3) supervision (PC)

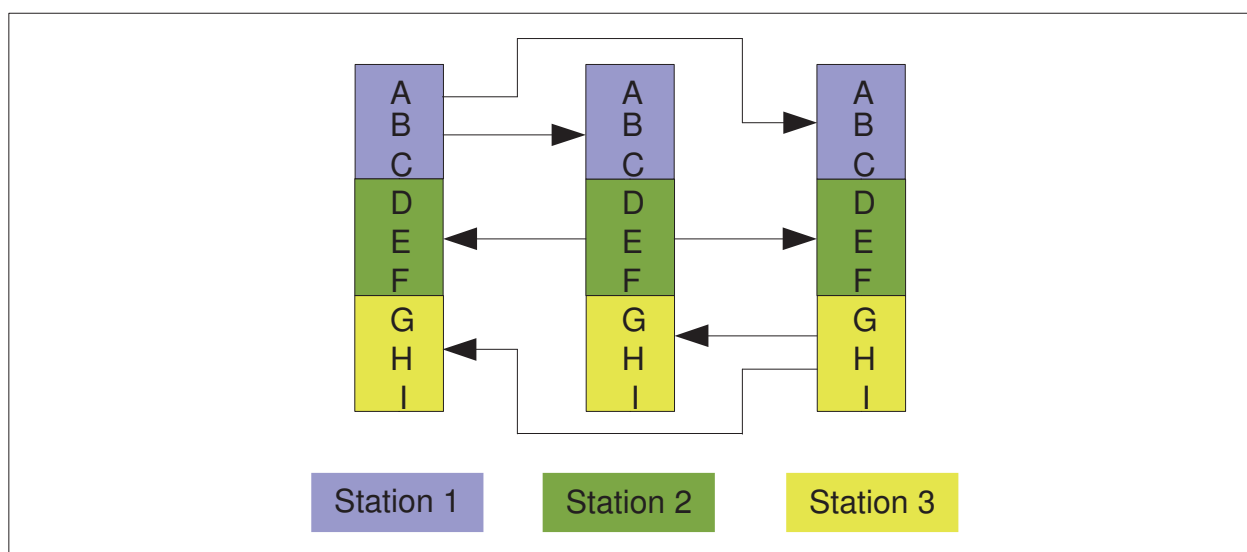
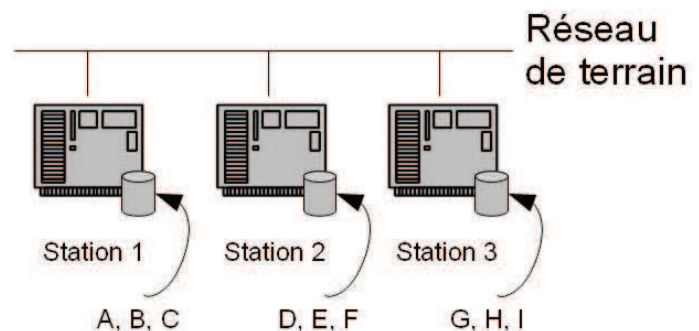
Le réseau industriel, ou encore réseau de terrain, se place au niveau 2, entre les automates.

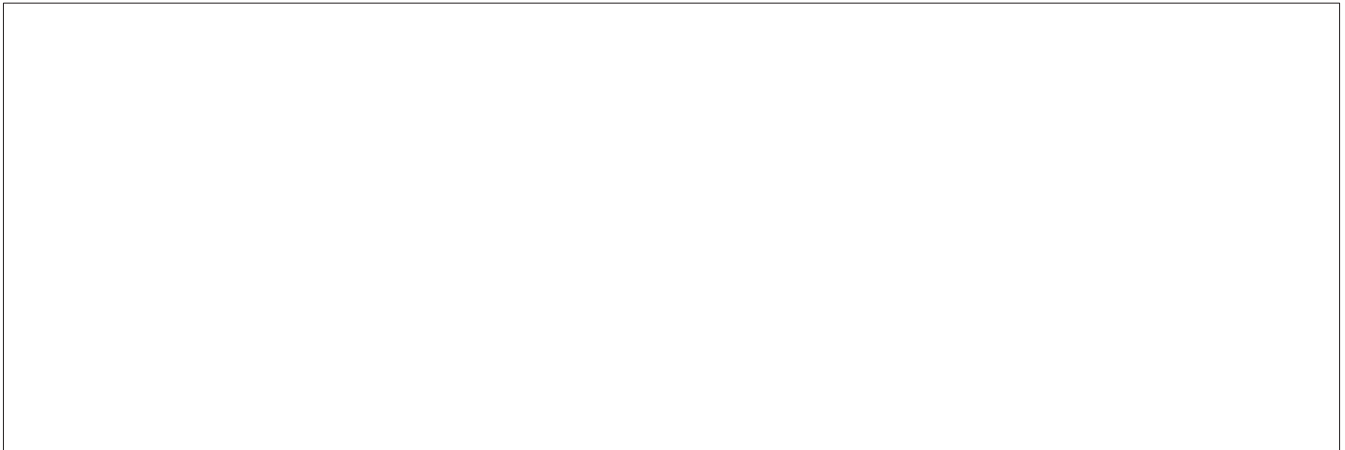
1.1 Echange de données

Dans un réseau industriel, les informations relatives aux entrées/sorties de la maquette sont réparties sur chacun des automates selon le câblage réalisé. Or, pour continuer de fonctionner, il est possible qu'une station ait besoin de connaître la valeur d'un capteur relié à une autre station.

Pour ce faire, une partie des données contenues sur chacun des automates sont périodiquement envoyées sur le réseau par chacun d'entre eux et à l'attention de tous. Ces données sont en réalité des mots mémoires appartenant à chacun des automates (exemple : mots A, B, C de la station 1).

L'échange de ces mots mémoires « partagés » entre les stations est représenté sur le schéma suivant :



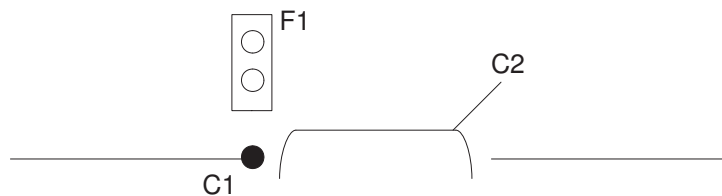
Remarques

1.2 Problématique

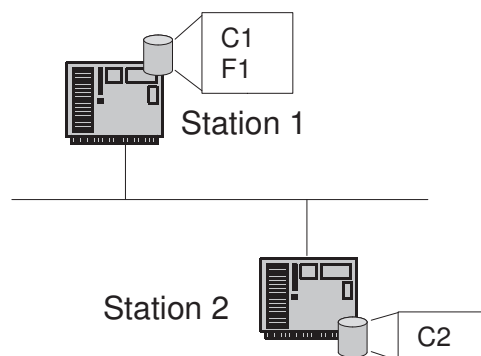
Cet échange se faisant périodiquement et automate après automate, un problème se pose : comment une station peut-elle être sûre d'avoir les dernières et bonnes valeurs des mots mémoires des autres stations ? Entre deux échanges, un capteur peut être activé et le mot mémoire lui correspondant être changé, une trame peut se perdre sur le réseau, un bit peut changer de valeur sous l'influence d'un champ électromagnétique.

Exemple

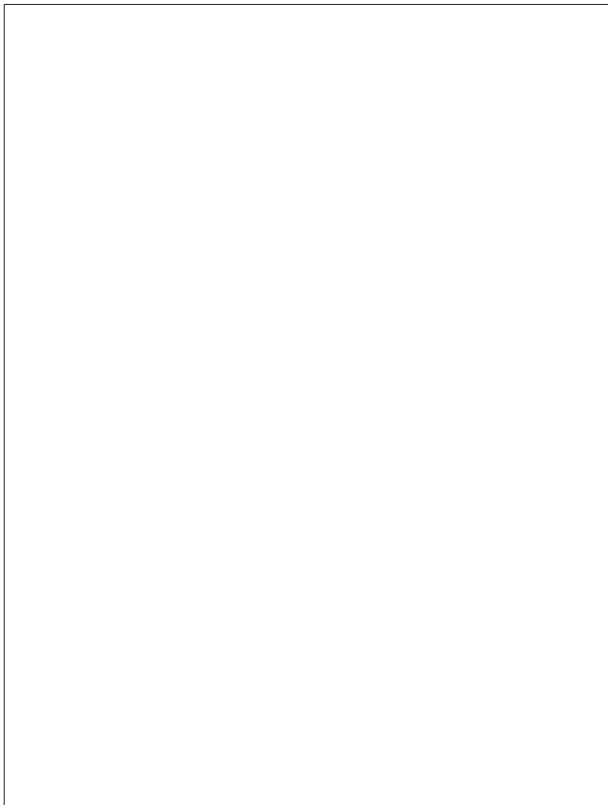
Sur le schéma suivant est représenté une route coupée d'un pont. Ce pont ne peut supporter le poids que d'un seul véhicule. Si un véhicule se présente en C1, il n'aura l'autorisation de passage, donné par le feu F1, que si le pont est libre (capteur C2).



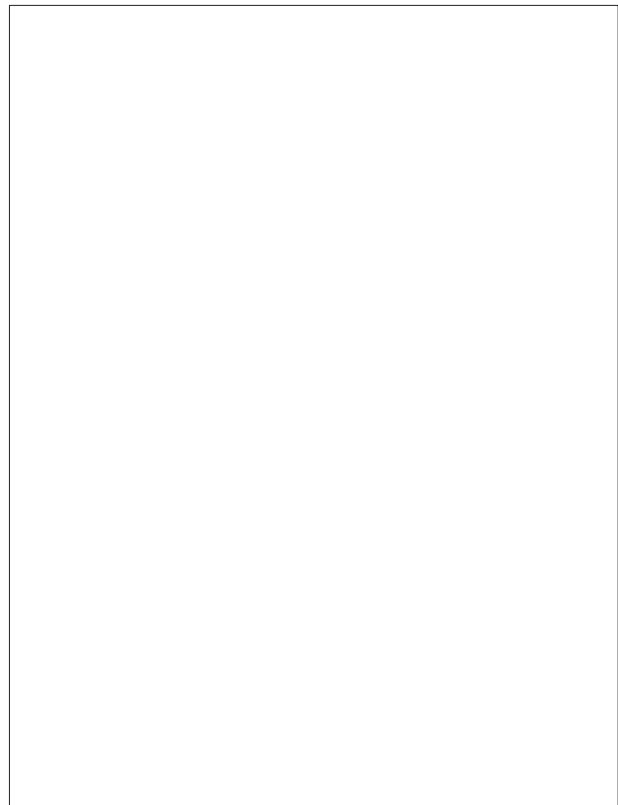
Le capteur C1 et le feu F1 sont reliés à la station 1 alors que, pour des raisons techniques, le capteur C2 est relié à la station 2.



Si la station 2 met dans un de ses mots partagés la valeur du capteur C2 et grâce à l'échange périodique de données sur le réseau, la station 1 aura un accès à cette information. La solution de cet exemple pourrait ainsi être la suivante :

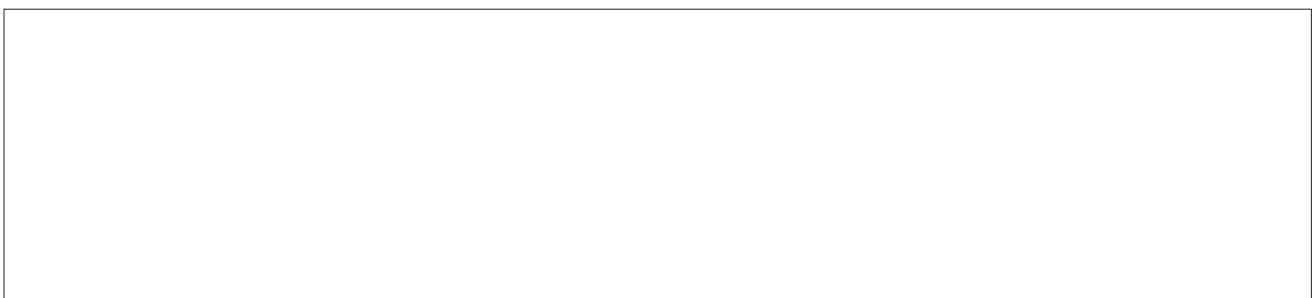


Station 1



Echange réseau

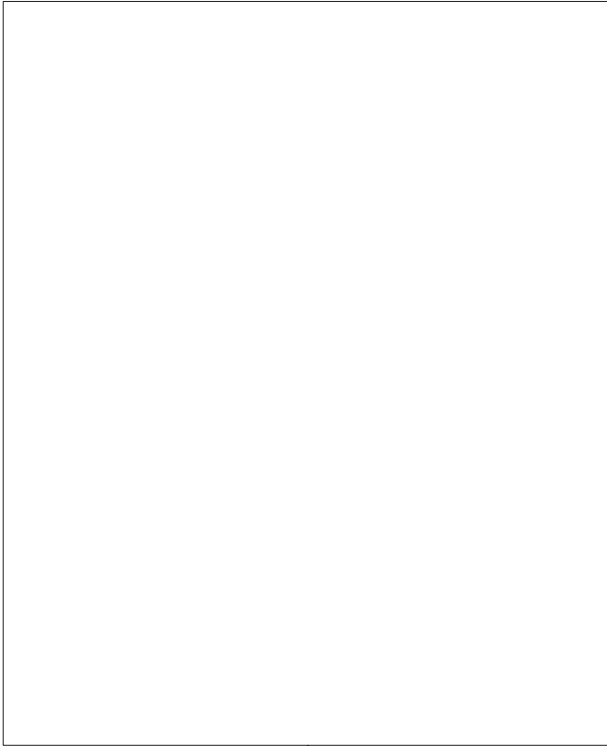
Inconvénients de cette solution

A large empty rectangular box intended for listing the disadvantages of the solution.

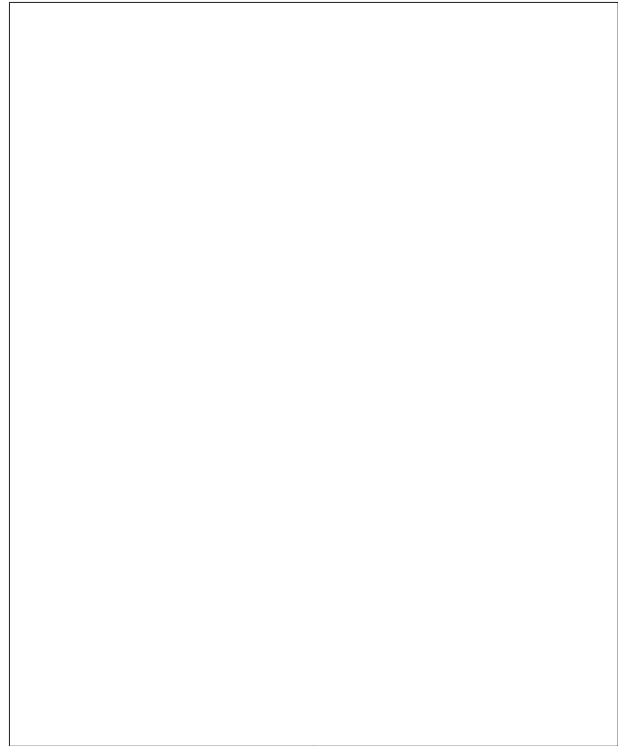
1.3 Sémaphores

Afin de répondre à cette problématique, une autre solution est à réaliser. Dès qu'un véhicule se présente au capteur C1, la station 1 prévient la station 2 du passage d'un véhicule en écrivant dans un de ses mots partagés une valeur qui correspond à « un véhicule est présent en C1 ». A nouveau, ces mots étant périodiquement envoyés sur le réseau, la station 2 en sera forcément informée à un moment.

Les grafçets correspondant à ces conditions sont alors les suivants :



Station 1



Station 2

A quel moment la station 1 doit-elle arrêter de prévenir la station 2 et ainsi autoriser le passage du véhicule ?



Cet exemple montre bien comme, dans un réseau industriel, il est nécessaire d'implémenter un système de synchronisation dans les grafçets des différentes stations de manière à ordonner la lecture dans les variable partagées (les échanges périodiques se font dans ordre qui dépend du type de réseau utilisé). Ce système de synchronisation porte un nom : les sémaphores, et se décompose en 4 temps :

- 1- demande / station 1
- 2- réception de la demande et conditions réunies permettant de répondre positivement à cette demande / station 2
- 3- réception de la réponse engendrant un arrêt de la demande / station 1
- 4- visualisation de l'arrêt de la demande engendrant un arrêt de la réponse / station 2

Un peu d'histoire...

A sa création en 1806, le sémaphore était un poste de guet établi sur la côte chargé de surveiller les approches maritimes et de signaler par signaux optiques toute activité ennemie (le mot sémaphore vient du grec sema : « signe » et phoros : « qui porte »).

En informatique, un sémaphore est une variable protégée permettant un accès à des ressources partagées. D'une manière générale, cela permet une restriction de l'accès.

2. Différents types de réseau

Quelques définitions :

- protocoles : ensemble de règles de communication qui définissent l'émission et la réception de données sur un réseau.
- réseau : ensemble d'équipements reliés entre eux pour échanger des informations.

Il existe différents protocoles de réseaux industriels principalement associés au type d'automate utilisé (Télémécanique utilise le protocole WorldFip avec un réseau FipWay). Lors de ce cours, deux réseaux seront étudiés : FipWay, et Profibus.

2.1 Réseau FipWay et protocole WordFip

2.2.1 Fonctionnement général

Dans un réseau FipWay, la communication est gérée par un arbitre de bus. Deux types de trafic sont proposés : périodique et apériodique. Chaque station possède deux types de fonctionnalités :

- l'arbitrage de bus : gestion de l'accès au medium.
- la fonction production/consommation.

Une seule station a sa fonctionnalité arbitre de bus active.

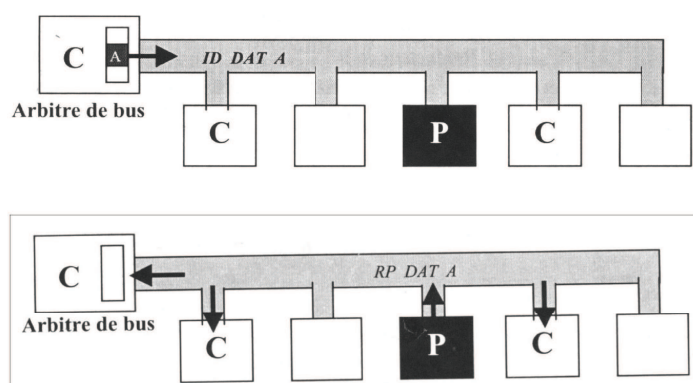
A) Trafic périodique

Le rôle de l'arbitre de bus est de gérer l'échange des informations sur le bus. Il dispose pour cela d'une table contenant la liste des identifiants à faire circuler.

L'arbitre de bus diffuse sur le bus par l'intermédiaire d'une trame question (ID_DAT), le nom d'un identifiant. Une et une seule des stations connectées sur le bus se reconnaît comme productrice de l'identifiant. Une ou plusieurs stations se reconnaissent consommatrices de cette variable.

P : station productrice de la variable.

C : station consommatrice de la variable.

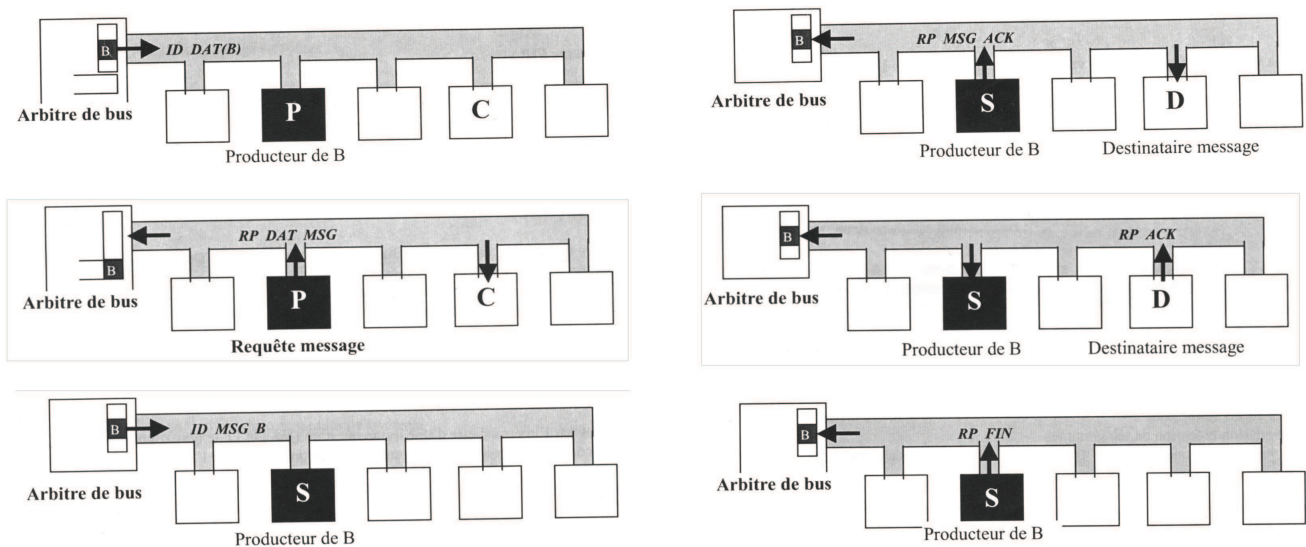


Le producteur de la variable diffuse ensuite la valeur relative à l'identifiant à l'intérieur d'une trame de réponse (RP_DAT). Cette valeur est simultanément captée par toutes les stations consommatrices. L'arbitre de bus passe ensuite à l'identifiant suivant et le même cycle question réponse recommence.

La table d'arbitrage est fixée une fois pour toute (hard du coupleur). Elle permet de définir avec précision la dynamique du réseau.

B) Trafic aperiodique

Lorsqu'un producteur émet l'identifiant qu'on lui a demandé de produire, il peut réclamer un droit de parole supplémentaire (bit MSG, du champ de contrôle). L'arbitre prend note de cette requête et la place en file d'attente. Dès que possible l'arbitre de bus fournira un droit de parole à cette station dans la fenêtre réservée au trafic aperiodique.



La station producteur de B demande la parole avec une requête message (RP_DAT_MSG) et l'arbitre de bus l'autorise à émettre un message (ID_MSG_B). Le producteur de la variable émet son message composé de son adresse, et de l'adresse du destinataire dans une trame du type RP_MSG_ACK. Le destinataire du message émet une trame d'acquittement (RP_ACK). L'émetteur signale ensuite sa fin de transaction à l'arbitre de bus (RP_FIN).

C) Procédure d'élection de l'arbitre de bus

Une station constatant un silence supérieur à un temps T, s'élit arbitre de bus. Ce temps $T=4*(n+1)+T_0$ avec $T_0=110.10^{-6}$ s et n l'adresse de la station.

D) Adresse de station

Chaque station a une adresse unique configurable au niveau de la carte. Cette adresse, sur 3 octets, est composée d'une adresse réseau et d'une adresse station. 128 réseaux maximums (de 0 à 127) et 64 stations maximums (de 0 à 63).

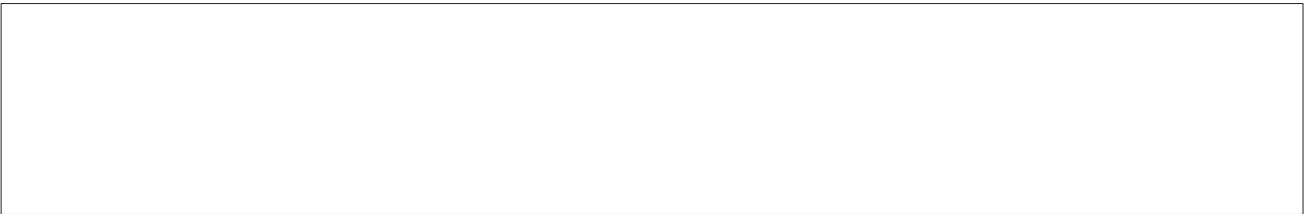
2.2.2 Syntaxe et utilisation

Il est possible d'avoir recours à trois méthodes pour communiquer via ce protocole.

- Mots communs (échange périodique) : ensemble de règles de communication qui définissent l'émission et la réception de données sur un réseau.



- Tables de données globales (échange périodique) : table créée par l'utilisateur qui permet de sélectionner la zone mémoire des automates que l'on veut mettre en commun.

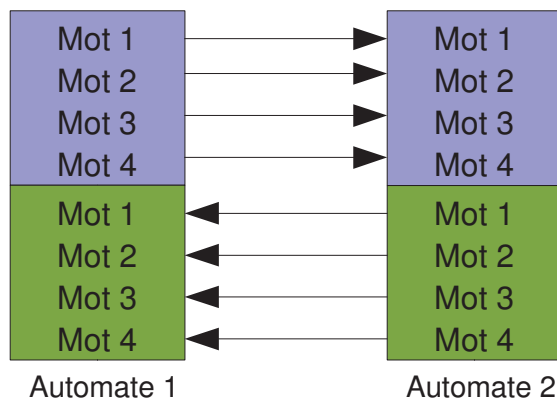


- Echanges aperiodiques : blocs fonctions permettant de copier la valeur d'un mot d'une station vers une autre.



A) *Mots communs*

L'échange est cyclique, la période étant de 40 ms. Périodiquement l'arbitre de bus envoie un identifieur aux stations connectées (adresse station entre 0 et 31). La station concernée répond en envoyant ses 4 mots communs (4*2 octets). Chaque automate peut accéder en lecture/écriture à ses 4 mots communs et en lecture à tous les autres.



Déroulement des échanges lors d'une communication entre l'automate et le coupleur (échange synchrone) :

Au début de chaque cycle automate, les mots communs des autres stations sont lus.

Le programme utilisateur gère l'application en tenant compte de ces nouveaux mots communs.

Il transfère en fin de cycle automate ses mots communs vers le coupleur.

Déroulement des échanges lors d'une communication entre stations :

Lorsque la station reçoit l'identifiant lui ordonnant de produire, elle envoie ses mots communs vers le réseau en diffusion, toutes les autres stations sont déclarées consommatrices des mots communs.

Lorsque la station est consommatrice de mots communs d'une autre station elle met à jour la zone mémoire correspondant de la mémoire partagée.

Les mots communs sont gérés comme du trafic périodique par l'arbitre de bus.

En ce qui concerne l'échange des mots communs entre le coupleur et l'unité centrale de l'automate, les coupleurs FipWay s'adaptent sur le bus d'entrée sortie des automates Schneider. Les informations entre l'automate et le coupleur sont échangées par l'intermédiaire d'une zone mémoire partagée composée de 128 mots de 16 bits. Chaque automate peut accéder en lecture/écriture à ses 4 mots communs et en lecture à tous les autres.

Sous PL7-Junior et PL7-Pro les mots communs sont adressés comme suit :

%NW{0}0	%NW{0}1	%NW{0}2	%NW{0}3	%NW{1}0	%NW{1}1	%NW{1}2											%NW{31}3
---------	---------	---------	---------	---------	---------	---------	-------	--	--	--	--	--	--	--	--	--	--	----------

%NW{n, i}j : Xk avec

n numéro de réseau (s'il n'y a qu'un réseau, ce numéro n'apparaît pas)

i n° de station [0-31] j n° de mot [0-3] k n° de bit [0-15]

Sous Unity Pro :

%NW0.0.0	%NW0.0.1	%NW0.0.2	%NW0.0.3	%NW0.1.0	%NW0.1.1	%NW0.1.2											%NW0.31.3
----------	----------	----------	----------	----------	----------	----------	-------	--	--	--	--	--	--	--	--	--	--	-----------

%NWn.i.j.k avec

n numéro de réseau i n° de station [0-31]

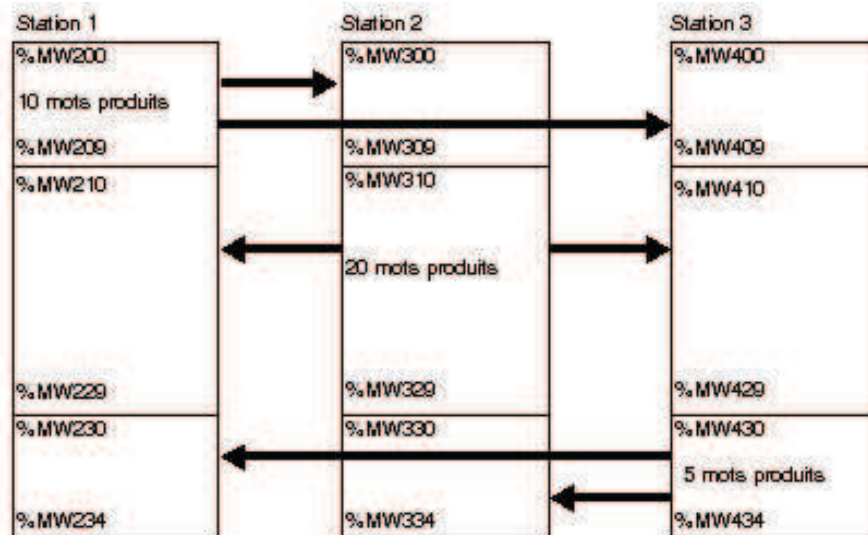
j n° de mot [0-3] k n° de bit [0-15]

B) Table de données globales

Sous PL7-Junior ou PL7-Pro (automates TSX37 et ultérieurs) le service d'utilisation des mots communs peut être remplacé par des tables de données globales. L'utilisateur doit donc choisir lors de la configuration du coupleur réseau le service qu'il souhaite activer. Il définit la décomposition de la zone « mémoire partagée » du coupleur. Chaque table doit être de même taille et au plus de 128 mots, tandis que chaque station produit au plus 32 mots.

Le graphe suivant représente le synoptique de production d'une table partagée constituée de 35 mots.

- La table de la station 1 débute en %MW200. Elle produit 10 mots de %MW200 à %MW209 et consomme de %MW210 à %MW234.
- La table de la station 2 débute en %MW300. Elle produit 20 mots de %MW310 à %MW329 et consomme de %MW300 à %MW309 et de %MW330 à %MW334.
- La table de la station 3 débute en %MW400. Elle produit 5 mots de %MW430 à %MW434 et consomme de %MW400 à %MW429.



Ce mécanisme est naturellement plus souple que les mots communs puisque la quantité d'information disponible pour chaque station est configurable par l'utilisateur. Il faut néanmoins savoir que ce service n'est pas compatible avec les automates TSX17 ou TSX47 (ancienne gamme).

C) Trafic aperiodique

Lorsqu'elle émet des messages périodiques une station peut demander un droit d'accès supplémentaire au réseau. L'arbitre autorise alors l'échange point à point d'un message (trafic aperiodique). Plusieurs fonctions sont à disposition afin de réaliser ces échanges : READ_VAR, WRITE_VAR et DATA_EXCH.

- Fonction READ_VAR

READ_VAR{adr#{s}SYS,type,i,j,typek:n,%MWm:4}

{s} numéro de la station objet de l'échange, du type {n°réseau.n°station}.

Type type de données lues '%MW' pour un mot, '%M' pour un bit, etc.

i adresse des données à lire

j : nombre de données à émettre

%typek:n adresse de stockage des données reçues
avec k l'adresse de base et n le nombre de données

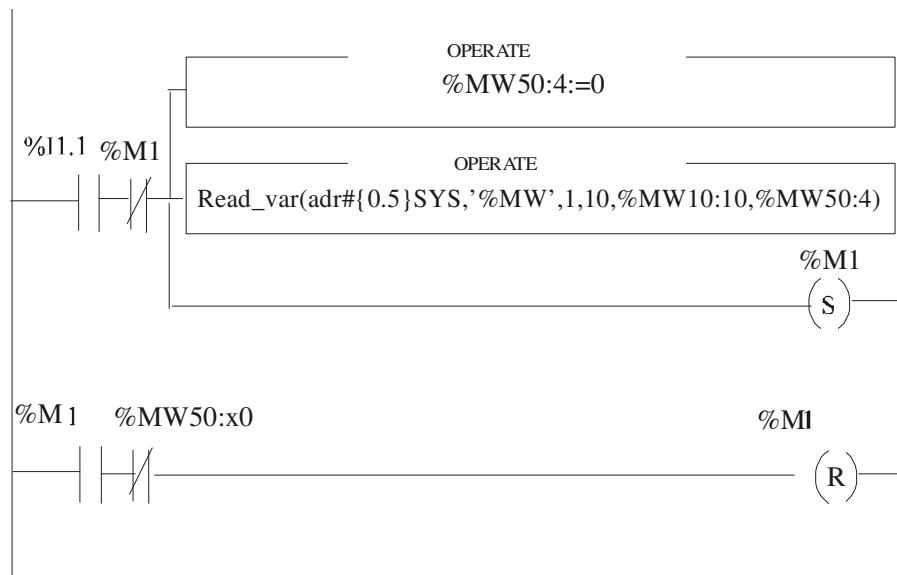
%MWm:4 4 mots de gestion, m décrit l'adresse de base des mots utilisés.

Parmi ces 4 mots, les deux premiers mots de gestion sont utilisés comme suit :

- %MWm : l'octet de poids fort contient un identifiant de l'échange, le bit de poids faible %MWm:x0 est le bit d'activité (à 1 lorsqu'un échange est en cours).
- %MWm+1 : l'octet de poids fort contient un rapport d'opération, l'octet de poids faible un rapport relatif à la communication. Ces codes renseignent sur le déroulement de la communication.

Exemple

Sur front montant de %I1.1, lecture des mots %MW1 à %MW10 de la station 5, ils seront stockés dans les mots %MW10 à %MW19. Les mots de gestion seront %MW50 à %MW53, le bit d'activité (communication en cours) est %MW50:X0, une transaction est initiée uniquement si la ligne est libre.



- Fonction WRITE_VAR

La fonction WRITE_VAR fonctionne suivant le même principe pour aller écrire dans un automate distant.

- Fonction DATA_EXCH

Data_exch(adr#{s}appI, [1-3], %MWi:j, %MWk:n, %MWm:4)

{s} numéro de la station objet de l'échange, du type {n°réseau.n°station}.

I numéro de bloc TXT ou de la fonction du correspondant

[1-3] transaction réalisée

1 émission suivie d'une réception (I/O)

2 émission

3 réception

%MWi:j données à émettre, i: l'adresse de base, j: le nombre de mots (en mode réception j=1)

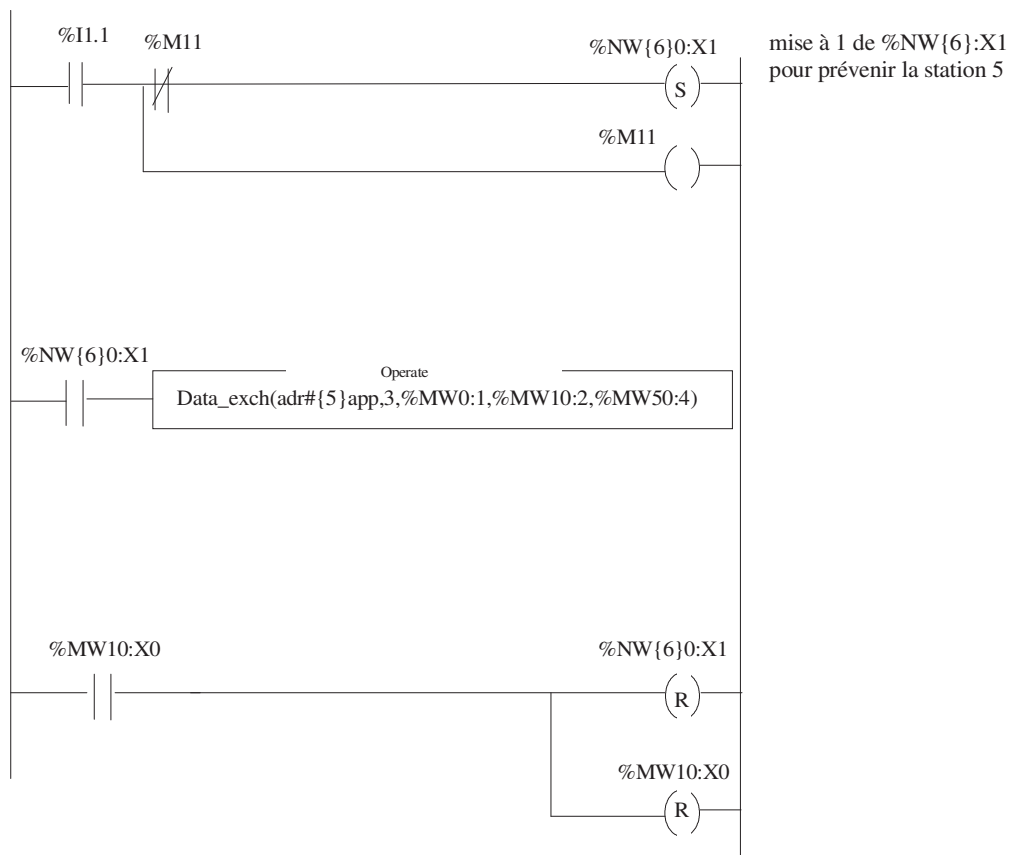
%MWk:n adresse de stockage des données reçues, k: l'adresse de base, n: le nombre de mots (en mode émission n=1)

%MWm:4 4 mots de gestion, m décrit l'adresse de base des mots utilisés. En émission, le quatrième mot doit contenir la taille en octet du message, c'est à dire 2*j octets. En réception, il s'ajuste automatiquement et contient le nombre d'octets reçus (2*n) octets.

Exemple

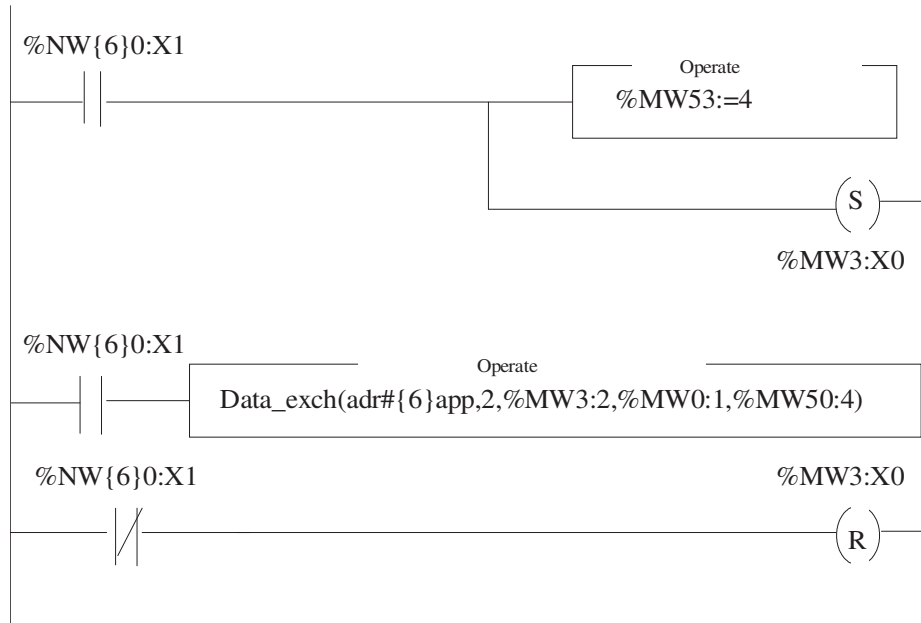
Station 6 (Station cliente)

Sur front montant de %I1.1, on demande 2 mots à stocker à partir de %MW10. Le bit 1 du mot commun 0 (%NW{6}0:X1) sert pour l'appel et %MW10:X0 sert d'accusé de réception.



Station 5 (Station serveur)

Lorsque %NW{6}0:X1 passe à 1, le nombre d'octets à envoyer est fixé dans %MW53. Les mots %MW3 et %MW4 sont envoyés, le bit %MW3:X0 sert d'accusé de réception. Il permettra au récepteur de garantir la réception des données.



2.2 Réseau MPI et protocole Profibus

2.2.1 Fonctionnement général

Dans un réseau Profibus, la communication est gérée selon un système d'anneau à jeton : des automates maîtres se passent un jeton alors que des esclaves sont interrogés par leur maître respectif.

Ce procédé est un procédé d'accès au bus hybride (jeton + réseau maître/esclave). La méthode du jeton garantit que toutes stations actives peuvent accéder au bus au moins une fois pendant un laps de temps défini. D'un autre côté, la méthode maître/esclave permet au maître possédant le jeton de scruter ses esclaves (stations passives). Il peut écrire ou lire sur ses esclaves.

Principe de l'anneau à jeton

Les stations actives se passent un jeton (télégramme) de manière périodique suivant un ordre préétabli (par ordre d'adresse ascendante). La station possédant le jeton reçoit un droit de parole, d'une durée préétablie, pour communiquer vers les autres stations actives et initier des transactions vers ses esclaves (lecture d'entrées/sorties sur des terminaux déportés). Lorsque la station a fini de communiquer ou que le temps de parole est expiré, elle passe le jeton à la station d'adresse immédiatement supérieure.

Un protocole d'interrogation des successeurs et des prédécesseurs permet de créer automatiquement cet anneau. Ce type de gestion d'accès fournit des bornes au temps de cycle réseau.

Le constructeur Siemens propose une couche supplémentaire reposant sur les protocoles Profibus ou MPI. Cette couche offre principalement 2 types de services :

- la communication par données globales basée sur un principe de diffusion permettant le dialogue entre 15 stations maximum.
- la communication en mode point à point entre 2 stations (Profibus DP). Elle permet d'initier des transactions entre 2 stations.

2.2.2 Utilisation des données globales

Les données globales (GD) sont des mémentos, des entrées, des sorties, des temporisations, des compteurs et des zones de blocs de données. La communication par données globales repose sur une table commune à chaque station. Cette table permet de définir l'ensemble des communications.

Chaque transaction par données globales est identifiée de la manière suivante :

$GD_{v,w,z}$ avec

v n° du cercle de connexion. Il regroupe plusieurs stations du réseau.

w n° du paquet, dans le cercle de connexion.

z n° des données dans le paquet. Un paquet peut regrouper plusieurs données.

Exemple 1

L'exemple qui suit présente une table de données globales qui paramètre la communication entre 3 stations. Les mnémoniques M1.0, M2.0, etc. correspondent aux bits M1.0, M2.0, etc. de la mémoire des stations.

Identificateur	Station 1	Station 2	Station 3
GD1.1.1	>M1.0		M6.0
GD2.1.1	>M10.2	M2.1	
GD2.1.2	>M10.4	M2.5	
GD2.2.1	M1.2	>M2.6	
GD2.2.2	M1.3	>M2.7	
GD3.1.1	>M10.5	M10.5	M6.1
GD3.2.1	M10.2	M10.2	>M6.3
GD3.2.2	M10.3	M7.2	>M7.2

La donnée globale GD2.1.2 désigne le 2^{ème} cercle de communication (il regroupe les stations d'adresse 1 et 2). Il s'agit du 1^{er} paquet de ce cercle (communication de la station 1 vers la station 2) et de la 2nde zone de données dans le paquet. Le symbole > précise que la station 1 est émettrice de ce paquet et que la zone de données est constituée du bit M10.4 de cette station. La station 2 est réceptrice et stocke le bit reçu dans M2.5.

Lorsque la station 1 a le jeton, les bits M1.0, M10.2, M10.4 et M10.5 de la station 1 sont copiés respectivement dans les bits M6.0 de la station 3, M2.1 de la station 2, M2.5 de la station 2 et M10.5 de la station 2.

Exemple 2

Exemple de table de données globales, vue de l'environnement Simatic manager.

	Identificateur GD	Station Magasin (4)\ CPU314 (4)	Station Maître (5)\ CPU314 (5)
1	GD 1.1.1	>MB80:10	MB20:10
2	GD 1.2.1	DB10.DBB0:20	>DB10.DBB0:20
3	GD		
4	GD		

Dans cet exemple, la station nommée Station Magasin\CPU314 envoie une zone de 10 octets commençant à MB80 à la station nommée Station Maître\CPU314 qui les dépose dans les octets MB20 à MB29.

La Station Maître est également définie comme émettrice et envoie 20 octets du DB10 (en commençant à l'adresse 0) à la CPU 314 de la Station Magasin qui les dépose dans les mêmes opérandes.

2.3 Réseau Modbus (compatible JBUS)

2.2.1 Fonctionnement général

Le réseau Modbus date de 1979. Il s'agit d'un standard de la communication de terrain. De nombreux équipements (entrées/sorties déportées, variateurs électroniques, automates, etc.) supportent ce protocole. On le retrouve parfois encapsulé dans un datagramme TCP/IP. Il permet d'atteindre des équipements industriels équipés de cartes Ethernet.

Il s'agit d'un réseau maître/esclaves. Le maître gère un mécanisme de questions/réponses avec les esclaves. La réponse renvoyée par l'esclave est identique à la requête lorsque celle-ci s'est correctement déroulée.

En cas de problème de transmission (erreur de code CRC) l'esclave reste muet, le maître renvoie, après un temps d'attente, la question, ceci pouvant se répéter jusqu'à trois fois. En l'absence de réponse le maître déclare l'automate esclave hors service.

En cas de problème d'interprétation de la requête par l'esclave (erreur du code fonction ou dans le champ de données) celui-ci génère une réponse dite d'exception interprétable par le maître.

Le maître peut également émettre un message en diffusion à l'intention de l'ensemble des stations, ce message reste sans réponse.

Ce protocole contient donc déjà une synchronisation des données basée sur un système de checksum (CRC16), il ne nous sera pas utile d'utiliser de sémaphores dans le cadre de son utilisation (pas pour synchroniser les échanges réseaux en tout cas).

Ce protocole regroupe plusieurs fonctions permettant d'aller lire ou écrire un nombre de bits définis sur un équipement industriel via un bus de communication.

2.2.1 Syntaxe et utilisation

A) Présentation de l'information

Afin d'utiliser ce protocole, 2 modes sont proposés, le mode ASCII et le mode RTU (Remote Terminal Unit).

- Mode ASCII

Chaque octet de données est codé en ASCII.

Par exemple : l'octet 0111 1011 soit en Hexa : x7B

devient x37 x42 soit en binaire 0011 0111 0100 0010

(x37 est le code ASCII de 7, et x42 et le code ASCII de B)

Les octets envoyés sont donc tous compris dans [A-F0-9], c'est à dire en hexa [x41-x46x30-x39]

La longueur maximale d'une trame est de 525 octets.

Chaque trame débute par une entête, l'octet x3A, code ASCII du caractère ":".

Chaque trame se termine par un délimiteur de fin, 2 octets x0Dx0A, code ASCII des caractères "CR/LF".

A chaque trame est associée une clé de contrôle LRC8 (checksum de 1 octet : somme de tous les octets de données avant codage ASCII, cette somme est ensuite complétée à 2, puis codée enASCII).

- Mode RTU

Aucun codage, ni délimiteur de fin, ni entête.

Tout caractère reçu après un silence supérieur à 3 caractères est considéré comme un début de trame.

Un silence sur la ligne au minimum égal à 3 caractères doit être respecté entre deux trames.

La longueur maximale des trames est de 261 octets.

Une clé CRC 16 (2 octets) est associée à la trame (voir l'algorithme sur la feuille jointe).

A) Format général des trames Jbus

Une trame Jbus est composée comme suit :

<i>Adresse esclave destination</i>	<i>Code fonction</i>	<i>Données</i>	<i>Code d'erreur</i>
--	--------------------------	----------------	--------------------------

avec :

- adresse de l'esclave (1 octet) : de 1 à 255, elle vaut 0 dans le cas d'une demande en diffusion,
- code fonction (1 octet) : indique la nature de la demande,
- données (1 octet) : données associées à la demande,
- code d'erreur CRC (2 octets) : permet de se préserver des erreurs de transmission.

B) Fonctions du protocole Jbus

Le détail des fonctions suivantes est fourni dans les annexes.

Fonction	1	:	lecture de n bits de sorties ou internes
	2	:	lecture de n bits d'entrées
	3	:	lecture de n mots de sorties ou internes
	4	:	lecture de n mots d'entrées
	5	:	écriture de 1 bit
	6	:	écriture de 1 mot
	7	:	lecture rapide de 8 bits

8	:	lecture des compteurs de diagnostic
11	:	lecture du compteur d'événements
12	:	lecture du buffer trace
13	:	commandes programmes
14	:	diagnostic de la fonction 13
15	:	écriture de n bits
16	:	écriture de n mots

A) Réponses d'exception

Une réponse d'exception renvoie le code fonction avec le bit de poids fort à 1, le champ de données est alors composé d'un code caractérisant le problème d'interprétation rencontré par l'esclave.

Code d'exception :	1	:	code fonction inconnu
	2	:	adresse incorrecte
	3	:	donnée incorrecte
	4	:	automate non prêt
	5	:	acquiescement
	6	:	non acquiescement
	7	:	défaut d'écriture
	9	:	chevauchement de zone

B) Compteurs de diagnostic

Chaque esclave est doté de 9 compteurs permettant de diagnostiquer le comportement du réseau. Ces compteurs sont accessibles par la fonction Jbus n°8.

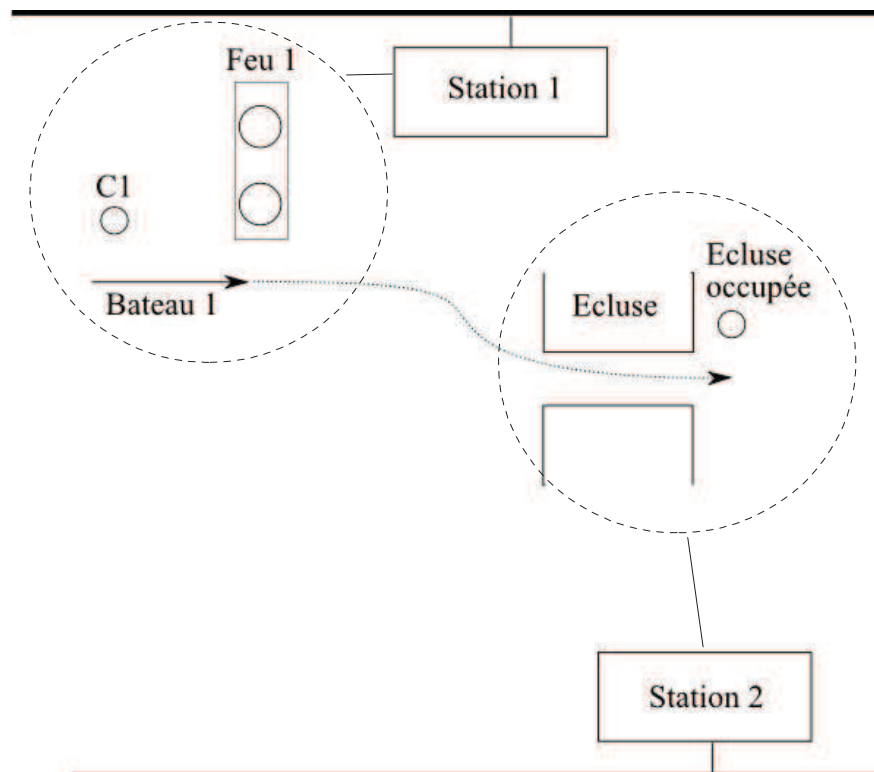
Compteur	CPT1	:	trames reçues sans erreur CRC
	CPT3	:	nombre de réponses d'exception
			etc...

EXERCICES

Exercice d'application 1 – Les sémaphores

Le schéma ci-dessous représente un écluse. Les bateaux arrivent en C1 où ils sont stoppés par le feu (rouge pour arrêt, vert pour passage). L'écluse doit être libre (aucun bateau dedans) pour que le bateau arrêté en C1 ait le droit de continuer sa route (passer dans l'écluse). Le capteur « Ecluse occupée » permet de connaître l'état de l'écluse (capteur de présence).

Proposez des grafçets (pour S1 et S2) permettant de gérer cette application en utilisant des sémaphores.



Remarque :

Dans cet exercice, la station 1 aura un bit « Demande de passage » qu'elle pourra renseigner (mettre à 1 ou à 0), et que la station 2 ne pourra que lire. De même, la station 2 possédera un bit « Ecluse libre ». Ces bits circuleront sur le réseau de manière périodique.

Exercice d'application 2 – Fonction Read Var

Sur activation de son entrée %I1.1, la station 2 doit lire les mots %MW10 à %MW19 de la station 1, et les stocker à partir de %MW1. Cette lecture doit être synchronisée, c'est-à-dire que la station 2 n'ira lire que lorsque la station 1 lui aura donné l'autorisation. Pour réaliser cette opération vous utiliserez les mots communs des stations.

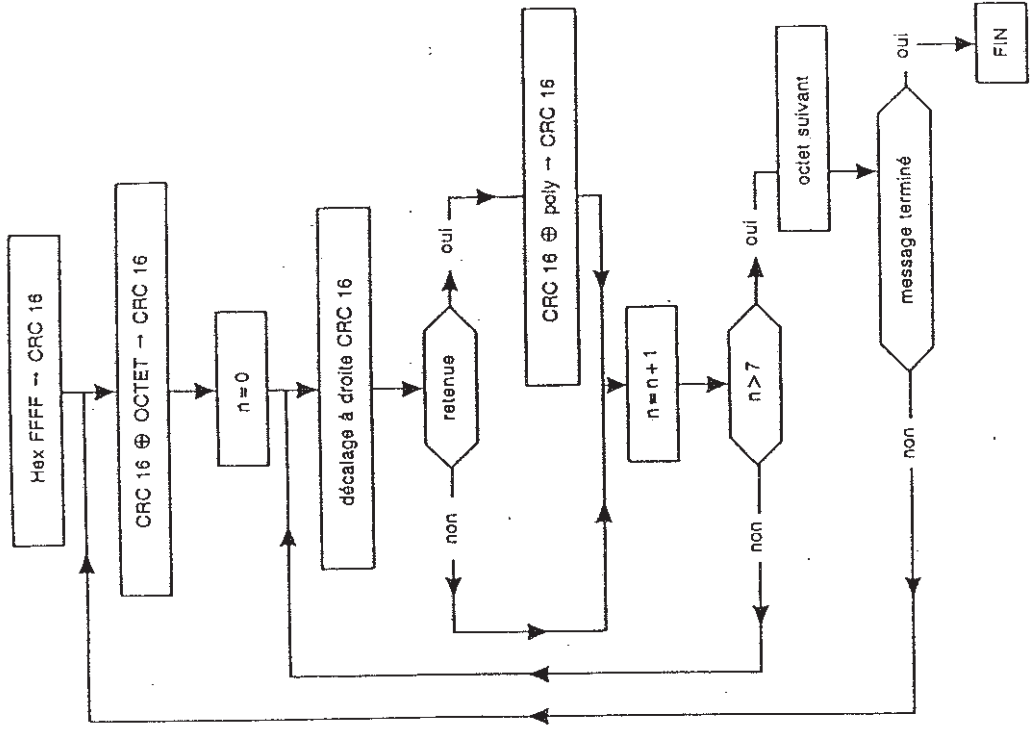
Station 2 (client)

Station 1 (serveur)

ANNEXES

MODBUS

6. Algorithme de calcul du CRC 16



⊕ = ou exclusif
 n = nombre de bits d'information
 POLY = polynôme de calcul du CRC 16 = 1010 0000 0000 0001
 (polynôme générateur = $x^4 + x^2 + x^{15} + x^{16}$)
 Dans le CRC 16, le 1^{er} octet émis est celui des poids faibles.

Exemple de calcul du CRC (trame 0207)

Initialisation registre CRC
 ⊕ du 1^{er} caractère

1111	1111	1111	1111
0000	0000	0000	0010

Décalage 1

1111	1111	1111	1101
0111	1111	1111	1101
1010	0000	0000	0001

Flag à 1, ⊕ polynôme

1101	1111	1111	1111
0110	1111	1111	1111
1010	0000	0000	0001

Décalage 2

1100	1111	1111	1110
0110	0111	1111	1111
1010	0011	1111	0011

Décalage 3

1010	0000	0000	0001
------	------	------	------

Décalage 4

Décalage 5

1001	0011	1111	1110
0100	1001	1111	1111
0010	0100	1111	1111
1010	0000	0000	0001

Décalage 6

Décalage 7

1000	0100	1111	1110
0100	0010	0111	1111
0010	0001	0011	1111
1010	0000	0000	0001

Décalage 8

⊕ 2^{ème} caractère

1000	0001	0011	1110
0000	0000	0000	0111

Décalage 1

1000	0001	0011	1001
0100	0000	1001	1100
1010	0000	0000	0001

Décalage 2

1110	0000	1001	1101
0111	0000	0100	1110
1010	0000	0000	0001

Décalage 3

1101	0000	0100	1111
0110	1000	0010	0111
1010	0000	0000	0001

Décalage 4

Décalage 5

1100	1000	0010	0110
0110	0100	0001	0011
0011	0010	0000	1001
1010	0000	0000	0001

Décalage 6

Décalage 7

1001	0010	0000	1000
0100	1001	0000	0100
0010	0100	1000	0010
0001	0010	0100	0001

Décalage 8

pois fort

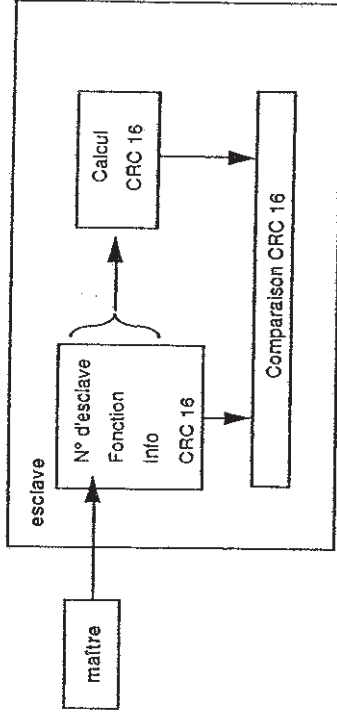
pois faible

Le CRC 16 de la trame est donc : 4112

Contrôle des messages reçus par l'esclave

Lorsque le maître émet une demande après avoir indiqué :

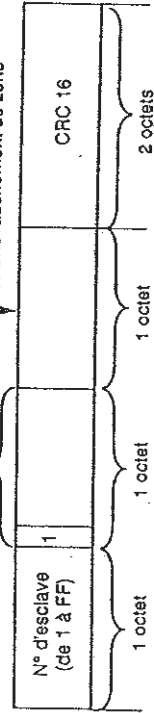
- le numéro d'esclave,
 - le code fonction,
 - les paramètres de la fonction.
- Il calcule et émet le contenu du mot de contrôle (CRC 16).
Lorsque l'esclave reçoit le message de demande, il le range en mémoire, calcule le CRC et le compare au CRC 16 reçu.



Si le message reçu est incorrect (inégalité des CRC 16) 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.

Contenu d'une réponse exception

- Code d'exception**
1. Code fonction inconnu
 2. Adresse incorrecte
 3. Donnée incorrecte
 4. Automate non prêt
 5. Acquiescement
 7. Non acquiescement
 8. Défaut d'écriture
 9. Chevauchement de zone



Exemple

demande : 1 9 0 0 0 0 0 CRC 16

réponse : 1 89_h 1 CRC 16

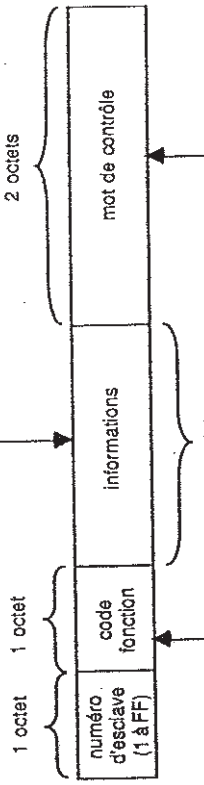
Remarque

Les réponses d'exception 5 et 7 sont relatives aux fonctions JBUS 13 et 14.

Présentation des trames de demande et de réponse

information nécessaire à la demande

- adresse bits, mots
- valeur bits, mots
- nombre de bits
- nombre de mots



Ce code permet de sélectionner les commandes disponibles

Lorsque le message est reçu par l'esclave, ce dernier lit le mot de contrôle et accepte ou refuse le message.

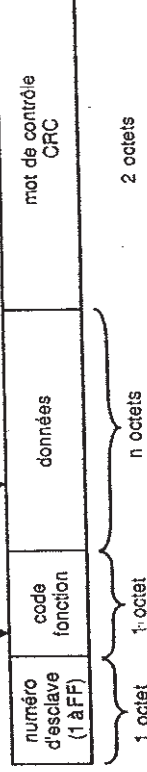
Le protocole JBUS possède 12 fonctions :

- Fonction 1 : lecture de n bits de sorties ou internes
- Fonction 2 : lecture de n bits d'entrées
- Fonction 3 : lecture de n mots de sorties ou internes
- Fonction 4 : lecture de n mots d'entrées
- Fonction 5 : écriture de 1 bit
- Fonction 6 : écriture de 1 mot
- Fonction 7 : lecture rapide de 8 bits
- Fonction 8 : diagnostic des échanges
- Fonction 11 : lecture du compteur d'événement
- Fonction 12 : lecture du buffer trace
- Fonction 13 : commandes programme
- Fonction 14 : diagnostic de la fonction 13
- Fonction 15 : écriture de n bits
- Fonction 16 : écriture de n mots

Nbr Trames reçues sur le Hôte pour les Services Transactions.

Commentaire, Commande API et Adresse.

- valeur des bits ou des mots lus
- valeur des bits ou des mots écrits
- nombre de mots ou nombre de bit



Le coupleur JBUS remplit et émet la trame de réponse sans aucune intervention de l'utilisateur.



1. Liste des messages d'erreur

Les erreurs susceptibles d'apparaître sont de deux types :

– les warnings (ou erreurs peu graves), qui n'interrompent pas l'exécution de la boîte fonctionnelle, ne sont pas signalés par le bit ERR. Seul le code correspondant est écrit dans WERR.

WERR =	- 1	coupleur occupé (par une BFC de même type)
	- 2	saturation du dialogue (nombre de tâches actives > 120)
	- 3	bloc mémoire non disponible (alloué par le système)
	- 4	boîte fonctionnelle non exécutée cause : traitement fin de cycle en cours (suite à une coupure secteur) ou UC en phase de mise au point (exécution du programme entité / entité ou cycle / cycle)
	- 5	perte dialogue sur le bus (sens UC → coupleur)
	- 65	exécution de la BFC suspendue car DSR ouvert.

– les erreurs qui interrompent l'exécution de la boîte fonctionnelle et provoquent le positionnement de ERR.

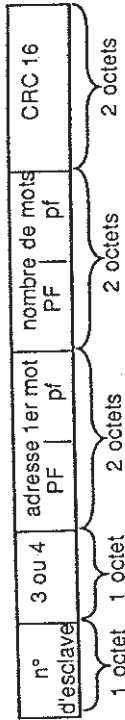
WERR =	0001	Time-out RTS/CTS dépassé	
	0002	Time-out en émission dépassé	
	0003	Pas de réponse de l'esclave	
	0004	Erreur de format, parité, overrun en réception (*)	
	0005	Erreur CRC (*)	
	0006	Time-out en réception dépassé	
	0007	Time-out en réception dépassé car DSR ouvert	
	0008	Trame de réponse incorrecte (N° esclave, code fonction, ...)	
	0009	Buffer des commandes répétitives saturé	
	0010	Time-out DTR/DSR dépassé (boîte modem)	
	0011	Table de stockage saturée avant d'avoir rencontré la condition de fin (BREAK, caractère de fin, ...)	
	0012	Réception d'un BREAK non attendu	
	0013	Nombre d'octets incompatible avec le type de données	
	0014	Type de la BFC incohérente avec le paramétrage de la voie (ex. : BFC READ sur voie paramétrée en PSP)	
	0015	paramètre N° esclave manquant (BFC MOD-COM)	
	0021	Code fonction inconnu] réponses d'exception du protocole JBUS
	0022	Adresse incorrecte	
	0023	Donnée incorrecte	
	0024	Automate non prêt	
	0025	Acquittement	
	0027	Non acquittement	
	0028	Défaut d'écriture	
	0029	Chevauchement de zones	
	003x	Pas de réponse aux trames de diagnostic (x = nombre de trames sans réponses)	
	0040	Perte de dialogue sur le bus (sens coupleur → UC)	
	0041	Erreur en lecture ou écriture de la mémoire de données	
	0042	Paramètre de la BFC incorrecte	
	0070	Carte non configurée	

(*) Ces défauts sont également signalés par la diode défaut voie (diode 2 pour voie 0, diode 5 pour voie 1).

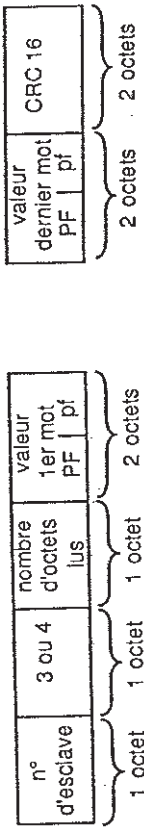
Lecture de N mots : fonction 3 ou 4

Le nombre de mots à lire doit être ≤ 125 .
 Fonction 3 : lecture de mots de sortie ou bits internes
 Fonction 4 : lecture de mots d'entrée

■ demande



■ réponse



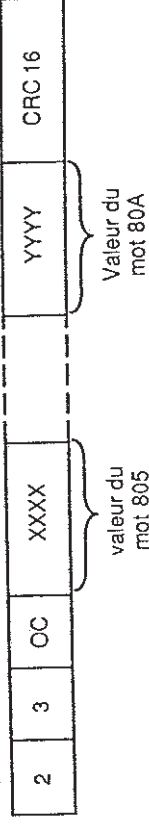
■ exemple

Lecture des mots 805 à 80A de l'esclave n° 2

Demande :



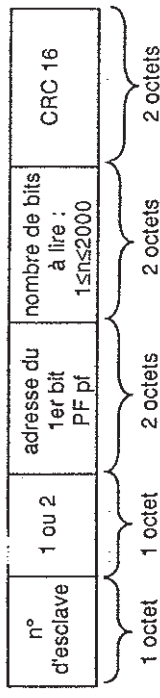
Réponse



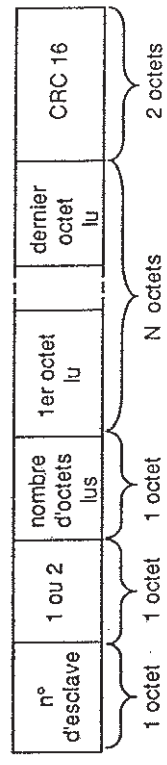
Lecture de N bits : fonction 1 ou 2

Fonction 1 : lecture de bits de sortie ou bits internes
 Fonction 2 : lecture de bits d'entrée
 Le nombre de bits à lire doit être ≤ 2000 .

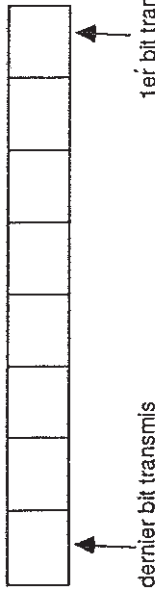
■ demande



■ réponse



détail d'un octet :

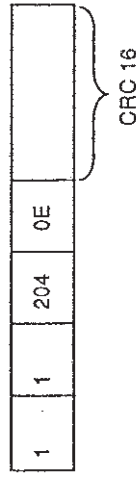


Les bits non utilisés dans l'octet sont mis à zéro.

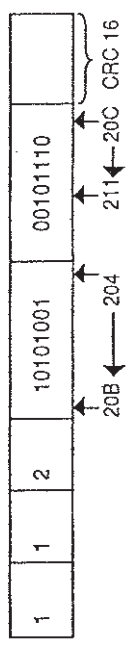
■ exemple

Lecture des bits 204 à 211 de l'esclave n° 1

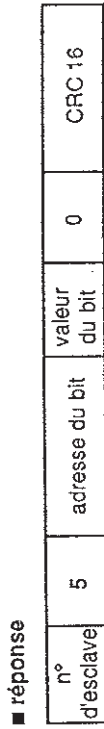
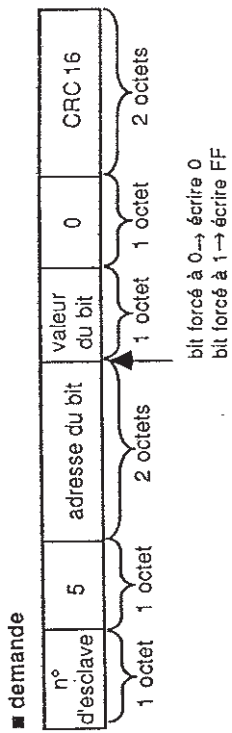
Demande :



Réponse :



Écriture d'un bit : fonction 5



Pour la fonction 5, la trame de réponse est identique à la trame de demande.

N.B. : si NES = 00, tous les esclaves exécutent le forçage sans émettre de réponse.

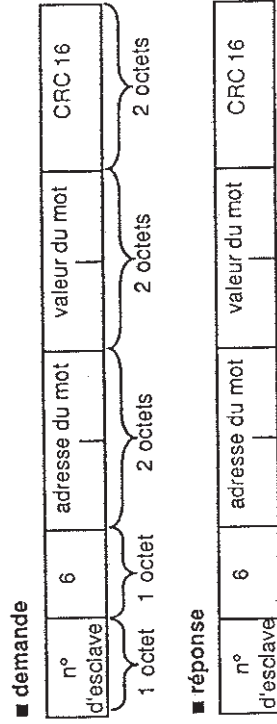


Exemple

Forçage à 1 du bit 210 de l'esclave n° 2.

2	5	210	FF	0	CRC 16
---	---	-----	----	---	--------

Écriture d'un mot : fonction 6



La réponse est un écho de la demande indiquant la prise en compte par l'automate de la valeur contenue dans la demande.

N.B. : si NES = 00, tous les automates exécutent l'écriture sans émettre de réponse.



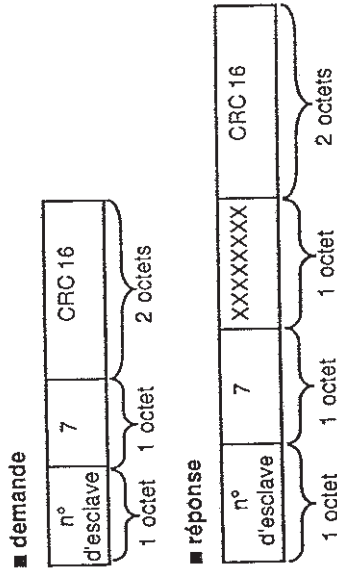
Exemple

Écriture de la valeur 1000 dans le mot d'adresse 810 de l'esclave 1.

1	6	810	1000	CRC 16
---	---	-----	------	--------



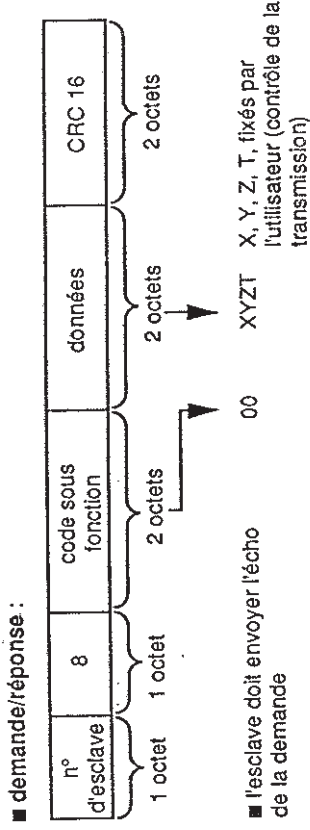
Lecture rapide de 8 bits : fonction 7



N.B.
Les adresses des 8 bits concernés sont fixés dans le coupleur esclave, au moment du paramétrage.

Lecture des compteurs de diagnostic : fonction 8

A chaque esclave sont affectés des compteurs d'événements (ou compteurs de diagnostic). Il y a au total 9 compteurs par esclave. Ces compteurs sont des mots de 16 bits :



- l'esclave doit envoyer l'écho de la demande

00	XYZT	X, Y, Z, T, fixés par l'utilisateur (contrôle de la transmission)
----	------	---
- remise à zéro des compteurs de diagnostic, annulation du mode déconnecté et reconfiguration du coupleur. Pas de réponse émise.
- remise à zéro des compteurs de diagnostic et du buffer trace.
- lecture du registre de diagnostic de l'automate

01	0000	
02	0000	voir ci-après contenu du registre
- modifié le caractère délimiteur de fin de trame (en mode ASCII). Par défaut ce caractère est LF (0A)
- mode déconnecté l'esclave est forcé à ne plus répondre
- remise à zéro des compteurs de diagnostic

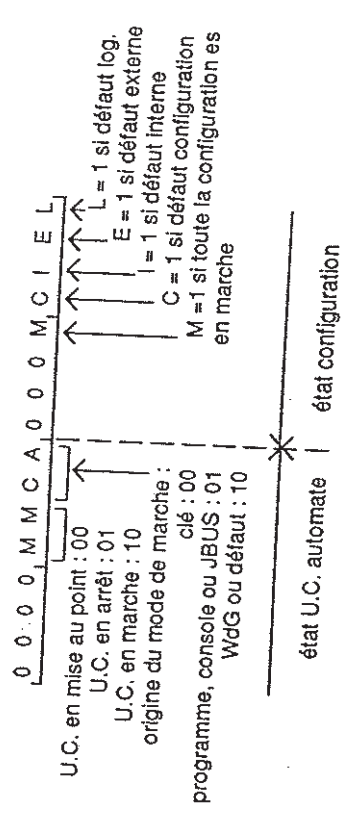
03	XY00	XY fixé par l'utilisateur : code ASCII du caractère de fin de trame
04	0000	(utiliser la sous-fonction 1 pour activer l'esclave)
0A	0000	
- lecture de nombre total des :

0B	XXXX	lors de la demande, XXXX vaut 00 00 lors de la réponse, XXXX est le contenu du compteur concerné
0C	XXXX	
0D	XXXX	
0E	XXXX	
0F	XXXX	
10	XXXX	
11	XXXX	
12	XXXX	

Détail du registre de diagnostic automate

(réponse de l'esclave à la fonction 8, sous code 02)

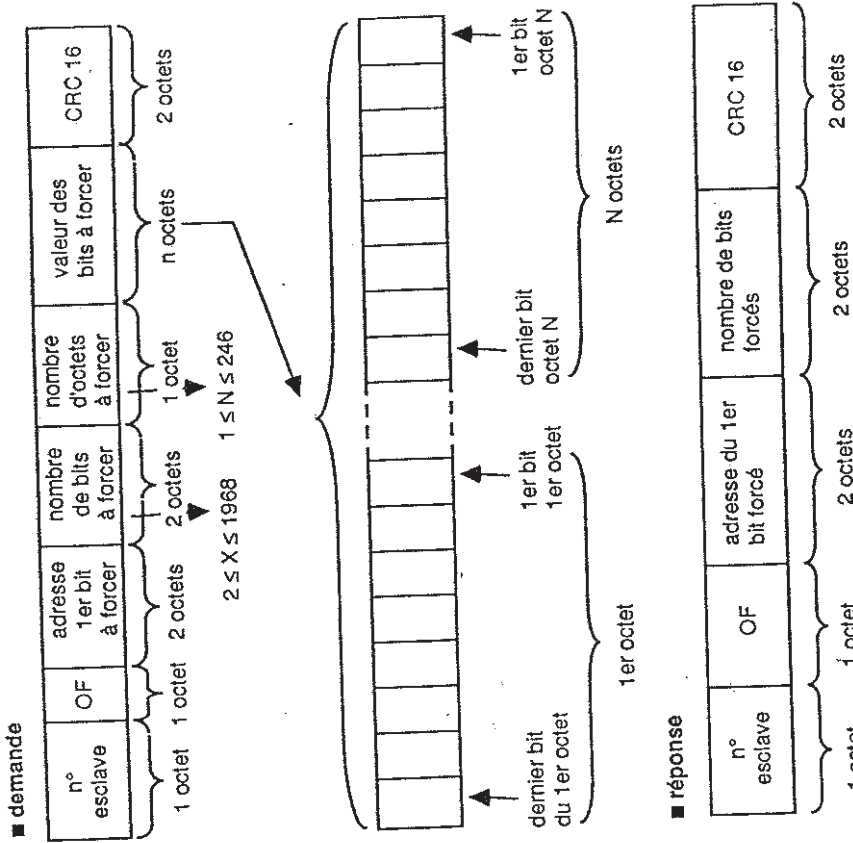
Le champ de données de la trame de réponse contient un mot de 16 bits représentatif de l'état de l'automate esclave :



Remarque

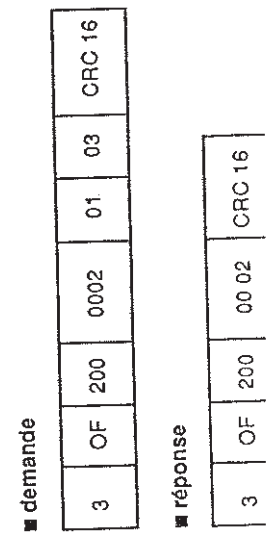
l'octet de poids faible est représentatif de l'état de l'ensemble de la configuration : Un défaut sur une seule carte de la configuration suffit à faire changer d'état le bit correspondant (C, I, E ou L). De même, le bit M ne passe à l'état 1 que si toutes les cartes de la configuration sont en marche.

Ecriture de n bits consécutifs : fonction 15

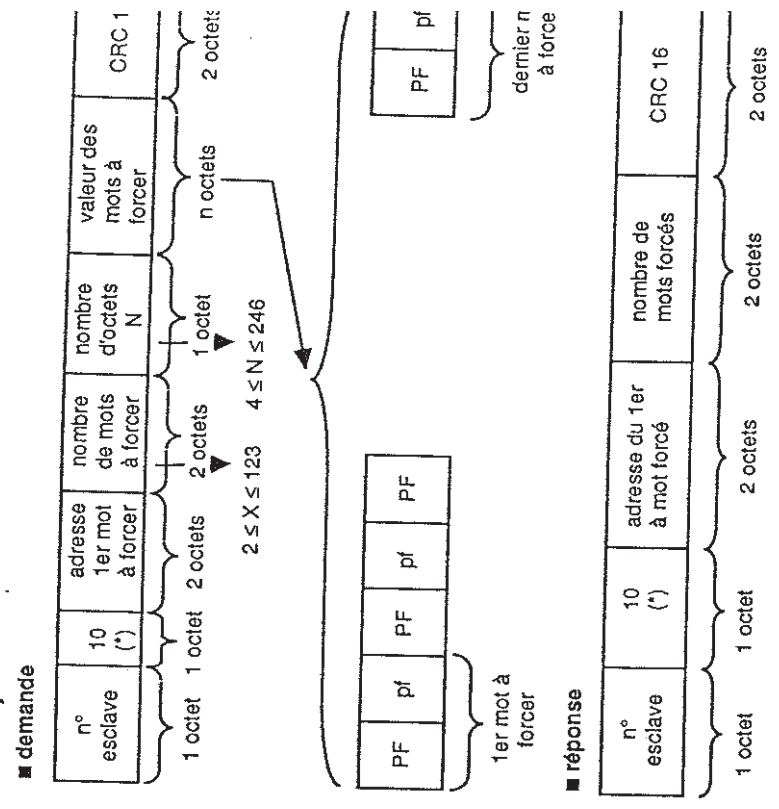


N.B. : si le numéro d'esclave est 0, tous les automates exécutent l'écriture sans émettre de réponse en retour.

■ exemple
Forcer à 1 les bits 200 et 201 de l'esclave 3.



Ecriture de n mots consécutifs : fonction 16 (commande 6)



N.B. : si le numéro d'esclave = 0, tous les automates exécutent l'écriture sans émettre de réponse en retour.

■ réponse
Forçage des mots 0800 à 0803 de l'esclave n° 1 :
(0800) = 000A
(0801) = 0010
(0802) = 0100
(0803) = 0000

