



<b>Chapitre</b>		<b>Page</b>
<b>1</b>	<b>Modbus</b>	<b>3</b>
	1.1 Définition d'un protocole	4
	1.2 Présentation du protocole Modbus	5
	1.3 Principe des échanges Modbus	6
<b>2</b>	<b>Gestion de protocole</b>	<b>7</b>
	2.1 Images des objets Modbus	8
	2.2 Echanges maître esclave	9
	2.3 Echanges processeur-coupleur esclave	11
	2.4 Echanges processeur-coupleur maître	13
	2.5 Fonctions gérées par le coupleur	15
	2.6 Détail des fonctions principales	16
	2.7 Détail des fonctions complémentaires	17
<b>3</b>	<b>Mise en œuvre matérielle</b>	<b>21</b>
	3.1 Présentation physique	22
	3.2 Installation du coupleur	22
	3.3 Raccordements	23
<b>4</b>	<b>Programmation du coupleur en esclave</b>	<b>25</b>
	4.1 Configuration du coupleur	26
	4.2 Echanges coupleur-automate	28
	4.3 Requêtes disponibles	32
	4.4 Modes de marches	34
<b>5</b>	<b>Programmation du coupleur en maître</b>	<b>37</b>
	5.1 Configuration du coupleur	39
	5.2 Echanges par coupleur-automate : fonctions principales	41
	5.3 Echanges par coupleur-automate : fonctions complémentaires	45
	5.4 Requêtes disponibles	50
	5.5 Modes de marche	52
<b>6</b>	<b>Bits tout ou rien registres</b>	<b>53</b>
	6.1 Interface tout ou rien	54
	6.2 Interface registre	56



<b>Chapitre</b>		<b>Page</b>
<b>7</b>	<b>Compléments de programmation</b>	<b>59</b>
	7.1 Signalisation des bits défauts	60
<b>8</b>	<b>Exemples de programmation</b>	<b>61</b>
	8.1 Programmation de l'esclave sur TSX 17-20	62
<b>9</b>	<b>Annexes</b>	<b>69</b>
	9.1 Performances de l'esclave	70
	9.2 Performances du maître	71
	9.3 Fonction Jbus	73
	9.4 Trames Modbus (RTU)	74



<b>Sous-chapitre</b>	<b>Page</b>
<b>1.1 Définition d'un protocole</b>	<b>4</b>
1.1-1 Généralités	4
1.1-2 Trame Modbus / Jbus	4
<b>1.2 Présentation du protocole Modbus</b>	<b>5</b>
1.2-1 Principe	5
<b>1.3 Principe des échanges Modbus</b>	<b>6</b>
1.3-1 Principe	6

## 1.1 Définition d'un protocole

---

### 1.1-1 Généralités

L'échange de données entre systèmes informatiques, automates programmables et autres systèmes intelligents doit s'effectuer dans un langage commun. Ce langage doit être le plus simple possible et compris par chaque interlocuteur; néanmoins, chaque échange doit pouvoir être contrôlé afin d'assurer l'intégrité des transferts. Les variables échangées sont alors insérées dans une trame constituée généralement de la façon suivante :



Chaque protocole définit la présence, le format, le contenu des différents groupes de variables entourant la zone données.

Cette structuration permet de définir le début des messages, la taille de ceux-ci, éventuellement le système auquel sont adressées les données, le type de fonction demandée, les variables elles-mêmes, un paramètre de contrôle et un code de fin, validant l'ensemble du message.

Cette trame est différente par son contenu et sa forme pour chaque type de protocole.

---

### 1.1-2 Trame Modbus / Jbus

En mode RTU, la trame définie pour le protocole Modbus ne comporte ni octet d'en tête de message, ni octet de fin de message. Sa définition est la suivante :



CRC 16 : Paramètre de contrôle polynomial (Cyclic Redundancy Check).

En mode ASCII, la trame est complète et se définit de la façon suivante :



LRC : Paramètre de contrôle (Longitudinal Redundancy Check),

CR : Retour chariot (Cariage Return),

LF : Nouvelle Ligne (Line Feed).

### Protocole Jbus

Le protocole Jbus utilise des fonctions identiques à celles de Modbus, ce qui permet avec ces fonctions communes de faire communiquer des équipements Modbus avec des équipements Jbus (pour plus de détails, se reporter en annexe 9.3).

## 1.2 Présentation du protocole Modbus

### 1.2-1 Principe

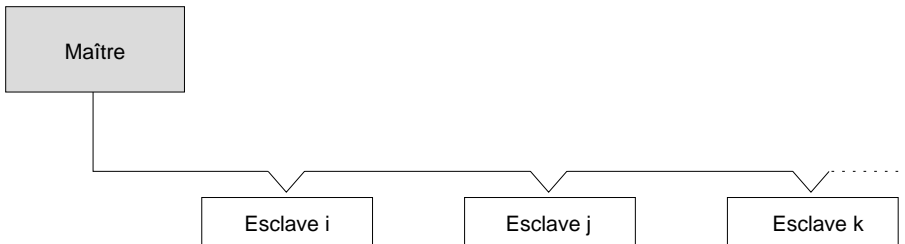
Le protocole Modbus est un protocole de dialogue créant une structure hiérarchisée (UN maître et plusieurs esclaves).

Le protocole Modbus permet d'interroger depuis le maître, un ou plusieurs esclaves intelligents. Une liaison multipoints relie entre eux, maître et esclaves.

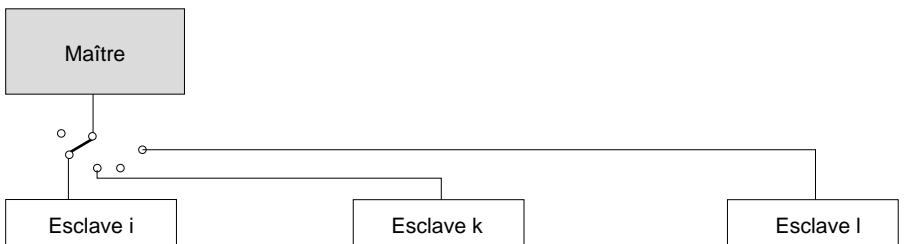
Deux types de dialogue sont possibles entre maître et esclaves :

- le maître parle à un esclave et attend sa réponse,
- le maître parle à l'ensemble des esclaves sans attente de réponse (principe de la diffusion générale).

Le maître gère les échanges et seul, en a l'initiative. Ce maître réitère la question lors d'un échange erroné et décrète l'esclave interrogé absent après une non réponse dans un temps enveloppe donné. Il ne peut y avoir sur la ligne qu'un seul équipement en train d'émettre. Aucun esclave ne peut de lui-même envoyer de message sans y avoir été invité.



Le principe du dialogue maître esclave peut-être schématisé sous une forme successive de liaisons point à point. Le maître gère la position d'un commutateur fictif, assimilé au numéro du destinataire qui est sélectionné.



Toute communication latérale (esclave à esclave) ne peut exister que si le logiciel application du maître a été conçu pour recevoir des données et les renvoyer d'un esclave à l'autre.

## 1.3 Principe des échanges Modbus

### 1.3-1 Principe

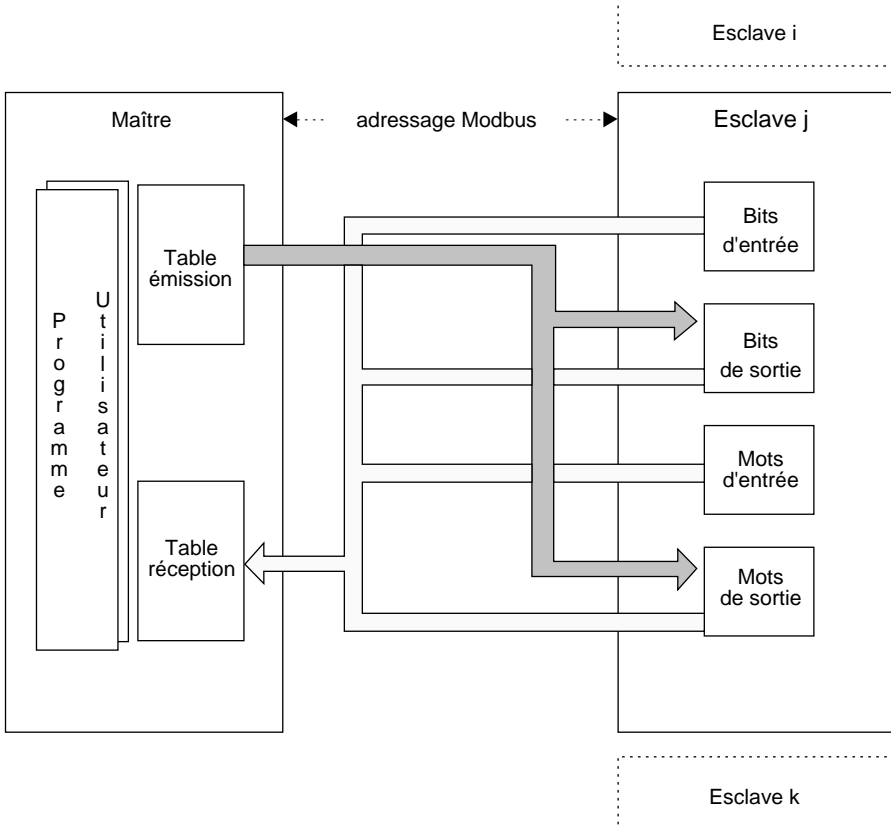
Le protocole Modbus permet d'échanger des données (bits et mots) entre un maître et des esclaves et assure le contrôle des échanges.

Par conséquent, dans chaque entité esclave, sont définies des zones de bits et de mots qui seront lues ou écrites par le maître.

Le protocole Modbus étant défini, il faut associer à chaque objet (bit ou mot) une adresse Modbus pour y accéder.

Un objet d'entrée peut être lu uniquement.

Un objet de sortie peut être lu ou écrit.





<b>Sous-chapître</b>	<b>Page</b>
<b>2.1 Images des objets Modbus</b>	<b>8</b>
<b>2.2 Echanges maître esclave</b>	<b>9</b>
2.2-1 Principe et contrôle des échanges	9
2.2-2 Surveillance de l'échange	9
2.2-3 Surveillance du maître	10
<b>2.3 Echanges processeur coupleur esclave</b>	<b>11</b>
2.3-1 Descriptions des échanges	12
2.3-2 Traitements coupleur et processeur	13
<b>2.4 Echanges processeur coupleur maître</b>	<b>13</b>
2.4-1 Description des échanges	14
2.4-2 Traitements coupleur et processeur	14
<b>2.5 Fonctions gérées par le coupleur</b>	<b>15</b>
<b>2.6 Détail des fonctions principales</b>	<b>16</b>
2.6-1 Lecture	16
2.6-2 Ecriture	16
<b>2.7 Détail des fonctions complémentaires</b>	<b>17</b>

---

## 2.1 Images des objets Modbus

---

Le protocole définit quatre types d'objets accessibles :

- bits d'entrée,
- bits de sortie,
- mots d'entrée,
- mots de sortie.

Tout esclave raccordé sur une ligne Modbus définit un nombre d'objets de chaque type.

A l'intérieur d'une même entité, tous les objets sont repérés par des adresses consécutives.

Le coupleur TSX SCG 11. esclave convertit une adresse Modbus en une adresse mémoire dans le processeur de l'automate esclave.

La table de correspondance est établie lors de la configuration du coupleur esclave. Tous les objets accessibles via Modbus sont localisés dans la zone Wi (mots internes) de l'automate.

### Objets bits

Pour les objets bits, le coupleur esclave connaît l'adresse i du mot interne Wi contenant le premier bit défini en position 0 (le bit concerné a donc pour image Wi,0) et son adresse Modbus correspondante.

L'adresse Modbus du premier bit configuré doit impérativement correspondre au bit de rang 0 d'un mot interne. Les autres bits sont consécutifs au premier.

Exemple : bits d'entrée

Adresse Modbus	Adresse processeur	Objet processeur	Objet physique
0100	50	W50,0	TXT1,D
0101	50	W50,1	T2,R
...	...	...	...
0115	50	W50,F	M3,S
0116	51	W51,0	I10,4

### Objets mots

Pour les objets mots, le coupleur esclave connaît l'adresse j du premier mot interne Wj et son adresse Modbus correspondante. Tous les objets sont situés dans la zone mots internes de l'automate.

Exemple : mots d'entrée

Adresse Modbus	Adresse processeur	Objet processeur	Objet physique
8000	200	W200	IW2,1
8001	201	W201	TXT,V
8002	202	W202	W202
...	...	...	...



---

## 2.2 Echanges maître esclave

---

### 2.2-1 Principe et contrôle des échanges

Le maître, ou organe de supervision, a l'initiative des échanges. Ce maître va s'adresser à un esclave en lui fournissant quatre types d'informations :

- adresse de l'esclave,
- fonction demandée à l'esclave,
- zone donnée (variable en fonction de la requête),
- contrôle d'échange.

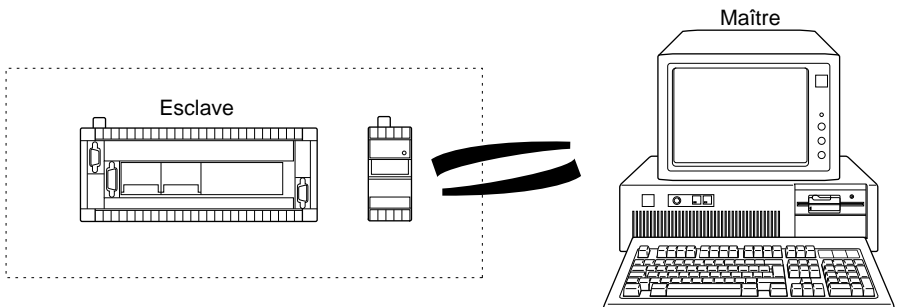
Le maître de la liaison attend la réponse de l'esclave avant d'émettre le message suivant, évitant ainsi tout conflit sur la ligne. Ceci autorise donc un fonctionnement en half-duplex.

Toute gestion d'échanges entre deux entités dialoguant par liaison série asynchrone, inclut évidemment des réponses d'exception lorsque sont apparus des défauts d'échange. Différents messages incohérents peuvent arriver au coupleur esclave. Dans ce cas, ce dernier répond sa non compréhension au maître qui prend ou non la décision de réitérer l'échange.

Cette gestion des messages erronés est transparente pour le programme utilisateur de l'esclave, qui n'est pas informé des erreurs détectées et traitées par le coupleur.

---

### 2.2-2 Surveillance de l'esclave



D'autre part, le maître a accès à un certain nombre d'informations détenues et gérées par le coupleur de l'esclave. Le maître accède à ces données par des codes fonctions particuliers (mode diagnostic, lecture du compteur d'événements...) qui se limitent à des échanges maître, coupleur TSX SCG 11. esclave.

---

Le coupleur de l'esclave gère par compteurs ses échanges avec le maître.  
Sont mémorisés les nombres de messages :

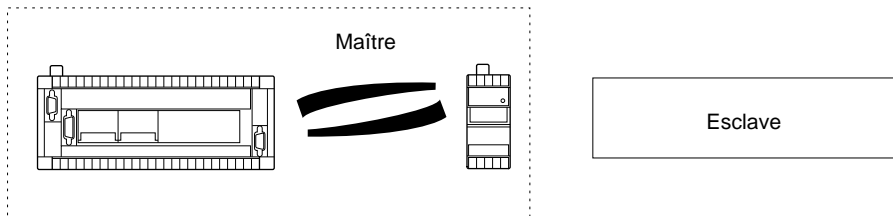
- reçus sur la ligne,
- reçus par le coupleur esclave avec erreur de checksum,
- émis avec code d'exception,
- reçus par le coupleur esclave,
- émis par le coupleur à son processeur sans réponse de ce dernier.

Toutes ces variables sont accessibles par le maître de façon transparente pour l'utilisateur de l'esclave.

Néanmoins, le programme utilisateur de l'automate esclave a également la possibilité de lire ces compteurs.

---

### 2.2-3 Surveillance du maître



Le coupleur maître gère par compteurs :

- les échanges entre processeur et coupleur maître,
- les échanges entre le maître et ses esclaves.

Ces compteurs peuvent être lus par le programme utilisateur de l'automate maître par l'utilisation du mode message (bloc fonction texte).

Les compteurs enregistrent les nombres suivants :

- demandes processeur prises en compte,
- demandes processeur refusées :

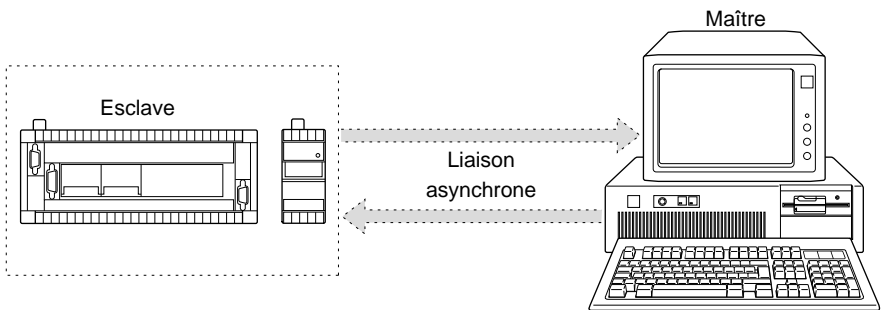
Le coupleur maître examine la demande du processeur et émet un refus dans les cas suivants :

- adresse esclave incorrecte,
- numéro de fonction inconnu,
- fonction interdite en diffusion générale,
- nombre d'objets à accéder trop élevé,
- longueur implicite de la demande processeur incorrecte.

- réponses esclave reçues correctes,
- réponses esclave reçues avec un code d'exception,

- messages sans réponse de l'esclave (délai écoulé),
- réponses esclave reçues avec erreur de checksum,
- réponses esclave reçues incohérentes :  
Le coupleur maître examine le message reçu de l'esclave et le déclare incohérent lorsque :
  - le message reçu présente un problème de cohérence vis à vis de la demande émise à cet esclave (adresse esclave, numéro de fonction ou nombre d'objets différent),
  - la longueur implicite du message reçu est incorrecte.
- messages réitérés :  
Le coupleur maître réitère un message à un esclave dans les cas suivants :
  - le message n'a pas eu de réponse (dans le délai configuré),
  - le message a été reçu avec une erreur de checksum,
  - le message reçu est incohérent,
  - le message comporte un défaut en réception.
- abandons de réitérations,
- délais d'attente de réponse arrivés à terme pendant la réception,
- réponses esclave reçues avec défaut en réception.

### 2.3 Echanges processeur-coupleur esclave



Le processeur esclave positionne le coupleur de liaison asynchrone TSX SCG 11. en attente de réception d'un message sur la ligne. Lorsqu'il reçoit un message, le coupleur l'analyse et transmet la demande du maître au processeur sous forme de requête.

Le programme utilisateur effectue alors l'opération demandée (écriture de tables, transfert de tables...). Le traitement terminé, le programme demande l'envoi au coupleur des données à échanger et la mise en réception du coupleur après traitement de la requête.

---

De façon asynchrone, le coupleur répond au maître en lui envoyant les quatre types d'informations suivantes :

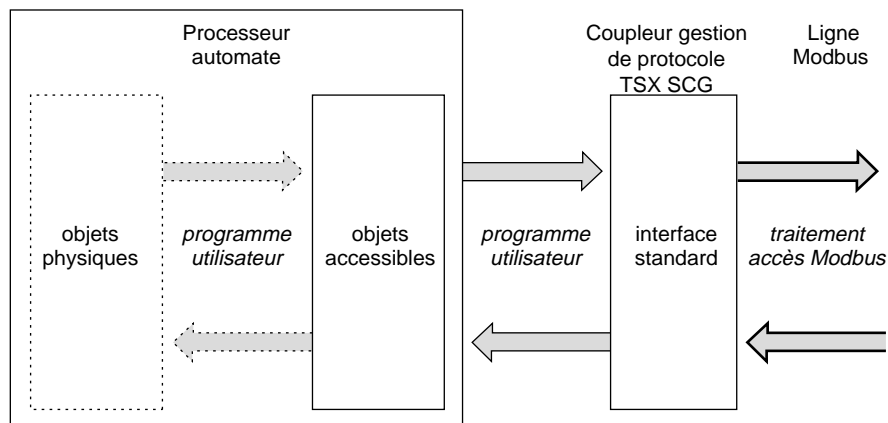
- numéro de l'esclave,
- type de fonction,
- zone données,
- contrôle d'échanges.

Le coupleur se retrouve alors en attente de réception d'un nouveau message afin de recommencer le même cycle d'échange.

---

### 2.3-1 Description des échanges

Les échanges entre le programme utilisateur et le coupleur TSX SCG 11. s'effectue par l'intermédiaire du mode message (table de mots de 16 bits). La programmation des échanges est réalisée à l'aide du bloc fonction texte. Elle nécessite quatre codes d'accès (ou requêtes). Le coupleur TSX SCG 11. n'utilise pas l'interface registre.



### 2.3-2 Traitements coupleur et processeur

#### Coupleur

Le coupleur effectue les opérations suivantes :

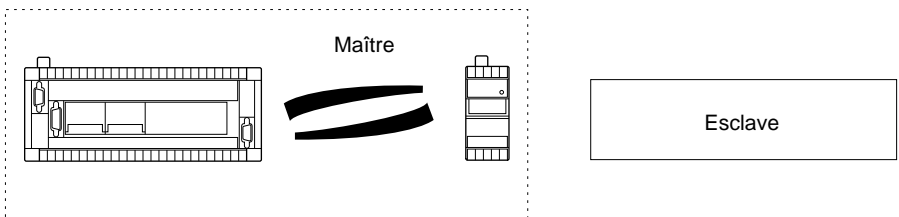
- réception et analyse des messages provenant du maître.
- mise en forme des messages pour la compréhension du processeur :  
le traitement de l'accès aux données Modbus par le coupleur TSX SCG 11. consiste notamment à transcrire les adresses Modbus des objets à accéder en adresses internes à l'automate.
- mise en forme de la réponse du processeur, et rajout du code de contrôle.
- envoi du message au maître.

#### Processeur

Le programme utilisateur gère :

- la prise en compte des demandes d'accès du module via l'interface message.
- la réalisation de l'accès demandé dans la table d'objets accessibles.
- éventuellement, les transferts entre la table d'objets accessibles et les objets physiques associés (valeurs de compteurs, temporisateurs, modules E/S...).

## 2.4 Echanges processeur-coupleur maître



Le processeur de l'automate a l'initiative des échanges. Il émet une requête à la voie 0 du coupleur de liaison asynchrone TSX SCG qui la prend en compte. Le coupleur émet un message à l'esclave concerné, comportant les quatre types d'informations suivantes :

- numéro de l'esclave,
- type de fonction,
- zone donnée,
- contrôle d'échange.

---

Le coupleur attend la réponse de l'esclave et transmet alors au processeur :

- un compte-rendu d'exécution,
- les données de l'esclave mises en forme.

Le cycle d'échange entre processeur et coupleur est terminé. Le coupleur est à nouveau prêt à traiter une requête du processeur.

---

### **2.4-1 Description des échanges**

Le dialogue entre le processeur et le coupleur s'effectue par l'interface message. Le code requête est identique au code de la fonction Modbus.

Les données échangées entre le processeur et le coupleur transitent par les tables d'émission et de réception associées au bloc texte.

---

### **2.4-2 Traitements coupleur et processeur**

#### **Coupleur**

Le coupleur effectue les opérations suivantes :

- contrôle de la validité de la requête processeur,
- mise en forme du message à émettre à l'esclave,
- envoi du message à l'esclave, et éventuellement réitération de ce message,
- mise en forme de la réponse à fournir au processeur.

#### **Processeur**

Le programme utilisateur gère :

- les informations nécessaires à l'échange,
  - le numéro de l'esclave,
  - le code fonction Modbus,
  - l'adresse Modbus des objets accédés,
  - les valeurs des objets à transférer,
- la transmission des demandes d'accès au module par l'envoi d'un bloc texte.
- le traitement de la réponse à la requête.

## 2.5 Fonctions gérées par le coupleur

Le coupleur TSX SCG 11. gère les fonctions Modbus dont la liste figure ci-dessous. Toutes les fonctions citées sont détaillées dans ce même document et sont les seules à être acceptées par le module. Toute autre requête sera rejetée, et la cause du rejet sera : "code fonction incorrect".

Dans le cas du coupleur maître, la requête "Modbus transparent" permet néanmoins de générer un numéro de fonction quelconque, compris entre 1 et 255. Dans ce cas, le programme utilisateur transmet au coupleur l'intégralité du message à transmettre, hormis le paramètre de contrôle rajouté par le coupleur.

Parmi les fonctions Modbus, on distingue :

- les fonctions principales qui permettent d'échanger un objet Modbus,
- les fonctions complémentaires utilisées par le maître pour obtenir des informations sur ses esclaves et contrôler les échanges.

Le module TSX SCG ne peut être utilisé qu'avec un automate TSX 17-20. Aussi une limitation apparait, due aux échanges par bloc texte (avec un maximum de 30 octets pour un automate TSX 17-20).

Les échanges de bits ou de mots ont des valeurs maximales qui sont indiquées dans le tableau ci-dessous.

La définition des fonctions (lecture, écriture...) s'entend vu du maître.

Les fonctions notées "D" peuvent être utilisées en diffusion générale.

Code	Nature des fonctions	D	maxTSX 17/20
01	Lecture de N bits de sortie		208
02	Lecture de N bits d'entrée		208
03	Lecture de N mots de sortie		14
04	Lecture de N mots d'entrée		14
05	Ecriture d'un bit de sortie	D	
06	Ecriture d'un mot de sortie	D	
07	Lecture du status d'exception		
08	Diagnostic		
11	Lecture du compteur d'évènements		
12	Lecture évènements connexion		
15	Ecriture de N bits de sortie	D	144
16	Ecriture de N mots de sortie	D	10
17	Identification esclave		

---

## **2.6 Détail des fonctions principales**

---

### **2.6-1 Lecture**

#### **Lecture de N bits de sortie**

##### **Code : 01**

Cette fonction permet d'accéder à des bits de sortie (bits 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 (bits que le maître ne peut que lire).

#### **Lecture de N mots de sortie**

##### **Code : 03**

Cette fonction permet la lecture de mots de sortie (mots pouvant être lus ou écrits) définis dans la mémoire d'un esclave.

#### **Lecture de N mots d'entrée**

##### **Code : 04**

Cette fonction, identique à la précédente possède les mêmes limites et s'adresse aux mots d'entrée (mots que le maître ne peut que lire).

---

### **2.6-2 Ecriture**

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

##### **Code : 05**

Cette fonction permet le positionnement à 0 ou 1 d'un bit de sortie (seuls accessibles en écriture) défini dans la mémoire d'un esclave.

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

##### **Code : 06**

Cette fonction effectue l'écriture d'un mot de 16 bits de sortie (seuls accessibles en écriture) défini dans la mémoire d'un esclave.

#### **Ecriture de N bits de sortie**

##### **Code : 15**

Cette fonction permet au maître d'écrire des bits de sortie (bits pouvant être lus ou écrits) dans la mémoire d'un esclave.

#### **Ecriture de N mots de sortie**

##### **Code : 16**

Cette fonction permet au maître d'écrire des mots de sortie (mots pouvant être lus ou écrits) dans la mémoire d'un esclave.



---

## 2.7 Détail des fonctions complémentaires

---

### Lecture du status d'exception

**Code : 07**

Cette fonction donne accès à huit bits de status enregistrant certains événements chez un esclave.

Dans le cas d'un automate TSX esclave, le status d'exception contient les poids forts du registre de sortie OWx,3, dont la définition des bits est laissée libre à l'utilisateur.

### Echo

**Code : 08/00**

Cette fonction diagnostic demande à l'esclave interrogé de retourner intégralement le message envoyé par le maître.

### Reprise communication

**Code : 08/01**

Cette fonction réalise la réinitialisation de la voie (suppression des messages en cours). La configuration de la voie est conservée.

### Lecture du registre diagnostic

**Code : 08/02**

Cette fonction permet l'accès à un mot de 16 bits contenant des informations sur l'état de l'esclave.

Dans le cas d'un automate TSX esclave, la structure du mot diagnostic est la suivante

- bit 0 = repli PWF
- bit 1 = non significatif
- bit 2 = état LOM (esclave en mode écoute seule)
- les autres bits sont non significatifs.

### Changement délimiteur ASCII

**Code : 08/03**

En mode ASCII, un octet est échangé par deux caractères ASCII, représentant son codage en hexadécimal. Les messages successifs sont séparés par un caractère délimiteur, initialisé à H'0A' (Line Feed). Ce caractère peut être changé par la fonction 08/03.

Cette fonction n'est pas supportée par le coupleur maître.

### Passage en mode écoute

**Code : 08/04**

Cette fonction force un esclave à passer en mode écoute seule (LOM).

Dans ce mode, l'esclave enregistre les messages qui lui sont adressés mais n'émet jamais de réponse.

Dans le cas d'un coupleur TSX esclave, le passage en mode écoute seule est signalé à son processeur par le bit 2 du registre IWx,1.

### Remise à zéro des compteurs

**Code : 08/0A**

Cette fonction effectue la remise à zéro de tous les compteurs d'un esclave, surveillant les échanges, ainsi que du registre diagnostic.

Dans le cas d'un esclave TSX, le mot diagnostic n'est pas remis à zéro.

---

### **Nombre de messages vus sur la ligne**

#### **Code : 08/0B**

Cette fonction permet l'accès à un compteur 16 bits (incrémentation de 0 à FFFF) totalisant le nombre de tous les messages vus sur la ligne et traités par l'esclave.

### **Nombre d'erreurs de checksum**

#### **Code : 08/0C**

Cette fonction permet l'accès à un compteur 16 bits, totalisant le nombre de messages reçus par l'esclave avec une erreur de checksum.

### **Nombre de réponses d'exception**

#### **Code : 08/0D**

Cette fonction permet l'accès à un compteur 16 bits, totalisant le nombre de réponses d'exception émises par le coupleur esclave vers le maître (après réception d'un message dont le contenu est incorrect).

### **Nombre de messages adressés à l'esclave**

#### **Code : 08/0E**

Cette fonction permet l'accès à un compteur 16 bits, totalisant le nombre de messages adressés à l'esclave, quelle que soit leur nature.

### **Nombre de non réponses du processeur esclave**

#### **Code : 08/0F**

Cette fonction permet l'accès à un compteur 16 bits, totalisant le nombre de messages émis par le coupleur esclave à son processeur et restés sans réponse de celui-ci.

### **Lecture du compteur d'événements**

#### **Code : 11**

Cette fonction permet de lire 2 mots de 16 bits :

- un status,
- un compteur d'événements.

Dans le cas d'un automate TSX esclave, le status est toujours nul. Le compteur d'événements est incrémenté à chaque réception de message correct (forme et contenu) sauf pour les réponses d'exception et les accès aux compteurs.

### **Lecture événements connexion**

#### **Code : 12**

Cette fonction permet d'accéder aux informations suivantes d'un esclave :

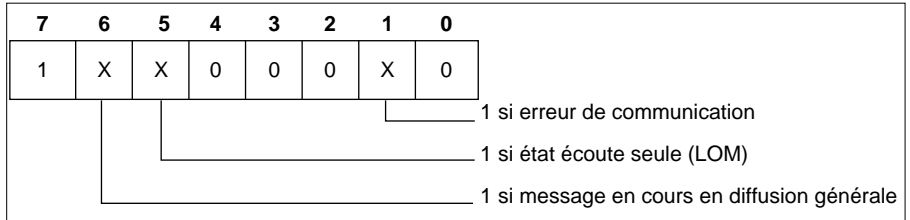
- mot status et compteur d'événements (identiques à la fonction 11).
- nombre de messages vus sur la ligne et traités par l'esclave (identique à la fonction 08 0B).
- contenu du buffer de trace (30 octets au maximum).

Cas de l'automate TSX esclave :

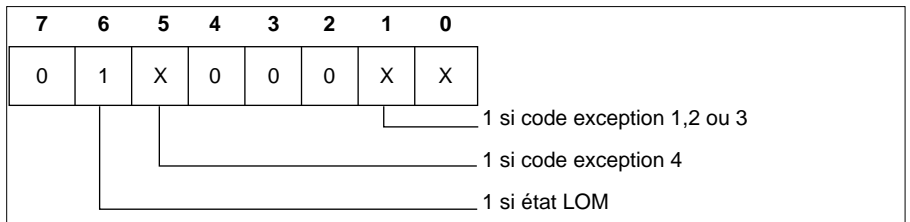
Le buffer de trace contient au maximum 30 octets (ou bien le nombre d'octets mis à jour). L'octet le plus récemment reçu est le premier transmis, le plus ancien est le dernier transmis (LIFO).

Les octets sont de quatre types :

- Réception esclave (mis à jour dès réception d'un message et avant traitement de celui ci).



- Emission esclave (mis à jour à la fin de l'émission d'un message ou du traitement d'un message dans le cas d'une écoute seule ou d'une diffusion générale).



- Entrée mode écoute (valeur hexadécimale H'20').
- Reprise communication (valeur hexadécimale H'xx').

### Identification de l'esclave

#### Code : 17

Cette fonction permet au maître d'obtenir d'un esclave les informations :

- identificateur esclave,
- état de fonctionnement,
- liste d'informations relatives au status et à la configuration de l'esclave.

Dans le cas d'un automate TSX esclave, la réponse comprend :

- le type d'automate connecté :
  - valeur H'17' pour un TSX 17-20.
- l'état de fonctionnement :
  - valeur H'00' pour PWF.
  - valeur H'FF' sinon.
- la liste des objets Modbus configurés (8 mots) :
  - nombre de bits de sortie,
  - adresse Modbus du premier bit de sortie,
  - nombre de bits d'entrée,
  - adresse Modbus du premier bit d'entrée,
  - nombre de mots de sortie,
  - adresse Modbus du premier mot de sortie,
  - nombre de mots d'entrée,
  - adresse Modbus du premier mot d'entrée.



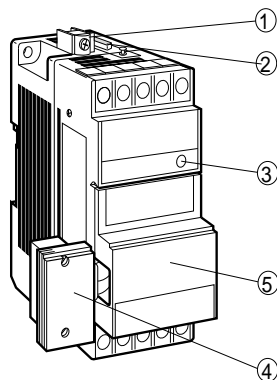


<b>Sous-chapître</b>	<b>Page</b>
<b>3.1 Présentation physique</b>	<b>22</b>
<b>3.2 Installation du coupleur</b>	<b>22</b>
<b>3.3 Raccordements</b>	<b>23</b>

### 3.1 Présentation physique

Le coupleur TSX SCG 1131 ou TSX SCG 1161 se présente sous la forme d'un boîtier, comprenant :

- ① Une borne de mise à la terre,
- ② Un connecteur 15 points pour raccordement à la liaison Modbus
- ③ Deux voyants de signalisation :
  - un voyant vert RUN (module sous tension, en état de fonctionnement),
  - un voyant rouge I/O (défaut du bus d'E/S).Les voyants NET et ADR ne sont pas utilisés en protocole Modbus,
- ④ Un connecteur 9 points et son câble pour raccordement au module précédent (bus TSX 17),
- ⑤ Un connecteur 9 points pour raccordement au module suivant (bus TSX 17).



Code de configuration du module : 63.

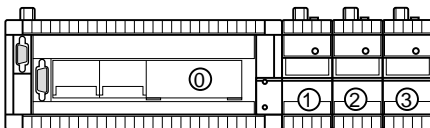
### 3.2 Installation du coupleur

Le coupleur TSX SCG 11 ne peut être utilisé qu'avec un automate TSX 17-20, équipé d'une cartouche micro-logicielle TSX P 17 20 FC (sans horodateur) ou FD (avec horodateur).

Le raccordement du coupleur à l'automate de base ou au bloc d'extension précédent (raccordement sur le bus) s'effectue par le câble intégré au coupleur. De ce fait le coupleur est toujours positionné à droite de ces éléments.

Il peut être installé indifféremment en première, deuxième ou troisième extension.

Au maximum deux coupleurs (TSX SCG 1131 ou TSX SCG 1161) peuvent être raccordés à un même automate TSX 17-20.



Le dernier bloc ou module d'extension d'une configuration doit être équipé sur son connecteur pour raccordement au module suivant, d'un adaptateur de ligne. Cet adaptateur est fourni séparément sous la référence TSX 17 ACC10.

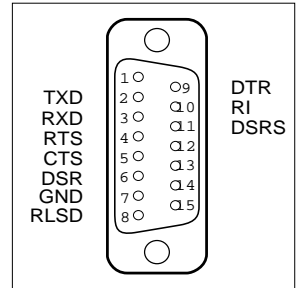
### 3.3 Raccordements

#### Raccordement TSX SCG 1131

Ce coupleur permet de réaliser une liaison conforme à la norme RS 232C, étendue au fonctionnement avec modem (9 signaux).

Le raccordement s'effectue par le connecteur femelle 15 points, situé sur la partie supérieure du coupleur. Le câblage de ce connecteur est le suivant :

- RXD : réception de données (3)
- TXD : émission de données (2)
- RTS : demande pour émettre (4)
- CTS : prêt à émettre (5)
- RI : indication d'appel (10)
- DTR : connecter le poste de données (9)
- DSR : poste de données prêt (6)
- RLSD : détection de la porteuse (8)
- DSRS : sélection de début binaire (11)

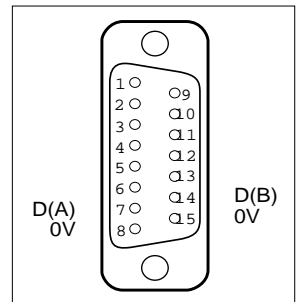


Pour plus de détails, se reporter au manuel TSX SCG 113 liaison asynchrone chapitre 5.2.

#### Raccordement TSX SCG 1161

Le coupleur permet de réaliser une liaison multipoint RS 485. Il comporte un circuit d'émission et un circuit de réception de signaux différentiels.

Le câblage du connecteur femelle 15 points, situé sur la partie supérieure du coupleur est le suivant :









<b>Sous-chapître</b>	<b>Page</b>
<b>4.1 Configuration du coupleur</b>	26
4.1-1 Définition de la table	26
4.1-2 Définition des paramètres	27
<b>4.2 Echanges coupleur-automate</b>	28
4.2-1 Description des échanges	28
4.2-2 Lecture de N mots	28
4.2-3 Ecriture de N mots	29
4.2-4 Lecture de N bits	30
4.2-5 Ecriture de N bits	30
4.2-6 Exemples	31
<b>4.3 Requêtes disponibles</b>	32
<b>4.4 Modes de marches</b>	34
4.4-1 Graphe général de fonctionnement du coupleur	34
4.4-2 Comportement du coupleur aux coupures et reprises secteur	35

---

## 4.1 Configuration du coupleur

---

Toutes les informations de configuration du coupleur sont transmises du processeur au coupleur par l'intermédiaire de bloc fonction texte.

Le coupleur n'ayant pas de mémoire sauvegardée, sa configuration est perdue sur chaque disparition de la tension secteur. Il est donc nécessaire d'effectuer l'envoi de la requête écriture configuration à chaque reprise secteur (SY0, SY1, IWx,1,F).

---

### 4.1-1 Définition de la table

La structure de la table à transmettre au coupleur est la suivante. Elle est constituée de 15 mots (30 octets) pouvant être définis dans l'automate en zone mémoire W ou en zone CW.

Les trois premiers mots sont codés en BCD (Binary Coded Decimal).

Tous les autres sont des valeurs binaires dont les 16 bits sont significatifs.

F			C	B			8	7			4	3			0
Type fonction : 2				Nombre bits				Parité				Bits de stop			
Vitesse de transmission - Bauds															
Réservé				Numéro de la station											
Adresse physique du premier mot de la zone bits de sortie															
Nombre de bits de sortie															
Adresse Modbus correspondante du premier bit															
Adresse physique du premier mot de la zone bits d'entrée															
Nombre de bits d'entrée															
Adresse Modbus correspondante du premier bit															
Adresse physique du premier mot de sortie															
Nombre de mots de sortie															
Adresse Modbus correspondante du premier mot															
Adresse physique du premier mot d'entrée															
Nombre de mots d'entrée															
Adresse Modbus correspondante du premier mot															

## 4.1-2 Définition des paramètres

### Type de fonction

paramètre spécifiant le mode de fonctionnement de la voie. Seul le traitement du protocole Modbus (esclave et maître) est défini dans le présent document.

- 1 = chaîne de caractères Full duplex
- 2 = protocole Modbus esclave

### Nombre de bits

définit le format des caractères échangés sur la ligne.

- 7 = transmission ASCII
- 8 = transmission RTU

### Parité

indique l'adjonction ou non d'un bit contrôlant une parité paire ou impaire.

- 0 = pas de parité
- 1 = parité impaire
- 2 = parité paire

### Nombre de stops

définit le nombre de bits de stops utilisés pour délimiter un caractère.

- 1 = un bit de stop
- 2 = deux bits de stop

### Important

Les trois paramètres précédents définissent le format de la transmission qui est lié aux possibilités de l'UART du coupleur TSX SCG. Les huit possibilités décrites ci-dessous sont offertes à l'exclusion de toute autre. La longueur des données (7 ou 8 bits) définit implicitement le mode ASCII ou RTU.

### Mode ASCII

- 7 bits + parité paire + 1 stop
- 7 bits + parité paire + 2 stops
- 7 bits + parité impaire + 1 stop
- 7 bits + parité impaire + 2 stops

### Mode RTU

- 8 bits + parité paire + 1 stop
- 8 bits + parité impaire + 1 stop
- 8 bits + 1 stop
- 8 bits + 2 stops

### Vitesse de la transmission

Le codage de la vitesse de transmission sur la ligne s'effectue sur 4 chiffres en **BCD** (exception faite pour la vitesse 19200 bauds qui sera codée H'1920').

Les possibilités ci-contre sont offertes.

- 19200 - 9600 - 4800 - 2400
- 1200 - 600 - 300 - 150.

---

## Numéro de la station Modbus

permet au module de reconnaître qu'un message lui est adressé.

Cette adresse, codée sur 3 chiffres en BCD prend une valeur comprise entre 1 et 247. B'001' à B'247'

Pour diffuser un message à tous les esclaves, l'adresse 0 est utilisée (diffusion générale).

## Tables de correspondance

Adresses automate - Adresses Modbus

Quatre tables de correspondance sont définies pour les quatre types d'objets accessibles suivants :

- bits de sortie,
- bits d'entrée,
- mots de sortie,
- mots d'entrée.

Chaque table est constituée de 3 paramètres (chaque paramètre est codé sur un mot) définissant :

- **l'adresse physique** du premier objet dans la mémoire automate. La zone concernée est obligatoirement située dans les mots internes.  
Exemple : l'adresse W125 est représentée par la valeur binaire 125,
- **le nombre total d'objets utilisés,**
- **l'adresse Modbus correspondante** du premier objet. Cette adresse doit correspondre à l'adresse émise sur la ligne par le maître pour accéder au premier objet dans la mémoire automate (adresse physique).

---

## 4.2 Echanges coupleur-automate

### 4.2-1 Description des échanges

Les transferts d'informations entre le coupleur et la mémoire de l'automate sont effectués sous forme de messages émis et reçus par bloc texte.

A l'initialisation des échanges, le programme utilisateur émet une requête vers le coupleur en utilisant un bloc texte en "exchange". Ceci positionne le coupleur en réception des messages Modbus émis par le maître sur la ligne.

Un cycle d'échange entre processeur et coupleur de l'esclave, nécessite les étapes suivantes :

### Etape 1

Dès réception d'un message Modbus, analysé et reconnu correct, le coupleur interprète la fonction demandée par le maître. Il transmet alors au programme utilisateur dans la table de réception du bloc texte, le code d'accès ainsi que les informations à traiter par le processeur.

Quatre codes d'accès sont possibles :

- lecture de N mots code d'accès 01 (fonctions Modbus 03 et 04)
- écriture de N mots code d'accès 02 (fonctions Modbus 06 et 16)
- lecture de N bits code d'accès 03 (fonctions Modbus 01 et 02)
- écriture de N bits code d'accès 04 (fonctions Modbus 05 et 15).

La définition des codes lecture et écriture s'entend vu du maître.

### Etape 2

Le programme utilisateur prend en compte et réalise l'accès aux données demandé par le coupleur.

### Etape 3

Le programme utilisateur lance un bloc texte en émission/réception avec les paramètres suivants :

- pour une lecture : les valeurs demandées par le maître sous le code requête lecture Modbus (TXTi,C = H'80').
- pour une écriture : le code requête fin d'écriture Modbus (TXTi,C = H'81') sans aucune donnée.

### Etape 4

Le coupleur traite la requête émise par le programme utilisateur à l'étape précédente. Il met en forme et envoie la réponse au maître.

Le bloc texte se retrouve en attente de réception par le coupleur d'un nouveau message Modbus.

## 4.2-2 Lecture de N mots

**Table de réception**

Code d'accès = 1
Adresse premier mot à lire
Nombre de mots à lire

**Table d'émission**

Valeur premier mot adressé
Valeur second mot adressé
Etc ...

### Variables

Code requête envoyé au coupleur : TXTi,C = H'80'

Longueur de la table d'émission : TXTi,L = 2 x (nombre de mots à lire)

---

### 4.2-3 Ecriture de N mots

**Table de réception**

Code d'accès = 2
Adresse du premier mot à écrire
Nombre de mots à écrire
Valeur du premier mot
Valeur du second mot
Valeur du troisième mot

**Table d'émission**  
vide

#### Variables

Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H}'81'$  (fin d'écriture)

Longueur de la table d'émission :  $\text{TXTi,L} = 0$

---

### 4.2-4 Lecture de N bits

**Table de réception**

Code d'accès = 3
Adresse premier mots à lire
Nombre de mots à lire

**Table d'émission**

Valeur premier mot adressé
Valeur second mot adressé
Etc ...

#### Variables

Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H}'80'$

Longueur de la table d'émission :  $\text{TXTi,L} = 2 \times (\text{nbre de mots à lire})$ .

---

### 4.2-5 Ecriture de N bits

**Table de réception**

Code d'accès = 4
Adresse du premier mot à écrire
Nombre de mots à écrire
Masque du premier mot = MSQ1
Masque du dernier mot = MSQ2
Valeur du premier mot
Valeur du second mot
Etc ...

**Table d'émission**  
vide

#### Variables

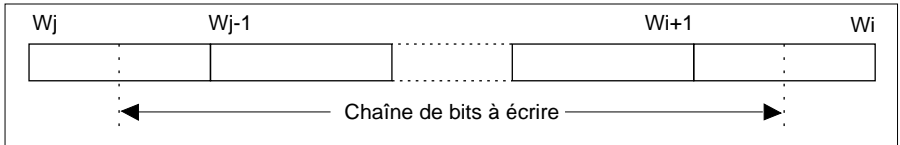
Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H}'81'$

Longueur de la table d'émission :  $\text{TXTi,L} = 0$

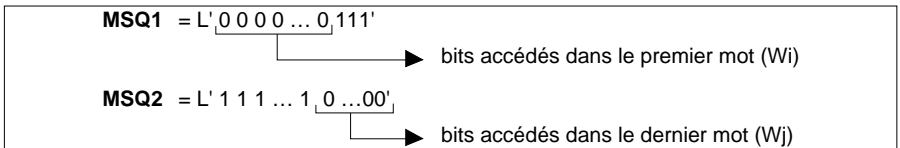
La chaîne de bits que le maître désire écrire se situe dans une suite de mots, dont le premier mot (qui contient les premiers bits) est défini par son adresse.

Avant d'être écrite, la chaîne de bits doit d'abord être remise à zéro, et ensuite un OU logique doit être effectué entre anciens et nouveaux tableaux.

Deux masques **MSQ1** et **MSQ2** sont définis en début et fin de tableau de mots afin de cerner la zone de bits accédée.



**MSQ1** et **MSQ2** sont utilisés avec un ET logique pour remettre à zéro la chaîne de bits



#### 4.2-6 Exemples

##### Lecture des mots W 28 à W 30

Table de réception

Wi = 1  
 Wi + 1 = 28  
 Wi + 2 = 3

Table d'émission

Wj = W 28  
 Wj + 1 = W 29  
 Wj + 2 = W 30

##### Ecriture des mots W 00 à la valeur 125 et W 01 à la valeur - 79

Table de réception

Wi = 2  
 Wi + 1 = 00  
 Wi + 2 = 2  
 Wi + 3 = 125  
 Wi + 4 = -79

##### Lecture des bits W 10,A à W 11,2

Table de réception

Wi = 3  
 Wi + 1 = 10  
 Wi + 2 = 2

Table d'émission

Wj = W 10  
 Wj + 1 = W 11

---

## Écriture des bits W 50,B à W 51,4 à L'11010 00101'

Table de réception

Wi = 4

Wi + 1 = 50

Wi + 2 = 2

Wi + 3 = H'07FF' = L'0000 0111 1111 1111'

Wi + 4 = H'FFE0' = L'1111 1111 1110 0000'

Wi + 5 = H'2800' = L'0010 1000 0000 0000'

Wi + 6 = H'001A' = L'0000 0000 0001 1010'

Traitement du programme utilisateur

W 50 AND Wi + 3 → W 50 (mise à zéro des bits accédés)

W 51 AND Wi + 4 → W 51 (mise à zéro des bits accédés)

W 50 OR Wi + 5 → W 50 (chargement des nouvelles valeurs)

W 51 OR Wi + 6 → W 51 (chargement des nouvelles valeurs).

---

## 4.3 Requêtes disponibles

---

### Écriture configuration

Code requête TXTi,C = H'40'

Compte-rendu positif TXTi,R = H'FE'

Compte-rendu négatif TXTi,R = H'FD'

Cette requête permet l'envoi de la configuration.

### Lecture configuration

Code requête TXTi,C = H'41'

Compte-rendu positif TXTi,R = H'71'

Compte-rendu négatif TXTi,R = H'FD'

Cette requête permet de relire la configuration envoyée à la voie.

Le bloc texte utilisé doit posséder une table de réception de taille égale à 30 octets pour contenir les variables de configuration.



### Lecture compteurs

Code requête	TXTi,C = H'70'
Compte-rendu positif	TXTi,R = H'A0'
Compte-rendu négatif	TXTi,R = H'FD'

Cette requête permet au processeur de lire les compteurs de surveillance des échanges. Le buffer de réception contient les 7 mots suivants :

- $W_i$  = nombre de messages reçus sur la ligne
- $W_i + 1$  = nombre de messages reçus par le coupleur avec erreur de checksum,
- $W_i + 2$  = nombre de messages émis avec code d'exception,
- $W_i + 3$  = nombre de messages reçus par le coupleur,
- $W_i + 4$  = nombre d'accès au processeur restés sans réponse,
- $W_i + 5$  = nombre d'événements,
- $W_i + 6$  = registre diagnostic.

### Lecture Modbus

Code requête	TXTi,C = H'80'
Compte-rendu positif	TXTi,R = H'B0'
Compte-rendu négatif	TXTi,R = H'FD»

Cette requête a deux utilisations précises :

- voie à l'état STOP : la requête permet de passer la voie à l'état RUN et de mettre le coupleur en attente d'un message sur la ligne, si l'ordre RUN voie est présent ( $OWx1,1 = 1$ ).

Le compte-rendu négatif apparait lorsque cette requête est envoyée alors que l'ordre RUN voie est absent.

- voie à l'état RUN : la requête constitue la réponse à une demande du maître de lecture de N bits ou de N mots.

### Fin d'écriture Modbus

Code requête	TXTi,C = H'81'
Compte-rendu positif	TXTi,V = H'B0

C'est la réponse à une demande du maître d'écriture de N bits ou N mots.

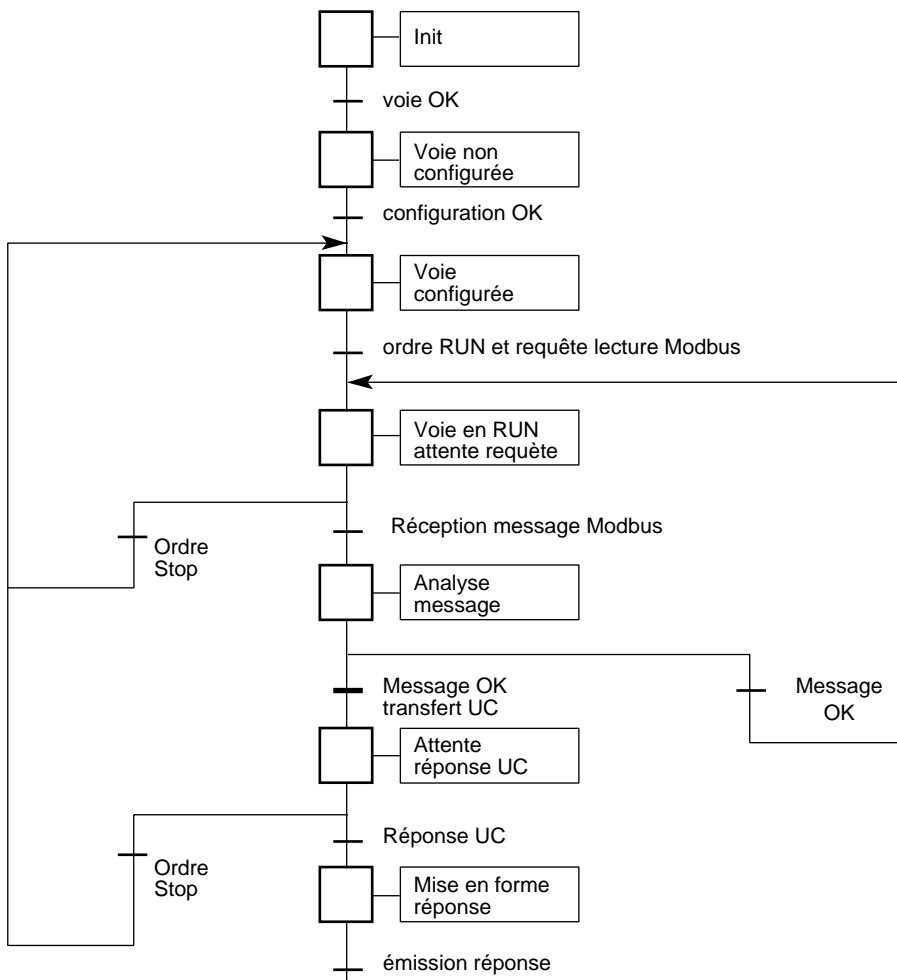
## 4.4 Modes de marche

### 4.4-1 Graphe général de fonctionnement du coupleur

Le graphe d'état représenté ci-dessous décrit le fonctionnement du coupleur TSX SCG 11, dont la configuration est le protocole Modbus esclave.

#### Important

La gestion des messages Modbus n'est prise en compte qu'après la demande de mise en RUN de la voie (bit  $OWx,1,1$ ), puis de l'envoi de la requête "lecture Modbus". Alors le bit RUN associé à la voie passe à l'état un (bit  $IWx,1,1$ ).



#### 4.4-2 Comportement du coupleur aux coupures et reprises secteur

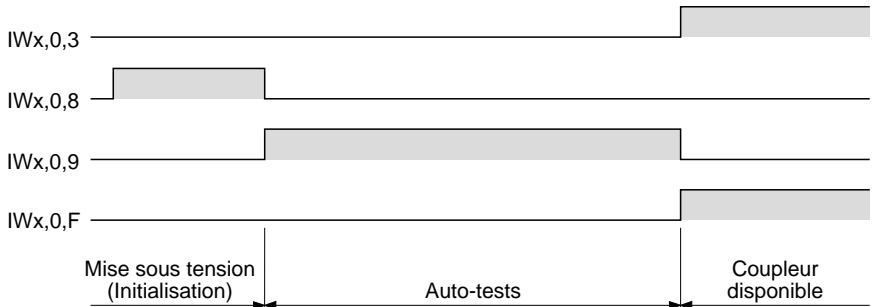
Le coupleur TSX SCG 11. n'a pas de mémoire sauvegardée, il perd sa configuration lorsqu'il n'est plus alimenté par l'automate.

Il est donc nécessaire de reconfigurer le coupleur :

- sur reprise à froid (SY0 = 1).
- sur reprise à chaud (SY1 = 1) lorsque la réserve d'énergie de l'alimentation a été épuisée.

##### Reprise secteur

A chaque mise sous tension le bit IWx,0,8 (Défaut bloquant) est forcé à 1 par l'automate. Après environ 300 ms, les interfaces TOR, registre et message devenant actifs dans le coupleur, celui-ci lance ses auto-tests (IWx,0,8 = 0 et IW x,0,9 = 1). En fin d'auto-tests, dont la durée est environ 1s, les bits Module disponible (IWx,0,3) et Repli PWF (IWx,1,F) sont mis à 1 pour indiquer que le coupleur est accessible en mode message. L'accquittement du bit PWF est possible, mais non obligatoire.



##### Conseil

Les alimentations 12V module et 5V automate peuvent avoir des comportements différents aux coupures secteur, provoquant une perte de la configuration alors que les bits SY0 et SY1 n'ont pas changé d'état. Pour cette raison il est préférable de tester en plus des bits SY0 et SY1, le bit IWx,1,F (Repli PWF).





<b>Sous-chapitre</b>	<b>Page</b>
<b>5.1 Configuration du coupleur</b>	<b>39</b>
5.1-1 Définition de la table	39
5.1-2 Définition des paramètres	39
<b>5.2 Echanges par coupleur-automate : fonctions principales</b>	<b>41</b>
5.2-1 Description des échanges	41
5.2-2 Lecture de N bits réception de sortie - Lecture de N bits d'entrée	41
5.2-3 Lecture de N mots de sortie - Lecture de N mots d'entrée	42
5.2-4 Ecriture d'un bit de sortie	42
5.2-5 Ecriture d'un mot de sortie	42
5.2-6 Ecriture de N bits de sortie	43
5.2-7 Ecriture de N mots de sortie	44
5.2-8 Exemples	44
<b>5.3 Echanges par coupleur-automate : fonctions complémentaires</b>	<b>45</b>
5.3-1 Lecture du status d'exception	45
5.3-2 Echo	45
5.3-3 Reprise communication	46
5.3-4 Lecture du registre diagnostic	46
5.3-5 Passage en mode écoute	46
5.3-6 Remise à zéro des compteurs	47
5.3-7 Lecture de nombre de messages reçus sur la ligne	47
5.3-8 Lecture du nombre d'erreurs de type checksum	47
5.3-9 Lecture du nombre de réponses d'exception	48
5.3-10 Lecture du nombre de messages adressés à l'esclave	48
5.3-11 Lecture du nombre de non réponse du processeur esclave	48
5.3-12 Lecture du compteur d'évènements	49
5.3-13 Lecture des évènements de connexion	49
5.3-14 Identification de l'esclave	49

---

---

<b>Sous-chapitre</b>	<b>Page</b>
<b>5.4 Requêtes disponibles</b>	<b>50</b>
<b>5.5 Modes de marche</b>	<b>52</b>
5.5-1 Graphe général de fonctionnement du compteur	52
5.5-2 Comportement du compteur sur coupures et reprises secteur	52

## 5.1 Configuration du coupleur

Toutes les informations de configuration du coupleur sont transmises du processeur au coupleur par l'intermédiaire de bloc fonction texte.

La table de configuration est constituée de 4 mots (8 octets) pouvant être définis dans l'automate en zone mémoire W ou en zone CW.

### 5.1-1 Définition de la table

	F	C	B	8	7	4	3	0
Wi	Type fonction		Nombre bits		Parité		Bits de stop	
Wi + 1	Vitesse de transmission - Bauds							
Wi + 2	Délai d'attente de réponse							
Wi + 3	0	0	0	0	0	0	0	Réitération

Les trois premiers mots définis dans la table de configuration sont codés en BCD (Binary Coded Decimal).

### 5.1-2 Définition des paramètres

#### Type de fonction

paramètre spécifiant le mode de fonctionnement de la voie. Seul le traitement du protocole Modbus (esclave et maître) est défini dans le présent document.

4 = protocole Modbus maître

#### Nombre de bits

définit le format des caractères échangés sur la ligne

7 = transmission ASCII

8 = transmission RTU

#### Parité

indique l'adjonction ou non d'un bit contrôlant une parité paire ou impaire.

0 = pas de parité

1 = parité impaire

2 = parité paire

#### Nombre de stops

définit le nombre de bits de stops utilisés pour délimiter un caractère

1 = un bit de stop

2 = deux bits de stop

---

## Important

Les trois paramètres précédents définissent le format de la transmission qui est liée aux possibilités de l'UART du coupleur TSX SCG.

Les huit possibilités décrites ci-dessous sont offertes à l'exclusion de toute autre. La longueur des données (7 ou 8 bits) définit implicitement le mode ASCII ou RTU).

### Mode ASCII

- 7 bits + parité paire + 1 stop
- 7 bits + parité paire + 2 stops
- 7 bits + parité impaire + 1 stop
- 7 bits + parité impaire + 2 stops

### Mode RTU

- 8 bits + parité paire + 1 stop
- 8 bits + parité impaire + 1 stop
- 8 bits + 1 stop
- 8 bits + 2 stops

### Vitesse de transmission

Le codage de la vitesse de transmission sur la ligne s'effectue sur 4 chiffres en BCD (exception faite pour la vitesse 19200 bauds qui sera codée H'1920'). Les possibilités ci-contre sont offertes.

19200 - 9600 - 4800 - 2400 -  
1200 - 600 - 300 - 150

### Délai d'attente de réponse

Temps au bout duquel le coupleur TSX SCG 11. maître estime que l'esclave n'a pas reçu la demande émise. La valeur de ce temps doit toujours être supérieure au maximum des temps de réponse des différents esclaves connectés. La valeur est codée en BCD sur un mot. Elle fixe l'intervalle de temps de 20 ms à 99,99s (1 minute et 40 secondes). (les valeurs B'0000', B'0001' et B'0002' fixent le temps à 20 ms).

**B'0000' à B'9999'** incréments de base de temps (10 ms).

### Nombre de réitérations

Précise au coupleur maître le nombre maximum de tentatives supplémentaires d'envoi d'une demande restée sans réponse, ou ayant occasionné une réponse incorrectement reçue.

Lorsque ce nombre est nul, le coupleur n'effectue qu'un seul essai d'envoi. Chaque nouvelle demande se fait après écoulement du délai d'attente précédemment cité.

**B'0' à B'9'**



## 5.2 Echanges coupleur-automate : fonctions principales

### 5.2-1 Description des échanges

Les transferts d'information entre la mémoire de l'automate et le coupleur sont effectués sous forme de messages véhiculés par bloc texte.

Les requêtes adressées au coupleur sont codées dans la variable  $TXTi,C$  du bloc fonction texte, et ont le même numéro que la fonction Modbus considérée.

Le programme utilisateur positionne les paramètres associés à chaque requête dans la table d'émission du bloc fonction texte :

- le numéro de l'esclave destinataire,
- l'adresse mémoire Modbus des objets à traiter,
- le nombre d'objets,
- etc...

Après traitement de la réponse de l'esclave, le coupleur complète la table de réception du bloc texte avec les informations nécessaires pour l'interprétation de la requête par le programme utilisateur.

Seules les fonctions Modbus principales, qui permettent d'échanger un objet Modbus, sont décrites ci-dessous.

### 5.2-2 Lecture de N bits réception de sortie - Lecture de N bits d'entrée

**Table d'émission**

Numéro de l'esclave
Adresse Modbus du 1er bit à lire
Nombre de bits à lire

**Table de réception**

Nombre de mots lus
Masque du dernier mot = MSQ
Valeur du premier mot
Valeur du second mot
Etc ...

#### Variables


Code requête envoyé au coupleur :  $TXTi,C = H'01'$  (bits de sortie)

$H'02'$  (bits d'entrée)

Longueur de la table d'émission :  $TXTi,L = 6$

Le premier bit de la chaîne de bits à lire par le maître correspond toujours au bit de poids le plus faible du premier mot.

Le masque du dernier mot permet d'identifier les derniers bits significatifs de la chaîne de bits.

$MSQ = L' 1 1 1 1 \dots 1, 0 \dots 00'$   


---

### 5.2-3 Lecture de N mots de sortie - Lecture de N mots d'entrée

**Table d'émission**

Numéro de l'esclave
Adresse Modbus du 1er mot à lire
Nombre de mots à lire

**Table de réception**

Nombre de mots lus
Valeur du premier mot
Valeur du second mot
Etc ...

#### Variables

Code requête envoyé au coupleur :

$\text{TXTi,C} = \text{H}'03'$  (mots de sortie)  
 $\text{H}'04'$  (mots d'entrée)

Longueur de la table d'émission :

$\text{TXTi,L} = 6$

---

### 5.2-4 Ecriture d'un bit de sortie

**Table d'émission**

Numéro de l'esclave
Adresse Modbus du bit de sortie
Valeur à écrire

**Table de réception**

vide

La valeur à écrire est égale à :

- $\text{H}'0000'$ , pour écrire le bit à zéro,
- $\text{H}'00FF'$ , pour écrire le bit à un.

#### Variables

Code requête envoyé au coupleur :

$\text{TXTi,C} = \text{H}'05'$

Longueur de la table d'émission :

$\text{TXTi,L} = 6$

---

### 5.2-5 Ecriture d'un mot de sortie

**Table d'émission**

Numéro de l'esclave
Adresse Modbus du mot de sortie
Valeur à écrire

**Table de réception**

vide

#### Variables

Code requête envoyé au coupleur :

$\text{TXTi,C} = \text{H}'06'$

Longueur de la table d'émission :

$\text{TXTi,L} = 6$

### 5.2-6 Ecriture de N bits de sortie

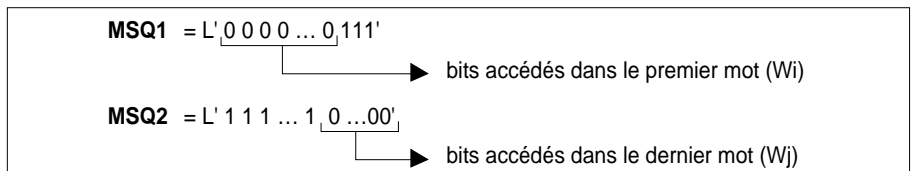
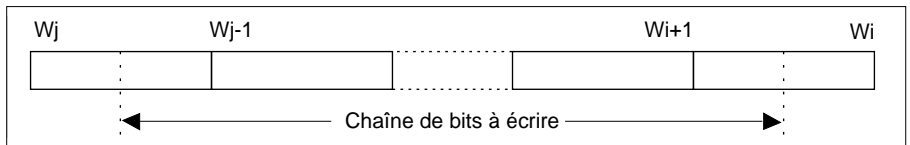
Table d'émission

Numéro de l'esclave
Adresse Modbus du 1er bit à écrire
Nombre de mots
Masque du premier mot = MSQ1
Masque du dernier mot = MSQ2
Valeur du premier mot
Valeur du second mot
Etc ...

Table de réception vide

#### Variables

La chaîne de bits que le maître désire écrire se situe dans une suite de mots, dont le premier mot (qui contient les premiers bits) est défini par son adresse. Avant d'être écrite, la chaîne de bits doit d'abord être remise à zéro, et ensuite un OU logique doit être effectué entre anciens et nouveaux tableaux.



**MSQ1** et **MSQ2** sont utilisés par le coupleur pour définir le nombre de bits à écrire et recalculer la chaîne de bits à écrire sur le premier bit, défini par son adresse Modbus.

#### Variables

Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H'0F'}$   
 Longueur de la table d'émission :  $\text{TXTi,L} = 10 + 2 \times (\text{nombre de mots à écrire})$

---

### 5.2-7 Ecriture de N mots de sortie

Numéro de l'esclave
Adresse Modbus du premier mot à écrire
Nombre de mots à écrire
Valeur du premier mot
Valeur du second mot
Etc ...

Table de réception  
vide

#### Variables

Code requête envoyé au coupleur :  $\text{TXT}_{i,C} = \text{H}'10'$   
Longueur de la table d'émission :  $\text{TXT}_{i,L} = 6 + 2 \times (\text{nombre de mots à écrire})$

---

### 5.2-8 Exemples

- **Lecture des mots d'entrée d'adresse Modbus 8100 et 8101 de l'esclave 32.**

Table d'émission	Table de réception
$W_i = 32$	$W_j = 2$
$W_i + 1 = 8100$	$W_j + 1 = \text{objet esclave d'adresse Modbus 8100}$
$W_i + 2 = 2$	$W_j + 2 = \text{objet esclave d'adresse Modbus 8101}$

- **Ecriture des mots de sortie d'adresse Modbus 4105, 4106 et 4107 de l'esclave 45 avec les valeurs 38, -342 et 2154.**

Table d'émission

$W_i = 45$
$W_i + 1 = 4105$
$W_i + 2 = 3$
$W_i + 3 = 38$
$W_i + 4 = -342$
$W_i + 5 = 2154$

- **Lecture de 18 bits de sortie, dont le premier a pour adresse Modbus 202, de l'esclave 228.**

Table d'émission	Table de réception
$W_i = 228$	$W_j = 2$
$W_i + 1 = 202$	$W_j + 1 = \text{H}'FFFC'$
$W_i + 2 = 18$	$W_j + 2 = \text{bits esclave d'adresse Modbus 202 à 217}$
	$W_j + 3 = \text{bits esclave d'adresse Modbus 218 à 233}$

- **Ecriture de 12 bits de sortie**, dont le premier a pour adresse Modbus 118, de l'esclave 191.

On supposera que les 12 bits à écrire sont les bits W50,E à W51,9 du processeur maître.

Table d'émission

$W_i = 191$   
 $W_i + 1 = 118$   
 $W_i + 2 = 2$   
 $W_i + 3 = H'3FFF' = L'0011111111111111'$   
 $W_i + 4 = H'FC00' = L'1111111000000000'$   
 $W_i + 5 = W50$   
 $W_i + 6 = W51$

---

### 5.3 Echanges coupleur-automate : fonctions complémentaires

---

#### 5.3-1 Lecture du status d'exception

**Table d'émission**

Numéro de l'esclave
---------------------

**Table de réception**

Status d'exception de l'esclave
---------------------------------

#### Variables

Code requête envoyé au coupleur :  $TXT_{i,C} = H'07'$   
 Longueur de la table d'émission :  $TXT_{i,L} = 2$

---

#### 5.3-2 Echo

**Table d'émission**

Numéro de l'esclave
Code diagnostic = 0
Donnée quelconque

**Table de réception**

vide

#### Variables

Code requête envoyé au coupleur :  $TXT_{i,C} = H'08'$   
 Longueur de la table d'émission :  $TXT_{i,L} = 6$

---

### 5.3-3 Reprise communication

**Table d'émission**

Numéro de l'esclave
Code diagnostic = 1
Donnée

**Table de réception**

vide

La donnée peut prendre une des deux valeurs suivantes :

- H'00FF' : poursuite sur erreur  
Dans ce cas, le buffer de trace enregistrant les événements de connexion sera effacé.
- H'0000' : arrêt sur erreur

#### Variables

Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H}'08'$   
Longueur de la table d'émission :  $\text{TXTi,L} = 6$

---

### 5.3-4 Lecture du registre diagnostic

**Table d'émission**

Numéro de l'esclave
Code diagnostic = 2

**Table de réception**

Registre diagnostic de l'esclave

#### Variables

Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H}'08'$   
Longueur de la tache d'émission :  $\text{TXTi,L} = 4$

---

### 5.3-5 Passage en mode écoute

**Table d'émission**

Numéro de l'esclave
Code diagnostic = 4

**Table de réception**

vide

#### Variables

Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H}'08'$   
Longueur de la table d'émission :  $\text{TXTi,L} = 4$

### 5.3-6 Remise à zéro des compteurs

**Table d'émission**

Numéro de l'esclave
Code diagnostic = 10

**Table de réception**

vide

#### Variables

Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H}'08'$   
 Longueur de la table d'émission :  $\text{TXTi,L} = 4$

### 5.3-7 Lecture du nombre de message reçus sur la ligne

**Table d'émission**

Numéro de l'esclave
Code diagnostic = 11

**Table de réception**

Nombre de messages vus sur la ligne
-------------------------------------

#### Variables

Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H}'08'$   
 Longueur de la table d'émission :  $\text{TXTi,L} = 4$

### 5.3-8 Lecture du nombre d'erreurs de type checksum

**Table d'émission**

Numéro de l'esclave
Code diagnostic = 12

**Table de réception**

Nombre d'erreurs de type checksum
-----------------------------------

#### Variables

Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H}'08'$   
 Longueur de la table d'émission :  $\text{TXTi,L} = 4$

---

### 5.3-9 Lecture du nombre de réponse d'exception

#### Table d'émission

Numéro de l'esclave
Code diagnostic = 13

#### Table de réception

Nombre de réponses d'exception
--------------------------------

#### Variables

Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H}'08'$

Longueur de la table d'émission :  $\text{TXTi,L} = 4$

---

### 5.3-10 Lecture du nombre de messages adressés à l'esclave

#### Table d'émission

Numéro de l'esclave
Code diagnostic = 14

#### Table de réception

Nbre de messages adressés à l'esclave
---------------------------------------

#### Variables

Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H}'08'$

Longueur de la table d'émission :  $\text{TXTi,L} = 4$

---

### 5.3-11 Lecture du nombre de non réponse du processeur esclave

#### Table d'émission

Numéro de l'esclave
Code diagnostic = 15

#### Table de réception

Nbre de non réponses du proces. esclave
---

#### Variables

Code requête envoyé au coupleur :  $\text{TXTi,C} = \text{H}'08'$

Longueur de la table d'émission :  $\text{TXTi,L} = 4$



### 5.3-12 Lecture du compteur d'événements

**Table d'émission**

Numéro de l'esclave
---------------------

**Table de réception**

Status
Compteur d'événements

#### Variables

Code requête envoyé au coupleur :

$\text{TXTi,C} = \text{H'0B'}$

Longueur de la table d'émission :

$\text{TXTi,L} = 2$

### 5.3-13 Lecture des événements de connexion

**Table d'émission**

Numéro de l'esclave
---------------------

**Table de réception**

Status
Compteur d'événements
Nombre de message vus sur la ligne
Nombre d'octets du buffer de trace
1er - 2ème - octets de trace
3ème - 4ème octets de trace
Etc ...

#### Variables

Code requête envoyé au coupleur :

$\text{TXTi,C} = \text{H'0C'}$

Longueur de la table d'émission :

$\text{TXTi,L} = 2$

### 5.3-14 Identification de l'esclave

**Table d'émission**

Numéro de l'esclave
---------------------

**Table de réception**

Identificateur de l'esclave
Etat de fonctionnement de l'esclave
Nombre d'octets suivants
1er - 2ème octets
3ème - 4ème octets
Etc ...

#### Variables

Code requête envoyé au coupleur :

$\text{TXTi,C} = \text{H'11'}$

Longueur de la table d'émission :

$\text{TXTi,L} = 2$

---

## 5.4 Requêtes disponibles

---

### Écriture configuration

Code requête	TXTi,C = H'40'
Compte-rendu positif	TXTi,R = H'FE'
Compte-rendu négatif	TXTi,R = H'FD'

Cette requête permet d'envoyer la configuration au coupleur.

Le compte-rendu est négatif lorsque :

- la voie n'est pas dans l'état STOP ou
- les paramètres de la requête sont incorrects.

### Lecture configuration

Code requête	TXTi,C = H'41'
Compte-rendu positif	TXTi,R = H'71'
Compte-rendu négatif	TXTi,R = H'FD'

Cette requête permet de relire la configuration envoyée à la voie.

Le compte-rendu est négatif si la voie n'est pas configurée.

### Lecture compteurs

Code requête	TXTi,C = H'70'
Compte-rendu positif	TXTi,R = H'A0'
Compte-rendu négatif	TXTi,R = H'FD'

Cette requête permet au processeur de lire les compteurs de surveillance des échanges.

Le buffer de réception contient les 12 mots suivants :

- $W_i$  = nombre de demandes processeur prises en compte par le coupleur,
- $W_i + 1$  = nombre de demandes processeur refusées par le coupleur,
- $W_i + 2$  = nombre de réponses esclave reçues correctes,
- $W_i + 3$  = nombre de réponses esclave reçues avec un code d'exception,
- $W_i + 4$  = nombre de messages maître émis sans réponse (délai d'attente écoulé),
- $W_i + 5$  = nombre de réponses esclave reçues avec erreur de checksum,
- $W_i + 6$  = nombre de réponses esclave reçues incohérentes,
- $W_i + 7$  = nombre de messages maître réitérés,
- $W_i + 8$  = nombre d'abandons de réitérations,
- $W_i + 9$  = nombre de délais d'attente de réponse arrivés à terme pendant la réception,
- $W_i + 10$  = nombre de réponses esclave reçues avec défaut en réception,
- $W_i + 11$  = registre diagnostic.

### Modbus transparent

Code requête	TXTi,C = H'81'
Compte-rendu positif	TXTi,R = H'FE'
Compte-rendu négatif	TXTi,R = H'17'

Cette requête permet au processeur de générer une fonction différente des fonctions Modbus gérées par le coupleur.

Le programme utilisateur prépare le message à émettre sur la ligne dans la table d'émission du bloc texte, avec le numéro de l'esclave destinataire mémorisé dans le premier octet.

Le coupleur vérifie le numéro de l'esclave et renvoie un compte-rendu négatif si ce numéro est invalide. Dans le cas contraire, le coupleur calcule le code de redondance et émet le message complet. Si le numéro de l'esclave est nul (diffusion générale), le coupleur émet le message sur la ligne, puis renvoie le compte-rendu positif au processeur sans attendre de réponse sur la ligne.

### Fonctions Modbus

Codes requêtes                      TXTi,C = numéro de la fonction Modbus

Comptes-rendus positifs        TXTi,R = H'FE'  
    TXTi,R = H'19' (demande adressée à un esclave en mode écoute)

### Comptes-rendus négatifs

Un compte-rendu négatif peut provenir de deux origines :

- le coupleur maître, lorsque celui-ci a détecté une erreur dans la demande du processeur ou lorsqu'il n'a pas reçu de réponse de l'esclave.
- le coupleur esclave, lorsque celui-ci a détecté une erreur dans la demande reçue du maître ou lorsque l'esclave a une impossibilité à traiter la demande du maître.

Dans ce cas, l'esclave renvoie au maître une réponse d'exception comprenant :

- l'adresse de l'esclave,
- le code fonction reçu par l'esclave, auquel l'esclave a rajouté la valeur H'80',
- le code d'exception indiquant la nature de l'erreur,
- le contrôle d'échange.

Le coupleur maître transmet alors au processeur le code d'exception reçu de l'esclave comme compte-rendu de la requête.

Compte-rendu (TXTi,R)	Signification
H'01' (réponse d'exception)	Numéro de fonction non défini chez l'esclave
H'02' (réponse d'exception)	Adresse Modbus inconnue de l'esclave
H'03' (réponse d'exception)	Valeur illégale pour l'adresse Modbus indiquée
H'04' (réponse d'exception)	Erreur fatale du processeur esclave
H'05' (réponse d'exception)	Aquittement : le processeur esclave a accepté et est en train de traiter la demande maître
H'06' (réponse d'exception)	Processeur esclave occupé
H'07' (réponse d'exception)	Acquittement négatif
H'15'	Aucune réponse correcte de l'esclave après réitérations
H'17'	Paramètres requêtes incorrects
H'18'	Voie dans l'état STOP
H'FD'	Fonction Modbus inconnue

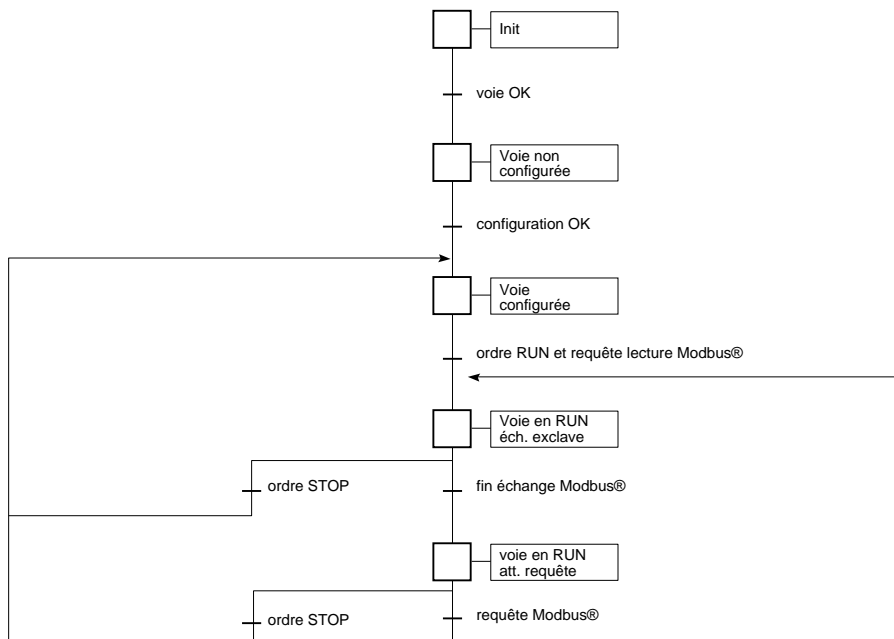
---

## 5.5 Modes de marche

---

### 5.5-1 Graphe général de fonctionnement du coupleur

Le graphe d'état représenté ci-dessous ne décrit que le fonctionnement de la voie 0 du coupleur TSX SCG 11. dont le configuration est le protocole Modbus maître.



La voie passe à l'état RUN sur réception de la première requête Modbus, accompagnée de la demande de mise en RUN de la voie (bit  $OWx,1,1$ ).

Le bit RUN associé à la voie passe alors à l'état RUN (bit  $IWx,1,1$ ).

Le passage de la voie à l'état STOP (sur mise à zéro du bit  $OWx,1,1$ ) est pris en compte à tout moment.

---

### 5.5-2 Comportement du coupleur sur coupures et reprises secteur

Voir chapitre 4.4-2



<b>Sous-chapitre</b>	<b>Page</b>
<b>6.1 Interface tout ou rien</b>	<b>54</b>
6.1-1 Description	54
6.1-2 Coupleur TSX SCG 113	54
6.1-3 Coupleur TSX SCG 116	55
<b>6.2 Interface registre</b>	<b>56</b>
6.2-1 Registres d'entrée	56
6.2-2 Registres de sortie	57

---

## 6.1 Interface tout ou rien

---

### 6.1-1 Description

L'interface tout ou rien du coupleur TSX SCG permet au programme utilisateur de gérer la transmission mise en œuvre sur la voie.

Cette interface est constituée par :

- 4 bits d'entrée : Ix,0 à Ix,3  
Ceux-ci reflètent l'état des signaux d'entrée des adaptateurs de ligne.
- 4 bits de sortie : Ox,0 à Ox,3  
Ceux-là permettent de positionner à l'état vrai ou faux certains signaux de sortie des adaptateurs de ligne.

Chaque bit tout ou rien a une définition spécifique au coupleur. Certains de ces bits ne sont pas uniquement recopiés par le coupleur, mais agissent de façon directe sur le fonctionnement de la voie.

Ci-après sont citées pour chaque coupleur, les affectations des bits tout ou rien, et leur incidence sur le fonctionnement de la voie.

L'état 0 (zéro) logique indique un signal actif.

L'état 1 (un) logique indique un signal non actif.

---

### 6.1-2 Coupleur TSX SCG 113



- **DTR** = connecter le poste de données  
Ce signal à utiliser pour piloter le modem n'a pas d'incidence sur le fonctionnement de la voie.  
La demande pour émettre est gérée par le protocole, donc transparente pour l'utilisateur.
- **DSRS** = sélection de débit binaire  
0 = la voie fonctionne à la vitesse configurée  
1 = la voie fonctionne à la vitesse configurée divisée par 2.

Lorsque le modem utilisé offre cette possibilité, ce signal permet de fonctionner à vitesse réduite lorsque des erreurs de transmissions répétées sont détectées.

---

Bits d'entrée Ix,i

3	2	1	0
RI	CD	DSR	CTS

- **CTS** = prêt à émettre : l'état logique 1 bloque le circuit d'émission,
- **DSR** = poste de données prêt : pas d'incidence sur le fonctionnement de la voie,
- **CD** = détection porteuse : l'état logique 1 bloque le circuit de réception,
- **RI** = indicateur d'appel : pas d'incidence sur le fonctionnement de la voie.

**Attention :**

Lorsque ces signaux ne sont pas utilisés, ils sont tous rappelés dans l'adaptateur à l'état 1 : les circuits d'émission et de réception sont en conséquence bloqués.

---

### 6.1-3 Coupleur TSX SCG 116

Cet adaptateur ne possède pas de signaux de contrôle de la transmission. La validation émetteur utilisée en RS 485 est gérée par le protocole afin de passer l'émetteur en haute impédance, de façon transparente pour l'utilisateur.

---

## 6.2 Interface registre

---

### 6.2-1 Registres d'entrée

#### Registres d'entrée IWx,0 à IWx,7

Ces registres, accessibles uniquement en lecture, fournissent des informations sur le fonctionnement du coupleur.

#### Registre d'entrée IWx,0

Bit	Fonction	Explications
0 1	non affectés	
2	RAZ en cours	Remise à zéro du système de message en cours.
3	Module disponible	Indique la fin des autos-tests communs et spécifiques donc la disponibilité du module
4	Défaut général	Ce bit passe à l'état 1 sur présence d'un défaut quelle que soit son origine, il réalise la fonction OU des bits 7 et 8 de ce même registre
5	non affecté	
6	non affecté	
7	Défaut application	Défaut lors de l'exécution d'une requête (erreur de parité en réception).
8	Défaut bloquant	Absence de 12V ou asynchronisme entre les alimentations 12V du module et 5V de l'automate. Vérifier que ces alimentations apparaissent en même temps, puis remettre ce bit à 0 par SY0 ou SY1. Défaut RAM, REPR0M ou logique interne du coupleur. Ce défaut provoque un blocage permanent du coupleur et nécessite son remplacement. Le voyant RUN est éteint et le voyant I/O est allumé.
9	Module en auto-test	Lors de chaque mise sous-tension de l'automate, le coupleur déroule une séquence d'auto-tests signalée par ce bit. Durant cette phase le coupleur non-disponible ne peut pas être configuré ou exploité.
A	non affecté	
B	Module non configuré	Ce bit est à l'état 1 lorsque le module n'est pas configuré
C	Module en RUN	Une requête est en cours d'exécution
D	réservé	
E	non affectés	
F		



**Registre d'entrée IWx,1**

Ce mot contient des informations sur l'état de la voie du coupleur

Bit	Fonction	Explications
0	non affecté	
1	RUN/STOP	0 si la voie est en STOP. 1 si la voie est en RUN
2	LOM	0 si la voie Modbus Esclave est en réception/émission 1 si la voie Modbus Esclave est en écoute seule
3	Coupleur configuré	0 si le coupleur n'est pas configuré 1 si le coupleur est configuré
4	Code adaptateur	101 = adaptateur RS 232 - Modem
5		100 = adaptateur RS 485
6		
7à E	non affectés	
F	Repli PWF	A l'état 1, il indique une reprise secteur Sa remise à zéro (par OWx,1,F) est nécessaire pour la prise en compte d'une nouvelle reprise secteur

**Registres d'entrée IWx,2 à IWx,7**  
inutilisés

**6.2-2 Registres de sorties****Registre de sortie OWx,0**

Le bit OWx,0,2 permet d'effectuer la remise à zéro du système de messagerie du module.

**Registre de sortie OWx,1**

Ce mot contient les informations de commande RUN/STOP

Bit	Fonction	Explications
0	non affecté	
1	RUN/STOP	0 si la voie est en STOP. 1 si la voie est en RUN
2à E	non affectés	
F	Acquittement PWF	La mise à 1 de ce bit permet d'acquitter (non obligatoire) le repli après une coupure secteur sur l'automate (repositionne IWx,1,F à l'état 0).

**Attention**

La remise à 0 des bits d'acquiescement ou de RAZ est à la charge du programme utilisateur.

---

**Registre de sortie OWx,2**

inutilisé

**Registre de sortie OWx,3 (esclave uniquement)**

Les poids forts correspondent à un registre d'état (voir chapitre 2.7 Lecture du status d'exception).

**Registres de sortie OWx,4 à OWx,7**

inutilisés.



<b>Sous-chapitre</b>	<b>Page</b>
<b>7.1 Signalisation des bits défauts</b>	<b>60</b>

---

---

## 7.1 Signalisation des bits défauts

---

La signalisation des défauts correspond à une série de bits accessibles en lecture en mode message, par la requête standard à tous les coupleurs intelligents "Lecture des bits défauts". Ces bits défauts au nombre de 48 (3 mots de 16 bits), seront reçus dans la table de réception du bloc texte concerné.

- code requête  $\text{TXTi,C} = \text{H}'47'$
- variable  $\text{TXTi,L} = 0$
- table émission non utilisée,
- table réception au moins 6 octets
- adresse coupleur et numéro de voie  $\text{TXTi,M} = 0x63$  avec  $x = 1, 2$  ou  $3$

### Etat des variables en fin d'échange

- nombre d'octets reçus  $\text{TXTi,S} = 6$
- compte-rendu  $\text{TXTi,R} = \text{H}'77'$

### Organisation de la table

- Bits défauts BDEF

Bit défaut	Défaut
0	Défaut de checksum EPROM
1	défaut RAM (interne, externe)
2	défaut de 12V
3 à 36	non utilisés
37	défaut de parité en réception
38 à 47	non utilisés



<b>Sous-chapitre</b>	<b>Page</b>
<b>8.1 Programmation de l'esclave sur TSX 17-20</b>	<b>62</b>
8.1-1 Description	62
8.1-2 Organisation	63
8.1-3 Définition des objets et des constantes	64
8.1-4 Programme, listing	65

---

## 8.1 Programmation de l'esclave sur TSX 17-20

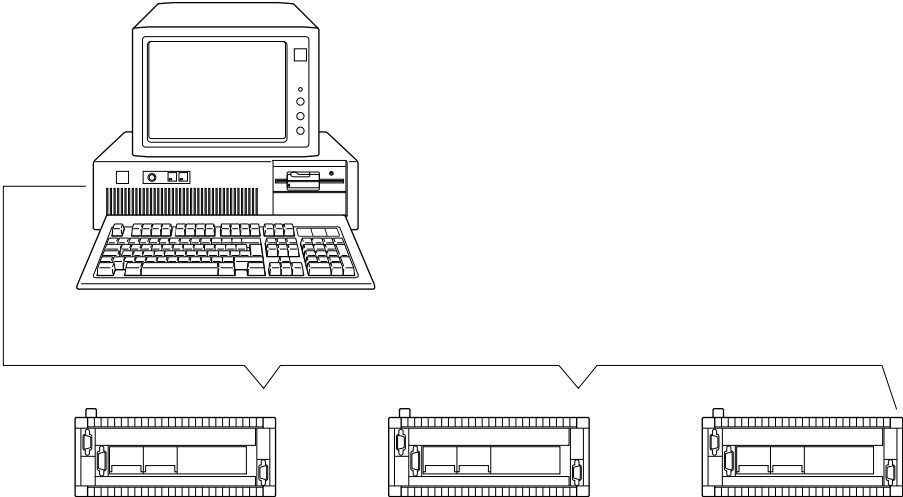
---

### 8.1-1 Description

Prenons l'exemple d'un automate TSX 17-20 raccordé à un maître sous le protocole Modbus.

L'esclave concerné a pour numéro de station 1.

Le programme décrit dans cet exemple assure la configuration du module après une mise sous tension de l'automate, puis la gestion du protocole Modbus esclave.



Nous allons définir dans cet esclave, pour une application donnée, 4 zones de bits et mots (objets accessibles). L'utilisateur aura à sa charge l'éventuelle définition d'objets physiques (entrées, sorties tout ou rien...) ainsi que le programme application liant ces objets physiques aux objets accessibles.

Soit une zone de 60 mots localisés à partir de W60, que nous allons définir de la façon suivante :

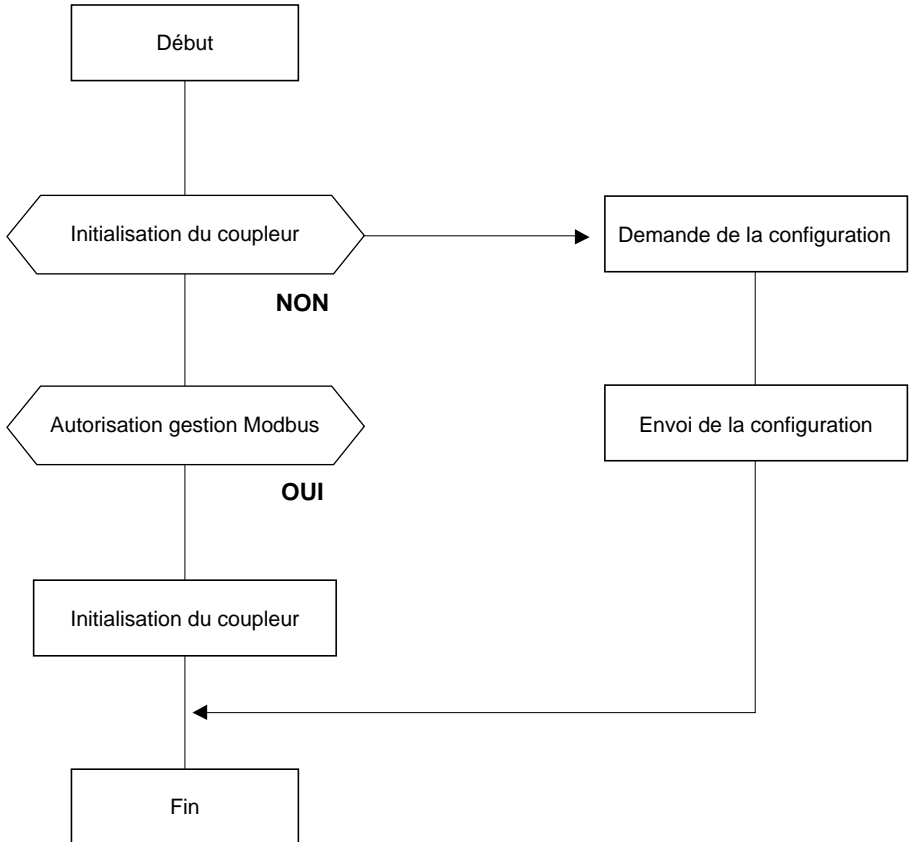
<b>W75 à W89</b>	table de 240 bits d'entrée
<b>W105 à W119</b>	table de 15 mots d'entrée
<b>W60 à W74</b>	table de 240 bits de sortie
<b>W90 à W104</b>	table de 15 mots de sortie

A chaque zone correspond une adresse Modbus maître (valeurs définies pour un maître donné).

bits d'entrée	<b>W75</b>	a pour adresse Modbus correspondante 0
mots d'entrée	<b>W105</b>	a pour adresse Modbus correspondante 0
bits de sortie	<b>W60</b>	a pour adresse Modbus correspondante 0
mots de sortie	<b>W90</b>	a pour adresse Modbus correspondante 0

### 8.1-2 Organisation

Le programme principal gère l'envoi de la configuration sur la voie 0 du coupleur et autorise ensuite la gestion du protocole Modbus sur cette voie.



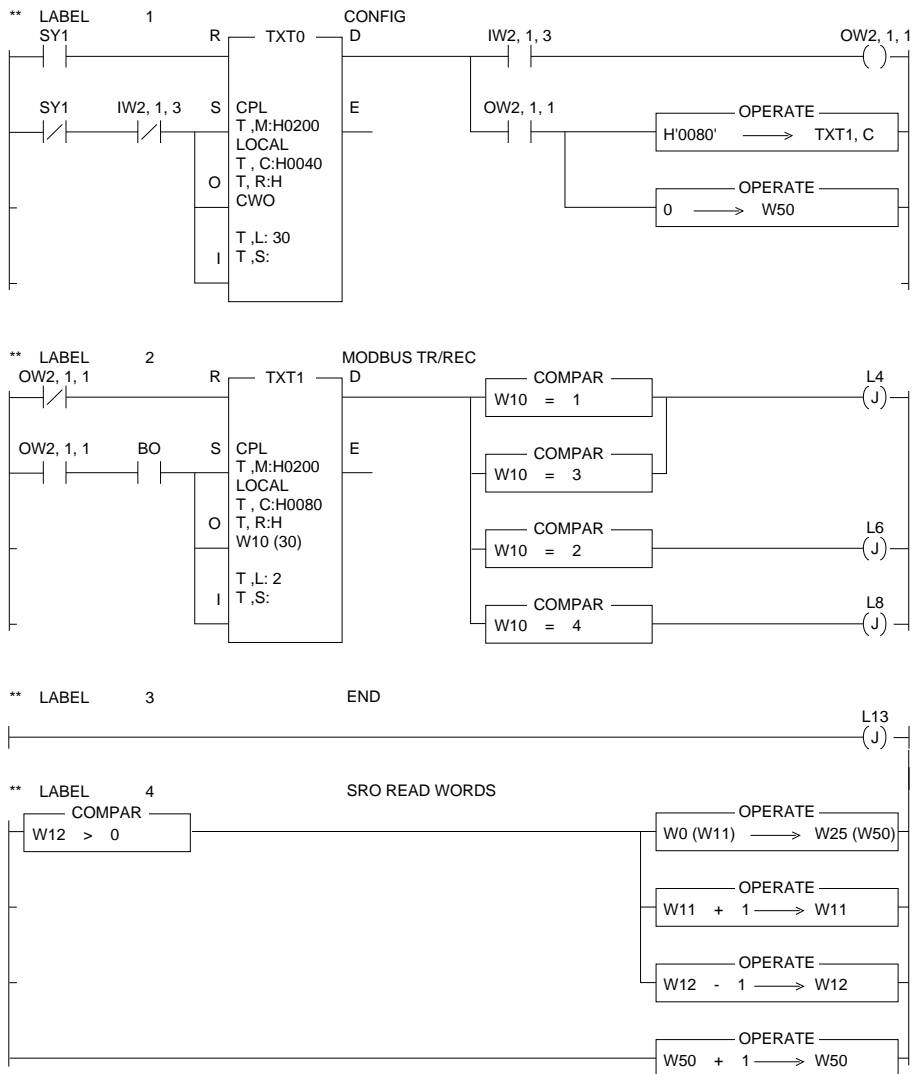
---

### 8.1-3 Définition des objets et des constants

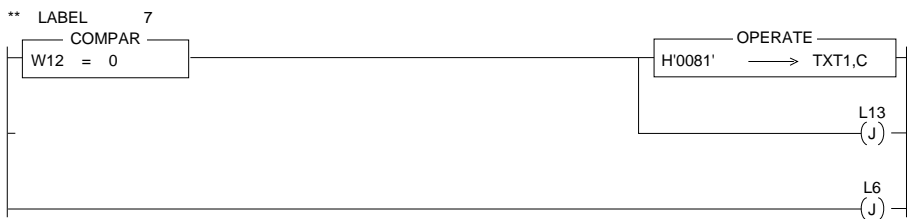
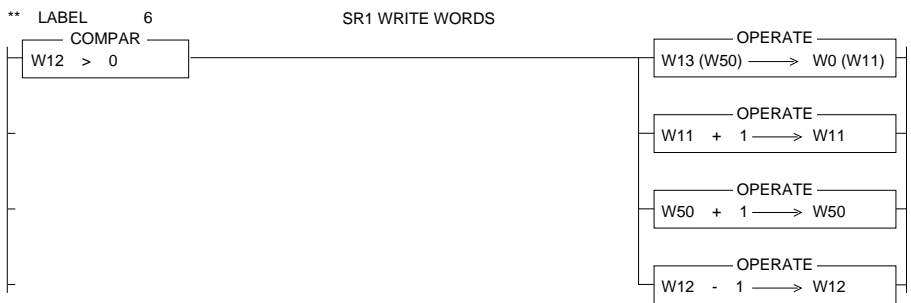
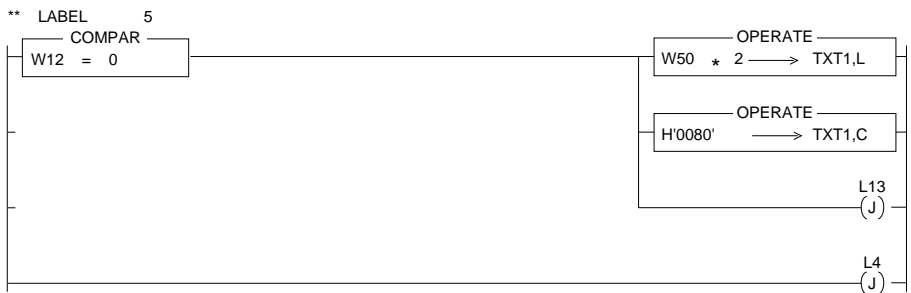
<b>TXT0</b>	:	configuration de la voie 0
<b>TXT1</b>	:	transfert des messages Modbus sur la voie 0
<b>B0</b>	:	autorisation de gestion du protocole Modbus
<b>CW0</b>	= H'2802'	: Modbus, mode RTU, pas de parité, 2 bits de stop
<b>CW1</b>	= H'9600'	: vitesse 9600 bauds
<b>CW2</b>	= 1	: station esclave numéro 1
<b>CW3</b>	= 60	: zone bits de sortie
<b>CW4</b>	= 240	: 240 bits de W60 à W74
<b>CW5</b>	= 0	: adresse Modbus du premier bit de sortie : 0
<b>CW6</b>	= 75	: zone bits d'entrée
<b>CW7</b>	= 240	: 240 bits de W75 à W89
<b>CW8</b>	= 0	: adresse Modbus du premier bit d'entrée : 0
<b>CW9</b>	= 90	: zone mots de sortie
<b>CW10</b>	= 15	: 15 mots de W90 à W104
<b>CW11</b>	= 0	: adresse Modbus du premier mot de sortie : 0
<b>CW12</b>	= 105	: zone mots d'entrée
<b>CW13</b>	= 15	: 15 mots de W105 à W119
<b>CW14</b>	= 0	: adresse Modbus du premier mot d'entrée : 0



## 8.1-4 Programme, listing



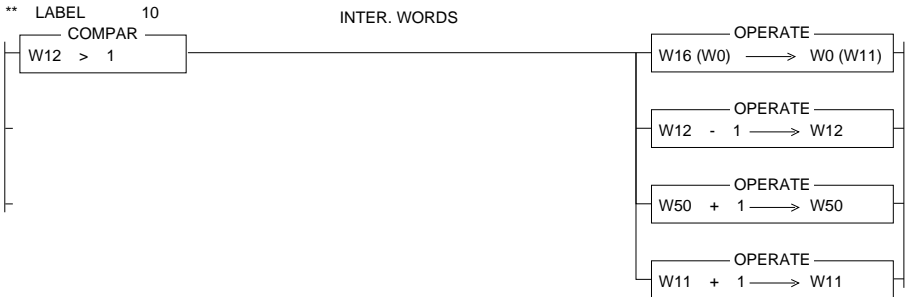
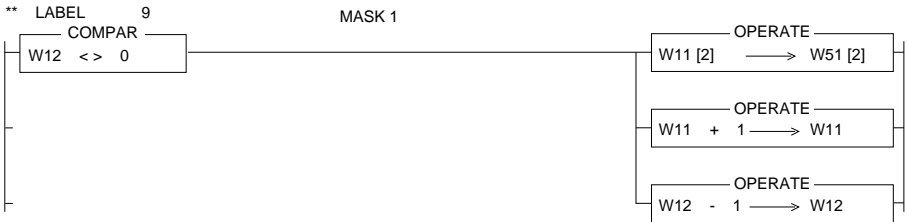
application MODBUS / ..... Télémechanique-/TSX	PROG.: MAST	rev	date 11-15-90	page 7- 1 7
--	-------------	-----	------------------	-------------------



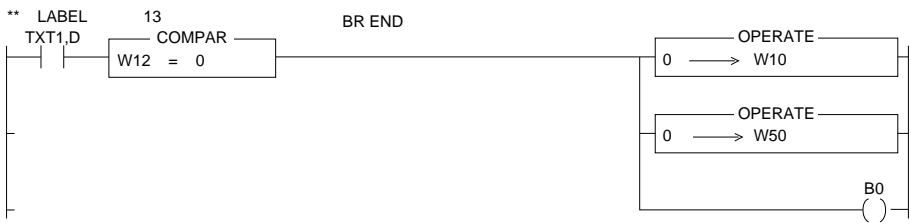
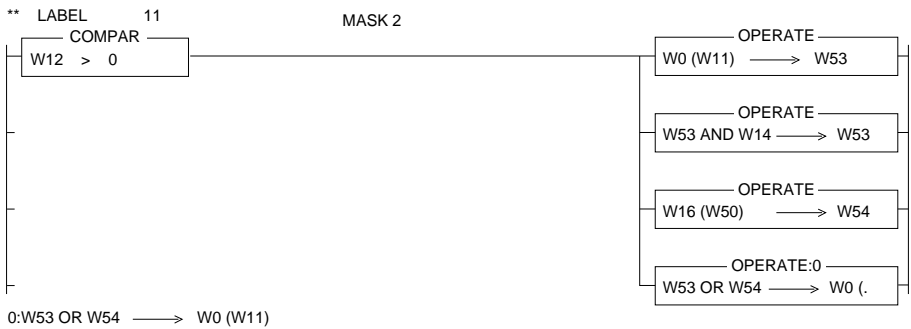
application	rev	date	page
MODBUS / ..... Télemécanique/-TSX	PROG.: MAST	11-15-90	7- 2 8



0:W53 OR W15 → W0 (W11)



application MODBUS / ..... Télémeccanique-/TSX	PROG.: MAST	rev	date 11-15-90	page 7- 3 9
--	-------------	-----	------------------	-------------------



END OF PROGRAM

application MODBUS / ..... Télémechanique-/TSX	PROG.: MAST	rev	date 11-15-90	page 7- 4 10
--	-------------	-----	------------------	--------------------



<b>Sous-chapitre</b>	<b>Page</b>
<b>9.1 Performances de l'esclave</b>	70
<b>9.2 Performances du maître</b>	71
<b>9.3 Fonctions Jbus</b>	73
<b>9.4 Trames Modbus (RTU)</b>	74
9.4-1 Détails des trames	74
9.4-2 Réponses d'exception	78
9.4-3 Calcul du CRC 16	79

---

## 9.1 Performances de l'esclave

---

### Approche des temps de réponse

Le calcul des temps de réponse est nécessaire pour évaluer les performances globales des échanges, et pour configurer le maître (définition du temps enveloppe pour interroger l'esclave suivant dans le cas d'une absence de réponse).

Tous les temps donnés sont valables dans les cas les plus défavorables.

Le temps de transfert se décompose comme suit :

- temps de transmission d'un message à l'esclave  $Trec$ ,
- le temps de traitement du coupleur  $Tdem$ ,
- temps de prise en compte et réponse du processeur  $Tpro$ ,
- temps de traitement de la réponse automate  $Trep$ ,
- temps de transmission du message au maître  $Témi$ .

Temps de transmission d'un message à l'esclave (N caractères)

$Trec = 1000 \times N \times (\text{nombre de bits par caractère}) / (\text{vitesse en bauds})$ .

Temps de traitement du coupleur

$Tdem$  est le temps nécessaire au coupleur pour reconnaître la fin du message (délai minimal de 3,5 caractères entre deux messages), pour l'analyser et émettre une requête vers le processeur.

Ce temps est fonction de la demande du maître.

Temps de prise en compte et réponse du processeur

$Tpro = 6 \times (\text{temps de cycle automate})$  lors d'échanges par bloc texte.

Temps de traitement de la réponse automate

$Trep$  est le temps utilisé par le coupleur pour prendre compte la réponse du processeur et préparer le message à renvoyer au maître.

Ce temps est fonction de la demande du maître.

Temps de transmission du message au maître (M caractères)

$Témi = 1000 \times M \times (\text{nombre de bits par caractère}) / (\text{vitesse en bauds})$ .

### Exemple

Soit une liaison à 9600 bauds, dont les caractères sont au format de un bit de start, huit bits de données, un bit de parité paire, un bit de stop, soit onze bits par caractère. Le maître effectue une demande de lecture de 32 bits.

Les échanges se font par l'interface message. Le temps de cycle automate est de 80 millisecondes.

$Trec = 1000 \times 8 \times 11 / 9600 = 9,2 \text{ ms}$

$Tdem = (1000 \times 3,5 \times 11 / 9600 + 4,5) = 8,5 \text{ ms}$

$Tpro = 6 \times 80 = 480 \text{ ms}$

$Trep = 1 \text{ ms}$

$Témi = 1000 \times 8 \times 11 / 9600 = 9,2 \text{ ms}$

Temps total = 508 ms

Le temps enveloppe du maître pourrait alors être fixé à 520 millisecondes.

---

## 9.2 Performances du maître

---

### Approche des temps de réponse

L'analyse qui suit permet d'évaluer le temps de réponse d'une transaction de Modbus. Tous les temps donnés sont valables dans les cas les plus défavorables.

Le temps de transfert se décompose comme suit :

- temps de prise en compte du processeur  $T_{pro1}$ ,
- temps de traitement du coupleur  $T_{dem}$ ,
- temps de transmission d'un message à un esclave  $T_{émi}$ ,
- temps de traitement de l'esclave  $T_{esc}$ ,
- temps de transmission de la réponse au maître  $T_{rec}$ ,
- temps de traitement de la réponse de l'esclave  $T_{rep}$ ,
- temps de réponse du processeur  $T_{pro2}$ .

Temps de prise en compte du processeur

$T_{pro1} = 2 \times$  (temps de cycle automate)

Temps de traitement du coupleur

$T_{dem}$  est le temps nécessaire au coupleur pour prendre en compte la demande du processeur et préparer le message à émettre à l'esclave. Ce temps dépend de la demande (fonction Modbus, nombre d'objets à traiter, etc...).

Temps de transmission d'un message à un esclave (N caractères)

$T_{émi} = 1000 \times N \times$  (nombre de bits par caractère) / (vitesse en bauds).

Temps de traitement de l'esclave

$T_{esc}$  constitue le temps de réponse de l'esclave interrogé, hors temps de transmission.

Temps de transmission de la réponse au maître (M caractères)

$T_{rec} = 1000 \times M \times$  (nombre de bits par caractère) / (vitesse en bauds).

Temps de traitement de la réponse de l'esclave

$T_{rep}$  est le temps utilisé par le coupleur pour analyser la réponse Modbus reçue de l'esclave et mettre en forme la réponse à transmettre au processeur. Ce temps dépend de la demande (fonction Modbus, nombre d'objets à traiter, etc...).

Temps de réponse du processeur

$T_{pro2} = 2 \times$  (temps de cycle automate).

---

## Exemples

Soit une liaison à 9600 bauds, dont les caractères sont au format de un bit de start, huit bits de données, un bit de parité paire, un bit de stop, soit onze bits par caractère. Le temps de cycle automate est de 60 millisecondes.

### • Ecriture de 16 mots

$T_{pro1}$		= 120 ms
$T_{dem}$		= 8,3 ms
$T_{émi}$	= $1000 \times 41 \times 11 / 9600$	= 47 ms
$T_{esc}$	temps fonction de l'esclave considéré	

$T_{rec}$	= $1000 \times 8 \times 11 / 9600$	= 9,2 ms
$T_{rep}$		= 2,1 ms
$T_{pro2}$		= 120 ms

Temps total : 307 ms + (temps de traitement de l'esclave)

### • Lecture de 256 bits

$T_{pro1}$		= 120 ms
$T_{dem}$		= 1,8 ms
$T_{émi}$	= $1000 \times 8 \times 11 / 9600$	= 9,2 ms
$T_{esc}$	temps fonction de l'esclave considéré	
$T_{rec}$	= $1000 \times 37 \times 11 / 9600$	= 42,4 ms
$T_{rep}$		= 8,6 ms
$T_{pro2}$		= 120 ms

Temps total : 302 ms + (temps de traitement de l'esclave)



---

### 9.3 Fonctions Jbus

---

Certaines fonctions utilisées par Jbus sont communes à celles utilisées par Modbus, d'autres présentent des différences. Ces fonctions sont listées ci-dessous (leur détail est donné au chapitre 2 "Gestion du protocole") :

#### Fonctions identiques avec les produits Modbus (maître / esclave) :

- 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 : écriture de 1 bit de sortie,
- 06 : écriture de 1 mot de sortie,
- 08/00 : fonction diagnostic,
- 08/01 : réinitialisation de voie,
- 15 : écriture de n bits,
- 16 : écriture de n mots.

#### Fonctions communes mais les résultats obtenus sont différents :

- 07 : pour Modbus, lecture de OWxy,6 d'un coupleur TSX SCM esclave,  
lecture de OWxy,3 d'un coupleur TSX SCG esclave,  
: pour Jbus, lecture d'une adresse spécifique à l'équipement.

Note :

Pour les fonctions suivantes, il est impératif que l'utilisateur dispose des spécifications des matériels utilisés (les compteurs de diagnostic sont spécifiques aux matériels).

- 08/02
- 08/03
- 08/0B
- 08/0C
- 08/0D
- 08/0E
- 08/0F
- 11
- 12

#### Fonctions Jbus non implémentées sur Modbus

- 08/11 : lecture du compteur 6,
- 08/12 : lecture du compteur 8.

## 9.4 Trames Modbus (RTU)

### 9.4-1 Détails des trames

#### Lecture de N bits : fonction 1 ou 2

Question

N° esclave	1 ou 2	N° du 1er bit PF   Pf	Nombre de bits PF   Pf	CRC 16
1 octet	1 octet	2 octets	2 octets	2 octets

Réponse

N° esclave	1 ou 2	Nombre d'octets lus	Valeur	.....	Valeur	CRC 16
1 octet	1 octet	1 octet				2 octets

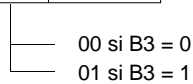
Exemple : lecture du bit B3 de l'esclave 2

Question

02	01	0003	0001	CRC 16
----	----	------	------	--------

Réponse

02	01	01	xx	CRC 16
----	----	----	----	--------



#### Lecture de N mots : fonction 3 ou 4

Question

N° esclave	3 ou 4	N° du 1er mot PF   Pf	Nombre de mots PF   Pf	CRC 16
1 octet	1 octet	2 octets	2 octets	2 octets

Réponse

N° esclave	3 ou 4	Nombre d'octets lus	Valeur du 1er mot PF   Pf	.....	Valeur dernier mot PF   Pf	CRC 16
1 octet	1 octet	1 octet	2 octets		2 octets	2 octets

Exemple : lecture des mots W20 à W24 de l'esclave 6

Question

06	04	14	05	CRC 16
----	----	----	----	--------

Réponse

02	01	0A	xxxx	.....	xxxx	CRC 16
			Valeur de W20		Valeur de W24	

**Ecriture d'un bit de sortie : fonction 5**

Question

N° esclave	5	N° du bit PF   Pf	Valeur du bit	CRC 16
1 octet	1 octet	2 octets	2 octets	2 octets

Le champ "Valeur du bit" a deux valeurs possibles à l'exclusion de toute autre :

- bit à 0 = 0000,
- bit à 1 = FF00.

Réponse

N° esclave	5	N° du bit PF   Pf	Valeur du bit	CRC 16
1 octet	1 octet	2 octets	2 octets	2 octets

Exemple : écriture de la valeur 1 dans le bit B3 de l'esclave 2

Question 

02	05	03	FF00	CRC 16
----	----	----	------	--------

Réponse 

02	05	03	FF00	CRC 16
----	----	----	------	--------

**Ecriture d'un mot de sortie : fonction 6**

Question

N° esclave	6	N° du mot PF   Pf	Valeur du mot PF   Pf	CRC 16
1 octet	1 octet	2 octets	2 octets	2 octets

Réponse

N° esclave	6	N° du mot PF   Pf	Valeur du mot PF   Pf	CRC 16
1 octet	1 octet	2 octets	2 octets	2 octets

Exemple : écriture de la valeur 3A15 dans le mot W12 de l'esclave 5

Question 

05	06	0C	3A15	CRC 16
----	----	----	------	--------

Réponse 

05	06	0C	3A15	CRC 16
----	----	----	------	--------

---

**Lecture du status d'exception** : fonction 7

Question

N° esclave	7	CRC 16
1 octet	1 octet	2 octets

Réponse

N° esclave	7	Status exception	CRC 16
1 octet	1 octet	1 octet	2 octets

**Diagnostic** : fonction 8

Question et réponse

N° esclave	8	Sous-code	Données	CRC 16
1 octet	1 octet	2 octets	2 octets	2 octets

**Lecture du compteur d'événements** : fonction 11 (H'0B')

Question

N° esclave	0B	CRC 16
1 octet	1 octet	2 octets

Réponse

N° esclave	0B	00   00	Valeur compteur PF   Pf	CRC 16
1 octet	1 octet	2 octets	2 octets	2 octets

**Lecture des événements de connexion : fonction 12 (H'0C')**

Question

N° esclave	0C	CRC 16
1 octet	1 octet	2 octets

Réponse

N° esclave	0C	Status	Compteur d'événements	Nb messages vus sur la ligne	Nombre d'octets buffer de trace
1 octet	1 octet	2 octets	2 octets	2 octets	2 octets

Trace 2 et 1	Trace 4 et 3	Trace n et n-1	CRC 16
2 octets	2 octets	2 octets	2 octets

**Ecriture de N bits de sortie : fonction 15 (H'0F')**

Question

N° esclave	0F	Adresse 1er bit à forcer	Nombre de bits à forcer	Nombre d'octets	Valeur des bits à forcer	CRC 16
1 octet	1 octet	2 octets	2 octets	1 octet	n octets	2 octets

Réponse

N° esclave	0F	Adresse 1er bit à forcer	Nombre de bits à forcer	CRC 16
1 octet	1 octet	2 octets	2 octets	2 octets

---

## Écriture de N mots de sortie : fonction 16 (H'10')

Question

N° esclave	10	Numéro du 1er mot	Nombre de mots	Nombre d'octets	Valeur du 1er mot	.....	CRC 16
1 octet	1 octet	2 octets	2 octets	1 octet	2 octets		2 octets

Réponse

N° esclave	10	Numéro du 1er mot	Nombre de mots	CRC 16
1 octet	1 octet	2 octets	2 octets	2 octets

Exemple : écriture des valeurs 1 et 2 dans les mots W16 et W17 de l'esclave 11

Question

0B	10	0010	0002	04	0001	0002	CRC 16
----	----	------	------	----	------	------	--------

Réponse

0B	10	0010	0002	CRC 16
----	----	------	------	--------

---

### 9.4-2 Réponses d'exception

Une réponse d'exception est retournée par un esclave lorsque celui-ci ne peut exécuter la demande qui lui est adressée.

**Format d'une réponse d'exception :**

N° esclave	Code réponse	Code erreur	CRC 16
1 octet	1 octet	1 octet	1 octet

Code réponse : C'est le code fonction de la demande + H'80' (le bit de rang le plus élevé est mis à 1),

Code erreur : Pour plus de détails concernant les codes d'erreur des coupleurs TSX SCG 1131 / 1161, se reporter au chapitre 5.4 de ce document.

Pour plus de détails concernant les codes d'erreur des coupleurs TSX SCM 22, se reporter au chapitre 4.4 du document TSX D24 002F.

---

### 9.4-3 Calcul du CRC 16

Le CRC 16 se calcule sur tous les octets du message en appliquant la méthode suivante :

- initialiser le CRC (registre de 16 bits ) à H'FFFF',
- faire du premier octet du message au dernier :
  - CRC XOR <octet> --> CRC
  - Faire 8 fois
    - Décaler le CRC d'un bit à droite
    - Si le bit sorti = 1, faire CRC XOR H'A001' --> CRC
  - Fin faire
- Fin faire

Le CRC obtenu sera émis poids faible d'abord, poids fort ensuite.

XOR = OU exclusif

