Table des matières

MODBUS

Introduction

Modbus (marque déposée par Modicon) est un protocole de communication utilisé pour des réseaux d'automates programmables (API). Il fonctionne sur le mode maître / esclave(s). Il est constitué de trames contenant l'adresse de l'automate concerné, la fonction à traiter (écriture, lecture), la donnée et le code de vérification d'erreur appelé contrôle de redondance cyclique sur 16 bits ou CRC16.

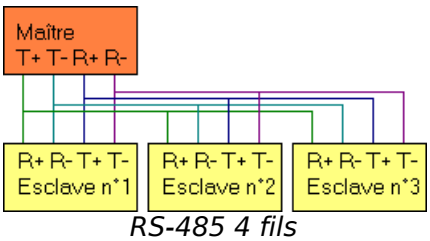
Les trames sont de 2 types :

- mode RTU (Remote Terminal Unit) : les données sont sur 8 bits
- mode ASCII : les données sont codées en ASCII (il faut deux caractères pour représenter un octet, exemple 0x03 sera codé '0' et '3')

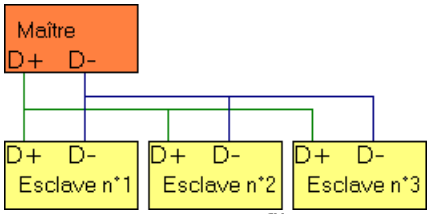
Le protocole Modbus peut être implémenté :

- sur une liaison série asynchrone de type RS-422 ou RS-485 ou TTY (boucle de courant), avec des débits et sur des distances variables ;
- sur TCP/IP sur Ethernet ; on parle alors de Modbus TCP/IP ;
- sur Modbus Plus. Modbus Plus est un réseau à passage de jetons à 1 Mb/s, pouvant transporter les trames Modbus et d'autres services propre à ce réseau.

Exemple : une liaison multipoints de type RS-485 relie un seul maître et un ou plusieurs esclave(s) sur une paire différentielle avec un débit jusqu'à 10 Mbits/s sur une distance d'environ 1 200 m. Sur 2 fils, la transmission est en half duplex (les données passent alternativement dans un sens puis dans l'autre).



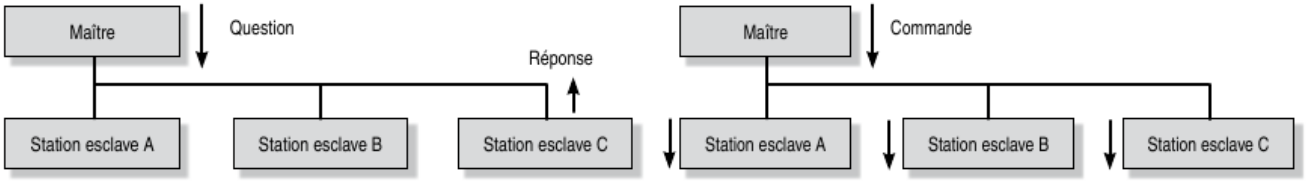
RS-485 4 fils



RS-485 2 fils

Échange entre un maître et un esclave

Le maître envoie une question et attend une réponse de l'esclave.



La structure des messages (question/réponse) est la suivante :

La question

Elle contient un code fonction indiquant à l'esclave adressé le type d'action demandé.
Les données contiennent des informations complémentaires dont l'esclave a besoin pour exécuter cette fonction.
Le mot de contrôle permet à l'esclave de s'assurer de l'intégralité du contenu de la question.

La réponse

Si une erreur apparaît, le code fonction est modifié pour indiquer que la réponse est une réponse d'exception (MSB*=0 : pas d'erreur ; MSB=1 : erreur).
Les données contiennent alors un code (code d'exception) permettant de connaître le type d'erreur.
Code d'exception :

01	Fonction illégale (erreur sur le code fonction)
02	Erreur sur l'adresse du registre ou du coil
08	Erreur de transmission (suite au contrôle du CRC ou du Timing)

*MSB : Most Significant Bit

Question :

N° station esclave	Code fonction + bit d'erreur	Information spécifique concernant la demande	Mot de contrôle
1 octet	1 octet	n octets	2 octets

Réponse :

N° station esclave	Code fonction + bit d'erreur	Données transmises	Mot de contrôle
1 octet	1 octet	n octets	2 octets

Réponse lors d'une erreur :

N° station esclave	Code fonction + bit d'erreur	Code d'exception	Mot de contrôle
1 octet	1 octet	1 octet	2 octets

MODBUS offre 19 fonctions différentes. Les équipements ne supportent pas obligatoirement toutes ces fonctions.

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

Question du maître

Remarque : le "mot" représente ici 2 octets soit 16 bits.

Lecture de n mots : fonction 3 ou 4

■ Demande.

N° esclave	3 ou 4	Adresse du 1er mot à lire : PF * | pf *	Nombre de mots à lire n ⩽ 125 : PF * | pf *	CRC 16 pf * | PF *
1 octet	1 octet	2 octets	2 octets	2 octets

■ Réponse.

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

Contenu d'une réponse exception.

N° esclave (1 à FF)	1		CRC 16 PF * | pf *
1 octet	1 octet	1 octet	2 octets

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

Code fonction reçu et bit de poids fort à 1.

Le message émis par le maître est le suivant :

04	03	00	02	00	01	25	9F

Exercice n°1

Décoder le message émis par le maître en complétant le tableau suivant.
Réponse :

Champs	**Valeur**	**Signification/Décodage**
Adresse de l'esclave		
Code fonction		
Mot de contrôle (CRC)		

Réponse(s) de l'esclave

L'esclave peut émettre deux types de réponse :

Réponse n°1 :

04	03	02	02	58	74	DE

Réponse n°2 :

04	83	02	D0	F0

Exercice n°2

Décoder le message « réponse n°1 » émis par l'esclave en complétant le tableau suivant.
Réponse :

Champs	**Valeur**	**Signification/Décodage**
Adresse de l'esclave		
Code fonction		
Mot de contrôle (CRC)		

Exercice n°3

Décoder le message « réponse n°2 » émis par l'esclave en complétant le tableau suivant.
Réponse :

Champs	**Valeur**	**Signification/Décodage**
Adresse de l'esclave		
Code fonction		
Mot de contrôle (CRC)		

Exercice n°4

D'après les messages transférés entre le maître et l'eclave, en déduire le type de trame (RTU ou ASCII) utilisé ici.
Réponse :

Étude de cas

Le pressostat TEDM et le thermostat ETTNM possèdent un port série RS485 et utilisent le protocole de communication Modbus RTU.

Les TEDM et ETTNM utilise 4 codes fonctions.

Code	Fonction	Action
01	Read Coils Status	Lecture de l'état des seuils 1 et 2 : ouvert ou fermé Lecture de la configuration des seuils : Normalement Ouvert (NO) ou Normalement Fermé (NC) Lecture de la position du point décimal (afficheur)
03	Read Holding Register	Lecture de la valeur mesurée. Pression (TEDM) ou Température (ETTNM) Lecture du code d'accès Lecture de la valeur des points de commutation haut et bas de chaque seuil Lecture des valeurs de temporisation de chaque seuil
05	Write Single Coil	Ecriture de la configuration des seuils : NO ou NC
06	Write Single Register	Ecriture du code d'accès Ecriture de la valeur des points de commutation haut et bas de chaque seuil Ecriture des valeurs de temporisation de chaque seuil Ecriture de l'adresse de l'esclave (TEDM ou ETTNM)

Ces codes fonctions permettent de récupérer :

- la valeur de mesurée de pression (TEDM) ou de température (ETTNM)
- l'état de chaque seuil (ouvert ou fermé)

et d'écrire l'état et le réglage des seuils, le code d'accès et l'adresse de l'esclave

Le maître veut interroger le TEDM/ETTNM sur la valeur de la mesure, du code, du réglage des points de commutation haut ou bas des temporisations des seuils. La question envoyée par le maître doit spécifier l'adresse du registre à lire. Les registres sont adressés à partir de zéro : les registres 1-11 sont adressés de 0 à 10.

Exercice n°5

Donner la requête émise par le maître pour lire le registre 1 (valeur mesure) de l'esclave 59.

Réponse :

	Nom du champ	Exemple
Question	Slave Address	
	Function	
	Starting Address Hi	
	Starting Address Lo	
	No. of Points Hi	
	No. of Points Lo	
	Error Check (CRC)	—

Adr. Reg	Action
00	Valeur mesure
01	Code
02	Valeur HSP1
03	Valeur LSP1
04	Valeur HSP2
05	Valeur LSP2
06	
07	Valeur TS1
08	Valeur TH1
09	Valeur TS2
10	Valeur TH2

L'esclave répond :

Exemple de réponse à la requête :

	Nom du champ	Exemple
Réponse	Slave Address	3B hex
	Function	03 hex
	Byte Count	02 hex
	Data Hi (Register 1)	0A hex
	Data Lo (Register 1)	2B hex
	Error Check (CRC)	—

Adr. Reg	Action
00	Valeur mesure
01	Code
02	Valeur HSP1
03	Valeur LSP1
04	Valeur HSP2
05	Valeur LSP2
06	
07	Valeur TS1
08	Valeur TH1
09	Valeur TS2
10	Valeur TH2

Exercice n°6

En décodant la réponse de l'esclave donner (en décimal) la valeur de la mesure.
Réponse :

Le code fonction 05 (*Write Single Coil*) est utilisé pour configurer à distance les seuils en NO ou en NC. Les coils concernés sont le 05 (NO) et le 06 (NC) avec en adresse respective adr 04 et adr 05.

Pour configurer le seuil en NO, la donnée 0xFF00 est envoyée à l'esclave.
Pour configurer le seuil en NC, la donnée 0x0000 est envoyée à l'esclave.

Exercice n°7

En sachant que l'esclave confirme l'écriture en renvoyant le même message que celui que lui a envoyé le maître, en déduire la signification de la requête envoyée par le maître.
Réponse :

L'esclave confime l'écriture en renvoyant le même message.

Exemple de réponse à la requête :

	Nom du champ	Exemple
Réponse	Slave Address	3B hex
	Function	05 hex
	Starting Address Hi	00 hex
	Starting Address Lo	04 hex
	No. of Points Hi	FF hex
	No. of Points Lo	00 hex
	Error Check (CRC)	—

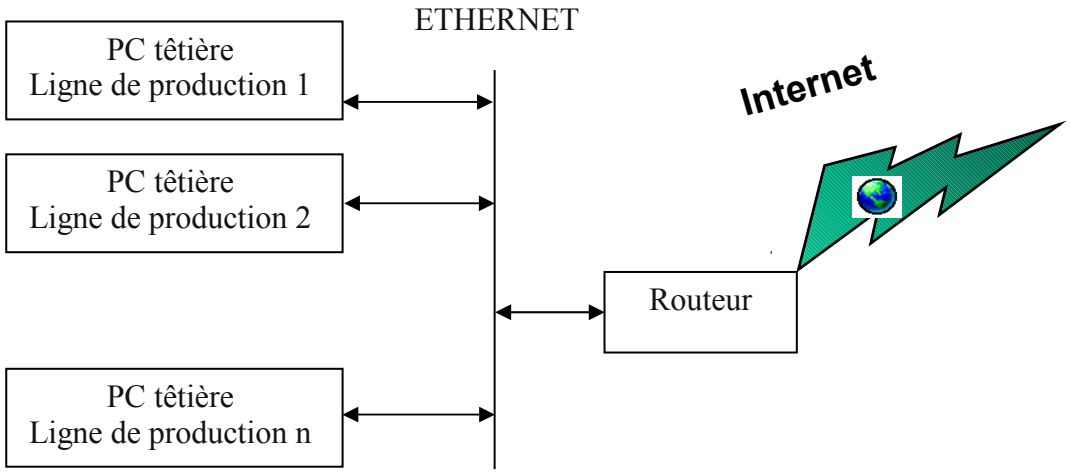
Adr. Coil	Action
00	
01	
02	
03	
04	NO-NC 1
05	NO-NC 2
06	
07	
08	
09	
10	
11	
12	
13	
14	
15	

Requête : forcer le seuil __ du coil __ en NC / NO (entourer le mode configuré).

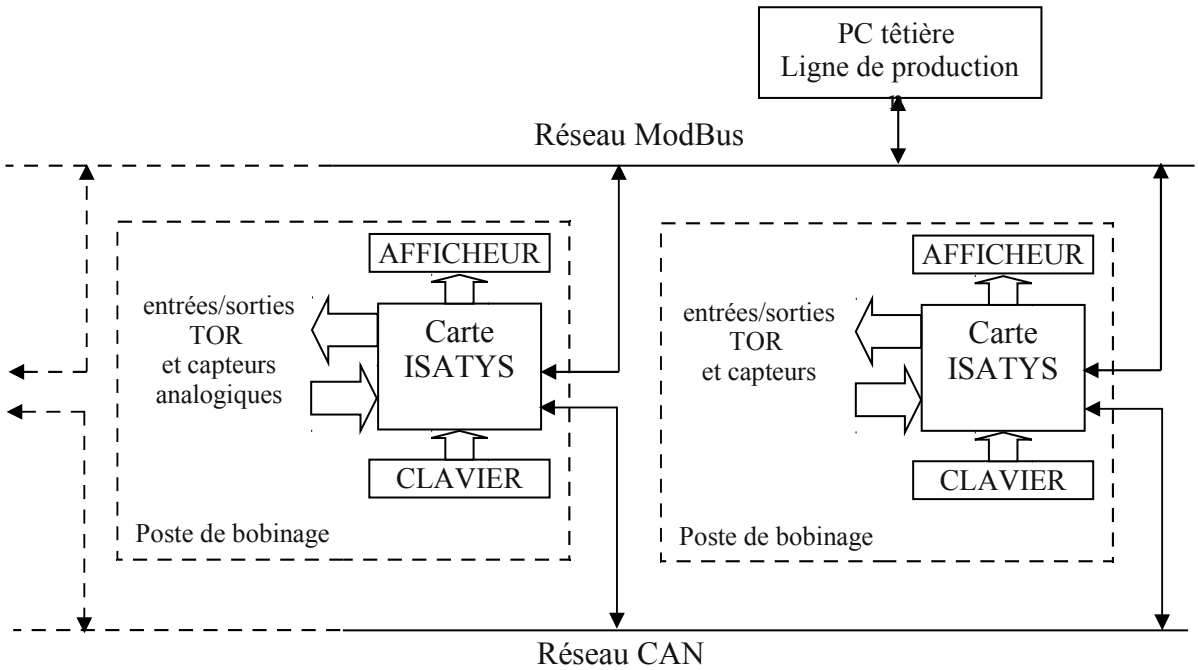
Programmation (ESI 2005)

Une machine est organisée en **lignes de production** constituées de **postes de bobinage**.

Le PC (appelé têtière) qui pilote chaque ligne de production est un PC industriel type Pentium avec système d'exploitation **LINUX**. Il permet de gérer la ligne de production composée de plusieurs postes de bobinage, de remonter les informations pour la supervision et de communiquer par Internet avec le fournisseur pour la télémaintenance.

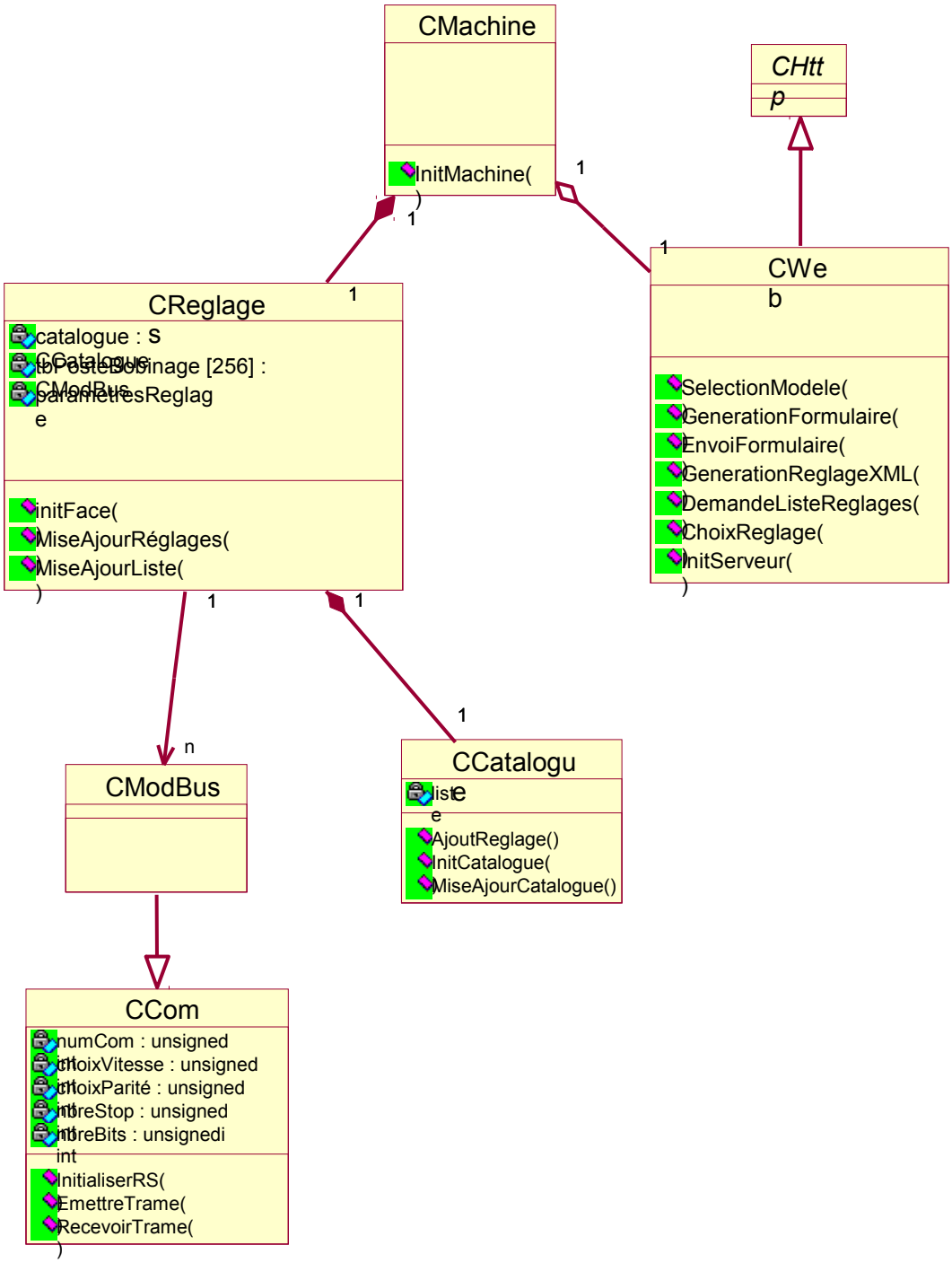


Chaque poste de bobinage est piloté par une carte à microcontrôleur (appelée carte ISATYS) reliée par **réseau** ModBus au PC têtière. Par ailleurs, les postes s'échangent des informations issues des capteurs grâce à un **réseau CAN** (non étudié ici).



Architecture d'une ligne de production d'enroulement de fils

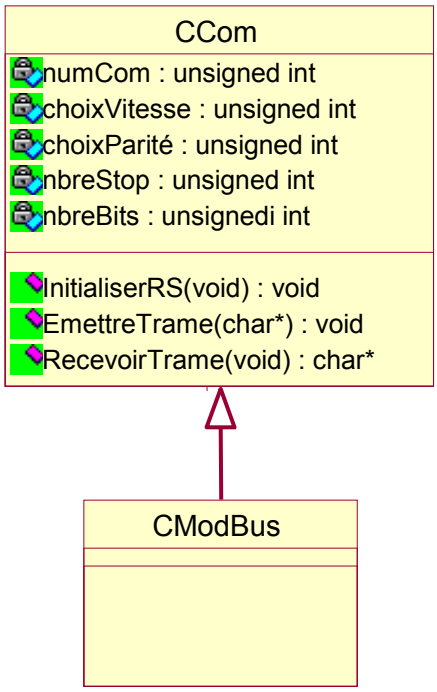
La modélisation du domaine de cette application est représentée par l'extrait du diagramme de classes suivant.



Exercice n°8

Que signifie la relation entre la classe CModBus et la classe CCom ?
Réponse :

Étude de la communication entre PC pilote ligne et poste de bobinage



La classe CModBus permet d'envoyer et recevoir les trames nécessaires à la communication avec les postes de bobinage. Elle assure la qualité des échanges en générant un CRC et en contrôlant celui reçu des esclaves.

Pour chaque poste de bobinage, le système instancie la classe CModBus. Le constructeur reçoit alors l'adresse ModBus de ce poste (sous forme d'un entier).

Il initialise les attributs :

- adrPoste de l'objet créé
- le CRC à sa valeur initiale
- une chaîne de caractères (prévue à la taille maximale pour envoyer ou recevoir tout type de trames)
- un compteur associé à la chaîne de caractères initialisé à 0.

Pour envoyer et recevoir des trames, la classe dispose de 2 méthodes auxquelles les arguments suivants sont fournis :

	Méthode EnvoyerCommande	Méthode RecevoirReponse
Arguments fournis aux méthodes	le code de la fonction	un pointeur sur le code de la fonction exécutée par l'esclave
	l'adresse du 1er bit ou mot à lire ou à écrire	un pointeur sur l'adresse du 1er bit ou mot lu ou écrit
	la longueur en nombre de bits ou mots	un pointeur sur la longueur en nombre de mots lus ou écrits
	pointeur sur le tableau contenant les bits ou mots à lire ou à écrire	un pointeur sur le tableau recevant les bits ou mots lus ou écrits

Le calcul du CRC fait l'objet d'une méthode dont le prototype est :
void CalculerCRC( );

Exercice n°9

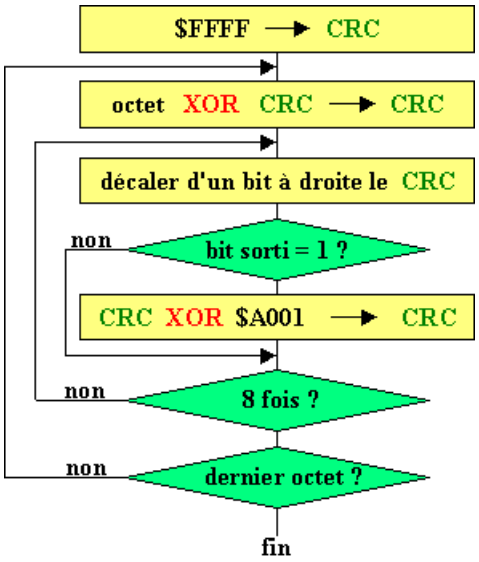
Compléter en C++ **la déclaration** de la classe CModBus en précisant les attributs et les méthodes (et leurs arguments) nécessaires à la gestion du réseau ModBus. *Réponse :*

Exercice n°10

Écrire en C++ le constructeur de la classe CmodBus. *Réponse :*

Le CRC (*Cyclical Redundancy Check*) est calculé par l'émetteur avant d'être transmis. Le récepteur calcule aussi le CRC et le compare avec le CRC reçu : des valeurs différentes indiqueront une erreur dans la transmission du message.

Le CRC, codé sur 2 octets (16 bits), est basé sur un OU EXCLUSIF (XOR) et se calcule de la façon suivante :



Traduction de l'organigramme en pseudo-code :

```
DEBUT
    CRC = FFFFh
    OCTET SUIVANT = premier octet de la trame

    REPETER
        CRC = CRC ⊕ OCTET SUIVANT
        POUR CPT VARIANT DE 1 A 8
            FAIRE
              CRC = CRC décalé d'un bit à droite
              SI BIT DECALE := 1 ALORS
                    FAIRE
                          CRC = CRC ⊕ A001h
                    FIN SI
            FIN POUR
        OCTET SUIVANT = octet suivant dans la trame
    TANT QU'IL RESTE DES OCTETS DANS LA TRAME
FIN
```

Le symbole ⊕ indique une opération 'OU exclusif '.

Exercice n°11

Écrire en C++ la méthode CalculerCRC( ) en traduisant l'algorithme proposé ci-dessus.
Réponse :

Exercice n°12

Quel est le nombre maximum d'appareils qu'on peut trouver sur un réseau ModBus ? Justifier la réponse.
Réponse :

Exercice n°13

Quelle est l'adresse de diffusion sur un réseau ModBus et quelle est sa fonction ?
Réponse :

