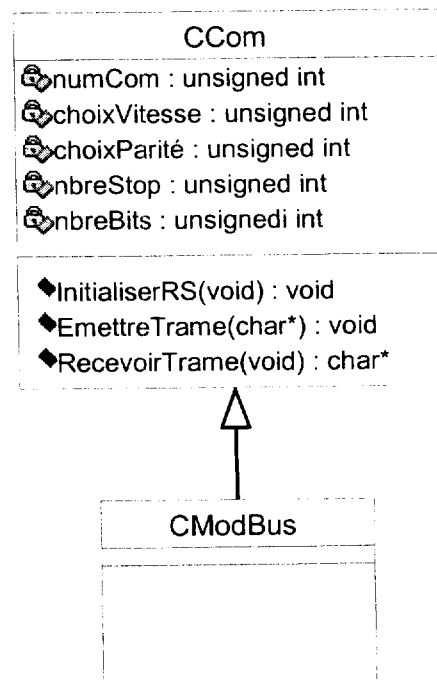


D. Etude du poste de bobinage

D.1 Etude de la communication entre PC pilote ligne et poste de bobinage

Utiliser les extraits de la documentation ModBus donnés en annexe 4 pour répondre aux questions suivantes.



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	le code de la fonction	un pointeur sur le code de la fonction exécutée par l'esclave
fournis	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
aux	la longueur en nombre de bits ou mots	un pointeur sur la longueur en nombre de mots lus ou écrits
méthodes	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();

Question D.1.1

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.

Question D.1.2

Ecrire en C++ le constructeur de la classe CModBus.

Question D.1.3

Ecrire en C++ la méthode CalculerCRC() en traduisant l'algorithme proposé dans la documentation.

Question D.1.4

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

Question D.1.5

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

D.2 Etude des trames échangées entre le poste pilote et les cartes ISATYS

Lors de la remise à zéro des compteurs, on relève sur l'analyseur les trames suivantes :

Question du contrôleur : 01 10 07 E6 00 03 06 00 00 00 00 00 18 BD

Réponse de l'esclave : 01 10 07 E6 00 03 60 8B

Question D.2.1

Interpréter ces trames relevées sur le réseau ModBus ? Détailler les champs des trames.

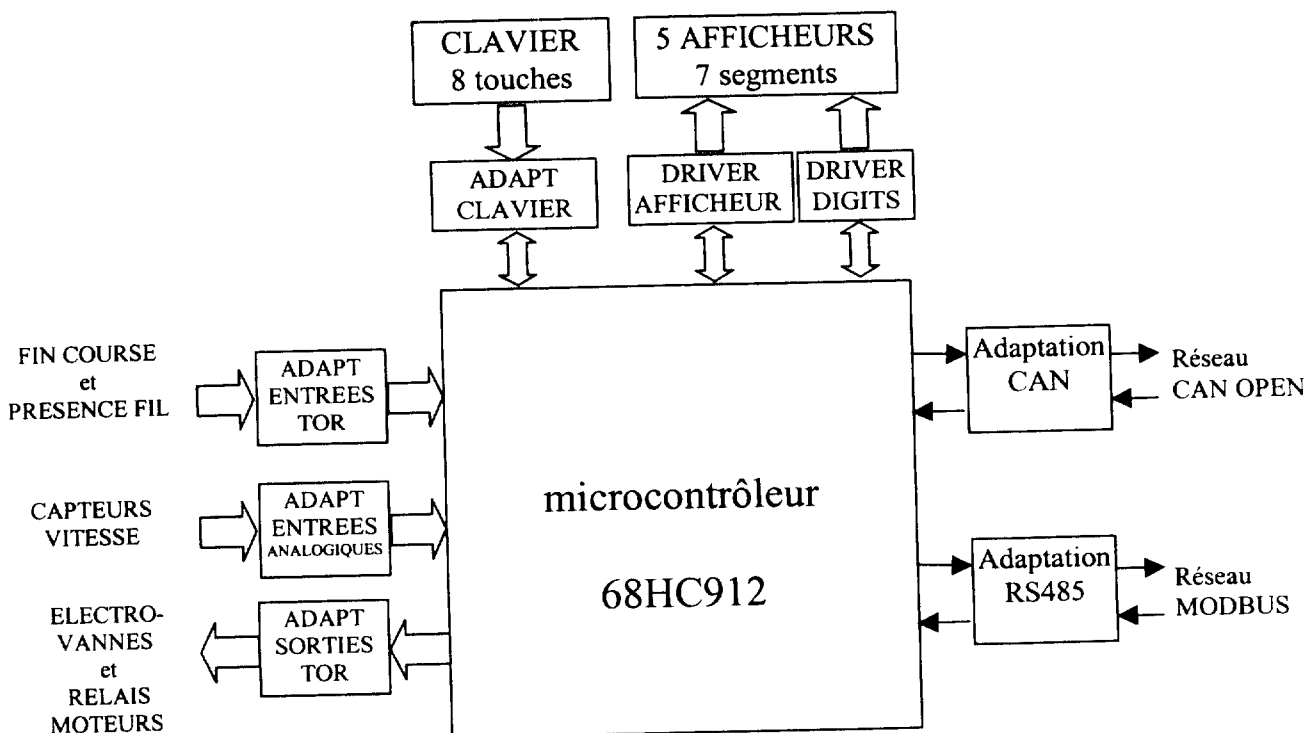
NB : dans le calcul du temps nécessaire à un échange, seules les durées de « Début de trame » sont à prendre en compte, les « Fin de trame » se chevauchant avec les temps d'attente nécessaires au maître ou à l'esclave ou avec le « Début de trame » de l'échange suivant.

Question D.2.2

Compte tenu de la configuration de la liaison (vitesse 4800 bauds, données sur 8 bits sans parité, 1 bit de stop) et du délai d'attente de l'esclave pour effectuer la commande, calculer le temps nécessaire pour cet échange ?

D.3 Configuration du poste de bobinage

Le poste de bobinage est géré par la carte ISATYS, architecturée autour d'un 68HC912. Le programme embarqué permet de superviser l'enroulement des fils sur les bobines.



Synoptique détaillé de la carte ISATYS (une par poste de bobinage)

Les fonctions suivantes sont disponibles :

- un clavier visu pour afficher le métrage et acquitter les alarmes.
- des entrées TOR permettant la surveillance capteurs « fin de course » et « présence fil ».
- des entrées ANALOGIQUES permettant la mesure des capteurs de vitesse.
- des sorties TOR pilotant les moteurs et les électrovannes.
- une liaison RS485 (réseau ModBus) pour recevoir le paramétrage des fabrications et remonter les alarmes.
- un bus CAN OPEN assurant un dialogue inter postes de bobinage lorsque le fil à assembler doit se faire sur plusieurs postes.

Les questions suivantes portent sur la configuration de la carte effectuée par le technicien lors du montage.

Une LIGNE DE PRODUCTION peut contenir un grand nombre de bobineurs. Le concepteur de la carte a prévu des cavaliers de configuration en fonction de la place occupée sur le réseau ModBus (seule la liaison esclave est utilisée).

Utiliser les annexes 5 à 7 (extraits du schéma carte ISATYS, documentation du composant LTC 1487 et la note d'application fournie par LINEAR TECHNOLOGIE) pour répondre aux questions suivantes.

Question D.3.1

Compléter le chronogramme des signaux de transmission relevés au niveau du connecteur X2 dans le cas d'une émission d'un esclave ?

Question D.3.2

Préciser la position des cavaliers de SW1 en fonction de la position des esclaves sur le ModBus ?

Question D.3.3

Quelles sont les raisons du choix du réseau ModBus ?

Question D.3.4

Citer un autre bus qui aurait pu être utilisé dans cette application ?

Utiliser l'annexe 8 (documentation du composant VN330SP) pour répondre à la question suivante.

Question D.3.5

Quel est le rôle des composants VN 330 SP ?

E. Multitâche

Chaque ligne de production est autonome. En tête, un PC appelé "tête" gère les cartes ISATYS par le réseau ModBus, la télémaintenance par un serveur WEB et la supervision par l'interface utilisateur XWindows. Le système d'exploitation est LINUX.

Au démarrage de ce PC, les processus suivants sont lancés automatiquement :

- Serveur WEB Apache pour la télémaintenance.
- Serveur FTP pour le téléchargement des mises à jour de fichiers (Documentation et exécutable).
- Un processus nommé "production" qui pilote la ligne de production en communiquant avec les postes de bobinage par le réseau ModBus. Ce processus crée une mémoire partagée avec une structure qui contiendra les valeurs de tous les capteurs et en assure la mise à jour périodique.
- Des processus de supervision dont les affichages sont déportés sur des terminaux X.

Les processus sont lancés en multitâche par les primitives *fork()* et *execl()* (processus lourds).

Question E.1.1

Préciser la différence de gestion de la mémoire entre un processus lourd et le parallélisme qui peut aussi être réalisé par un thread (processus léger).

Le nombre maximum de postes de bobinage géré par un PC "tête" est 130.

Sur chaque poste il y a 4 capteurs de vitesse et les entrées TOR sont regroupées bit à bit dans un entier.

Les données à stocker dans la mémoire partagée sont :

- Référence de la production en cours (texte de 40 caractères).
- Tableau des entrées TOR des 130 postes de bobinage (entier dans lequel chaque bit correspond à l'état d'un capteur.).
- Tableau des 4 capteurs de vitesse de chacun des 130 postes (chaque vitesse étant un réel).

Question E.1.2

Ecrire en C++ la définition de la structure de la mémoire partagée appelée "infoBobinage".

Le lancement du processus "*production*" se fait en lui passant comme paramètre un numéro de clé qui servira d'accès à la mémoire partagée.

Lors de son initialisation ce processus :

- transforme ce paramètre en un entier avec lequel il crée la mémoire partagée nécessaire à la structure *infoBobinage*.
- récupère le pointeur sur cette zone mémoire pour pouvoir y accéder.
- lance en multitâche l'exécution du programme "*supervision*" en lui passant en paramètre le numéro de la clé afin qu'il puisse utiliser cette mémoire partagée.

Question E.1.3.1

Compléter l'extrait du fichier source *production.cpp* en utilisant la documentation sur les primitives UNIX (annexe 9).

Le processus "*production*" se met ensuite en attente de réception d'une trame. A l'arrivée de cette dernière le décodage permet de stocker le numéro du poste dans l'entier *n* et la valeur des capteurs TOR dans l'entier *valCapteursTOR*.

Question E.1.3.2

Ecrire la ligne transférant la variable *valCapteursTOR* vers le champ prévu dans la mémoire partagée.

La mémoire partagée est accessible en écriture par le processus *production* et en lecture par les processus de *supervision* et de *télémaintenance*. L'accès devant être exclusif pour chaque processus, on prévoit de créer un sémaphore booléen pour en contrôler l'accès.

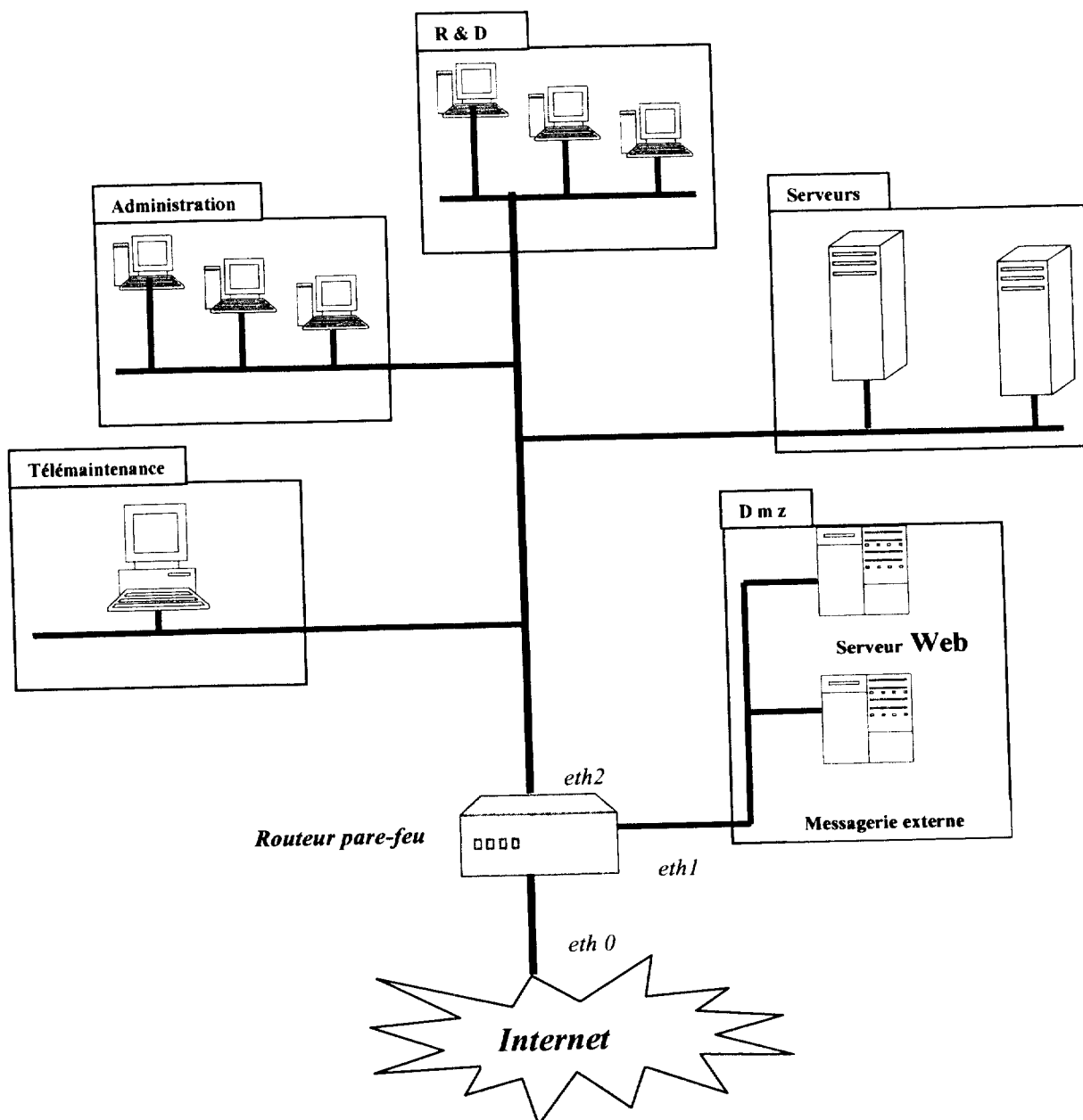
Question E.1.4

La valeur initiale de ce sémaphore étant "1", expliquer le fonctionnement lors des opérations de prise (P) et de libération (V) de la ressource.

F. Réseaux

F.1 Architecture du réseau du site fournisseur

On dénombre 4 sous-réseaux dont la télémaintenance reliés au routeur WAN.
Une zone Dmz gérée par le pare-feu intègre le serveur Web de la société.



F.2 Ethernet

Question F.2.1

Décrire la méthode d'accès au support de transmission utilisé par le réseau Ethernet.
Peut-on qualifier ce réseau de probabiliste ou déterministe ?

Le support de transmission est de type 100 base T.

Question F.2.2

Expliquer le terme 100 base T.

Question F.2.3

Donner le nom du codage utilisé par Ethernet pour la transmission du signal.
Expliquer le principe général de ce type de codage et son rôle.

F.3 Adressage IP

Les 4 sous réseaux sont reliés à l'interface eth2. L'adresse réseau est 128.128.0.0

Question F.3.1

Donner la classe d'adresse de ce réseau. Justifier votre réponse.

Question F.3.2

Donner le masque des sous-réseaux. Justifier votre réponse.

Question F.3.3

Donner les adresses des différents sous-réseaux ainsi que la plage maximum d'adresses utilisables pour chacun des sous-réseaux.

Question F.3.4

Qu'est-ce qu'une adresse « *broadcast* » ?
Compléter dans le tableau réponse les adresses broadcast de chaque sous-réseau.

F.4 Accès Internet

Pour l'accès au Web, la société a acquis un routeur WAN pare-feu avec un abonnement spécifique à un fournisseur Internet.

Après l'installation du routeur, il faut configurer les stations clientes :

Question F.4.1

Quelle doit être l'adresse de passerelle à préciser dans les paramètres réseau des stations souhaitant accéder à Internet ?

L'acronyme *Dmz* signifie « demilitarized zone » ou « zone démilitarisée »

Question F.4.2

Expliquer la fonction de cette zone Dmz .

Question F.4.3

Quelles seraient les fonctionnalités apportées par l'installation d'un serveur proxy HTTP ?