

**BTS INFORMATIQUE ET RÉSEAUX**  
**POUR L'INDUSTRIE ET LES SERVICES TECHNIQUES**

SESSION 2005

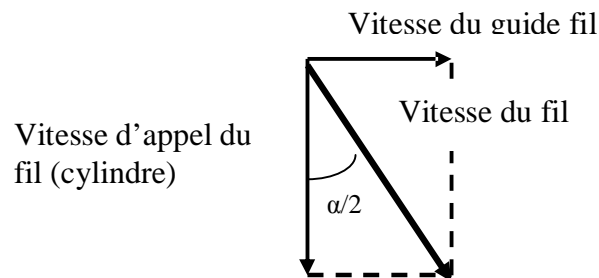
**Épreuve E.4**  
**Étude d'un système informatisé**  
*Chaîne de bobinage textile*

**Corrigé (14 pages)**

**Réponses aux questions B.1**

---

**B.1.1**



Vitesse du fil = vitesse d'appel du fil /  $\cos (\alpha / 2) = 50 / \cos 30^{\circ} = \mathbf{57.74 \text{ m/min}}$

**B.1.2** Unité de mesure de la torsion : ... **tours / mètre** .....

**B.1.3** Temps de fabrication d'une bobine =  $15000 / 57.74 \approx \mathbf{260 \text{ min}}$  .....

## Réponses aux questions B.2

---

**B.2.1** Titrage du fil assemblé :

**Nm 2/10** = 2 fils simples de 1/10 assemblés donnent **Nm 1/5**

**B.2.2** Poids du fil assemblé :

15000 / 5 = **3000 g**

## Réponses aux questions B.3

---

**B.3.1** Principe de fonctionnement d'un capteur capacitif :

Les capteurs capacitifs sont des capteurs de proximité qui permettent de détecter des objets métalliques ou isolants. Lorsqu'un objet entre dans le champ de détection des électrodes sensibles du capteur, il provoque des oscillations en modifiant la capacité de couplage du condensateur. Un circuit de communication met en forme cette information.

**B.3.2** Intérêt de ce type de capteur :

Cette technologie rend le détecteur insensible à l'encrassement et aux vibrations.

## Réponses aux questions C.1

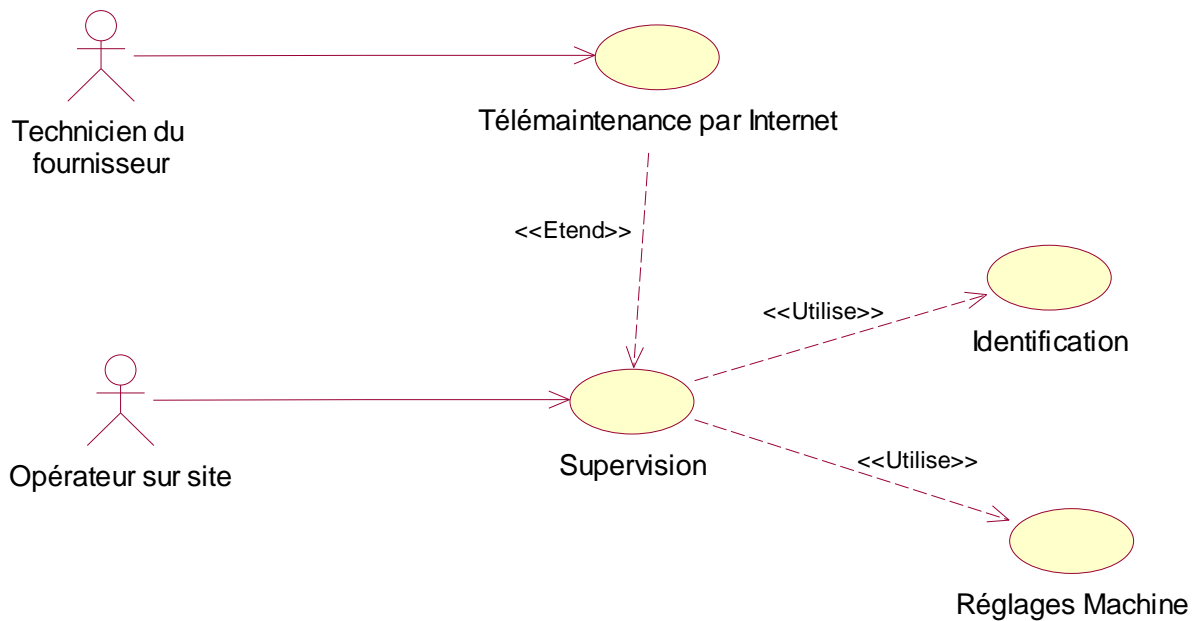
---

### C.1.1 Rôle des cas d'utilisation :

Les cas d'utilisation décrivent le comportement du système du point de vue de l'utilisateur. Ils permettent de définir les limites du système et les relations entre le système et l'environnement.

L'expression des besoins est recentrée sur les utilisateurs.

### C.1.2



## Réponses aux questions C.2

---

### C.2.1 Implication au niveau de ses méthodes :

La classe CModBus hérite de la classe CCom.

### C.2.2 Implication au niveau de ses méthodes :

La classe CHttp possède au moins une fonction membre virtuelle pure.

Utilisation d'une classe abstraite :

Une classe abstraite est une classe qui n'est pas instanciable directement.

La fonction membre virtuelle pure devra être implémenté dans les classes dérivées.

### C.2.3

Relation : composition encore appelée agrégation par valeur ;

Contrainte : la durée de vie d'un objet CReglages est liée à celle de l'objet CMachine.

Traduction C++ :

```
class CReglages ;
class CMachine
{
```

.....

```
private :
    CReglages reglage ;
```

.....

```
};
```

## C.2.4

Relation : Agrégation par référence

Traduction C++ :

```
class CWeb ;  
class CMachine  
{  
.....
```

```
private:  
    CWeb *navigateur;  
.....
```

```
};
```

```
# include "cweb.h"
```

```
CMachine::CMachine  
{  
.....
```

```
navigateur = new CWeb ( );  
.....
```

```
}
```

```
CMachine::~~CMachine  
{  
.....
```

```
delete navigateur;  
.....
```

```
}
```

## Réponses aux questions D.1

---

### D.1.1 Déclaration de la classe CModBus

```
class CModBus : public CCom
{.....
..... private :
..... unsigned char  adrPoste ; // entre 1 et 255
..... unsigned int CRCModBus ; // permet le contrôle des échanges
..... unsigned char  chTrameModBus [260] ; // long maxi trame :125 mots
..... unsigned int lgTrame ; // permet de connaître la longueur de la chaîne envoyée ou reçue
..... public :
..... void CModBus(unsigned int) ; //le constructeur reçoit l'adresse de l'esclave
..... void EnvoyerCommande( unsigned char codeFonct, unsigned int adPremierMot,
                           unsigned int nbreMots, unsigned int *tbMots );
..... void RecevoirReponse( unsigned char *codeFonctExecute, unsigned int *adPremierMot,
                           unsigned int *nbreMots, unsigned int *tbMots );
.....void CalculerCRC() ;
};
```

### D.1.2 constructeur de la classe CModBus

```
CModBus::CModBus (unsigned int adrModBus) : CCom ( )
{
adrPoste =(unsigned char) adrModBus ; // met à jour l'attribut de la classe sur un octet
CRCModBus= 0xFFFF ;// valeur d'initialisation
lgTrame =0 ; //trame vide
};
```

**D.1.3** donner le code de la méthode `CalculerCRC()` en traduisant l'algorithme proposé dans la documentation.

```

..... CModBus::CalculerCRC()
..... {
.....     unsigned int nbreBits , i=0 , octetDecale;
.....     CRCModBus =0xFFFF ;
.....     octetDecale = ( chTrameModBus[i++] ) << 8;
.....     do
.....     {
.....         CRCModBus = CRCModBus ^ octetDecale;
.....         for (nbreBits =0 ; nbreBits <8 ; nbreBits ++ )
.....         {
.....             if(CRCModBus&0x01)= = 0x01) //test si dépassement
.....             {
.....                 CRCModBus = CRCModBus >>1 ;
.....                 CRCModBus = CRCModBus ^ 0xA001 ;
.....             }
.....             else
.....             {
.....                 CRCModBus = CRCModBus >>1 ;
.....             }
.....         }
.....         octetDecale = ( chTrameModBus[i] ) << 8;
.....     }
.....     while (i++ < lgTrame ); //jusqu'à la fin de la chaîne
..... }

```

**D.1.4** nombre maximum d'appareils sur ModBus? Justifier la réponse.

...Adresse sur 8 bits, donc 255 adresses possibles (car l'adresse 0 est réservée à la diffusion)  
permettant 254 esclaves (appareils possibles) + 1 contrôleur.

**D.1.5** quelle est l'adresse de diffusion sur un ModBus et quelle est sa fonction ?

.....Adresse 0 : adressage d'un message à tous les esclaves .....

## Réponses aux questions D.2

---

**D.2.1** interpréter ces trames relevées sur le ModBus? Détailler les champs des trames.

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

..... 01 : esclave 1

..... 10 : code fonction écriture

..... 07 E6 : adresse de départ à laquelle il faut écrire

..... 00 03 06: longueur 3 mots (6 octets)

..... 00 00 00 00 00 00 : contenu à écrire : reset des compteurs

..... 18 BD: CRC

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

..... 01 : esclave 1

..... 10 : écriture faite sans erreur

..... 07 E6 : adresse de départ

..... 00 03 : nombre de mots écrits

..... 60 8B: CRC



**D.2.2** compte tenu de la configuration de la liaison (vitesse 4800 bds, 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 ?

...en 4800 N81 chaque octet émis ou reçu nécessite 10 bits (start, 8 bits de données, 1 stop) = 2ms

...nombre d'octets émis par le contrôleur : 3,5 + 15

... délai d'attente : 20 ms \* 3 mots

... nombre d'octets émis par le récepteur : 3,5 + 8

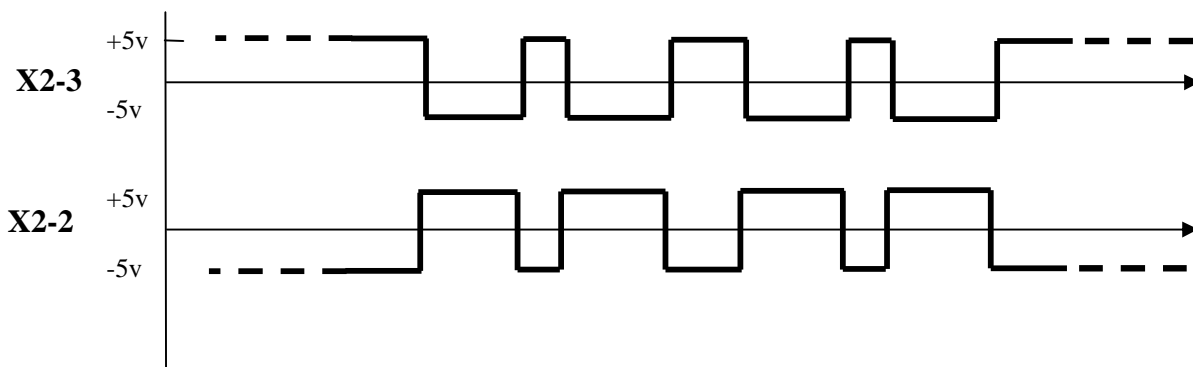
.....total : 30 octets \* 2ms + 60ms = 120 ms

*Ne pas pénaliser une réponse qui prendrait en compte une ou plusieurs fois les durées de « Fin de trame » et de « Début de trame » dans le calcul.*

### Réponses aux questions D.3

---

**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 ?



Quelles tensions peut-on mesurer en différentiel entre X2-3 et X2-2 ? ..... -10v et +10 volts

Correspondant à quel niveau logique sur Rx ? ..... 0 et 1 ...

**D.3.2** préciser la position des cavaliers de SW1 en fonction de la position des esclaves sur le ModBus ?

Dans le tableau suivant, compléter la position (ON / OFF) des cavaliers pour chaque carte :

NB : répondre par ON : contact fermé      ou OFF : contact ouvert      ou X : sans importance

	12-1	11-2	10-3	9-4	8-5	7-6
Esclave 1	X	X	X	ON	ON	ON
Esclave 2	X	X	X	OFF	OFF	OFF
Esclave 3	X	X	X	OFF	OFF	OFF
.....	X	X	X	OFF	OFF	OFF
Esclave N-1	X	X	X	OFF	OFF	OFF
Esclave N	X	X	X	ON	OFF	ON

**D.3.3** quelles sont les raisons du choix du réseau MODBUS ?

- ..... économique et simple à mettre en œuvre (câbles et connectique),
- ..... performant en nombre d'équipement et distances,
- ..... résistant aux parasites, adapté aux automatismes .....

**D.3.4** citer un autre bus qui aurait pu être utilisé dans cette application ?

- .....réponse attendue : .....bus CAN .....
- ..... autres possibilités : PROFIBUS, AXINA, ARYNC, AS-I, UNITELWAY, TCPIP ModBus,

**D.3.5** quel est le rôle des composants VN 330 SP ?

- ... Ils pilotent les relais en limitant la puissance et protégeant des courts-circuits et des températures élevées : AMPLIFICATION ET PROTECTION .....

## Réponses aux questions E.1

---

**E.1.1** différence entre processus lourd et processus léger :

	<b>thread (processus léger)</b>	<b>fork / execl (processus lourd)</b>
<b>Gestion de la mémoire</b>	Les codes exécutés en parallèle partagent le même espace mémoire et accèdent aux mêmes variables. ..... ..... ..... ..... ..... .....	Chaque processus a son propre espace mémoire et ne peut donc pas communiquer avec un autre sauf par des moyens de communication gérés par le noyau (par exemple une mémoire partagée). ..... ..... ..... .....

**E.1.2** définition de la structure de la mémoire partagée :

```

... const int Lg = 40 ; .....
... const int nbPosteBob = 130 ; .....
... struct infoBobinage { .....
..... char reference [Lg] ; .....
..... int capteursTOR [ nbPostesBob ] ; .....
..... float vitesses [ nbPostesBob ] [4] ; .....
..... } infoBobinage ; .....
.....

```

**E.1.3.1** extrait du fichier production.cpp :

```
int main ( int argc, char *argv[] )
{
// Calcul de la clé (entier) à partir du paramètre passé au lancement
... int key = atoi ( argv [1] ); .....

// Création de la mémoire partagée : shmget()
...int idShm = shmget (key, sizeof(infoBobinage ), IPC_CREAT/0666); .....

// Récupération du pointeur sur cette mémoire : shmat()
... infoBobinage * pInfo = (infoBobinage *) shmat (idShm, NULL, 0); .....

// Duplication du processus : fork()
...pid_t idSupervision = fork ( ) ; .....

// Recouvrement du processus fils créé par l'exécutable supervision : execl() :
... if (idSupervision == 0) execl("supervision", "supervision", argv[1], NULL );
```

**E.1.3.2** Ecriture de la variable *valCapteursTOR* dans la mémoire partagée :

```
... pInfo -> capteursTOR [n] = valCapteursTOR ; .....
```

**E.1.4** fonctionnement d'un sémaphore :

Le sémaphore booléen est une variable système initialisée à 1

Avant d'accéder à la ressource critique, chaque processus appelle une primitive réalisant l'opération P (prise de la ressource) :

- Si la valeur du sémaphore est supérieure à 0, l'opération P rend la main après avoir décrémenter le sémaphore (mis à 0 dans le cas d'un sémaphore booléen).
- Si la valeur du sémaphore est 0, l'opération est bloquante jusqu'à ce que sa valeur remonte à 1.

Après l'accès à la ressource il faut réaliser l'opération V qui incrémente la valeur du sémaphore, ce qui libérera la ressource et débloquera éventuellement un autre processus en attente.

## Réponses aux questions F.2

---

### F .2.1

Méthode d'accès : C S M A / CD

Réseau : Probabiliste (temps d'accès aléatoire)

F.2.2 100 : 100 M bits / s

Base : codage en bande de base

T : Twisted-Pair ( paires torsadées)

### F.2.3

Nom du codage utilisé par Ethernet: **Manchester**

Principe général de ce type de codage:.....

**La transmission en bande de base est une transmission sans transposition de fréquence par modulation.**

Les codages numériques sont utilisés pour plusieurs raisons :

- Ø La récupération du signal d'horloge facilitée par des variations du signal pour chaque bit d'information transmis
- Ø Le spectre d'un signal binaire est concentré sur les fréquences basses qui sont les plus affaiblies sur la ligne.
- Ø Les perturbations subies par un signal sont proportionnelles à la largeur de sa bande de fréquence.

**Les codages en bande de base vont donc essentiellement avoir pour rôle de diminuer la largeur de bande du signal binaire et de transposer celle-ci vers des fréquences plus élevées**

### Réponses aux questions F.3

---

**F.3.1** classe d'adresse : ... classe B ... 128 -> 10 0 0 0 0 0 0 .....  
↙  
Classe B

**F.3.2** masque de sous-réseau : 255.255.192.0  
 Classe B -> masque de la forme 255.255.0.0  
 4 sous-réseaux -> 2 bits attribués au réseau

#### F.3.3

Nom	Adresse Sous-Réseau	Adresse début / adresse fin des machines	Adresse Broadcast
Serveurs	128.128.0.0	128.128.0.1 / 128.128.63.254	128.128.63.255
Administration	128.128.64.0	128.128.64.1 / 128.128.127.254	128.128.127.255
R & D	128.128.128.0	128.128.128.1 / 128.128.191.254	128.128.191.255
Télémaintenance	128.128.192.0	128.128.192.1 / 128.128.255.254	128.128.255.255

**F.3.4** Définition d'une adresse Broadcast :  
 C'est une adresse de diffusion qui permet d'envoyer un message à toutes les machines situées sur le réseau.

### Réponses aux questions F.4

---

**F.4.1** adresse passerelle = adresse du routeur *eth2*

**F.4.2** zone Dmz : C'est une zone qui n'est ni publique, ni interne. Elle permet l'accès du serveur Web aux clients de l'entreprise tout en sécurisant le réseau interne. Cette zone ne peut exister que si le pare-feu possède plus de deux interfaces réseau, une pour la connexion au réseau externe et l'autre pour la connexion au réseau.

**F.4.3** serveur proxy http :

Un serveur proxy HTTP possède 3 fonctions principales :

- Ø Cache : capacité à garder en mémoire les pages les plus souvent visitées par les utilisateurs du réseau local afin de pouvoir les leur fournir le plus rapidement possible.
- Ø Filtrage : suivi des connexions via la constitution de journaux d'activité
- Ø Authentification : à l'aide d'un nom d'utilisateur et d'un mot de passe par exemple