

<p style="text-align: center;"><b>BREVET DE TECHNICIEN SUPÉRIEUR INFORMATIQUE ET RÉSEAUX POUR L'INDUSTRIE ET LES SERVICES TECHNIQUES</b></p>
--

**SESSION 2005**

**Épreuve E.4**

**Étude d'un système informatisé**

<p style="text-align: center;"><b>Durée : 6h 00</b></p>
---

<p style="text-align: center;"><b>Coefficient 5</b></p>
---

<p><i>"Calculatrice autorisée (conformément à la circulaire n° 99-186 DU 16 NOVEMBRE 1999)."</i></p>
--

<p style="text-align: center;"><i>Aucun document autorisé.</i></p>
--

Ce document comprend :

<b>Sujet</b>	: pages 1 à 18	sur papier rose
<b>Annexes</b>	: pages 1 à 15	sur papier vert
<b>Document réponse</b>	: pages 1 à 14	sur papier blanc
<b>à rendre obligatoirement (même vierge).</b>		

**Dès que le sujet vous est remis, assurez-vous qu'il soit complet.**

# **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***

**Sujet**

**Durée : 6h 00      Coefficient 5**

*"Calculatrice autorisée (conformément à la circulaire n° 99-186 DU 16 NOVEMBRE 1999)."*

**Aucun document autorisé.**

**Toutes les réponses sont à fournir sur le livret "Document Réponse" à l'exclusion de tout autre support.**

**Les réponses doivent être exclusivement fournies dans les emplacements prévus à cet effet. Si nécessaire le candidat a la possibilité de rectifier ses réponses sur la page non imprimée en regard.**

**On ne justifiera une réponse que si le document le demande.**

#### **Temps conseillés et barèmes indicatifs**

<b>Lecture du sujet :</b>	<b>30 minutes.</b>	
<b>B) Assemblage des fils</b>	<b>45 minutes.</b>	<b>15 points</b>
<b>C) Modélisation UML</b>	<b>30 minutes.</b>	<b>15 points</b>
<b>D) Etude du poste de bobinage</b>	<b>120 minutes.</b>	<b>30 points</b>
<b>E) Multitâche</b>	<b>60 minutes.</b>	<b>20 points</b>
<b>F) Réseaux</b>	<b>60 minutes.</b>	<b>20 points</b>
<b>Relecture :</b>	<b>15 minutes.</b>	

# SOMMAIRE

A. Présentation du système.....	4
A.1 Présentation du contexte .....	4
A.2 La télémaintenance .....	5
A.3 Architecture matérielle du site client.....	6
B. Assemblage des fils .....	7
B.1 Principe .....	7
B.2 Titrages des fils.....	8
B.3 Contrôles du fil .....	8
C. Modélisation UML .....	9
C.1 Les cas d'utilisation .....	9
C.2 Diagramme de classes.....	10
D. Etude du poste de bobinage .....	12
D.1 Etude de la communication entre PC pilote ligne et poste de bobinage.....	12
D.2 Etude des trames échangées entre le poste pilote et les cartes ISATYS .....	14
D.3 Configuration du poste de bobinage.....	14
E. Multitâche.....	16
F. Réseaux .....	18
F.1 Architecture du réseau du site fournisseur .....	18
F.2 Ethernet.....	19
F.3 Adressage IP .....	19
F.4 Accès Internet .....	19

# A. Présentation du système

## A.1 Présentation du contexte

Depuis 1984, une société française fabrique des machines de production pour l'industrie du textile. L'utilisation de fibres aux propriétés variées a rendu les procédés de fabrication de plus en plus complexes. Les principaux débouchés sont l'habillement, l'ameublement, l'automobile, les pneumatiques, l'électronique, l'hygiène, l'aéronautique, la chimie, etc.

Sa maîtrise technique porte sur les outillages de transformation de fils (constitués de plusieurs fils de base aux qualités différentes) et leur enroulement en bobine.



*Vue d'une chaîne de fabrication de fils*

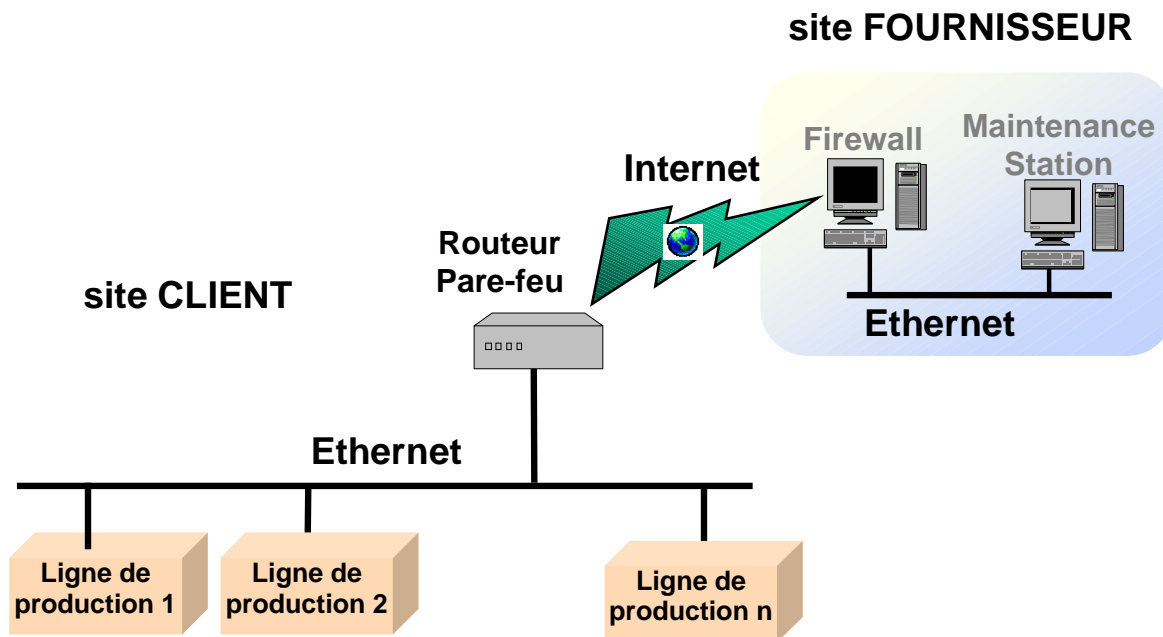
Dans les années 2000, sa restructuration industrielle a été menée après le rachat par un groupe européen, ce qui permit à l'entreprise d'aborder des marchés en Asie.

Constituée de 70% cadres et techniciens, cette société a recruté de nombreux techniciens supérieurs pour constituer son bureau d'étude.

## A.2 La télémaintenance

Ces dernières années, la stratégie de développement s'est organisée autour d'une idée puissante : la télémaintenance. En effet, les outils de communication d'Internet permettent d'offrir aux clients des services de « proximité » malgré la distance. Parmi ceux-ci, la société propose :

- de télécharger des mises à jour de logiciels,
- de faire la mise à jour de la documentation des machines,
- d'effectuer des tests de maintenance à distance.



*Synoptique général d'une application proposée au client*

Pour réaliser cet objectif, la société doit répondre aux difficultés suivantes :

- permettre au matériel de supporter des autotests
- obtenir des informations sans perturber les fabrications en cours
- assurer la sécurité des réseaux des clients

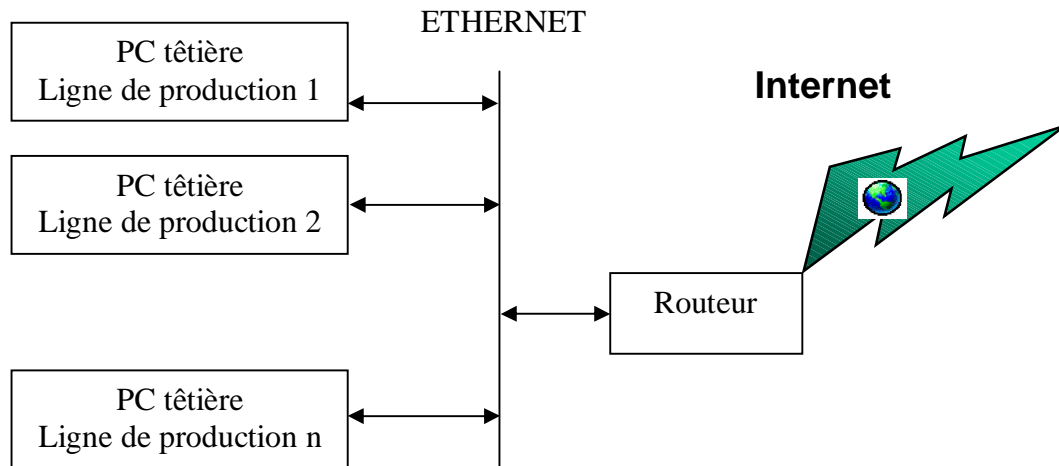
Après une phase de prise de connaissances du système technique, un technicien supérieur (sous la responsabilité d'un chef de projet) est chargé de mettre en place cette télémaintenance.

*L'étude portera donc d'une part sur la configuration des matériels utilisés pour gérer l'enroulement des fils sur les bobines, d'autre part sur les outils de communication à mettre en place.*

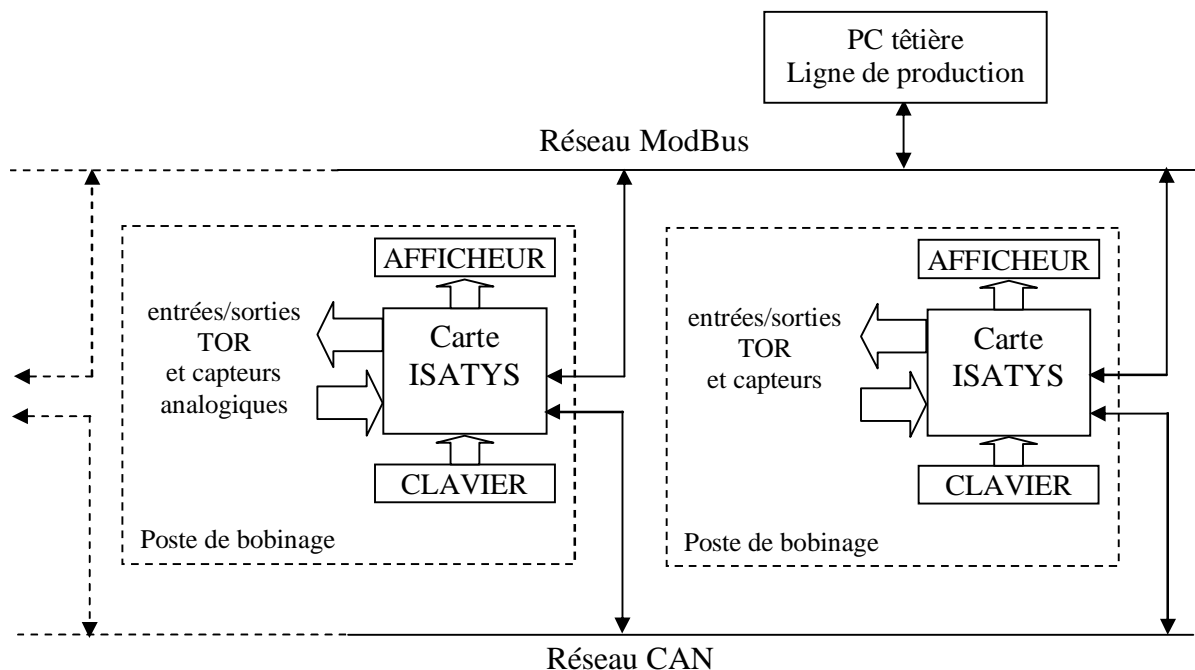
### A.3 Architecture matérielle du site client

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

Le PC (appelé têtère) 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 têtère. Par ailleurs, les postes s'échangent des informations issues des capteurs grâce à un **réseau CAN** (non étudié dans le sujet).

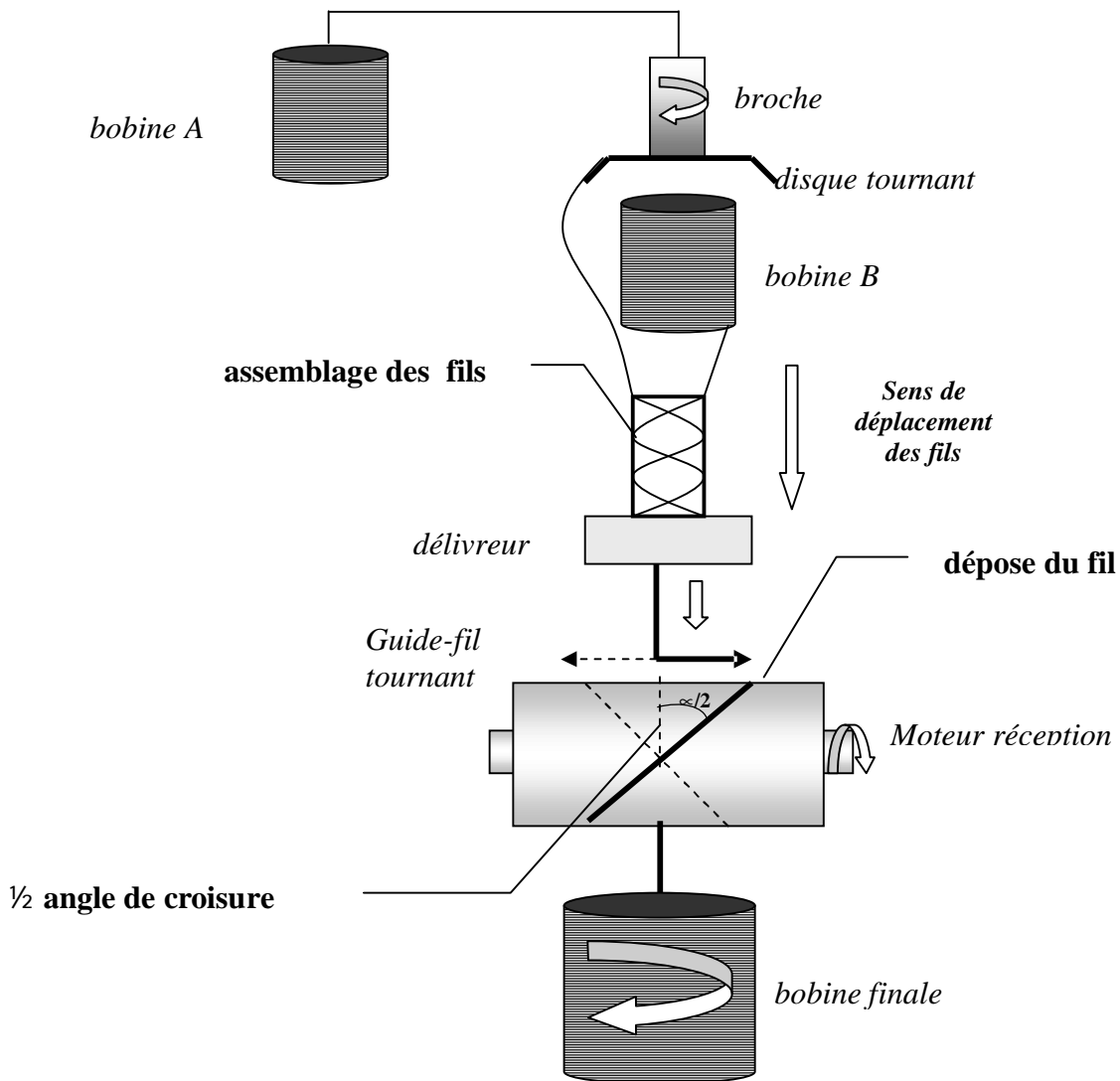


*Architecture d'une ligne de production d'enroulement de fils*

## B. Assemblage des fils

### B.1 Principe

La technique d'assemblage de 2 fils est schématisée par la figure suivante :



**Synoptique de l'assemblage**

On estime que le vecteur vitesse d'appel du fil est perpendiculaire au vecteur vitesse du guide fil. Les paramètres de réglage de la machine sont référencés dans le fichier xml de l'annexe 1.

**Question B.1.1**

Compléter le diagramme vectoriel permettant de déterminer le vecteur vitesse du fil en fonction du  $\frac{1}{2}$  angle de croisure  $\alpha/2$ .

Calculer sa valeur numérique pour un angle  $\alpha = 60^\circ$

**Question B.1.2**

Donner l'unité de mesure de la torsion.

**Question B.1.3**

Calculer le temps de fabrication d'une bobine en fonction du métrage donné.

**B.2 Titrages des fils**

Le titrage est la mesure permettant de déterminer la finesse d'un fil (voir annexe 2).

On procède à l'assemblage de 2 fils identiques de 1000 déciTex

**Question B.2.1**

Donner le titrage du fil assemblé en Numéro métrique.

**Question B.2.2**

Calculer le poids total du fil sur la bobine conformément aux réglages de l'annexe 1.

**B.3 Contrôles du fil**

Différents capteurs sont placés sur la ligne d'assemblage.

Le MCD-F est un capteur capacitif sans contact permettant le contrôle de la présence de fils.  
(voir annexe 3).

**Question B.3.1**

Quel est le principe de fonctionnement d'un capteur capacitif ?

**Question B.3.2**

Quel est l'intérêt de ce type de capteur dans l'environnement industriel étudié?



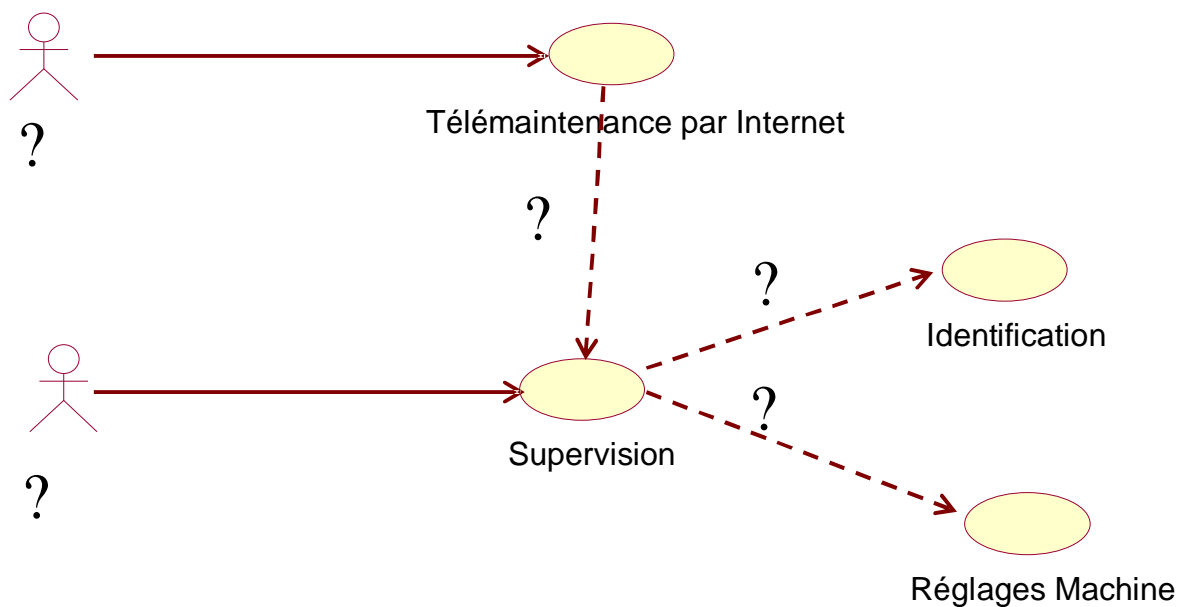
# C. Modélisation UML

## C.1 Les cas d'utilisation

Ce diagramme permet de préciser les différents acteurs et cas d'utilisation du système.

### Question C.1.1

Quel est le rôle des cas d'utilisation au sein de la modélisation objet avec UML ?

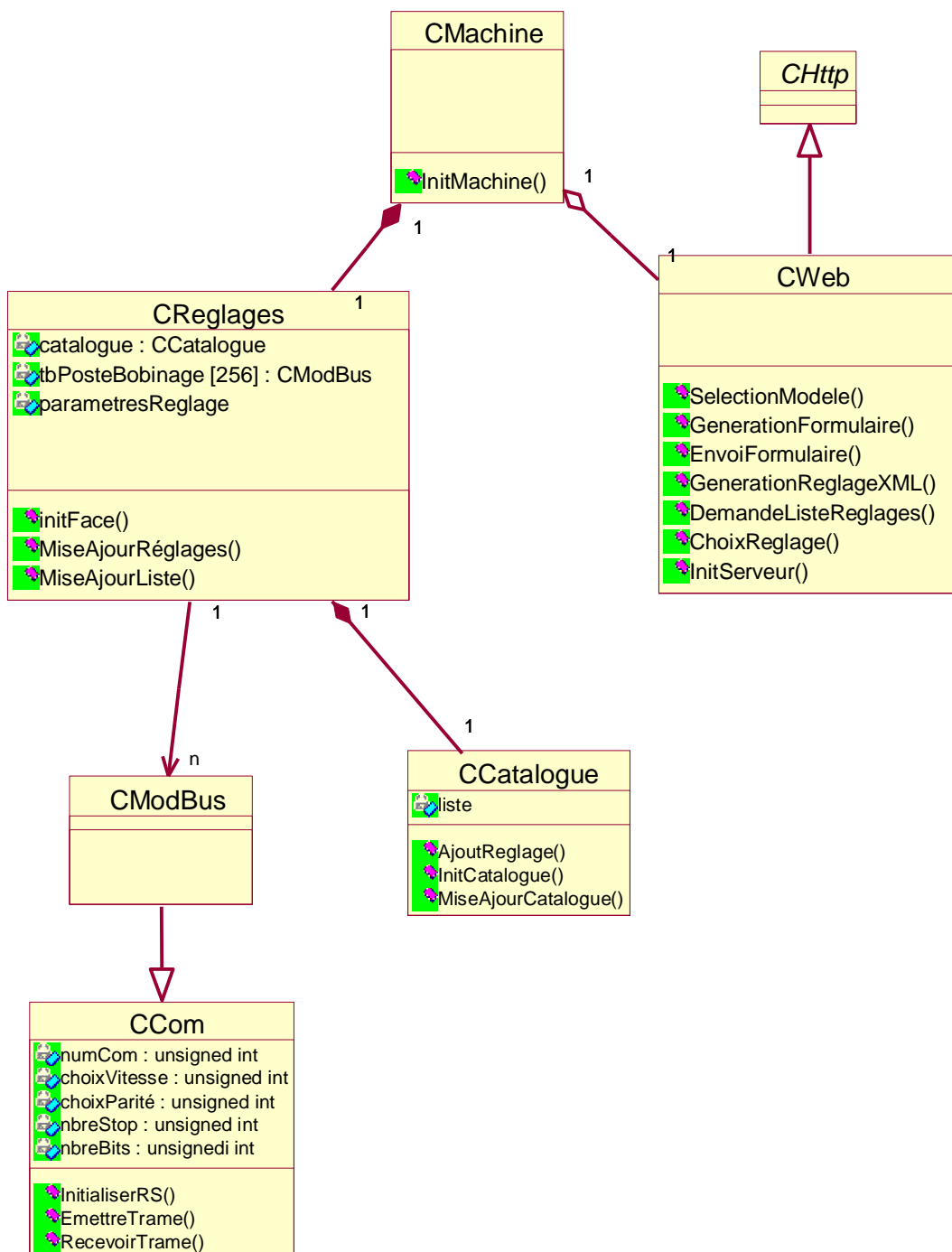


### Question C.1.2

Compléter ce diagramme en précisant le nom des acteurs et des relations entre cas d'utilisation.

## C.2 Diagramme de classes

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



**Question C.2.1**

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

La classe *CHttp* est une classe abstraite.

**Question C.2.2**

Qu'est ce que cela implique au niveau de ses méthodes ?

Décrire l'utilisation d'une classe abstraite.

La machine textile est pilotée par le système suivant différents types de réglages.

**Question C.2.3**

Définir le type de relation entre la classe CMachine et la classe CReglages.

Quelle est la contrainte imposée par ce type de relation ?

Traduire en C++ cette relation.

La télémaintenance de chaque machine est assurée via un serveur Web.

**Question C.2.4**

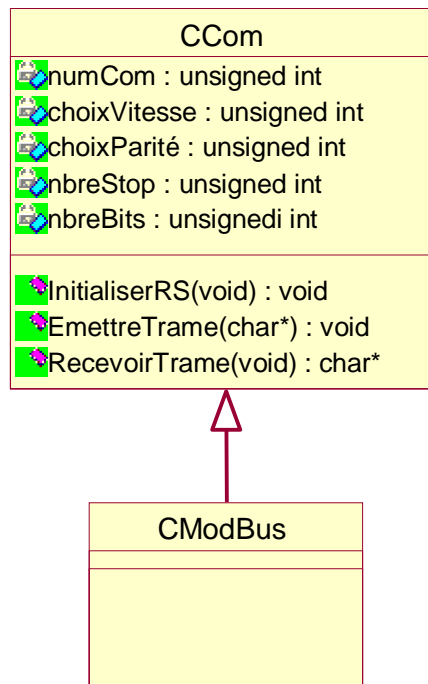
Définir le type de relation entre la classe CMachine et la classe CWeb.

Traduire en C++ cette relation.

## 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.

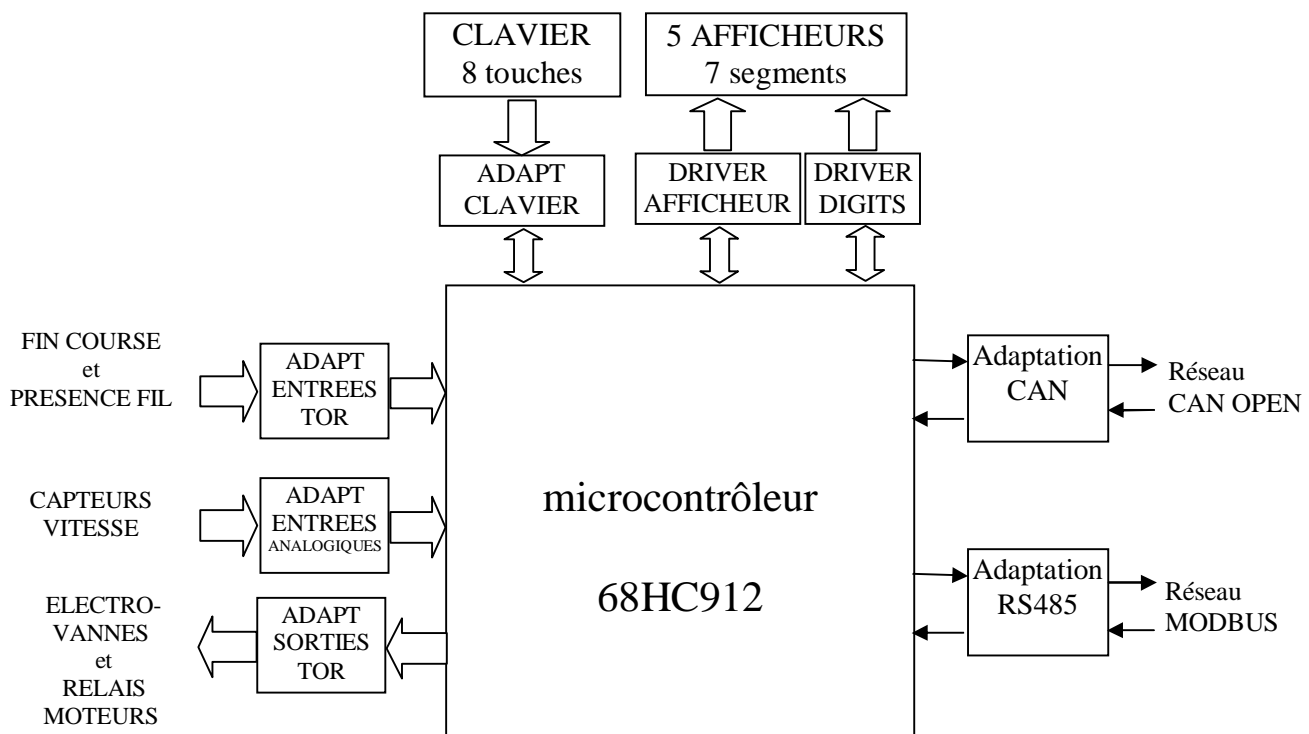
NB : dans le calcul du temps nécessaire à un échange, seules les durées de « Début de trame » sont à prendre en compte, les « Fin de trame » se chevauchant avec les temps d'attente nécessaires au maître ou à l'esclave ou avec le « Début de trame » de l'échange suivant.

### 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 exécutables).
- 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.

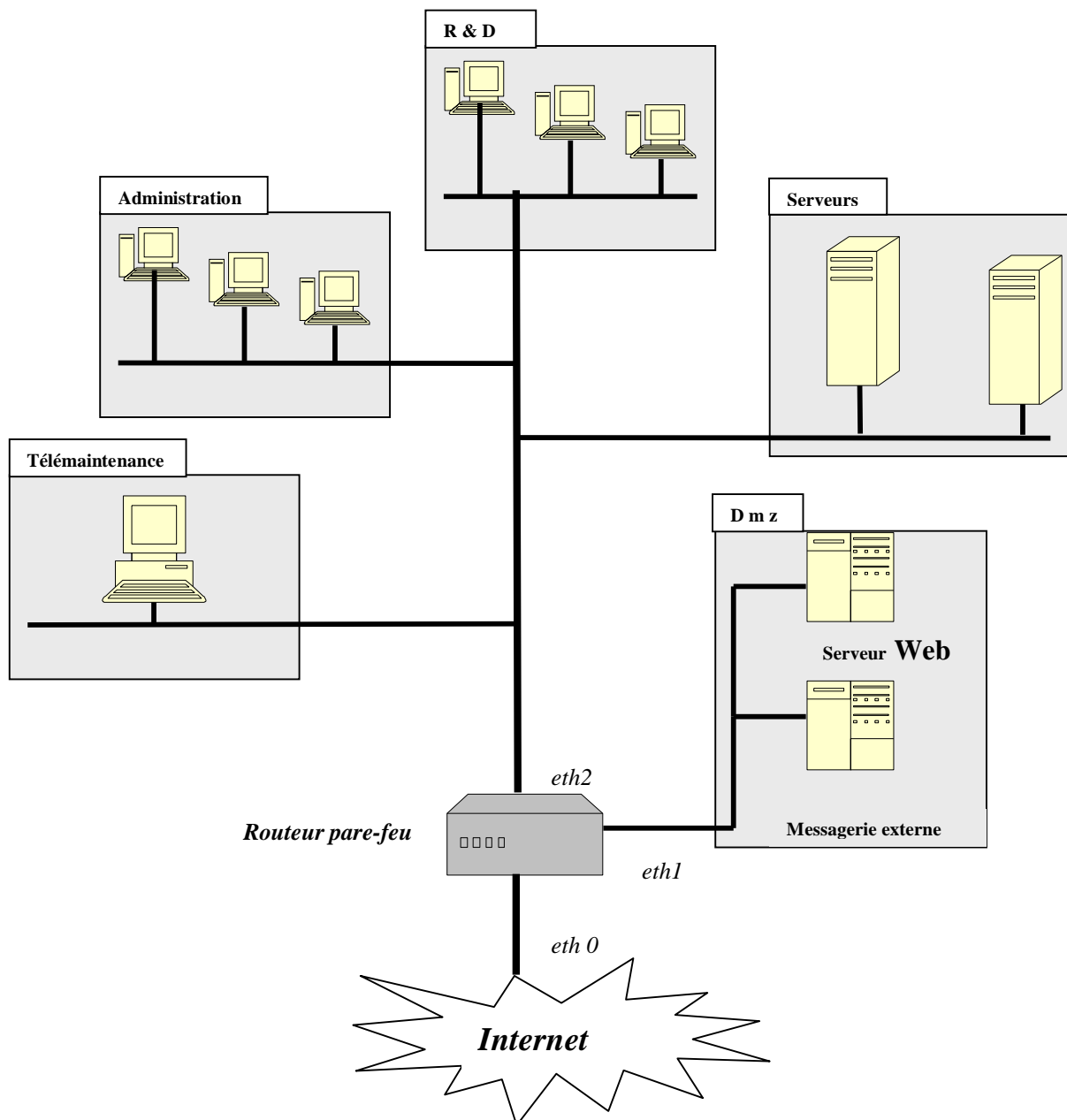
#### **Question E.1.4**

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 ?

## Annexe 1

### Fichiers de réglages machine textile

Code	Description	*sCode	*sNom
<b>Métrage</b>	valeur à produire	*metrage_lMetProg	km.
<b>Reception</b>	vitesse d'appel du fil	*reception_wVitesse	m/min
	tempo accélération	*reception_wTempoAcc	sec.
	tempo décélération	*reception_wTempoDec	sec.
	temps d'accélération	*reception_wTpsAcc	sec.
	temps de décélération	*reception_wTpsDec	sec.
<b>Broches</b>	vitesse	*broche_wVitesse	trs/min
	sens de torsion	*wSensRotation	S/Z
	torsion = (vitesse broche *2)/vitesse reception	*wTorsion	
<b>Guide-fil</b>	course	*croisure_wCourseGuideFil	mm
	croisure	*guideFil_wAngle	°
	vitesse	*guideFil_wVitesse	m/min

<?xml version="1.0" ?>

<all>

</param>

<param>

<data>metrage\_lMetProg</data>

<value>15</value>

<type>float</type>

<min>0</min>

<max>45</max>

<ratio>1</ratio>

<mode>W</mode>

</param>

<param>

<data>broche\_wVitesse</data>

<value>4500</value>

<type>float</type>

<min>1000</min>

<max>8000</max>

<ratio>1</ratio>

<mode>W</mode>

</param>

<param>

<data>reception\_wVitesse</data>

<value>50</value>

<type>float</type>

<min>50</min>

<max>200</max>

<ratio>1</ratio>

<mode>W</mode>

</param>

<param>

<data>reception\_wTempoAcc</data>

<value>10</value>

<type>float</type>

<min>0</min>

<max>60</max>

<ratio>1</ratio>

<mode>W</mode>

</param>

<param>

<data>reception\_wTempoDec</data>

<value>10</value>

<type>float</type>

<min>0</min>

<max>60</max>

<ratio>1</ratio>

<mode>W</mode>

</param>

<param>

<data>reception\_wTpsAcc</data>

<value>10</value>

<type>float</type>

<min>0</min>

<max>60</max>

<ratio>1</ratio>

<mode>W</mode>

</param>

<param>

<data>reception\_wTpsDec</data>

<value>10</value>

<type>float</type>

<min>0</min>

<max>60</max>

<ratio>1</ratio>

<mode>W</mode>

</param>

<param>

<data>croisure\_wCourseGuideFil</data>

<value>250</value>

<type>int</type>

<min>0</min>

<max>250</max>

<ratio>1</ratio>

<mode>R</mode>

</param>

<param>

<data>wSensRotation</data>

<value>0</value>

<type>int</type>

<min>0</min>

<max>1</max>

<ratio>1</ratio>

<mode>W</mode>

</param>

<param>

<data>wTorsion</data>

<value></value>

<type>float</type>

<min>1</min>

<max>500</max>

<ratio>1</ratio>

<mode>R</mode>

</param>

</all>

## Annexe 2

### Titrages des fils

**Le titrage est la mesure utilisée pour déterminer la finesse d'un fil.**

L'unité normalisée est le Tex ou le Décitex (masse par unité de longueur).

Sont encore utilisés le Deniers (pour la soie), le Numéro métrique (pour les filés de fibres) ou encore le Numéro anglais (pour le coton, le lin ou la laine).

#### **Nm (numéro métrique) :**

**C'est l'unité utilisée pour le titrage des fils discontinus (fibres).**

Le numéro métrique ou Nm est exprimé par le nombre de mètres au gramme ou de kilomètre au kilogramme.

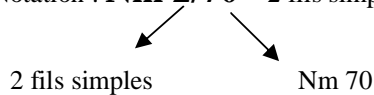
Exemple: = 70 mètres de ce fil pèsent 1 gramme

**Formule:  $Nm = L/P = \text{Longueur/poids}$**

#### **Notation spécifique pour les fils retors:**

L'assemblage par retordage de fils d'un même numéro: le Nm sera celui des fils simples réunis.

Notation : **Nm 2/70** = 2 fils simples de 1/70 assemblés donnent **Nm 1/35**



**Nm 4/80** = 4 fils simples de 1/80 assemblés donnent **Nm 1/20**

#### **Tex :**

C'est l'unité normalisée internationale depuis 1956

Unité de titrage des fils, c'est le **poids en grammes de 1000 m de fil**

Le sous-multiple décitex (dtx) est plus couramment employé. C'est le poids en grammes de 10000 m de fil

Exemple: 20 dtx = 10000 mètres de ce fil pèsent 20 grammes

**20 tex** = 1000 mètres de ce fil pèsent **20 grammes**

## Annexe 3

### Contrôle de la présence de fils

**FC FIL CONTROL**  
GROUPE ELSIDEV  
Electronique électromécanique  
Service  
Industrialisation  
Développement



# MCD-F

## Gamme des détecteurs capacitifs

EL CONTROL DEV-3

#### Application

Le **MCD-F** est un **détecteur capacitif sans contact**.

**FONCTION PRINCIPALE :** Contrôle de la présence de fils (et mèches) en mouvement rectiligne et de fils en ballon.

Dès que la mèche (ou le fil) casse, le MCD-F avertit l'utilisateur que la position est en défaut au moyen d'une LED clignotante et / ou d'un signal électrique.

Tous fils (ou mèches) de matériaux pouvant stocker des charges électrostatiques sont contrôlables par le **MCD-F**.

**SECURITE D'EMPLOI :** Le **MCD-F** est protégé contre les inversions de branchement et les courts-circuits sur sa sortie. Son niveau de protection électromagnétique (CEM) l'isole de son environnement (>4kV).

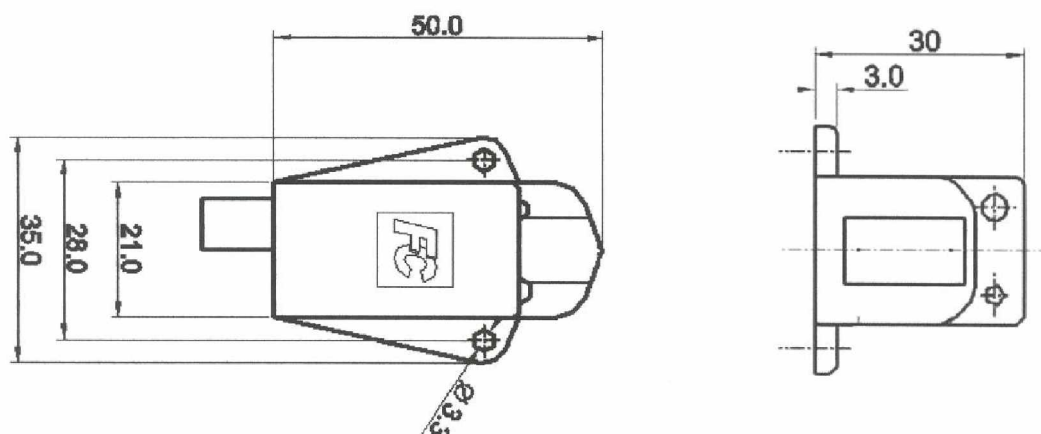


#### Caractéristiques :

- ☐ Alimentation : 18 à 30 V DC
- ☐ 1 sortie PNP
- ☐ Alarme visible (LED)
- ☐ Sortie protégée contre les courts-circuits
- ☐ Connexion par câble ou connecteur Lumberg 2,5 MSFW 5
- ☐ Sensibilité du détecteur ajustable selon le besoin de l'utilisateur

Ces caractéristiques sont adaptables en fonction des besoins de l'utilisateur (voir le tableau de codification).

#### Encombrement (mm)



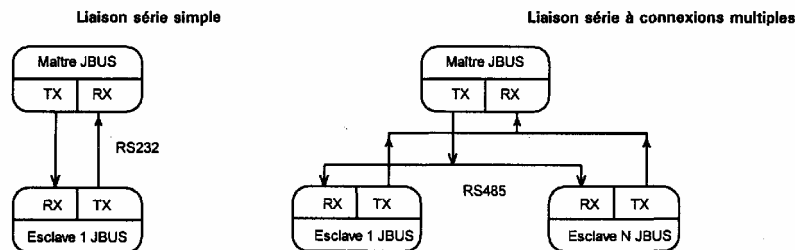
## Annexe 4

### Extraits de la documentation ModBus

## PRINCIPES DE BASE DES PROTOCOLES MODBUS ET JBUS

Un protocole de communication de données définit les règles et la structure des messages utilisés par l'ensemble des périphériques d'un réseau pour l'échange de données. Ce protocole définit également l'échange correct des messages et la détection des erreurs.

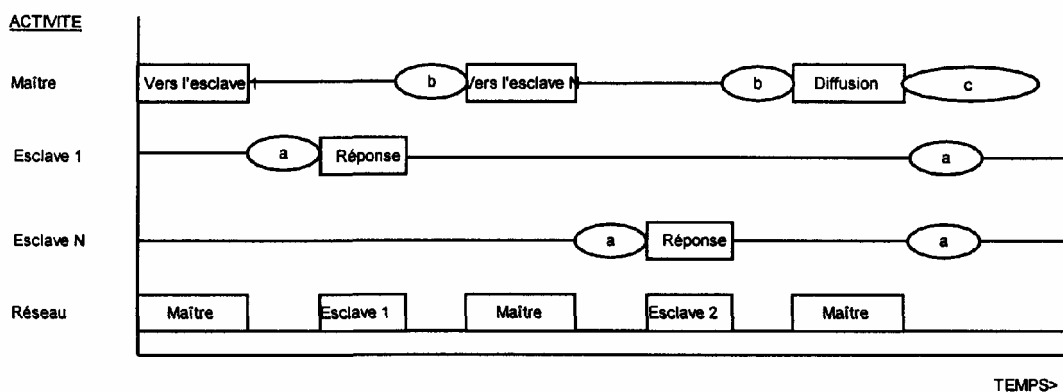
MODBUS définit un réseau de communication logique ayant un seul MAITRE et un ou plusieurs ESCLAVES. Un réseau simple ou à connexions multiples est possible. Les deux types de réseaux de communications sont illustrés sur le schéma ci-dessous.



- Une transaction-type est composée d'une demande envoyée par le maître, suivie d'une réponse de l'esclave.
- Chaque esclave possède une adresse de périphérique unique sur 8 bits.
- L'envoi d'une communication avec l'adresse de périphérique '0' est une communication de diffusion qui envoie des informations à tous les périphériques du réseau. Chacun effectue l'action demandée mais sans envoyer de réponse.
- L'envoi d'une communication avec une adresse de périphérique unique déclenche une réponse du seul périphérique possédant cette adresse. Ce périphérique recherche les erreurs, effectue la tâche demandée puis répond en donnant sa propre adresse, les données et un total de contrôle.

## ACTIVITE-TYPE D'UNE LIGNE DE TRANSMISSION

Cette figure illustre la suite-type des événements sur une ligne de transmission Modbus.



- Période « a » : temps de traitement nécessaire à l'esclave pour exécuter la commande et élaborer une réponse.  
 Période « b » : temps de traitement nécessaire au maître pour analyser la réponse de l'esclave et formuler la commande  
 Période « c » : temps d'attente calculé par le maître pour que les esclaves exécutent l'opération

Pour les fonctions d'échange de paramètres (écriture ou lecture de n bits ou de n mots), le temps d'attente est de 20 millisecondes par paramètre transmis (125 mots maxi).

## SYNTAXE DE LA TRAME DU MESSAGE

Le schéma qui suit montre l'ordre définissant la syntaxe de la trame du message utilisée par JBUS et MODBUS :

Début de la trame	Adresse du périphérique	Code fonction	Données accompagnant la fonction	Données de détection d'erreur	Fin de la trame
	1 octet	1 octet	n octets	2 octets	

Le début de la trame est une période d'inactivité égale à au moins 3,5 fois la durée de transmission d'un caractère unique. Par exemple, à 9600 bauds, un caractère comportant un bit de départ, un bit d'arrêt et 8 bits de données a besoin d'un début de trame de 3,5 millisecondes. Cette période est la fin de transmission implicite d'une transmission antérieure.

- L'adresse du périphérique est un seul octet (8 bits) propre à chaque périphérique du réseau.
- Les codes fonction sont une instruction à un seul octet destinée à l'esclave et décrivant l'action à exécuter.
  - Le segment de données d'un message dépend du code fonction et le nombre d'octets varie en conséquence.
  - En règle générale, le segment de données contient une adresse de paramètres et le nombre de paramètres à lire ou écrire.
  - Le contrôle de redondance cyclique est un code de détection d'erreur qui a une longueur de 2 octets.
  - Le segment Fin de la transmission est une période d'inactivité égale à 3,5 fois la durée de transmission d'un seul caractère. Le segment Fin de la transmission à la fin d'un message indique à l'appareil récepteur que la transmission suivante sera un nouveau message et par conséquent un caractère d'adresse de périphérique.

## CONTROLE DE REDONDANCE CYCLIQUE

Le contrôle de redondance cyclique (CRC) est un code de détection d'erreur qui a une longueur de deux octets (16 bits). Après construction d'un message (données uniquement, absence de bits de départ, d'arrêt et de parité), l'appareil émetteur calcule un code CRC et l'ajoute à la fin du message. Un appareil récepteur calcule un code CRC à partir du message qu'il a reçu. Si ce code CRC est différent du code émis, il y a eu une erreur de communication. Traduction de l'organigramme en pseudo-code :

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

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



Le symbole  $\oplus$  indique une opération 'OU exclusif'.

## Extrait des fonctions de ModBus : ECRITURE DE N MOTS

Code fonction : **16, (10h)**

Commande :

Adresse du périphérique	Code fonction 10	Adresse du premier mot		Nombre de mots à écrire		Nombre d'octets de données (n)	Données	CRC	
1 octet	1 octet	octet MSB	octet LSB	octet MSB	octet LSB	1 octet	n octets	octet MSB	octet LSB

Le nombre maximal de mots qui peuvent être transmis est de 125 mots, ce qui correspond à 250 octets de données.

Les deux premiers octets sont des données qui comportent la valeur exigée du premier paramètre, le bit de poids fort étant le premier. Les paires suivantes d'octets sont des données pour les adresses de paramètres consécutives.

Réponse :

Adresse du périphérique	Code fonction 10	Adresse du premier mot		Nombre de mots écrits		CRC	
1 octet	1 octet	octet MSB	octet LSB	octet MSB	octet LSB	octet MSB	octet LSB

### Exemple :

**Commande :** Ecrire dans l'esclave situé à l'adresse 2.

Consigne 1 = 379 à l'adresse 164

Consigne 2 = 918 à l'adresse 165

Consigne 3 = 250 à l'adresse 166

Adresse du périphérique	Code fonction	Adresse du premier mot		Nombre de mots à écrire		Nombre d'octets de données	Données	CRC
02	10	00	A4	00	03	06	Cf. ci-dessous	20 71

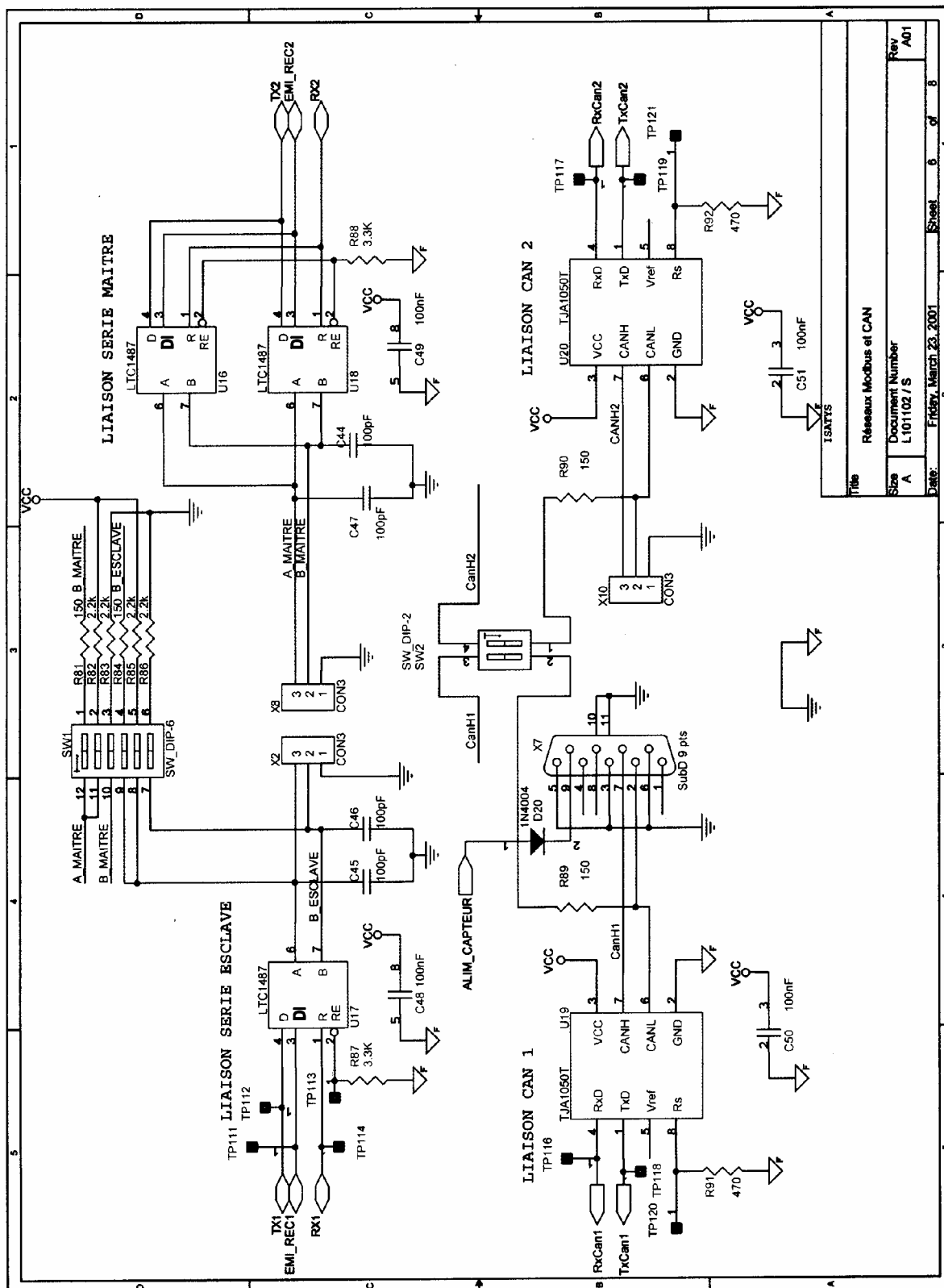
Données (379) pour l'adresse 164		Données (918) pour l'adresse 165		Données (250) pour l'adresse 166	
01	7B	03	96	00	FA

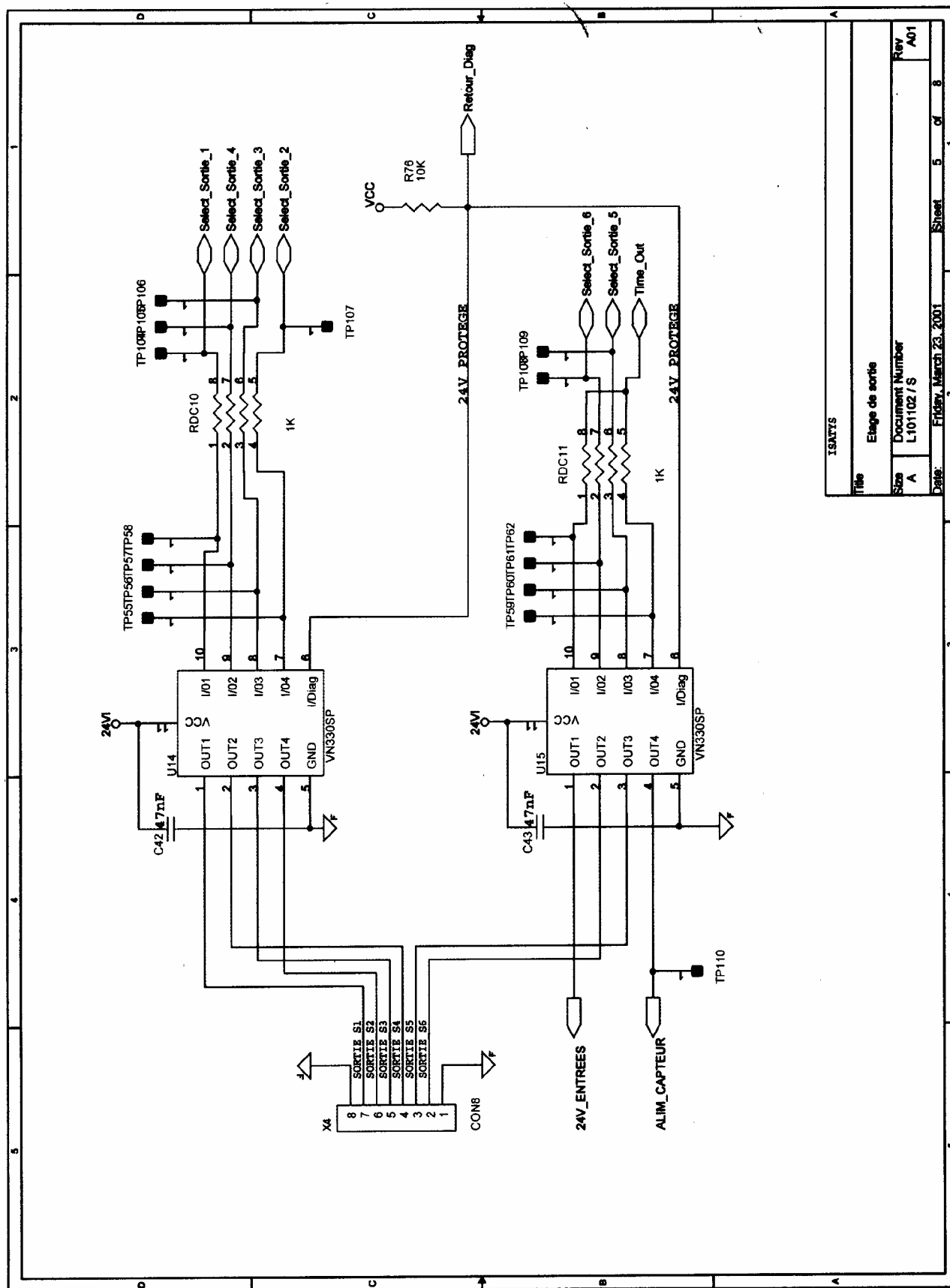
**Réponse :**

Adresse du périphérique	Code fonction	Adresse du premier mot		Nombre de mots écrits		CRC	
02	10	00	A4	00	03	C1	D8

# Annexe 5

## Extraits des schémas de la carte ISATYS





ISAT7S	
Etage de sortie	
Size	Document Number
A	L101102 / S
Date:	Friday, March 23, 2001
Sheet	5 of 8

## Annexe 6

### Extraits de la documentation Linear Technologie



LTC1487

Ultra-Low Power RS485  
with Low EMI, Shutdown  
and High Input Impedance

### FEATURES

- **High Input Impedance: Up to 256 Transceivers on the Bus**
  - **Low Power:  $I_{CC} = 120\mu A$  Max with Driver Disabled**
  - $I_{CC} = 200\mu A$  Max with Driver Enabled, No Load
  - **$1\mu A$  Quiescent Current in Shutdown Mode**
  - Controlled Slew Rate Driver for Reduced EMI
  - Single 5V Supply
  - ESD Protection to  $\pm 10kV$  On Receiver Inputs and Driver outputs
  - $-7V$  to  $12V$  Common-Mode Range Permits  $\pm 7V$  Ground Difference Between Devices on the Data Line
  - Thermal Shutdown Protection
  - Power Up/Down Glitch-Free Driver Outputs Permit Live Insertion or Removal of Transceiver
  - Driver Maintains High Impedance in Three-State or with the Power Off
  - Pin Compatible with the LTC485
- 
- Battery-Powered RS485/RS422 Applications
  - Low Power RS485/RS422 Transceiver
  - Level Translator

### ES R T N

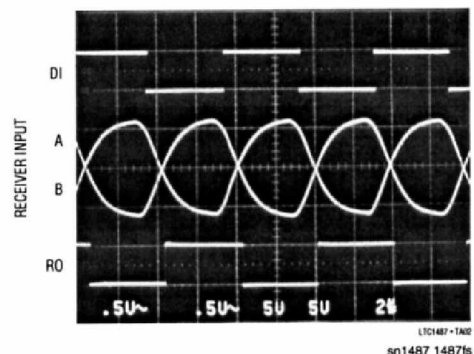
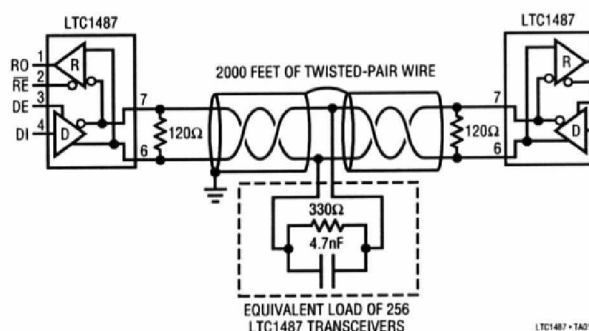
The LTC<sup>®</sup>1487 is an ultra-low power differential line transceiver designed with high impedance inputs allowing up to 256 transceivers to share a single bus. It meets the requirements of RS485 and RS422. The LTC1487 features output drivers with controlled slew rate, decreasing the EMI radiated from the RS485 lines, and improving signal fidelity with misterminated lines. The CMOS design offers significant power savings without sacrificing ruggedness against overload or ESD damage. Typical quiescent current is only  $80\mu A$  while operating and  $1\mu A$  in shutdown.

The driver and receiver feature three-state outputs, with the driver outputs maintaining high impedance over the entire common-mode range. Excessive power dissipation caused by bus contention or faults is prevented by a thermal shutdown circuit which forces the driver outputs into a high impedance state. The receiver has a fail-safe feature which guarantees a high output state when the inputs are left open. I/O pins are protected against multiple ESD strikes of over  $\pm 10kV$  using the Human Body Model.

The LTC1487 is fully specified over the commercial temperature range and is available in 8-pin DIP and SO packages.

LT, LTC and LT are registered trademarks of Linear Technology Corporation.

### TYPICAL APPLICATION



1

## PIN FUNCTIONS

**RO (Pin 1):** Receiver Output. If the receiver output is enabled ( $\overline{RE}$  LOW), and  $A > B$  by 200mV, RO will be HIGH. If  $A < B$  by 200mV, then RO will be LOW.

**$\overline{RE}$  (Pin 2):** Receiver Output Enable. A LOW enables the receiver output, RO. A HIGH input forces the receiver output into a high impedance state.

**DE (Pin 3):** Driver Outputs Enable. A HIGH on DE enables the driver output. A and B and the chip will function as a line driver. A LOW input will force the driver outputs into a high impedance state and the chip will function as a line receiver. If  $\overline{RE}$  is HIGH and DE is LOW, the part will enter a low power (1 $\mu$ A) shutdown state.

**DI (Pin 4):** Driver Input. If the driver outputs are enabled (DE HIGH) then a LOW on DI forces the outputs A LOW and B HIGH. A HIGH on DI with the driver outputs enabled will force A HIGH and B LOW.

**GND (Pin 5):** Ground.

**A (Pin 6):** Driver Output/Receiver Input.

**B (Pin 7):** Driver Output/Receiver Input.

**V<sub>CC</sub> (Pin 8):** Positive Supply.  $4.75V < V_{CC} < 5.25V$ .

## FUNCTION TABLES

LTC1487 Transmitting

INPUTS			OUTPUTS	
$\overline{RE}$	DE	DI	B	A
X	1	1	0	1
X	1	0	1	0
0	0	X	Z	Z
1	0	X	Z*	Z*

\* Shutdown mode

LTC1487 Receiving

INPUTS			OUTPUTS
$\overline{RE}$	DE	A - B	RO
0	0	$\geq 0.2V$	1
0	0	$\leq -0.2V$	0
0	0	Inputs Open	1
1	0	X	Z*

\* Shutdown mode

## TEST CIRCUITS

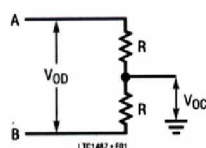


Figure 1. Driver DC Test Load

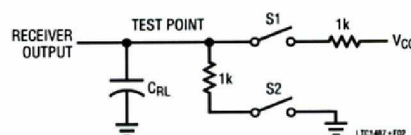


Figure 2. Receiver Timing Test Load

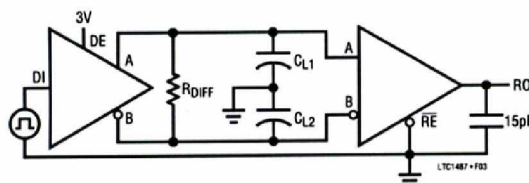


Figure 3. Driver/Receiver Timing Test Circuit

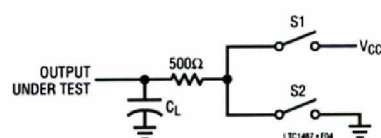


Figure 4. Driver Timing Test Load

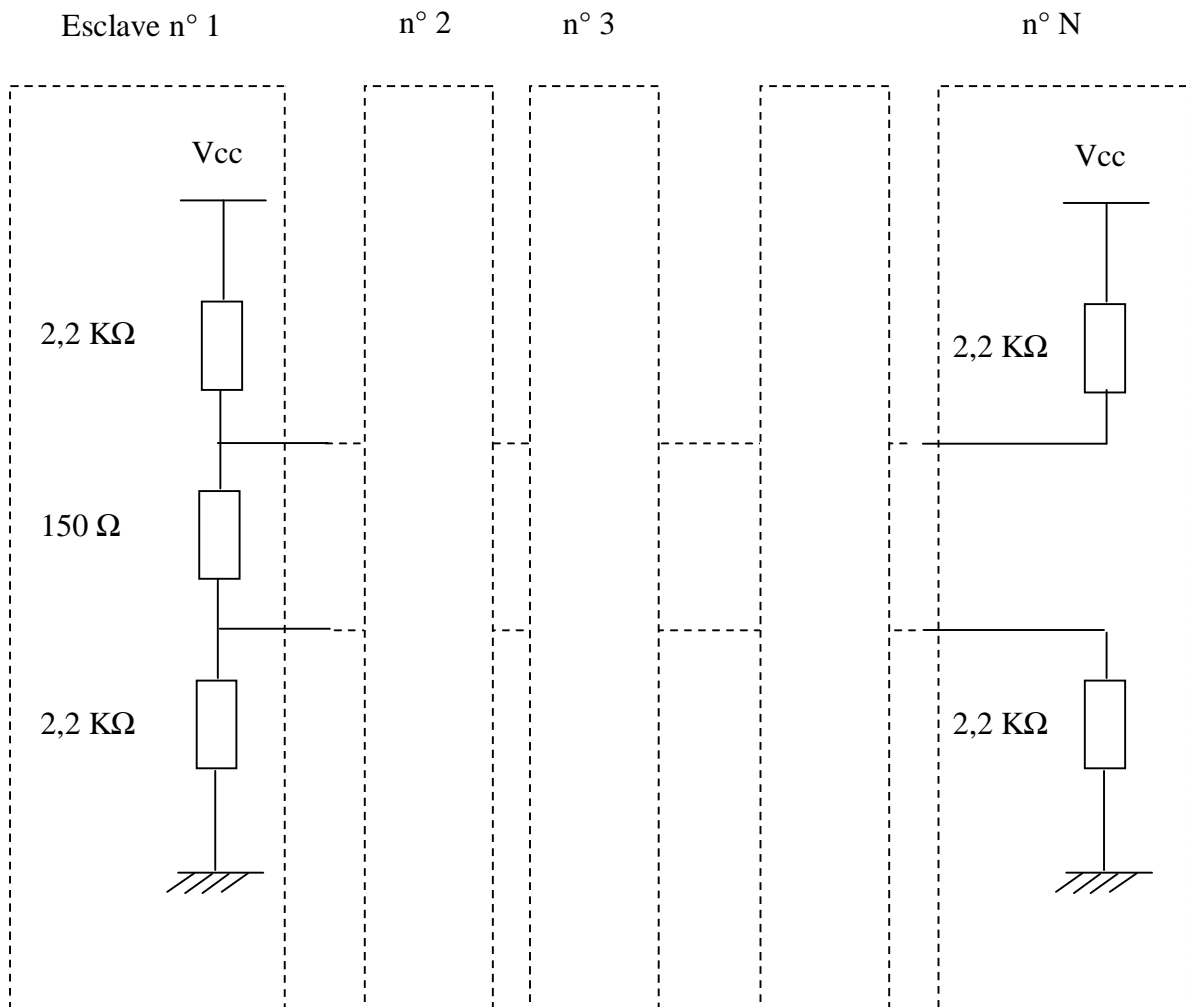
sn1487 14871s



## Annexe 7

### Note d'application

Note d'application fournie par LINEAR TECHNOLOGIE utilisée pour configuration du ModBus (réduction du bruit sur les lignes) :



## Annexe 8

### Extraits de la documentation STMicroelectronics



## VN330SP

### QUAD HIGH SIDE SMART POWER SOLID STATE RELAY

TYPE	V <sub>demag</sub> (*)	R <sub>DSon</sub> (*)	I <sub>out</sub> (*)	V <sub>CC</sub>
VN330SP	V <sub>CC</sub> -55V	0.32Ω (**)	0.7A	36 V

(\*) Per channel  
(\*\*) at T<sub>J</sub>=85°C

- OUTPUT CURRENT: 0.7A PER CHANNEL
- DIGITAL INPUTS CLAMPED AT 32V MINIMUM VOLTAGE
- SHORTED LOAD AND OVERTEMPERATURE PROTECTIONS
- BUILT-IN CURRENT LIMITER
- UNDERVOLTAGE SHUT-DOWN
- OPEN DRAIN DIAGNOSTIC OUTPUT
- FAST DEMAGNETIZATION OF INDUCTIVE LOADS

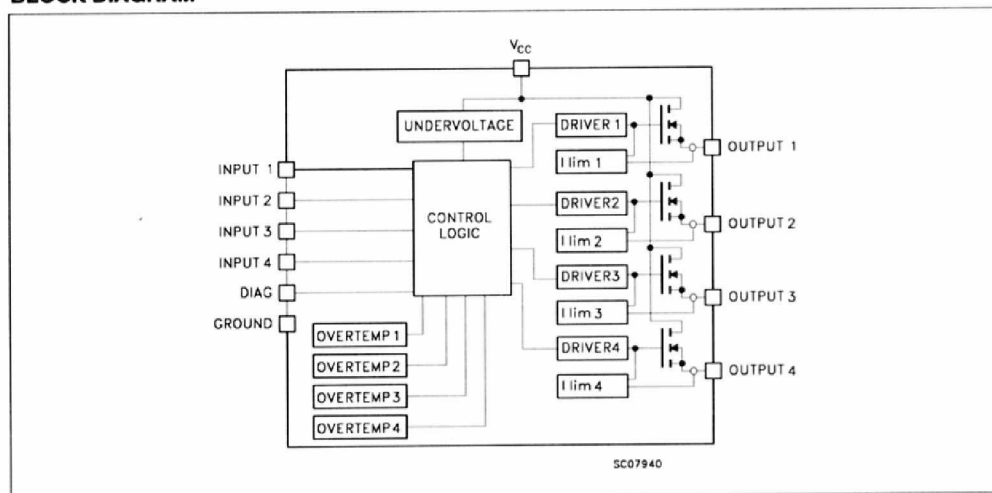


#### DESCRIPTION

The VN330SP is a monolithic device made using STMicroelectronics VIPower technology, intended for driving four independent resistive or inductive loads with one side connected to ground. Active current limitation avoids dropping the system power supply in case of shorted load. Built-in thermal shut-down protects the chip from

overtemperature and short circuit. The open drain diagnostic output indicates short circuit and overtemperature conditions.

#### BLOCK DIAGRAM



August 2000

1/9

**ELECTRICAL CHARACTERISTICS** (continued)**LOGIC INPUT** (Per each channel)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
$V_{IL}$	Input low level voltage				2	V
$V_{IH}$	Input high level voltage (see note 1)		3.5			V
$V_{I(HYST)}$	Input hysteresis voltage			0.5		V
$I_{IN}$	Input current	$V_{IN} = 0$ to 30V $V_{IN} = 0$ to 2V	25		600	$\mu A$ $\mu A$
$I_{LGND}$	Output current in ground disconnection	$V_{CC} = V_{INn} = GND = DIAG = 24V$ ; $T_J = 25^\circ C$			25	mA
$V_{ICL}$	Input clamp voltage (see note 1)	$I_{IN} = 1mA$ $I_{IN} = -1mA$	32	36 -0.7		V V

Note 1: The input voltage is internally clamped at 32V minimum, it is possible to connect the input pins to an higher voltage via an external resistor calculate to not exceed 10mA.

**PROTECTION AND DIAGNOSTICS**

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
$V_{DIAG} (*)$	Status voltage output low	$I_{STAT} = 5mA$ (Fault condition)			1	V
$V_{SCL} (*)$	Status clamp voltage	$I_{DIAG} = 1mA$ $I_{DIAG} = -1mA$	32	36 -0.7		V V
$V_{USD}$	Undervoltage shut down		5		8	V
$V_{OL}$	Low state output voltage	$V_{IN} = V_{ILj}$ ; $R_{LOAD} \geq 10M\Omega$			1.5	V
$I_{LIM}$	DC Short circuit current	$V_{CC} = 24V$ ; $R_{LOAD} < 10m\Omega$	0.7		2.5	A
$I_{OVPK}$	Peak short circuit current	$V_{CC} = 24V$ ; $V_{IN} = 30V$ ; $R_{LOAD} < 10m\Omega$ (see figure 2)			4	A
$I_{DIAGH}$	Leakage on diag pin in high state	$V_{DIAG} = 24V$			100	$\mu A$
$I_{LOAD}$	Output leakage current	$V_{CC} = 10$ to 36V; $V_{IN} = V_{IL}$			50	$\mu A$
$t_{SC}$	Delay time of current limiter				100	$\mu s$
$T_{TSD}$	Thermal shut down temperature		150	170		$^\circ C$
$T_R$	Reset temperature		135	155		$^\circ C$

(\*) Status determination > 100 $\mu s$  after the switching edge.

Note: If INPUT pin is left floating the corresponding channel will automatically switch off. If GND pin is disconnected, all channels will switch off provided  $V_{CC}$  does not exceed 36V.





## Annexe 9

### Primitives UNIX

#### Extrait du Manuel du programmeur Linux (2) et (3)

### **FORK**

NOM

fork - Créer un processus fils.

SYNOPSIS

```
#include <sys/types.h>
#include <unistd.h>
```

```
pid_t fork(void);
```

DESCRIPTION

fork crée un processus fils qui diffère du processus parent uniquement par ses valeurs PID et PPID et par le fait que toutes les statistiques d'utilisation des ressources sont remises à zéro. Les verrouillages de fichiers, et les signaux en attente ne sont pas hérités.

VALEUR RENVOYÉE

En cas de succès, le PID du fils est renvoyé au processus parent, et 0 est renvoyé au processus fils. En cas d'échec -1 est renvoyé dans le contexte du parent, aucun processus fils n'est créé, et errno contient le code d'erreur.

---

### **EXECL**

NOM

execl - Exécuter un programme.

SYNOPSIS

```
#include <unistd.h>
```

```
int execl (const char *path, const char *arg, ...);
```

DESCRIPTION

La fonction execl remplace l'image mémoire du processus en cours par un nouveau processus. L'argument initial est le chemin d'accès du fichier à exécuter.

Les arguments const char \*arg ainsi que les points de suspension peuvent être vues comme des arg0, arg1, ..., argn. Ensemble ils décrivent une liste d'un ou plusieurs pointeurs sur des chaînes de caractères terminées par des caractères nuls, qui constituent les arguments disponibles pour le programme à exécuter. Par convention le premier argument doit pointer sur le nom du fichier associé au programme à exécuter. La liste des arguments doit se terminer par un pointeur NULL.

---

## SHMGET

NOM

shmget - Allouer un segment de mémoire partagée.

SYNOPSIS

```
#include <sys/ipc.h>
```

```
#include <sys/shm.h>
```

```
int shmget(key_t clé, int size, int shmflg);
```

DESCRIPTION

shmget() renvoie l'identificateur du segment de mémoire partagée associé à la valeur de l'argument clé. Un nouveau segment mémoire, de taille size arrondie au multiple supérieur de PAGE\_SIZE, est créé.

shmflg est composé de IPC\_CREAT pour créer un nouveau segment et du mode d'accès (les 9 bits de poids faibles) indiquant les permissions pour le propriétaire, le groupe et les autres. Actuellement la permission d'exécution n'est pas utilisée par le système.

---

---

## SHMAT

NOM

shmat - Opérations sur la mémoire partagée.

SYNOPSIS

```
#include <sys/types.h>
```

```
#include <sys/shm.h>
```

```
void *shmat ( int shmid, const void * shmaddr, int shmflg);
```

DESCRIPTION

La fonction shmat attache le segment de mémoire partagée identifié par shmid au segment de données du processus appelant. L'adresse d'attachement est indiquée par shmaddr avec comme critère :

Si shmaddr vaut NULL, le système essaye de trouver une adresse libre pour attacher le segment.

Utiliser shmat avec shmaddr égale à NULL est la manière conseillée, portable, d'attacher un segment de mémoire partagée.

La valeur courante de shmflg est 0.

---