

BTS INFORMATIQUE INDUSTRIELLE

Session 2001

Epreuve Etude d'un Système Informatisé

Système d'acquisition de température de bobinages

Sujet – deuxième partie : conception d'éléments du système

Durée : 4h30 Coefficient 3,5/5

*"Calculatrice autorisée (conformément à la circulaire n° 99-186 du 16 novembre 1999)."
"Le candidat peut utiliser tous les documents personnels qu'il estime nécessaire."*

Ce document comprend 12 pages composées de :

Sujet : pages 1 à 10 (papier blanc)

Documents réponses : pages 11 à 12 (papier de couleur)

à rendre obligatoirement (même vierge) avec la copie

Toutes vos réponses doivent être accompagnées d'une justification.

D. Architecture matérielle

D.1 Etude des caractéristiques du bus IEEE 488

Une documentation partielle du bus est fournie en annexe D4.

Question D.1.1 – Répondez directement sur le document Réponse.

Complétez le tableau de configuration donnant les fonctions et les adresses de chacun des appareils.
(de nombreuses réponses sont possibles pour les adresses)

	Rack de commutation	Multimètre 1	Multimètre 2	Multimètre 3	Multimètre 4
Fonction	Ecouteur ou parleur				
Adresse	7				
Code Ecouteur	\$27				
Code Parleur	\$47				

Nota : On supposera le bit de parité toujours à 0

La chaîne de caractères *IDN? est envoyée sur le bus à partir du rack de commutation vers l'un des multimètres.

Question D.1.2

Quelles sont les valeurs binaires présentes sur les lignes de données au moment de l'envoi du caractère * (code ASCII 42 décimal) ?

Question D.1.3

Quelle sont les informations contenues dans la chaîne de caractères renvoyée par le multimètre concerné ?

D.2 Adressage des cartes

L'organisation de l'espace adressable est fixée par la carte CPU BAB40. L'annexe S1 donne une documentation partielle de la carte BAB40.

Question D.2.1

Quelle est dans cet espace la plage d'adresses disponible pour le bus VME (première et dernière adresse) en accès court (données sur 8 ou 16 bits) et la capacité (nombre d'octets) correspondante ?

La carte d'extension VMOD-IO est connectée sur le bus VME (voir Figure 8 du dossier technique).

Un extrait de la documentation de la carte VMOD-IO est fourni en Annexe S2.

Le Jumper J4 est fixé en position 2-3. Les commutateurs S2 et S1 sont positionnés respectivement à C et 8.

Question D.2.2

Quelles sont les lignes d'adresse (A15, A14...) fixées par les commutateurs S2 et S1 ?

Question D.2.3

Quelle est la plage d'adresses relatives utilisées par la carte VMOD-IO et la capacité correspondante ?

Question D.2.4

Quel est l'espace d'adresse absolu utilisé par la carte VMOD-IO dans le plan mémoire fixé par la carte CPU BAB 40 ?

Le module VMOD GPIB intègre un circuit contrôleur de bus IEEE 488 (bus GPIB) μ PD7210 dont une documentation partielle est donnée en annexe S4.

Question D.2.5

Quelle est l'espace d'adresse absolu utilisé par les 16 registres internes du contrôleur μ PD7210 ?

D.3 Etude du module VMOD-RL

Les 12 sorties du module VMOD-TTL/O sont reliées aux 2 modules VMOD-RL (6 sorties par module) de la carte d'extension VMOD-IO (voir Figure 8 du dossier technique).

Ces 12 sorties correspondent aux 8 lignes du Port B et aux 4 lignes du port C du CIO Z8536 intégré au module VMOD-TTL/O et sont accessibles par un connecteur 20 broches (voir page 3 de

l'annexe S3). Elle permettent de programmer les commutations du 230V, des bobines du four, de la sonde de température et du multimètre connecté (voir Figure 9 du dossier technique).

Le schéma structurel du module VMOD-RL est donné annexe S6.

Question D.3.1

Comment doit-on configurer les cavaliers J1 à J6 pour que le module VMOD-RL soit relié au port B du CIO Z8536 ?

Question D.3.2

Quel est le rôle du circuit ULN2003 (Documentation en Annexe S7)?

Question D.3.3

Quel niveau logique doit être présent en sortie du circuit ULN2003 pour alimenter les bobines des relais.

Question D.3.4

Quel est le niveau logique présent en entrée des circuits 74LS04 en cas de déconnexion de la carte ?

Dans quel état sont alors les relais ?

Quel est le rôle des circuits 74LS04 ?

Question D.3.5

Sachant que les commutateurs sont représentés au repos, quelle valeur doit t-on écrire sur le port B pour commuter la bobine 3 du four sur le multimètre ?

E. Programmation

E-1 Utilisation du matériel IEEE

Les questions qui suivent s'intéressent à la manipulation de la carte VMOD-GPIB. Ce module enfichable conforme à la norme IEEE 488-2 permet le pilotage des multimètres qui mesurent les résistances des bobines, et effectue la récupération des données de ces multimètres.

Lors de la mise en route de l'application, on devra initialiser le driver et créer le périphérique (device) correspondant au module GPIB (voir Annexe S5). Une fois ces opérations effectuées, l'application, pour accéder au module GPIB doit obtenir un descripteur de fichier de niveau 1, en utilisant la fonction `open()` de la bibliothèque standard avec comme paramètre le nom utilisé lors de la création du périphérique (device). Cette ouverture se fait en mode lecture seule, bien que les opérations d'écriture ultérieures soient autorisées.

Question E.1.1

En vous référant à la documentation fournie en annexe S8, et au schéma décrit dans le dossier technique, proposez un prototype pour la méthode `CGPIB ::Open()` qui :

- initialise le driver GPIB,
- crée le device correspondant,
- obtient le descripteur de fichier nécessaire pour la suite des opérations.

Vous tiendrez compte pour établir ce prototype des variations possibles d'implantation du module et de la carte support ?

Nota : Le vecteur d'interruption est fixé à **0x80** et le niveau d'interruption à **5**. Ces valeurs sont indicatives, et ne sont jamais utilisées dans le reste du sujet.

Question E.1.2

Proposez une solution complète en C++ pour cette méthode ?

Question E.1.3

Quels sont d'après le dossier technique et les questions D.3.1 à D.3.5 les valeurs des paramètres qu'il faudra passer à cette méthode?

Une fois le descripteur de fichier obtenu, l'envoi de la chaîne ***IDN?** vers l'adresse présumée d'un multimètre comme vu plus haut, permet de s'assurer de la présence d'un multimètre.

On souhaite pouvoir réinitialiser un multimètre et vider éventuellement la file d'attente des messages d'erreur de l'appareil.

Question E.1.4

Quelle fonction en langage C de la bibliothèque GPIB peut-on utiliser pour envoyer cette commande au multimètre ? A l'aide de cette fonction, quelle chaîne de caractères doit on envoyer au multimètre ? Quel sera l'effet de l'ajout de la commande ***OPC?** à la fin de cette chaîne ?

E-2 Analyse des résultats

Soit la structure :

```
#define NB_FASTACQUIRE 5

struct Measure
{
    long    m_Time ;
    float   m_Res[NB_FASTACQUIRE] ;
} ;
```

Cette structure reçoit les données brutes liées à une acquisition du multimètre. On souhaite ajouter à cette structure une information susceptible de recevoir la moyenne de ces mesures, en mettant de côté systématiquement la plus petite et la plus grande des 5 mesures, pour ne garder que la moyenne des trois valeurs centrales.

Question E.2.1

Proposez une modification de la structure Measure qui prend en compte cet ajout.

Question E.2.2

Proposez un codage en C++ de la fonction **float CBobine::GetCentralValue(...)** qui prend en argument une référence à une structure Measure, met à jour cette structure, et renvoie la valeur centrale.

Question E.2.3

Quel est l'intérêt de la constante **NB_FASTACQUIRE** ?

F. Communication et réseau

F.1 Liaison série RS232

La liaison RS232 est seulement utilisée en phase d'initialisation pour configurer différents paramètres réseau du rack.

Le constructeur donne les indications suivantes pour le raccordement d'un terminal ou d'un ordinateur à la carte CPU BAB40 :

- liaison sur 3 fils
- 9600 bit/s
- 8 bits de données
- parité paire
- 1 bit d'arrêt
- contrôle de flux Xon/Xoff

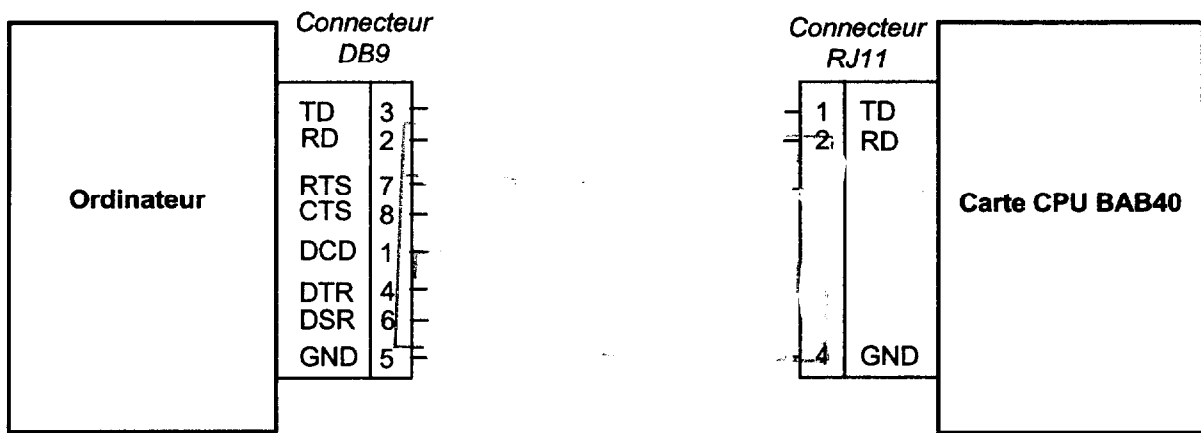
Question F.1.1

La liaison est-elle de type asynchrone ou synchrone ?

Physiquement, la liaison est réalisée à l'aide d'un câble muni d'un connecteur DB9 côté ordinateur et d'un connecteur RJ11 côté carte CPU.

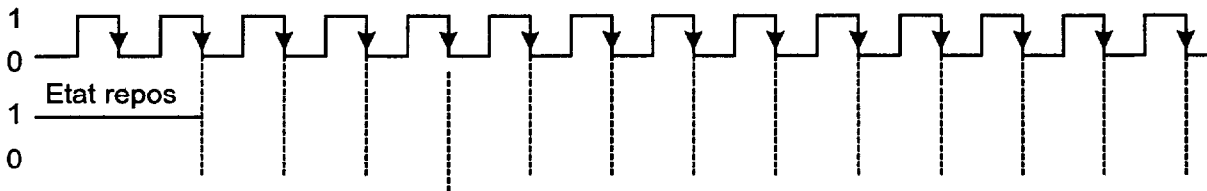
Question F.1.2 – Répondez directement sur le document Réponse.

Complétez le schéma de connexion entre l'ordinateur et la carte CPU.



Question F.1.3 – Répondez directement sur le document Réponse.

Représenter le chronogramme correspondant à l'émission du caractère 'A' (code ASCII 41H) en faisant figurer les différents bits de la trame (bit de départ, bit de données...) à partir d'un état de repos au niveau haut.



Question F.1.4

Quel est le temps minimum nécessaire pour transmettre, lors de mise sous tension du rack, la chaîne de caractères «VxWorks System Boot»?

Question F.1.5

Expliquez le principe du contrôle de flux utilisé sur cette liaison.

F.2 Organisation du réseau

Le réseau mis en œuvre est à base d'une architecture Ethernet 10baseT. On utilise le service FTP de TCP/IP pour le transfert de fichiers (voir Figure 2 du dossier technique).

Question F.2.1 – Répondez directement sur le document Réponse.

Complétez le tableau en donnant le numéro de la couche du modèle OSI concernée par les différentes entités ou protocoles présents sur le réseau utilisé.

Entité/protocole	Câble UTP	Routeur ADSL	Ethernet	Connecteur RJ45	TCP	IP	FTP
Couche(s)	1						

Le réseau possède une adresse IP de classe C : 195.212.5.0

Question F.2.2 – Répondez directement sur le document Réponse.

Proposez un plan d'adressage pour les différentes machines connectées.

	Routeur	Serveur	PC de contrôle	Rack VME
Adresse IP				
Masque				

F.3 Dialogue FTP

La configuration réalisée à l'aide de la liaison RS232 est mémorisée en RAM non volatile. Dans cette configuration les paramètres réseau du rack ont été donnés :

- adresse IP du rack
- adresse IP du serveur
- chemin et nom du fichier à télécharger
- nom et mot de passe pour la connexion FTP

La deuxième phase d'initialisation consiste à télécharger à partir du serveur le fichier correspondant au module du noyau VxWorks. Le client FTP est donc le rack qui demande automatiquement, après mise sous tension, une connexion FTP anonyme avec le serveur et le chargement du fichier VxWorks.

Les deux trames reproduites ci-dessous sont extraites d'un relevé réalisé par un analyseur de protocole lors de la séquence de démarrage à la mise sous tension du rack.

L'analyse présente, pour les trames 17 et 18, la description des différents protocoles utilisés : Ethernet, IP, TCP et FTP.

Pour chaque protocole, la première ligne donne les caractéristiques principales, les lignes suivantes les valeurs et la signification des différents champs.

La dernière partie donne pour chaque trame le contenu brut en hexadécimal suivi d'une représentation ASCII.

Network Monitor trace Mon 06/05/98 16:02:32 BootVxWoks.TXT

```
*****
Frame   Time      Src MAC Addr  Dst MAC Addr  Protocol  Description
17      10.944    00005B001BDB 0050046A8DCA  FTP       Req. from Port 1024
```

+ FRAME: Base frame properties

```
ETHERNET: ETYPE = 0x0800 : Protocol = IP: DOD Internet Protocol
+ ETHERNET: Destination address : 0050046A8DCA
+ ETHERNET: Source address : 00005B001BDB
ETHERNET: Frame Length : 96 (0x0060)
ETHERNET: Ethernet Type : 0x0800 (IP: DOD Internet Protocol)
ETHERNET: Ethernet Data: Number of data bytes remaining = 82 (0x0052)
```

IP: ID = 0x3EE; Proto = TCP; Len: 82

IP: Version = 4 (0x4)

IP: Header Length = 20 (0x14)

+ IP: Service Type = 0 (0x0)

IP: Total Length = 82 (0x52)

IP: Identification = 1006 (0x3EE)

- IP: Flags Summary = 0 (0x0)

IP:0 = Last fragment in datagram

IP:0. = May fragment datagram if necessary

IP: Fragment Offset = 0 (0x0) bytes

IP: Time to Live = 30 (0x1E)

IP: Protocol = TCP - Transmission Control

IP: Checksum = 0x174F

IP: Source Address = 195.212.5.24

IP: Destination Address = 195.212.5.1

IP: Data: Number of data bytes remaining = 62 (0x003E)

TCP: .AP..., len: 42, seq: 320061-320102, ack: 207440213, win: 4096, src: 1024, dst: 21 (FTP)

TCP: Source Port = 0x0400

TCP: Destination Port = FTP [control]

TCP: Sequence Number = 320061 (0x4E23D)

TCP: Acknowledgement Number = 207440213 (0xC5D4955)

TCP: Data Offset = 20 (0x14)

TCP: Reserved = 0 (0x0000)

+ TCP: Flags = 0x18 : .AP...

TCP: Window = 4096 (0x1000)

TCP: Checksum = 0x5528

TCP: Urgent Pointer = 0 (0x0)

TCP: Data: Number of data bytes remaining = 42 (0x002A)

FTP: Req. from Port 1024, 'RETR tornado\target\config\bab40\vxworks'

FTP: FTP Command = RETR

FTP: FTP Data: Number of data bytes remaining = 38 (0x0026)

```
00000: 00 50 04 6A 8D CA 00 00 5B 00 1B DB 08 00 45 00 .P.j....[.....E.
00010: 00 52 03 EE 00 00 1E 06 17 4F C3 D4 05 18 C3 D4 .R.....O.....
00020: 05 01 04 00 00 15 00 04 E2 3D 0C 5D 49 55 50 18 .....=.]IUP.
00030: 10 00 55 28 00 00 52 45 54 52 20 74 6F 72 6E 61 ..U(..RETR torna
00040: 64 6F 5C 74 61 72 67 65 74 5C 63 6F 6E 66 69 67 do\target\config
00050: 5C 62 61 62 34 30 5C 76 78 77 6F 72 6B 73 0D 0A \bab40\vxworks..
```

```
*****
Frame   Time      Src MAC Addr  Dst MAC Addr  Protocol  Description
18      10.945    0050046A8DCA 00005B001BDB  FTP       Resp. to Port 1024
```

+ FRAME: Base frame properties

ETHERNET: ETYPE = 0x0800 : Protocol = IP: DOD Internet Protocol

```

+ ETHERNET: Destination address : 00005B001BDB
+ ETHERNET: Source address : 0050046A8DCA
  ETHERNET: Frame Length : 150 (0x0096)
  ETHERNET: Ethernet Type : 0x0800 (IP: DOD Internet Protocol)
  ETHERNET: Ethernet Data: Number of data bytes remaining = 136 (0x0088)
IP: ID = 0x1D7E; Proto = TCP; Len: 136
  IP: Version = 4 (0x4)
  IP: Header Length = 20 (0x14) + IP: Service Type = 0 (0x0)
  IP: Total Length = 136 (0x88)
  IP: Identification = 7550 (0x1D7E)
- IP: Flags Summary = 0 (0x0)
  IP: .....0 = Last fragment in datagram
  IP: .....1. = Cannot fragment datagram
  IP: Fragment Offset = 0 (0x0) bytes
  IP: Time to Live = 128 (0x80)
  IP: Protocol = TCP - Transmission Control
  IP: Checksum = 0x5B88
  IP: Source Address = 195.212.5.1
  IP: Destination Address = 195.212.5.24
  IP: Data: Number of data bytes remaining = 116 (0x0074)
TCP: .AP..., len: 96, seq: 207440213-207440308, ack: 320103, win: 8659, src: 21 (FTP),
dst: 1024
  TCP: Source Port = FTP [control]
  TCP: Destination Port = 0x0400
  TCP: Sequence Number = 207440213 (0xC5D4955)
  TCP: Acknowledgement Number = 320103 (0x4E267)
  TCP: Data Offset = 20 (0x14)
  TCP: Reserved = 0 (0x0000)
+ TCP: Flags = 0x18 : .AP...
  TCP: Window = 8659 (0x21D3)
  TCP: Checksum = 0xEECE
  TCP: Urgent Pointer = 0 (0x0)
  TCP: Data: Number of data bytes remaining = 96 (0x0060)
FTP: Resp. to Port 1024, '150 Opening BINARY mode data connection...'
  FTP: FTP Error Return Code = 150
  FTP: FTP Command Arg1 = Opening
  FTP: FTP Data: Number of data bytes remaining = 85 (0x0055)

00000: 00 00 5B 00 1B DB 00 50 04 6A 8D CA 08 00 45 00  ..[....P.j....E.
00010: 00 88 1D 7E 40 00 80 06 5B 88 C3 D4 05 01 C3 D4  ...~@...[.....
00020: 05 18 00 15 04 00 0C 5D 49 55 00 04 E2 67 50 18  ....]IU...gP.
00030: 21 D3 EE CE 00 00 31 35 30 20 4F 70 65 6E 69 6E  !.....150 Openin
00040: 67 20 42 49 4E 41 52 59 20 6D 6F 64 65 20 64 61  g BINARY mode da
00050: 74 61 20 63 6F 6E 6E 65 63 74 69 6F 6E 20 66 6F  ta connection fo
00060: 72 20 74 6F 72 6E 61 64 6F 5C 74 61 72 67 65 74  r tornado\target
00070: 5C 63 6F 6E 66 69 67 5C 62 61 62 34 30 5C 76 78  \config\bab40\vx
00080: 77 6F 72 6B 73 28 34 37 32 33 36 39 20 62 79 74  works(472369 byt
00090: 65 73 29 2E 0D 0A                                     es)...
```

Question F.3.1

Indiquez les adresses Ethernet et les adresses IP du serveur et du rack.

Question F.3.2

Donner pour la trame 17 le nombre d'octets total et le nombre d'octets correspondant à chaque protocole.

Question F.3.3

Les paquets IP sont-ils fragmentés ? Justifiez votre réponse.

Question F.3.4

Quelles sont les commandes et les réponses FTP utilisées dans les trames 17 et 18? Quelle partie du dialogue est représentée par ces deux trames ?

F.4 Client-Serveur

La communication des résultats des mesures entre le rack de communication et le PC de contrôle utilise le réseau Ethernet 10baseT au travers du protocole IP .

On décide d'établir une communication en mode **non connecté**.

Question F.4.1

En vous référant à l'annexe D7 , précisez les valeurs des 3 paramètres d'ouverture de la socket de communication. Ces paramètres adoptent-ils les mêmes valeurs sur le PC de contrôle et sur le rack de communication ? Expliquez.

La socket de communication étant ouverte avec succès tant au niveau du PC de contrôle que du rack de communication, l'application attache maintenant cette socket à un point de communication. Cet attachement (binding) nécessite la fourniture d'une structure de type **sockaddr_in**.

Extrait de l'annexe D7:

```
struct sockaddr_in
{
    sa_family_t    sin_family; /* address family: AF_INET */
    u_int16_t      sin_port;   /* port in network byte order */
    struct in_addr sin_addr;   /* internet address */
};

/* Internet address. */
struct in_addr
{
    u_int32_t      s_addr;     /* IPv4 address in network byte order */
};
```

Dans notre application, c'est le PC de contrôle qui sera le premier en attente de réception de datagrammes sur le port **1057**. Les adresses IP des différentes machines sont récupérées dans les trames du paragraphe F.3...

Les questions qui suivent permettent de définir les valeurs des différents membres de la structure de type **sockaddr_in** à fournir à l'appel **bind()**.

Question F.4.2

Le type **u_int16_t** correspond à un entier non signé sur 16 bits.

Quelle valeur doit être fournie pour le champ **sin_port** de la structure sur le PC de contrôle?

Question F.4.3

Quelle valeur doit être fournie pour le champ **sin_port** de la structure sur le rack de communication?

Question F.4.4

Comment le PC de contrôle connaît-il le port utilisé par le rack de communication ?

Question F.4.5

Comment le rack de communication connaît-il le port utilisé par le PC de contrôle?

G. Annexes du sujet

- Annexe S1** *Carte CPU BAB40 (extraits)*
- Annexe S2** *Carte d'extension VMOD-IO (extraits)*
- Annexe S3** *Module VMOD TTL/O*
- Annexe S4** *Module VMOD-GPIB(extraits)*
- Annexe S5** *Coupleur parallèle CIO Z8536*
- Annexe S6** *Schéma structurel du module VMOD-RL*
- Annexe S7** *Circuit ULN2003*
- Annexe S8** *Driver GPIB*

BTS INFORMATIQUE INDUSTRIELLE

Session 2001

Epreuve Etude d'un Système Informatisé

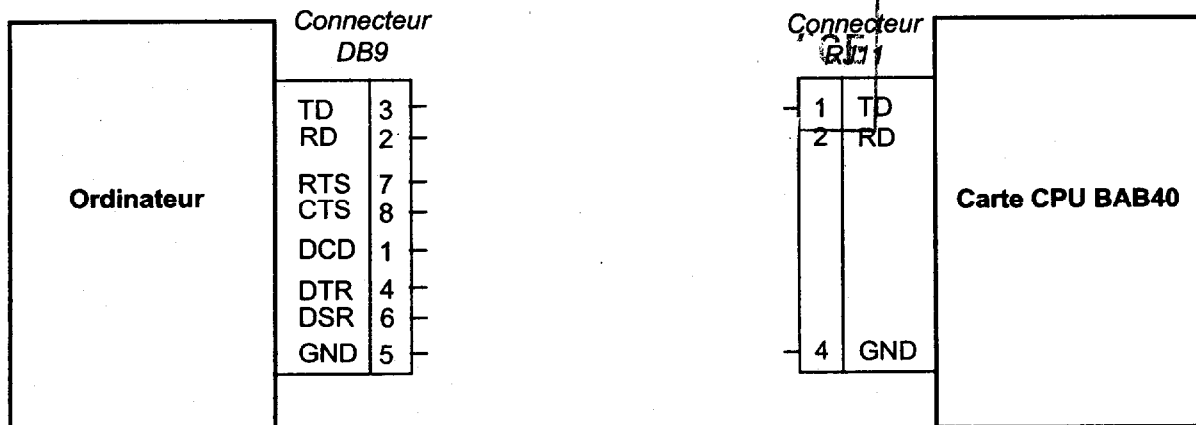
Système d'acquisition de température de bobinages

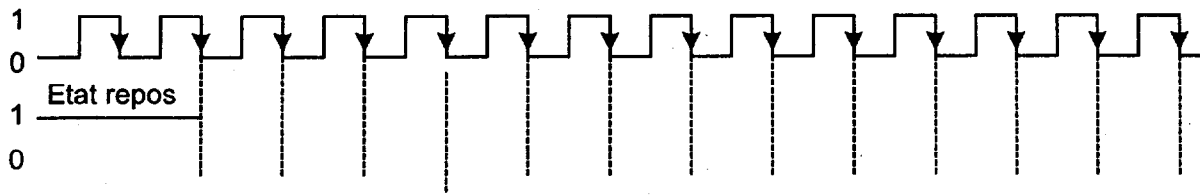
Sujet – deuxième partie : Document Réponse

Réponse D.2.1

	Rack de commutation	Multimètre 1	Multimètre 2	Multimètre 3	Multimètre 4
Fonction	Ecouteur ou parleur				
Adresse	7				
Code Ecouteur	\$27				
Code Parleur	\$47				

Réponse F.1.2



Réponse F.1.3**Réponse F.2.1**

Entité/protocole	Câble UTP	Routeur ADSL	Ethernet	Connecteur RJ45	TCP	IP	FTP
Couche(s)	1						

Réponse F.2.2

	Routeur	Serveur	PC de contrôle	Rack VME
Adresse IP				
Masque				