

14/96
الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التربية الوطنية
MINISTERE DE L'EDUCATION NATIONALE

ECOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التخصصات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

SUJET

RESEAUX DE PC A L'AIDE DE

COURANTS PORTEURS :

CARTE DE COMMUNICATION

Proposé par :

Mme Beddek
&
Mr Sadoun

Etudié par :

Mr y. Sail
&
F. Terranti

Dirigé par :

Mme Beddek
&
Mr Sadoun

PROMOTION

Septembre 1996

E. N. P 10, Avenue Hassen Badi - EL-HARRACH - ALGER

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التربية الوطنية

MINISTERE DE L'EDUCATION NATIONALE

ECOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
المكتبة — BIBLIOTHEQUE
Ecole Nationale Polytechnique

DEPARTEMENT D'ELECTRONIQUE

PROJET DE FIN D'ETUDES

SUJET

RESEAUX DE PC A L'AIDE DE

COURANTS PORTEURS :

CARTE DE COMMUNICATION

Proposé par :

Mme Beddek
&
Mr Sadoun

Etudié par :

Mr y. Sail
&
F. Terranti

Dirigé par :

Mme Beddek
&
Mr Sadoun

PROMOTION

Septembre 1996

DEDICACES

A toute ma famille
A mes amis
A mes proches
Je dédie ce modeste travail

Yazid.

A toute ma famille
A mes amis
A mes proches
Je dédie ce modeste travail

Fayçal

REMERCIEMENTS

Nous tenons tout d'abord à exprimer notre vive reconnaissance à nos promoteurs Mme BEDDEK et Mr SADOON pour leur suivi, leur aide et leur soutien tout au long de ce projet, nous les remercions encore pour les moyens qu'ils ont mis à notre disposition.

Nos remerciements vont également à tous les enseignants qui ont contribué à notre formation.

Nous remercions aussi KRIMO de la bibliothèque centrale de l'ENP.

Enfin nous remercions toute personne ayant de près ou de loin contribué avec un effort pour l'aboutissement de ce travail.

SOMMAIRE

RESUME.

INTRODUCTION GENERALE.

CHAPITRE 1 : GENERALITES SUR LA TRANSMISSION DE DONNEES.....3

- 1 _ Introduction.
- 2 _ Transmission analogique
- 3 _ Transmission numérique.
- 4 _ Transmission en bande de base.
- 5 _ Transmission en bande transposée.
- 6 _ Modulations analogiques discrètes.
- 7 _ Sens de transmission.
- 8 _ Milieux de transmission.
- 9 _ Types de liaisons.

CHAPITRE 2 : LES RESEAUX D'ORDINATEURS.....12

- 1 _ Généralités.
- 2 _ La gestion des réseaux
- 3 _ Transmission et topologies.
- 4 _ Infrastructure des réseaux.

CHAPITRE 3 : TRANSMISSION SERIE.....20

- 1 _ Introduction.
- 2 _ Modes de transmission.
- 3 _ La norme RS232_C.
- 4 _ Programmation du port série du micro ordinateur.
- 5 _ Modems.

CHAPITRE 4 : TRANSMISSION PAR COURANT PORTEUR38

- 1 _ Introduction.
- 2 _ Sources de bruits.
- 3 _ Choix du type de modulation.
- 4 _ Choix des fréquences.
- 5 _ La modulation de fréquence.

CHAPITRE 5 : PRESENTATION DE LA REALISATION.....42

- 1 _ Introduction.
- 2 _ Etude de la carte d'interface
 - 2.1 _ Partie émission.
 - 2.2 _ Partie réception.
- 3 _ Le logiciel d'exploitation du réseau.

CONCLUSION GENERALE

يهدف هذا المشروع إلى إنشاء بطاقة اتصال لتبادل المعلومات بين الحواسيب
و ذلك باستعمال الشبكة الكهربائية كناقل للمعلومات .
بطاقة الاتصال هذه عنصرها الأساسي الدارة المتكاملة EF7910 .
إضافة إلى القسم التجهيزي تم تطوير برنامج لتسيير عملية تبادل المعلومات .

Résumé

L'objet de ce projet consiste à réaliser une carte de communication entre micro-ordinateurs utilisant les courants porteurs.

Le réseau électrique jouera le rôle de support de transmission.

La carte de communication est bâtie autour du modem EF 7910.

Outre la partie matérielle, nous avons développé une partie logicielle qui gère l'échange de données.

Abstract

The object of this project consists of designing of one card for communication between computers using the mains supply.

The heart of this card is the modem EF 7910.

In addition to the hardware, a software gives all the necessary commands for the communication.



INTRODUCTION GENERALE

Les mutations technologiques opérées au cours de la seconde moitié des années soixante dix (70) ont favorisé l'émergence de mini et de micro-ordinateurs de plus en plus puissants, accessibles à des prix très compétitifs.

Cette catégorie d'ordinateurs va faire jouer à l'informatique un rôle nouveau, celui de la décentralisation dans une première étape et la constitution de véritables réseaux d'ordinateurs dans une seconde étape. Il y a lieu, cependant, de remarquer qu'en absence de standard à cette époque, l'évolution n'a pas joué en faveur des réseaux hétérogènes, mais plutôt en faveur des réseaux homogènes ou réseaux constructeurs tels que les réseaux DNA de DEC et le réseau SNA d' IBM.

Le rôle des standards est de permettre à des équipements provenant de constructeurs différents de communiquer et d'interpréter les informations échangées sans que l'utilisateur n'ait à opérer des modifications importantes à cette fin.

Deux organismes internationaux importants se disputent l'élaboration des standards:

- Le CCITT (Comité Consultatif International pour le Télégraphe et le Téléphone)
- L' ISO (International Standards Organisation) .

Les réseaux d'ordinateurs peuvent être classés selon leurs portée en deux catégories :

- Les réseaux locaux : les plus connus sont ETHERNET créé par DEC, INTEL et XEROX (norme DIX) et TOKEN RING créé par IBM.
- Les réseaux large portée : ce sont des réseaux couvrants de larges étendues géographiques, le plus connu est INTERNET.

L'interconnexion entre les micro-ordinateurs utilise plusieurs types de supports parmi lesquels on cite: la paire torsadée, le câble coaxial et la fibre optique. Ces types de supports nécessitent un câblage direct additionnel entre les micro-ordinateurs.

D'autres supports, ne nécessitant pas de câblage additionnel, peuvent être utilisés. On cite l'interconnexion par ligne téléphonique, radio et courant porteur.

Notre projet a pour but la réalisation d'une carte de communication utilisant le courant porteur pour l'échange d'informations entre micro-ordinateurs. Le réseau électrique jouera le rôle de support de transmission; ce qui peut constituer un moyen bon marché pour la réalisation d'un réseau d'ordinateurs.

Le réseau électrique a été choisi comme support de transmission pour l'interconnexion d'un réseau d'ordinateurs, car c'est le support le plus répandu et le moins sujet à des restrictions légales.

Pour pouvoir relier les ports séries des micro-ordinateurs au réseau électrique nous avons réalisé une carte d'interface. Cette carte d'interface doit adapter la nature numérique du micro-ordinateur à la nature analogique du réseau électrique. Elle est basée autour du modem intégré EF7910 programmé pour fonctionner en mode BELL 103.

La carte d'interface peut être divisée en deux parties :

1- Partie émission : elle est constituée des blocs fonctionnels suivants:

- le modem EF7910 (partie émission)
- le modulateur
- l'amplificateur de puissance

2- Partie réception : elle est constituée des blocs fonctionnels suivants:

- le filtre primaire
- l'amplificateur
- le démodulateur
- le filtre secondaire

CHAPITRE 1

GENERALITES SUR LA TRANSMISSION DE DONNEES



1 - INTRODUCTION

L'échange de données comprend l'ensemble des moyens techniques nécessaires à l'acheminement aussi fidèle et fiable que possible des données entre deux points a priori quelconque.

La liaison physique entre deux équipements informatiques, peut être représentée schématiquement comme indiqué sur la figure.1; le circuit de données est constitué d'un canal de transmission et de deux équipements de terminaison du circuit de données appelés ETCD dont le rôle est d'établir la communication et d'assurer une mise en forme de données numériques pour leur transmission à travers le canal.

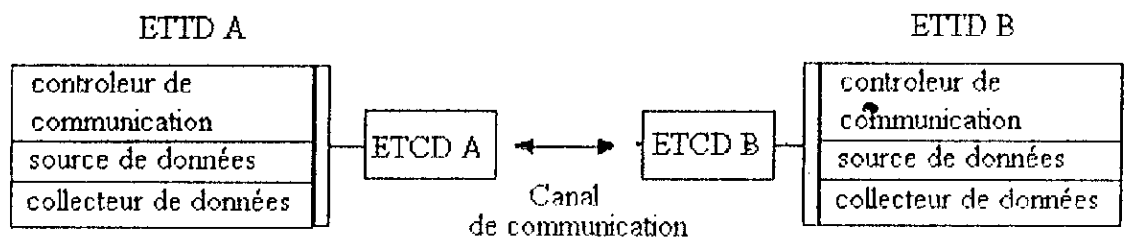


Figure.1 Structure d'une liaison entre deux ETTD

Ce circuit de données relie entre eux les deux équipements informatiques de traitement de données appelés ETTD.

Dans un ETTD nous distinguons en fait deux parties qui réalisent des fonctions différentes.

- Le système de traitement qui peut être source ou collecteur de données.
- Le contrôleur de communication qui regroupe les organes chargés des fonctions de communication.

2 - TRANSMISSION ANALOGIQUE

Pour pouvoir intégrer des micro-ordinateurs dans un réseau de transmission analogique il est nécessaire de convertir les signaux numériques en signaux analogiques et vice versa. La conversion peut être effectuée par des modems (modulateur démodulateur).

3 - TRANSMISSION NUMERIQUE

L'ordinateur est parfaitement adapté pour la transmission des informations numériques. Ce sont les signaux analogiques qu'il faudra convertir en numériques.

4 - TRANSMISSION EN BANDE DE BASE

Dans ce type de transmission le signal de données est appliqué directement sur le support de transmission sans transposition de fréquence (modulation) ce qui permet d'utiliser des ETCD simples.

5-TRANSMISSION EN BANDE TRANSPOSEE

Pour pouvoir adapter le signal représentant les données aux caractéristiques du canal on procède à une transposition de fréquence (Modulation), la modulation est appelée analogique discrète lorsque ce signal est numérique. Si le signal à transmettre est numérique la transposition considérée est dite modulation analogique discrète.

6 - MODULATION ANALOGIQUE DISCRETE

On désigne par modulations analogiques discrètes des procédés dans lesquels :

- Le signal primaire ou signal modulant est binaire noté $a(t)$
- La porteuse est un signal sinusoïdal noté $U_p(t) = U_p \cos(2 \pi f_p t)$
- Le signal modulé est obtenu par la variation discrète d'un des paramètres(amplitude, fréquence, phase) de la porteuse, il est noté $U_s(t)$.

Ces modulations sont analogiques dans le sens qu'elles ne modifient pas la nature de l'information portée par les signaux avant et après modulation.

Suivant le paramètre modulé on distingue les types suivants:



Suivant le paramètre modulé on distingue les types suivants:

6.1- Modulation ASK (Amplitude Shift Keying)

Le signal modulé a pour expression:

$U_s(t) = a(t) U_p \cos(2 \pi f_p t)$; où $a(t)$ prend les valeurs 0 ou 1. La figure.2 représente le signal $U_s(t)$ et sa densité spectrale $S(f)$.

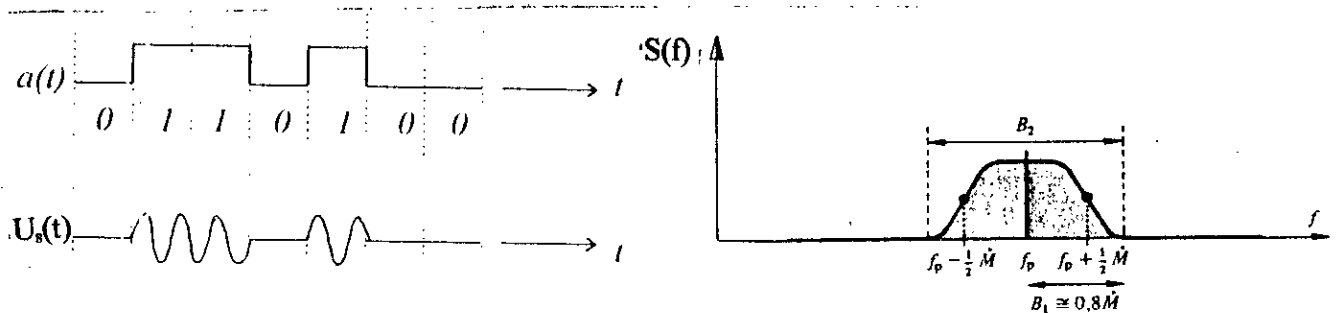


Figure.2 Représentation d'un signal ASK et de sa densité spectrale

La densité spectral de puissance $S(f)$ du signal ASK contient:

- Une raie à f_p .
- deux bandes latérales de part et d'autre de f_p correspondant chacune à la densité spectrale de puissance du signal primaire.

La largeur de bande B_2 nécessaire en ASK est le double de celle qui est nécessaire en bande de base, elle est donnée par l'équation suivante:

$$B_2 = 2 B_1 = 1.6 D, \text{ D est le débit binaire du signal primaire.}$$

Ce type de modulation est sensible aux bruits.

6.2 - Modulation FSK (Frequency Shift Keying)

Le signal modulé a comme expression:

$$U_s(t) = U_p \cos[2 \pi (f_p + a(t) 2 \pi \Delta f - \Delta f) t]$$

Le principe consiste donc à associer à chaque niveau logique 0 ou 1, une fréquence particulière de la porteuse ainsi le 0 correspond à une fréquence $f_1 = f_p - \Delta f$ et le 1 correspond à une fréquence $f_2 = f_p + \Delta f$. La figure.3 représente $U_s(t)$ et sa densité spectrale $S(f)$.

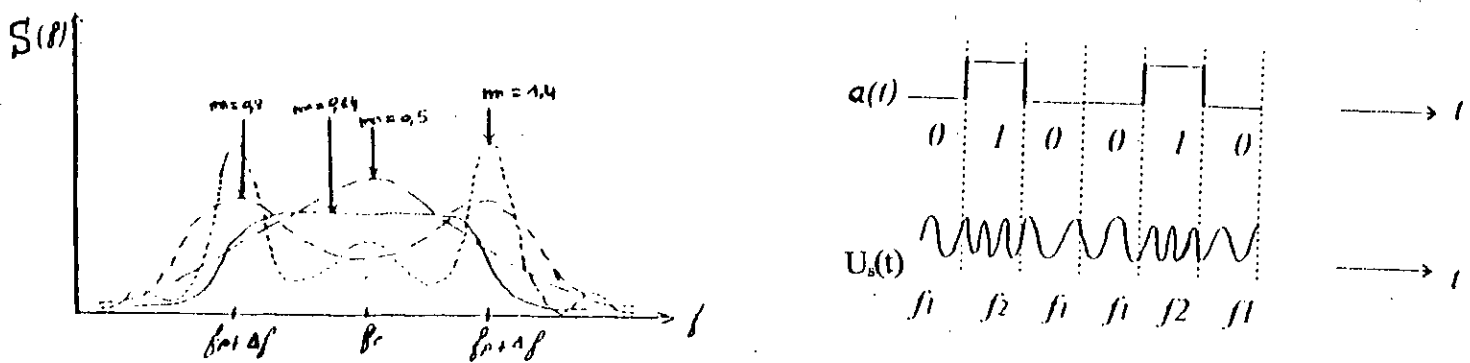


Figure.3 Représentation d'un signal FSK et de sa densité spectrale

Le spectre du signal $U_s(t)$ est décrit par la figure.3 pour différentes valeurs du paramètre $m = 2 \Delta f \cdot T$ appelé indice de modulation (T correspond à la durée d'un bit).

On voit que le spectre devient plus accentuée en $f + \Delta f$ et $f - \Delta f$ lorsque m augmente. La largeur de bande nécessaire est donnée par la formule suivante:

$$B_2 = 1,6 D + 2 \Delta f$$

D est le débit binaire du signal primaire $a(t)$. Ce type de modulation permet une transmission de meilleure qualité car elle est moins sensible aux bruits que la ASK

6.3 - Modulation PSK (Phase Shift Keying)

Dans ce cas, c'est la phase de la porteuse qui est modifiée en fonction de la forme du signal $a(t)$. Le signal modulé a comme expression:

$$U_s(t) = U_p \cos[2 \pi f_p t + a(t) \pi];$$

Le signal $U_s(t)$ prend la forme décrite dans la Figure.4. La modulation de phase permet d'envoyer n fois plus d'information sur la même bande passante.

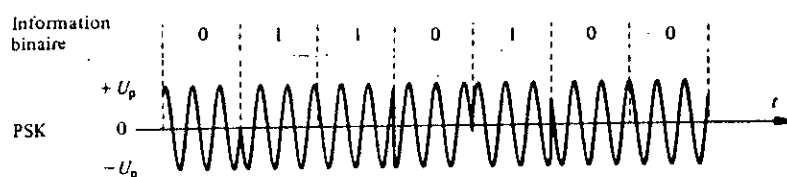


Figure.4 Représentation d'un signal PSK

7 - SENS DE TRANSMISSION

7.1 - Transmission Simplex

L'information est transmise sans possibilité de retour; l'exploitation est en mode unidirectionnel de A vers B.(figure.5)



Figure.5 - Liaison simplex

7.2 - Transmission Half Duplex

Les informations circulent alternativement de A vers B puis de B vers A et vice versa, l'exploitation est en mode bidirectionnel.(figure.6)

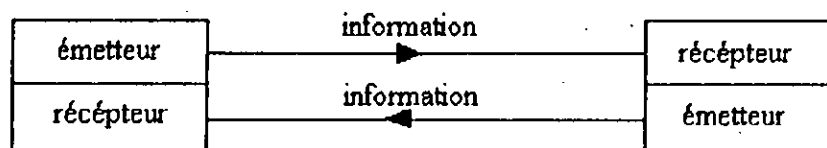


Figure.6 - Liaison Half duplex

7.3 - Transmission Full Duplex

Les informations circulent dans les deux sens simultanément de A vers B et de B vers A, l'exploitation est en mode bidirectionnel simultané.(figure.7)

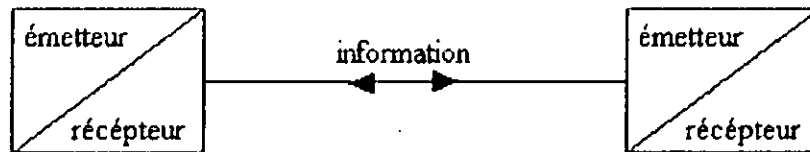


Figure.7 - Liaison Full duplex

8 - MILIEUX DE TRANSMISSION

On distingue deux sortes de milieux de transmission :

- Les supports matériels
- Les supports immatériels.

8.1 - Supports matériels

Les supports matériels peuvent être des fils, des câbles électriques ou des fibres optiques.

8.1.1 - Transmission par réseau électrique

Le principe de la transmission par réseau électrique est d'utiliser les fils d'installation électrique en superposant le signal à transmettre au signal du réseau électrique.

8.1.2 - Transmission par réseau téléphonique

Le réseau téléphonique peut être utilisé pour transmettre les signaux ayant leurs spectres compris dans la bande réservée aux communications téléphoniques.

8.1.3 - Transmission par fibre optique

Une transmission par fibre optique est caractérisée par trois éléments : une source lumineuse, des fibres optiques et un détecteur.

8.2 - Supports immatériels

On peut classer les supports immatériels en trois catégories : les ondes radio, les ultrasons et les infrarouges.



8.2.1 - Transmission par ultrasons

Dans une transmission par ultrason l'émetteur est caractérisé par un haut parleur spécialement conçu pour les fréquences hautes et appelé transducteur. Le récepteur est un transducteur identique qui transforme les ondes reçues en signaux électriques (Microphone).

8.2.2 - Transmission par infrarouge

Dans un émetteur infra rouge les signaux utiles sont transformés en rayonnement infrarouge au moyen d'une LED. Une photo diode ou un photo transistor permettent au récepteur de capter les rayonnements et de les convertir en signaux électriques.

8.2.3 - Transmission par ondes radio

Les ondes restent un véhicule important pour toute sorte de transmission à courte, moyenne ou longue distance. Le rôle de l'émetteur consiste à produire la puissance suffisante pour faire passer le message, celui du récepteur est de l'isoler de la multitude des signaux captés par l'antenne.

9 - TYPES DE LIAISONS

9.1 - Liaison point à point

elle ne comporte que deux extrémités (figure.8)

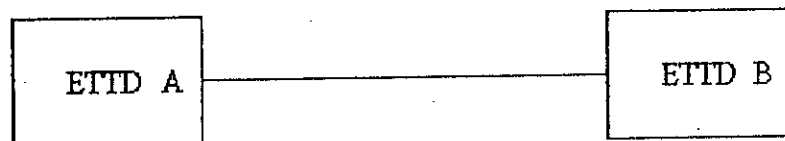


Figure.8 - Liaison point à point

9.2 - Liaison multipoints

lorsqu'on doit connecter plusieurs terminaux à un équipement central, il est préférable de les relier à l'équipement central par une liaison multipoints. Si les données émises par

l'équipement central sont reçus simultanément par tous les terminaux connectés, les transmissions vers l'équipement central ne peuvent s'effectuer que l'une après l'autre d'une manière ordonnée. (figure.9)

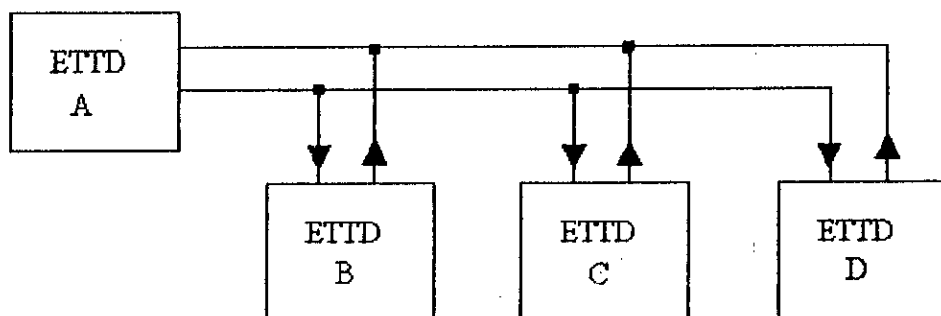


Figure.9 - Liaison multipoints

9.3 - Liaison en boucle

Dans une liaison en boucle les deux extrémités de la liaison sont reliés à l'équipement central, la transmission peut s'établir dans un sens déterminé et une procédure spéciale est nécessaire pour permettre aux terminaux connectés d'émettre ou de recevoir d'une manière ordonnée. (figure.10)

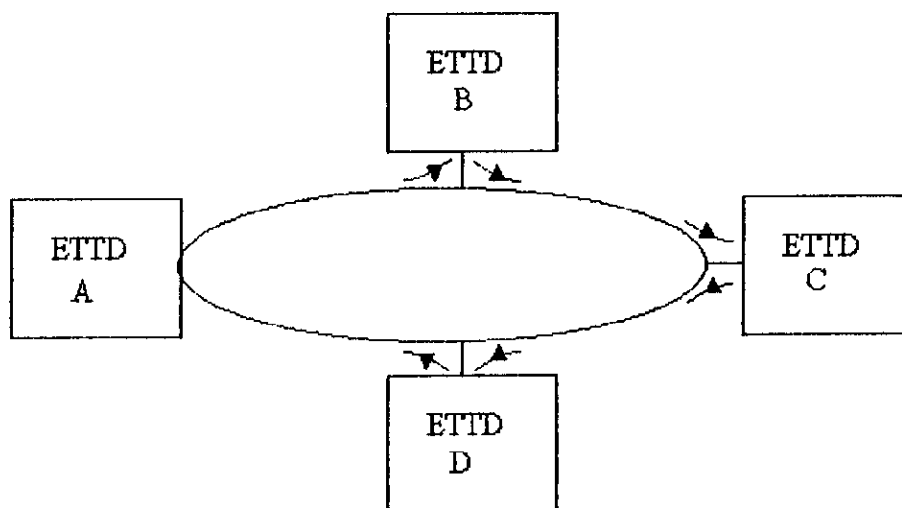


Figure.10 - Liaison en boucle

CHAPITRE 2

RESEAUX D'ORDINATEURS



1 - GENERALITES

1.1 - Définition

On appelle réseau le raccordement de micro-ordinateurs et périphériques par un câblage grâce auquel ces divers éléments peuvent communiquer. Le raccordement de PC présente l'avantage de centraliser le stockage des fichiers pour assurer leur intégrité et leur sauvegarde.

Les réseaux les plus courants sont ETHERNET et TOKEN RING. On peut décrire le réseau ETHERNET comme une longueur de câble servant d'épine dorsale à l'ensemble des éléments raccordés. Le réseau TOKEN RING (anneau à jeton), à l'inverse, comme son nom l'indique met en oeuvre un câble en anneau fermé relié à chaque PC en deux points.

Les réseaux peuvent être organisés pour un travail de groupe, ou pour tout un département, ou être reliés à d'autres réseaux. Le réseau WAN (Wide Area Network) est constitué de plusieurs réseaux situés dans divers lieux géographiques, la communication se fait par ligne téléphonique ou par satellite.

1.2 - Le serveur du réseau

Sur un réseau, il est courant d'attribuer à un PC le rôle de serveur du réseau. Sa fonction principale est de stocker les fichiers de données pour les utilisateurs et d'assurer l'envoi et la réception des fichiers à la demande. Les autres ordinateurs du réseau sont appelés stations.

Les serveurs tournent sous des logiciels d'exploitation pour réseaux (NOS :Network Operating System) tel que NetWare, et disposent d'une large capacité de mémoire.

1.3 - Matériel et logiciel

En plus d'un serveur, de stations et d'un câble, il faut des adaptateurs de réseau, que l'on connecte aux stations et au serveur pour établir la communication. Il faut aussi un logiciel d'exploitation spécial comme NetWare ou LAN Manager pour commander le fonctionnement du réseau.



1.4 - Gestion

Comme dans tous les systèmes impliquant la circulation d'un flux d'informations dans différentes directions sur des voies pré-définies, on a besoin d'assurer une gestion et une surveillance pour que cet ensemble fonctionne efficacement.

Le serveur peut abriter des informations confidentielles non accessibles à tous les utilisateurs du réseau. Il est essentiel d'assurer une certaine sécurité, afin que les accès aux fichiers ne soient possibles que pour les utilisateurs habilités à consulter certains types d'informations.

2 - LA GESTION DES RESEAUX

2.1- Généralités

Les réseaux de PC constituent l'épine dorsale de nombreux systèmes d'information en milieu professionnel. Ils ne cessent de croître en taille et en complexité.

La nécessité d'optimiser la productivité, de réduire le temps d'immobilisation et de gérer les services va donc revêtir une importance croissante, il faudra ici faire appel à une panoplie d'outils spéciaux.

2.2 - Outils de gestion

On distingue deux grandes catégories pour la gestion et l'optimisation des réseaux: les outils de surveillance et les outils d'analyse. Ces outils s'attachent à détecter les anomalies et à assurer la fiabilité de l'infrastructure du réseau.

2.2.1 - Outils de surveillance

Ces outils fournissent une vue d'ensemble sur le réseau et facilitent l'exécution des fonctions de gestion.

2.2.2 - Outils d'analyse

Les outils d'analyse agissent plutôt comme des microscopes, ils détectent des défaillances et permettent de réduire les temps d'immobilisation.



2.3 - Gestion locale

On constate que les réseaux sont de plus en plus utilisés pour les applications les plus sensibles des entreprises, d'où une forte demande de gestion et pour des services destinés à ces réseaux de PC. On l'appelle la gestion locale.

La nouvelle génération d'outils de gestion locale pour réseaux permet aux administrateurs de piloter un grand nombre de tâches à partir d'une console, grâce à un logiciel intégré.

3 - TRANSMISSION ET TOPOLOGIES

3.1 - Transmission

Les unités d'un réseau communiquent au travers d'un câble essentiellement par trois méthodes: la détection de signal et de collision CSMA/CD d'ETHERNET, le passage à jeton de TOKEN RING ou l'interrogation ('polling').

3.1.1 - Détection de signal et de collision

Cette technique englobe trois domaines : La transmission et la réception de données, la détection de collision des paquets de données en cas d'accès simultanés.

La transmission est couverte par la première partie 'CSMA' de l'acronyme. Avant de transmettre des données sur le réseau CSMA/CD, l'émetteur se met à l'écoute pour savoir si quelqu'un d'autre utilise le câble. L'émetteur ne peut transmettre que si le câble est libre; les données sont transmises par 'paquets' ou 'trames'. On appelle paquet un bloc d'informations véhiculé comme une seule entité, comportant des identifications d'émission et de réception. Tous ceux qui sont sur le réseau CSMA/CD sont à l'écoute des envois de paquets. Le destinataire regarde si le paquet comporte sa propre adresse, et il le lit si tel est le cas, sinon il ignore le paquet.

La détection de collision (partie CD de CSMA/CD) est une technique dans laquelle l'émetteur examine ce qui a été transmis et le compare à celui qui avait été envoyé, si le message diffère, l'émetteur comprend qu'il y a eu collision avec un message expédié par une autre station; chaque station détectant une collision met fin à sa propre émission, attend un certain

X

temps puis fait une nouvelle tentative de transmission. Le principe se poursuit jusqu'à ce que le message soit correctement transmis.

3.1.2 - Passage de jeton

Cette technique alloue à chaque station un créneau de temps pré-défini pour transmettre ces paquets de données. Cela se fait par la circulation d'un jeton autour du réseau, ce jeton passe d'une station à l'autre dans un ordre défini, la station en possession du jeton peut transmettre ces données, ou simplement passe le jeton à la station suivante (figure.1). A la fin du créneau de temps, la station doit passer le jeton à la suivante, qu'elle ait ou non fini de transmettre tous ces paquets.

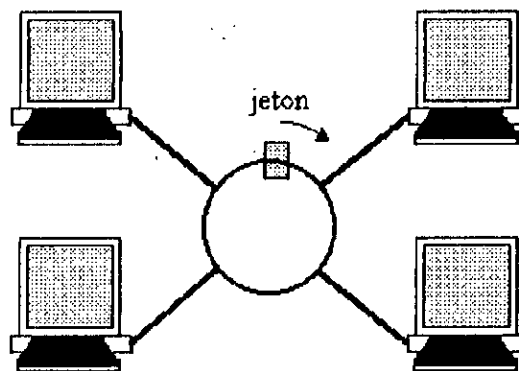


Figure.1 - Transmission par anneau à jeton

3.1.3 - Le polling

Le POLLING ou interrogation est généralement employé sur des anciens réseaux en étoile, au cours de l'interrogation le serveur scrute les stations une à une pour savoir s'il y a des messages en attente, le serveur commande toutes les transmissions de données sur le réseau.

3.2 - Topologies

La topologie d'un réseau définit l'organisation logique du réseau, c'est à dire la répartition du câblage et des unités. Parmi les diverses topologies, citons : le bus, l'étoile, l'étoile répartie et l'anneau en étoile câblée.

X

3.2.1 - Topologie en bus

C'est la topologie la plus courante, elle est principalement employée avec les réseaux ETHERNET; une simple longueur de câble constitue l'épine dorsale du réseau.(figure.2)

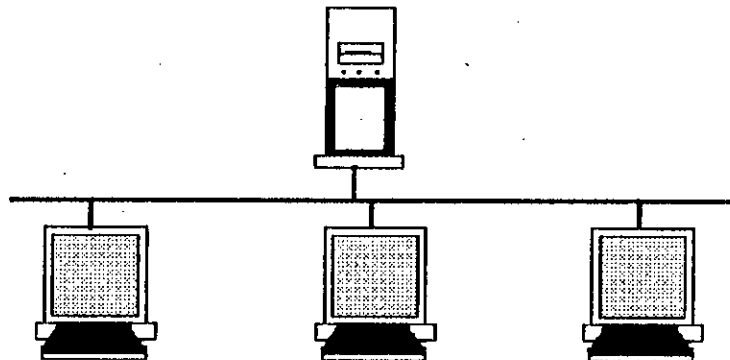


Figure.2 - Topologie en Bus

3.2.2 - Topologie en étoile

Un câblage part en étoile du PC serveur, si un câble est défaillant cela ne touche que cette station, et le reste du réseau n'est pas affecté.(figure.3)

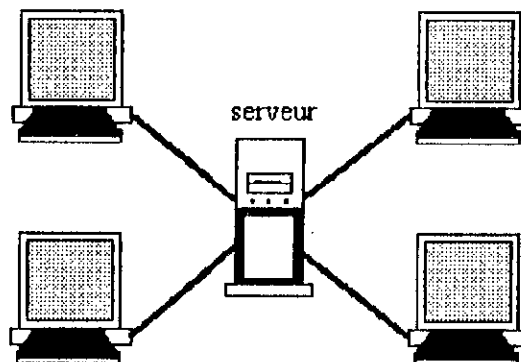


Figure.3 - Topologie en étoile

3.2.3 - Topologie en étoile répartie

Les stations sur le réseau ne sont pas connectées au PC serveur mais à un concentrateur.(figure.4)

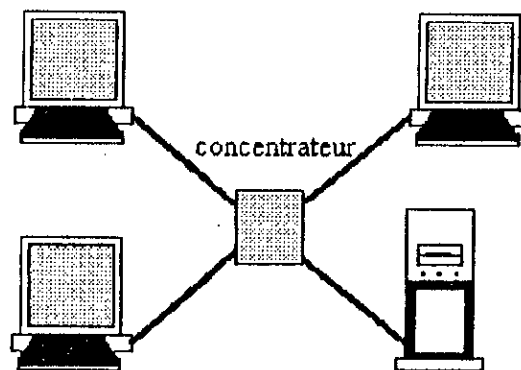


Figure.4 - Topologie en étoile répartie

3.2.4 - Topologie en anneau en étoile câblée

Elle est employée par les réseaux à passage de jeton (TOKEN RING d'IBM). (figure.5)

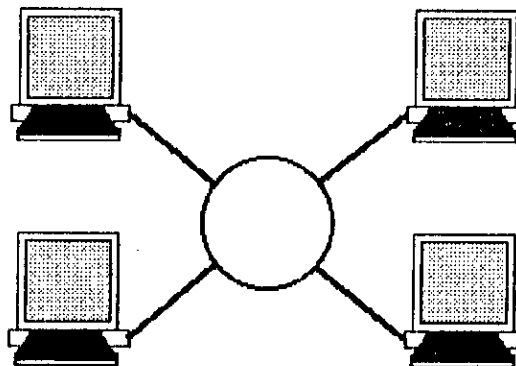


Figure.5 - Topologie en anneau en étoile câblée

4 - INFRASTRUCTURE DES RESEAUX

4.1 - Câblage

Quelque soit le réseau, il faut tenir compte du type de câblage employé, chacun ayant ces avantages et ces inconvénients.(figure.6)

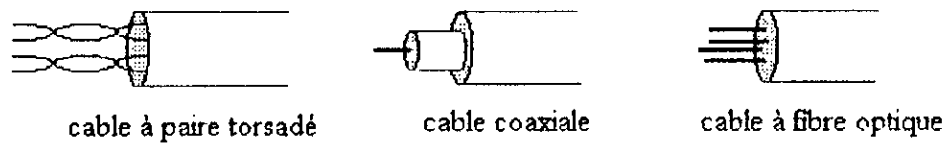


Figure.6 - Types de câblages pour réseau

4.2 - les adaptateurs de réseau

Les stations à relier doivent être équipées de cartes d'interface (adaptateur). Ces cartes trouvent place dans l'un des emplacements d'extension de la station.

On appelle adaptateur les interfaces situées entre la station, le serveur et les ressources du serveur.(figure.7)

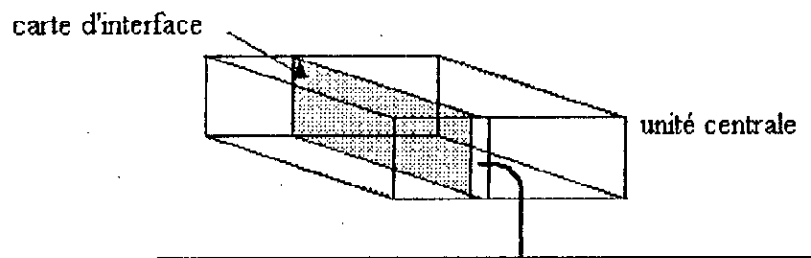


Figure.7 - Adaptateur de réseau

4.3 - Les concentrateurs

Les concentrateurs sont le point central d'un réseau en étoile, il joue le rôle de répéteur de signaux. Les concentrateurs classiques sont des systèmes indépendants. Les produits les plus récents se présentent sous la forme de carte additionnelle à installer dans le serveur.

CHAPITRE 3

TRANSMISSION SERIE



1 - INTRODUCTION

Le transfert de données par une liaison série vient pallier aux problèmes qui apparaissent lorsque ce transfert est effectué par une liaison parallèle sur longues distances, en effet le coût des câbles de liaison devient important et leurs volumes encombrant, de plus, leurs propriétés électriques allongent le temps de transition des signaux (de 0 à 1 ou inversement) ce qui favorise des pertes de données.

La liaison série consiste à envoyer les bits de données séparément l'un après l'autre sur un seul circuit consistant en deux fils seulement. A la réception, le processus est inversé, les bits séparés sont rassemblés pour reconstituer l'octet d'origine.

Cette méthode réduit l'encombrement et le coût, malheureusement ces avantages sont liés à une diminution de rendement, il faut au moins huit fois plus de temps pour transmettre huit bits séparément l'un après l'autre que pour les transmettre simultanément en parallèle.

Heureusement, cette perte de données s'avère sans gravité dans la plupart des applications si l'on considère certains périphériques série, on observe une caractéristique commune : Ils sont lents, chacun d'eux gaspille une partie du temps, généralement en fonctionnement mécanique, ce qui réduit beaucoup la vitesse. Ainsi, la vitesse autorisée par les transferts de données en parallèle est mal utilisée sur ces périphériques.

La méthode série, par conséquent, peut permettre de sacrifier une partie de la vitesse, tout en gérant les périphériques de façon satisfaisante. Dans ce cas, la perte de vitesse est sans importance par rapport à l'accroissement de fiabilité et à l'amélioration de la portée de la transmission.

2 - MODES DE TRANSMISSION

Il y a deux modes de transmission série, la transmission synchrone et la transmission asynchrone, ces deux modes utilisent la notion de la durée de bit.

Comme les bits sont transmis sur la ligne de liaison un par un, il est nécessaire de disposer d'une technique pour déterminer quand un bit s'arrête et le suivant commence. Cette technique

X

consiste à définir l'intervalle de temps pendant lequel un bit est présent sur la ligne, cet intervalle de temps s'appelle la durée de bit.

2.1 - Transmission série asynchrone

Ce mode de transmission est appelé ainsi parce qu'il n'y a pas de durée définie entre la transmission d'un caractère et celle du suivant.

Comme il n'y a pas de synchronisation entre caractères, chacun doit être complété par des bits de synchronisation.

En format asynchrone, chaque caractère transmis est précédé par un bit START (0 logique), ceci prévient le récepteur qu'un caractère va arriver. Le caractère est suivi d'un ou plusieurs bits STOP (1 logique), ce qui laisse au récepteur une période de repos avant la transmission du caractère suivant. On voit donc que le début d'un caractère est repéré sans ambiguïté par la transition '1' vers '0' de la ligne qui correspond à l'arrivée du bit START, les instants d'échantillonnage des différents bits du caractère sont déterminés à partir du bit START lors de la réception.

Pour vérifier que le caractère a été correctement transmis, un bit supplémentaire est ajouté à l'ensemble des bits START, STOP et les bits de données proprement dit (les bits de données peuvent être au nombre de 5, 6, 7 ou 8), ce bit prend la valeur '0' ou '1' de façon à ce que le nombre total de bits '1', y compris le bit de parité soit pair (c'est alors un contrôle de parité) ou impair (c'est alors un contrôle d'impairité), la figure.1 illustre le format d'une transmission asynchrone pour le caractère 1101001.

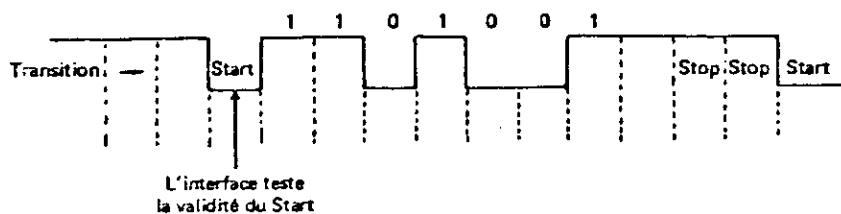


Figure.1 - Transmission asynchrone du caractère 1101001



2.2 - Transmission série synchrone

La transmission synchrone est utilisée lorsque le flux de données entre l'émetteur et le récepteur est régulier et notamment lorsque les quantités d'informations à transmettre sont importantes. Dans ce cas, les données sont envoyées par bloc de N caractères, deux caractères n'ayant aucun temps mort entre eux; la synchronisation se fait à l'aide d'un référentiel temporel commun qui peut être mis en oeuvre par différents procédés.

3 - LA NORME RS232 C

Pour faciliter l'interconnexion physique entre équipements de constructeurs différents, des normes qui spécifient la nature de l'interface entre deux dispositifs de communication série ont été développées. La norme d'une interface définit essentiellement les caractéristiques électriques, mécaniques, fonctionnelles et procédurales.

La RS232 C est une norme qui couvre les règles et les spécifications électriques, mécaniques et fonctionnelles nécessaires à la transmission série d'informations binaires entre un équipement terminal de terminaison de données (ETTD) et un équipement terminal de communication de données (ETCD).

3.1 - Caractéristiques électriques

3.1.1 - définition des niveaux logiques

Le niveau logique "zéro" est représenté sur le fil de données par une tension positive comprise entre +5 et +15 volts, le niveau logique "un" est représenté par une tension négative comprise entre -5 et -15 volts.

3.1.2 - Connexion avec les circuits TTL

Il existe sur le marché des circuits intégrés différents pouvant réaliser la conversion des niveaux TTL vers la RS232 C et vice versa. Parmi ceux-ci, on cite l'émetteur de ligne MC1488 et le récepteur de ligne MC1489.

3.1.3 - Vitesse de transmission

La durée de bit détermine la cadence maximale à laquelle les caractères peuvent être transmis et définit ainsi la fréquence de bit à laquelle fonctionne une interface série déterminée. Les vitesses de transmission (débit) normalisées pour les liaisons asynchrones sont 50, 75, 110, 134.5, 150, 300, 600, 1200, 2400, 4800, 9600 et 19200 bits par seconde.

3.2 - Caractéristiques mécaniques

Les connecteurs de branchement se présentent sous forme de connecteurs de type D, de 9 ou de 25 broches appelés respectivement DB9 et DB25. La figure.2 donne la forme et le brochage des deux connecteurs. Le tableau.1 donne les positions et les noms des signaux utilisés dans la liaison RS232 C ainsi que l'équivalence des deux connecteurs.

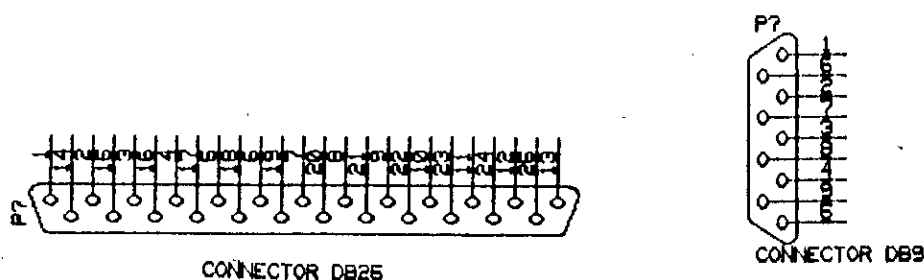


Figure.2 - Brochage des connecteurs DB9 et DB25.

Tableau 1: Equivalence DB25 DB9

SIGNAUX	DB9	DB25
DCD\	1	8
RxD	2	3
TxD	3	2
DTR\	4	20
GND	5	7
DSR\	6	6
RTS\	7	4
CTS\	8	5
RI	9	22

3.3 - Caractéristiques fonctionnelles

Le connecteur RS232 C est un modèle à 25 broches, vingt sont affectées aux signaux RS232 C, deux sont réservées (9,10) et trois non affectées (11,18 et 25), le tableau.2 donne la liste de tous les signaux RS232 C avec la broche sur laquelle ils se trouvent ainsi que celle de leur emploi.

En fait il n'existe que 8 signaux fondamentaux de la norme RS232 C, ils se trouvent sur les broches 2 à 8 et sur la broche 20, ces signaux sont répartis en deux catégories :

3.3.1 - les signaux de données (broches 2, 3 et 7)

Les lignes de données TxD et RxD, appelées respectivement émission de données et réception de données, ils sont indispensables du fait que la liaison est bilatérale. La masse logique (broche 7) constitue la ligne de retour pour les signaux qui circulent sur les lignes TxD et RxD.

3.3.2 - les signaux de contrôle et de vérification

La norme prévoit des signaux pour le contrôle et la commande d'une liaison entre un ETTD et un modem:

CTS : (Clear To Send) Modem prêt à recevoir des données

RTS : (Request To Send) Interface d'E/S prête à émettre

DSR : (Data Set Ready) Modem prêt à émettre des données

DTR : (Data Terminal Ready) Interface d'E/S prête à entrer en communication

DCD : (Data Carrier Detect) Détection de porteuse

Rq: Les signaux précédents ont été désignés du point de vue de l'ETTD.

3.4 - Caractéristiques procédurales

Par caractéristiques procédurales on entend la division en séquences des différents signaux de contrôle qui permet un transfert de données correct entre deux équipements.

Tableau 2 : Signaux de la RS232 C

BROCHE	SIGNAL	REMARQUES
1	Terre de protection (FG)	
2	Données d'émission (TxD)	ETTD --> ETCD
3	Données de réception (RxD)	ETTD <-- ETCD
4	Demande d'émission (RTS)	ETTD --> ETCD
5	Prêt à émettre (CTS)	ETTD <-- ETCD
6	Modem Prêt (DSR)	ETTD <-- ETCD
7	Masse signal (GND)	
8	Détection de porteuse (DCD)	ETTD <-- ETCD
9	-	
10	-	
11	-	
12	CD secondaire (S DCD)	ETTD <-- ETCD
13	CTS secondaire (S CTS)	ETTD <-- ETCD
14	TxD secondaire (S TxD)	ETTD --> ETCD
15	Horloge émission (TxClk)	ETTD --> ETCD
16	RxD secondaire (S RxD)	ETTD <-- ETCD
17	Horloge de réception (RxClk)	ETTD <-- ETCD
18	-	
19	RTS secondaire (S RTS)	ETTD --> ETCD
20	Terminal de données Prêt (DTR)	ETTD --> ETCD
21	Détection signal qualité (SQ)	
22	Indicateur de sonnerie (RI)	ETTD <-- ETCD
23	Choix cadence de données	ETTD --> ETCD
24	Horloge émission externe ETClk	ETTD --> ETCD
25	-	

4 - PROGRAMMATION DU PORT SERIE DU MICRO-ORDINATEUR

La programmation du port série du micro-ordinateur peut se faire de différentes manières:

- Par des fonctions DOS.
- Par des fonctions BIOS.
- Par programmation directe du port.

4.1 - Les fonctions DOS

Les deux fonctions DOS qui concernent la communication série sont appelées en exécutant une interruption 21H. La première fonction sert aux entrées série, celle-ci est réalisée en mettant le registre AH à 3 et en exécutant une instruction INT 21H, la deuxième fonction concerne les sorties série, elle est effectuée en mettant le registre AH à 4, en plaçant le caractère à émettre dans le registre DL et en exécutant une instruction INT 21H. Le caractère sera envoyé à COM1.

Ces deux fonctions présentent de sérieux handicaps qui restreignent leur domaine d'application. En effet, elles ne permettent pas d'accéder au contrôle de flux et n'offrent aucun moyen pour initialiser les paramètres de communication (vitesse de transfert, format de mot, etc...).


4.2 - Les fonctions BIOS

L'interruption 14H fait appel aux quatre fonctions BIOS concernant les communications série. Il faut placer un nombre de 0 à 3, dans le registre AH pour indiquer laquelle des fonctions est appelée, puis indiquer ensuite un numéro de port dans le registre DX, 0 pour COM1 et 1 pour COM2.

4.2.1 - La fonction initialisation des paramètres de communication

La première fonction BIOS, la fonction 0, sert à initialiser les paramètres de communication. On y accède en mettant le registre AH à 0 et le registre DX au numéro de port, en plaçant un octet représentant les paramètres vitesse, parité, longueur du mot et nombre de bits STOP dans le registre AL, puis en exécutant une instruction 14H.

Lorsque les paramètres ont été initialisés, la fonction retourne l'état courant du port dans le registre AX.



4.2.2 - La fonction émission de caractères

La deuxième fonction BIOS, fonction 1, sert à l'émission des caractères. On l'appelle en mettant le registre AH à 1 et DX au numéro de port, ensuite on place le caractère à envoyer dans le registre AL, puis on exécute l'instruction INT 14H. Le caractère n'est pas envoyé tant que les lignes de contrôle entrantes (DSR et CTS) ne sont pas à l'état haut.

La mise en oeuvre conventionnelle consiste tout d'abord, à exécuter la fonction saisie de l'état du port, puis la fonction émission de caractères une fois que l'on sait que les conditions sont favorables c'est à dire les lignes de contrôle DSR et CTS sont actives.

En retour, le registre AH rend compte de toute condition d'erreur. Si le bit 7 est à zéro, alors l'émission a réussi, s'il est à un, les bits restant indiquent le type d'erreur qui s'est produit.

4.2.3 - La fonction réception de caractères

La fonction réception de caractères, fonction 2, est appelée en mettant AH à 2, DX à la valeur du numéro du port, puis en exécutant l'instruction INT14H. Le BIOS attend qu'un caractère en provenance du port série soit reçu ou bien qu'un dépassement de temps soit atteint, lorsqu'un caractère est reçu, il est placé dans AL et toute condition d'erreur sera reportée dans AH.

Si AH vaut zéro, aucune erreur ne s'est produite. Si AH est différent de zéro, les bits 0 à 7 indiquent alors la condition d'erreur. Toutefois, si le bit 7 est mis à 1, cela indique une erreur de dépassement de temps, les bits restant n'ont aucune signification particulière.

4.2.4 - La fonction saisie de l'état du port

La fonction saisie de l'état du port, fonction 3, est appelée en mettant AH à 3, DX à la valeur du numéro de port, puis en exécutant l'instruction INT14H. Elle fournit diverses informations à propos de l'état du port série, puis retourne cet état dans le registre AX.

4.2.5 - Contrôle de flux avec le BIOS

Le BIOS se comporte très bizarrement vis à vis du contrôle de flux. La fonction Réception de caractères élève le signal DTR et abaisse le signal RTS, elle attend ensuite l'arrivée d'un caractère et retourne une erreur de dépassement s'il n'en parvient aucun après un certain

X

laps de temps. La fonction émission de caractères active DTR et RTS, puis attend que DSR et CTS soient tous les deux mis à l'état haut par l'appareil connecté au port série. Si les lignes ne sont pas mises à l'état haut. La fonction retourne une erreur de dépassement de temps. On voit que le BIOS requiert pour émettre deux signaux de contrôle alors qu'il n'en délivre qu'un seul en réception, de plus, il met RTS à zéro lorsqu'il est en attente de réception.

Par ailleurs, les signaux de contrôle sortant sont à l'état bas, tant que la fonction émission ou la fonction Réception n'a pas été appelée.

Cette interruption BIOS, bien que nettement plus intéressante que celle du DOS présente plusieurs désavantages, car elle ne propose pas la possibilité de gestion libre du flux, de plus elle limite les possibilités qu'offre l'UART du PC.

4.3 - Programmation directe du port série

Cette technique de programmation consiste à accéder directement au port série en utilisant des instructions appropriées. Elle permet une gestion libre des lignes de contrôle de flux ainsi qu'une exploitation complète des possibilités qu'offre l'UART. Toutefois, la connaissance des différents registres de ce circuit est indispensable.

L'UART (Universal Asynchrone Receiver Transmitter) est un circuit d'interface travaillant en mode asynchrone. Elle permet de réaliser la liaison série entre le micro-ordinateur et ses interfaces. Ce circuit possède des registres internes qui sont répartis comme suit :

4.3.1 - Registres de contrôle

Quatre registres de contrôle servent à recevoir les commandes provenant de l'unité centrale de traitement.

Le registre de contrôle de ligne

Il est utilisé pour initialiser les paramètres de communication, la signification de chaque bit est représentée par le tableau.3



Tableau.3 : Description du registre de contrôle de ligne

Bit	Signification
0	Longueur du mot
1	Longueur du mot
2	Bit d'arrêt
3	Valide la parité
4	Sélectionne la parité
5	Bit de parité à 1.
6	Signal break
7	Bit d'accès à la bascule de division.

Le registre de contrôle de modem

Le registre de contrôle de modem commande les signaux de contrôle émis par l'UART. Les bits 0,1 permettent de mettre les sorties DTR et RTS aux niveaux désirés. Les bits 2 et 3 contrôlent les sorties auxiliaires. Le bit 4 valide le mode de test par diagnostic. Les bits 5 à 7 sont en permanence maintenus à zéro.

Registre de validation des interruptions

Ce registre sert à indiquer à l'UART, la cause d'une éventuelle interruption. Le tableau.4 décrit la signification de chacun de ses bits.

X

Tableau 4 : Description du registre de validation des interruptions

Bit	Signification
0	Donnée disponible
1	Registre d'émission vide
2	Etat de la ligne de réception
3	Etat du modem
4,5,6,7	Forcés à zéro

Les bascules de division de la vitesse de transfert

La valeur de la vitesse de transfert est établie en introduisant dans deux registres le nombre par lequel la fréquence de l'horloge d'entrée (1,8432 Mhz) doit être divisée. La fréquence résultante vaut 16 fois la vitesse de transfert.

Ces deux registres sont d'une part l'octet le moins significatif de la bascule de division et d'autre part l'octet le plus significatif de la bascule de division.

4.3.2 - Registres d'Etats

Registre d'état de la ligne

Ce registre sert à fournir des informations relatives à la réception et à l'émission des données. La signification des bits individuels est illustrée par le tableau.5

Tableau 5 : Description du registre d'état de la ligne

Bit	Abréviation	Nom	Signification (S'il est mis à 1)
0	DR	Donnée prête	Un caractère a été reçu et placé dans le registre tampon de réception.
1	OE	Erreur de cadence	Un caractère a été reçu avant que le précédent n'ait été retiré.
2	FPE	Erreur de parité	Le bit de parité est incorrect.
3	FE	Erreur de trame	Un caractère reçu ne contient pas de bit d'arrêt valide.
4	BJ	signal break	Un break a été reçu.
5	THRE	registre de saisie de l'émetteur vide.	L'UART est prêt à recevoir un nouveau caractère pour l'émettre.
6	TSRE	Registre à décalage de l'émetteur vide réservé.	Le caractère a été émis.
7			Ce bit est forcé à zéro.

Registre d'état du modem

Ce registre donne des informations sur l'état des lignes de contrôle de flux. La signification de ses bits individuels est indiquée dans le tableau.6

Tableau 6: Description du registre d'état du modem

Bit	Nom	Signification (S'il est mis à 1)
0	DELTA CTS	Le signal CTS a changé.
1	DELTA RTS	La ligne RTS a changé.
2	TERI	L'indicateur d'appel a changé de haut vers le bas.
3	DELTA RLSD	La détection signal de la ligne de la réception a changé.
4	CTS	L'entrée prêt à émettre est au niveau haut.
5	PSR	L'entrée poste de donnée prêt est au niveau haut.
6	RI	L'indication d'appel est haut.
7	RLSD	La détection signal de la ligne de réception est haut.

Les quatre bits de poids faible indiquent qu'un changement s'est produit depuis la dernière lecture du registre, tandis que les quatre autres bits donnent l'état réel des lignes considérées.

Le registre identificateur d'interruption

Il fournit l'état courant des interruptions en attente. Le bit 0 est mis à un s'il n'y en a aucune, s'il est mis à zéro, les bits 1 et 2 indiquent laquelle est en attente selon la configuration présentée dans le tableau 7. Les bits 3 à 7 sont forcés à zéro.

Tableau 7 : Description des bits 1 et 2 du registre d'interruption

Bit 2	Bit 1	Interruption en attente
1	1	Etat de la ligne
1	0	Donnée reçue disponible.
0	1	Registre de saisie de l'émetteur vide.
0	0	Registre du modem.

4.3.3 - Registres Tampons

Les registres tampons constituent la troisième catégorie des registres de l'UART. Il y a le registre tampon de réception et le registre tampon d'émission.

Le registre tampon de réception

Le registre tampon de réception extrait le dernier caractère reçu, une fois lu, le registre d'état de la ligne indique que le tampon de réception est vide tant qu'un autre caractère n'est pas reçu.

Le registre de saisie de l'émission

Le registre de saisie de l'émetteur recueille le nouveau caractère à émettre, il est placé là par le programme.

Tous les registres internes de l'UART sont localisés à des emplacements mémoires, ci-dessous, les adresses utiles des ports et des différents registres (Tableau .8).

Tableau.8 : Registres internes de l'UART et leurs adresses mémoire

Port 1	Port 2	Registre sélectionné
3F8	2F8	Tampon de réception
3F8	2F8	Tampon d'émission
3F8	2F8	Octet fort du diviseur
3F9	2F9	Octet faible du diviseur
3F9	2F9	Registre de validation des interruptions
3FA	2FA	Registre d'identification des interruptions
3FB	2FB	Registre de contrôle de ligne
3FC	2FC	Registre de contrôle de modem
3FD	2FD	Registre d'état du modem.
3FE	2FE	Registre d'état de ligne.

5 - LES MODEMS

5.1 - Introduction

Deux équipements informatiques peuvent être reliés au moyen d'une liaison série. Cette liaison peut atteindre quelques dizaines de mètres avec un câblage direct sans problèmes apparents, mais lorsqu'on désire effectuer des liaisons plus longues, par exemple entre deux villes, on peut toujours conserver un mode de transmission série mais il faut utiliser un support mieux adapté comme le réseau téléphonique.

Un modem est le périphérique qui permet d'adapter les signaux issus d'un micro-ordinateur aux caractéristiques d'une ligne téléphonique.

5.2 - Définition d'un modem

Le terme Modem est la contraction de deux mots : Modulateur et Démodulateur. Un modem convertit les signaux numériques issus d'une liaison série d'un micro-ordinateur en signaux analogiques compatibles avec les lignes téléphoniques.

Les modems sont généralement regroupés suivant la fréquence à laquelle ils peuvent transmettre des informations.

La grandeur qui permet de définir cette fréquence est le baud. Un baud mesure le nombre de transitions rencontrées en une seconde.

Les modems sont divisés en quatre catégories :

- Les modems à basse vitesse [0,600 bauds]
- Les modems moyenne vitesse [1200,2400 bauds]
- Les modems à haute vitesse [3600,16000 bauds]
- Les modems à large bande [> 19600 bauds]



5.3 - Types de modems

5.3.1 - Modems ASK

Ce type de modem est peu répandu. L'émission d'un niveau logique 1 se traduit par l'émission d'un signal ou porteuse alors que pour un 0 logique le modem reste silencieux.

Le choix de la porteuse doit être limité dans la bande téléphonique.

5.3.2 - Modems FSK

Les modems à basse vitesse emploient généralement comme codage le FSK qui signifie "Frequency Shift keying" qui se traduit modulation par déplacement de fréquence, ils utilisent deux fréquences pour représenter les éléments binaires "1" et "0".

Comme la bande passante de la ligne téléphonique est comprise entre 300 hz et 3khz donc le choix des deux fréquences f_1 et f_2 est limité à cette bande.

Une liaison par modem doit pouvoir être bidirectionnelle et ce sans conflit. Cette bidirectionnalité est assurée en attribuant aux deux modems gérant la liaison un couple de fréquences porteuses (f_1, f_2) différent. Ce choix du plan de fréquences doit être tel qu'il ne produise aucune interaction ou diaphonie entre les canaux montant et descendant. En particulier, une fréquence ne doit pas être un multiple d'une autre fréquence afin d'éviter une intermodulation avec les fréquences harmoniques.

5.3.3 - Modems PSK

Du fait de la limitation en bande passante des lignes téléphoniques il est impossible de transmettre des informations modulées en FSK lorsque le débit est supérieur à 1200 bauds. Afin de lever cette restriction, d'autres techniques de modulation ayant un encombrement spectral plus faible sont apparus parmi lesquelles la modulation PSK. Avec la PSK les informations binaires "1" et "0" sont modulées à une fréquence porteuse fixe et sont différenciées par leur phase.

5.4 - Modes de fonctionnement d'un modem

Suivant leur mode de fonctionnement, les Modems peuvent être regroupés en trois catégories. Les modems full duplex, Half duplex et simplex. Cette classification se fait en



fonction du type de liaison retenue : Liaison bidirectionnelle simultanée, bidirectionnelle alternée et unidirectionnelle.

Suivant le mode de transmission utilisé les modems sont classés en modems synchrones, asynchrones et synchrones/Asynchrones.

CHAPITRE 4

TRANSMISSION PAR COURANT PORTEUR

1 - INTRODUCTION

La transmission par courant porteur utilise le réseau électrique existant pour la transmission des signaux. Dans ce cas un signal haute fréquence est émis à travers le réseau électrique et superposé au signal (220 volts, 50 hertz) et distribué partout dans le réseau électrique. La figure.1 montre le signal émis superposé au signal du réseau électrique.

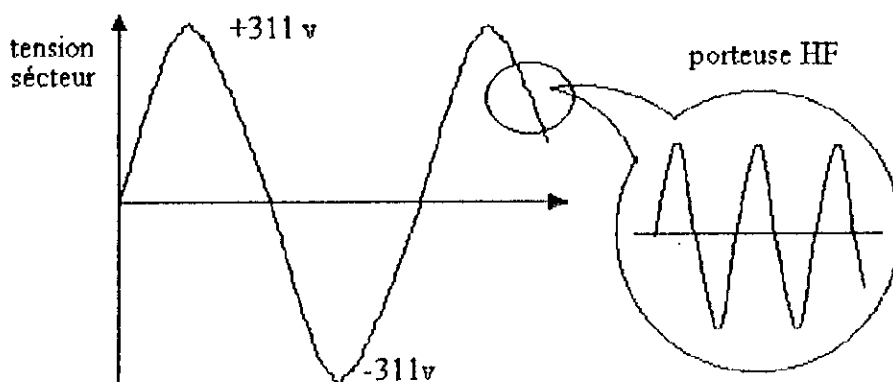


Figure.1

2.- SOURCES DE BRUITS

Compte tenu que le réseau électrique est un support destiné au transport de l'énergie électrique il présente quelques limitations :

- il présente souvent des transitoires très violentes qui peuvent perturber les signaux transmis.

- exposition permanente aux parasites générés par les équipements électriques .

- Possibilité d'interférence entre les signaux provenant de différents émetteurs.

- exposition aux bruits intenses provenant de sources différentes :

émissions radio,moto,train,etc...

- En cas d'alimentation triphasé il est impossible d'établir la liaison entre l'émetteur et le récepteur s'ils sont connectés sur des prises reliées à des phases différentes.

3 - CHOIX DU TYPE DE MODULATION

Pour transmettre de l'information numérique sur un support de mauvaise qualité comme le réseau électrique, il n'existe bien sûr qu'une méthode qui est la modulation.

Vu que le réseau électrique est sujet à des parasites très violents il faudra choisir un type de modulation qui minimise les distorsions du signal transmis. Une comparaison entre les différents types de modulations montre que la modulation de fréquence est le type de modulation le plus adéquat dans ce contexte.

4 - CHOIX DES FREQUENCES

Dans la transmission par courant porteur, l'expérience montre que les fréquences situées entre 30 khz à 500 khz offrent le meilleur compromis propagation / faible influence des parasites. L'impédance caractéristique du réseau électrique est de 600 ohm.

5 - LA MODULATION DE FREQUENCE

5.1 - Généralités

Supposons qu'on ait une porteuse de fréquence f_c et que l'on veut moduler par un signal $m(t)$. Dans le cas de la modulation de fréquence, la fréquence de la porteuse varie linéairement en fonction du signal modulant $m(t)$, la fréquence instantanée sera :

$f_c + k_f m(t)$ où k_f est une constante.

Si l'expression de la porteuse est $A_c \cos(2\pi f_c t)$, le signal modulé aura comme expression :

$$f(t) = A_c \cos \left[2\pi f_c t + 2\pi k_f \int m(t) dt \right]$$

En supposant $m(t)$ un signal de type $A_m \cos(2\pi f_m t)$, l'expression de $f(t)$ sera :

$$f(t) = A_c \cos \left[2\pi f_c t + k_f (A_m / f_m) \sin(2\pi f_m t) \right]$$

On définit la déviation maximale de fréquence par $\delta = k_f A_m$ ainsi que l'indice de modulation m_f d'un signal modulé en fréquence par $m_f = \delta / f_m$.

La déviation maximale de fréquence dépend de l'amplitude du signal modulant quelque soit le type de ce signal.

X

5.2 - Propriétés du spectre d'un signal FM

- Un signal modulé en fréquence a un nombre infini de composantes spectrales distantes de f_m , $2f_m$, $3f_m$,.... de la porteuse.

- La bande de fréquence du signal FM dépend de l'indice de modulation.

- Le spectre de fréquence présente une symétrie par rapport à la porteuse.

- Si $m_f \ll 1$ la bande de fréquence est $W = 2f_m$; f_m représente la fréquence maximale du signal modulant.

Si $1 < m_f < 20$ la bande de fréquence est $W = 2 N f_m$, où N est le nombre de raies spectrales dont l'amplitude des coefficients de Bessel demeure supérieure à 0,01.

Si $m_f > 20$ la bande de fréquence est $W = 2(\delta + f_m)$.

CHAPITRE 5

PRESENTATION DE LA REALISATION

1 - INTRODUCTION

Nous avons vu dans les chapitres précédents des généralités concernant les réseaux d'ordinateurs, l'échange de données entre équipements informatiques et la liaison série. Notre projet a pour objet de réaliser un réseau de micro-ordinateurs pour l'échange de données en utilisant comme support de communication le réseau électrique.

Le réseau conçu est organisé de telle sorte que l'un des micro-ordinateurs joue le rôle du serveur du réseau et les autres celui de stations. La figure.1 montre le raccordement des micro-ordinateurs au réseau électrique; pour établir les liaisons il faut une carte d'interface spéciale au micro-ordinateur serveur et une carte d'interface pour chaque micro-ordinateur station; ces cartes d'interface doivent adapter la nature numérique du micro-ordinateur à la nature analogique du réseau électrique.

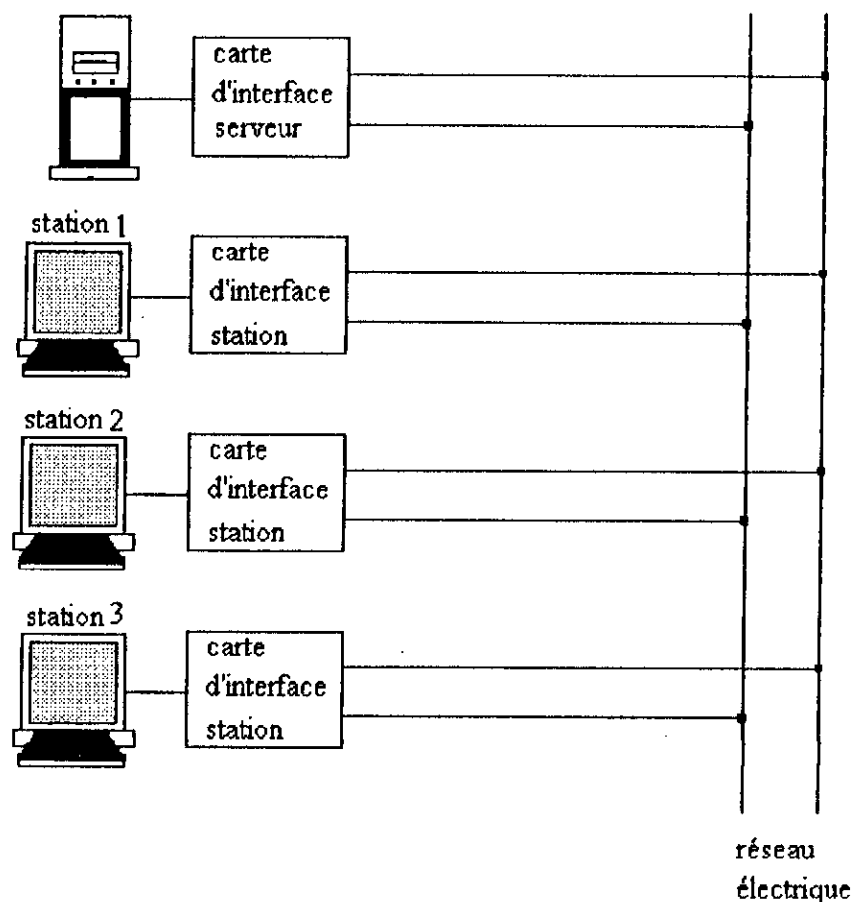


Figure.1 - Structure du réseau d'ordinateurs

X

La liaison entre le micro-ordinateur serveur et un des micro-ordinateurs stations peut être schématisée par la figure.2.

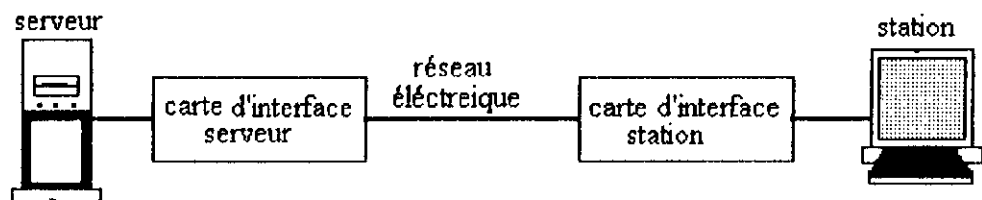


Figure.2 - Liaison entre le serveur et la station

Cette liaison permet de relier le micro-ordinateur serveur au réseau électrique par son port série à travers une carte d'interface (carte serveur); de l'autre côté, le micro-ordinateur station sera relié au réseau électrique par son port série à travers une carte d'interface (carte station).

Les deux cartes d'interface sont bâties autour du modem intégré EF7910 programmé en mode BELL 103, ce qui permet une liaison en full duplex avec un débit de 300 bits par seconde.

En plus de l'établissement de la liaison, il nous a fallu mettre au point un logiciel pour commander le fonctionnement du réseau.

2 - ETUDE DE LA CARTE D'INTERFACE

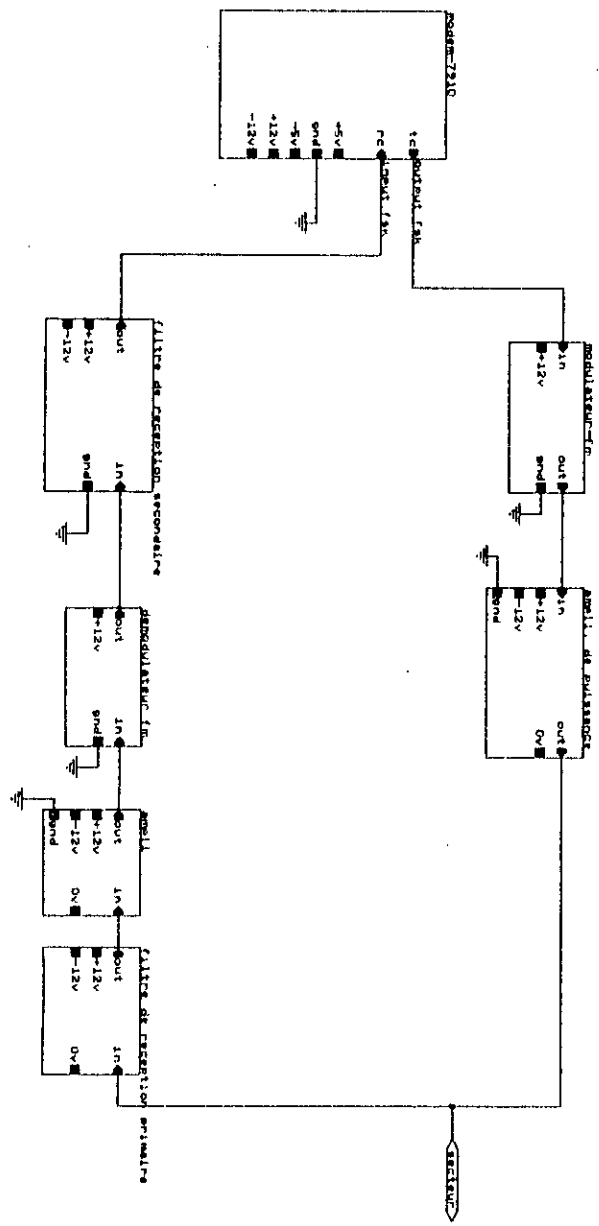
2.1 - INTRODUCTION

Le schéma synoptique de la carte d'interface est représenté par la figure.3. Cette carte peut être divisée en deux parties :

* **Partie émission** : elle est constituée des blocs suivants :

- Le modem EF7910 (partie émission).
- Le modulateur
- L'amplificateur de puissance.

X



7-114	SCHEMA SYNOPSIS DE LA CARTE D'INTERFACE
B	REVISION N°1
1	REV
1	REV

**** Partie réception :** elle est constituée des blocs suivants :

- Le filtre primaire
- L'Amplificateur
- Le démodulateur
- Le filtre secondaire
- Le modem EF7910 (partie réception).



La différence entre la carte serveur et la carte station réside dans la programmation du modem EF7910 et les fréquences d'émission et de réception choisis pour chaque carte.

2.2 - PARTIE EMISSION

2.2.1 - Le modem

Introduction

La réalisation d'un modem à partir d'éléments discrets est très délicate vue les critères qu'il doit satisfaire dans le cas de notre réalisation à savoir, une liaison en Full Duplex conformément à la norme RS232 C et un contrôle de flux de données. Ceci nous a conduit à choisir un modem intégré.

Dans notre recherche bibliographique nous avons rencontré une variété de modems mais les critères cités ci-dessus n'étaient pas complètement respectés sauf pour le EF7910 qui est capable de travailler dans les modes normalisés les plus répandus, le choix du mode de fonctionnement se fait simplement par programmation des broches prévues à cet effet.

Il dispose d'une interface aux normes RS232 C dont il sait gérer les principaux signaux automatiquement en fonction des conditions de réception. Les différentes normes acceptées par le EF7910 sont données par le tableau.1.

Le BELL 103 a été choisis car il permet un écart entre les fréquences d'émission et de réception (moins de conflit entre l'émission et la réception) tout en gardant une liaison en Full Duplex.

Tableau 1: Normes acceptées par le modem EF7910

NORME	VITESSE DE TRANSMISSION (BAUD)	DUPLEX	émission		réception	
			0	1	0	1
BELL 103 mode original	300	Full	1070	1270	2025	2225
BELL 103 mode answer	300	Full	2052	2225	1070	1270
CCITT V 21 mode original	300	Full	1180	980	1850	1650
CCITT V 23 mode answer	300	Full	1850	1650	1180	980
CCITT V 23 mode 1	600	Half	1700	1300	1700	1300
CCITT V23 mode 2	1200	Half	2100	1300	2100	1300
BELL 202	1200	Half	2200	1200	2200	1200
BELL 202 égalisé	1200	Half	2200	1200	2200	1200
CCITT V 23 back channel	75	Half	2200	1200	2200	1200
BELL 202 back channel	5	Half	450	390	450	390

Synoptique fonctionnel du EF7910

Le modem EF7910 est logé dans un boîtier 28 pattes. Il renferme deux convertisseurs (un analogique/digital et un digital/analogique), et un filtre numérique programmable associé à 1,3 Kilo bits de RAM et 24 Kilobits de ROM. Son synoptique fonctionnel est représenté en figure.4 où on y trouve les éléments essentiels d'un modem de type FSK sous forme numérique.

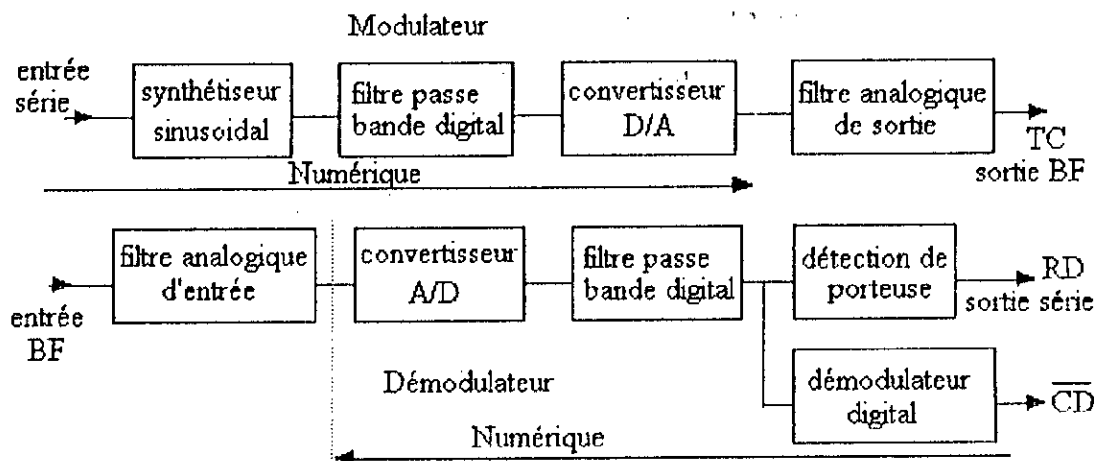


Figure.4 - Synoptique fonctionnel du EF7910

La partie modulateur est constituée par un synthétiseur sinusoïdal qui à partir des données logiques à émettre, génère une sinusoïde à la fréquence désirée cette génération a lieu sous forme numérique, c'est à dire que ce n'est pas un signal sinusoïdal qui sort de ce bloc fonctionnel mais une suite de données numériques qui représente la sinusoïde elle-même. Cette génération, étant de type discret, un filtre numérique, lui aussi, vient parfaire le signal qui est alors converti de numérique en analogique. Un filtre passe bas de sortie élimine tout résidu de signal logique et permet au circuit de délivrer une sinusoïde affûtée d'un taux de distorsion harmonique correct pour l'application envisagée.

Du côté démodulateur, après un filtre analogique passe bande d'entrée destiné à éliminer les signaux indésirables, un convertisseur analogique digital transforme le signal reçu en numérique.

Tout le traitement de démodulation a ensuite lieu en numérique grâce à un système à corrélateur.

Cette numérisation permet à ce circuit d'accepter de multiples normes de fonctionnement sans avoir à commuter le moindre composant passif externe. En effet comme tout est fait sous forme numérique en interne il suffit de changer quelques niveaux logiques sur des pattes de sélection pour adapter le 7910 aux principales normes FSK actuelles.

Synoptique interne du 7910

La figure.5 montre le synoptique interne du circuit EF7910.

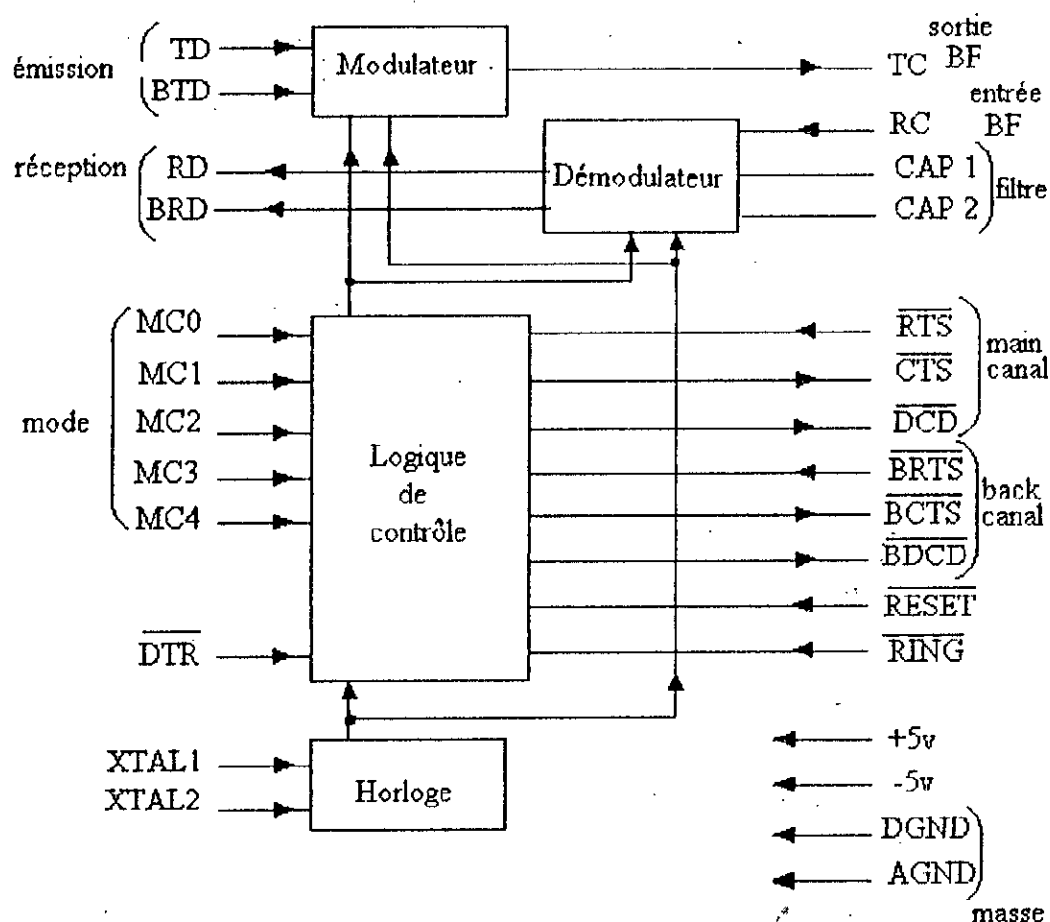


Figure.5 - Synoptique interne du EF7910

Ce circuit se compose de quatre sous ensembles fondamentaux: le modulateur transforme les données logiques en sinusoïdes BF, le démodulateur réalise l'opération inverse et le bloc logique de contrôle qui gère les signaux de l'interface RS232 C en fonction des conditions de réception et qui interprète les informations de sélection des divers modes de fonctionnement et enfin un oscillateur à quartz à partir duquel sont générées toutes les fréquences utilisées par le 7910.

Les lignes MCO à MC4 servent à sélectionner les divers modes de fonctionnement du modem selon les combinaisons du tableau.2.

X

Tableau.2 : Sélection des modes de fonctionnement du EF7910

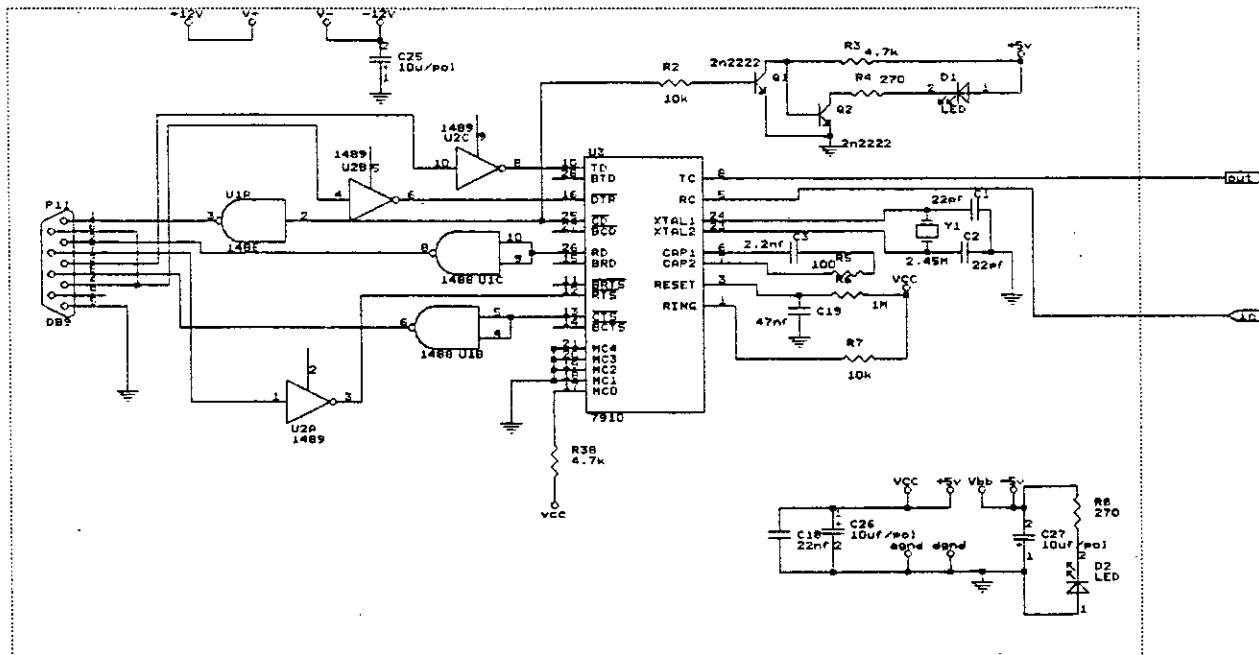
MC4	MC3	MC2	MC1	MCO	MODE
0	0	0	0	0	Bell 103 originate
0	0	0	0	1	Bell 103 answer
0	0	0	1	0	Bell 202
0	0	0	1	1	Bell 202 égalisé
0	0	1	0	0	CCITT V21 originate
0	0	1	0	1	CCITT V21 answer
0	0	1	1	0	CCITT V23 mode 2
0	0	1	1	1	CCITT V23 mode 2 égalisé
0	1	0	0	0	CCITT V21 mode 1

Schéma complet du modem

Les figures.6 et 7 montrent les deux modems utilisés dans la carte serveur et la carte station respectivement.

Le 7910 est programmé en mode Bell 103 answer dans la carte station. L'entrée RING n'est pas utilisée. Son rôle est de réaliser un modem à réponse automatique. Le 7910 nécessite une alimentation symétrique +5 Volts et -5Volts, toutes ses broches sont compatibles TTL, ses deux masses analogique (AGND) et digitale (DGND) sont indépendantes et sont généralement reliées. La broche RESET est reliée à une cellule R-C réalisant son initialisation automatique lors de la mise sous tension.

Le signal émis par le micro-ordinateur est reçu par le modem à travers l'entrée TD. Les broches DTR, CD, RTS, CTS servent au contrôle de flux de données.



SCHEMA DU MODEM POUR LA CARTE STATION
 Size Document Number: FIGURE.7 REV
 B
 Date: August 30, 1996 Sheet 67



L'adaptation du modem au micro-ordinateur (port série) se fait par les deux circuits : MC 1488 (conversion TTL vers RS232 C) et MC 1489 (conversion RS232 C vers TTL). Le quartz est de fréquence 2,4576 Mhz. Le couple de transistors Q1 et Q2 allume une led lorsque la ligne DCD est active «détection de porteur valide ».

La broche TC est la sortie du signal FSK à destination du modulateur. Ce signal est de fréquence (1070 hz pour le 0, 1270 Hz pour le 1) pour la carte serveur et de fréquence (2025 Hz pour le 0, 2225 Hz pour le 1) pour la carte station.

2.2.2 - Le modulateur FM

Introduction

Le signal FSK à la sortie du modem ne peut pas être transmis directement sur le secteur. Il doit moduler en FM une porteuse de fréquence beaucoup plus élevée pour être exploitable. Le choix de la modulation de fréquence a été déjà explicité dans le paragraphe (3) chapitre (4).

La plupart des modulateurs utilisés dans ce type de modulation font appel à la varicap qui est une diode dont la capacité de jonction est inversement proportionnelle à la polarisation inverse et utilisent des porteuses qui atteignent les Mégahertz, ce qui est loin d'être utilisable en transmission par courant porteur. Nous avons opté, pour réaliser notre modulateur, pour le circuit intégré XR2206 utilisé comme oscillateur sinusoïdal modulé par le signal issu de la sortie du modem.

Le XR 2206

Le XR 2206 est un circuit intégré utilisé dans les générateurs de signaux à basses fréquences. La figure.8 représente le schéma de branchement et le bloc diagramme interne. Le coeur de ce circuit intégré est un oscillateur commandé en tension. La fréquence de l'oscillateur est définie par la valeur du condensateur et de la résistance externes C_{ext} et R_{ext} .

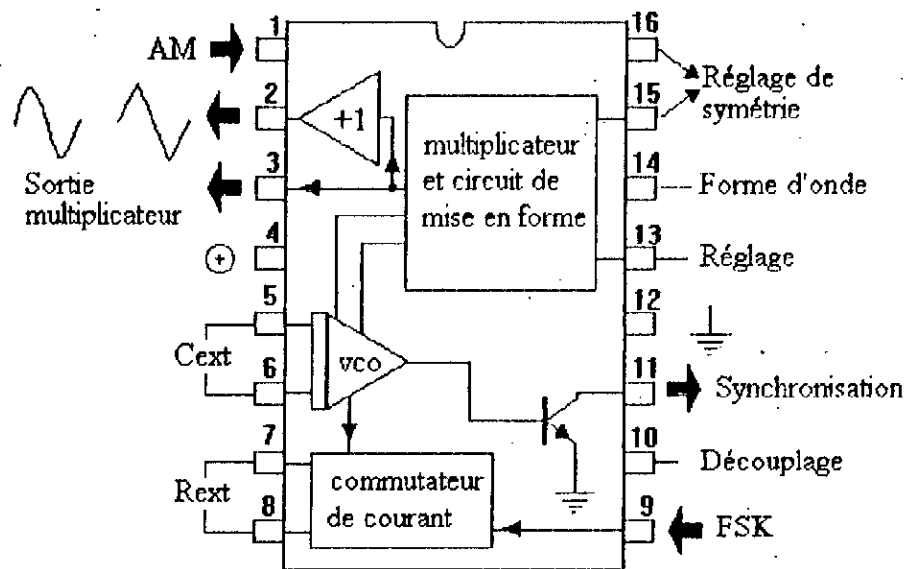


Figure.8 - Structure interne du XR2206

Un courant de commande I_f est commuté vers l'une des deux sorties de courant (broche 7 ou 8) du circuit intégré suivant l'état logique du sélecteur d'entrée (broche 9).

Les broches 13 à 16 permettent d'ajuster la portée sinusoïdale et la symétrie. Les ondes de forme sinusoïdale, triangulaire et en dent de scie sont tamponnées par un étage suiveur de tension puis transmises à la sortie basse impédance sur la broche 2.

La tension disponible entre les broches 7 et 8 est stabilisée à 3 volts à l'intérieur du circuit intégré. Le courant de commande I_f peut varier entre 1mA et 3mA.

La fréquence du VCO est définie par ce courant I_f et par la valeur du condensateur externe C_{ext} , le courant de commande étant réglé au moyen de la résistance R_f connectée entre les broches 7 ou 8 et la masse. L'équation donnant la fréquence est la suivante :

$$f = I_f / 3 C_{ext}$$

$$f = 1 / R_{ext} C_{ext}$$

Il résulte de l'équation précédente que le graphe de la fréquence en fonction de la valeur de R_{ext} n'est pas linéaire mais hyperbolique. On peut avoir une réponse approximativement linéaire en procédant comme suit :

Il existe sur la broche 7 du circuit intégré une tension constante de 3 volts. Le courant qui s'écoule depuis cette broche jusqu'à la masse est directement proportionnel à la fréquence de sortie de sorte qu'une variation linéaire de courant entraînera naturellement une variation de fréquence linéaire.

On voit sur la figure.9 que cette variation de courant est obtenue au moyen du diviseur de tension constitué par R_4 , P_1 , P_6 et R_7 , les valeurs de composants de ce diviseur sont bien choisis pour limiter la tension U_f . Ce diviseur détermine la chute de tension aux bornes de R_5 soit le courant I_f traversant cette résistance donc la valeur de la fréquence.

$$I_f = (3\text{volts} - U_f) / R_5$$

$$f = (3\text{volts} - U_f) / 3 R_5 \text{ Cext}$$

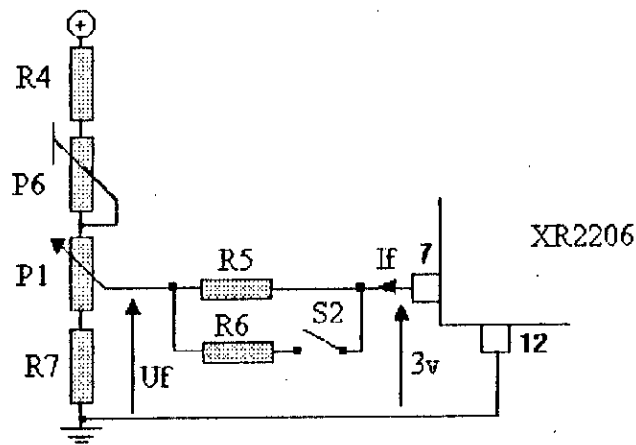


Figure.9

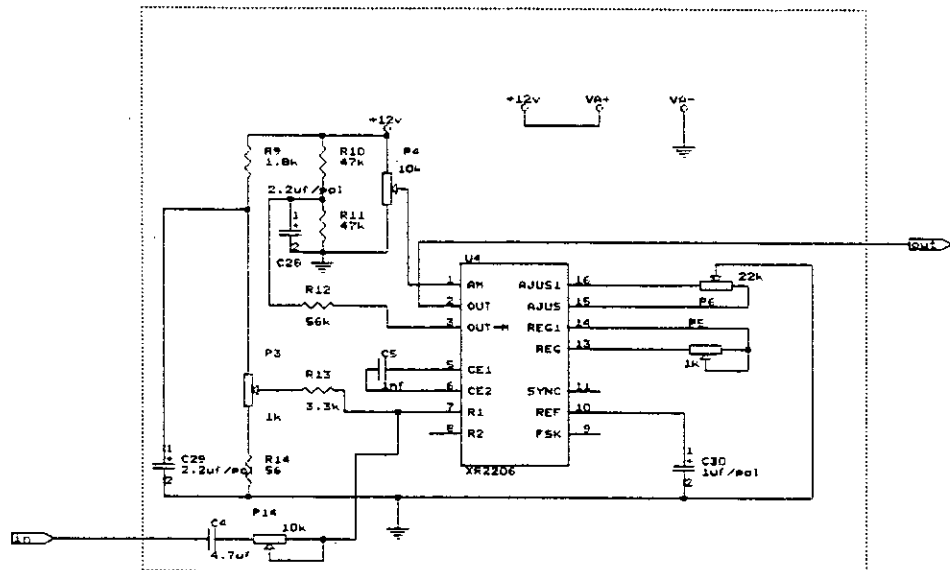
Puisqu'il existe une relation linéaire entre la chute de tension et le courant, il est possible d'obtenir à l'aide d'un potentiomètre linéaire une graduation linéaire du réglage de la fréquence.

Procédure de modulation

Le schéma du modulateur de fréquence est représenté par la figure.10, le coeur de ce modulateur étant le XR 2206. La broche 2 de ce circuit est la sortie du modulateur, l'ensemble R_9 , R_{14} , R_{13} , P_3 et C_5 définissent la fréquence de sortie en l'absence de signal modulant. Les condensateurs C_{28} , C_{29} et C_{30} sont des condensateurs de découplage.

La composante continue du signal de sortie est fixé à 6 volts au moyen du pont diviseur R_{10} , R_{11} , R_{12} alors que l'ajustable P_4 permet de varier l'amplitude du signal de sortie. La distorsion est réglée par le biais de l'ajustable P_5 .

2



Title		
SCHEMA DU MODULATEUR DE FREQUENCE		
Size	Document Number	REV
8	FIGURE 10	
Date:	August 30, 1995	Sheet 07

X

La modulation se fait en modifiant la fréquence de sortie du XR 2206 en agissant sur le courant de commande I_f . Comme la tension présente à la broche 7 est réglée à 3 volts, la seule limitation de fréquence (sans présence de modulation) se fait à travers l'ensemble R_9, R_{14}, R_{13}, P_3 , et C_5 .

Lorsque le signal modulant (FSK) est appliqué au circuit, les débits provoqués par ce signal s'ajoute au débit R_{13} via P_{14} . La fréquence d'oscillation est donc modifiée. C_4 a pour rôle de bloquer une composante continue éventuelle qui risquerait de provoquer un décalage symétrique de la fréquence centrale. P_{14} permet de régler le taux de modulation.

Pour la carte serveur le signal modulant est un signal FSK de fréquence (1070 Hz, 1270 Hz). La porteuse FM est réglée à 200 KHz par le choix de C_5, R_{14}, R_9, R_{13} et le réglage de P_3 .

Pour la carte station, le signal modulant est un signal FSK de fréquence 2025 hz-2225 hz. La porteuse FM est réglée à 50 KHz par le choix de C_5, R_9, R_{13}, R_{14} le réglage de P_3 .

2.2.3 - L'Amplificateur de puissance

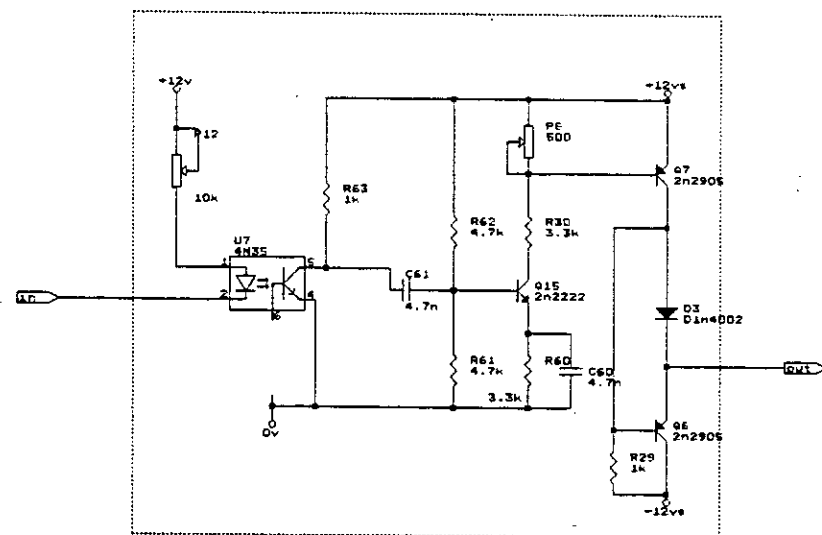
Introduction

Pour que le signal transmis à travers le secteur puisse atteindre la carte de destination, il faut que sa puissance soit suffisante pour parcourir la distance qui sépare les deux cartes dans les conditions les plus extrêmes. Cette puissance ne peut en aucun cas être atteinte sans l'adjonction d'un amplificateur de puissance.

Un amplificateur de puissance à transistor a été choisi car il allie simplicité, grande puissance et large bande passante. Celui ci est constitué d'un coupleur optique, un préamplificateur et d'un étage de sortie.

Le coupleur optique

Pour des raisons de sécurité, autant pour l'utilisateur que pour le matériel, une isolation entre le secteur et la partie de la carte reliée au micro-ordinateur doit être établie. Pour cela,



Title		
SCHEMA DE L'AMPLIFICATEUR DE PUISSANCE		
Size Document Number		
B	FIGURE 11	REV
Date: August 30, 1986 Sheet 3 of 8		

nous avons utilisé un coupleur optique à six broches constitué d'une LED (émettrice du signal optique) et d'un phototransistor (Récepteur du signal optique).¹

La figure.11 montre la structure interne du photocoupleur et son branchement dans l'amplificateur de puissance. L'ajustable P_{12} polarise le coupleur optique.

Le signal issu de la sortie du modulateur de fréquence attaque la cathode de la LED, sera reçu par le phototransistor et transmis vers le préamplificateur.

Le préamplificateur

Le signal à la sortie du coupleur optique est atténué. Cette atténuation est due à la faible bande passante du coupleur optique et au spectre relativement important du signal modulé en fréquence. Pour pallier à ce problème, ce signal est amplifié par un étage émetteur commun bâti autour du transistor Q_{15} comme indiqué par la figure.11.

L'étage de sortie

Cet étage est constitué du transistor Q_7 monté en source de courant, du transistor Q_6 et de la diode D_3 . La polarisation de la base de Q_7 est assurée par l'ajustable P_8 . Le courant de sortie de Q_7 est déterminé par la tension présente à sa base qui est commandée à travers R_{30} par le préamplificateur comme indiqué par la figure.11.

Lorsque la tension du secteur est inférieure de 0,7 volts ou plus à la tension de sortie de Q_7 , la diode D_3 conduit et Q_6 reste bloqué jusqu'à ce que la tension du secteur dépasse cette limite. A ce moment, la diode D_3 devient bloquée et Q_6 devient une charge au secteur.



¹ Voir la référence [6]

2.3 - PARTIE RECEPTION

2.3.1 - Le filtre primaire

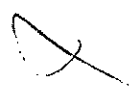
Introduction

Le Signal émis par chaque carte d'interface est modulé en fréquence. Le spectre de ce signal est symétrique par rapport à la fréquence porteuse. Pour pouvoir éliminer le maximum de parasites en dehors de la bande utilisée, un filtre passe bande est nécessaire. Parmi ces derniers, il existe le circuit (L-C), les filtres (R-C) ainsi que les filtres actifs.

Notre choix s'est orienté vers les filtres actifs pour leur simplicité de mise au point et de réglage. Le 741 s'est avéré intéressant pour la réalisation de notre filtre. Malgré qu'il présente une bande passante limitée, on a pu surmonter ce problème en faisant un compromis entre le gain du filtre à la fréquence centrale et sa bande passante.

Description du filtre primaire

Le filtre primaire est constitué de deux filtres passe bande identiques montés en série pour améliorer la réponse en fréquence comme indiqué par la figure.12.



Le filtre passe bande utilisé est représenté par la figure.13. Il est caractérisé par sa fréquence centrale f_0 , sa bande passante $\Delta\omega$ et son gain à la fréquence centrale $G(f_0)$.

Ceux-ci sont données en fonction des éléments du filtre par les formules suivantes:

$$G(f_0) = R_3 / R_1$$

$$f_0 = [R_3 R_1 R_2 / (R_1 + R_2)]^{1/2} / 2 \pi C$$

$$\Delta\omega = 2 / C R_3$$

En fixant $G(f_0)$, $\Delta\omega$ et f_0 , on peut déterminer les éléments du filtre passe bande qui sont donnés par les formules suivantes:

$$R_1 = 1 / C G(f_0) \Delta\omega^2$$

$$R_2 = \Delta\omega / [2 \omega_0^2 - G(f_0) \Delta\omega^2] C$$

$$R_3 = 2 / C \Delta\omega$$

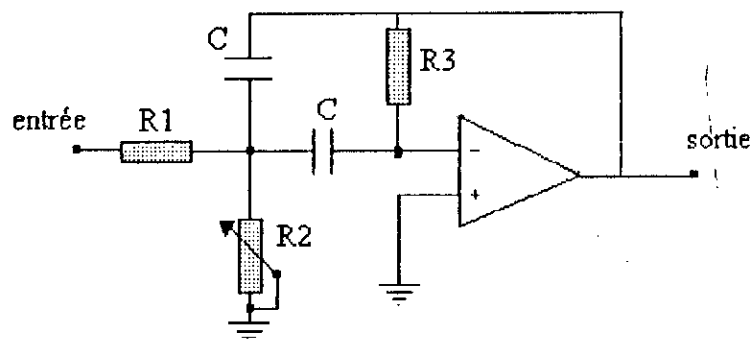
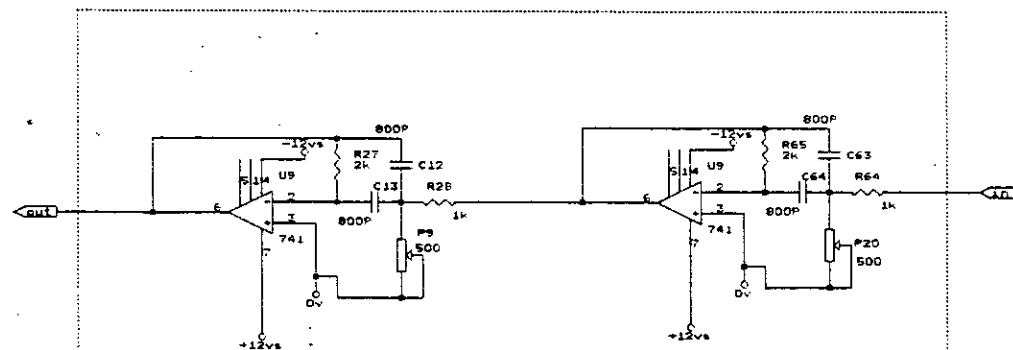


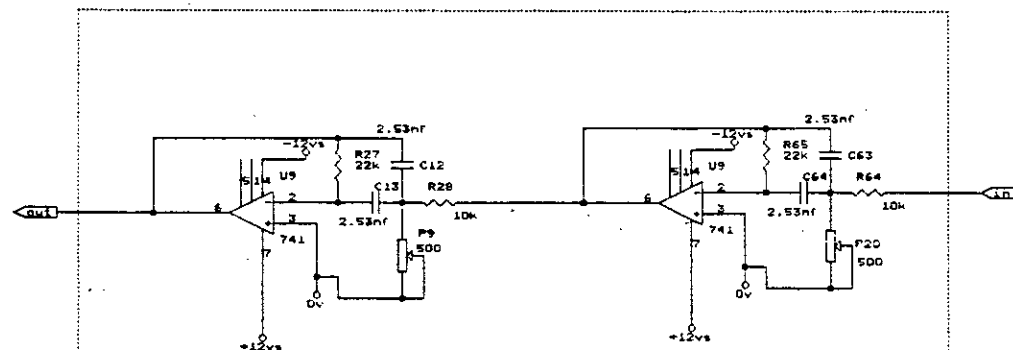
Figure.13 - Schéma du filtre passe bande

Pour assurer un réglage fin de la fréquence de résonance, nous avons remplacé R_2 par un ajustable.

Le filtre primaire de la carte serveur est accordé sur la fréquence 50 KHz et reçoit le signal modulé en fréquence, issu de la carte station. Celui de la carte station est accordé sur la fréquence 200 KHz et reçoit le signal modulé en fréquence, de porteuse 200 KHz, émis par la carte serveur. La bande passante du filtre est approximativement 10 kHz



SCHEMA DU FILTRE PRIMAIRE POUR LA CARTE STATION



SCHEMA DU FILTRE PRIMAIRE POUR LA CARTE SERVEUR

Title		
SCHEMA DU FILTRE PRIMAIRE		
Size Document Number		
B	FIGURE 18	REV
Date: August 30, 1988 Sheet 01		

2.3.2 - L'amplificateur

Introduction

Le signal en provenance de la ligne électrique est atténué et bruité. Après l'avoir filtré à travers le filtre de réception primaire, il faudra l'amplifier à un niveau acceptable. Cette tâche est confiée à un amplificateur constitué de deux étages (comme indiqué par la figure.14).

- un amplificateur de liaison
- un amplificateur d'entrée

L'amplificateur de liaison

Le signal en provenance du filtre de réception primaire est injecté à la base de Q_5 via le condensateur C_{11} qui doit être bien choisi pour que l'étage n'entre pas en oscillation. La résistance de l'émetteur est découplée par le condensateur C_{10} , C_{31} sert à découpler la tension d'alimentation.

Le transistor Q_4 monté en collecteur commun réalise une adaptation d'impédance entre Q_5 et le coupleur optique (voir la figure.14).

L'amplificateur d'entrée

Comme pour le cas de l'émission, pour isoler le micro-ordinateur du secteur, nous avons fait appel au coupleur optique 4N35. Le signal à la sortie du coupleur optique est injecté à la base du transistor Q_3 à travers le condensateur C_{21} . (voir la figure.14).

A la sortie de l'amplificateur le signal est d'un niveau qui permet d'attaquer le trigger de Schmitt.

2.3.2 - L'amplificateur

Introduction

Le signal en provenance de la ligne électrique est atténué et bruité. Après l'avoir filtré à travers le filtre de réception primaire, il faudra l'amplifier à un niveau acceptable. Cette tâche est confiée à un amplificateur constitué de deux étages (comme indiqué par la figure.14).

- un amplificateur de liaison
- un amplificateur d'entrée

L'amplificateur de liaison

Le signal en provenance du filtre de réception primaire est injecté à la base de Q_5 via le condensateur C_{11} qui doit être bien choisi pour que l'étage n'entre pas en oscillation. La résistance de l'émetteur est découplée par le condensateur C_{10} , C_{31} sert à découpler la tension d'alimentation.

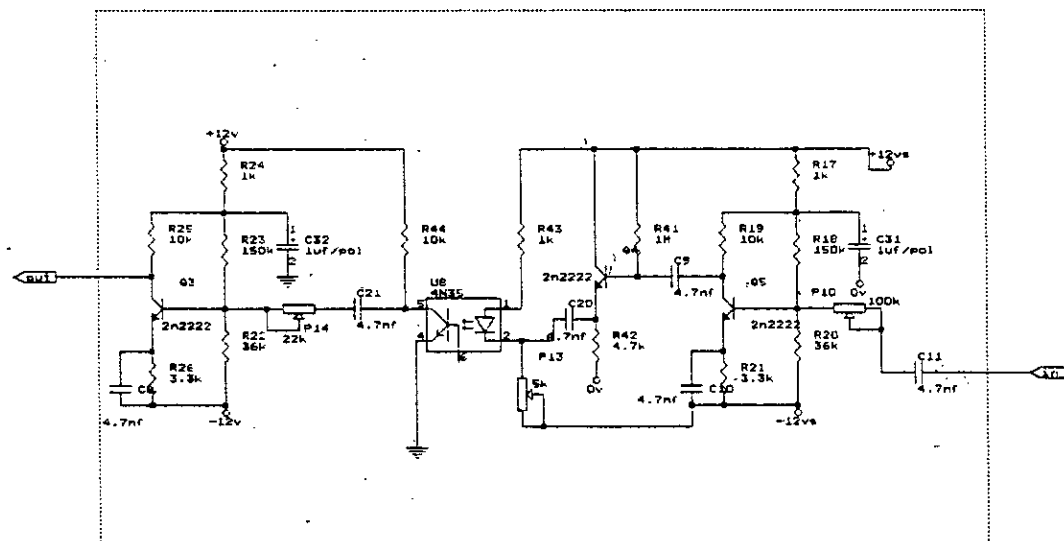
Le transistor Q_4 monté en collecteur commun réalise une adaptation d'impédance entre Q_5 et le coupleur optique (voir la figure.14).

L'amplificateur d'entrée

Comme pour le cas de l'émission, pour isoler le micro-ordinateur du secteur, nous avons fait appel au coupleur optique 4N35. Le signal à la sortie du coupleur optique est injecté à la base du transistor Q_3 à travers le condensateur C_{21} . (voir la figure.14).

A la sortie de l'amplificateur le signal est d'un niveau qui permet d'attaquer le trigger de Schmitt.

Q



Title		
SCHEMA DE L'AMPLIFICATEUR		
Size/Document Number		
B FIGURE 14		
Date: August 30, 1988 Sheet 4 of 8		

2.3.3 - Le démodulateur

Introduction

Pour démoduler un signal modulé en fréquences plusieurs solutions sont possibles, la première est descripteur Foster-seely qui est un dispositif complexe, il est de moins en moins utilisé car sa sensibilité aux fluctuations d'amplitude est importante. En conséquence on doit le faire précéder d'un circuit limiteur.

Le deuxième système est le détecteur de rapport faisant lui aussi appel à des couplages magnétiques de réalisation très délicate.

C'est la raison pour laquelle nous avons opté pour la PLL (phase Locked loop) appelée boucle à verrouillage de phase. Le circuit HEF4046 a été choisis parmi les circuits PLL pour sa simplicité d'emploi.

Mise en forme

Pour que le signal issu de l'amplificateur puisse être démodulé par la PLL, il lui faut une mise en forme pour le rendre compatible avec le CMOS. Cette fonction a été réalisée par un trigger de Schmitt en utilisant des portes logiques NAND (circuit 4011).

Présentation de la PLL

On appelle PLL un système bouclé d'asservissement de phase dans lequel la grandeur asservie est la phase d'un signal périodique.

La figure.15 représente le schéma fonctionnel d'une PLL.

Les éléments constitutifs sont :

- Un comparateur
- Un filtre passe bas
- Un oscillateur contrôlé en tension



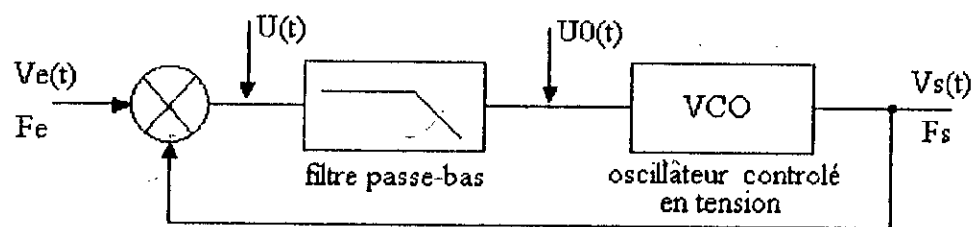


Figure.15 - Schéma fonctionnel de la PLL

Le comparateur de phase

Ce circuit compare la phase de deux signaux alternatifs ou considérés comme tels, et fournit une tension moyenne d'erreur $U(t)$ proportionnelle à leur déphasage ϕ lorsque la boucle est verrouillée soit $f_s = f_e$.

La tension $U(t)$ récupérée à partir d'un comparateur de phase dans les conditions de verrouillage comprend :

- Des harmoniques de fréquence $2f_e$ et plus.
- Une composante continue U_0 à l'image du déphasage.

Le filtre passe bas

La tension $U(t)$ est inutilisable à cause de ses harmoniques. Il faut supprimer ceux-ci afin de ne conserver que la composante continue $U_0(t)$. Deux types de filtres peuvent être utilisés, les filtres passifs et les filtres actifs.

L'oscillateur contrôlé

C'est un oscillateur fournissant une onde alternative, dont la fréquence varie sous l'action d'une tension ou d'un courant. L'oscillateur oscille à f_0 lorsque $U_0(t) = V_{DD} / 2$ (V_{DD} est la tension d'alimentation):

Il existe deux plages caractéristiques de fonctionnement correct de la PLL en boucle fermée : la plage de verrouillage et la Plage de capture

Handwritten mark resembling a stylized 'X' or a signature.

Plage de verrouillage

Elle correspond à l'écart de fréquence par rapport à la fréquence centrale f_0 à partir duquel la PLL se déverrouille.

Plage de capture

Elle correspond à l'écart de fréquence par rapport à la fréquence centrale à partir duquel la PLL se verrouille.

Présentation du HEF 4046

Le HEF 4046 est un circuit intégré PLL en technologie CMOS. Il comprend un oscillateur linéaire commandé en tension (VCO), deux comparateurs de phase différents ayant un amplificateur d'entrée de signaux commun et une entrée de comparateur commune. Une diode régulatrice (Zener) de 7 Volts assure la régulation de la tension d'alimentation. La fréquence maximale de fonctionnement du HEF 4046 est de 2 Mhz. La figure.16 montre le schéma du HEF 4046.

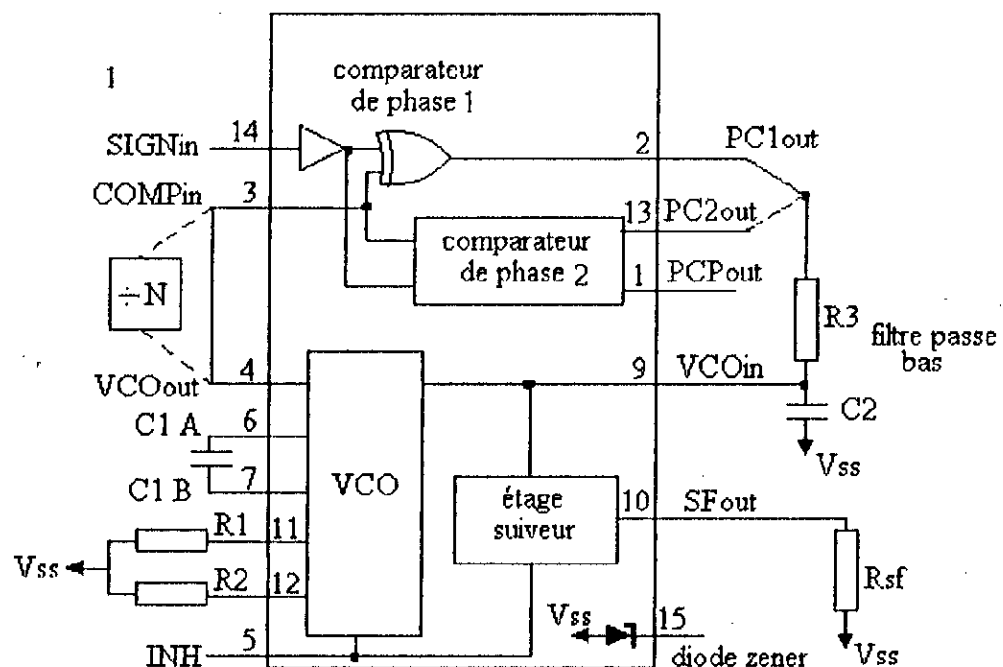


Figure.16 - Schéma fonctionnel du HEF 4046

Partie VCO

Le VCO nécessite un condensateur externe C1 et une ou deux résistances externes R1 ou R1 et R2. La résistance R1 et le condensateur C1 déterminent la plage de fréquence du VCO. La résistance R2 permet au VCO des déviations de fréquence éventuelles.

L'impédance d'entrée élevée du VCO simplifie la conception du filtre passe bas. Un niveau bas à la borne (5) valide le VCO et l'étage suiveur, tandis qu'un niveau haut les bloque.

Le comparateur de phase

Le comparateur de phase 1 est un circuit ou exclusif, la fréquence du signal d'entrée à la borne 14 doit avoir un rapport cyclique de 50 % pour obtenir la plage de verrouillage maximale. Le comparateur de phase 2 est un réseau de mémoires numérique commandées par des fronts d'impulsions. Son utilisation nécessite une étude approfondie.

Le filtre passe bas

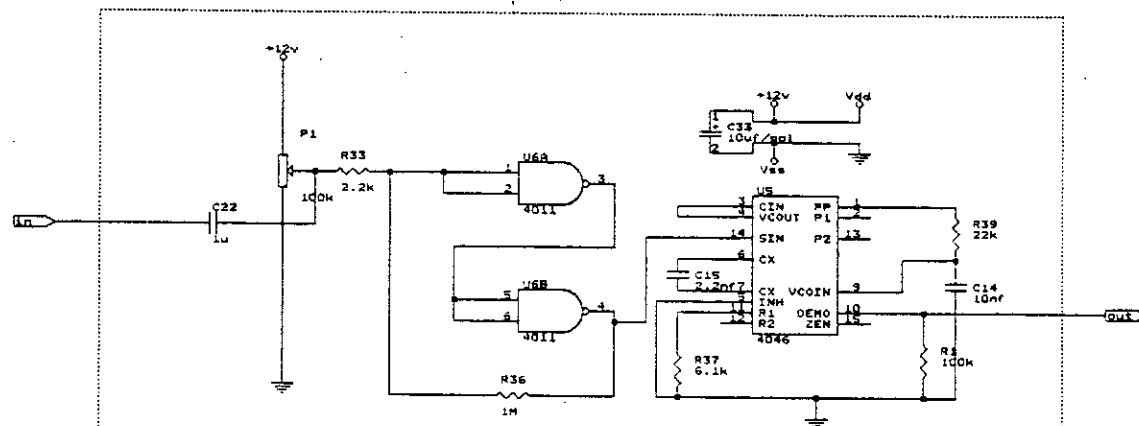
Le filtre passe bas est constitué d'un filtre R₃-C₂ dont la fonction de transfert est donnée par :

$$U_o(P) / U(p) = 1 / (1 + R_3 C_2 P)$$

Procédure de démodulation

Le schéma du démodulateur est donné par la figure.17. Le signal issu de la sortie du trigger de Schmitt attaque l'entrée du comparateur de phase 1 qui compare la phase de ce signal avec celle du VCO. A sa sortie, nous obtenons un signal comprenant la somme des fréquences d'entrée et du VCO ainsi que la fréquence différence. Le filtre passe bas élimine la fréquence somme et transmet la fréquence différence si celle-ci se situe sous sa fréquence de coupure. Ainsi, toute différence entre la fréquence du VCO et la fréquence d'entrée se traduit par une tension d'erreur qui commande le VCO et ajuste la période de ce dernier pour la rendre égale à la période du signal d'entrée. A ce moment, la boucle est verrouillée. A la sortie du filtre passe bas, nous obtiendrons le signal démodulé qui est le signal FSK émis par le modem.

Dans la carte serveur le VCO oscille en l'absence de signal à la broche 14, à la fréquence 50 Khz qui est la porteuse émise par la carte station. Tandis que dans la carte station le VCO oscille, en l'absence de signal à la broche 14, à la fréquence 200 Khz qui est la porteuse émise par la carte serveur (voir annexe 1 pour le calcul des fréquences).



$R37=6.1k; C15=2.2nf \rightarrow f_{vco}=50khz$
 $R37=1.13k; C15=2.2nf \rightarrow f_{vco}=200khz$

Titre	SCHEMA DU DEMODULATEUR DE FREQUENCE
Size	Document Number
3	FIGURE 17
Date:	August 30, 1996 Sheet 5 of 8

Le signal FSK à la sortie du démodulateur est de fréquence 1070 hz pour le 0 et 1270 hz pour le 1 pour la carte station et est de fréquence 2025 hz pour le 0 et 2250 hz pour le 1 pour la carte serveur.

2.3.4 - Le filtre secondaire

Le signal à la sortie du démodulateur attaque un filtre passe bande bâti autour du 741 comme indiqué par la figure.18. Les éléments de ce filtre (R_{15} , P_2 , R_{16} , C_6 , C_7) sont calculés comme pour le cas du filtre primaire. Ce filtre est destiné à fournir au modem un signal FSK pur.

Pour la carte serveur, le filtre de réception secondaire est accordé sur 2125 hz (fréquence centrale entre 2025 hz et 2225 hz), tandis que, pour la carte station, il est accordé sur 1170 hz (fréquence centrale entre 1070 hz et 1270hz). La bande passante est de 700 Hz calculée selon la formule du paragraphe (6.2) chap (1), avec $D = 300$ bits /s et $\Delta f = 100$ Hz.

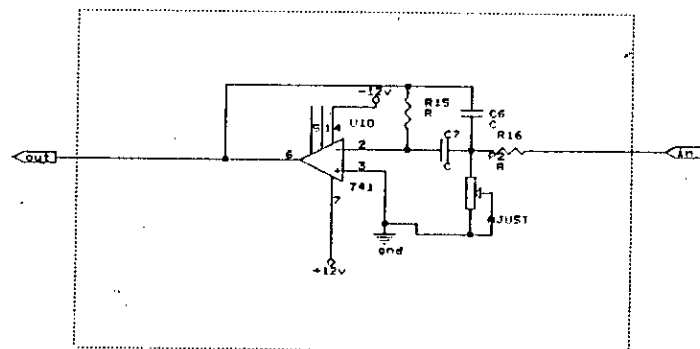


Figure.18 - Schéma du filtre secondaire

2.3.5 - Le modem EF7910(Partie réception)

Le signal à la sortie du filtre est injecté à l'entrée RC du modem qui le démodule et en obtient, à la sortie RD, un signal binaire qui est converti à la norme RS232 C par le MC1488 et transmis vers le port série du micro-ordinateur.

3 - SCHEMA COMPLET DE LA CARTE SERVEUR

Le schéma complet de la carte reliée au micro-ordinateur serveur est donné par la figure.19 (voir annexe 2). Le modem 7910 est programmé par les pattes MCO à MC4 en mode Bell 103 originate (appellant).

Dans la partie émission, le modem convertit le signal binaire de son entrée provenant du micro-ordinateur en un signal FSK de fréquence 1070hz pour le 0 et 1270 hz pour le 1. Ce signal module la porteuse de 200 khz générée par le XR 2206.

Le signal modulé, en fréquence attaque un coupleur optique (4N35) qui isole le micro-ordinateur du secteur. Il est transmis vers l'amplificateur de puissance puis à travers le secteur via deux condensateurs de capacité bien choisis (C_{16}, C_{52}).

Dans la partie réception, le signal émis par la carte station, et modulé en fréquence de porteuse 50 khz, est reçu et transmis via les capacités (C_{16}, C_{52}) au filtre de réception primaire accordé sur 50 khz. Il attaque ensuite un amplificateur qui l'élève à un niveau acceptable. Le micro-ordinateur est isolé du secteur par le coupleur optique 4N35. Le signal est ensuite mis en forme par un trigger de Schmitt (4011) et démodulé par le 4046 dont le VCO est accordé sur 50 khz.

Ensuite le signal passe par un filtre secondaire (filtre passe bande) accordé sur 2125 hz et qui fournit à sa sortie le signal FSK de fréquence 2025 hz pour le 0 et 2225 hz pour le 1. Ce signal attaque l'entrée RC du modem 7910 pour être démodulé et transmis vers le micro-ordinateur à travers le MC1489.

4 - SCHEMA COMPLET DE LA CARTE STATION

Le schéma complet de la carte reliée au micro-ordinateur station est donné par la figure.20 (voir annexe 3) . Le modem est programmé par les pattes MC0 à MC4 en mode Bell 103 answer (appelé).

Dans la partie émission, le modem convertit le signal binaire de son entrée provenant du micro-ordinateur en un signal FSK de fréquence 2025 hz pour le 0 et 2225 hz pour le 1. Ce signal module la porteuse de 50 khz générée par le XR2206

Le signal modulé en fréquence attaque un coupleur optique (4N35) qui isole le micro-ordinateur du secteur et est transmis vers l'amplificateur de puissance puis à travers le secteur via deux condensateurs de capacité bien choisis (C16 ,C52).

Dans la partie réception, le signal émis par la carte serveur, et modulé en fréquence de porteuse 200 khz, est reçu et transmis au filtre de réception primaire accordé sur 200 khz. Il attaque ensuite un amplificateur qui l'élève à un niveau acceptable. Le micro-ordinateur est isolé du secteur par le coupleur optique 4N35. Le signal est ensuite mis en forme par un trigger de schmitt (4011) et démodulé par le 4046 dont le VCO est accordé sur 200 khz. Ensuite, il passe par un filtre passe bande accordé sur 1170 hz qui fournit le signal FSK de fréquence 1070 hz pour le 0 et 1270 hz pour le 1.

Ce signal attaque l'entrée RC du modem 7910 pour être démodulé et transmis vers le micro-ordinateur à travers le MC 1488.

5 - CIRCUITS IMPRIMES

Les deux circuits imprimés de la carte serveur et de la carte station sont représentés par les figures.21 et 22 respectivement. Ces deux circuits sont simple face et nécessitent l'adjonction de quelques straps.

La réalisation des deux circuits imprimés a été faite à l'aide du logiciel ORCAD.

L'emplacement des composants dans chaque carte est donné par les figures.23 et 24.



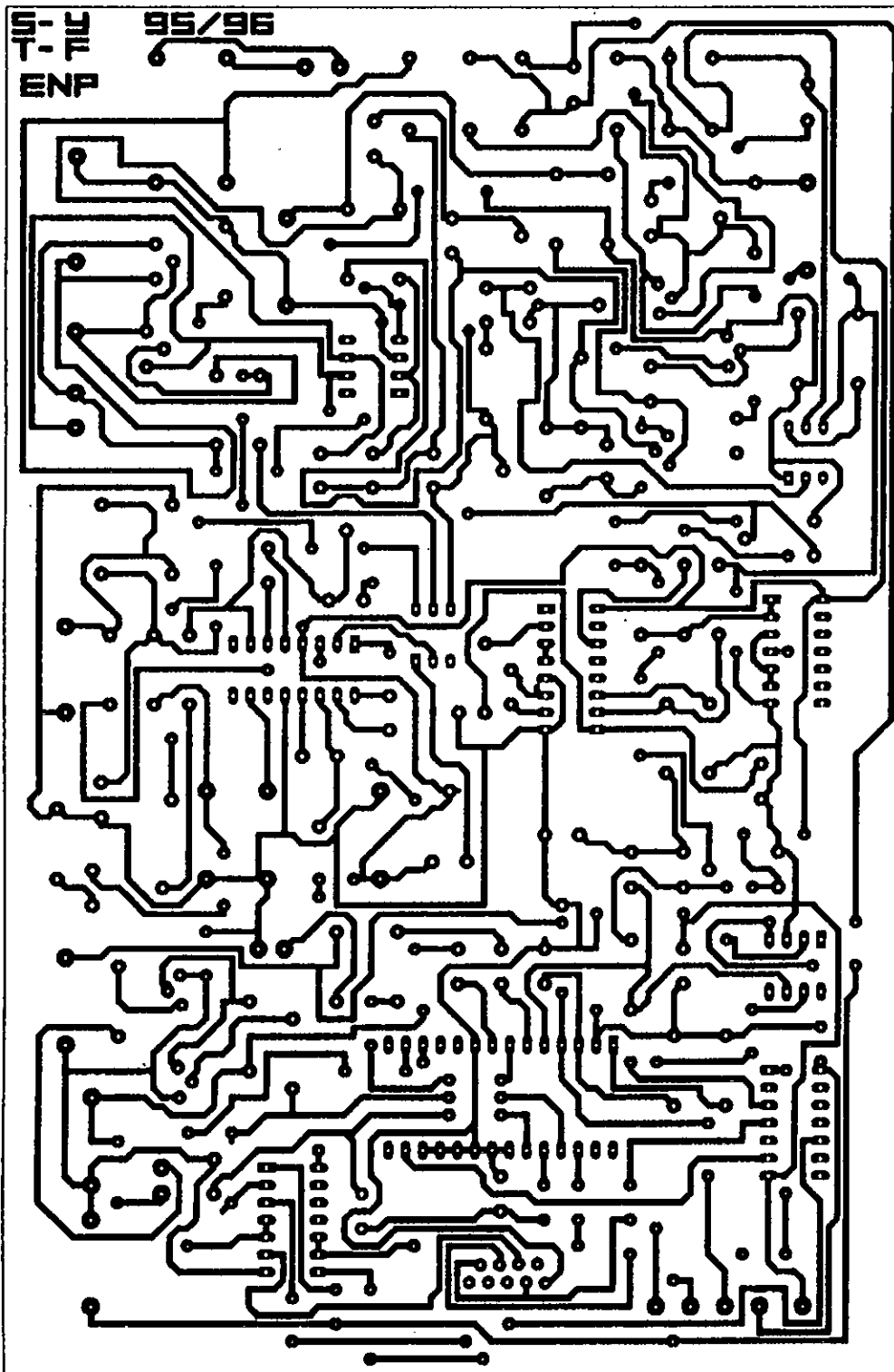
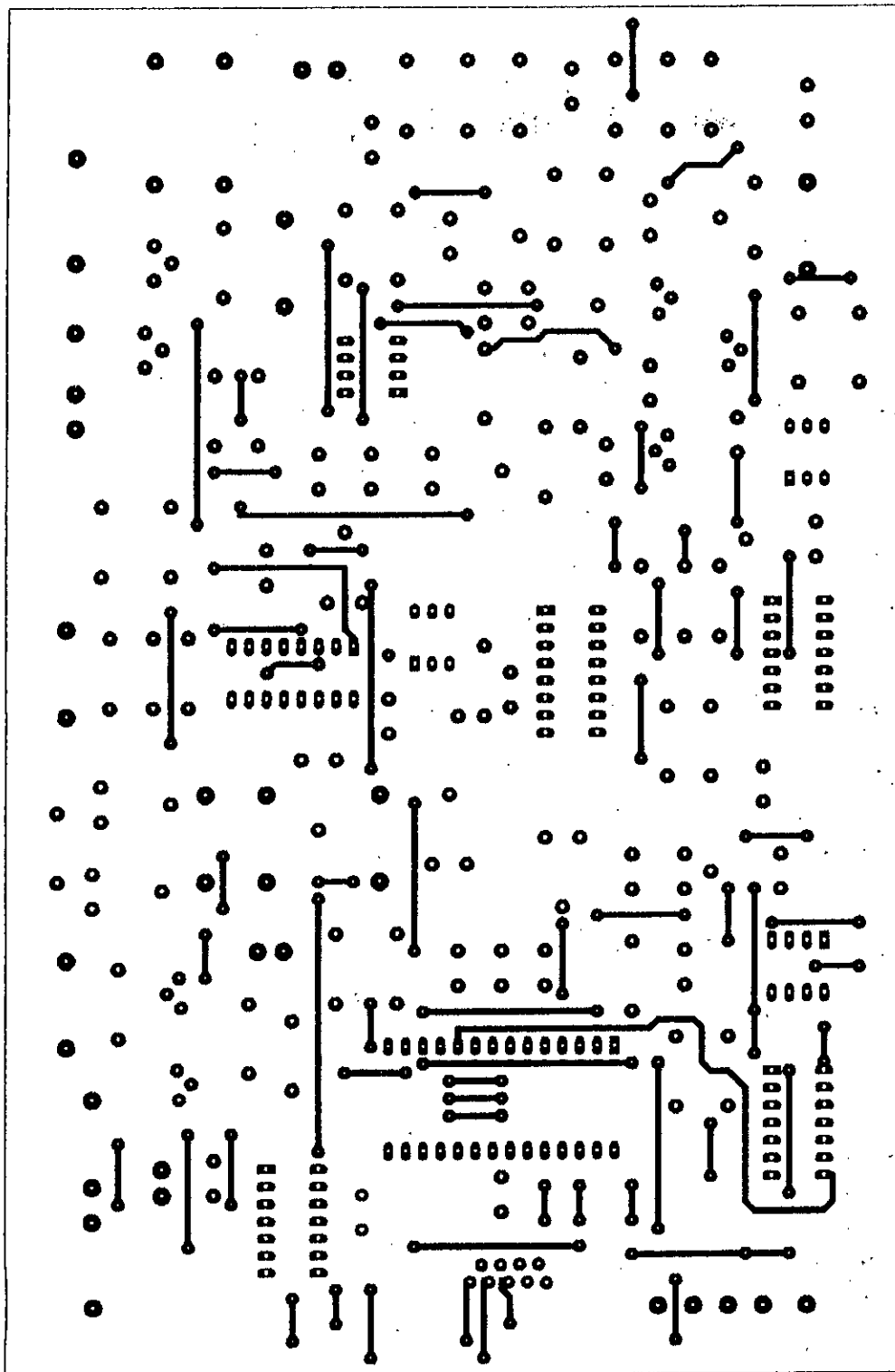


Figure. 21

✓



2

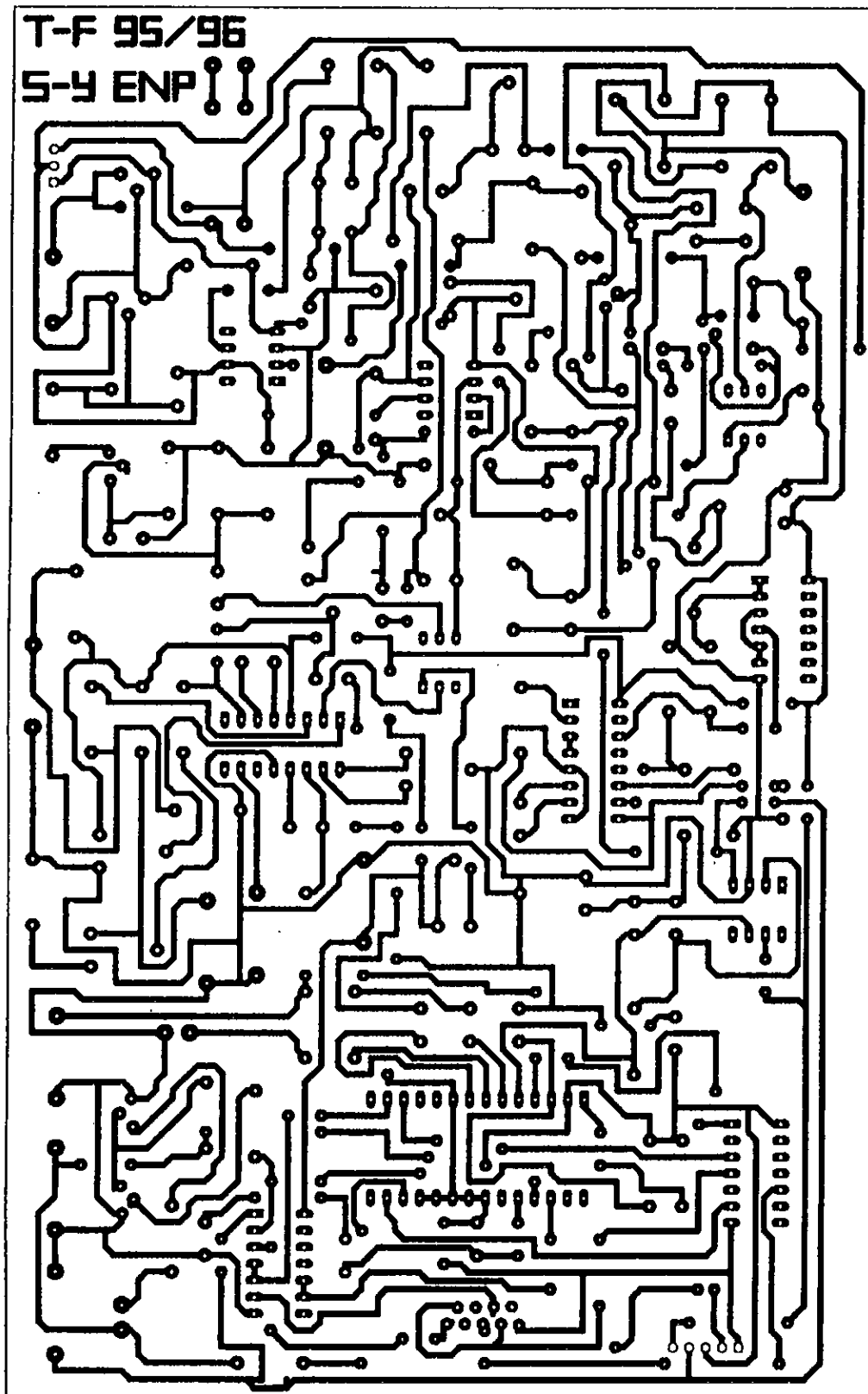
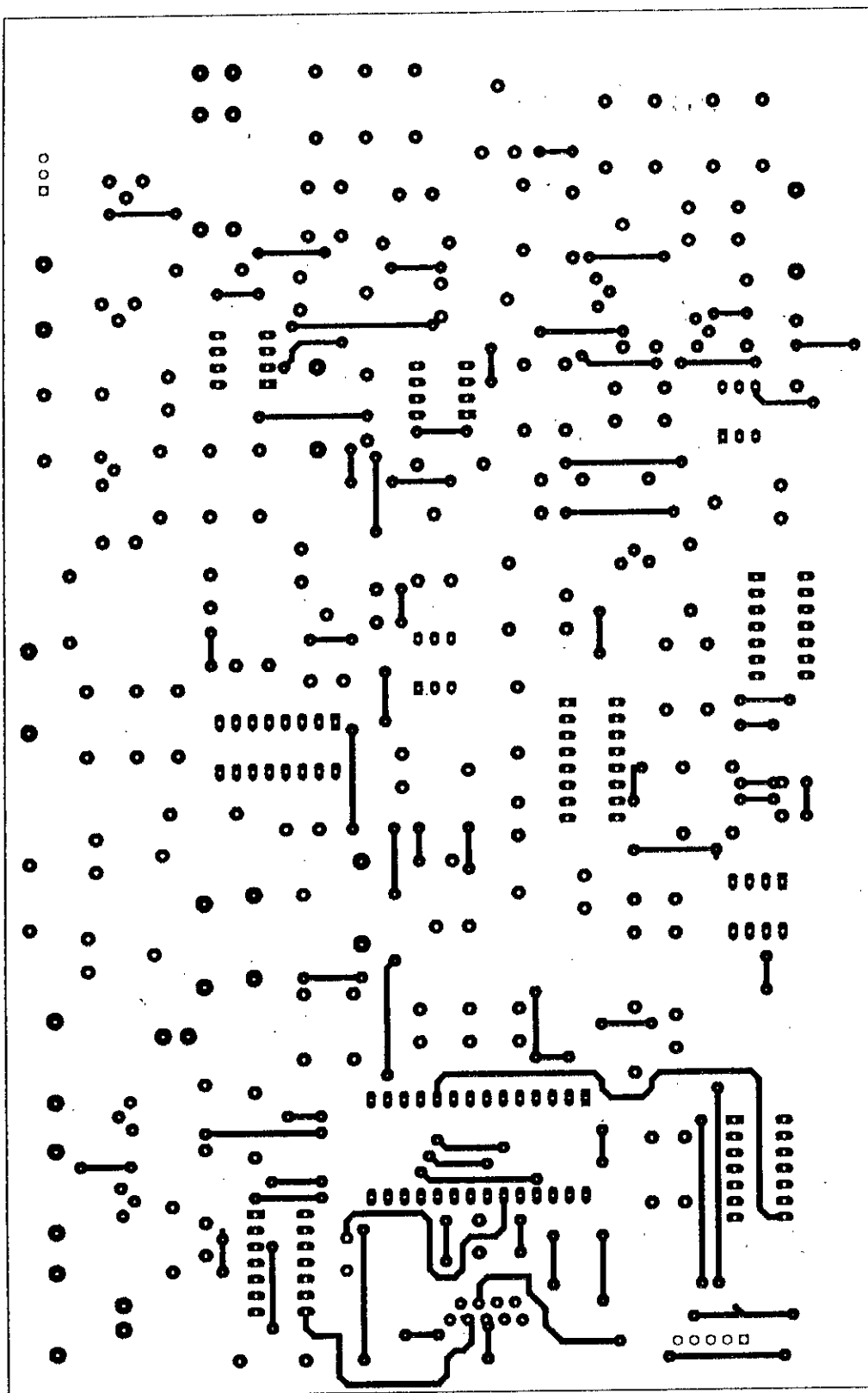


Figure. 22

X



8

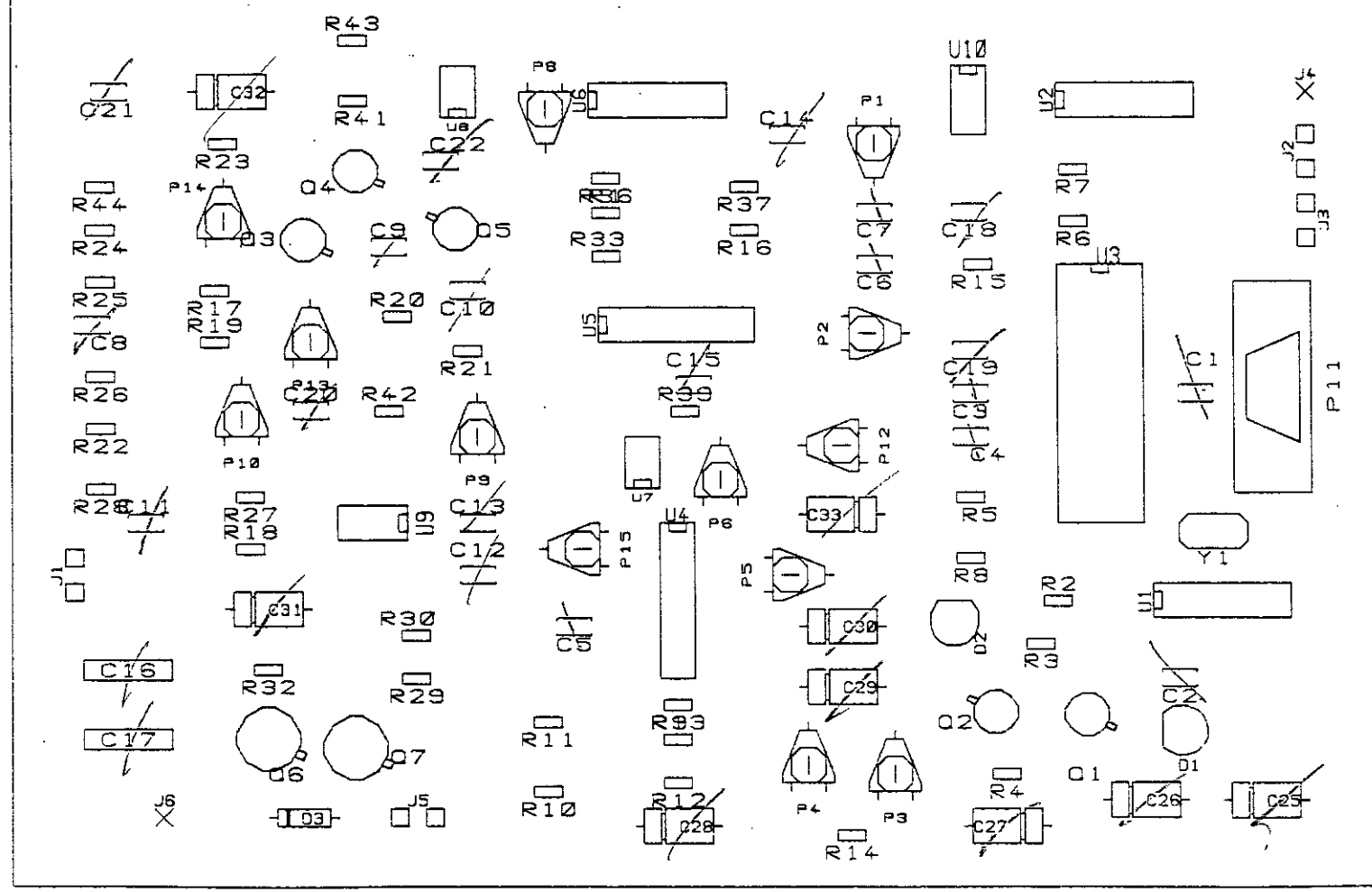


Figure 23

X

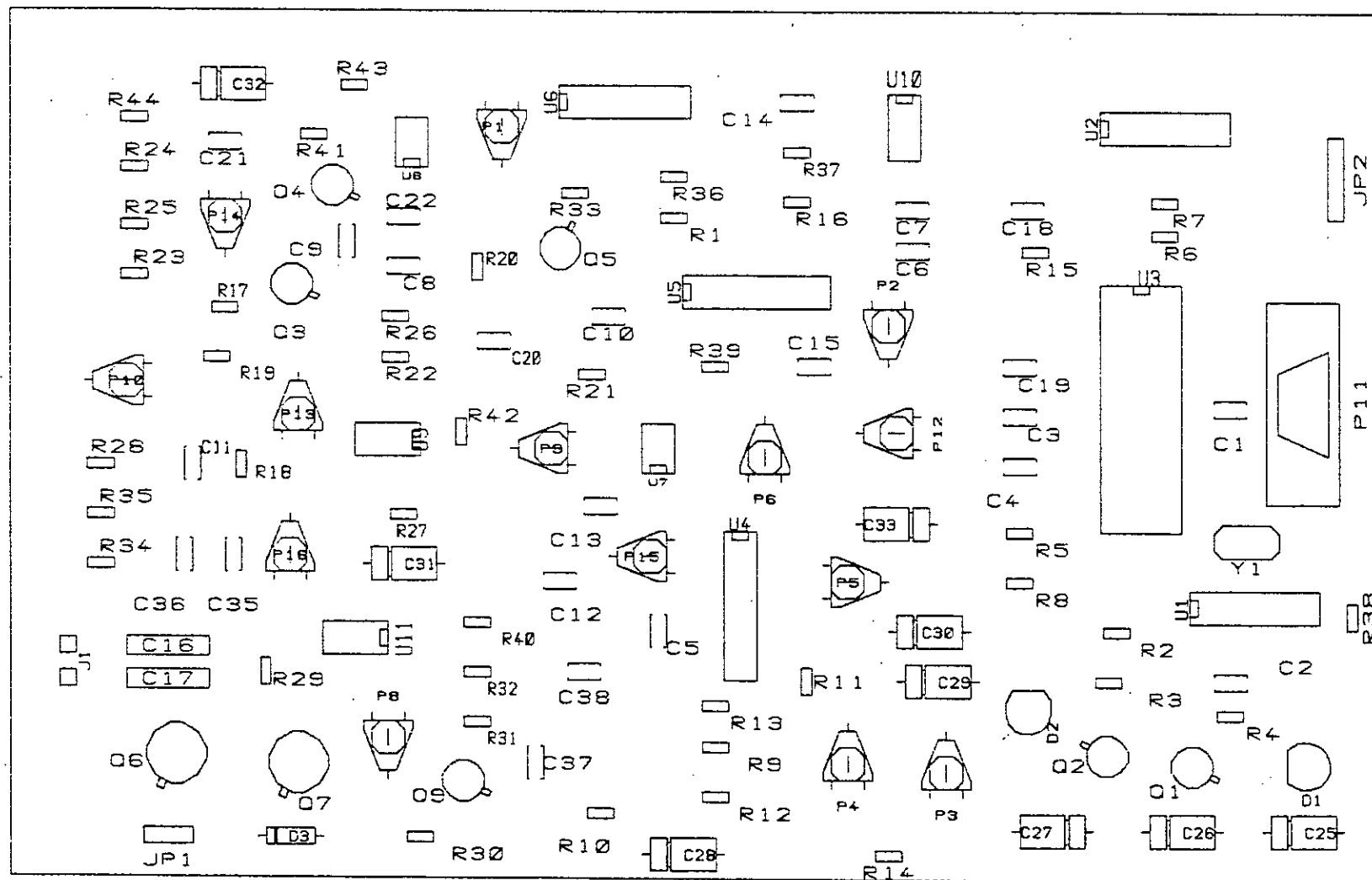


Figure. 24

6 - LA SIMULATION

Pour vérifier la fonctionnalité des étages constituant la carte d'interface avant sa mise au point, nous avons fait appel à la simulation à l'aide du logiciel SPICE.

6.1 - Simulation de l'amplificateur de puissance

La figure 25 montre l'analyse temporelle des signaux d'entrée et de sortie de l'amplificateur de puissance.

Le signal d'entrée est vu au niveau de la cathode de la led du coupleur optique, celui de sortie au niveau de la cathode de la diode D1.

6.2 - Simulation du filtre primaire

Les figures.26 et 27 montrent l'analyse fréquentielle des deux filtres de réception primaires, le premier accordé sur la fréquence 200 khz, le second accordé sur la fréquence 50 khz.

6.3 - Simulation de l'amplificateur

La figure.28 représente l'analyse temporelle des signaux d'entrée et de sortie de l'amplificateur.

Le signal d'entrée est vu au niveau de la capacité C6, celui de sortie au niveau du collecteur de Q1.

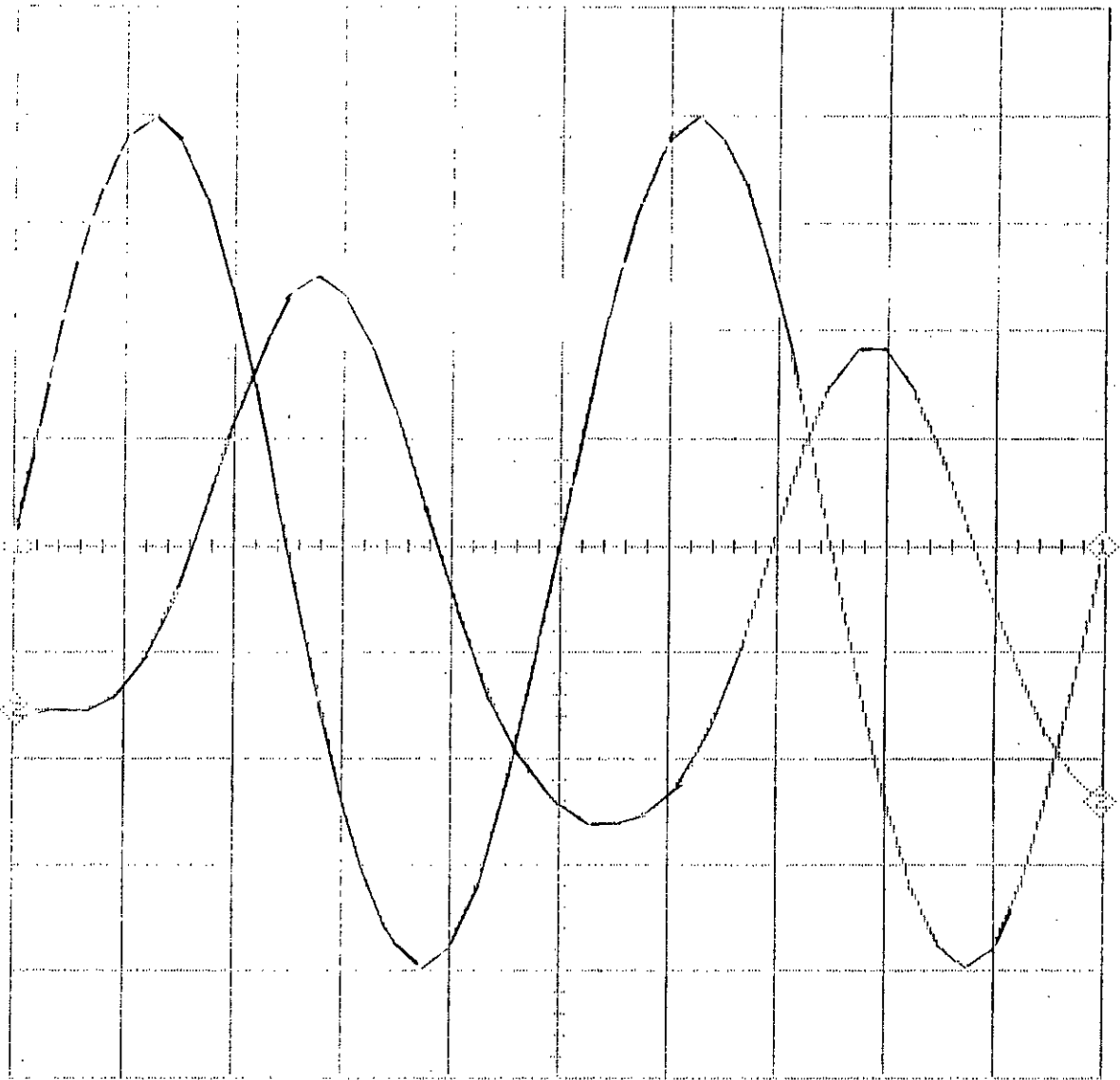
6.4 - Simulation du filtre secondaire

Les figures.29 et 30 montrent l'analyse fréquentielle des deux filtres secondaires

Le premier est accordé sur 2125 hz avec une bande passante de 700 hz, le second sur 1170 hz avec une bande passante de 700 hz également.

6.5 - Conclusion

La réalisation pratique de ces différents étages a donné approximativement les mêmes résultats que ceux de la simulation.



CH 1. V(10) vs. TIME	CURSOR	LEFT	RIGHT	DIFFERENCE
YSCALE 500MV/DIV				
YZERO 0.00E0V	VER	-0.00E0V	-3.12MV	-3.12MV
XSCALE 2USEC/DIV				
XZERO 10.00USEC	HOR	-36.9FSEC	19.9USEC	19.9USEC

CH 2. V(7) vs. TIME	CURSOR	LEFT	RIGHT	DIFFERENCE
YSCALE 2V/DIV				
YZERO 1.70 V	VER	1.70 V	-3.63 V	-1.73 V
XSCALE 2USEC/DIV				
XZERO 10.00USEC	HOR	-36.9FSEC	19.9USEC	19.9USEC

Figure.25 Analyse temporelle de l'amplificateur de puissance

Date/Time run: 05/16/96 16:11:30

* C:\CONCEPT_E\WSPICE\FILT200.sch

Temperature: 27.

(A) C:\CONCEPT_E\WSPICE\FILT200.DAT

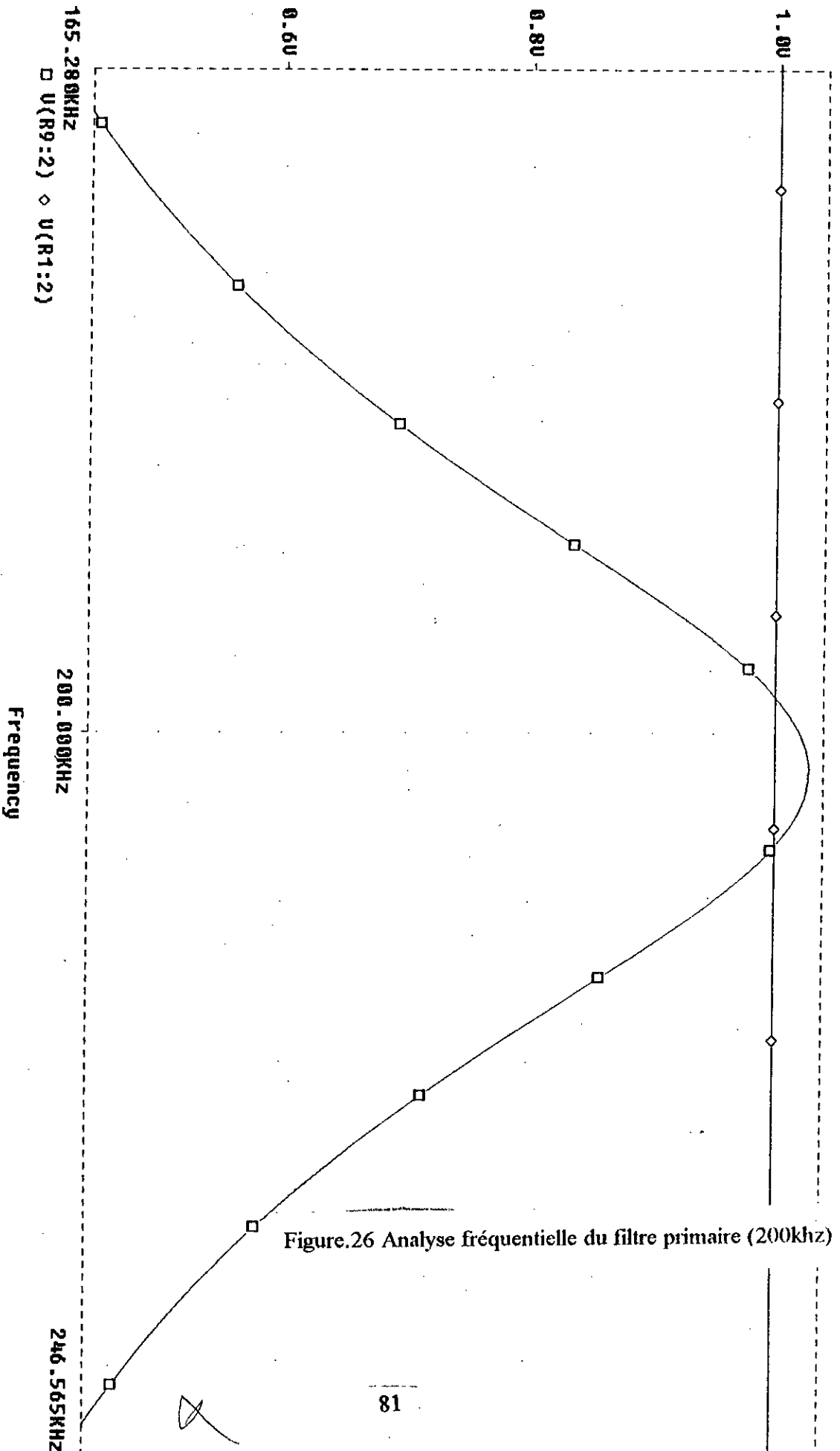
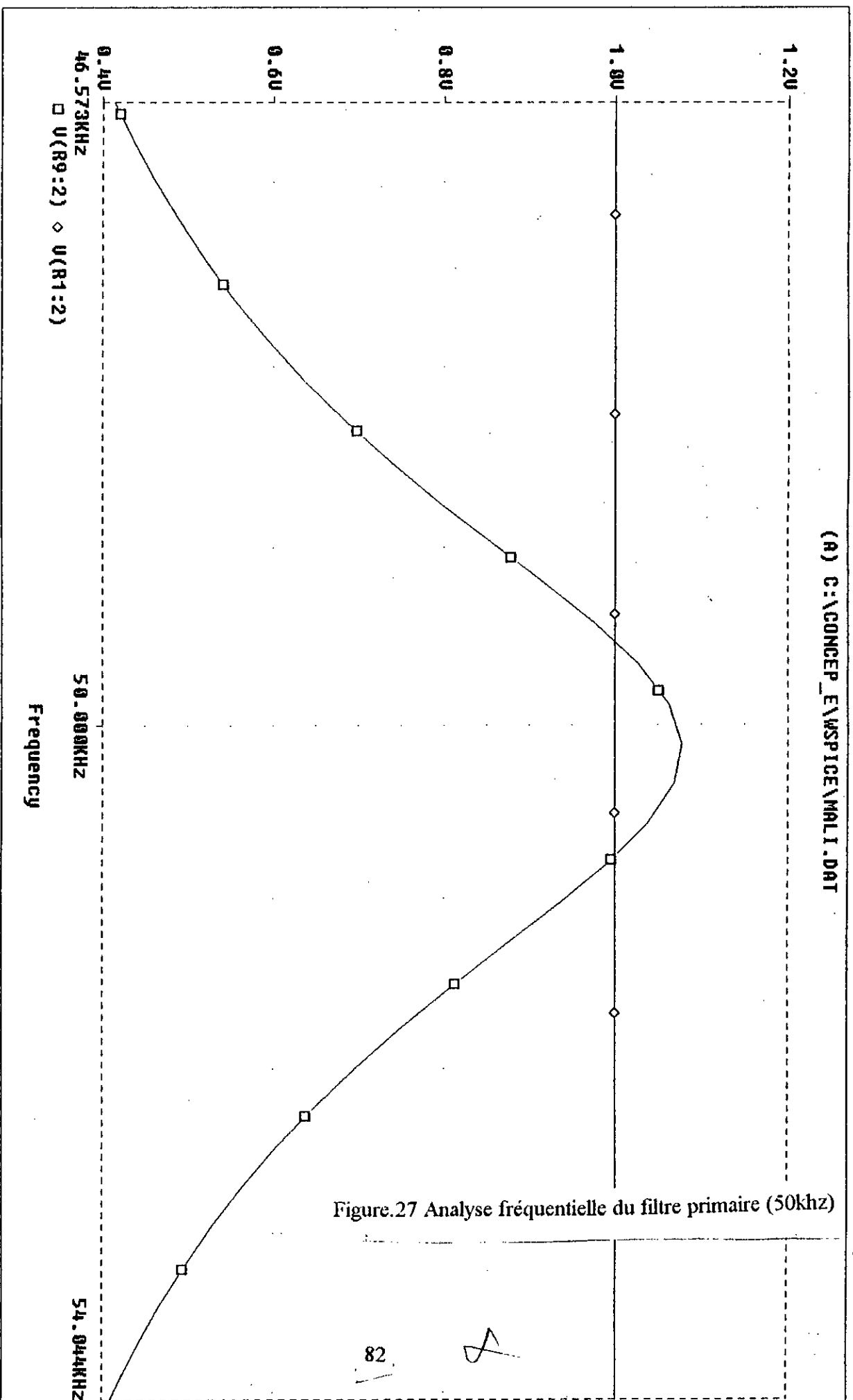
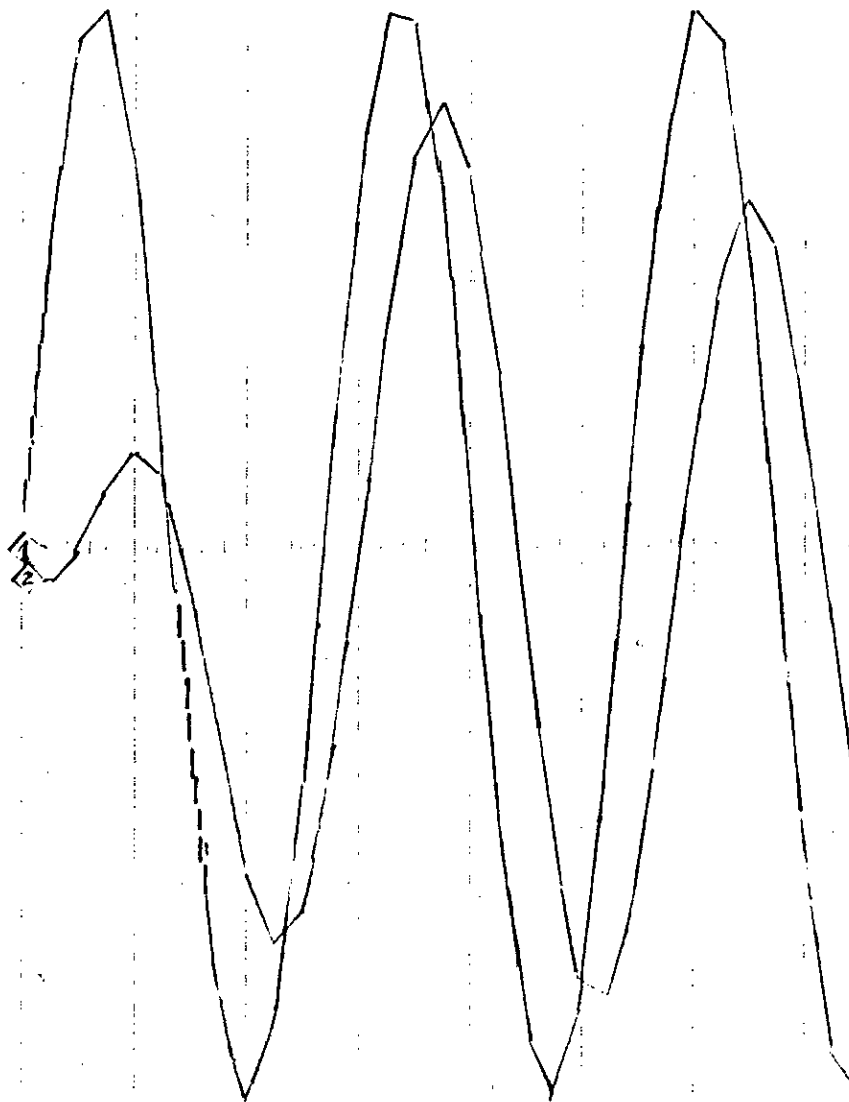


Figure.26 Analyse fréquentielle du filtre primaire (200khz)

(A) C:\CONCEPT\E\WSPICE\MALI.DAT

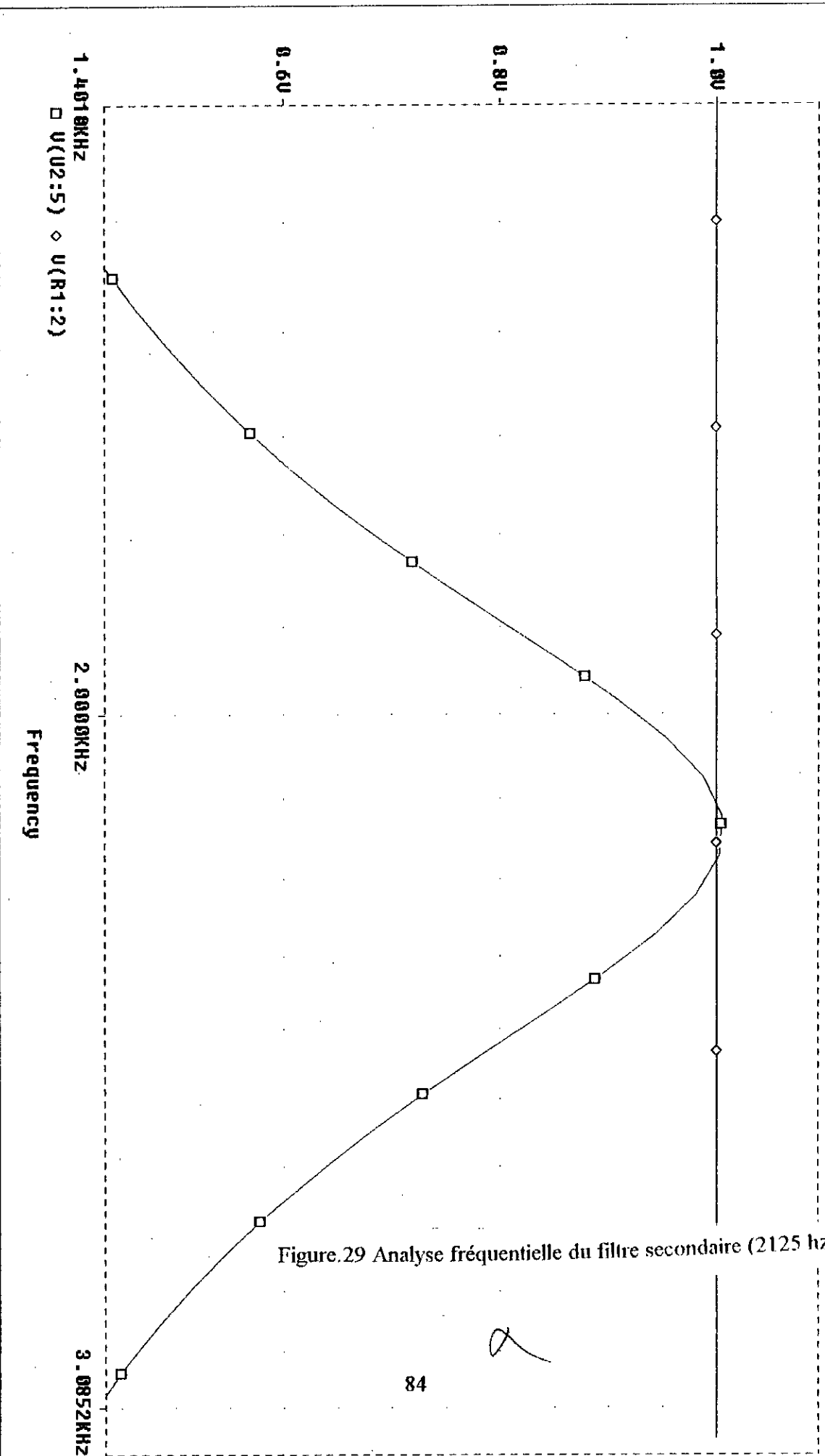




CH 1, V(1) VOLTAGE	CH 2, V(2) VOLTAGE	CH 3, V(3) VOLTAGE	CH 4, V(4) VOLTAGE	CH 5, V(5) VOLTAGE
SCALE 200mV/DIV	SCALE 200mV/DIV	SCALE 200mV/DIV	SCALE 200mV/DIV	SCALE 200mV/DIV
WATER 0.000000	WATER 0.000000	WATER 0.000000	WATER 0.000000	WATER 0.000000
WATER 0.000000	WATER 0.000000	WATER 0.000000	WATER 0.000000	WATER 0.000000
WATER 0.000000	WATER 0.000000	WATER 0.000000	WATER 0.000000	WATER 0.000000
CH 2, V(2) VOLTAGE	CH 3, V(3) VOLTAGE	CH 4, V(4) VOLTAGE	CH 5, V(5) VOLTAGE	CH 6, V(6) VOLTAGE
SCALE 200mV/DIV	SCALE 200mV/DIV	SCALE 200mV/DIV	SCALE 200mV/DIV	SCALE 200mV/DIV
WATER 0.000000	WATER 0.000000	WATER 0.000000	WATER 0.000000	WATER 0.000000
WATER 0.000000	WATER 0.000000	WATER 0.000000	WATER 0.000000	WATER 0.000000
WATER 0.000000	WATER 0.000000	WATER 0.000000	WATER 0.000000	WATER 0.000000

Figure.28 Analyse temporelle de l'amplificateur

(A) C:\CONCEPT\E\WSPICE\FILTRE11.DAT



(A) C:\CONCEPT_E\WSPICE\FILTRE11.DAT

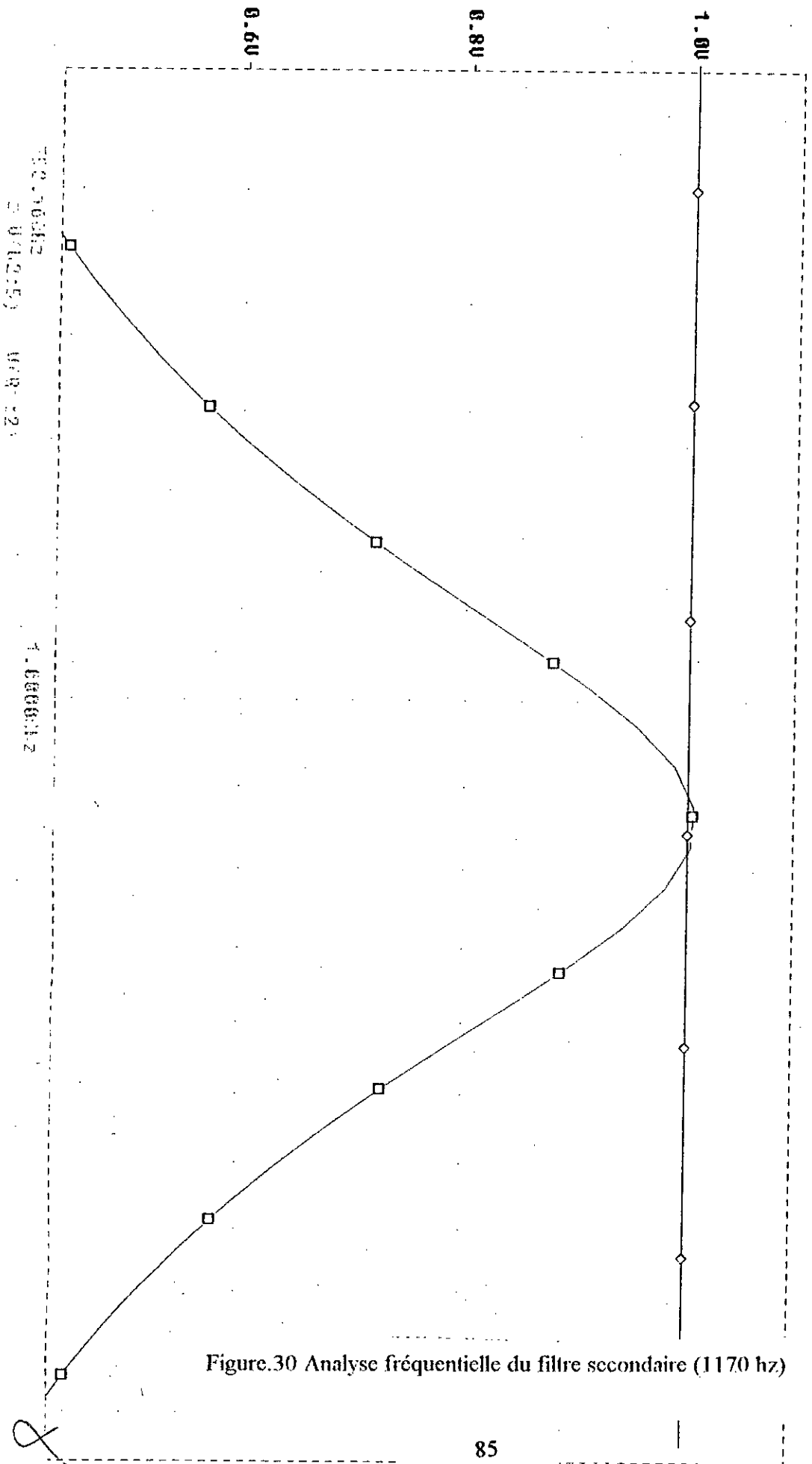


Figure.30 Analyse fréquentielle du filtre secondaire (1170 hz)

7 - LOGICIEL D'EXPLOITATION DU RESEAU

7.1 - Introduction

Soit une liaison qui relie deux micro-ordinateurs par l'intermédiaire de deux modems EF7910, l'un programmé en mode Bell 103 originate (appelant) et l'autre programmé en mode Bell 103 answer (appelé). Le dialogue entre les deux modems est donné selon le raisonnement suivant.

Lorsque les deux entrées DTR (Terminal de données prêt) des deux modems sont activées par les micro-ordinateurs, pour commencer la séquence d'émission, le micro-ordinateur appelant active sa sortie RTS (demande à émettre) vers le modem. Ce dernier envoie un ton spécifique au modem appelé. En réponse à ce ton, ce dernier active sa sortie CD (détection de porteuse) vers le micro-ordinateur appelé. Le modem appelant répond au micro-ordinateur par sa sortie CTS (prêt à émettre) pour l'informer qu'il est prêt à émettre.

Le micro-ordinateur appelant envoie les données vers l'entrée TD (Transmission de données) du modem qui les transmet à son tour vers le modem appelé. Ces données seront transmises au micro-ordinateur à travers son entrée RD (réception de données).

Lorsque le micro-ordinateur (appelant) désire cesser l'émission, il désactive sa sortie RTS (demande à émettre) vers le modem, ce qui désactive CTS (prêt à émettre) et enlève le signal CD (détection de la porteuse) de l'autre côté de la liaison.

Parallèlement à la réception de données par le micro-ordinateur appelé, il peut en transmettre. Pour commencer, il active l'entrée RTS (demande à émettre) du modem. Ce dernier envoie un ton spécifique au modem appelant. En réponse à ce ton, le modem appelant active sa sortie CD (détection de porteuse) vers le micro-ordinateur. Le modem appelé répond au micro-ordinateur par sa sortie CTS (prêt à émettre) pour l'informer qu'il est prêt à émettre. Le micro-ordinateur appelé envoie les données vers l'entrée TD du modem qui les transmet à son tour au modem appelant. Ces données seront transmises au micro-ordinateur par son entrée RD.

La figure.31 montre les chronogrammes des signaux de dialogue entre les deux modems EF7910 programmés en Bell 103.

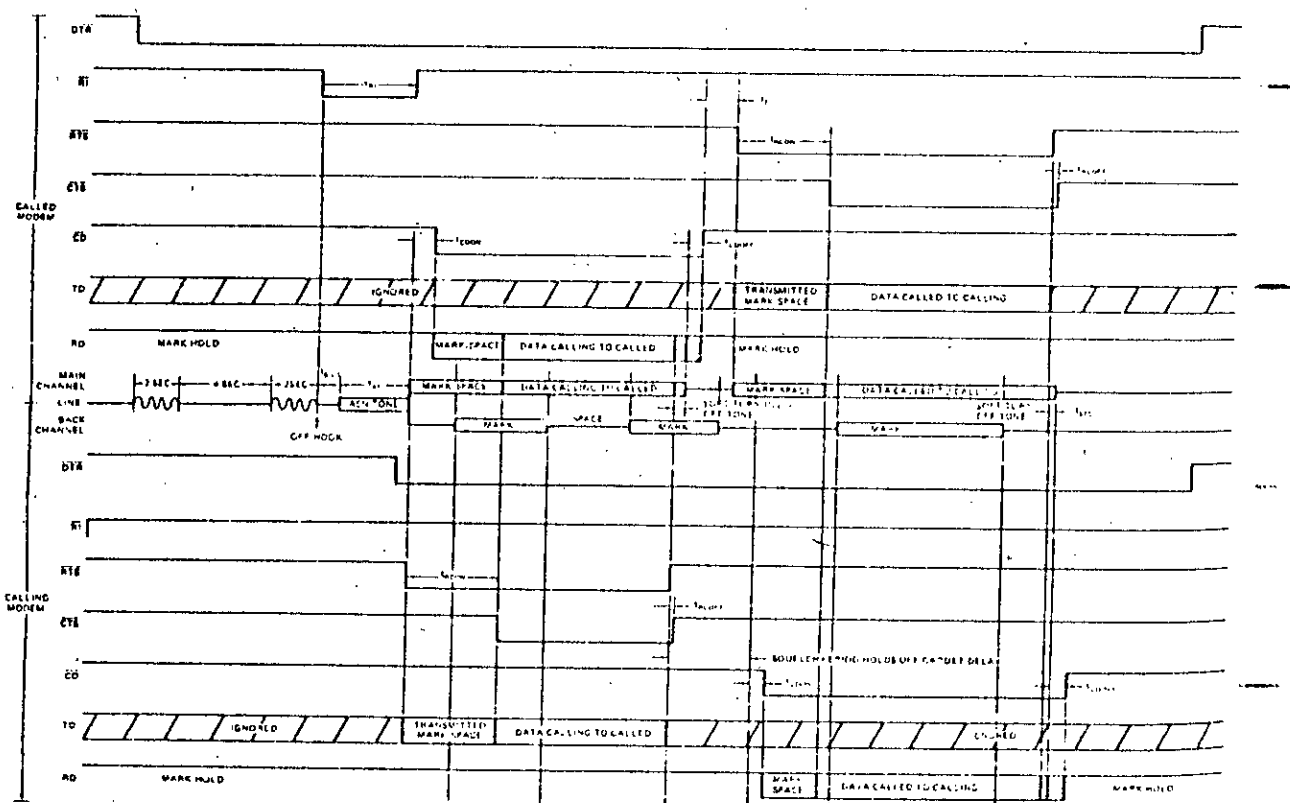


Figure. 31

7.2 - Gestion du réseau

La gestion du réseau d'ordinateur conçu est assurée en attribuant à chaque micro-ordinateur station un numéro d'adresse. Le micro-ordinateur serveur sélectionne le micro-ordinateur station, par l'envoi au début de la communication du caractère contenant le numéro d'adresse. Tous les micro-ordinateurs stations reçoivent ce caractère mais seul le micro-ordinateur possédant le même numéro répond au serveur.

7.3 - L'échange de données

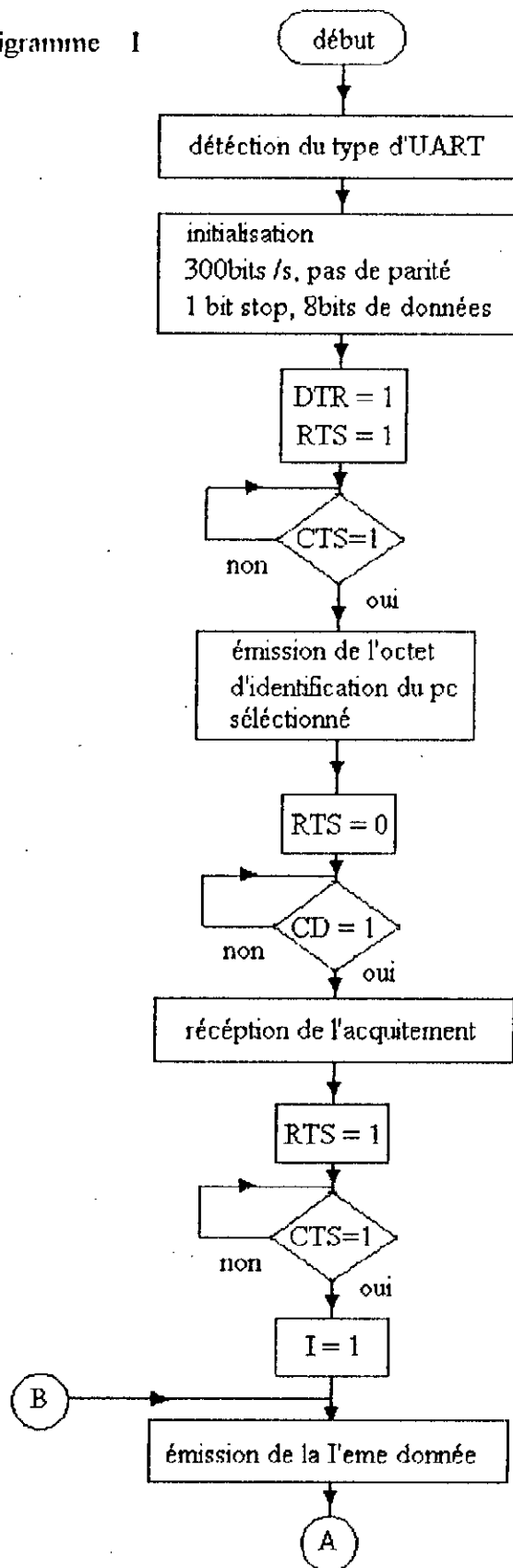
Après une initialisation des paramètres de communication des micro-ordinateurs du réseau (à savoir le débit 300 bauds, le pas de parité, 8 bits de données et un bit STOP), les sorties DTR reliées aux modems sont activées pour chaque micro-ordinateur. Si le micro-ordinateur serveur désire établir la communication avec l'un des micro-ordinateurs stations, il active sa sortie RTS vers le modem et attend que sa sortie CTS soit activée par le modem. Ceci fait, il commence la séquence d'émission par l'envoi d'un octet représentant le numéro du micro-ordinateur station désiré. Tous les micro-ordinateurs stations reçoivent cet octet mais seul le micro-ordinateur sélectionné répond. Sa réponse est assurée en envoyant un octet d'acquiescement positif vers le micro-ordinateur serveur. Ce dernier, en recevant l'acquiescement positif, commence l'envoi de données octet par octet. Le micro-ordinateur station reçoit les données octet par octet, en cas de donnée erronée, il émet un acquiescement négatif au micro-ordinateur serveur.

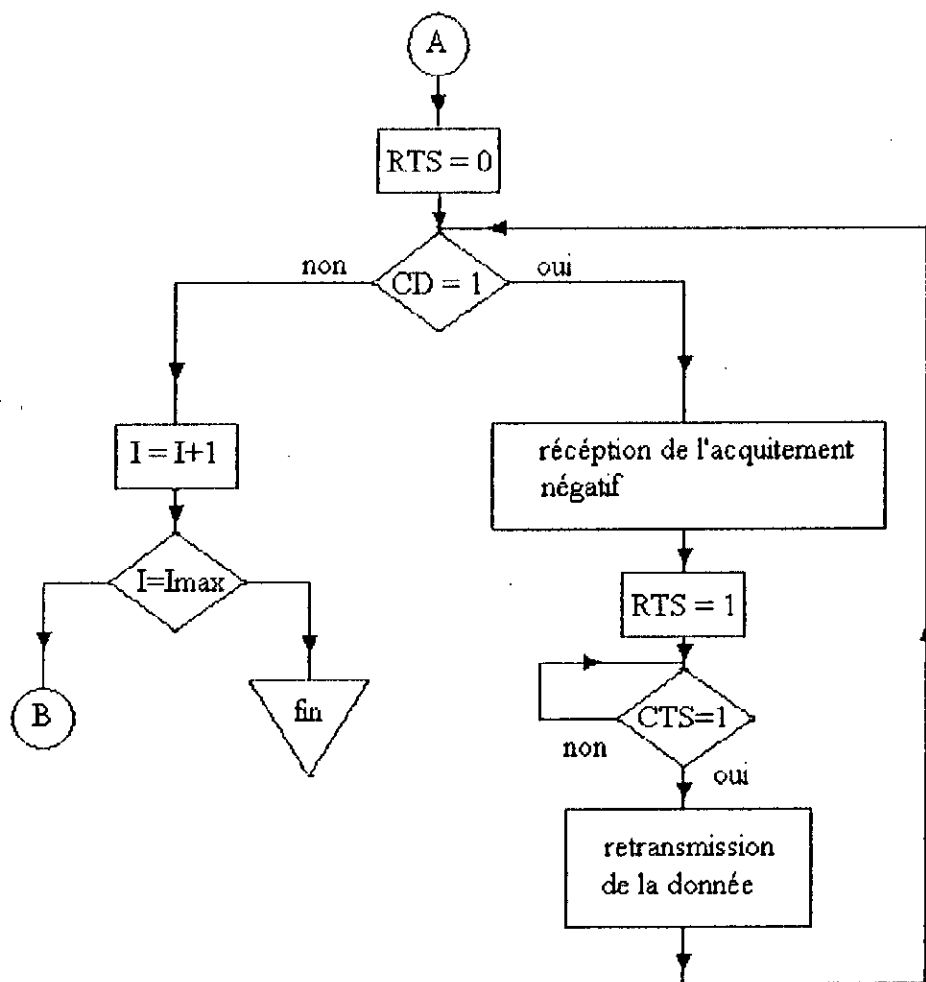
Le micro-ordinateur serveur teste, à chaque fois, si son entrée CD (détection de porteuse) est active. Si c'est le cas, il reçoit l'acquiescement négatif indiquant la réception d'une donnée erronée par le micro-ordinateur station et retransmet la donnée.

La séquence d'émission se poursuit jusqu'à l'envoi de toutes les données correctement. Pour cesser l'émission, le micro-ordinateur serveur désactive sa sortie RTS ce qui désactive CTS et enlève CD de l'autre côté de la ligne.

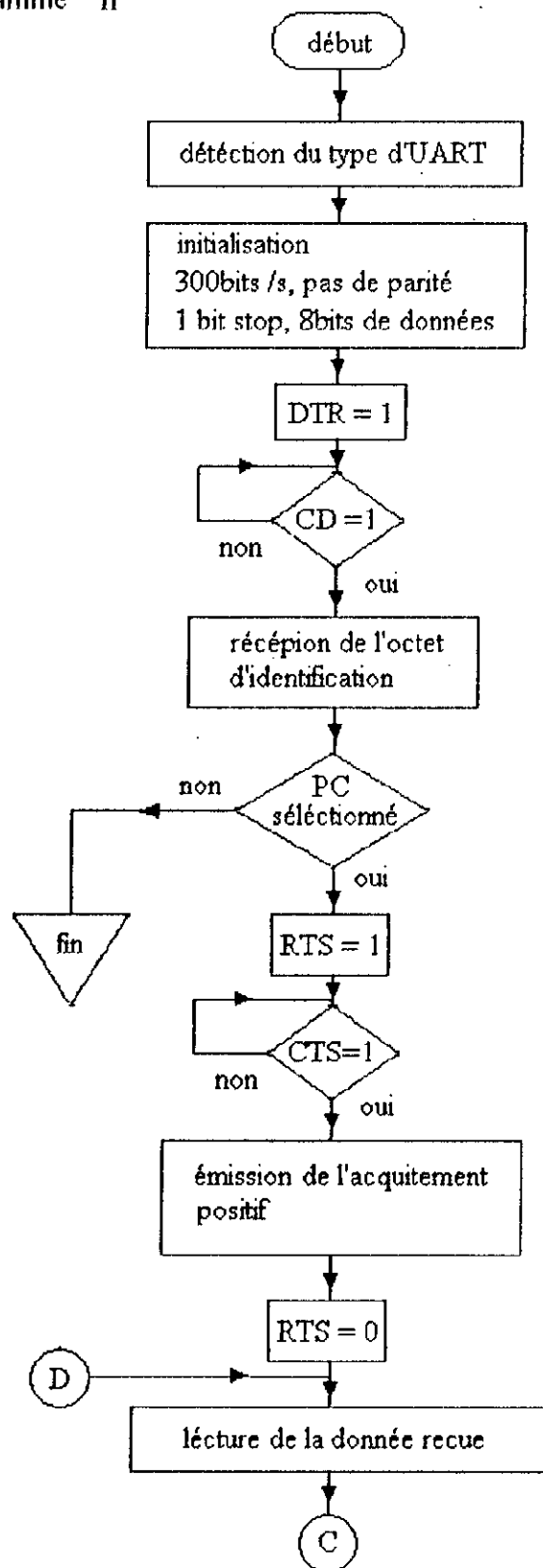
Les organigrammes I et II montrent la gestion de la communication. Les deux programmes de gestion de la communication sont donnés en annexe 4.

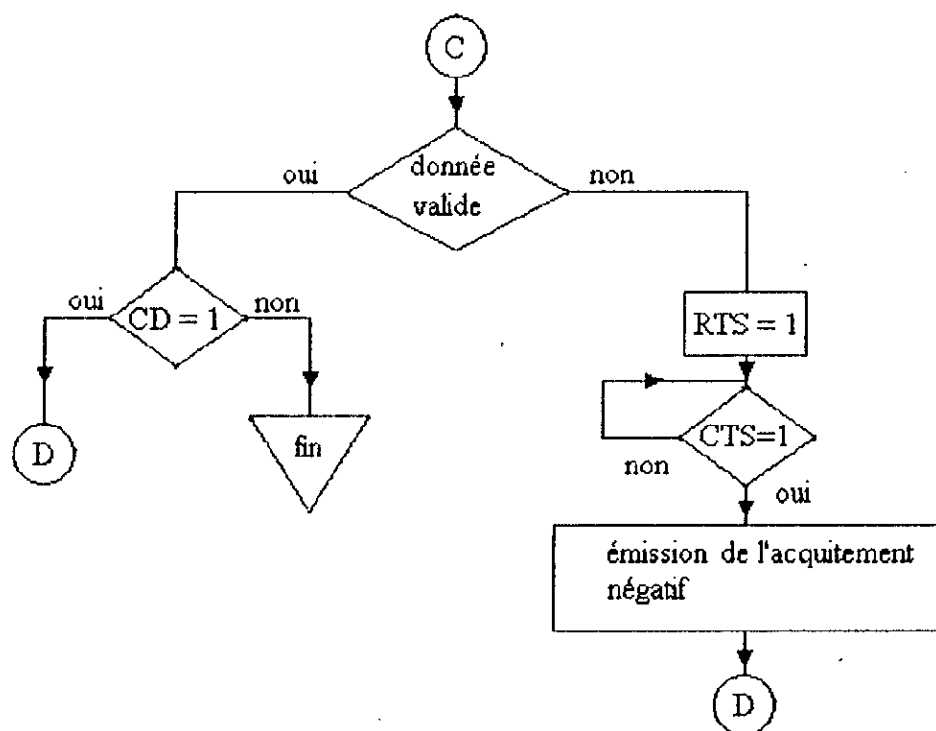
organigramme 1





organigramme II





9

CONCLUSION

La communication entre micro-ordinateurs nécessite une interface et un support de transmission flexible, ce fût le but de cette réalisation.

L'utilisation de la carte d'interface ainsi que le réseau électrique comme support de transmission a répondu à cette exigence tout en restant à la portée de tous.

La carte d'interface que nous avons réalisée, outre son utilisation en transmission par courant porteur peut être exploitée en transmission d'informations sur de grandes distances par câble ou par onde radio.

L'utilisation du modem EF7910 programmé en mode " BELL 103 " se voit très intéressante vu ses caractéristiques qui lui permettent d'interfacer les micro-ordinateurs aux supports analogiques tout en respectant les normes internationales.


Les fréquences issues du EF7910 demeurent trop faibles pour attaquer le réseau électrique, il a fallu utiliser un modulateur additionnel constitué du XR2206 qui présente l'avantage de faciliter la mise au point et le choix des fréquences.

De l'autre côté les fréquences injectées au réseau électrique sont trop grandes pour attaquer le EF7910, il a été nécessaire d'utiliser un démodulateur additionnel constitué de la PLL (boucle à verrouillage de phase), le HEF4046.

L'ensemble des circuits intégrés utilisés a permis d'exploiter au maximum les possibilités offertes par le EF7910 dans le mode " BELL 103 " à savoir une communication en fullduplex sur 300 Bauds.

Des tests finaux sur divers sites ainsi que diverses distances permettront de garantir la fonctionnalité et la fiabilité de la carte d'interface conçue ce qui n'était pas possible dans notre cas. Les tests que nous avons effectués (au laboratoire 11, département d'électronique sur courtes distances) ont été assez concluants.

Avec des systèmes microprogrammés remplaçant les micro ordinateurs stations, toutes les possibilités sont envisageables; entre autres : la commande à distance de machines outils à commande numérique dans la même usine, ou la surveillance permanente des matériels médicaux dans les chambres des malades dans un hôpital.



BIBLIOGRAPHIE

- [1] S.CIARCIA Circuits 2 Edit M.G.HILL 1986
- [2] C.TAC.TAVERNIER Modems Edit ETSF 1993
- [3] S.LEIPSON Manuel des interfaces Edit M.G.HILL 1984
- [4] M.TISCHER Bible du PC, micro-applications 1992
- [5] M.BETTAS La communication dans les réseaux d'ordinateurs 1993
- [6] Optoelectronics GE/RCA 1987
- [7] M.MARSTON Cent dix études pratiques de générateurs de signaux 1982
- [8] M.GIRARD Boucles à verrouillage de phase Edit science 1988.
- [9] C.MACCHI Téléinformatique Edit Dunod 1987
- [10] H.KHALDI "Transmissions par courant porteur en HT"
thèse pour obtention du diplôme d'ingénieur d'état ENP 1967.
- [11] BOUROUBI & CHARA "Etude et réalisation d'un système de liaison sans câble
entre un PC et un périphérique".
thèse pour obtention du diplôme d'ingénieur d'état ENP 1995.
- [12] J.CAMPBELL La liaison RS232 Edit Sybex
- [13] Guide PSI du matériels PC,PS et compatible

[14] Electronique Applications N°185 1987

[15] Guide INTEL des réseaux(logiciel INTEL) 1993

ANNEXES

ANNEXE 1

Sélection des composants du VCO

Valeurs recommandées pour R1 et R2 : de 10 k Ω à 1 M Ω ; pour C1 : 50 pF jusqu'à une valeur normalisée quelconque.

1. VCO sans déviation de la fréquence (R2 = ∞)
 - a. Connaissant f_0 , calculer la valeur de R1 et C1 à partir des courbes de la Fig. 7.
 - b. Connaissant f_{\max} , calculer la valeur de f_0 à partir de $f_0 = 1/2 f_{\max}$; puis déterminer R1 et C1 à partir de f_0 et des courbes de la Fig. 7.
2. VCO avec déviation de la fréquence
 - a. Connaissant f_0 et f_L , calculer f_{\min} à l'aide de l'équation $f_{\min} = f_0 - f_L$; déterminer R2 et C1 à partir de f_{\min} et des courbes de la Fig. 8 ; calculer

$$\frac{f_{\max}}{f_{\min}} \text{ d'après l'équation } \frac{f_{\max}}{f_{\min}} = \frac{f_0 + f_L}{f_0 - f_L} ;$$

puis à partir de l'équation $\frac{f_{\max}}{f_{\min}}$ et des courbes de la Fig. 9, calculer le rapport R2/R1

pour obtenir R1.

- b. Connaissant f_{\min} et f_{\max} , déterminer R2 et C1 à partir de f_{\min} et des courbes de la Fig. 8 ; calculer $\frac{f_{\max}}{f_{\min}}$; puis, à l'aide de $\frac{f_{\max}}{f_{\min}}$ et des courbes de la Fig. 9, déterminer R1/R2 pour obtenir R1.

9.4.

série HE 4000B, 1984, p. 297 et 298)

RTC HEF 4046B

RTC HEF 4046B

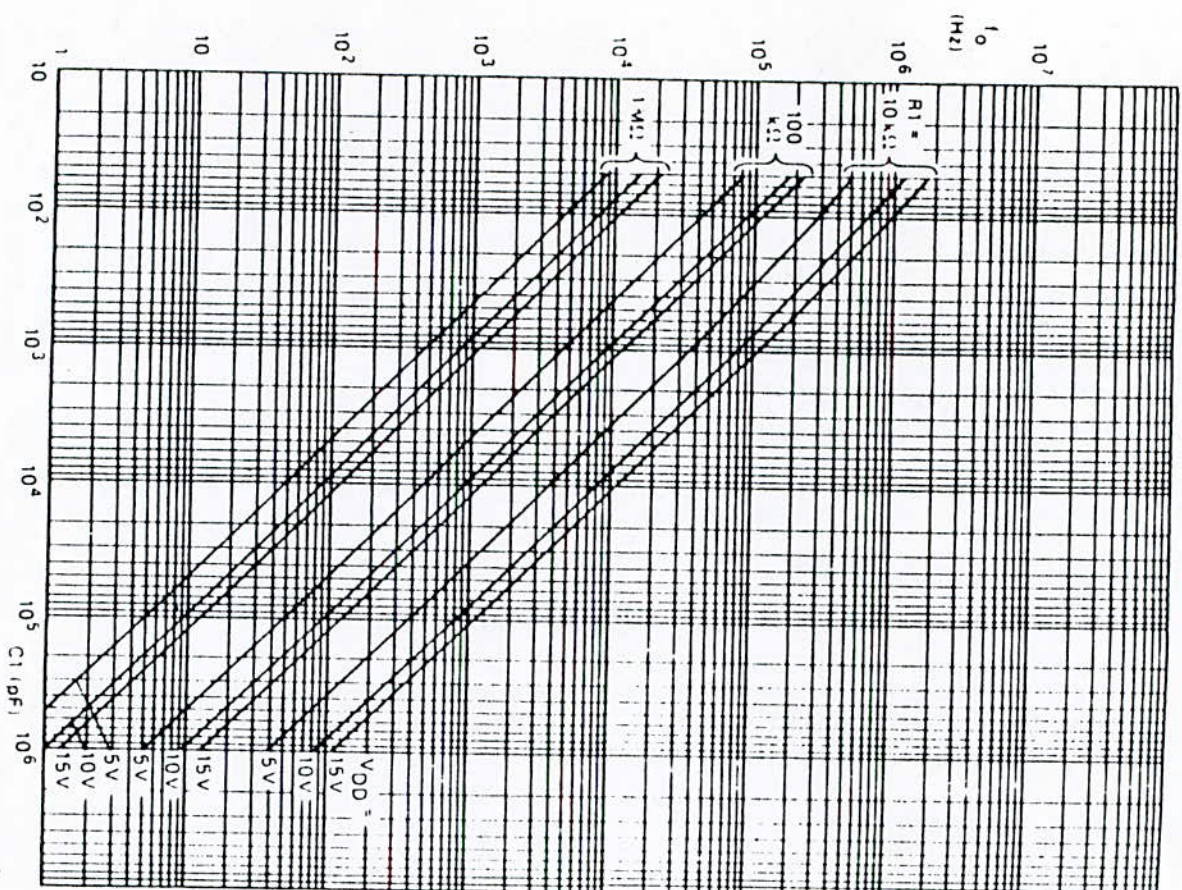


Fig. 7 Fréquence centrale typique en fonction de la capacité C_1 ; $T_{amb} = 25^\circ C$; V_{COIN} à $1/2$ V_{DD} ; I_{INH} à V_{SS} ; $R_2 = \infty$.

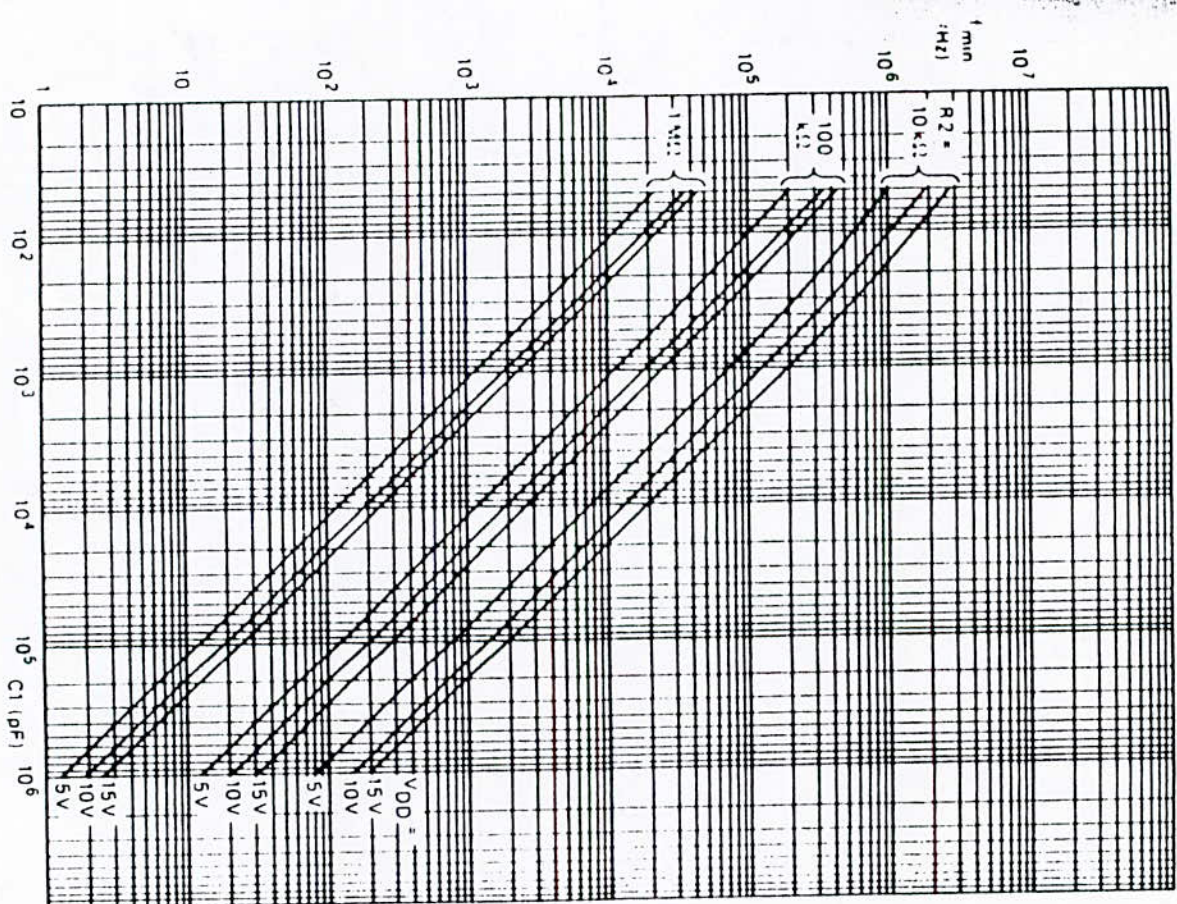


Fig. 8 Déviation de fréquence typique en fonction de la capacité C_1 ; $T_{amb} = 25^\circ C$; V_{COIN} à V_{SS} ; I_{INH} à V_{SS} ; $R_1 = \infty$.

Figure

(Document RTC, Manuel technique

9.5.

série HE 4000B, 1984, p. 299 et 300)



HEF 40468

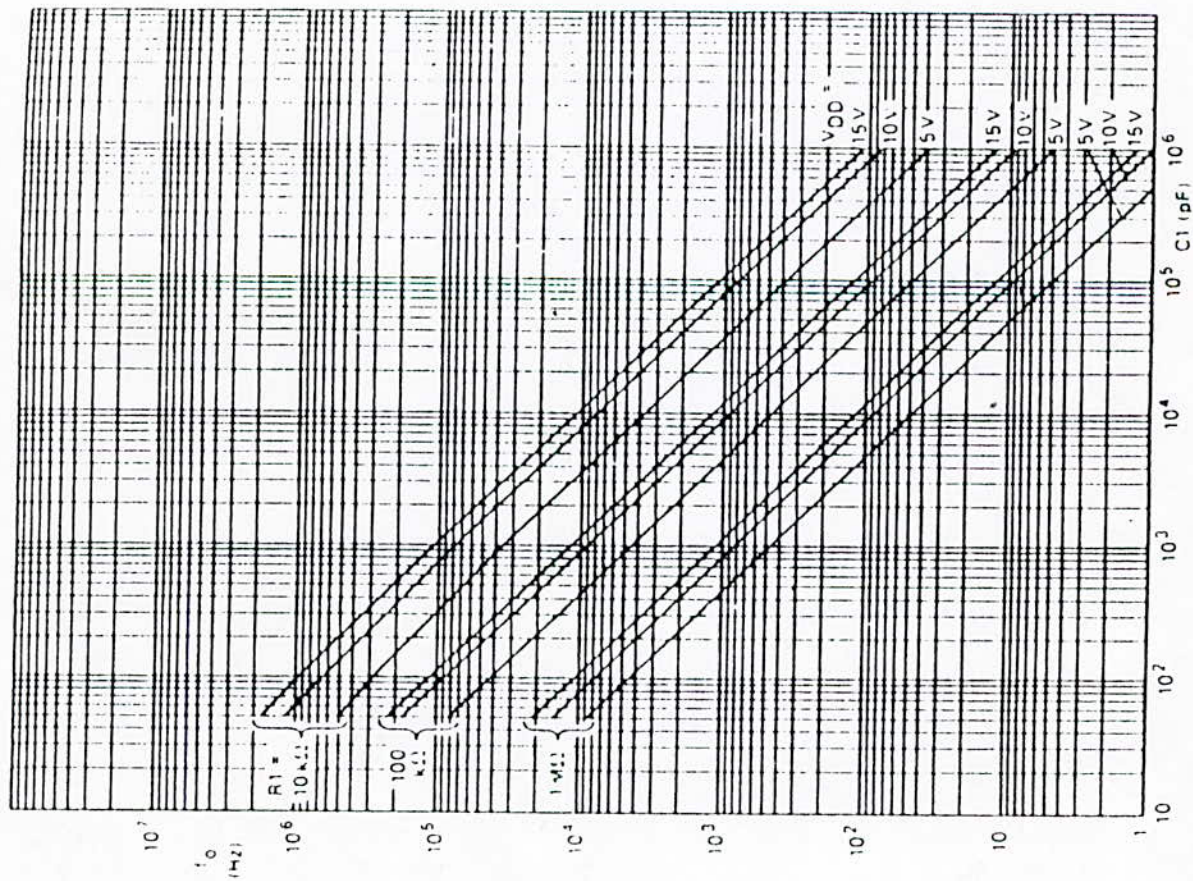


Fig. 7 Fréquence centrale typique en fonction de la capacité C_1 ; $T_{amb} = 25^\circ\text{C}$; $V_{COIN} = 1/2$ V_{DD} ; INH à V_{SS} ; $R_2 = \infty$.

Figure
(Document RTC, Manuel technique



HEF 40468

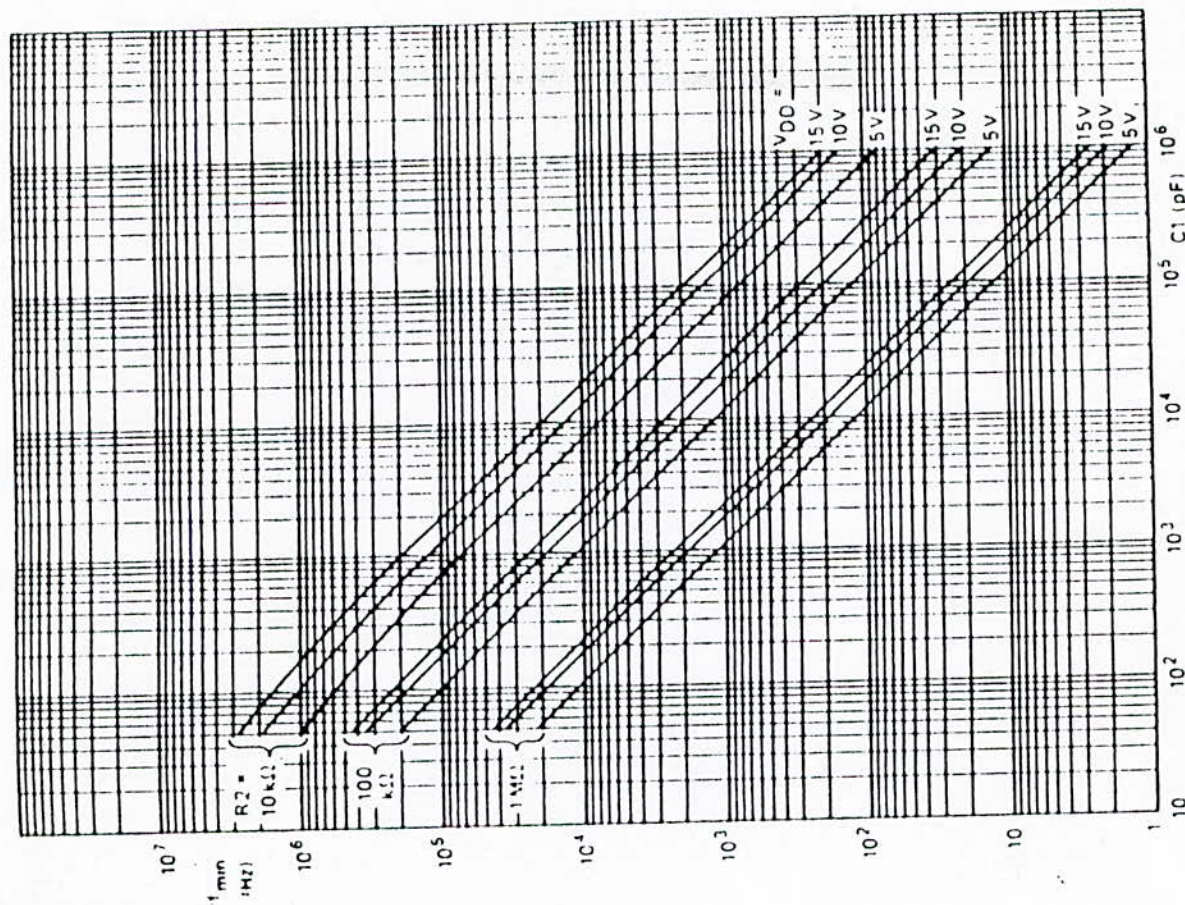


Fig. 8 Déviation de fréquence typique en fonction de la capacité C_1 ; $T_{amb} = 25^\circ\text{C}$; V_{COIN} à V_{SS} ; INH à V_{SS} ; $R_1 = \infty$.

9.5.
série HE 4000B, 1984, p. 299 et 300)

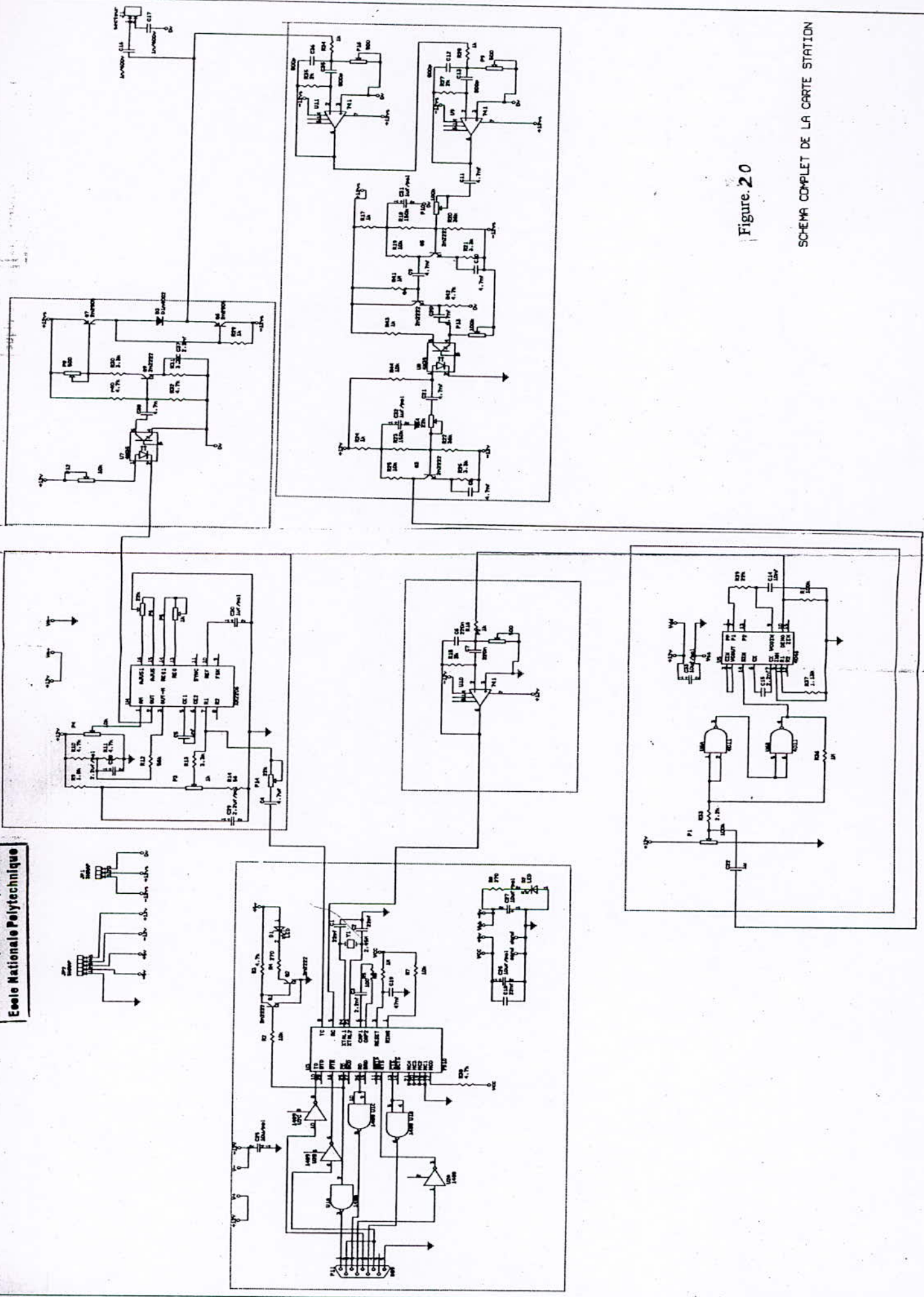


Figure.20

SCHEMA COMPLET DE LA CARTE STATION

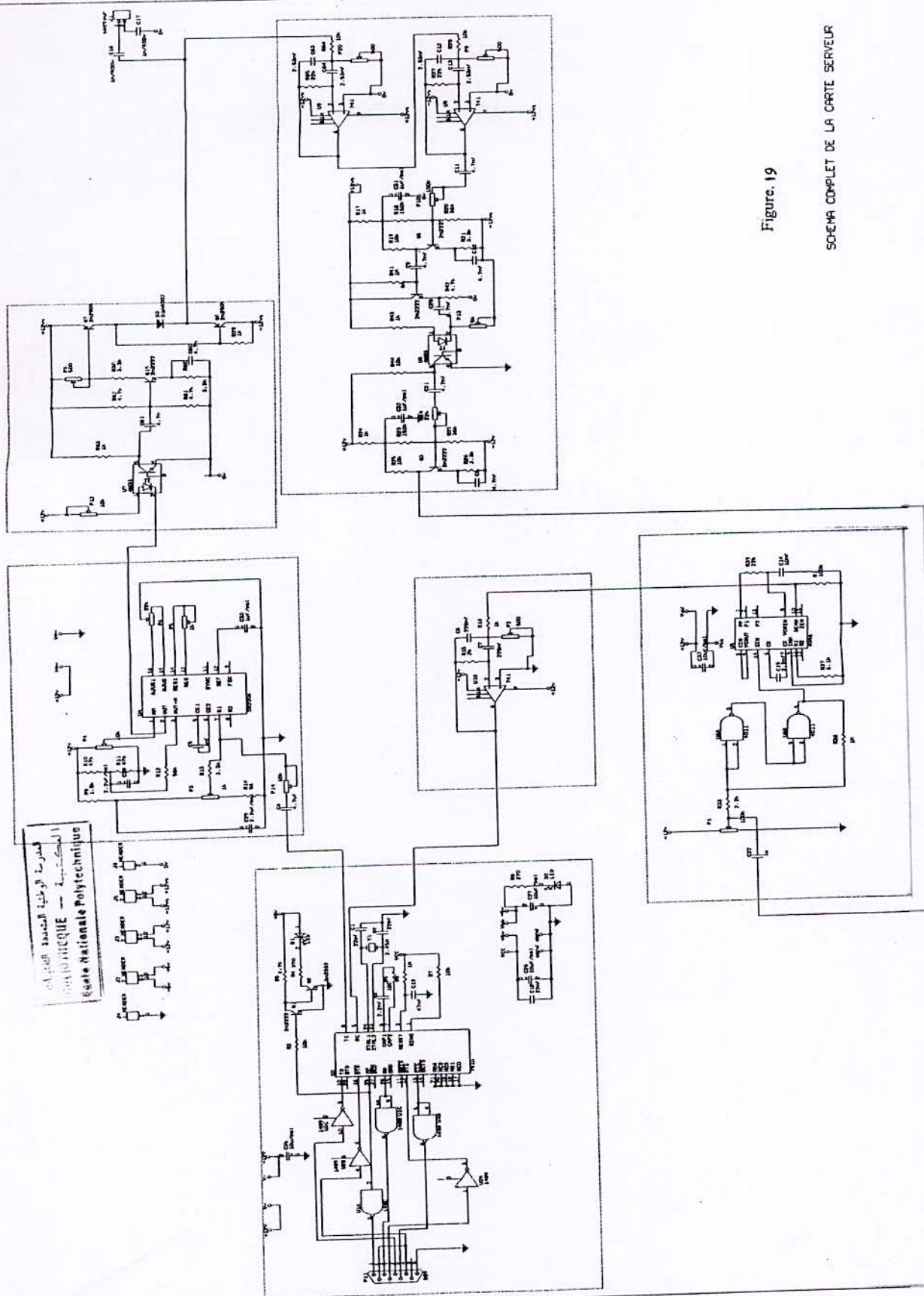


Figure. 19

SCHEMA COMPLET DE LA CARTE SERVEUR


```

[*****
***** Programme principal *****
*****]

```

BEGIN

```

  if ser_uarttype($2f8) = noser then
    begin
      writeln ('Il n''ya pas d''interface!');
      halt;
    end;
  ia:= ser_init($2f8,300, SER_LCR_8bits or ser_lcr_1stopbit or ser_lcr_noparity);
  if ia= noser then
    begin
      writeln('Il n''ya pas d''interface!');
      halt ;
    end;

```

```

  ser_setmodemcontrol($2f8, ser_MCR_DTR );
  repeat
  until ser_ismodemstatusset($2f8,ser_msr_cd);
  clrscr;
  yt := '';
  ib:= ser_readbyte($2f8,ad,$182,0,0);
  if((ib<> ser_success) and (ad<>code )) then
    begin
      writeln ('pc non selectionné ');
      halt;
    end;

```

```

  ser_setmodemcontrol($2f8,ser_mcr_dtr or ser_mcr_rts);
  repeat
  until ser_ismodemstatusset($2f8,ser_msr_cts);

  ic := ser_writebyte($2f8, ascii_ack,$182,0,0);
  if ic<> ser_success then
    begin
      writeln('erreur1');
      halt;
    end;

```

```

  ser_setmodemcontrol($2f8,ser_mcr_dtr);

```

```

  while ser_ismodemstatusset($2f8, ser_msr_cd) do
    begin
      id:=ser_readByte($2f8,ahmed, $182,0,0);
      while id <> ser_success do
        begin
          ser_setmodemcontrol($2f8,ser_mcr_dtr or ser_mcr_rts);
          repeat
          until ser_ismodemstatusset($2f8,ser_msr_cts);
          ie:=ser_writeByte($2f8,ascii_nak,$182,0,0);
          if (ie<>ser_success) then
            begin
              writeln('erreur 2');
              halt;
            end;
          id:=ser_readbyte($2f8,ahmed, $182,0,0);
        end;

```

```

      yt:= yt+chr(ahmed);
    end;

```

```

  writeln ('La chaine reçue est : ',yt);
  End.

```



```

[      bSigMask - Masque de bits des lignes de signal à tester ]
[      (RTS, CTS, CD, RI) ]
[      bSigVals - Etat des lignes de signal après ]
[      avoir démasqué avec le masque ci-dessus. ]
[ Sortie : = 0 - octets ont été envoyés ]
[      <> 0 - Erreurs ]
[*****]
Function ser_ReadByte(   iSerPort      : Integer;
                        var Data       : Byte;
                        uTimeout       : Word;
                        bSigMask, bSigVals : Byte ) : Integer;

Begin
  if uTimeout <> 0 then                                [ Boucle Timeout ]
  Begin
    while( not ser_IsDataAvailable( iSerPort )
           and ( uTimeout <> 0 ) )
    do Dec( uTimeout );
    if uTimeout = 0 then
    Begin
      ser_ReadByte := SER_ERRTIMEOUT;
      Exit;
    End;
  End
  else
    [ Attente ! ]
    Repeat
      Until ser_IsDataAvailable( iSerPort );

    {-- Tester les lignes de signal -----}
    if ( port[iSerPort+SER_MODEM_STATUS] and bSigMask ) = bSigVals then
    Begin
      [ lit octet reçu par l'interface ]
      Data := port[iSerPort + SER_RXBUFFER];
      ser_ReadByte := port[iSerPort + SER_LINE_STATUS] and
        SER_LSR_ERRORMSK;
    End
    else
      ser_ReadByte := SER_ERRSIGNALS;
    End;

    uTimeout : Word;
    bSigVals , bSigMask : Byte ) : Integer;

var i, e : Integer;
    BufPtr : SerBufPtr;

Begin
  BufPtr := pData;
  if ilen < 0 then { si la longueur n'est pas indiquée }
  Begin
    [ il faut d'abord chercher le premier ]
    [ byte zéro du bloc ]
    ilen := 0;
    While BufPtr[ilen] <> 0 do
      Inc(ilen);
    End;

    {-- parcourir le bloc et envoyer tous les octets un à un -}
    for i := 0 to ilen - 1 do
    Begin
      e := ser_WriteByte( iSerPort, BufPtr[i], uTimeout,
                          bSigMask, bSigVals );
      if e <> 0 then
      Begin
        ser_WritePaket := e;
        Exit;
      End;
    End;
    ser_WritePaket := SER_SUCCESS;
  End;

```



```

Begin
  port[iSexPort + SER_MODEM_CONTROL] := bNewControl;
End;

{*****}
{ ser_WriteByte : Envoi d'un octet }
{*****}
{
  Entrée : iSexPort - Port de base de l'interface pour
  envoyer un octet.
  {
    bData - octet à envoyer
    {
      uTimeOut - Nombre de passages effectués dans la boucle
      avant que l'échec d'une émission n'est signalé
      par une erreur TimeOut.
      {
        (Si iTimeOut = 0 l'attente continue
        {
          bSigMask - Masque de bits des lignes de signal à tester
          {
            (RTS, CTS, CD, RI)
            {
              bSigVals - Etat des lignes de signal après avoir
              démasqué le masque ci-dessus.
              {
                Sortie : = 0 - octets ont été envoyés
                {
                  <> 0 - Erreurs
                  {*****}
}
Function ser_WriteByte( iSexPort      : Integer;
                        bData          : Byte;
                        uTimeOut       : Word;
                        bSigMask, bSigVals : Byte ) : Integer;

Begin
  if uTimeOut <> 0 then
    [ Boucle TimeOut ]
    Begin
      While( not ser_IsWritingPossible( iSexPort )
        and ( uTimeOut <> 0 ) )
        do Dec( uTimeOut );
      if uTimeOut = 0 then
        Begin
          ser_WriteByte := SER_ERRTIMEOUT;
          Exit;
        End;
      End
    else [ Attente ! ]
      Repeat
        Until ser_IsWritingPossible( iSexPort );

      [-- test de lignes de signal -----]
      if ( port[iSexPort + SER_MODEM_STATUS] and bSigMask ) = bSigVals
      then
        Begin [ transmission des octets à émettre vers l'interface ]
          port[iSexPort + SER_TXBUFFER] := bData;
          [ Retourne erreur interface ]
          ser_WriteByte := port[iSexPort + SER_LINE_STATUS] and
            SER_LSR_ERRORMSK;
        End
      else
        ser_WriteByte := SER_ERRSIGNALS;
      End;
End;

{*****}
{ ser_ReadByte : Réception d'un octet }
{*****}
{
  Entrée : iSexPort - Port de base de l'interface utilisée
  pour recevoir un octet
  {
    Data - la variable byte accueille l'octet
    reçu.
    {
      uTimeOut - Nombre de passages dans la boucle
      avant qu'une erreur TimeOut ne signale
      l'échec de la réception.
      {
        (Si iTimeOut = 0 l'attente continue )
        {

```



```

port[iSerPort + SER_DIVISOR_LSB] := LO( uDivisor );
port[iSerPort + SER_DIVISOR_MSB] := HI( uDivisor );

port[iSerPort + SER_LINE_CONTROL] := {interdit l'accès au diviseur }
port[SER_LINE_CONTROL] and not SER_LCR_SETDIVISOR;

{-- Pour fixer les autres paramètres attendre le positionnement en }
{ arrière des taux de baud Latch parce que l'opération efface tous }
{ les paramètres de l'interface }

port[iSerPort + SER_LINE_CONTROL] := bParams;
    { Ligne un byte pour placer les erreurs éventuellement }
    { présentes en arrière. }

b := port[iSerPort + SER_TXBUFFER];
ser_Init := uART;
End;

```

```

Function ser_IsDataAvaliable( iSerPort : Integer ) : Boolean;

```

```

Begin
    ser_IsDataAvaliable := ( port[iSerPort + SER_LINE_STATUS]
and SER_LSR_DATA RECEIVED ) <> 0;
End;

```

```

{*****}
{ ser_IsWritingPossible: L'interface, peut-elle envoyer encore un bit? }
{*****}
{ Entrée : iSerPort - Port de base de l'interface à tester }
{ Sortie : = 0 : le byte ne peut pas être envoyé. }
{ <>0 : l'interface est prête à émettre. }
{*****}
{ Info : Il ne faut pas utiliser une interface série pour envoyer }
{ des octets dans les cas suivants : }
{ 1. Un octet reçu n'a pas encore été appelé par l'interface }
{ 2. Une demande d'émission n'a pas encore été exécutée }
{*****}
Function ser_IsWritingPossible( iSerPort : Integer ) : Boolean;

```

```

Begin
    ser_IsWritingPossible := ( port[iSerPort + SER_LINE_STATUS]
and SER_LSR_TSREMPY ) <> 0;
End;

```

```

{*****}
{ ser_IsModemStatusSet : Vérifier l'état d'entrée des lignes }
{*****}
{ Entrée : iSerPort - Port de base de l'interface. }
{ bTestStatus - Bit échantillon des lignes à tester }
{ (CTS, DSR, RI, CD) }
{*****}
Function ser_IsModemStatusSet( iSerPort : Integer;
bTestStatus : Byte ) : Boolean;

```

```

Begin
    ser_IsModemStatusSet := ( port[iSerPort + SER_MODEM_STATUS]
and bTestStatus ) = bTestStatus;
End;

```

```

{*****}
{ ser_SetModemControl : Etablir connexions de signal pour une communi- }
{ cation avec le modem. }
{*****}
{ Entrée : iSerPort - Port de base de l'interface. }
{ bNewControl - Etat nouveau des lignes DTR, RTS }
{*****}
Procédure ser_SetModemControl( iSerPort : Integer; bNewControl : Byte);

```



```

port[iSerPort + SER_LINE_CONTROL] := (interdit l'accès au diviseur )
port[SER_LINE_CONTROL] and not SER_LCR_SETDIVISOR;

[ -- Pour fixer les autres paramètres attendre le positionnement en ]
[ arrière des taux de baud latch parce que l'opération efface tous ]
[ les paramètres de l'interface ]

port[iSerPort + SER_LINE_CONTROL] := bParams;
[ Lixe un byte pour placer les erreurs éventuellement ]
[ présentes en arrière. ]

b := port[iSerPort + SER_TXBUFFER];
ser_Init := uART;
End;

```

```

Function ser_IsDataAvaible( iSerPort : Integer ) : Boolean;

```

```

Begin

```

```

    ser_IsDataAvaible := ( port[iSerPort + SER_LINE_STATUS]
                          and SER_LSR_DATA RECEIVED ) <> 0;

```

```

End;

```

```

[*****]
[ ser_IsWritingPossible: L'interface, peut-elle envoyer encore un bit? ]
[*****]
[ Entrée : iSerPort - Port de base de l'interface à tester ]
[ Sortie : = 0 : le byte ne peut pas être envoyé. ]
[ <> 0 : l'interface est prête à émettre. ]
[*****]
[ Info : Il ne faut pas utiliser une interface série pour envoyer ]
[ des octets dans les cas suivants : ]
[ 1. Un octet reçu n'a pas encore été appelé par l'interface ]
[ 2. Une demande d'émission n'a pas encore été exécutée ]
[*****]
Function ser_IsWritingPossible( iSerPort : Integer ) : Boolean;

```

```

Begin

```

```

    ser_IsWritingPossible := ( port[iSerPort + SER_LINE_STATUS]
                              and SER_LSR_TSREMPY ) <> 0;

```

```

End;

```

```

[*****]
[ ser_IsModemStatusSet : Vérifier l'état d'entrée des lignes ]
[*****]
[ Entrée : iSerPort - Port de base de l'interface. ]
[ bTestStatus - Bit échantillon des lignes à tester ]
[ (CTS, DSR, RI, CD) ]
[*****]

```

```

Function ser_IsModemStatusSet( iSerPort : Integer;
                               bTestStatus : Byte ) : Boolean;

```

```

Begin

```

```

    ser_IsModemStatusSet := ( port[iSerPort + SER_MODEM_STATUS]
                              and bTestStatus ) = bTestStatus;

```

```

End;

```

```

[*****]
[ ser_SetModemControl : Etablir connexions de signal pour une communi- ]
[ cation avec le modem. ]
[*****]
[ Entrée : iSerPort - Port de base de l'interface. ]
[ bNewControl - Etat nouveau des lignes DTR, RTS ]
[*****]

```



```

    port[iSerPort + SER_FIFO] := $CF;
    if ( port[iSerPort + SER_IRQ_ID] and $C0 ) <> $C0
then
    UartDetect := NS16450
else
    Begin
        port[iSerPort + SER_FIFO] := 0;
        [ le registre Alternate-Function est présent ? ]
        port[iSerPort + SER_LINE_CONTROL] :=
SER_LCR_SETDIVISOR;
        port[iSerPort + SER_2Function] := $07;
        if port[iSerPort + SER_2Function] <> $07 then
            Begin
                port[iSerPort + SER_LINE_CONTROL] := 0;
                UartDetect := NS16550A;
            End
        else
            Begin
                port[iSerPort + SER_LINE_CONTROL] := 0;
                [Repositionner en arrière ]
                port[iSerPort + SER_2Function] := 0;
                UartDetect := NS16C552;
            End;
        End;
    End;
End;
End;
End;

```

```

    ser_UARTType := UartDetect;
End;

```

```

{*****}
[ ser_Init : initialise l'interface série ]
{*****}
[ Entrée : iSerPort - Port de base pour l'interface ]
[ à initialiser ]
[ lBaud - taux de baud ( de 1 à 115200 ) ]
[ bParams - Masque de bit pour les autres paramètres ]
[ (à voir SER_LCR...-Bits) ]
[ Sortie : TRUE - l'interface a été initialisée ]
[ FALSE - interface introuvable ]
{*****}

```

```

Function ser_Init( iSerPort : Integer;
    lBaudRate : longint;
    bParams : Byte ) : Integer;

```

```

var uDivisor : Word;
    b : Byte;
    uART : Integer;

```

```

Begin
    uART := ser_UARTType( iSerPort );
    if uart = NOSER then
        Begin
            [ calcule le diviseur des taux de baud ]
            ser_Init := NOSER;
            exit;
        end;

    uDivisor := ( SER_MAXBAUD div lBaudRate );

    port[iSerPort + SER_LINE_CONTROL] := [permet l'accès au diviseur ]
    port[SER_LINE_CONTROL] or SER_LCR_SETDIVISOR;

    port[iSerPort + SER_DIVISOR_LSB] := LO( uDivisor );
    port[iSerPort + SER_DIVISOR_MSB] := HI( uDivisor );

```


SER_LCR_SENDBREAK = \$40; [place une fois que le bit de transmission est à 1]
SER_LCR_SETDIVISOR = \$80; [Pour accéder au diviseur des taux de baud]

[bits du registre de contrôle du modem (contrôle du signal)]
SER_MCR_DTR = \$01; [placer signal DTR]
SER_MCR_RTS = \$02; [placer signal RTS]
SER_MCR_UNUSED = \$04;
SER_MCR_IRQENABLED = \$08; [signale IRQ's au contrôleur de IRQ]
SER_MCR_LOOP = \$10; [autotest]

[Registre de l'état des lignes - Bits (Erreurs de transmission)]
SER_LSR_DATARECEIVED = \$01; [données ... (5 - 8 Bits) reçus]
SER_LSR_OVERRUNERROR = \$02; [données antérieures perdues]
SER_LSR_PARITYERROR = \$04; [Erreur de parité]
SER_LSR_FRAMINGERROR = \$08; [Erreur Bit de stop/de départ]
SER_LSR_BREAKDETECT = \$10; [Break détecté]
SER_LSR_ERRORMSK = [SER_LSR_OVERRUNERROR ou SER_LSR_PARITYERROR ou
SER_LSR_FRAMINGERROR ou SER_LSR_BREAKDETECT];
SER_LSR_THREMPY = \$20;
SER_LSR_TSREMPY = \$40;

[Registre d'état du modem - Bits]
[(Quels sont les signaux qui ont été positionnés ?) Les bits Delta]
[indiquent si l'état des signaux en question depuis la]
[dernière lecture du registre de l'état du modem s'est changé.]
SER_MSR_DCTS = \$01; [Delta CTS (Etat dans CTS)]
SER_MSR_DDSR = \$02; [Delta DSR (Etat dans DSR)]
SER_MSR_DRI = \$04; [Delta RI (Etat dans RI)]
SER_MSR_DCD = \$08; [Delta CD (Etat dans CD)]
SER_MSR_CTS = \$10; [Fixe Clear To Send]
SER_MSR_DSR = \$20; [Fixe Data Set Ready]
SER_MSR_RI = \$40; [Fixe indicateur Rin]
SER_MSR_CD = \$80; [Fixe Carrier Detect]

NOSER = 0;

NOSER = 0;
INS8250 = 1; [UART National Semiconductor]
NS16450 = 2;
NS16550A = 3;
NS16C552 = 4;

SER_MAXBAUD = 115200; [taux de baud max.]
SER_SUCCESS = 0;
SER_ERRSIGNALS = \$0300;
SER_ERRTIMEOUT = \$0400;

Type

SenBuf = array [0..65534] of byte;
SenBufPtr = ^SenBuf;
var
j,io,iw,is,ia,ib,ic,id,ie,il : Integer ;
ai,bi,na,ci,dat,adx,ahmed,i : byte ;
xl : string;

```
*****
[ ser_UARTType : Détecte type du chip UART ]
[-----*-----]
[ Entrée : iSerPort - Test du port de base de l'interface ]
[ Sortie : 0 (NOSER) - pas de chip UART ]
[ 1 (INS8250) - INS8250 ou INS8250-B Chip ]
[ 2 (NS16450) - INS8250A, INS82C50A, NS16450, NS16C450 ]
[ 3 (NS16550A) - NS16550A Chip ]
[ 4 (NS16C552) - NS16C552 Chip ]
*****
Function ser_UARTType( iSerPort : Integer ) : Integer;
```



```

Begin
    port[iSerPort + SER_SCRATCH] := 0;
    port[iSerPort + SER_FIFO] := $CF;
    if ( port[iSerPort + SER_IRQ_ID] and $C0 ) <> $C0
then
    UartDetect := NS16450
    else
    Begin
        port[iSerPort + SER_FIFO] := 0;
        [ le registre Alternate-Function est présent ? ]
        port[iSerPort + SER_LINE_CONTROL] :=
SER_LCR_SETDIVISOR;
        port[iSerPort + SER_2Function] := $07;
        if port[iSerPort + SER_2Function] <> $07 then
            Begin
                port[iSerPort + SER_LINE_CONTROL] := 0;
                UartDetect := NS16550A;
            End
        else
            Begin
                port[iSerPort + SER_LINE_CONTROL] := 0;
                [Repositionner en arrière ]
                port[iSerPort + SER_2Function] := 0;
                UartDetect := NS16C552;
            End;
        End;
    End;
End;
End;
End;
End;

sex_UARTType := UartDetect;
End;

{*****}
{ sex_Init : initialise l'interface série }
{*****}
{ Entrée : iSerPort - Port de base pour l'interface }
{ à initialiser }
{ lBaud - taux de baud ( de 1 à 115200 ) }
{ bParams - Masque de bit pour les autres paramètres }
{ (à voir SER_LCR...-Bits) }
{ Sortie : TRUE - l'interface a été initialisée }
{ FALSE - interface introuvable }
{*****}
Function sex_Init( iSerPort : Integer;
    lBaudRate : longint;
    bParams : Byte ) : Integer;

var uDivisor : Word;
    b : Byte;
    uART : Integer;

Begin
    uART := sex_UARTType( iSerPort );
    if uart = NOSER then
        Begin
            [ calcule le diviseur des taux de baud ]
            sex_Init := NOSER;
            exit;
        end;

    uDivisor := ( SER_MAXBAUD div lBaudRate );

    port[iSerPort + SER_LINE_CONTROL] := [permet l'accès au diviseur ]
    port[SER_LINE_CONTROL] or SER_LCR_SETDIVISOR;

```



```

{*****}
Function ser_UARTType( iSerPort : Integer ) : Integer;

var b          : Byte;
    UartDetect : integer;
Begin
    UartDetect := -1; { -1 indique : pas encore initialisé }

    { - Vérifie fonctionnalités de base ----- }
    port[iSerPort + SER_LINE_CONTROL] := $AA; {Diviseur Latch positionné}
    if port[iSerPort + SER_LINE_CONTROL] <> $AA then
        UartDetect := NOSER
    else
        Begin
            port[iSerPort + SER_DIVISOR_MSB] := $55; {Décrète diviseur}
            if port[iSerPort + SER_DIVISOR_MSB] <> $55
            then UartDetect := NOSER
            else
                {Efface diviseur latch }
                Begin
                    port[iSerPort + SER_LINE_CONTROL] := $55;
                    if port[iSerPort + SER_LINE_CONTROL] <> $55 then
                        UartDetect := NOSER
                    else
                        Begin
                            port[iSerPort + SER_IRQ_ENABLE] := $55;
                            if port[iSerPort + SER_IRQ_ENABLE] <> $05 then
                                UartDetect := NOSER
                            else
                                Begin
                                    port[iSerPort + SER_FIFO] := 0; {efface FIFO et IRQ}

                                    port[iSerPort + SER_IRQ_ENABLE] := 0;
                                    if port[iSerPort + SER_IRQ_ID] <> 1 then
                                        UartDetect := NOSER
                                    else
                                        Begin
                                            port[iSerPort + SER_MODEM_CONTROL] := $F5;
                                            if port[iSerPort + SER_MODEM_CONTROL] <> $15
then
                                                UartDetect := NOSER
                                            end;
                                        end;
                                    end;
                                end;
                            end;
                        end;
                    end;
                end;
            end;
        end;

        if UartDetect = -1 then { échappé au filtre ? }
        Begin { Boucle }
            port[iSerPort + SER_MODEM_CONTROL] := SER_MCR_LOOP;
            b := port[iSerPort + SER_MODEM_STATUS];
            if ( port[iSerPort + SER_MODEM_STATUS] and $F0 ) <> 0 then
                UartDetect := NOSER
            else
                Begin
                    port[iSerPort + SER_MODEM_CONTROL] := $1F;
                    if ( port[iSerPort + SER_MODEM_STATUS] and $F0 ) <> $F0 then
                        UartDetect := NOSER
                    else
                        Begin
                            port[iSerPort + SER_MODEM_CONTROL] := SER_MCR_DTR or
                                SER_MCR_RTS;
                                { le registre Scratch existe ? }
                            port[iSerPort + SER_SCRATCH] := $55;
                            if port[iSerPort + SER_SCRATCH] <> $55 then
                                UartDetect := INS8250
                            else
                                { FIFO existe ? }

```



```

SER_LCR_PARITYCLR = $38;      [ le bit de parité est tjs absent]
SER_LCR_PARITYMSK = $38;

SER_LCR_SENDBREAK = $40;     ['Break' tant que le bit est en position]
SER_LCR_SETDIVISOR = $80;    [Pour accéder au diviseur des taux de baud]

[ bits du registre de contrôle du modem (contrôle du signal) ]
SER_MCR_DTR = $01;           [ placer signal DTR ]
SER_MCR_RTS = $02;           [ placer signal RTS ]
SER_MCR_UNUSED = $04;
SER_MCR_IRQENABLED = $08;     [ signale IRQ's au contrôleur de IRQ]
SER_MCR_LOOP = $10;           [ autotest ]

[ Registre de l'état des lignes - Bits (Erreurs de transmission) ]
SER_LSR_DATARECEIVED = $01;   [ données ... (5 - 8 Bits) reçus ]
SER_LSR_OVERRUNERROR = $02;   [ données antérieures perdues ]
SER_LSR_PARITYERROR = $04;    [ Erreur de parité ]
SER_LSR_FRAMINGERROR = $08;   [ Erreur Bit de stop/de départ ]
SER_LSR_BREAKDETECT = $10;    [ Break détecté ]
SER_LSR_ERRORMSK = ( SER_LSR_OVERRUNERROR or SER_LSR_PARITYERROR or
SER_LSR_FRAMINGERROR or SER_LSR_BREAKDETECT );
SER_LSR_THREMPY = $20;
SER_LSR_TSREMPY = $40;

[ Registre d'état du modem - Bits ]
[ (Quels sont les signaux qui ont été positionnés ?) Les bits Delta ]
[ indiquent si l'état des signaux en question depuis la ]
[ dernière lecture du registre de l'état du modem s'est changé. ]
SER_MSR_DCTS = $01;           [ Delta CTS (Etat dans CTS) ]
SER_MSR_DDSR = $02;           [ Delta DSR (Etat dans DSR) ]
SER_MSR_DRI = $04;            [ Delta RI (Etat dans RI) ]
SER_MSR_DCD = $08;            [ Delta CD (Etat dans CD) ]
SER_MSR_CTS = $10;            [ Fixe Clear To Send ]
SER_MSR_DSR = $20;            [ Fixe Data Set Ready ]
SER_MSR_RI = $40;             [ Fixe indicateur Rin ]
SER_MSR_CD = $80;             [ Fixe Carrier Detect ]

NOSER = 0;
INS8250 = 1;                  [ UART National Semiconductor ]
NS16450 = 2;
NS16550A = 3;
NS16C552 = 4;

SER_MAXBAUD = 115200;         [ taux de baud max. ]

SER_SUCCESS = 0;
SER_ERRSIGNALS = $0300;
SER_ERRTIMEOUT = $0400;
code = $01                    [ adresse du microordinateur station ]
type
SexBuf = array[0..65534] of byte;
SexBufPtr = ^SexBuf;
var
i, ia, ib, ic, id, ie : Integer ;
ai, bi, ci, ahmed : byte ;
ad : byte ;
yt : string;
{*****}
[ sex_UARTType : Detecte type du chip UART ]
{*****}
[ Entrée : iSexPort - Test du port de base de l'interface ]
[ Sortie : 0 (NOSER) - pas de chip UART ]
[ 1 (INS8250) - INS8250 ou INS8250-B Chip ]
[ 2 (NS16450) - INS8250A, INS82C50A, NS16450, NS16C450 ]
[ 3 (NS16550A) - NS16550A Chip ]
[ 4 (NS16C552) - NS16C552 Chip ]

```



```

program serie2;
uses crt;
Const
  ASCII_ACK = $06;
  ASCII_NAK = $15;
  SER_COM1 = $3F8;           { Adresse de base COM1 }
  SER_COM2 = $2F8;           { Adresse de base COM2 }

  SER_IRQ_COM1 = 4;           { IRQ 4 = Vecteur $0C }
  SER_IRQ_COM2 = 3;           { IRQ 3 = Vecteur $0B }

  SER_TXBUFFER = $00;         { Registre Transmit }
  SER_RXBUFFER = $00;         { Registre Receive }
  SER_DIVISOR_LSB = $00;       { Diviseur des taux de baud LSB }
  SER_DIVISOR_MSB = $01;       { Diviseur des taux de baud MSB }
  SER_IRQ_ENABLE = $01;        { Registre d'interruption Enable }
  SER_IRQ_ID = $02;            { Registre d'interruption ID }
  SER_FIFO = $02;              { Registre FIFO }
  SER_2Function = $02;         { Registre Alternate Function }
  SER_LINE_CONTROL = $03;      { Contrôle des lignes }
  SER_MODEM_CONTROL = $04;      { Contrôle du modem }
  SER_LINE_STATUS = $05;        { Etat des lignes }
  SER_MODEM_STATUS = $06;       { Modemstatus }
  SER_SCRATCH = $07;           { Scratch-Registre }

  { (IRQ-Enable-Registre-Bits (Interruptions en/hors service) }
  SER_IER_RECEIVED = $01;       { IRQ après réception des données }
  SER_IER_SENT = $02;           { IRQ après l'envoi d'un bit }
  SER_IER_LINE = $04;           { IRQ après modification de l'état de ligne }
  SER_IER_MODEM = $08;          { IRQ après modification de l'état du modem }

  { IRQ-ID - Bits (Origine du IRQ ?) }
  SER_ID_PENDING = $01;         { le IRQ est il serial ? }
  SER_ID_MASK = $06;            { codage de l'ID en Bits 1 et 2 }
  SER_ID_LINESTATUS = $06;       { Etat de lignes (Error ou Break) }
  SER_ID_RECEIVED = $04;         { recevoir données }
  SER_ID_SENT = $02;            { Byte a été envoyé }
  SER_ID_MODEMSTATUS = $00;      { modification de CTS, DSR, RI ou RLSD }

  { Occupation de bits dans le registre FIFO }
  { (il faut au moins une UART 16550A) }
  SER_FIFO_ENABLE = $01;
  SER_FIFO_RESETRECEIVE = $02;
  SER_FIFO_RESETTRANSMIT = $04;

  { FIFO-Bits (L'IRQ se déclenche après ... bits dans le FIFO : ) }
  SER_FIFO_TRIGGER0 = $00;       { Normal }
  SER_FIFO_TRIGGER4 = $40;       { 4 Bytes }
  SER_FIFO_TRIGGER8 = $80;       { 8 Bytes }
  SER_FIFO_TRIGGER14 = $C0;      { 14 Bytes }


  { Registre du contrôle de ligne - Bits (paramètres de transmission) }
  SER_LCR_WordLEN = $03;         { nombre de bits à transmettre }
  SER_LCR_5BITS = $00;
  SER_LCR_6BITS = $01;
  SER_LCR_7BITS = $02;
  SER_LCR_8BITS = $03;
  SER_LCR_2STOPBITS = $04;       { 2 bzw. 1.5 bits de stop }
  SER_LCR_1STOPBIT = $00;        { 1 Bit de stop }

  SER_LCR_NOPARITY = $00;         { desactive le contrôle parité }
  SER_LCR_ODDPARITY = $08;        { parité impaire }
  SER_LCR_EVENPARITY = $18;       { parité paire }
  SER_LCR_PARITYSET = $28;        { le bit de parité est tjs présent }

```



```
if ((ie<>sex_success) and (na<>ASCII_NAK)) then
begin
writel('erreur 4');
halt;
end;
sex_setmodemcontrol($368,sex_mcr_dtr or sex_mcr_rts);
repeat
until sex_ismodemstatusset($368,sex_msr_cts);
id:=sex_Writebyte($368,adr, $182,0,0);
if id<>sex_success then
begin
writel('erreur 5');
halt;
end;
writel('erreur N° j ');
end;
end ;
End.
```




```

End;
ser_WritePaket := SER_SUCCESS;
End;

```

```

{*****
***** Programme principal *****

```

```

*****}
Begin
if ser_uarttype($368) = noser then
begin
writeln('Il n''ya pas d''interface!');
halt;
end
else
ia:= ser_init($368,300, SER_LCR_8bits or ser_lcr_1stopbit or ser_lcr_noparity);
if ia= noser then
begin
writeln('Il n''ya pas d''interface!');
halt;
end;

```

```

ser_setmodemcontrol($368, ser_MCR_DTR or SER_MCR_RTS);

```

```

repeat
until ser_ismodemstatusset($368, ser_MSR_CTS) ;
clrscr;
writeln('donner le N° de votre PC destination');
readln(i);
is:=ser_writebyte($368,i,$8000,0,0);
if is<> ser_success then
begin
writeln('erreur');
end;
ser_setmodemcontrol($368,ser_mcr_dtr);
repeat
until ser_ismodemstatusset($368,ser_msr_cd);
iw:=ser_readbyte($368,dat,$8000,0,0);
if(( iw<> ser_success)and (dat<>ascii_ack)) then
begin
writeln('erreur1');
halt;
end;
ser_setmodemcontrol($368,ser_mcr_dtr or ser_mcr_rts);
repeat
until ser_ismodemstatusset($368,ser_msr_cts);

writeln('donner le texte à transmettre');
readln (xt);
writeln('xt ',xt);
for j:= 1 to length(xt) do
begin
adr:=ord(xt[j]);
id:=ser_WriteByte($368, adr, $182,0,0);
if id <> 0 then
begin
writeln('erreur 2');
halt;
end;
ser_setmodemcontrol($368,ser_mcr_dtr);

while ser_ismodemstatusset($368, ser_MSR_CD) do
begin
ie:=ser_readByte($368,na,$182,0,0);

```



```

[
    (Si iTimeOut = 0 l'attente continue )
]
[
    bSigMask - Masque de bits des lignes de signal à tester
]
[
    (RTS, CTS, CD, RI)
]
[
    bSigVals - Etat des lignes de signal après
]
[
    avoir démasqué avec le masque ci-dessus.
]
[
    Sortie : = 0 - octets ont été envoyés
]
[
    <> 0 - Erreurs
]
[
    .....
]
Function ser_ReadByte( iSerPort      : Integer;
                      var Data       : Byte;
                      uTimeOut       : Word;
                      bSigMask, bSigVals : Byte ) : Integer;

Begin
    if uTimeOut <> 0 then
        [ Boucle TimeOut ]
        Begin
            while( not ser_IsDataAvaliable( iSerPort )
                and ( uTimeOut <> 0 ) )
            do Dec( uTimeOut );
            if uTimeOut = 0 then
                Begin
                    ser_ReadByte := SER_ERRTIMEOUT;
                    Exit;
                End;
            End
        else
            [ Attente ! ]
            Repeat
                Until ser_IsDataAvaliable( iSerPort );

            [ -- Tester les lignes de signal ----- ]
            if ( port[iSerPort+SER_MODEM_STATUS] and bSigMask ) = bSigVals then
                Begin
                    [ lit octet reçu par l'interface ]
                    Data := port[iSerPort + SER_RXBUFFER];
                    ser_ReadByte := port[iSerPort + SER_LINE_STATUS] and
                        SER_LSR_ERRORMSK;
                End
            else
                ser_ReadByte := SER_ERRSIGNALS;
            End;
        End;

nction ser_WritePaket( iSerPort : Integer;
                      pData      : pointer;
                      ilen       : Integer;
                      uTimeOut   : Word;
                      bSigVals , bsigmask: Byte ) : Integer;

var i, e : Integer;
    BufPtr : SerBufPtr;

Begin
    BufPtr := pData;
    if ilen < 0 then { si la longueur n'est pas indiquée }
        Begin
            [ il faut d'abord chercher le premier ]
            [ byte zéro du bloc ]
            ilen := 0;
            While BufPtr^[ilen] <> 0 do
                Inc(ilen);
            End;
        End;

    [ -- parcourir le bloc et envoyer tous les octets un à un - ]
    for i := 0 to ilen - 1 do
        Begin
            e := ser_WriteByte( iSerPort, BufPtr^[i], uTimeOut,
                               bSigMask, bSigVals );
            if e <> 0 then
                Begin
                    ser_WritePaket := e;
                    Exit;
                End;
            End;
        End;
    End;

```



```
Procedure ser_SetModemControl( iSerPort : Integer; bNewControl : Byte);
```

```
Begin
```

```
  port[iSerPort + SER_MODEM_CONTROL] := bNewControl;
```

```
End;
```

```
{*****}
{ ser_WriteByte : Envoi d'un octet }
{*****}
{ Entrée : iSerPort - Port de base de l'interface pour }
{           envoyer un octet. }
{           bData    - octet à envoyer }
{           uTimeout - Nombre de passages effectués dans la boucle }
{           avant que l'échec d'une émission n'est signalé }
{           par une erreur Timeout. }
{           (Si iTimeout = 0 l'attente continue }
{           bSigMask - Masque de bits des lignes de signal à tester }
{           (RTS, CTS, CD, RI) }
{           bSigVals - Etat des lignes de signal après avoir }
{           démasqué le masque ci-dessus. }
{ Sortie : = 0 - octets ont été envoyés }
{           <> 0 - Erreurs }
{*****}
```

```
Function ser_WriteByte( iSerPort      : Integer;
                        bData         : Byte;
                        uTimeout      : Integer;
                        bSigMask, bSigVals : Byte ) : Integer;
```

```
Begin
```

```
  if uTimeout <> 0 then { Boucle Timeout }
```

```
  Begin
```

```
    While( not ser_IsWritingPossible( iSerPort )
           and ( uTimeout <> 0 ) )
```

```
      do Dec( uTimeout );
```

```
    if uTimeout = 0 then
```

```
      Begin
```

```
        ser_WriteByte := SER_ERRTIMEOUT;
```

```
        Exit;
```

```
      End;
```

```
    End
```

```
  else { Attente ! }
```

```
    Repeat
```

```
    Until ser_IsWritingPossible( iSerPort );
```

```
  {-- test de lignes de signal -----}
```

```
  if ( port[iSerPort + SER_MODEM_STATUS] and bSigMask ) = bSigVals
  then
```

```
    Begin { transmission des octets à émettre vers l'interface }
```

```
      port[iSerPort + SER_TXBUFFER] := bData;
```

```
      { Retourne erreur interface }
```

```
      ser_WriteByte := port[iSerPort + SER_LINE_STATUS] and
                        SER_LSR_ERRORMSK;
```

```
    End
```

```
  else
```

```
    ser_WriteByte := SER_ERRSIGNALS;
```

```
End;
```

```
{*****}
{ ser_ReadByte : Réception d'un octet }
{*****}
{ Entrée : iSerPort - Port de base de l'interface utilisée }
{           pour recevoir un octet }
{           Data    - la variable byte accueille l'octet }
{           reçu. }
{           uTimeout - Nombre de passages dans la boucle }
{           avant qu'une erreur Timeout ne signale }
{           l'échec de la réception. }
{*****}
```


ANNEXE 4

```

Program series;
Uses cat;
Const
  ASCII_ACK = $06;
  ASCII_NAK = $15;
  SER_COM1 = $3F8;           [ Adresse de base COM1 ]
  SER_COM2 = $2F8;           [ Adresse de base COM2 ]

  SER_IRQ_COM1 = 4;           [ IRQ 4 = Vecteur $0C ]
  SER_IRQ_COM2 = 3;           [ IRQ 3 = Vecteur $0B ]

```

```

SER_TXBUFFER = $00;           [ Register Transmit ]
SER_RXBUFFER = $00;           [ Register Receive ]
SER_DIVISOR_LSB = $00;        [ Diviseur des taux de baud LSB ]
SER_DIVISOR_MSB = $01;        [ Diviseur des taux de baud MSB ]
SER_IRQ_ENABLE = $01;         [ Register d'interruption Enable ]
SER_IRQ_ID = $02;             [ Register d'Interruption ID ]
SER_FIFO = $02;               [ Register FIFO ]
SER_2Function = $02;          [ Register Alternate Function ]
SER_LINE_CONTROL = $03;       [ Contrôle des lignes ]
SER_MODEM_CONTROL = $04;       [ Contrôle du modem ]
SER_LINE_STATUS = $05;        [ Etat des lignes ]
SER_MODEM_STATUS = $06;       [ Modemstatus ]
SER_SCRATCH = $07;            [ Scratch-Register ]

```

[IRQ-Enable-Register-Bits (Interruptions en/hors service)]

```

SER_IER_RECEIVED = $01;       [ IRQ après réception des données ]
SER_IER_SENT = $02;           [ IRQ après l'envoi d'un bit ]
SER_IER_LINE = $04;           [ IRQ après modification de l'état de ligne ]
SER_IER_MODEM = $08;          [ IRQ après modification de l'état du modem ]

```

[IRQ-ID - Bits (Origine du IRQ ?)]

```

SER_ID_PENDING = $01;         [ Le IRQ est il serial ? ]
SER_ID_MASK = $06;            [ codage de l'ID en Bits 1 et 2 ]
SER_ID_LINESTATUS = $06;       [ Etat de lignes (Error ou Break) ]
SER_ID_RECEIVED = $04;         [ recevoir données ]
SER_ID_SENT = $02;            [ Byte a été envoyé ]
SER_ID_MODEMSTATUS = $00;      [ modification de CTS, DSR, RI ou RLSD ]

```

[Occupation de bits dans le registre FIFO]

[il faut au moins une UART 16550A]

```

SER_FIFO_ENABLE = $01;
SER_FIFO_RESETRCEIVE = $02;
SER_FIFO_RESETRANSHIT = $04;

```

[FIFO-Bits (L'IRQ se déclenche après ... bits dans le FIFO :)]

```

SER_FIFO_TRIGGER0 = $00;       [ Normal ]
SER_FIFO_TRIGGER4 = $40;       [ 4 Bytes ]
SER_FIFO_TRIGGER8 = $80;       [ 8 Bytes ]
SER_FIFO_TRIGGER14 = $C0;      [ 14 Bytes ]

```

[Register du contrôle de ligne - Bits (paramètres de transmission)]

```

SER_LCR_WordLEN = $03;         [ nombre de bits à transmettre ]
SER_LCR_5BITS = $00;
SER_LCR_6BITS = $01;
SER_LCR_7BITS = $02;
SER_LCR_8BITS = $03;
SER_LCR_2STOPBITS = $04;       [ 2 bzw. 1.5 bits de stop ]
SER_LCR_1STOPBIT = $00;        [ 1 Bit de stop ]

```

```

SER_LCR_NOPARITY = $00;         [ desactive le contrôle parité ]
SER_LCR_ODDPARITY = $08;        [ parité impaire ]
SER_LCR_EVENPARITY = $18;       [ parité paire ]
SER_LCR_PARITYSET = $28;        [ le bit de parité est tjs présent ]
SER_LCR_PARITYCLR = $38;        [ le bit de parité est tjs absent ]
SER_LCR_PARITYEN = $35;

```