

Institut Supérieur des Etudes Technologiques en Communications de Tunis

Projet de fin d'études

**INDICATION PAR SYNTHÈSE VOCALE DU NUMÉRO DE
L'APPELANT POUR PERSONNES NON VOYANTES**

Réalisé par :

**Mansour Mahassen
&
Azaza Feyza**

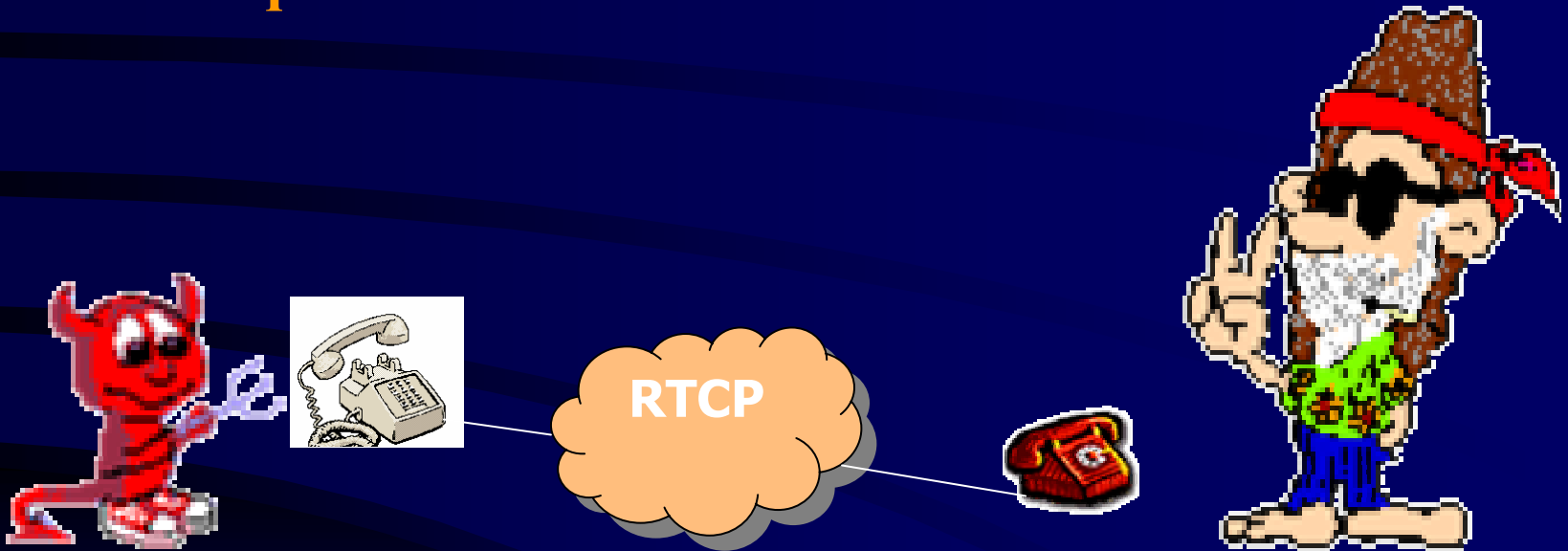
TS5- Télécommunications

Encadrées par :

**Mr. Ridha Bouallegue
Mlle Hela Gneba**

2001-2002

Problématique :



- Possibilité d'offrir le service CLIP pour personnes non voyantes.

Indication par synthèse vocale du numéro de l'appelant pour personnes non voyantes

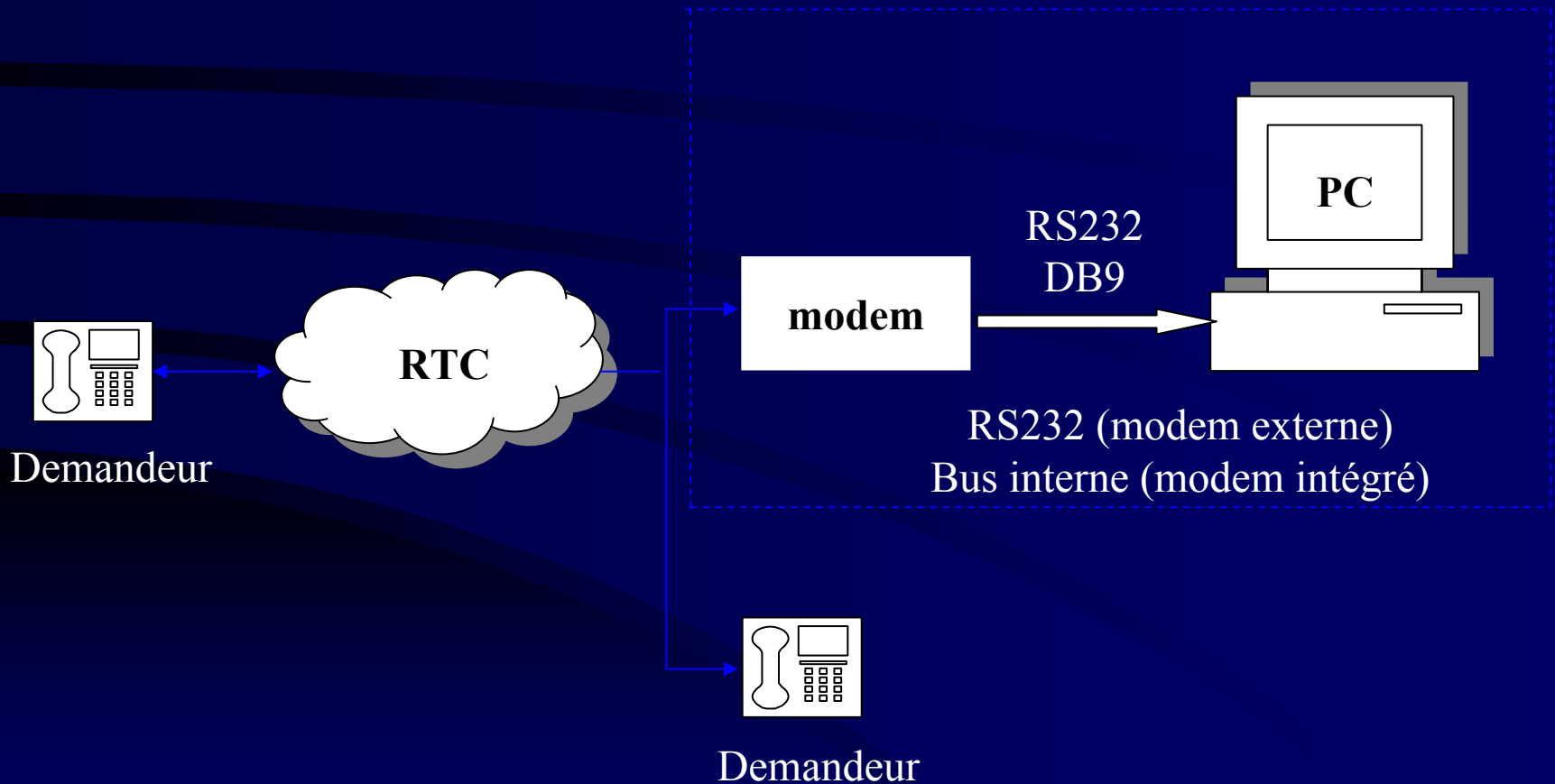
- **Réalisé par :**
 - **MANSOUR Mahassen**
 - **AZAZA Feyza**
- **Encadrées par :**
 - **Mr. BOUALLEGUE Ridha**
 - **Mlle. GNEBA Hela**

Année Universitaire : 2001-2002

Plan de l'exposé

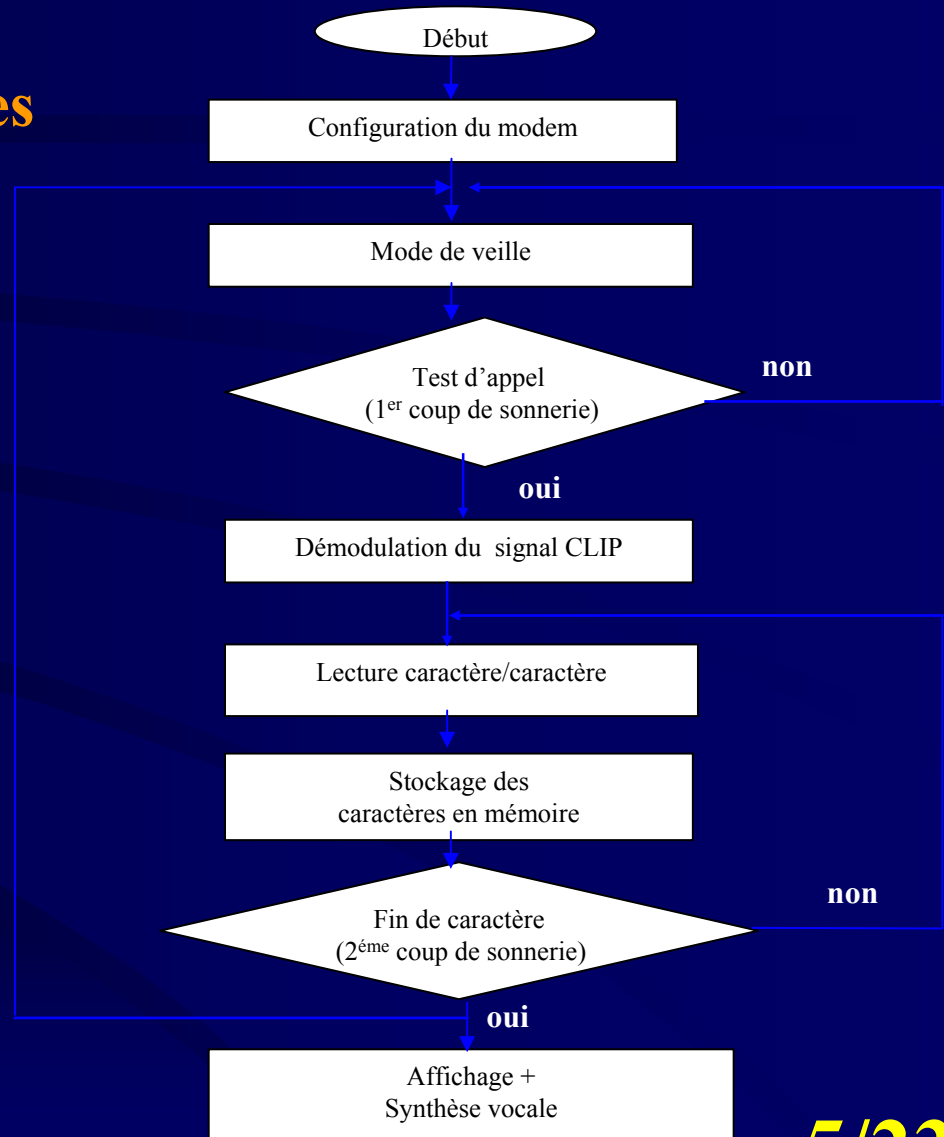
- ❖ **Synoptique du cahier des charges**
- ❖ **Etude théorique**
- ❖ **Etude de conception**
- ❖ **Etude de réalisation**
- ❖ **Conclusion et perspectives**

Synoptique du cahier des charges



Etude théorique (1/9)

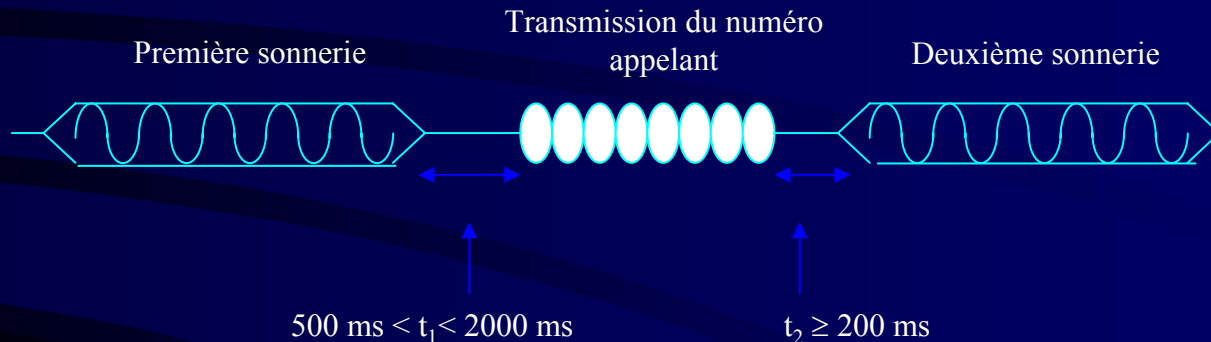
1. Transcription du cahier des charges :



Etude théorique (2/9)

2. Le service CLIP :

- En quoi consiste le service CLIP ?
- Procédure d'émission du message CLIP :



- Mode de transmission du message CLIP :

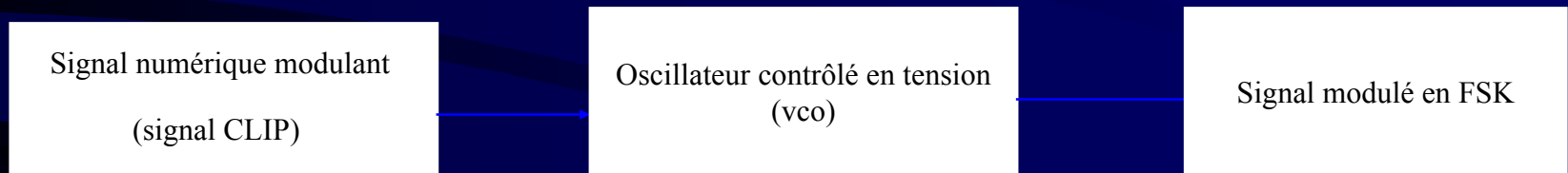
Etude théorique (3/9)

Modulation FSK :

Problème : dégradation du signal

Solution : la modulation FSK.

- La modulation FSK :
 - f_0 → « 0 » logique
 - f_1 → « 1 » logique
- Un modulateur FSK peut être schématisé comme suit :



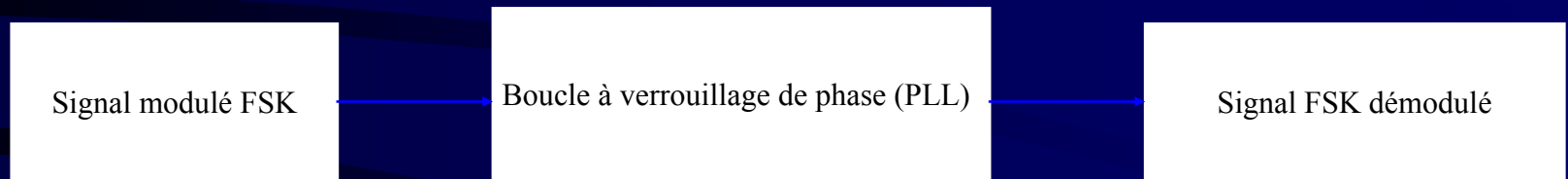
La liaison série asynchrone :

- Principe de fonctionnement d'une liaison série
- Le port série
- La norme RS232

Etude théorique (4/9)

Démodulation FSK :

- Une boucle à verrouillage de phase (PLL).
- Un démodulateur peut être schématisé comme suit :



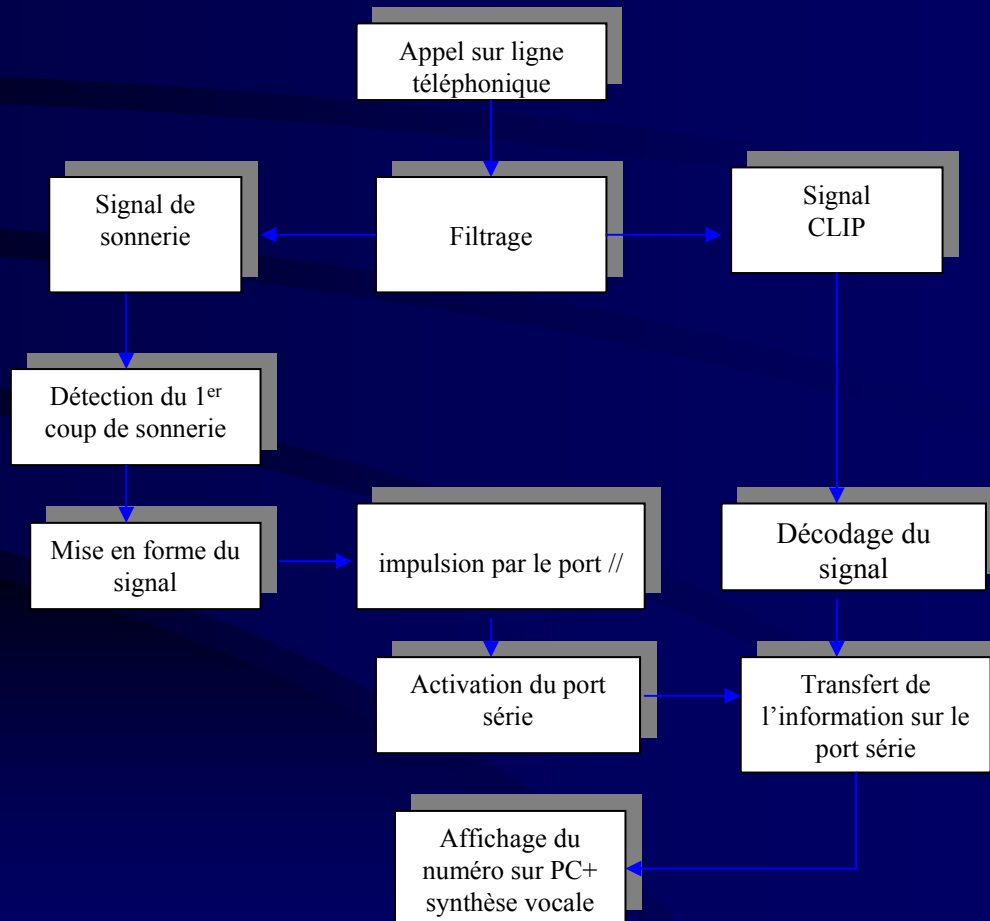
Les éléments constitutifs de ce circuit (PLL) :

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

Etude théorique (5/9)

3. Solution matérielle :

- Synoptique du cahier des charges :

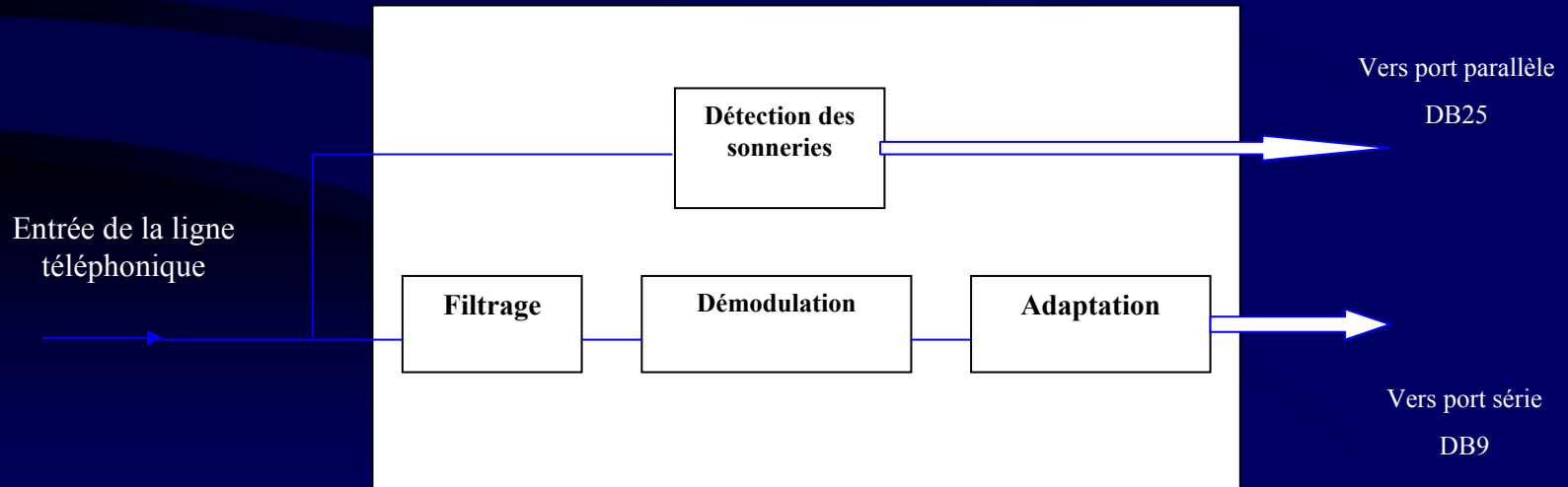


Etude théorique (6/9)

- **Objectifs de cette solution :**

- Etude et conception d'une carte électronique.
- Développement d'une application informatique

- **Carte électronique réalisée :**



Etude théorique (7/9)

4. Solution logicielle :

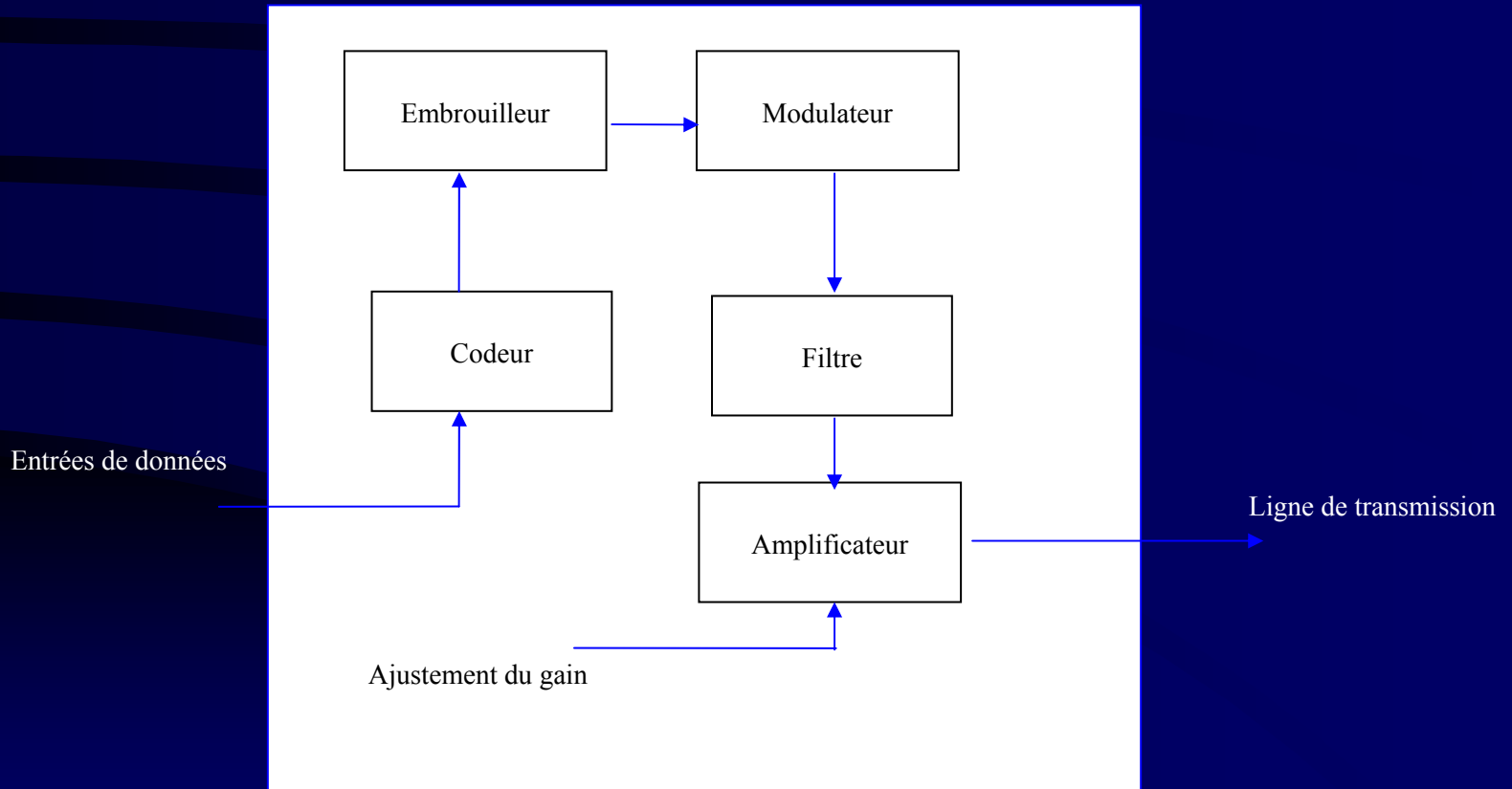
Les objectifs de cette solution sont déjà définis dans notre cahier des charges. Elle consiste à élaborer une interface logicielle qui permet la programmation d'une carte modem intégrée afin d'assurer :

- L'extraction des signaux de sonnerie et des signaux CLIP
- L'affichage du numéro de l'appelant
- Sa synthèse vocale.

Etude théorique (8/9)

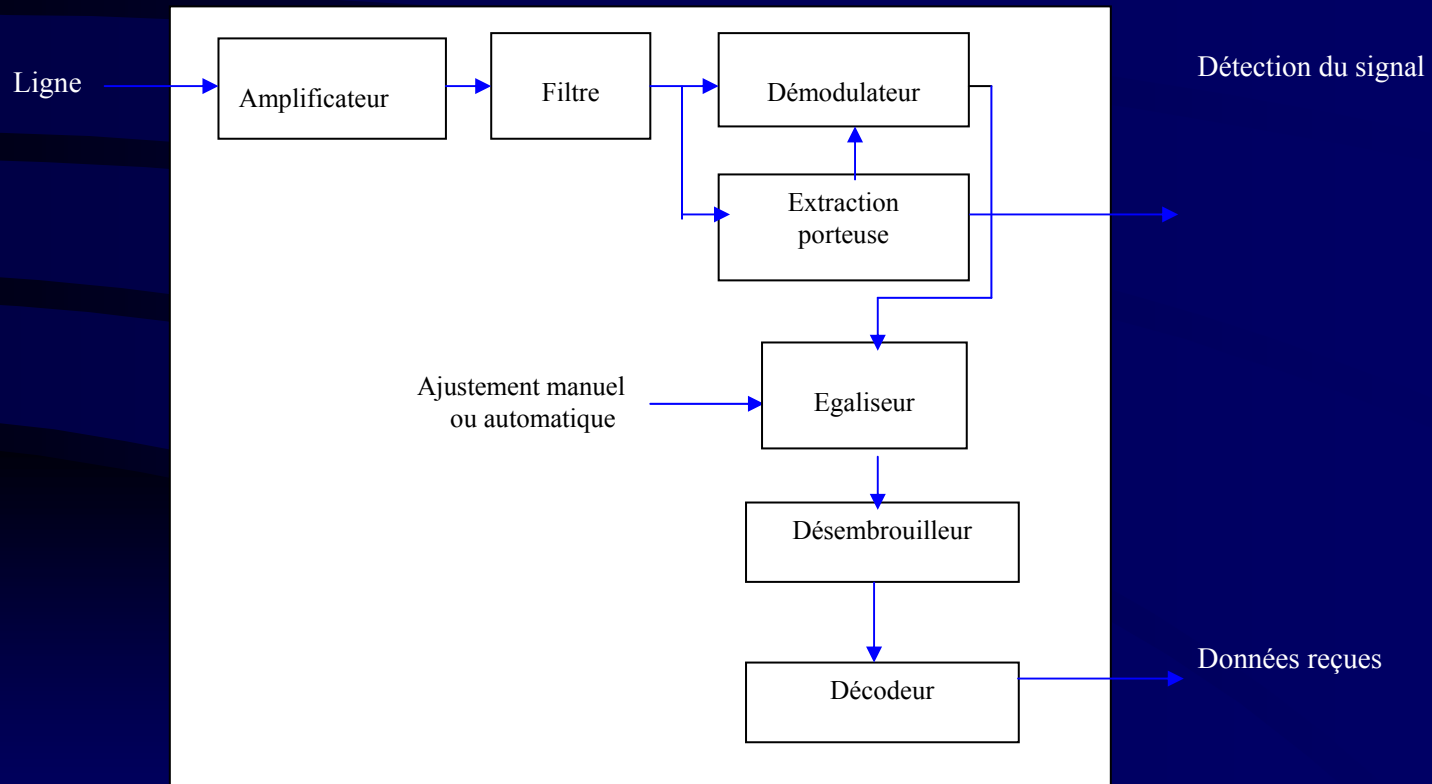
5. Les modems :

- Principe des modems en émission :



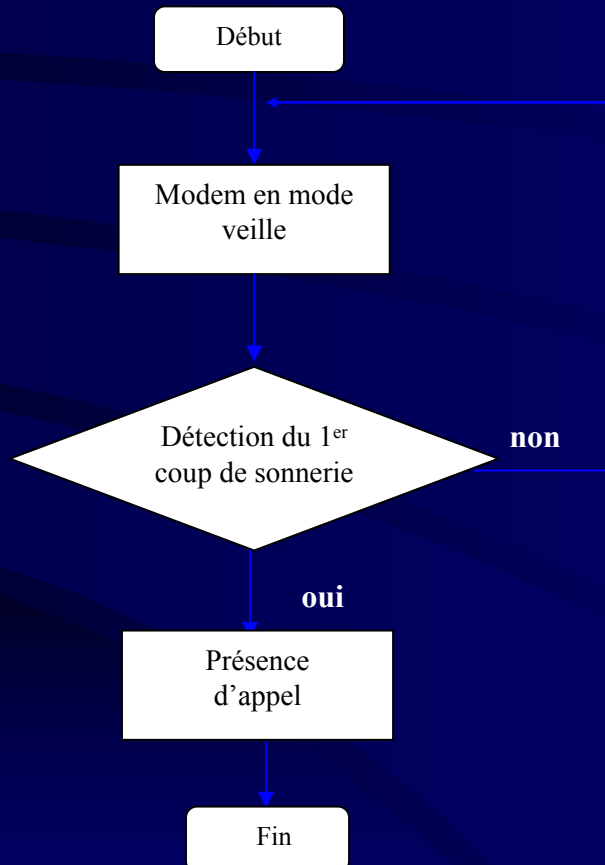
Etude théorique (9/9)

- Principe des modems en réception :



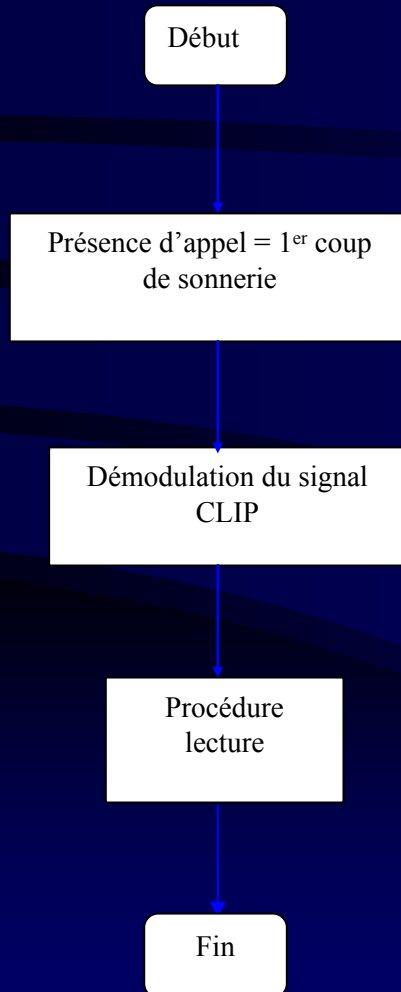
Etude de conception (1/6)

1. Extraction des signaux de sonnerie :

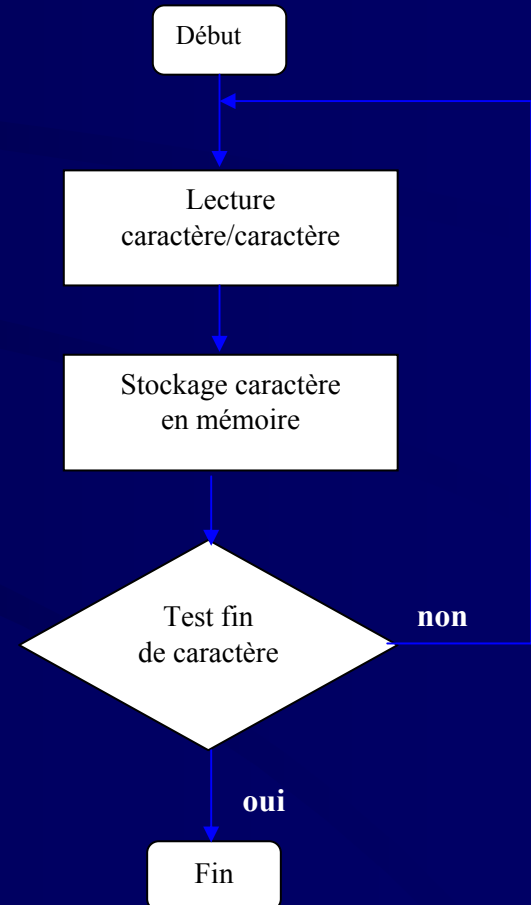


Etude de conception (2/6)

2. Extraction des signaux CLIP :

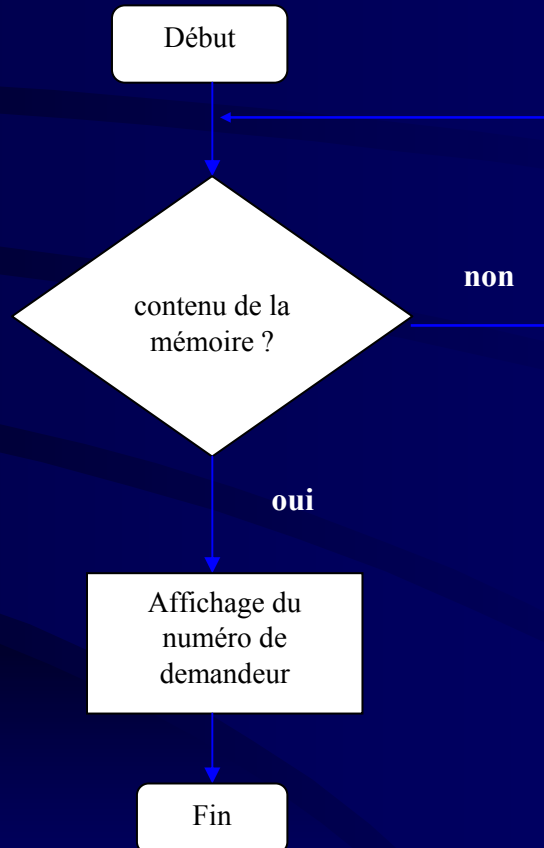


• Procédure de lecture :



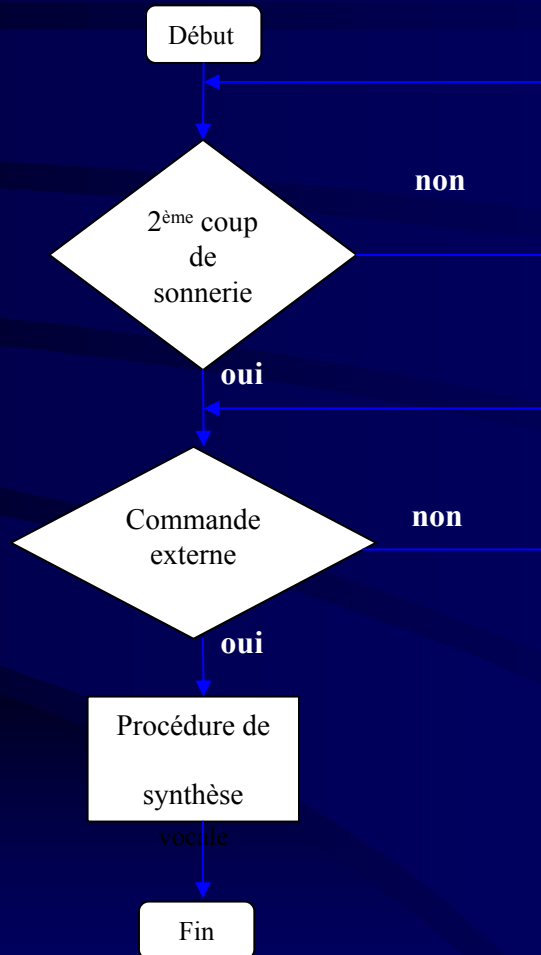
Etude de conception (3/6)

3. Affichage du numéro de l'appelant :



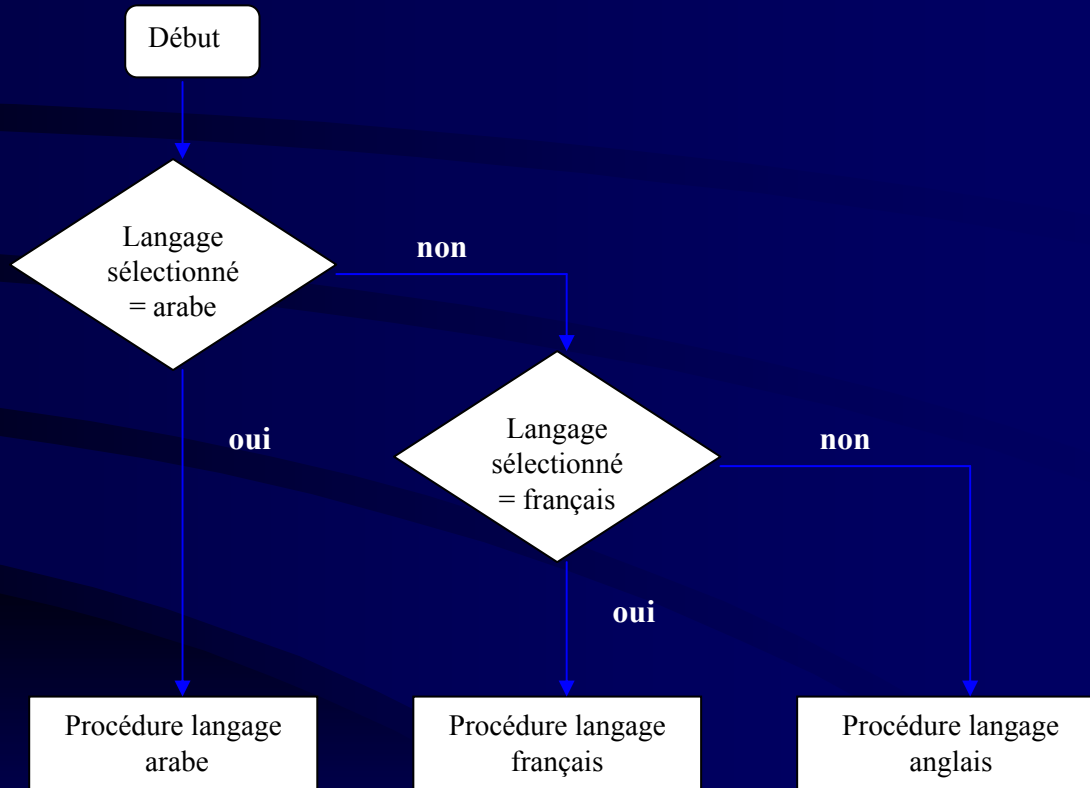
Etude de conception (4/6)

4. Synthèse vocale :



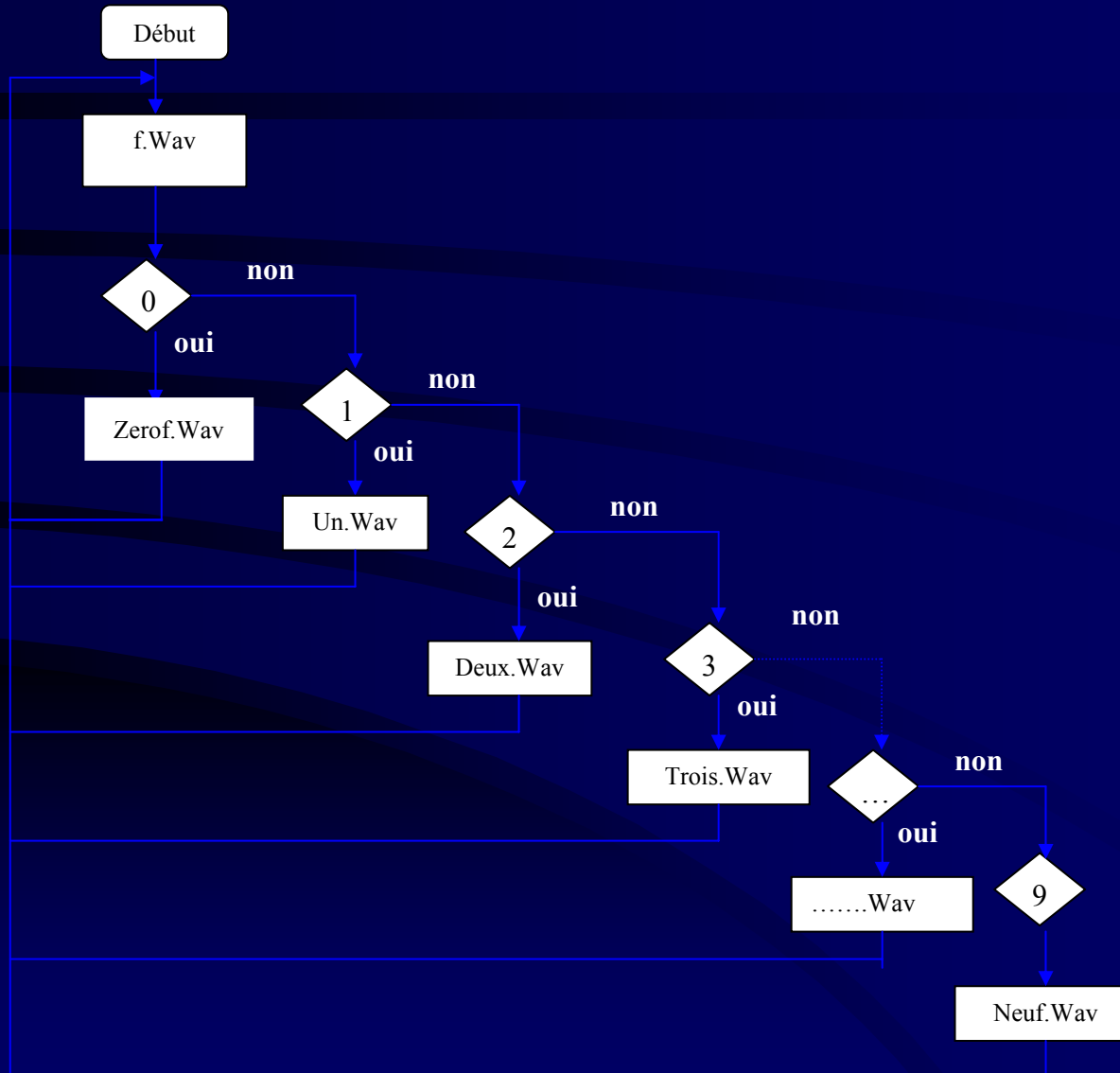
Etude de conception (5/6)

- Procédure de la synthèse vocale :



Etude de conception (6/6)

- Procédure du langage français :



Etude de réalisation (1/3)

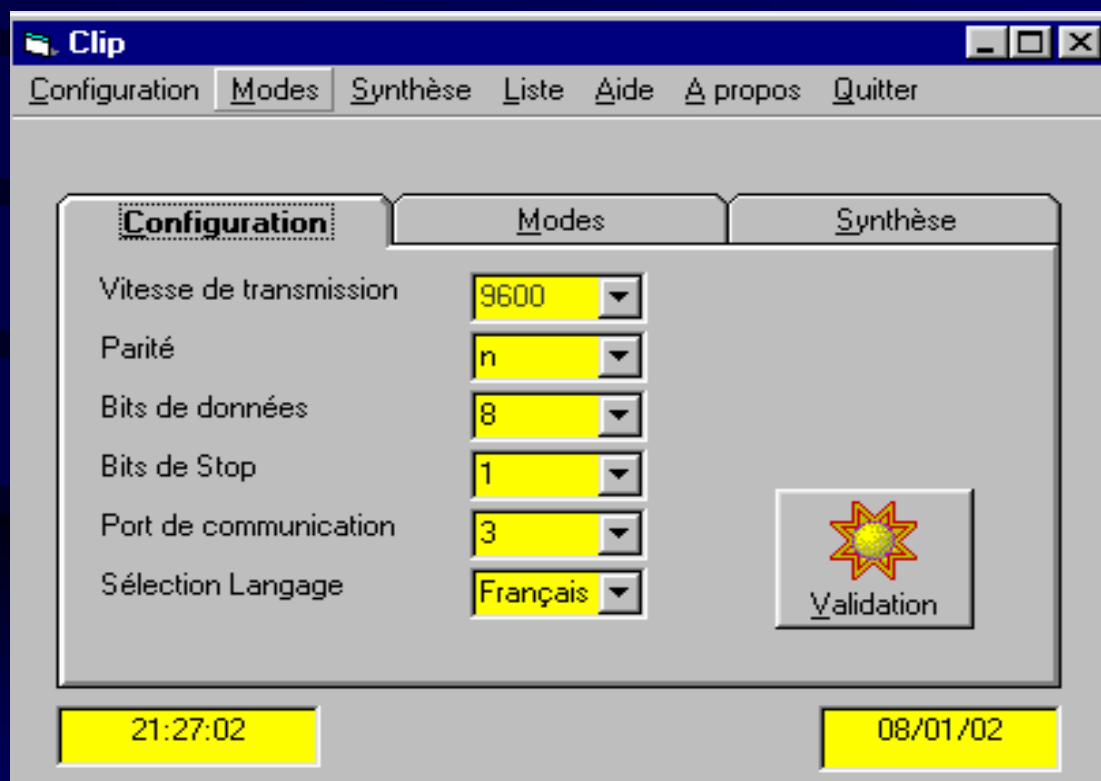
1. Organigramme de programmation de l'interface :



Etude de réalisation (2/3)

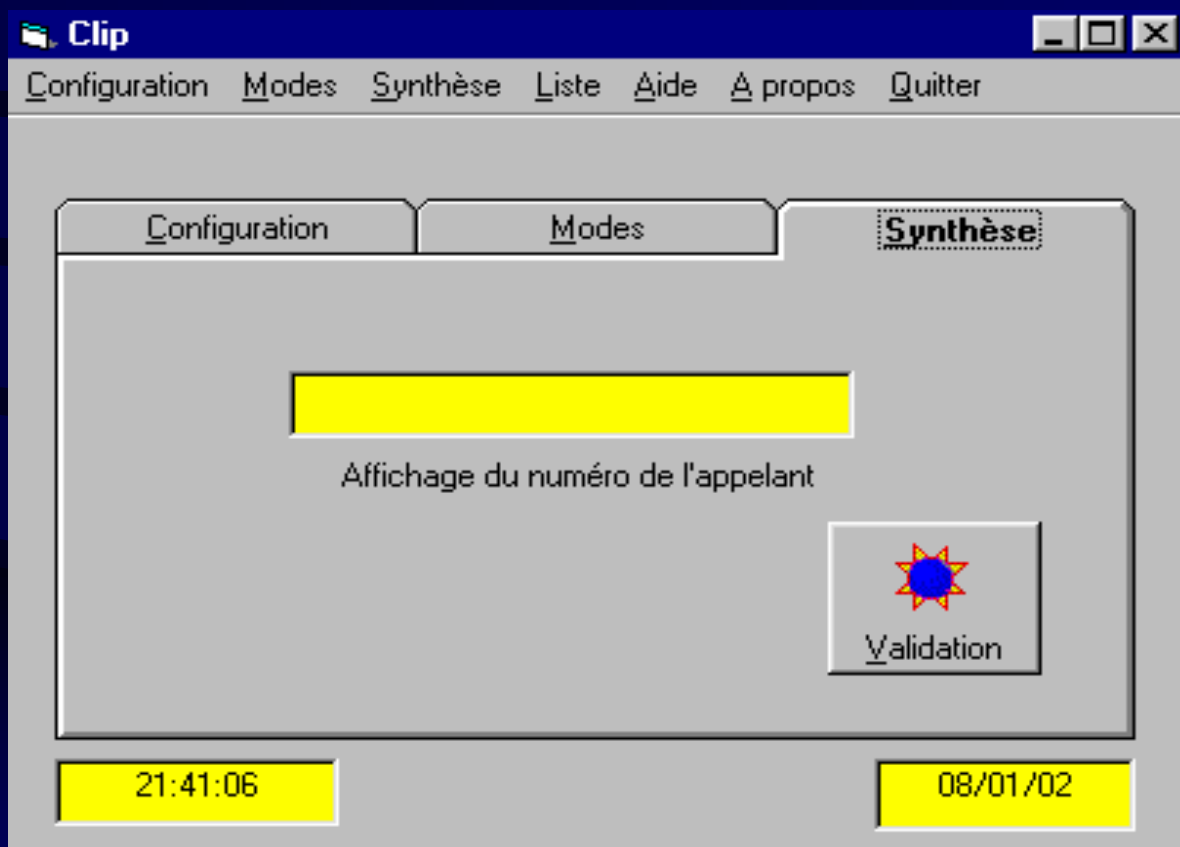
2. Programmation :

- Fenêtre principale :



Etude de réalisation (3/3)

- Menu synthèse :



Conclusion & perspectives

Conclusion :

- Etude de conception et de réalisation d'un prototype logiciel.
- Utilisation d'un composant programmable, qui est le modem, et un PC piloté par un programme Visual Basic.

Perspectives :

- Application autonome et indépendante d'un équipement informatique.
- Développement d'une carte à base d'un microcontrôleur et d'un afficheur à cristaux liquides inséré directement sur la ligne téléphonique et moyennant un synthétiseur vocal.

Dédicaces

A ma chère mère Mabrouka

*Pour l'amour, la récompense et les sacrifices qu'elle a contenté
pour moi.*

A mon cher père Salah

*En témoignage de ma sincère reconnaissance pour les efforts, les
sacrifices qu'il a consenti pour la famille et l'accomplissement de
mes études.*

A mes frères et mes sœurs

*Qu'ils trouvent dans ce travail tous les signes de ma profonde
affective, en leurs souhaitant plein succès dans leurs vie.*

A L'âme de mon cher frère Miloud.

Je dédie ce modeste travail.

 Feyza

Dédicaces

A ma mère qui a tout souffert sans me faire souffrir,

A mon père que je dois ma réussite,

Que dieu les protège et leurs préserve bonheur et santé.

A mes frères


A mes sœurs

A mon amour

A mes amis

Pour tous leurs sacrifices

Je dédie ce modeste travail.

Mahassen

Remerciements

Pour tous ceux qui nous ont aidé de près ou de loin à bien mener et à profiter de ce travail trouvent ici l'expression de nos sincères remerciements et de nos profondes gratitude. Particulièrement, nous tenons à remercier :

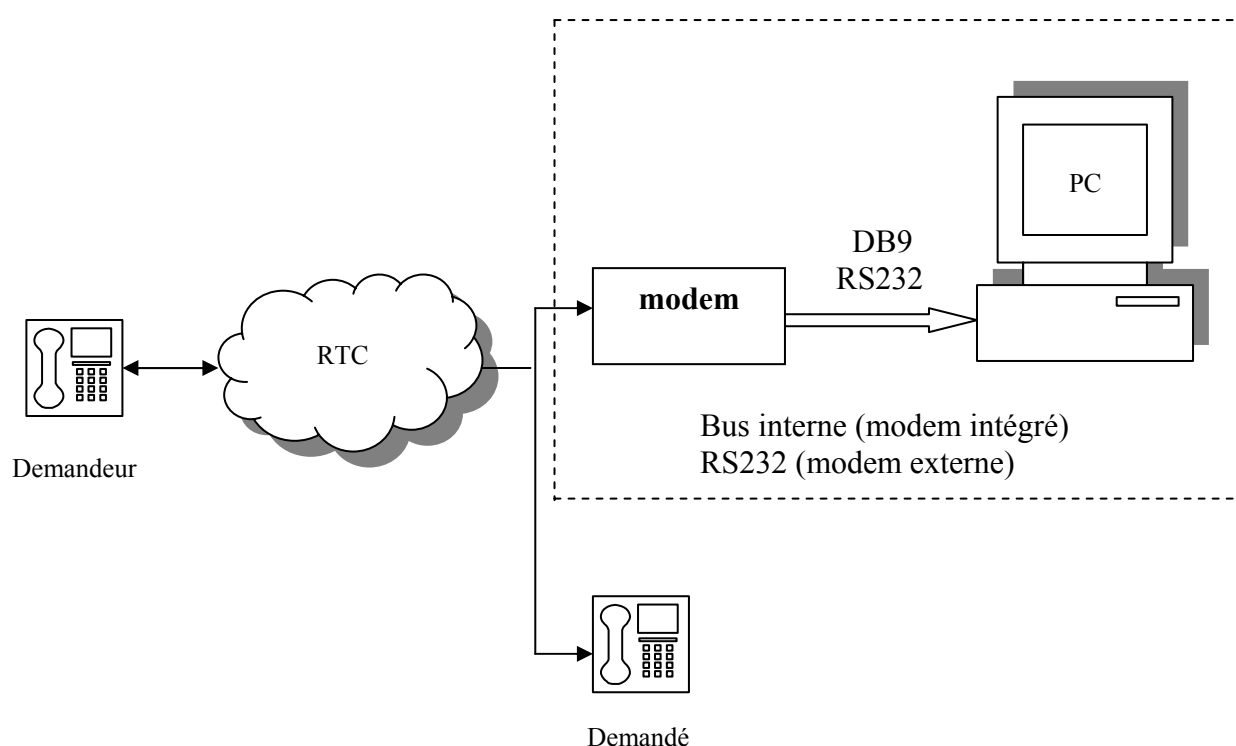
- *Nos encadreurs Mr Bouallegue Ridha et Mlle Gneba Hela pour leurs écoutes, leurs patiences leurs conseils et leurs disponibilités tout le long de ce travail malgré leurs occupations.*
- *L'ISSET'COM ainsi que tous nos enseignants pour les précieuses orientations qu'ils ont bien voulu nous fournir.*
- *Les membres de jurys que nous avons honneur qu'ils acceptent de juger ce modeste travail.*



Feyza et Mahassen

Cahier des charges

L'objectif de notre projet de fin d'études consiste à élaborer une interface logicielle pour la programmation d'un modem permettant à un abonné non voyant, raccordé au réseau téléphonique commuté (RTC), d'identifier le numéro de son correspondant par voie vocale.



Synoptique du cahier des charges

Pour cela, nous sommes amenés à réaliser les étapes suivantes :

- ◆ Etude de conception des divers organigrammes assurant l'extraction des signaux de sonnerie et des signaux CLIP, l'affichage du numéro de l'appelant et sa synthèse vocale.
- ◆ Développement d'une application logicielle permettant la gestion du modem (sa configuration) et la traduction de ces organigrammes en programmation Visual Basic afin d'aboutir à la synthèse vocale du numéro de l'appelant.

Sommaire

Introduction générale	1
Cahier des charges	3
Chapitre I : Etude théorique	4
I. Transcription du cahier des charges	4
II. Le service CLIP	6
II.1. En quoi consiste le service CLIP	6
II.2. Procédure d'émission du message CLIP	7
II.3. Mode de transmission du message CLIP	8
II.3.1. La liaison série asynchrone	8
II.3.2. Le port série	10
II.3.3. La norme RS232	12
III. Présentation des solutions proposées pour notre projet	13
III.1. Présentation de la solution matérielle	13
III.1.1. Synoptique du cahier des charges	14
III.1.2. Carte électronique réalisée	16
III.2. Présentation de la solution logicielle	18
IV. Les modems	18
IV.1. Principe des modems	18
IV.1.1. Modem en émission	18
IV.1.2. Modem en réception	20
IV.2. Caractéristiques d'un modem	21
IV.3. La modulation et la démodulation	22
IV.3.1. La modulation	22
IV.3.2. La démodulation	25
IV.4. Présentation de la boucle à verrouillage de phase (PLL)	26
IV.4.1. Généralités	26
IV.4.2. Principe de fonctionnement	27
V. Les commandes Hayes	29
V.1. Historique	29
V.2. L'ensemble des commandes Hayes	30
V.3. Liste des commandes Hayes	30
V.4. Règles syntaxiques d'écriture d'une commande Hayes	32
V.5. Programmation des modems, le code AT	33
VI. La synthèse vocale	34
 Chapitre II : Etude de conception	 36
I. Extraction des signaux de sonnerie	36
II. Extraction des signaux CLIP	37
III. Affichage du numéro de l'appelant	39
IV. Synthèse vocale	39

Chapitre III : Etude de réalisation	45
I. Présentation du logiciel de développement	47
I.1. Critères du choix	47
I.2. Présentation du Visual Basic	47
I.3. Description des objets	48
II. Programmation	50
II.1. Création des fenêtres	50
II.2. Fenêtre principale	51
II.3. Les procédures de programmation	55
II.3.1. Procédure de configuration du port série	55
II.3.2. Procédure du choix de modes	55
II.3.3. Procédure d'appels des formes	55
II.3.4. Procédure du temps et de la date	56
II.3.5. Procédure de fermeture des formes	56
II.3.6. Procédure de synthèse vocale	57
Conclusion générale	64
Bibliographie	

Liste des tableaux et figures

Fig.1.1. Transcription du cahier des charges	5
Fig.1.2. Dispositif d'affichage du numéro du demandeur.....	7
Fig.1.3. Méthode de transmission du numéro de l'appelant	7
Fig.1.4. Signal de sonnerie	8
Fig.1.5. Transmission série asynchrone	9
Tableau.1.1. Attribution des broches	11
Fig.1.6. Schéma synoptique du projet par solution Hard	14
Fig.1.7. Synoptique du cahier des charges	15
Fig.1.8. Synoptique de la carte réalisée	16
Fig.1.9. Courbe 1	17
Fig.1.10. Courbe 2	17
Fig.1.11. Schéma de principe d'un modem en émission	19
Fig.1.12. Schéma de principe d'un modem en réception	20
Tableau.1.2. Normes et vitesses de transmission	22
Fig.1.13. Modulation en fréquence du signal	24
Fig.1.14. Schéma fonctionnel d'un modulateur	24
Fig.1.15. Schéma de principe d'un modulateur /démodulateur	25
Fig.1.16. Schéma fonctionnel d'un démodulateur	26
Fig.1.17. La boucle à verrouillage de phase	27
Fig.1.18. Schéma d'un comparateur de phase	28
Fig.1.19. Schéma global du VCO	29
Tableau.1.3. Messages Hayes	34
Fig.2.1. Organigramme d'extraction des signaux de sonnerie	37
Fig.2.2. Organigramme d'extraction des signaux CLIP.....	38
Fig.2.3. Organigramme de la procédure de lecture.....	38
Fig.2.4. Organigramme d'affichage du numéro de l'appelant	39
Fig.2.5. Organigramme de la synthèse vocale	40
Fig.2.6. Organigramme de la procédure de synthèse vocale	41
Fig. 3.1. Organigramme de programmation de l'interface	46
Fig.3.2. L'environnement de développement du Visual Basic	47

Introduction générale

Dans les années 60 et au début des années 70, les services en matière de télécommunications étaient relativement limités. Ils y avaient, essentiellement, le téléphone dans le domaine de la voix et le télex dans celui de l'écrit.

L'évolution très rapide des besoins des utilisateurs a généré la création d'un grand nombre de nouveaux services tels que la télécopie, la transmission de donnéesetc.

L'environnement des systèmes de télécommunications évolue actuellement vers des systèmes qui fournissent une grande mobilité, une flexibilité des services et des fonctionnalités contrôlables directement par le client. Ceci mène à des situations où les opérateurs de télécommunications doivent déployer de nouveaux services pour répondre à la demande grandissante des utilisateurs. Ce nouveau environnement impose le passage des anciennes architectures de commutateurs vers des systèmes répartis qui sont dynamiquement configurables. Il est nécessaire donc de concevoir de nouvelles architectures de réseaux, de nouveaux modems, techniques et outils pour faire face à cette croissance du déploiement de nouveaux services.

Ces services peuvent varier d'un simple appel de base à un service de vidéo à la demande aussi bien sur un réseau fixe que sur un réseau mobile. En plus, les réseaux modernes ont pu offrir une large gamme de services supplémentaires à leurs abonnés soit en phase de conversation (identification d'appels malveillant, appel en attente, conférence...) soit en phase d'établissement d'appel (transfert d'appel sur non réponse ou sur occupation, identification de la ligne appelante...).

Parmi les services supplémentaires offerts dans les réseaux RNIS, GSM ... et bientôt le RTCP est le service CLIP.

En quoi consiste ce service ?

Comment permettre à une personne non voyante de profiter d'un tel service ?

La réponse à ces questions fera l'objet de notre projet de fin d'études. Afin d'atteindre ces objectifs, le présent rapport est découpé en trois chapitres:

- ◆ Le premier chapitre traitera l'étude théorique du service CLIP, des modems et la présentation d'une solution hard (matérielle).
- ◆ Le deuxième chapitre portera sur la conception des divers organigrammes permettant l'extraction des signaux de sonnerie et des signaux CLIP, l'affichage du numéro de l'appelant et sa synthèse vocale.
- ◆ Le troisième chapitre sera consacré pour le développement d'une interface logicielle permettant la traduction de l'information récupérée en un signal visuel et vocal.

A rectangular area with a marbled paper pattern in shades of grey, white, and light brown. The text is centered within this area.

Chapitre I

Etude Théorique

Chapitre1

Etude théorique

Introduction :

Lors d'un appel, le signal sur la ligne téléphonique du demandé résulte de la superposition de la sonnerie et du signal CLIP.

A ce propos, nous nous intéressons au signal d'identification de l'appelant et à son traitement pour pouvoir le récupérer à la réception et le traduire en un message vocal.

I. Transcription du cahier des charges :

Le signal CLIP est modulé dans le central selon le principe de la modulation par déplacement de fréquence (FSK). A la réception, un modem est nécessaire pour assurer la démodulation de ce signal, l'affichage du numéro de demandeur sur le PC et sa synthèse vocale.

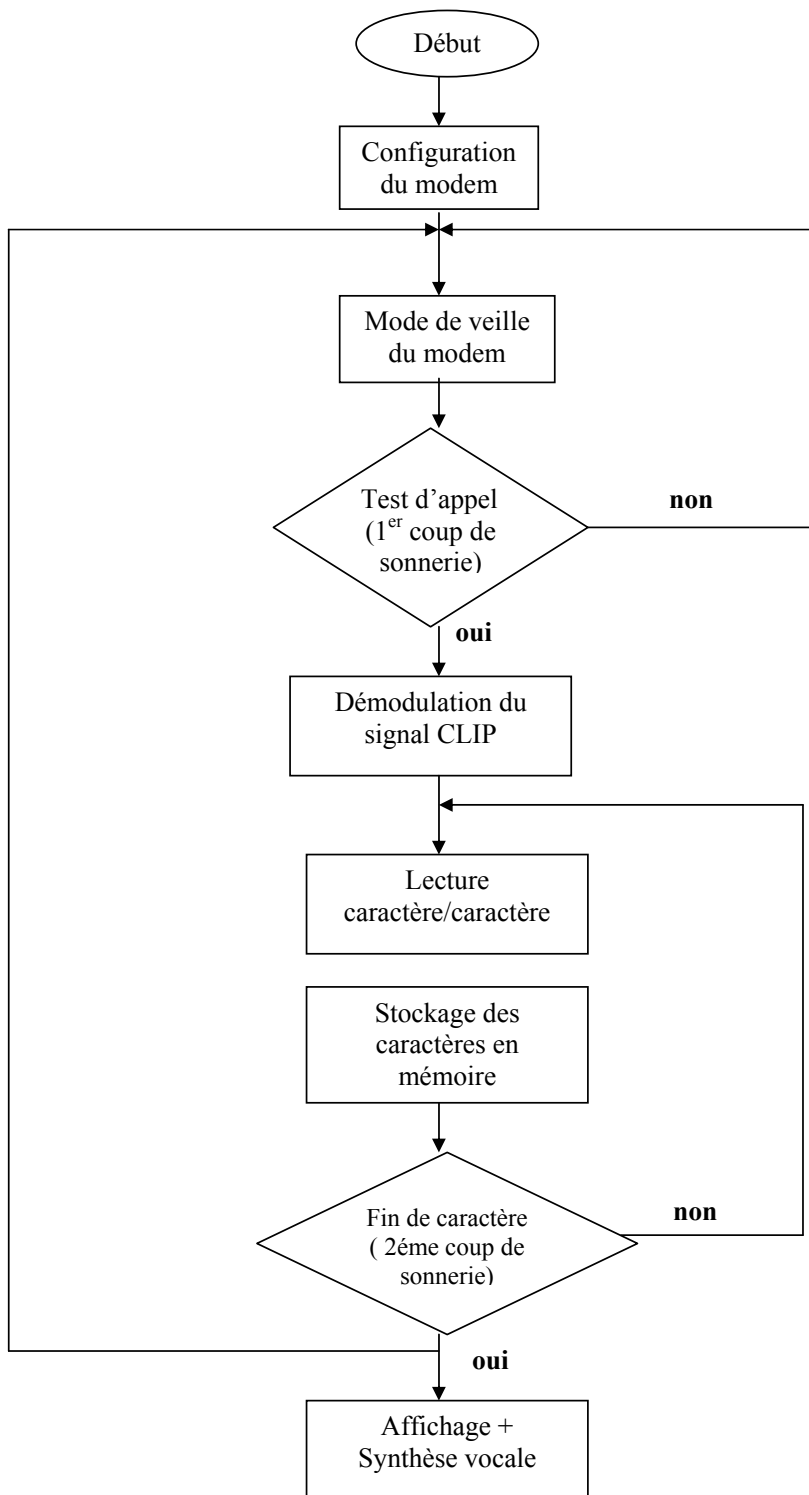


Fig.1.1. Transcription du cahier des charges

II. Le service CLIP :

Introduction :

Dans le secteur de télécommunication, l’affichage des numéros de téléphone des appelants est l’une des nouvelles technologies qui permet d’intervenir dans la vie privée des gens. Cette facilité est appelée identification de la ligne appelante ou service CLIP (Calling Line Identification Presentation).

II .1. En quoi consiste le service CLIP :

Le service CLIP permet à l’abonné demandé de voir apparaître le numéro d’appel ou le nom de son correspondant même avant de décrocher. Ce qui lui donne le choix de décider s’il va répondre ou non à un tel appel. Les utilisateurs GSM (système de téléphonie mobile) et tous les utilisateurs d’appareils fixes munis d’une ligne RNIS (réseau numérique à intégration de service) peuvent également voir apparaître sur l’écran de leurs appareils le numéro d’appel.

Pour se bénéficier de ce service, l’abonné doit disposer d’un terminal spécial ou d’un équipement supplémentaire permettant le décodage, l’affichage et l’emmagasinement des numéros.

Si toutefois, l’abonné demandeur ne désire pas diffuser son numéro, il suffit de demander à la clientèle d’activer pour lui le service CLIR (Calling Line Identification Restriction).

Par conséquent, le service CLIP est offert par le réseau RNIS et le réseau GSM et bientôt par le réseau RTCP (Réseau Téléphonique Commuté Public).

Le dispositif d’affichage du numéro de l’appelant est schématisé par la figure (1.2).

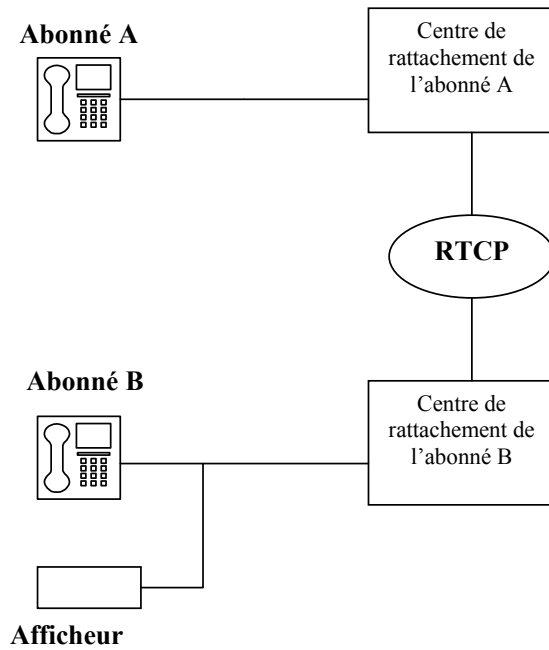


Fig.1.2. Dispositif d'affichage du numéro du demandeur

II.2. Procédure d'émission du message CLIP :

Le message CLIP est transmis durant la première pause de sonnerie. La transmission de ce message commence au moins 500 millisecondes (ms) après la première sonnerie, mais cette pause ne doit pas dépasser les 2000 ms. Une autre pause d'au moins 200 ms doit également être observée à la fin de la transmission.

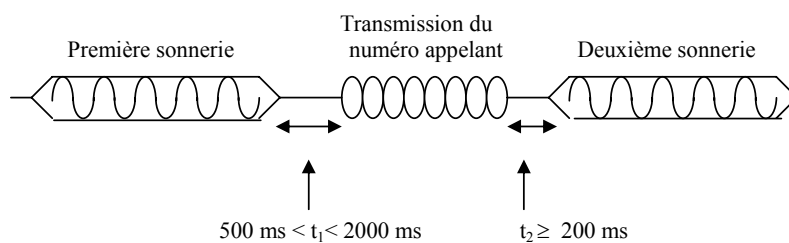


Fig.1.3. Méthode de transmission du numéro appelant

Par conséquent, un appel reçu se manifeste par la superposition de deux signaux : le signal CLIP et le signal de sonnerie qui se caractérise par des signaux de même sens de l'ordre de 80 à 100 volts (v) et d'une fréquence de l'ordre de 25 à 50 Hz, à la cadence de 1.7 s de présence et 3.3 s d'absence.

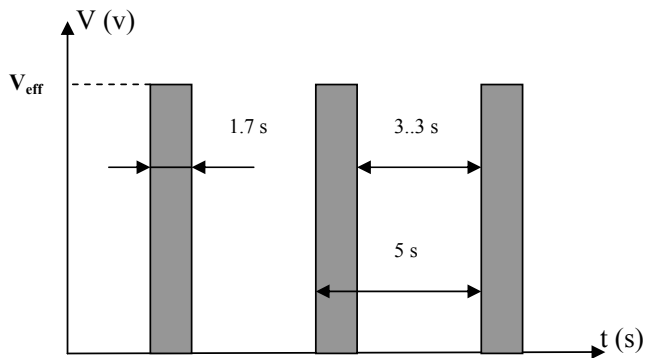


Fig.1.4. Signal de sonnerie

II.3. Mode de transmission du message CLIP :

Le signal CLIP est un signal numérique émis par des émetteurs FSK dans le central. Il est transmis dans la bande 300-3400 Hz, selon le principe de la modulation par déplacement de fréquence (FSK : Frequency Shift Keying), en mode série asynchrone. Du côté réception, l'abonné demandé dispose d'un modem pour la démodulation de ce signal et la restitution de l'information numérique.

II.3.1. La liaison série asynchrone :

La liaison série asynchrone s'effectue caractère par caractère qui sont séparés par un intervalle de temps variable et à basse vitesse. La longueur d'un caractère est comprise entre 4 et 8 bits. Ce type de transmission nécessite l'ajout d'un bit **START** (bit de départ) dont l'état logique est toujours « 0 », un ou deux bits de **STOP** (bits d'arrêt) dont l'état logique est « 1 », et un bit de parité pour détecter les erreurs de transmission.

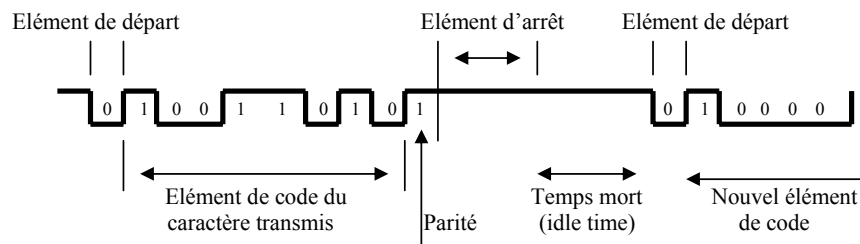


Fig.1.5. Transmission série asynchrone

❖ Le principe de fonctionnement d'une liaison série :

La communication série nécessite trois fils au minimum : une masse pour référencer les signaux, un fil émetteur et un fil récepteur. Notre liaison série est en effet full- duplex, c'est à dire que l'on peut émettre et recevoir en même temps (comme le téléphone par exemple).

La différence principale entre le port parallèle et le port série est que les informations ne sont pas transmises simultanément sur des fils séparés (D0 à D7) mais les unes après les autres sur un même fil. Cela amène à une économie de câble (un fil au lieu de 8) mais un montage décodeur devient nécessaire pour retransformer les données sérialisées.

Lors d'une communication série, les différents paramètres rentrant en jeu sont :

- **longueur du mot** : sur le PC, le BIOS ne permet une longueur de mot que de 7 ou 8 bits.
- **Parité** : le mot transmis peut être suivi d'un bit de parité qui sert à détecter les erreurs éventuelles de transmission. Il existe deux parités : la parité paire et la parité impaire. Dans le cas de la parité paire, et pour le mot 10110101 contenant 5 états à « 1 », le bit de parité sera « 1 » amenant ainsi le nombre total de « 1 » à un nombre pair (6). Dans le cas de la parité impaire, le bit de parité aurait été « 0 » car le nombre total de « 1 » est déjà impair. L'intérêt de ce rajout est le suivant : si jamais lors de la transmission, un état « 1 » est transformé en état « 0 » (par exemple perturbation du canal par des parasites), le nombre total de « 1 » change et donc le bit de parité recalculé par le récepteur ne correspond plus à celui reçu. L'erreur est donc détectée. Evidemment, si deux états à « 1 » passent à « 0 », l'erreur ne sera pas détectée mais la probabilité pour que cela arrive est très faible.

- **Bit de start** : lorsque rien ne circule sur la ligne, celle-ci est à l'état haut. Pour indiquer qu'un mot va être transmis, la ligne passe à l'état bas avant de commencer le transfert. Cette précaution permet de resynchroniser le récepteur.
- **Bits de stop** : ces bits signalent la fin de la transmission. Selon le protocole utilisé, il peut y avoir 1, 1.5, ou 2 bits de stop (ces bits sont toujours à « 1 »).
- **Vitesse de transmission** : la plupart des cartes série permettent de choisir une vitesse entre 300 et 9600 bauds (par exemple à 300 bauds, un bit est transmis tout les un trois-centième de seconde). D'autre part, la liaison série est avant tout censée pour les liaisons téléphoniques par modems, dont la bande passante est très limitée.

II.3.2. Le port série :

Le port série permet un grand nombre d'applications avec une faible contrainte d'encombrement au point de vue fils de connexions. En outre, la prise série est généralement utilisée en conjonction avec une souris ou un modem. Ce port a l'avantage d'être compact et de ne posséder que quelques signaux utiles. Plus difficile à mettre en oeuvre que le port parallèle, cependant la prise série se révèle plus puissante et plus universelle.

a/ Brochage et performances :

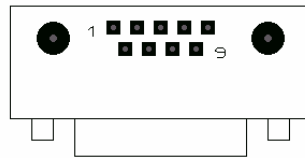
Chaque ordinateur possède deux ports série. Ils prennent la forme de deux connecteurs mâles à 9 broches (pour les PC un peu plus anciens, un des deux possède 25 broches), nommés en standard COM1 et COM2. Le schéma ci-contre montre le brochage.

A l'origine, le port série était destiné à être connecté à un modem, c'est de là que les noms des broches sont tirés. En effet, seulement trois broches sont nécessaires pour communiquer avec un périphérique (TxD, RxD et la masse), les autres broches n'étaient que des signaux de contrôle et d'état. Le port série est très bien protégé contre les courts-circuits (avec une intensité de 20 mA) et les broches sont limitées en courant. Ainsi, il est tout à fait possible d'alimenter un petit montage électronique (par exemple les souris séries sont alimentées par la ligne RTS) mais il est quand même préférable d'avoir recourt à une alimentation externe.

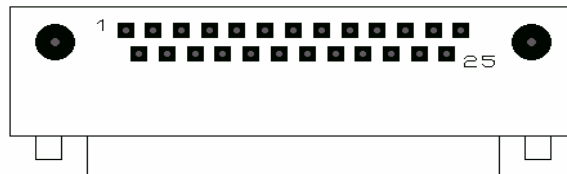
Avec le port parallèle, tous les signaux sont transmis les uns à côté des autres dans un câble qui prend bien souvent l'allure d'une nappe. Ce procédé est rapide mais on remarque que la mise en parallèle des fils très proches induit entre eux un couplage par capacité. Pour réduire

cet effet, il est possible d'intercaler des fils reliés à la masse entre ceux acheminant les signaux.

♦ DB9 mâle (vue de devant) :



♦ DB25 mâle (vue de devant) :



b/ Description et attribution des signaux :

Broche DB9	Broche DB25	Nom
1	8	DCD
2	3	RX
3	2	TX
4	20	DTR
5	7	GND
6	6	DSR
7	4	RTS
8	5	CTS
9	22	RI

Tableau1.1. Attribution des broches

- ◆ **DCD** (Data Carrier Detect) : cette ligne est une entrée active haute. Elle signale à l'ordinateur qu'une liaison a été établie avec un correspondant.
- ◆ **RX** (Receive Data) : cette ligne est une entrée. C'est ici que transitent les informations correspondantes vers l'ordinateur.
- ◆ **TX** (Transmit Data) : cette ligne est une sortie. Les données de l'ordinateur vers le correspondant sont véhiculées par l'intermédiaire de TX.
- ◆ **DTR** (Data Terminal Ready) : cette ligne est une sortie active haute. Elle permet à l'ordinateur de signaler au correspondant que le port série a été libéré et qu'il peut être utilisé s'il le souhaite.
- ◆ **GND** (GrouND) : c'est la masse.
- ◆ **DSR** (Data Set Ready) : cette ligne est une entrée active haute. Elle permet au correspondant de signaler qu'une donnée est prête.
- ◆ **RTS** (Request To Send) : cette ligne est une sortie active haute. Elle indique au correspondant que l'ordinateur veut lui transmettre des données.
- ◆ **CTS** (Clear To Send) : cette ligne est une entrée active haute. Elle indique à l'ordinateur que le correspondant est prêt à recevoir des données.
- ◆ **RI** (Ring Indicator) : cette ligne est une entrée active haute. Elle permet à l'ordinateur de savoir qu'un correspondant veut initier une communication avec lui.

D'un point de vue électronique, les signaux transmis par les TX et RX en sortie des prises répondent aux normes RS232, c'est à dire : « 1 » logique compris entre -3 et -25 V et « 0 » logique compris entre +3 et +25 V.

II.3.3. La norme RS232 :

La transmission série est largement utilisée pour la communication entre les équipements se trouvant éloignés les uns des autres. Le standard RS232 est une interface entre un équipement terminal de données (DTE, Data Terminal Equipment, c'est à dire un ordinateur ou un terminal) et un équipement de communication de données (DCE, Data Communication Equipment, par exemple un modem). Ces deux équipements se transmettent des informations sous forme binaire en série. Comme le montre la définition, le RS232 est tout simplement un « standard ». Celui-ci stipule les règles nécessaires à la transmission d'informations entre

deux machines, terminaux, imprimantes, ordinateurs ou autres équipements utilisant des communications séries

Dans le paragraphe suivant nous allons expliquer l'identifiant « standard » :

Lorsque deux machines, étaient connectés, elles pouvaient ne pas être physiquement compatibles. Les dimensions et la forme des fiches de connexion n'étaient pas identiques. On pourrait dire qu'elles n'étaient pas matériellement compatibles. Il est en effet difficile d'adapter une fiche carrée dans une prise ronde !. Les incompatibilités électriques représentaient également un risque : connecter des signaux électriques incompatibles entre eux est une expérience traumatisante. Ces différents problèmes illustrent bien la nécessité de disposer d'un standard qui spécifie où, quand et comment une information doit être transmise d'une machine à une autre.

III. Présentation des solutions proposées pour notre projet :

L'indication par synthèse vocale du numéro de l'appelant pour personnes non voyantes peut être effectuée par deux solutions :

- Une solution matérielle qui a été élaborée précédemment par un projet de fin d'études.
- Une solution logicielle qui fera l'objet de notre projet.

III.1. Présentation de la solution matérielle :

La solution matérielle consiste à la réalisation d'une carte électronique pilotée par un PC via une interface logicielle permettant à un abonné non voyant d'identifier le numéro de la ligne appelante par voie vocale .

L'objectif de cette solution est de réaliser les étapes suivantes :

- ◆ Etude et conception d'une carte électronique qui assure l'extraction et le décodage de l'information contenant le numéro de l'abonné demandeur.
- ◆ Développement d'une application informatique qui permet l'affichage sur l'écran d'un PC du numéro présenté sur le port série et sa synthèse vocale.

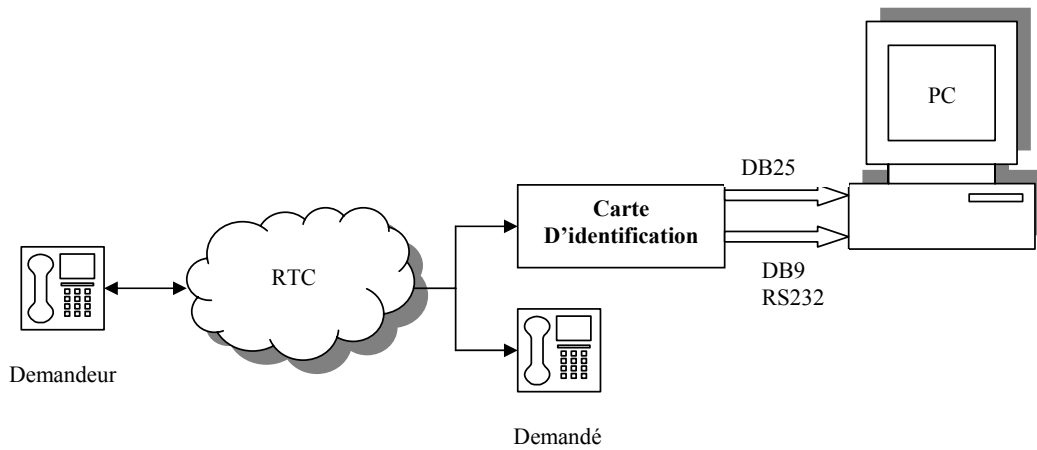


Fig.1.6. Schéma synoptique du projet par solution hard

III.1.1. Synoptique du cahier des charges :

Lors d'un appel, le signal sur la ligne téléphonique du demandé résulte de la superposition de la sonnerie et du signal CLIP. Pour permettre la séparation de deux signaux, il sera utile de les filtrer.

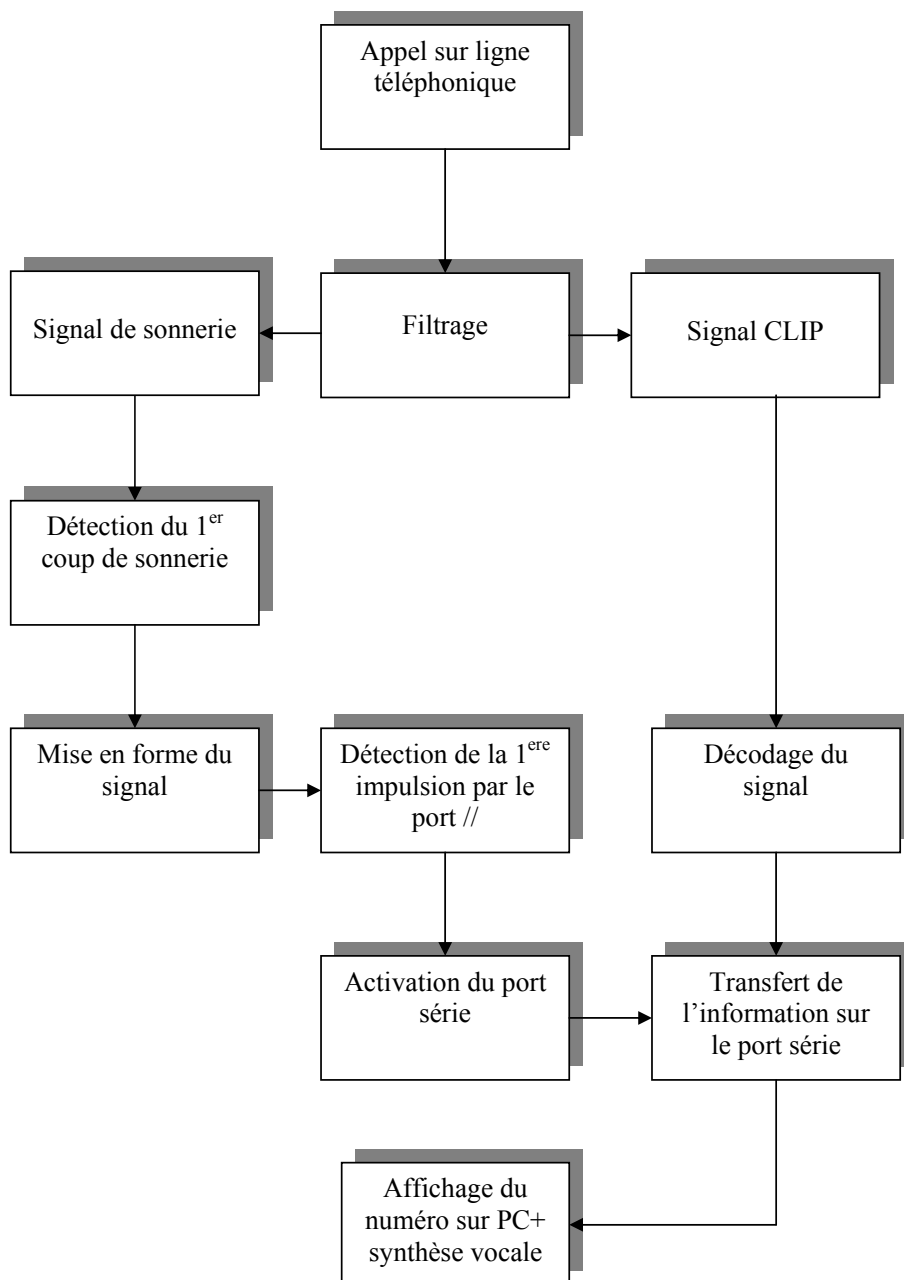


Fig.1.7. Synoptique du cahier des charges

III.1.2. Carte électronique réalisée :

Cette carte englobe la détection de sonnerie, le filtrage, la démodulation et l'adaptation du signal CLIP au PC.

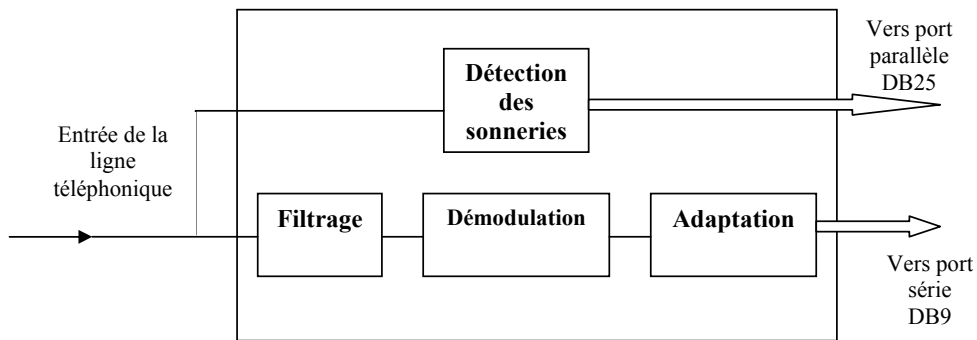


Fig.1.8. Synoptique de la carte réalisée

♦ Le filtrage :

Il faut noter que pour une démodulation efficace, il est utile de filtrer le signal d'entrée pour rejeter le signal de sonnerie et la composante continue de 48 V relevée sur la ligne en état de raccrochage et ne garder que le signal FSK dont la fréquence est supérieure à celle de la sonnerie.

Un filtre passe-haut, composé d'une résistance et d'une capacité, assure cette fonction.

♦ La démodulation :

Pour la restitution de l'information initiale, ils ont utilisé un démodulateur FSK nommé XR2211 qui se présente sous forme d'un circuit intégré à 14 broches .

♦ La détection de sonnerie :

Pour que le PC se rende compte de la présence d'un signal d'identification de l'appelant, il lui faut un signal d'alerte. Ce dernier n'est autre que le signal de sonnerie qui sera transformé

en une suite d'impulsions dont la première servira pour activer le PC et la deuxième pour le désactiver, puisque l'information est présentée dans la première pause de sonnerie.

♦ Adaptation du signal au PC :

Les ordinateurs fonctionnent exclusivement avec des tensions électriques qui représentent des « 1 » ou des « 0 » de l'arithmétique en base 2. Toute information sera donc composée de « 1 » et de « 0 » correspondant à deux niveaux de tension électrique successifs, se produisant à des intervalles de temps égaux.

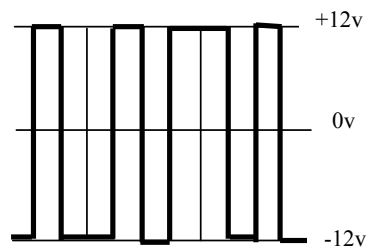


Fig.1.9. Courbe 1

Par contre, le signal à la sortie du démodulateur présente un niveau de tension conforme à la logique TTL : +5 V qui représente un « 1 » logique et 0 V un « 0 » logique.

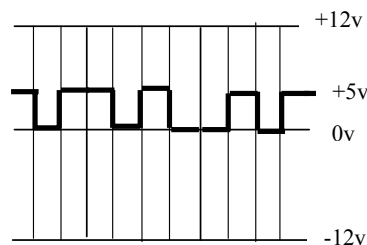


Fig.1.10. Courbe 2

La courbe (2) correspond à la logique TTL et la courbe (1) à la RS232 (Ordinateurs).

Il est remarquable tout de suite qu'il y a deux paramètres essentiels :

- ♦ Les niveaux des tensions sont différents : 0 et 5 V pour la logique TTL, +12 et -12 V pour la logique RS232.
- ♦ La logique est inversée.

Il faut donc adapter le signal à l'entrée RS232. Cette adaptation est assurée dans ce cas par le circuit MAX232.

III.2. Présentation de la solution logicielle :

Les objectifs de cette solution sont déjà définis dans notre cahier des charges. Elle consiste à élaborer une interface logicielle qui permet la programmation d'une carte modem intégrée afin d'assurer l'extraction des signaux de sonnerie et des signaux CLIP, l'affichage du numéro de l'appelant et sa synthèse vocale.

Pour cela, nous nous sommes intéressés de connaître le principe des modems, les commandes **Hayes** à utiliser et la programmation du modem en mode réception afin d'aboutir à la synthèse vocale du numéro de l'appelant.

IV. Les modems :

IV.1. Principe des modems :

Vu l'impossibilité de véhiculer des signaux numériques sur une ligne analogique et à grande distance, on a recours à se servir d'un **MODEM** qui permet d'adapter, à l'émission, les signaux numériques à la transmission sur une ligne analogique et de convertir, à la réception, un signal analogique en un signal numérique. Il est suffisant donc d'associer à chaque niveau logique « 0 » ou « 1 » un signal analogique de fréquence relativement basse. D'où, la simplicité de principe de base d'un modem.

Le modem assure un rôle prépondérant aussi bien en émission qu'en réception de données. La conversion des signaux numériques en signaux analogiques basse fréquence s'effectue dans la partie modulateur du modem en émission des données, alors que la conversion inverse, à savoir des signaux basse fréquence en signaux numériques, a lieu dans la partie démodulateur du modem lors d'une réception des données.

IV.1.1. Modem en émission :

L'adaptation des signaux numériques à la ligne de transmission est assurée essentiellement par :

- ♦ Filtrage : pour éliminer les interférences entre les impulsions successives.
- ♦ Codage : cette opération s'effectue si on utilise des modems en bande de base. Si les modems sont analogiques, le signal numérique sera modulé.

Le filtrage constitue une étape importante après la modulation puisqu'il permet d'éliminer le maximum des parasites tout en ayant un rapport signal sur bruit (caractéristique d'une liaison analogique) plus important. Ce qui facilite la tâche du démodulateur avant le décodage à la réception.

Cependant, on a besoin d'autres opérations telles que le codage qui s'effectue juste à l'entrée du modem et permettant de concentrer l'énergie transmise dans une bande conforme à la bande passante du filtre. Aussi, l'embrouillage est indispensable pour modifier les données à transmettre avant l'étape de la modulation, cette modification consiste à mélanger les données par une série des bits pseudo-aléatoires. Son rôle est donc d'éviter qu'une trop longue succession de 0 ou de 1 n'empêche le récepteur de récupérer le rythme d'horloge dans le train d'informations reçues.

Vu que les signaux analogiques vont circuler sur des grandes distances, on a intérêt à les amplifier et de se servir de signaux de commande et de synchronisation.

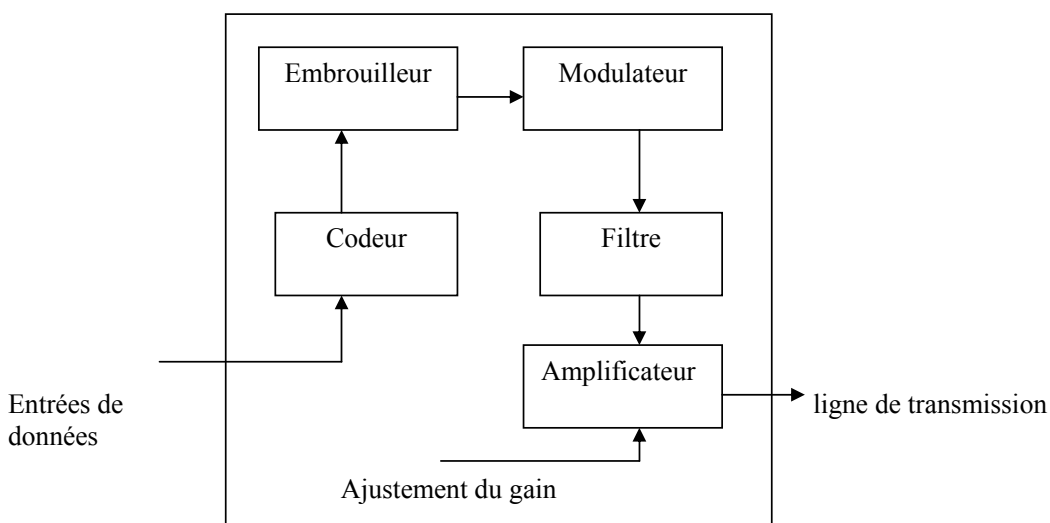


Fig.1.11. Schéma de principe d'un modem en émission

IV.1.2. Modem en réception :

Tout d'abord, le signal doit être amplifié puisqu'il arrive très affaibli au modem de réception. Puis, il sera filtré pour éliminer les parasites amplifiés. Ensuite, la porteuse est extraite pour la détection du signal reçu.

Pour restituer l'information numérique, le modem effectue la démodulation du signal analogique.

Afin de diminuer les interférences dues aux distorsions d'amplitude et de phase, un égaliseur est nécessaire. C'est un équipement qui compense les différences d'affaiblissements ou de retard de transmission subies par les signaux transmis dans la gamme de fréquence du canal occupé sur le réseau téléphonique.

A la sortie de l'égaliseur, le désembrouilleur intervient pour supprimer le brouillage qui a eu lieu à l'émission et retrouver les données initiales.

Enfin, le décodeur restitue la suite des données binaires émises à partir des échantillons provenant de l'égaliseur.

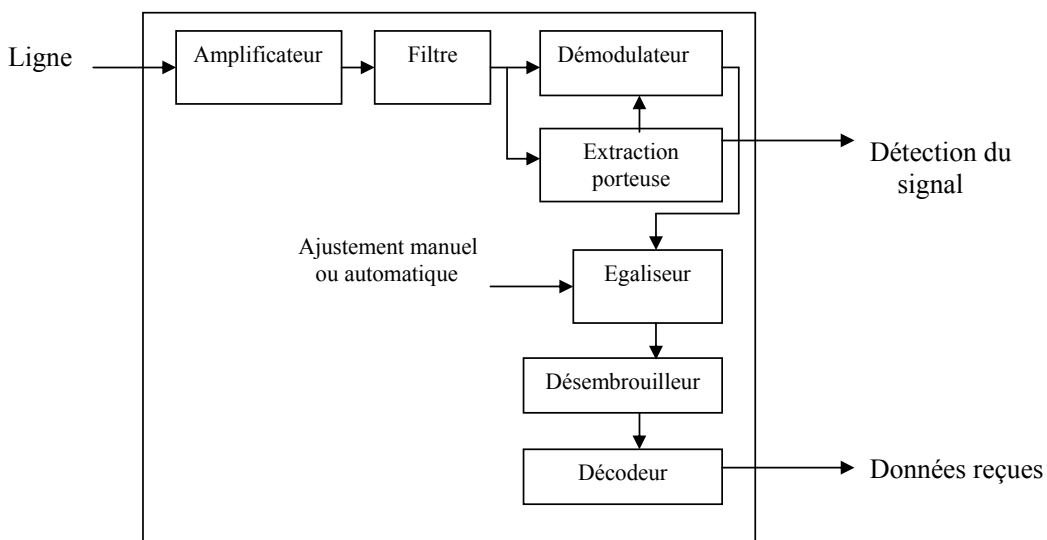


Fig.1.12. Schéma de principe d'un modem en réception

IV.2. Caractéristiques d'un modem :

Un modem (**MO**dulateur **DE**Modulateur) est un convertisseur digital/analogique ou adaptateur digital/digital destiné à convoyer des données sur des lignes habituellement réservées au téléphone.

Il y a deux familles principales de Modems :

- ♦ Les modems pour lignes commutées (lignes domestiques) qui utilisent les mêmes circuits que le téléphone classique – Mode Asynchrone.
- ♦ Les modems pour lignes permanentes dédiées aux transmissions de données point à point entre deux sites reliés par des circuits loués aux opérateurs des télécoms – Mode Synchrone.

Le modem pour les lignes commutées peut être configuré en mode appelant ou en mode appelé.

Mode appelant :

- ♦ Génération des tons ou des impulsions de composition du numéro de téléphone.
- ♦ Adaptation de la vitesse de transmission en fonction des conditions (**Fallback**).
- ♦ Gestion de la réception de la porteuse (Carrier Detect) ou de sa perte.

Mode appelé :

- ♦ Détection de sonnerie (Ring indicator) pour signaler la réception d'un appel.
- ♦ Etablissement de la connexion avec l'appelant, échange des modes de fonctionnement.
- ♦ En cas de dégradation de la qualité de la ligne : **Fallback**.
- ♦ Gestion de la réception de la porteuse (Carrier Detect) ou de sa perte.

Note :

En cas de dégradation de la qualité de la ligne durant la transmission ou à l'initialisation, un des modems partenaire peut demander le **Fallback**, ce qui veut dire que la vitesse de transmission sera descendue par paliers, pour tenter d'améliorer la qualité des messages.

Les normes peuvent être représentées par le tableau suivant :

AVIS CCITT	VITESSE(Bits/s)	MODULATION
V21/Bell 103	300	FSK
V22/Bell 212a	1200	DPSK
V23	1200	DPSK
V22bis	2400	QAM
V32	9600	QAM
V32bis	14.400	QAM
V34	28.800	QAM
V34+	33.600	QAM
V90	56.600	QAM

Tableau.1.2. Normes et vitesses de transmission

IV.3. La modulation et la démodulation :

IV.3.1. La modulation :

La transmission d'un signal numérique sur des longues distances à travers des supports adaptés à la transmission numérique nécessite le recours à la modulation car il y'a risque de dégradation du signal en fonction de la distance parcourue.

La modulation consiste à émettre sur le canal de transmission un signal qui n'a aucune relation avec l'information, mais qui est adapté par le support. Ce choix garantit la réception d'une information avec un taux d'erreurs minimal. Sur ce signal, on va greffer l'information originale qui est l'objet de la transmission.

En transmission de données, le transporteur sera un signal sinusoïdal pur appelé « porteuse » et l'information qui est numérique modulera un ou plusieurs paramètres de la porteuse .

$$s(t)=A.Cos(\omega_0 t+\alpha_0)$$

Il existe trois types de modulation :

- ♦ Modulation par déplacement de fréquence FSK : les niveaux logiques sont représentés par la variation de la fréquence de la porteuse. Cette modulation est utilisée pour la

transmission à faible débit sur le réseau téléphonique commuté. En outre, le principe de cette modulation sera plus détaillé dans notre projet.

- ♦ Modulation par déplacement de phase PSK (Phase Shift Keying) qui associe à un code binaire une valeur de la phase de la porteuse . La vitesse peut être facilement augmentée en utilisant un code binaire sur 2, 3 bits ou plus sans augmenter la fréquence de la porteuse.
- ♦ Modulation par déplacement d'amplitude ASK (Amplitude Shift Keying) qui s'applique en faisant varier l'amplitude du signal en fonction des bits à coder .

La modulation par déplacement de fréquence FSK :

La modulation FSK consiste à émettre un signal sinusoïdal, soit de fréquence f_0 lorsque le bit du signal numérique à transmettre vaut « 0 » logique, soit de fréquence f_1 dans le cas contraire. Afin que la largeur de la bande de fréquence soit aussi faible que possible, il faut que f_0 et f_1 soient plus proches l'une de l'autre. Typiquement, nous retenons que $(f_1 - f_0)/2$ doit être de l'ordre de 10% de $(f_1 + f_0)/2$. D'autre part, afin que le signal de fréquence f_1 ou f_0 soit défini sans ambiguïté, il est nécessaire que la période correspondante à f_1 ou f_0 soit nettement plus petite que la durée d'un état électrique stable du signal numérique initial (voir figure 1.13). Si f_h représente la fréquence du signal numérique initial, il faut prendre donc $f_0 \gg 2f_h$.

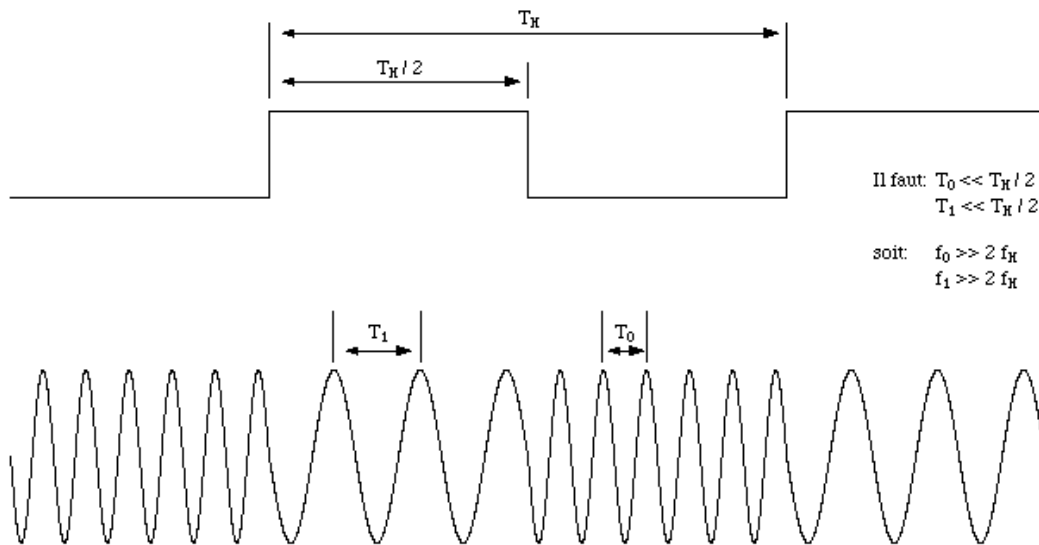


Fig.1.13. Modulation en fréquence du signal

Ainsi, un modulateur FSK peut être schématisé comme suit :

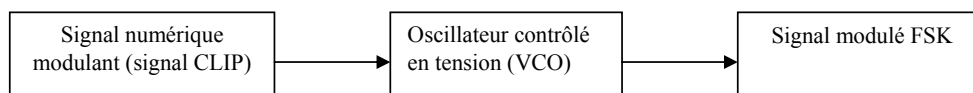


Fig.1.14. Schéma fonctionnel du modulateur

Les oscillateurs contrôlés en tension (VCO) sont des convertisseurs tension – fréquence. Cette variation de tension, lente ou rapide, à leurs entrées se traduit par une variation de fréquence en sortie : on peut aussi utiliser l'expression Vobuler (oscillateur vobulé). Un VCO doit présenter les qualités suivantes :

- bonne linéarité de fréquence en fonction de la tension d'entrée $f_s(u_0)$
- bonne stabilité de fréquence centrale f_0
- grande variation possible de la tension d'entrée u_0

Un VCO est accompagné toujours d'un condensateur ou d'un réseau RC dont la ou les valeurs sont définies pour que la fréquence de sortie f_s soit égale à la fréquence centrale f_0 à tension de commande nulle ($u_0 = 0\text{ V}$).

IV.3.2. La démodulation :

Si l'émetteur module le signal qu'il transmet, le récepteur doit pouvoir effectuer l'opération inverse, c'est à dire la démodulation. Mais, cette opération n'est pas, à priori, évidente. Il s'agit d'employer une boucle à verrouillage de phase (figure 1.15). Le principe de ce montage est de prendre le même circuit pour la démodulation que celui qui a été employé pour la modulation (circuit à base de VCO) puis d'asservir le circuit de démodulation de telle sorte que sa sortie soit égale à l'entrée du circuit de modulation. Nous retrouvons ainsi à l'entrée du circuit récepteur le signal qui était à la sortie du circuit émetteur, c'est à dire le signal modulé.

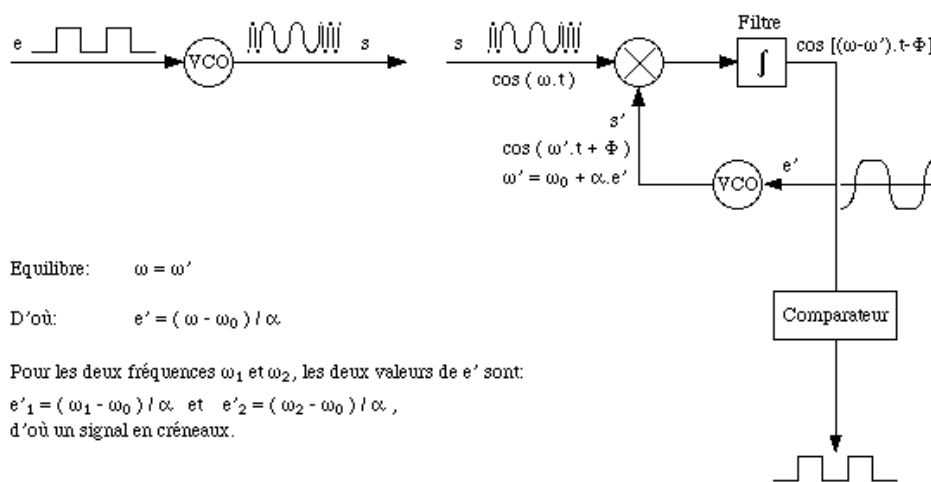


Fig.1.15. Schéma de principe du modulateur / démodulateur

Le montage démodulateur comporte un oscillateur, réglable au moyen d'une inductance variable, donnant la fréquence de référence à partir de laquelle il essaie d'égaliser celle du signal qu'il a démodulé. Ainsi il faut que cette fréquence de référence soit assez proche de f_0 et f_l . En fait, si cette fréquence est correctement réglée, on dit qu'il y a accrochage de la boucle de phase sur le signal transmis.

Par conséquent, la démodulation FSK consiste à l'émission des fréquences en mode verrouillé. Lorsque nous désirons transmettre des signaux binaires « 0 » ou « 1 » par voie téléphonique ou radio téléphonique, nous associons à ces niveaux des signaux sinusoïdaux de fréquences fixées par des standards. La bande passante des lignes téléphoniques étant de 3 KHz (300 à 3400 Hz), ces fréquences sont incluses dans cette marge :

« 1 » pour 1270 Hz	} Lignes téléphoniques
« 0 » pour 1070 Hz	
« 1 » pour 2225 Hz	
« 0 » pour 2025 Hz	

Nous remarquons que le « 0 » binaire correspond à la fréquence basse et le « 1 » binaire correspond à la fréquence haute.

Ainsi, un démodulateur FSK peut être schématisé comme suit :

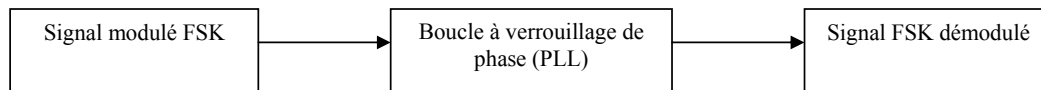


Fig.1.16. Schéma fonctionnel d'un démodulateur

IV.4. Présentation de la boucle à verrouillage de phase PLL :

IV.4.1. Généralités :

La PLL (Phase Locked Loop) ou boucle à verrouillage de phase PLL ou boucle d'asservissement de phase est un système bouclé dans lequel la grandeur asservie est la phase d'un signal alternatif. Son principe a été étudié par Bellescize en 1932. Ce dispositif était destiné à améliorer les conditions de réception des signaux radioélectriques noyés dans le bruit en modulation d'amplitude. Il était aussi utilisé en réception des signaux à porteuses supprimées ou réduites.

Ainsi, le principe fondamental de la PLL était en soi extrêmement intéressant mais la technologie de l'époque ne permettait pas de réaliser des montages simples et économiques. Il a fallu, comme pour beaucoup de principes découverts et mis en attente, l'avènement des circuits intégrés de grande intégration (LSI : Large Scale Integration). Donc, il existe aujourd'hui de nombreux circuits intégrés (PLL). Certains sont à usages multiples, d'autres à usages très spécifiques.

Par conséquent, les PLL peuvent être utilisées dans de nombreuses applications touchant aussi bien l'instrumentation et les mesures que la radio (émission et la réception).

Citons quelques unes de ces applications :

- démodulation cohérente d'amplitude AM
- démodulation de fréquence FM
- détection FSK
- synthétiseur de fréquence
- asservissement de vitesse ...

IV.4.2. Principe de fonctionnement :

♦ Constitution générale :

La figure (1.17) représente le schéma fonctionnel d'une PLL. C'est un système asservi (contre- réaction) à retour unitaire. Entrée et retour sont caractérisés par la phase d'un signal de même fréquence.

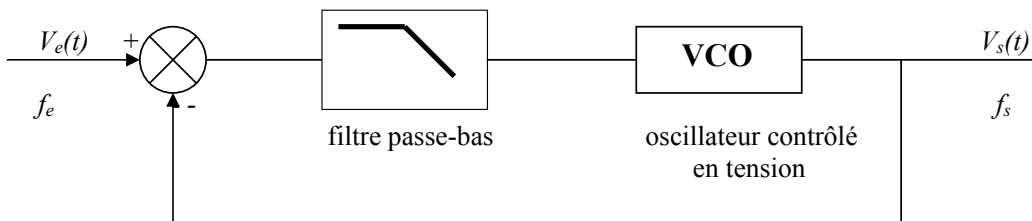


Fig.1.17. La boucle à verrouillage de phase (PLL)

Les éléments constitutifs de ce circuit sont :

- un comparateur de phase,
- un filtre passe-bas,
- un oscillateur contrôlé.

a/ Comparateur de phase :

Ce circuit compare la phase de deux signaux alternatifs et fournit une tension moyenne d'erreurs $u(t)$ proportionnelle à leurs déphasage φ lorsque la boucle est verrouillée, soit $f_e = f_s$.

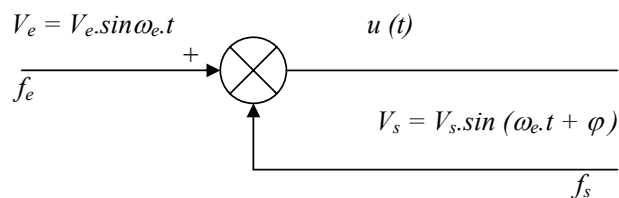


Fig.1.18. Schéma d'un comparateur de phase

La tension $u(t)$ récupérée à partir du comparateur de phase dans les conditions de verrouillage comprend :

- des harmoniques de fréquences $2.f_e$ et plus,
- une composante continue u_0 à l'image de déphasage qui sera directement exploitable

b/ Filtre passe-bas :

La tension $u(t)$ est inutilisable à cause des harmoniques. Il faut supprimer ceux-ci afin de ne conserver que la composante continue $u_0(t)$. La fonction de transfert du filtre influence les propriétés de l'asservissement et permet, par le choix des paramètres introduits, de modifier les performances du dispositif. C'est l'utilisateur qui fixe la ou les fréquences de cassure de ce filtre. On rencontre :

- le filtre passif constitué de résistances et de condensateurs, c'est le cas le plus fréquent lorsque on utilise des circuits intégrés PLL.

- le filtre actif qui permet, en plus de sa fonction initiale, d'apporter un gain supplémentaire dans la chaîne directe .

c/ Oscillateur contrôlé :

C'est un oscillateur fournissant une onde alternative, dont la fréquence varie à l'image d'une tension ou d'un courant. Cet oscillateur est appelé oscillateur vobulé.

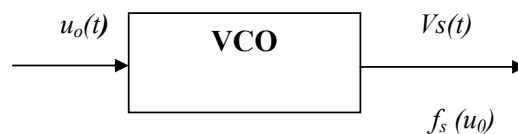


Fig.1.19. Schéma global du VCO

Remarque :

Les circuits PLL peuvent être alimentés symétriquement (+V, -V) ou dissymétriquement (+V, 0). Dans tous les cas, la fréquence centrale est définie pour u_0 située au milieu de la tension totale d'alimentation. Le VCO seul est peu stable en fréquence, il ne peut être utilisé qu'en boucle fermée. Sa sensibilité K, pouvant être exprimée en Hz / V, ne doit pas être trop élevée afin de préserver la stabilité de la boucle.

V. Les commandes Hayes :

V.1. Historique :

Pendant les 1970s, une petite compagnie des communications appelée Bizcom a développé le concept d'ajouter l'intelligence aux modems en utilisant des codes -un ensemble des commandes- que le modem reconnaît et à base desquelles il va agir.

En plus du travail novateur de Bizcom, les autres fabricants du modem ont développé des modems de la même façon. Malheureusement, les ensembles des commandes construites dans les modems fabriqués par les vendeurs différents étaient habituellement incompatibles les

unes avec les autres pendant les premiers jours d'apparition des modems intelligents. Donc, beaucoup de produits du logiciel des communications, originalement, ont opéré avec un modem spécifique ou quelques modems, déterminés par leurs ensembles des commandes. Ainsi, les modèles de la Hayes Micro-ordinateur, Produits série de Smartmodems, ont capturé sur 50% du marché des modems utilisés avec des ordinateurs personnels. L'ensemble des commandes de cette entreprise a émergé comme un niveau de l'industrie standard. Pendant les 1980s beaucoup de fabricants du modem ont incorporé un ensemble de commandes semblables mis dans leurs produits, en résultant en compatibilité entre logiciel des communications qui supporte l'ensemble de commandes utilisé par Hayes Smartmodems et les modems fabriqués par les autres vendeurs.

Aujourd'hui, beaucoup de fabricants du modem font de la publicité pour leurs produits comme « Hayes-compatible ».

V.2. L'ensemble des commandes Hayes:

L'ensemble des commandes Hayes consiste en un ensemble de commandes de base et des commandes d'extensions. Les commandes de base, tel que placer le modem connecté (l'action qui se produit quand le combiné du téléphone est soulevé) et composer un numéro, sont communes à tous les modems Hayes-compatibles. Les commandes d'extensions, tel que placer un modem dans une vitesse du fonctionnement spécifique ou commençant certains traits - y compris compression des données ou un type spécifique de découverte de l'erreur et de correction - sont applicables aux modems seulement construits pour transmettre et recevoir les données à cette vitesse ou modems qui supportent le trait demandé.

Dans notre cas, nous nous intéressons à l'utilisation des commandes Hayes de base pour la programmation de notre modem intégré.

Le format de base exigé pour transmettre des commandes à un modem intelligent de type Hayes-compatible est comme suit, avec des paramètres facultatifs indiqués entre crochets :

AT Nom de la commande [Paramètre (s)] Nom de la commande[Paramètre] . . . Retour

V.3. Liste des commandes Hayes:

Les commandes AT du mode Hayes permettent, à partir du terminal, de configurer le modem, de mémoriser un numéro téléphonique et de recevoir ou de transmettre des appels sur le réseau commuté. Il existe plusieurs types des commandes Hayes qui sont :

♦ **Commandes d'exploitation et de numérotation:**

AT	Indication de début d'une commande. Elle précède toutes les commandes sauf A/.
O	On line / retour en transmission de données c-à-d retour au mode transfert de données (inverse de la commande d'échappement).
Hn	Hook Switch Control / connexion- déconnexion directe: 0 : déconnexion 1 : connexion
A	Answer / passage en mode réponse avec émission du 2100Hz en exploitation réseau commuté.
D	Dial number / connexion avec numérotation. C'est une commande de numérotation et seuls les caractères suivants sont pris en compte après D: 0 à 9 - chiffres du numéro T - numérotation multifréquence P - numérotation décimale R - appel en mode réponse (émission voie haute, Réception Voie basse). R doit être placé en fin de chaîne. W - attente de tonalité ; - retour en mode commande après la numérotation U - répétition automatique d'un appel qui n'a pas abouti. Sans paramètres, la commande D provoque la connexion sans numérotation sur le réseau commuté (RC) et le passage en transmission de données sur la ligne en mode appelant.
DSn	Dial with stored number / appel avec numéro en EEPROM. La DSn est une commande de numérotation automatique du numéro

mémo-

risé dans le répertoire téléphonique à l'adresse n ($0 \leq n \leq 4$).

Commentaire :

♦ **Commandes diverses:**

A/	Repeat last command / demande de répétition de la dernière commande.
-----------	--

+++	Escape command / commande d'échappement : il y'a passage du mode transfert des données au mode commande.
B	Select the method of modem modulation / sélection de la méthode de modulation du modem.

♦ **Commandes de test:**

&T	Test mode / commandes de test
In	Product identification / numéro d'identification

V.4. Règles syntaxiques d'écriture d'une commande Hayes :

♦ **1^{ère} règle :**

Toute chaîne de commande doit commencer en transmettant un code d'attention au modem suivi par la commande appropriée ou l'ensemble des commandes que l'utilisateur veut que son modem le rende effectif, à l'exception de la commande **A/**. Le code d'attention est la séquence du caractère **AT**. Cette exigence de préfixer toutes les lignes de commandes avec ce code AT a mené beaucoup de fabricants de modems à dénoter leurs modems comme " Hayes AT- Compatible ".

♦ **2^{ème} règle :**

Plusieurs commandes peuvent être chaînées pour être traitées séquentiellement.

♦ **3^{ème} règle :**

Les codes des commandes peuvent être entrés au clavier indifféremment en majuscules ou en minuscules.

♦ **4^{ème} règle :**

Une ligne de commande ne doit pas dépasser 255 caractères. En cas de débordement, le modem envoie l'indication « ERROR » et la ligne est ignorée.

♦ **5^{ème} règle :**

Une commande ou une chaîne de commandes est prise en compte après un retour chariot et le modem répond « OK ».

♦ 6^{ème} règle :

En cas d'erreur de syntaxe, la correction est possible en utilisant la touche DEL (H7F) ou BS (CTRLH=H08).

♦ 7^{ème} règle :

Il est possible d'insérer des tirets et des espaces pour améliorer la présentation d'une chaîne de commandes. Ces caractères ne sont pas pris en compte par le modem.

♦ 8^{ème} règle :

En cours de numérotation, le modem affiche l'ensemble des digits du numéro qu'il est en train de composer. L'appui sur une touche quelconque du terminal provoque l'arrêt de la numérotation et l'envoi du message « ABORTED ».

V.5 . Programmation des modems, le code AT :

Les séquences de programmation **AT** pour *Attention Code* inventées par les constructeurs de modems Hayes sont devenus un standard de facto.

Ce standard définit un certain nombre de commandes (citées précédemment) qui permettent de dialoguer avec le modem, qui commencent toutes par les deux lettres AT.

Exemple : **ATDT 7773456** signifie **Dial Tone No** de téléphone : **7773456**.

A chaque requête acceptée, le modem répond par **OK** ou par **ERROR** en cas de fonction erronée.

Considérons un ordinateur DTE qui permet à son port série connecté au modem DCE et qui est configuré à 57600 Bits/s, 8 bits, sans parité. Les deux lettres **AT** sont envoyées dans ce format.

A la réception, le modem qui lui est configuré (par exemple) 9600Bits/s, 7 bits, avec parité, comparera dans une table les deux caractères erronés reçus, ce qui lui permet d'ajuster ses propres paramètres à celui de l'ordinateur(DTE).

Il répondra par **OK** pour faire savoir qu'il travaille selon les mêmes paramètres.

Hayes a défini un certain nombre de **codes résultants** de base aux commandes AT. Ces codes sont mentionnés dans le tableau suivant :

Code	Message	Signification
0	OK	La ligne de commande est exécutée sans erreur
1	CONNECT	La connexion a été établie
2	RING	Détection d'un signal de sonnerie
3	NO CARRIER	La connexion n'a pu s'établir ou perte de porteuse
4	ERROR	Erreur dans la ligne de commande
5	CONNECT 1200	Connexion établie à 1200 bits/s (V.22)
6	NO DIAL TONE	La tonalité d'invitation à numéroté n'a pas été détectée.
7	BUSY	Détection de la tonalité d'occupation
8	NO ANSWER	La tonalité de réponse n'a pas été détectée
15	DIALING xxxx	Le modem compose le numéro xxxx
16	BLACKLIST FUL	La mémoire de stockage des numéros retardés ou interdits est pleine.
17	FORBIDDEN CALL	Numéro brûlé (2 appels erronés)
18	DELAYED CALL mn	Numéro retardé de minutes
19	ABORTED	Appel abandonné (appui sur une touche pendant la numérotation).

Tableau.1.3. Messages Hayes

Note : Les messages de base sont en caractère gras.

VI. La synthèse vocale :

La synthèse vocale est une technique qui permet de faire prononcer de façon agréable un texte écrit par exemple, autrement dit la création d'une voix artificielle semblable à celle de l'homme.

Les synthétiseurs vocaux peuvent être utilisés dans le domaine de l'industrie (pour faire parler une machine), en informatique dans le secteur de multimédia, dans les serveurs vocaux et bien sûr dans le domaine du handicap.

Ils permettent à une personne aveugle d'accéder aux données (lecture, bloc-notes, service de messagerie ...).

Pour créer une voix artificielle, il faut convertir l'information source en mots. Ces derniers doivent être représentés en code phonologique puis convertis en sons véritables.

Comment produire les sons ?

Certains chercheurs du XVIII^e siècle ont essayé de reproduire les formes de l'appareil vocal en fabriquant des machines qui produisaient des sons de quelques voyelles et quelques consonnes.

D'autres chercheurs, comme Alexandre Graham Bell, ont fabriqué des machines encore plus perfectionnées comme les Gamophones basées sur les techniques d'enregistrement sur disque qui codent l'onde sonore toute entière.

Vers les années 30 paraissait la première méthode électrique (le vocoder) qui permettait de produire quelque chose vaguement ressemblante à la parole humaine, mais la manipulation était très complexe.

Au début des années 50 une deuxième méthode permettait la restitution des sons à partir de l'onde sonore.

Ces deux méthodes sont exploitées par les programmes de synthèse vocal.

Actuellement, les ordinateurs sont utilisés pour mettre en mémoire des échantillons d'ondes numérisés. La procédure consiste à convertir les ondes en trains d'impulsions qui n'occupent pas beaucoup de place dans la mémoire de l'ordinateur. Cette technique ne tient compte ni de l'intonation ni du ton de la voix.

Conclusion :

Ce chapitre nous a donné une idée générale sur le service CLIP (sa procédure d'émission, son mode de transmission ...), le principe de fonctionnement du modem et quelques exemples de programmation des modems.

A rectangular area with a marbled paper pattern in shades of grey, white, and light brown. The text is centered within this area.

Chapitre II

Etude de conception

Chapitre II

Etude de conception

Introduction :

Après avoir étudié théoriquement le service CLIP et les modems, nous nous intéressons dans ce chapitre aux sous programmes nécessaires pour la récupération de l'information numérique et sa synthèse vocale.

Ces sous programmes sont schématisés par des organigrammes assurant l'extraction des signaux de sonnerie et des signaux CLIP, l'affichage du numéro de demandeur et sa synthèse vocale.

I. Extraction des signaux de sonnerie :

Pour que le modem se rende compte de la présence d'un signal d'identification de l'appelant, il lui faut un signal d'alerte. Ce dernier n'est autre que la sonnerie.

Le sous programme d'extraction des signaux de sonnerie est tout aussi simple et peut être symbolisé par l'organigramme de la figure (2.1).

Description :

Le modem est en mode veille (en attente). A chaque instant il teste s'il y a un appel (détection de la première sonnerie) :

- ♦ Si oui, il détecte la présence d'un appel.
- ♦ Si non, il revient à son état de veille.

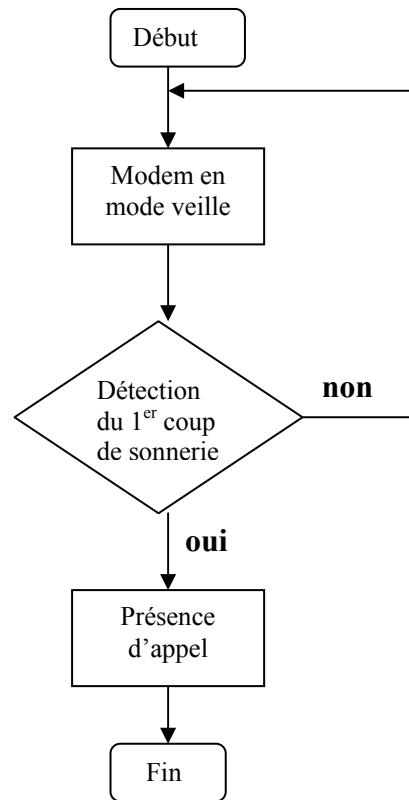


Fig.2.1. Organigramme d'extraction des signaux de sonnerie

II. Extraction des signaux CLIP :

Le sous programme d'extraction des signaux CLIP est symbolisé par l'organigramme de la figure (2.2).

Description :

Après la détection de la présence d'un appel, le modem démodule le signal CLIP pour récupérer l'information numérique. Ensuite, la deuxième opération consiste à la procédure de lecture.

*** La procédure de lecture :**

Elle est représentée par l'organigramme de la figure (2.3). Chaque caractère lu sera stocké en mémoire. Pour que le modem soit informé de la fin des caractères, un test est nécessaire :

- ♦ Si ce test est valide (oui), alors c'est la fin de la procédure de lecture et le modem détecte le deuxième coup de sonnerie.
- ♦ S'il est invalide (non), le modem revient à lire le caractère suivant.

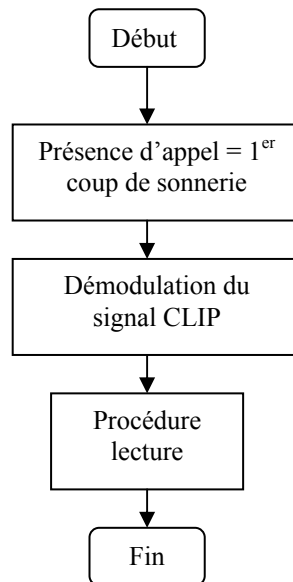


Fig. 2.2. Organigramme d'extraction des signaux CLIP

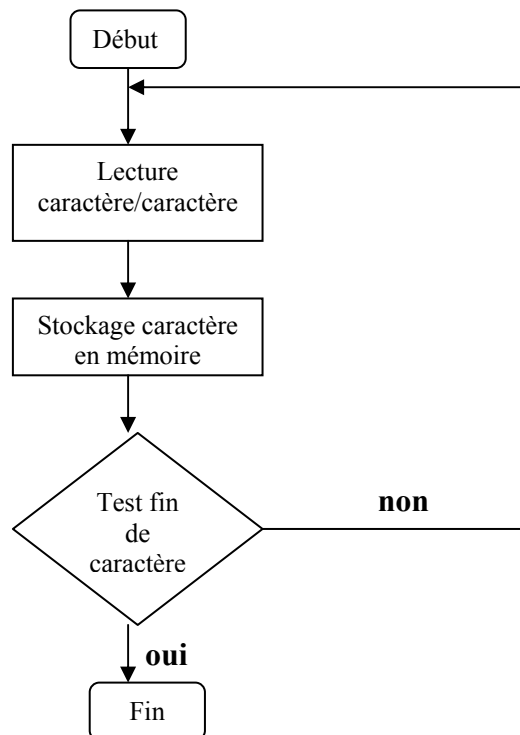


Fig.2.3. Organigramme de la procédure de lecture

III. Affichage du numéro de l'appelant:

Le sous programme est symbolisé par l'organigramme de la figure(2.4).

Description:

L'opération consiste à tester le contenu de la mémoire (présence ou absence d'un numéro).

- ♦ Si ce test est valide, alors affichage du numéro de l'appelant en question sur PC.
- ♦ S'il est invalide, alors le modem teste de nouveau le contenu de la mémoire.

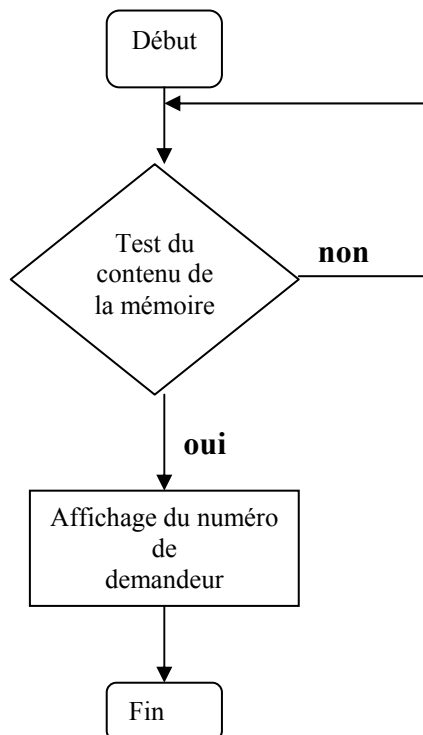


Fig.2.4. Organigramme d'affichage du numéro de demandeur

IV. Synthèse vocale:

Le sous programme de la synthèse vocale est symbolisé par l'organigramme de la figure (2.5).

Description:

Pour réaliser la synthèse vocale, le modem effectue un double test:

- ♦ 1^{er} test (test du deuxième coup de sonnerie) :

Si ce test est valide, le modem attend l'entrée d'une commande externe programmée à l'avance par l'utilisateur.

Si le test est invalide, le modem reste en attente du deuxième coup de sonnerie.

♦ 2^{ème} test (entrée de commande externe) :

Si ce test est valide, le modem passe à la procédure de synthèse vocale.

Si non, attente de la commande.

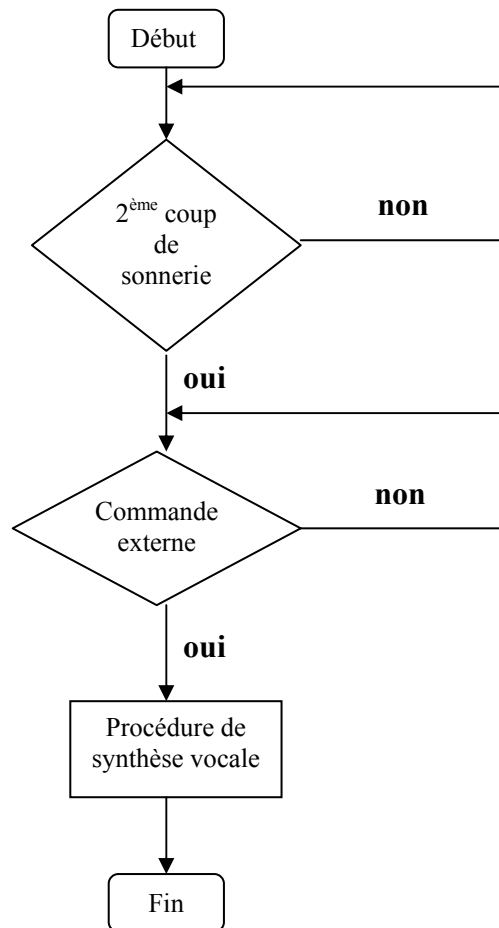


Fig.2.5. Organigramme de la synthèse vocale

*** Procédure de la synthèse vocale:**

Il s'agit dans notre application de prononcer des textes convertis en fichiers '.Wav' (format couramment employé pour les fichiers de type son) séquentiellement en fonction du numéro du demandeur affiché. Pour cela, il faudrait déjà enregistrer des morceaux sonores traduisant vocalement des textes qui seront utilisés par la synthèse vocale.

Ces textes sont :

- ' Le numéro de l'appelant est : '
- 'zéro', 'un', 'deux', 'neuf'.

En plus du langage français, nous pouvons accéder à la synthèse vocale du numéro par les deux autres langues arabe et anglais sélectionnés à l'avance.

La procédure de la synthèse vocale peut être symbolisée par l'organigramme de la figure (2.6). Pour prononcer le numéro de l'appelant, le modem a recours aux fichiers son : "f.wav" (cas du langage français) ou "a.wav" (cas du langage arabe) ou "e.wav" (cas du langage anglais). Le modem effectue un test cyclique sur le chiffre à prononcer. Si le chiffre testé est valide, il passe à son fichier approprié pour le prononcer. Si non, le modem passe pour tester le chiffre suivant.

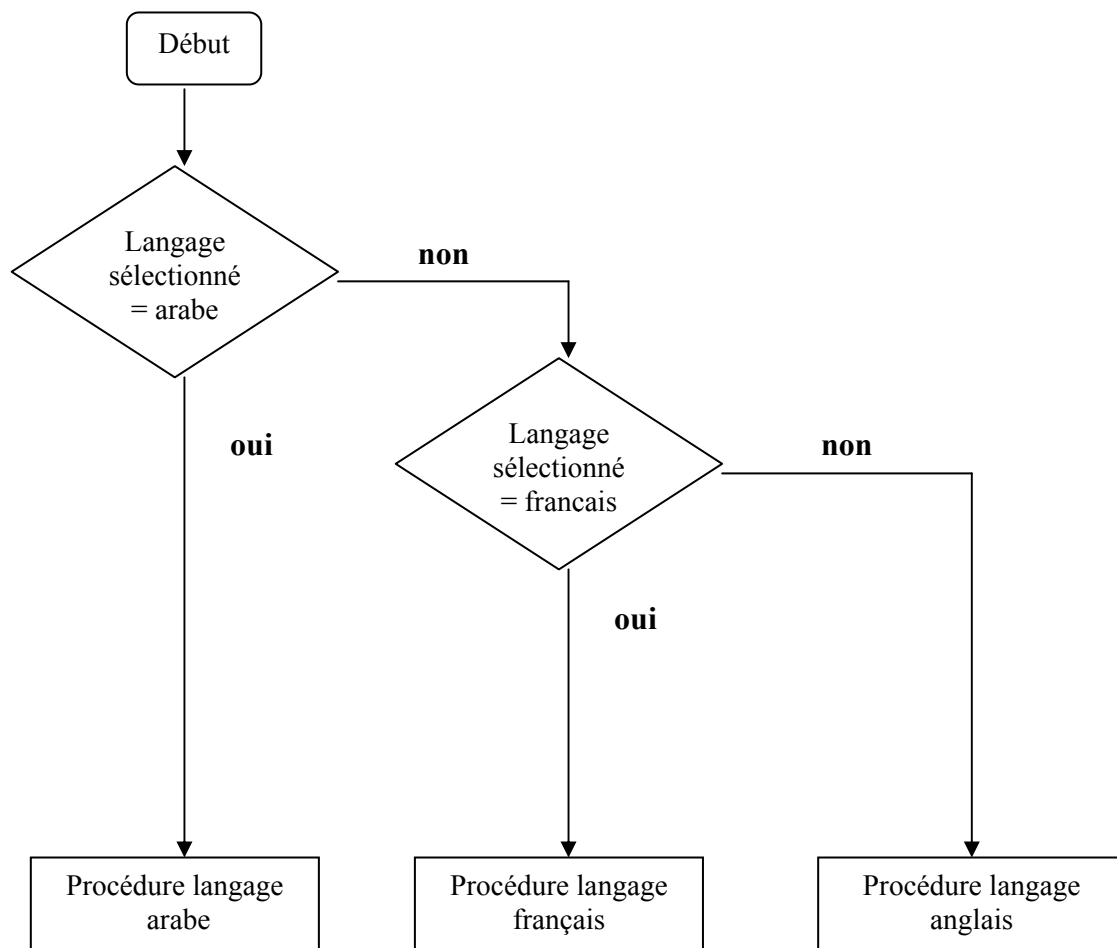
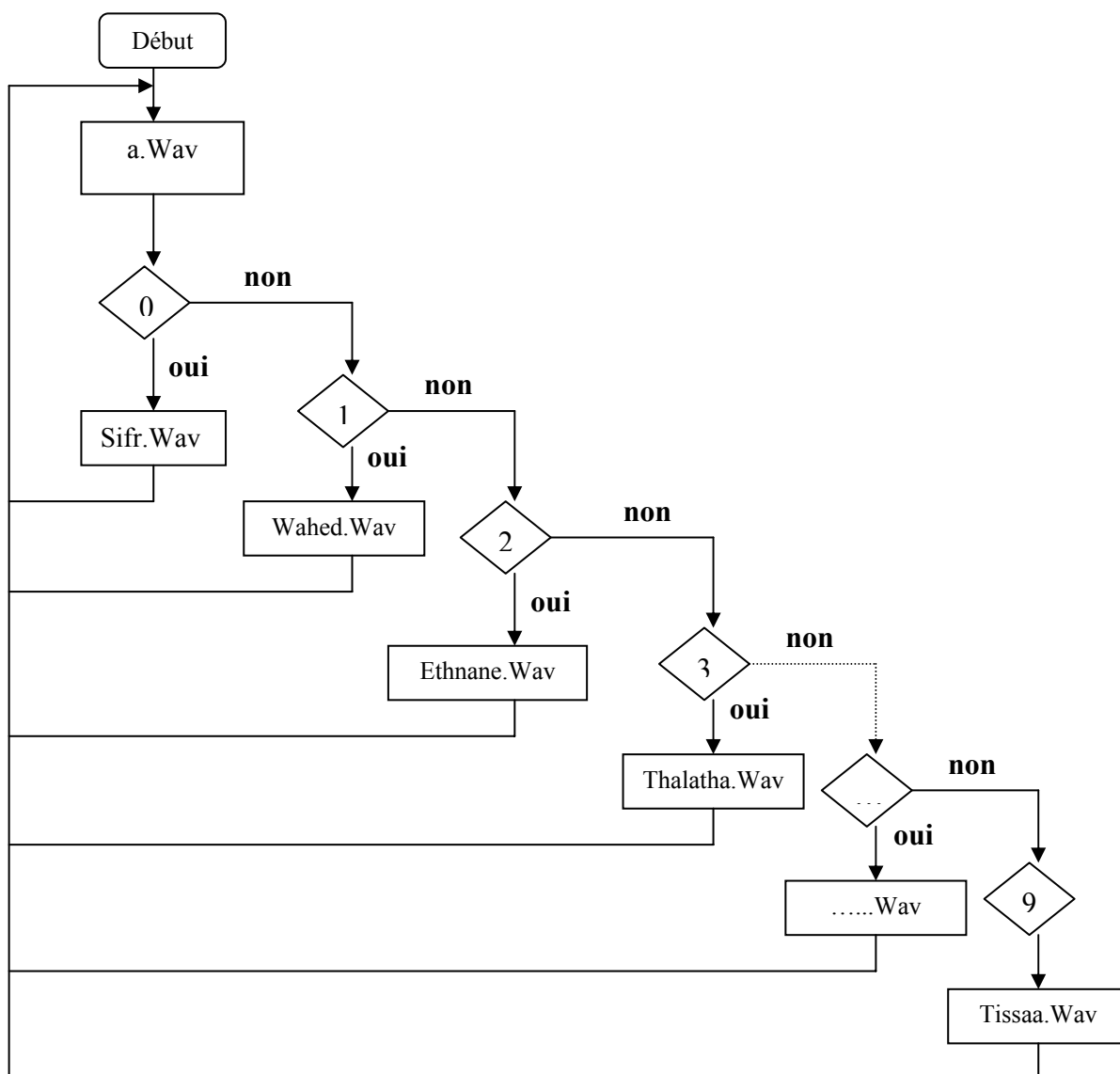
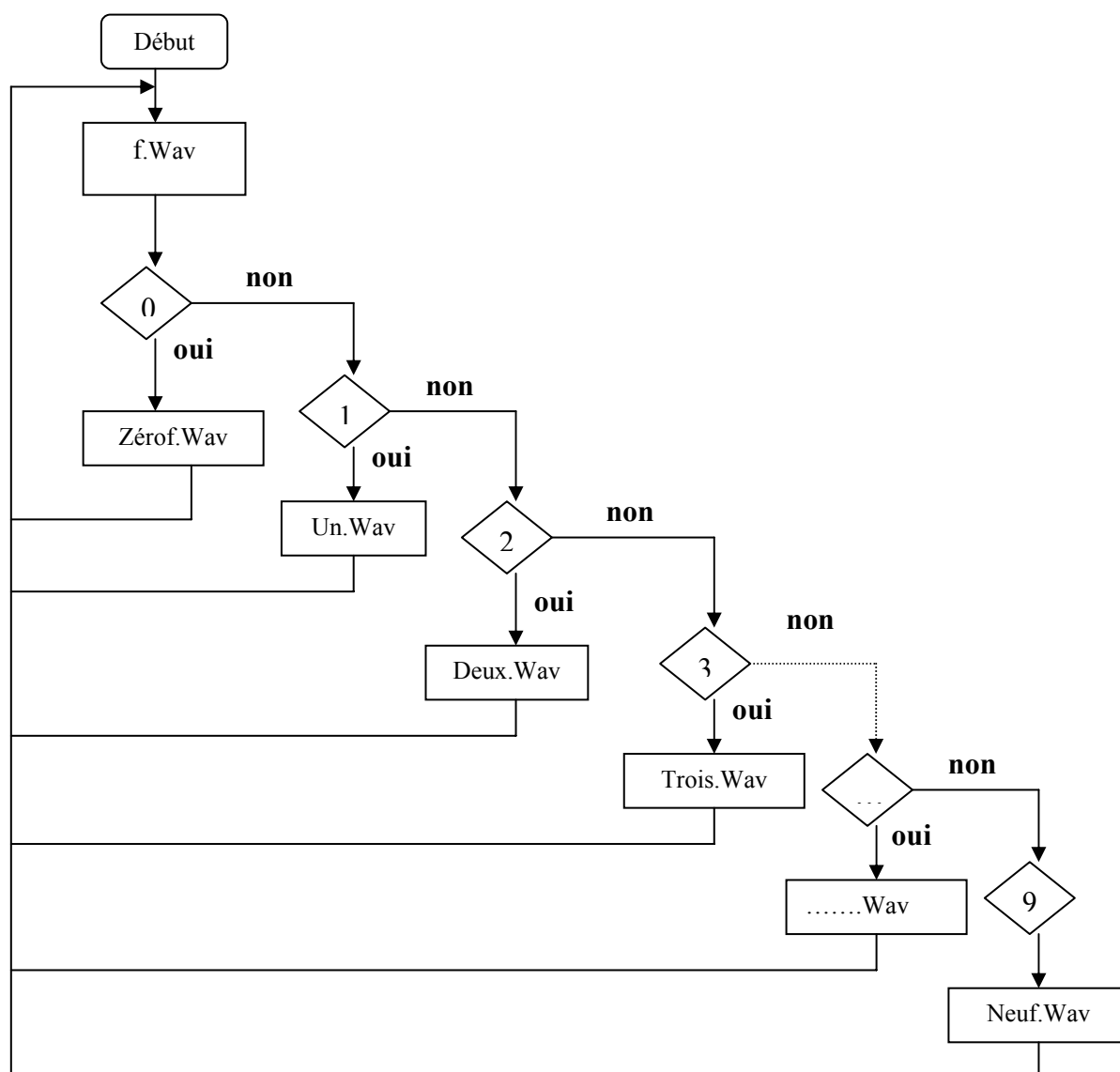
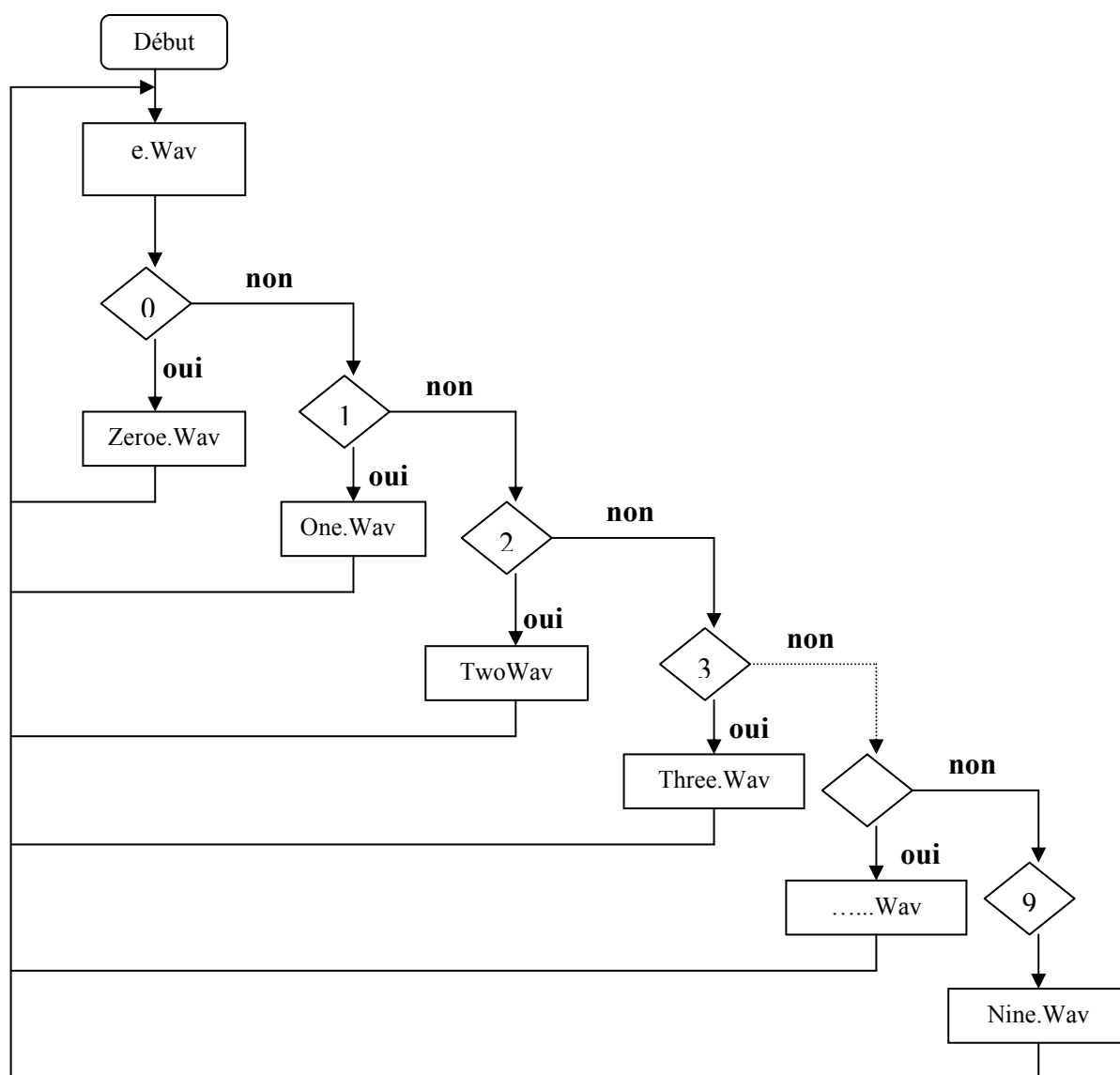


Fig.2.6. Organigramme de la procédure de synthèse vocale

*** Procédure du langage arabe :**

*** Procédure du langage français :**

*** Procédure du langage anglais :**



Conclusion:

Ce chapitre nous a permis de présenter les différents sous programmes nécessaires pour le traitement du message CLIP par le modem et la récupération de l'information à la réception.

A rectangular area with a marbled paper pattern in shades of grey, white, and light brown. The text is centered within this area.

Chapitre III

Etude de réalisation

Chapitre III

Etude de réalisation

Introduction :

Après avoir symboliser les divers sous- programmes par des organigrammes, nous devons concrétiser le bon fonctionnement de notre organigramme globale (voir figure 3.1).

Cela nécessite le développement d'un programme sous *Visual Basic* mettant en œuvre le numéro de l'appelant et sa synthèse vocale. Ce programme est composé par des fenêtres résultantes des différentes procédures utilisées.

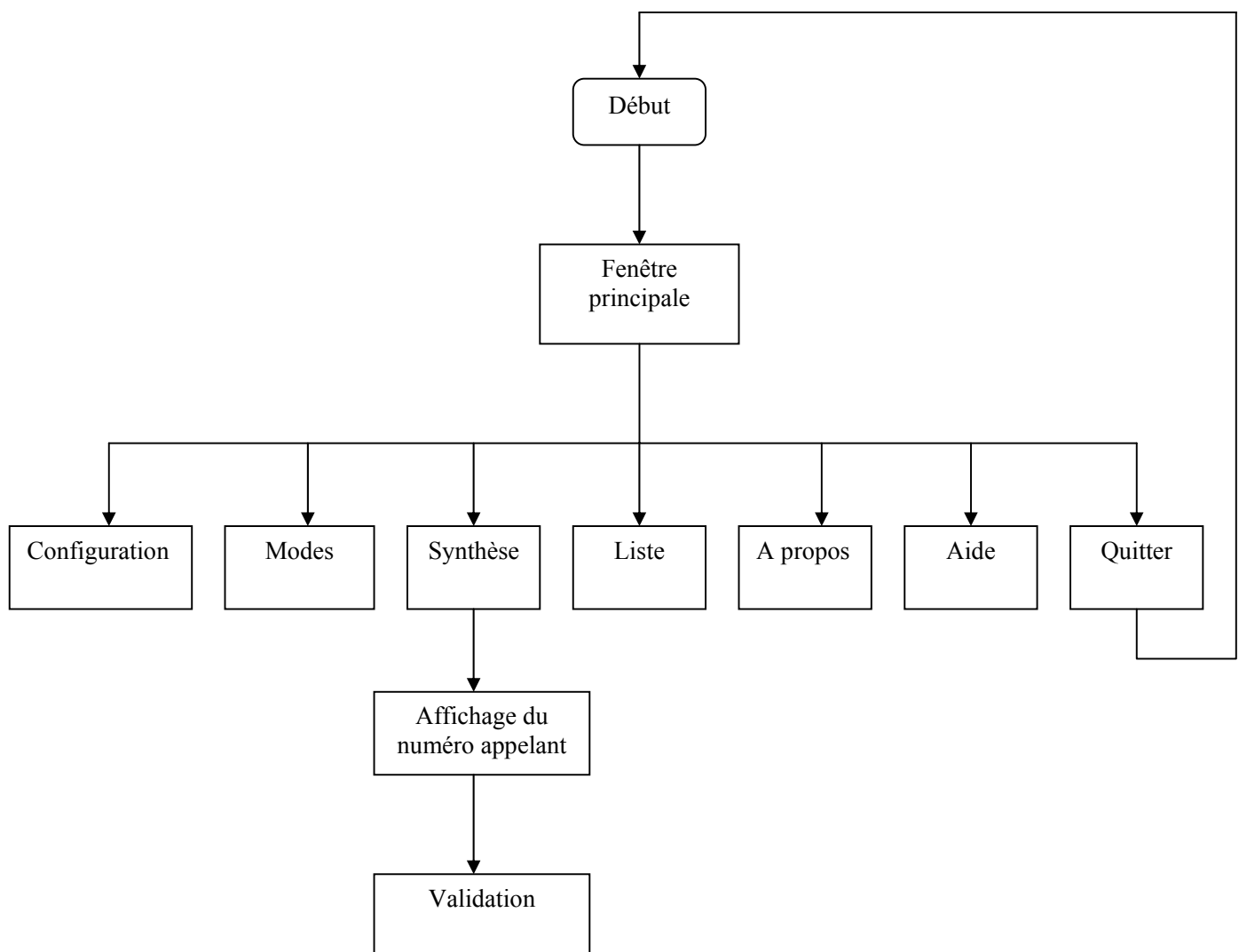


Fig.3.1. Organigramme de programmation de l'interface

I. Présentation du logiciel de développement :

I.1. Critères de choix :

De nombreux logiciels de développement orientés objets (Visual C++ , Delphi, Visual Basic....) sont disponibles. Le choix dépend de notre application et de la souplesse qu'offre l'outil.

Pour cette raison, nous choisissons Visual Basic comme langage de programmation vue la simplicité de sa maîtrise.

I.2. Présentation du Visual Basic :

Visual Basic représente aujourd'hui un environnement de programmation moderne qui permet de développer des applications, créer et manipuler tous les objets de Windows.

Par défaut, cet outil de programmation affiche une feuille vierge appelée FORM1. Cette dernière servira à l'élaboration d'une future application. Elle possède toutes les caractéristiques d'une fenêtre Windows 95 (barre de titre, icônes, etc). il suffit d'y placer les contrôles nécessaires.

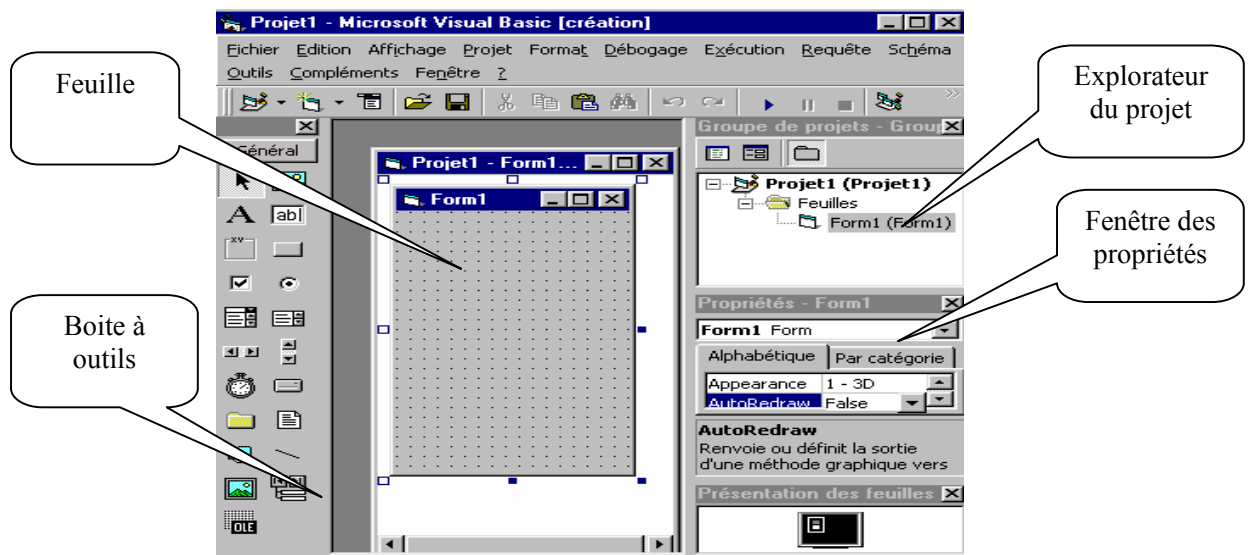


Fig.3.2. L'environnement de développement du Visual Basic

Les divers boutons de la boîte à outils représentent les contrôles qu'il faut utiliser pour élaborer l'interface graphique d'une application. Les contrôles ainsi que la feuille permettent à l'utilisateur de gérer l'application. Le terme « contrôle » désigne tous les éléments (objets) que la boîte à outils permet de placer sur une feuille afin de définir la présentation et les fonctionnalités de l'application. La boîte à outils propose un grand nombre de contrôles standard Windows qui sont déjà rencontrés dans d'autres applications. Ces contrôles vont aider à établir la structure de base d'une future application.

La fenêtre des propriétés affiche dans la colonne de gauche toutes les propriétés de contrôle sélectionné, classées par ordre alphabétique. La colonne de droite présente les valeurs de ces propriétés. Dans la partie inférieure de la fenêtre, Visual Basic affiche une petite aide sur la propriété sélectionnée.

Les propriétés des objets doivent être toujours visible au programmeur afin de l'utiliser pour la modification éventuelle de son application.

I.3. Description des objets :

Pour le cas de notre interface, l'implémentation du programme est élaboré sous VB. Cela nous mène à l'utilisation de plusieurs objets (contrôles) tels que MSComm, SStab.....

I.3.1. Le contrôle MSComm :

Le composant MSComm (MicroSoft comm Control) offre à notre application des possibilités de communication série en l'autorisant à transmettre et à recevoir des données par le biais d'un port série. Ce contrôle est fourni en standard dans la suite Visual Basic. Cet objet vous permet d'accéder aux ports COM de façon très intuitive et très complète. Il existe beaucoup d'options et d'événements associés.

Bien que MSComm possède de nombreuses propriétés, nous proposons de définir les plus importantes :

- ❖ **CommPort** : c'est un entier qui permet de définir et de renvoyer le numéro du port de communication.

- ❖ **Settings** : Cette propriété définit et renvoie les paramètres de vitesse en bauds (BBBB), de parité(P), de nombre de bits de données(D) et de nombre de bits d'arrêt (S). Elle est constituée donc de quatre valeurs et se présente sous le format suivant : "BBBB,P,D,S".
- ❖ **CommEvent** : Cette propriété renvoie l'erreur ou l'événement de communication le plus récent. Elle n'est pas disponible au moment de la création et est en lecture seule au moment de l'exécution.
- ❖ **PortOpen** : C'est une expression booléenne qui définit et renvoie l'état du port de communication (ouvert ou fermé). Cette propriété n'est pas disponible au moment de la création.
- ❖ **InputLen** : définit et renvoie le nombre de caractères lus dans le tampon de réception par la propriété Input.
- ❖ **Input** : permet de renvoyer et de supprimer les caractères provenant du tampon de réception.
- ❖ **Output** : c'est une propriété permettant d'écrire une chaîne de caractères dans le tampon de transmission.

I.3.2. Le contrôle Onglet :

Le contrôle Onglet (nommé SStab) est un outil pratique qui permet de présenter plusieurs boîtes de dialogue ou plusieurs écrans d'informations sur une même feuille en utilisant la même interface utilisée par la plupart des applications commerciales Microsoft Windows.

Ce contrôle fournit une série d'onglets, faisant chacun office de conteneur pour les autres contrôles. Il n'est possible d'activer qu'un seul onglet à la fois, il affiche alors les contrôles qu'il contient, tandis que les autres contrôles restent cachés dans les autres onglets.

Pour utiliser ce contrôle, il faut au préalable déterminer comment agencer les contrôles placés dans les divers onglets. Lors de la création, il suffit de cliquer sur chaque onglet pour le sélectionner, puis dessiner pour chaque onglet les contrôles à afficher.

Au cours de l'exécution, les utilisateurs peuvent se déplacer d'un onglet à l'autre en appuyant sur les touches CTRL+TAB ou en utilisant les mnémoniques définis dans la légende de chaque onglet.

I.3.3. Le contrôle TextBox :

Un contrôle TextBox (zone de texte), parfois appelé " champ d'édition " ou " contrôle d'édition ", affiche des informations entrées au moment de la création, tapées par l'utilisateur ou affectées au contrôle au moment de l'exécution.

I.3.4. Le contrôle ComboBox :

Un contrôle ComboBox (liste modifiable) réunit les fonctionnalités des contrôles **TextBox** (zone de texte) et **ListBox** (zone de liste). Il permet aux utilisateurs d'entrer des informations dans la partie zone de texte ou de sélectionner un élément dans la partie zone de liste.

I.3.5. Le contrôle CommandButton :

Le contrôle CommandButton (bouton de commande) permet à l'utilisateur de commencer ou d'interrompre une opération, ou encore d'y mettre fin. Lorsque le bouton de commande correspondant est choisi, il semble s'enfoncer à l'écran, ce qui explique pourquoi on appelle parfois les boutons de commande " boutons-poussoirs ".

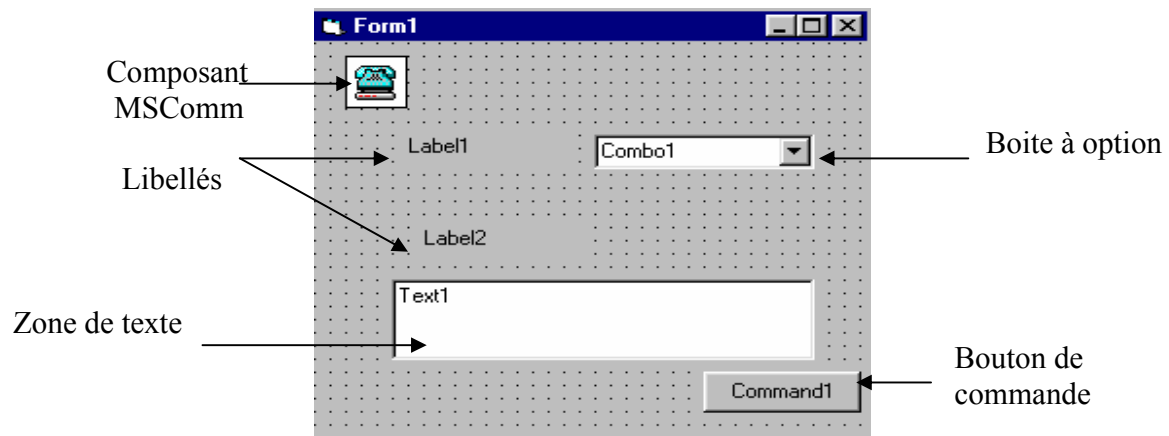
I.3.6. Le contrôle Label :

Un contrôle Label (étiquette) est un contrôle graphique qui permet d'afficher du texte que l'utilisateur ne peut pas modifier directement.

II. Programmation :

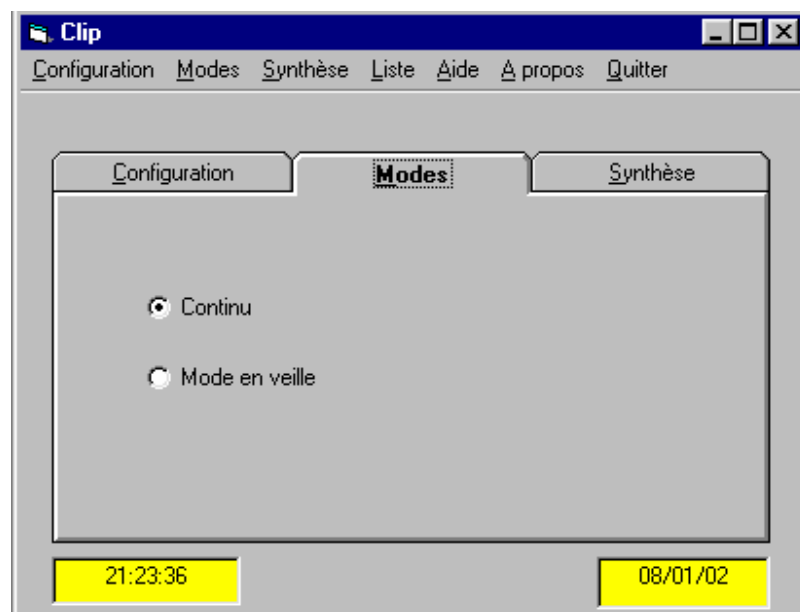
II.1. Création des fenêtres :

Pour élaborer un programme sous Visual Basic, nous devons créer une nouvelle forme dans laquelle nous plaçons des objets selon nos besoins. L'utilisation du composant MSComm , par exemple, nécessite l'emplacement de cet objet sur la feuille Form1, les champs nécessaires à l'affichage utilise le composant 'Text1', 'ComboBox1' ...etc.

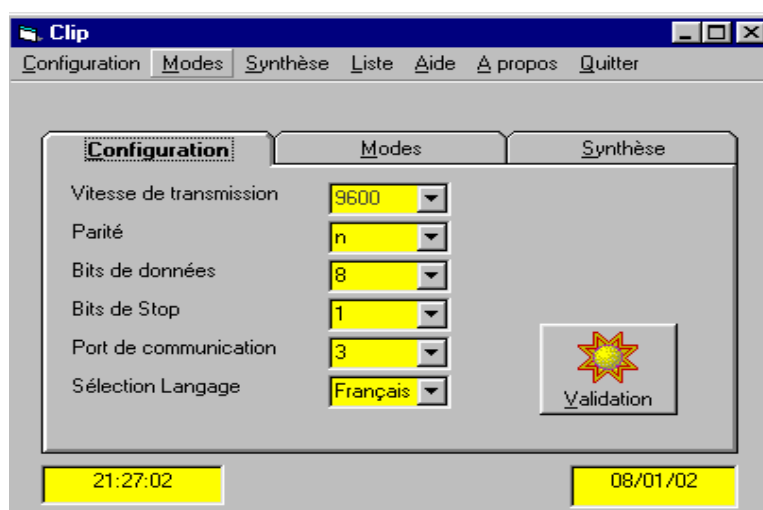


II.2. Fenêtre principale :

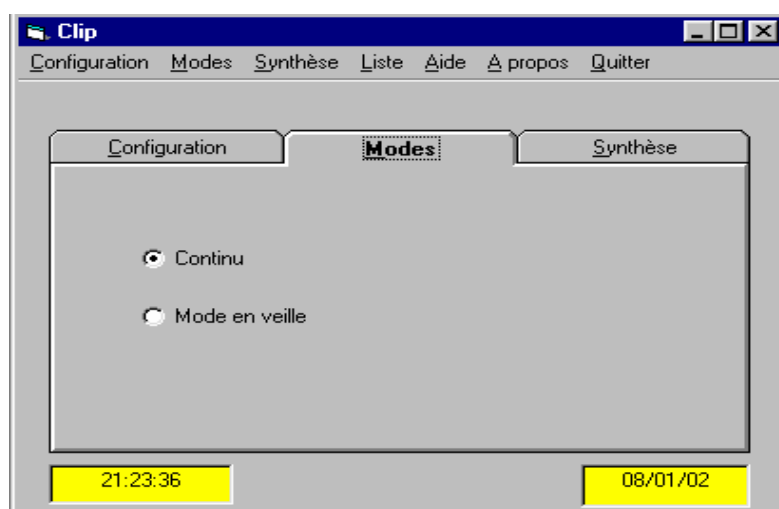
Cette fenêtre a la même forme qu'une fenêtre Windows avec une barre de titre et une barre de menu permettant d'accéder à la suite de l'application.



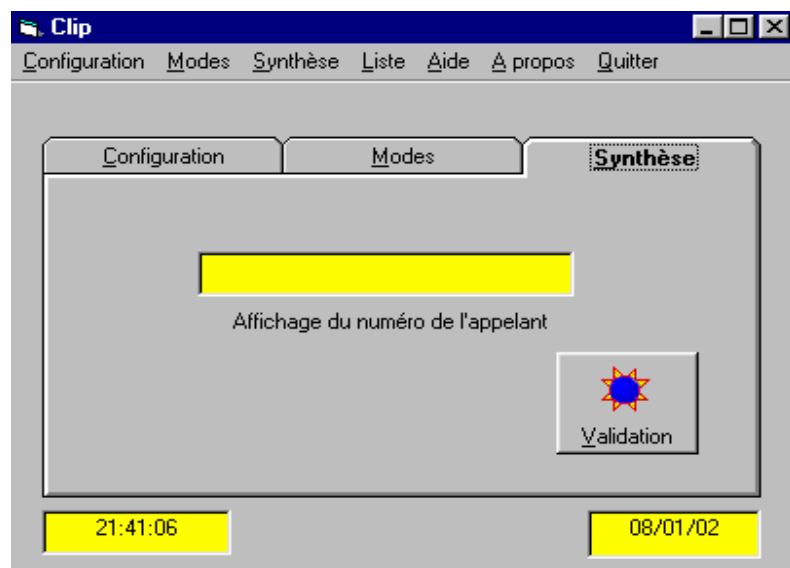
- ❖ **Le menu Configuration** : permet de configurer le port de communication en précisant le numéro du port et les quatre paramètres qui le caractérisent à savoir : la vitesse exprimée en Bauds, la parité, le nombre de bits de données et le nombre de bits de STOP.
- Nous avons ajouté a ce menu un contrôle ComboBox pour sélectionner le langage approprié (arabe, français..) pour la prononciation du numéro de l'appelant.



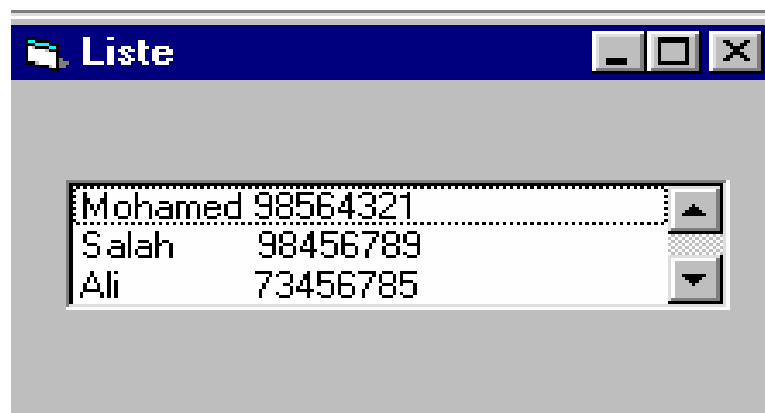
- ❖ **Le menu Modes** : ce menu définit deux états pour notre application. En mode continu, la fenêtre principale est visible sur l'écran et le programme est en cours d'exécution. Par contre, en mode veille, cette fenêtre sera réduite sans arrêt de l'exécution du programme.



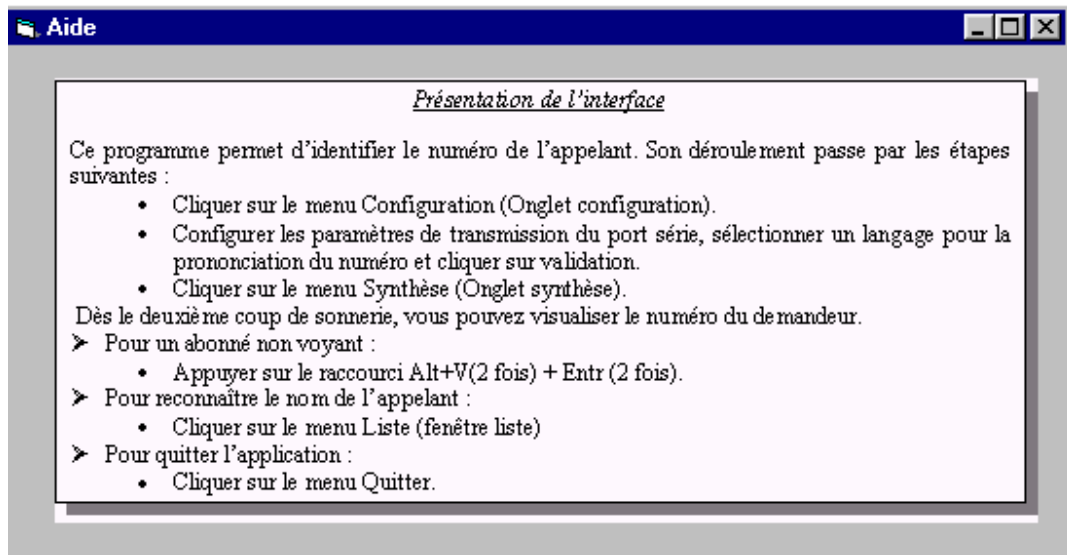
- ❖ **Le menu Synthèse :** C'est le menu principal à travers lequel nous pouvons accéder au numéro de l'appelant et à sa synthèse vocale. La fenêtre correspondante comprend une zone de texte réservée à l'affichage des numéros et un bouton de commande (nommé Validation) permettant de récupérer le message vocal par double clic.



- ❖ **Le menu Liste :** ce menu permet d'accéder à la fenêtre 'Liste' qui contient des numéros associés à des noms de demandeurs.



- ❖ **Le menu Aide :** C'est une fenêtre qui définit les différentes tâches permettant l'exécution du programme.



- ❖ **Le menu A propos :** se présente sous forme d'une fenêtre qui décrit le cadre de notre projet.



- ❖ **Le menu Quitter :** un simple clic sur ce menu permet de quitter l'application.

II.3. Les procédures de programmation :

II.3.1. Procédure de configuration du port série :

La configuration du port série prend en compte les paramètres introduits par l'utilisateur. Sa procédure est la suivante :

```
Private Sub Command1_Click()
```

```
MScomm1.CommPort = Combo5.Text
```

```
MScomm1.Settings = Combo1.Text + "," + Combo2.Text + "," + Combo3.Text + "," +  
Combo4.Text
```

```
End Sub
```

II.3.2. Procédure du choix de modes :

Pour choisir l'un de deux modes pour notre application, nous avons recours à la procédure suivante :

```
Private Sub Option2_Click()
```

```
Option1.Value = True
```

```
Option2.Value = False
```

```
Form1.WindowState = 1
```

```
End Sub
```

II.3.3. Procédure d'appel des Formes :

Vu que les boîtes de dialogue des menus (Configuration, Modes et Synthèse) appartiennent à la fenêtre principale, pour faire appel à ces dernières, on utilise les procédures suivantes :

```
Private Sub config_Click()
```

```
SSTab2.Tab = 0
```

```
End Sub
```

```
Private Sub Mode_Click()
```

```
SSTab2.Tab = 1
```

```
End Sub
```

```
Private Sub synt_Click()
```

```
SSTab2.Tab = 2
```

```
End Sub
```

A l'exception du menu Quitter, a chaque menu de la fenêtre principale (Liste, Aide et A propos) correspond une fiche. Pour y accéder, nous faisons appel à la procédure suivante :

```
Private Sub Form_Click()
```

```
Form N.Show
```

```
End Sub
```

N étant le numéro de la forme.

II.3.4. Procédure du temps et de la date :

```
Private Sub Timer1_Timer()
```

```
Text2.Text = Time
```

```
Text3.Text = Date
```

```
End Sub
```

II.3.5. Procédure de fermeture des formes :

La procédure à appliquer au bouton « Quitter » permet de sortir de notre application.

```
Private Sub quitter_Click()
```

```
End
```

```
End Sub
```

II.3.6. Procédure de la synthèse vocale :

Visual Basic offre un certain nombre de possibilités pour intégrer à notre application des sons. Sous la version standard du Visual Basic, pour produire du multimédia, nous devons emprunter la route des API de Windows 95. Une carte son est indispensable à la diffusion des fichiers son.

Qu'est ce que une API ?

Visual Basic permet d'appeler les fonctions prédéfinies du système d'exploitation Windows 95. Ces fonctions sont groupées dans une API (Application Programming Interface). Cette API se présente sous la forme d'un certain nombre de bibliothèques de lien dynamiques (DLL, Dynamic Link Libraries). Ainsi, les fonctions de l'API forment la base de Windows.

Pour utiliser une fonction de l'API dans notre application, nous devons la déclarer à Visual Basic : quelle est cette fonction, à quelle DLL appartient- elle et quels sont ces paramètres ?

Faire appel à une DLL revient, en quelque sorte, à faire appel à un module Visual Basic existant. La DLL ne fait pas partie du fichier EXE résultant de la compilation du projet, elle est simplement rattachée à l'application Visual Basic lors de l'exécution.

Dans le cas de notre application, le module Visual Basic que nous avons développé pour la déclaration de la fonction 'sndPlaySound' est le suivant :

Option Explicit

Global NomFic As String

Global TypeFichier As String

'API de support du son de haut niveau.

Declare Function JoueMusique Lib "WINMM.DLL" Alias "sndPlaySoundA" _

(ByVal NomFic As String, ByVal Mode As Long) As Long

La procédure à développer pour l’affichage du numéro de l’appelant et sa prononciation par les fichiers sons, lors de la manœuvre de la touche de raccourci [Alt+V(2 fois)+Entr (2 fois)] ou un double clic sur le bouton ‘ Validation’, est la suivante :

Private Sub Command2_Click()

Dim Nomfic, mode, chemin, chaine As String

Dim ReturnCode As Long

Dim j As Integer

Text1.Text = "0123456789"

 'Indique au contrôle qu'il doit lire la totalité du

 ' tampon lors de l'utilisation de la propriété Input.

MSComm1.InputLen = 0

 ' Ouvre le port.

If MSComm1.PortOpen = True Then

MSComm1.PortOpen = False

Else MSComm1.PortOpen = True

 ' Envoie la commande Attention au modem.

MSComm1.Output = "AT" + Chr\$(13)

 ' Attend le retour des données vers le port série.

Do

 Dummy = DoEvents()

Loop Until MSComm1.InBufferCount >= 2

 ' Lit les données composant la réponse "OK" au niveau

 ' du port série.

Text1.Text = MSComm1.Input

 ' Ferme le port série.

MSComm1.PortOpen = False

End If

 ' Pour gérer le son, une carte son avec le gestionnaire CODEC Microsoft ADPCM Audio est

 ' nécessaire. Ce gestionnaire est inclu dans Microsoft Windows 95 et

```
' Microsoft Windows NT 3.51. Pour Microsoft Windows 3.1x, le gestionnaire CODEC
' est commercialisé avec Microsoft Video for Windows version 1.1 ou avec
' Microsoft Sound System version 2.0
chemin = "D:\mahassen&faiza\prog2\"
chaine = "0123456789"
If Combo6.Text = "Français" Then
Nomfic = chemin + "f.wav"
ReturnCode = JoueMusique(Nomfic, 0)
For j = 1 To Len(chaine)
If Left(chaine, j) = 0 Then
Nomfic = chemin + "zerof.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 1 Then
Nomfic = chemin + "un.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 2 Then
Nomfic = chemin + "deux.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 3 Then
Nomfic = chemin + "trois.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 4 Then
Nomfic = chemin + "quatre.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 5 Then
Nomfic = chemin + "cinq.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 6 Then
Nomfic = chemin + "six.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 7 Then
PFE
```

```
Nomfic = chemin + "sept.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 8 Then
Nomfic = chemin + "huit.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 9 Then
Nomfic = chemin + "neuf.wav"
ReturnCode = JoueMusique(Nomfic, 0)
End If
Next j
```

```
*****
```

```
ElseIf Combo6.Text = "Anglais" Then
Nomfic = chemin + "e.wav"
ReturnCode = JoueMusique(Nomfic, 0)
For j = 1 To Len(chaine)
If Left(chaine, j) = 0 Then
Nomfic = chemin + "zeroe.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 1 Then
Nomfic = chemin + "one.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 2 Then
Nomfic = chemin + "two.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 3 Then
Nomfic = chemin + "three.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 4 Then
Nomfic = chemin + "four.wav"
ReturnCode = JoueMusique(Nomfic, 0)
```

```
ElseIf Right(Left(chaine, j), 1) = 5 Then
Nomfic = chemin + "five.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 6 Then
Nomfic = chemin + "sixe.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 7 Then
Nomfic = chemin + "seven.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 8 Then
Nomfic = chemin + "eight.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 9 Then
Nomfic = chemin + "nine.wav"
ReturnCode = JoueMusique(Nomfic, 0)
End If
Next j
```

```
ElseIf Combo6.Text = "Arabe" Then
Nomfic = chemin + "a.wav"
ReturnCode = JoueMusique(Nomfic, 0)
For j = 1 To Len(chaine)
If Left(chaine, j) = 0 Then
Nomfic = chemin + "sifr.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 1 Then
Nomfic = chemin + "wahed.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 2 Then
Nomfic = chemin + "ethnane.wav"
ReturnCode = JoueMusique(Nomfic, 0)
```



```

ElseIf Right(Left(chaine, j), 1) = 3 Then
Nomfic = chemin + "thalatha.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 4 Then
Nomfic = chemin + "arbaaa.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 5 Then
Nomfic = chemin + "khamssa.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 6 Then
Nomfic = chemin + "sitta.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 7 Then
Nomfic = chemin + "sabbaa.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 8 Then
Nomfic = chemin + "thamania.wav"
ReturnCode = JoueMusique(Nomfic, 0)
ElseIf Right(Left(chaine, j), 1) = 9 Then
Nomfic = chemin + "tissaa.wav"
ReturnCode = JoueMusique(Nomfic, 0)
End If
Next j
'*****
End If
End Sub

```

Note :

- Le paramètre 'chemin' varie selon le répertoire dans lequel se trouve notre application.
- Les commentaires entre guillemets sont des explications pour le programme.
- La chaîne « 0123456789 » est un exemple pour tester l'application.

Conclusion :

Dans ce chapitre, nous avons abouti au développement de l'interface logicielle qui a permis d'atteindre les objectifs de notre projet à savoir l'extraction, l'affichage du numéro de l'appelant et sa traduction en un message vocal.

Conclusion générale

Dans ce projet de fin d'études, il nous a été confié à l'étude de conception et de réalisation d'un prototype logiciel permettant à un abonné non voyant raccordé au réseau téléphonique commuté d'identifier son correspondant par voie vocale.

Nous avons alors procéder de la manière suivante, après avoir étudié la procédure d'émission ainsi que le mode de transmission du message CLIP, il nous a été possible d'établir une configuration qui répond aux exigences du cahier des charges. En effet, nous avons opté pour une solution moyennant l'utilisation d'un composant programmable, qui est le modem, et d'un PC piloté par un programme Visual Basic développé par nos soins.

En perspectives, nous pouvons rendre l'application autonome et indépendante d'un équipement informatique par développement d'une carte à base d'un microcontrôleur et d'un afficheur à cristaux liquide inséré directement sur la ligne téléphonique et moyennant un synthétiseur vocal., ce qui pourrait faire l'objet d'un travail d'études pour d'autres étudiants.