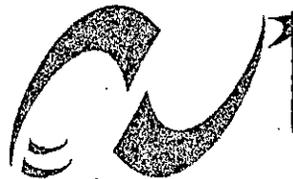


9/01

République Algérienne Démocratique et Populaire  
Ministère de l'enseignement supérieur et de la recherche scientifique

**ECOLE NATIONALE POLYTECHNIQUE**  
**DEPARTEMENT : D'ELECTRONIQUE**



المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

**PROJET DE FIN D'ETUDES**

Thème

**ETUDE ET CONCEPTION**

**D'UNE PLC**

**Proposé par :**

Mr : O. STIHI

**Etudié par :**

Mr : R. ZEGHILET

Mr : A. REGUIG

*Promotion Octobre 2001*

**E.N.P 10, Avenue HASSEN BADI-EL HARRACH-ALGER**

République Algérienne Démocratique et Populaire  
Ministère de l'enseignement supérieur et de la recherche scientifique

**ECOLE NATIONALE POLYTECHNIQUE**  
**DEPARTEMENT : D'ELECTRONIQUE**



**PROJET DE FIN D'ETUDES**

**Thème**

**ETUDE ET CONCEPTION**

**D'UNE PLC**

**Proposé par :**

Mr : O.STIHI

**Etudié par :**

Mr : R. ZEGHILET  
Mr : A. REGUIG

***Promotion Octobre 2001***

**E.N.P 10, Avenue HASSEN BADI-EL HARRACH-ALGER**

## Dédicaces

*Je dédie ce mémoire :*

*A mes très chers parents qui m'ont beaucoup soutenu et encouragé  
durant toute la période de mes études.*

*A ma grand mère que j'aime beaucoup.*

*A ma sœur Anissa.*

*A mes frères Seddik et lotfi.*

*A mes tantes , oncles, cousins et cousines.*

*A tous ce qui m'ont aidé de loin ou de près pour achever ce travail.*

*A tous mes ami(e)s, en particulier :*

*Amine, Chafik, Lotfi, Mohamed, Samy, Yasmine, Amel, Belaid, Mohsen*

*A mon binôme Riad ainsi qu'à tous les membres de sa famille.*

*Amine*

## Dédicaces

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

*Je dédie ce mémoire :*

*A mes très chers parents qui m'ont soutenu et encouragé durant toute la période de mes études.*

*A mes grands parents que j'aime beaucoup.*

*A mes sœur Houda, Karima , Meriem et Amina.*

*A mes frères Karim et Hakim.*

*A mes tentes, oncles, cousins et cousines*

*A tous ce qui m'ont aidé de loin ou de près pour mon travail.*

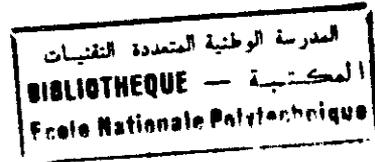
*A tous mes amis, en particulier :*

*Omar, Mustapha, Belaid, Mohsen et Mustapha.*

*A mon binôme Amine ainsi qu'à tous les membres de sa famille*

*Riad*

Remerciements



Remerciements

*Nous remercions M.Stihi pour avoir proposé le sujet.*

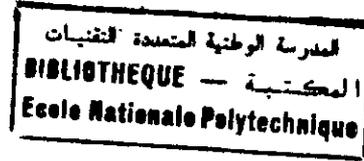
*Nous remercions vivement M.Trabelsi pour avoir présidé notre jury et pour nous avoir soutenu dans les moments difficiles, et Mlle Moussaoui membre du jury.*

*Notre gratitude et notre reconnaissance vont à tous les enseignants et tout le personnel du département d'électronique de l'Ecole Nationale Polytechnique.*

*Riad et Amine*

Résumé

Résumé



La saturation du réseau téléphonique, et l'indisponibilité des autres moyens de connexion à Internet, a conduit à la recherche de nouveaux moyens plus accessibles.

Et qu'y a-t-il de plus accessible que le réseau électrique. Les recherches se sont multipliées dans ce sens pour essayer de satisfaire une demande qui ne cesse d'augmenter.

C'est ainsi que nous avons consacré notre projet de fin d'études *Etude et conception d'une PLC* au transfert de données à travers le réseau électrique.

Notre travail se résume à transférer des données numériques (paquets TCP/IP) entre un serveur et des clients sur le câble d'alimentation électrique.

### Mots Clés

PLC

Adresse IP

Adresse de classe C

## Introduction

### Chapitre I le réseau Internet

#### I.1 Introduction

#### I.2 Le protocole Internet

##### I.2.1 Introduction

##### I.2.2 Présentation de TCP/IP

###### I.2.2.1 Présentation du modèle OSI

###### I.2.2.2 Le modèle OSI

###### I.2.2.3 Le modèle TCP/IP

###### I.2.2.4 Encapsulation des données

###### I.2.2.5 Format du datagramme IP

###### I.2.2.6 Les adresses IP

###### I.2.2.6.1 La classe A

###### I.2.2.6.2 La classe B

###### I.2.2.6.3 La classe C

#### I.3 Les liaisons physiques

##### I.3.1 Introduction

###### I.3.2.1 La connexion par la ligne téléphonique

###### I.3.2.2 La connexion par le câble

###### I.3.2.3 La connexion par satellite

#### I.4 conclusion

### Chapitre II Principe de fonctionnement de la PLC

#### II.1 Introduction

#### II.2 La modulation numérique

##### II.2.1 Introduction

##### II.2.2 Modulation numérique en amplitude

##### II.2.3. Modulation numérique de phase

###### II.2.3.1 Modulation d'amplitude et de phase

##### II.2.4 Modulation numérique en fréquence

#### II.3 Fonctionnement de la PLC

##### II.3.1 Introduction

##### II.3.2 Schéma synoptique de l'émetteur principal

##### II.3.3 principe de fonctionnement de l'émetteur

##### II.3.4 Principe de fonctionnement du récepteur

##### II.3.5 Aiguillage des données

## II.4 Interface avec l'ordinateur

### II.4.1 Protocole d'échange

### II.4.2 La liaison série

## II.5 interface avec les lignes électriques

## II.6 Conclusion

## Chapitre III Le PIC 16F84

### III.1 Introduction

### III.2 Description générale

#### III.2.1 Structure externe

#### III.2.2 Structure interne

#### III.2.3 Organisation de la mémoire

##### III.2.3.1 La mémoire programme

##### III.2.3.2 La mémoire RAM

##### III.2.3.3 Le SFR (Special Function Registers)

##### III.2.3.4 Le GPR (General Purpose Register)

#### III.2.4 Les registres

##### III.2.4.1 Introduction

##### III.2.4.2 Le registre INDF (ADRESSE 00, 80H)

##### III.2.4.3 Le registre FSR (ADRESSE 04, 84H)

##### III.2.4.4 Le registre TMRO (ADRESSE 01H)

##### III.2.4.5 Le registre OPTION (ADRESSE 81H)

###### III.2.4.5.1 Description des BITS

##### III.2.4.6 Le registre STATUS (ADRESSE 03 ET 83H)

###### III.2.4.6.1 Description des BITS

##### III.2.4.7 Le registre TRISA (ADRESSE 85H)

##### III.2.4.8 Le registre PORTA (ADRESSE 05H)

##### III.2.4.9 Le registre TRISB (ADRESSE 86H)

##### III.2.4.10 Le registre PORTB (ADRESSE 06H)

##### III.2.4.11 Le registre EEDATA (ADRESSE 08H)

##### III.2.4.12 Le registre EEADR (ADRESSE 09H)

##### III.2.4.13 Le registre EECON1 (ADRESSE 88H)

###### III.2.4.13.1 Description des BITS

##### III.2.4.14 Le registre EECON2 (ADRESSE 89H)

##### III.2.4.15 Le registre INTCON (ADRESSE 0B et 8BH)

###### III.2.4.15.1 Description des BITS

##### III.2.4.16 Le registre PCL (ADRESSE 02, 82H)

##### III.2.4.17 Le registre PCLATH (ADRESSE 0A, 8AH)

#### III.2.5 La périphérie

##### III.2.5.1 Le PORTA ET B

III.2.5.2 Le TIMER

III.2.5.3 Le diviseur programmable ou PRESCALER

III.2.5.4 Le chien de garde

III.2.5.5 L'EEPROM

III.2.6 Les interruptions

III.2.6.1 Déroulement d'une interruption

III.2.6.2 Les différentes sources d'interruption

III.2.7 La configuration de l'oscillateur

III.2.8 Le mode SLEEP

III.2.8.1 Introduction

III.2.8.2 Le mode SLEEP

III.2.8.3 Le "réveil" du microcontrôleur

III.2.9 Le jeu d'instruction du PIC16F84

**Chapitre IV Conception d'une PLC**

**IV.1 Introduction**

**IV.2 Réception des données en provenance du web**

**IV.3 Codage du signal issu de la FIFO**

**IV.4 l'émetteur**

IV.4.1 Introduction

IV.4.2 la boucle à verrouillage de phase

IV.4.3 Le MC145151-2

IV.4.4 La modulation des signaux

IV.4.5 L'isolation

**IV.5 Les récepteurs**

**IV.6 Le décodeur FSK**

**IV.7 l'alimentation du montage**

**IV.8 Conclusion**

**Conclusion générale**

# Introduction

## I-Introduction

L'Internet est certainement le système le plus complexe jamais conçu et déployé par l'homme. Cette infrastructure de communication permet de connecter à grande échelle de multiples réseaux et ordinateurs de technologies différentes. Son succès grandissant repose sur la mise en œuvre de quelques grands principes généraux des réseaux qui ont fait la preuve de leur robustesse lors de la croissance exponentielle de l'Internet.

Il existe des milliers et des milliers de réseaux télématiques, indépendants les uns des autres, et Internet est un système d'interconnexion entre ceux qui le souhaitent. Internet relie les réseaux qui ont choisi la même norme de communication (TCP/IP : Transport Control Protocol / Internet Protocol). Et chaque machine connectée à l'un de ces réseaux possède une adresse unique, attribuée exclusivement par une instance habilitée par « The Internet Society ».

La richesse et la diversité des services accessibles par Internet sont telles qu'une association n'en tire profit que dans le cadre de projets de communication clairement définis. Mais il présente l'avantage d'un monde sans frontière et ouvert à tout savoir et à toute initiative, propice aux associations.

Ainsi l'utilisation de l'Internet se fait à travers trois grandes périphéries connues de tout le monde, qui sont :

La connexion à travers le téléphone.

La connexion à travers le câble.

La connexion à travers le satellite.

Mais voilà vu l'engouement que connaît l'Internet et surtout la saturation que connaît le réseau téléphonique, qui est sans aucun doute le moyen le plus courant et surtout le plus accessible pour se connecter à Internet, on a presque été poussé à trouver un nouveau réseau qui serait une sorte de compromis entre ces trois grands réseaux et leurs avantages, tout en essayant d'être le plus accessible possible, et que pouvait-on trouver de plus accessible que le réseau électrique.

Ainsi, nous avons consacré la totalité de notre projet de fin d'études à ce procédé révolutionnaire qui va bientôt envahir nos maisons, et nous fera oublier les lourdes factures de téléphone.

Nous tenterons dans ce mémoire l'étude et la conception d'une PLC (Power Line Communication), à travers quatre grands chapitres :

Le 1<sup>ier</sup> chapitre donne un aperçu sur le protocole Internet et les différents moyens de connexion.

Le 2ieme chapitre présente brièvement le principe de fonctionnement de la PLC.

Le 3ieme chapitre est consacré à la description du microcontrolleur PIC16F84..

Le 4ieme chapitre illustre la conception d'une PLC.

# Chapitre I :

# Le Réseau

# Internet

## I le réseau Internet

### I.1 Introduction

Aux débuts de l'informatique des ordinateurs ont été mis au point, dès qu'ils furent aptes à fonctionner seuls, des personnes eurent l'idée de les relier entre eux afin qu'ils puissent échanger des données, c'est le concept de réseau. Il a donc fallu mettre au point des **liaisons physiques** entre les ordinateurs pour que l'information puisse circuler, mais aussi un langage de communication pour qu'il puisse y avoir un réel échange, on a décidé de nommer ce langage un **protocole**.

Des réseaux hétérogènes se sont développés aux quatre coins du globe ; des personnes décidèrent donc de relier ces réseaux entre eux (des universités par exemple ou l'armée). Les protocoles ont donc évolué pour permettre la communication de tous ces réseaux pour former le réseau des réseaux, formant petit à petit une gigantesque toile d'araignée ("web") formant le réseau le plus vaste, puisque contenant tous les réseaux, que l'on appelle **Internet**

### I.2 Le protocole Internet

#### I.2.1 Introduction

Sur Internet, de nombreux protocoles sont utilisés, ils font partie d'une suite de protocole qui s'appelle TCP/IP. TCP/IP est basé sur le repérage de chaque ordinateur par une adresse appelée adresse IP qui permet d'acheminer les données à la bonne adresse. Puis on a associé à ces adresses des noms de domaine pour permettre de s'en souvenir plus facilement.

#### I.2.2 Présentation de TCP/IP [FR]

TCP/IP est une suite de protocoles (utilisé sur Internet). Il signifie **Transmission Control Protocol/Internet Protocol**. Il représente la façon de laquelle les ordinateurs communiquent sur Internet. Pour cela il se base sur l'adressage IP, c'est-à-dire le fait de fournir une adresse IP à chaque machine du réseau afin de pouvoir acheminer des paquets de données. Etant donné que la suite de protocoles TCP/IP a été créée à l'origine dans

un but militaire, elle doit répondre à un certain nombre de critères parmi lesquels :

- Fonctionnement des messages en paquets.
- Utilisation d'un système d'adresse.
- Acheminement des données sur le réseau (routage).
- Contrôle des erreurs de transmission de données.

Afin de pouvoir appliquer le modèle TCP/IP à n'importe quelles machines, c'est-à-dire indépendamment du système d'exploitation, le système de protocoles TCP/IP a été décomposé en plusieurs modules effectuant chacun un rôle précis. De plus, ces modules effectuent des tâches les uns après les autres dans un ordre précis, on a donc un système stratifié, c'est la raison pour laquelle on parle de **modèle en couches**.

Le terme de couche est utilisé pour évoquer le fait que les données qui transitent sur le réseau traversent plusieurs **niveaux de protocoles**. Ainsi, les données (paquets d'informations) qui circulent sur le réseau sont traitées successivement par chaque couche, qui vient rajouter un élément d'information (appelé *en-tête*) puis sont transmises à la couche suivante.

Le modèle TCP/IP s'inspire du modèle OSI (modèle comportant 7 couches) qui a été mis au point par l'organisation internationale des standards (ISO, *international standard organisation*) afin de normaliser les communications entre ordinateurs.

### I.2.2.1 Présentation du modèle OSI [ISO]

OSI signifie (*Open Systems Interconnection*, ce qui se traduit par *Interconnexion de systèmes ouverts*). Ce modèle a été mis en place par l'ISO afin de mettre en place un standard de communications entre les ordinateurs d'un réseau, c'est-à-dire les règles qui gèrent les communications entre des ordinateurs. En effet, aux origines des réseaux chaque constructeur avait un système propre (on parle de système propriétaire). Ainsi de nombreux réseaux incompatibles coexistaient. C'est la raison pour laquelle l'établissement d'une norme a été nécessaire.

Le rôle du modèle OSI consiste à standardiser la communication entre les machines afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles (pour peu qu'ils respectent scrupuleusement le modèle OSI).

Le but d'un système en couches est de séparer le problème en différentes parties (les couches) selon leurs niveaux d'abstraction.

Chaque couche du modèle communique avec une couche adjacente (celle du dessus ou celle du dessous). Chaque couche utilise ainsi les services des couches inférieures et en fournit à celle de niveau supérieur.

### 1.2.2.2 Le modèle OSI [ISO]

Le modèle OSI est un modèle qui comporte 7 couches :

- **La couche physique** : définit la façon de laquelle les données sont converties en signaux numériques
- **La couche liaison donnée** : définit l'interface avec la carte réseau
- **La couche réseau** : permet de gérer les adresses et le routage des données
- **La couche transport** : elle est chargée du transport des données et de la gestion des erreurs
- **La couche session** : définit l'ouverture des sessions sur les machines du réseau
- **La couche présentation** : définit le format des données (leur représentation, éventuellement leur compression et leur cryptage)
- **La couche application** : assure l'interface avec les applications

### 1.2.2.3 Le modèle TCP/IP [FR]

Le modèle TCP/IP a été développé à peu près au même moment que le modèle OSI, c'est la raison pour laquelle il s'en inspire mais n'est pas totalement conforme aux spécifications du modèle OSI.

Le modèle TCP/IP reprend l'approche modulaire (utilisation de modules ou couches) mais en contient uniquement quatre :

Modèle TCP/IP	Modèle OSI
Couche Application	Couche Application Couche Présentation Couche Session
Couche Transport (TCP)	Couche Transport
Couche Internet (IP)	Couche Réseau
Couche Accès réseau	Couche Liaison données Couche Physique

Comme on peut le remarquer, les couches du modèle TCP/IP ont des tâches beaucoup plus diverses que les couches du modèle OSI, étant donné que certaines couches du modèle TCP/IP correspondent à plusieurs couches du modèle OSI.

Les rôles des différentes couches sont les suivants :

- **Couche accès réseau** spécifie la forme sous laquelle les données doivent être acheminées quel que soit le type de réseau utilisé
- **Couche Internet** : elle est chargée de fournir le paquet de données (datagramme)
- **Couche Transport** : elle assure l'acheminement des données, ainsi que les mécanismes permettant de connaître le résultat de la transmission
- **Couche Application** : elle englobe les applications standard du réseau (Telnet, SMTP, FTP,...)

#### 12.2.4. Encapsulation des données [FR]

Lors d'une transmission, les données traversent chacune des couches au niveau de la machine émettrice. A chaque couche, une information est ajoutée au paquet de données, il s'agit d'un **en-tête**, ensemble d'informations qui garantit la transmission. Au niveau de la machine réceptrice, lors du passage dans chaque couche, l'en-tête est lu, puis supprimé. Ainsi, à la réception, le message est dans son état originel...

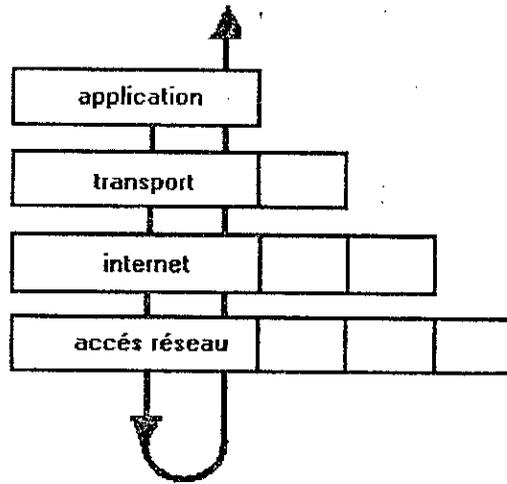


Figure I.2.2.4 :Encapsulation des données

A chaque niveau, le paquet de données change d'aspect, car on lui ajoute un en-tête, ainsi les appellations changent suivant les couches :

- Le paquet de données est appelé **message** au niveau de la couche application
- Le message est ensuite encapsulé sous forme de **segment** dans la couche transport
- Le segment une fois encapsulé dans la couche Internet prend le nom de **datagramme**
- Enfin, on parle de **trame** au niveau de la couche accès réseau

#### I.2.2.5 Format du datagramme IP [FR]

Un datagramme IP se présente de la manière suivante :

Version	I.H.L	Type de service	Longueur totale	
Identificateur			drapeaux	Offset de fragmentation
Durée de vie	Protocole		Checksum de l'entête	
Adresse source				
Adresse destination				
Options				Rembourrage
Données				

**Version** : 4 bits

Ce champ indique la version du format d'en-tête utilisée. La version utilisée actuellement est la version 4.

**I.H.L** : (Internet header length) 4 bits

Le champ Longueur d'En-tête code la longueur de l'en-tête Internet, l'unité étant le mot de 32 bits, et de ce fait, marque le début des données. Notez que ce champ ne peut prendre une valeur en dessous de 5 pour être valide.

**Type de service** : 8 bits

Le Type de Service donne une indication sur la qualité de service souhaitée, qui reste cependant un paramètre "abstrait". Ce paramètre est utilisé pour "guider" le choix des paramètres des services actuels lorsqu'un datagramme transite dans un réseau particulier. Certains réseaux offrent un mécanisme de priorité, traitant préférentiellement un tel trafic par rapport à un trafic moins prioritaire (en général en acceptant seulement de véhiculer des paquets d'un niveau de priorité au-dessus d'un certain seuil lors d'une surcharge momentanée). Principalement, le choix offert est une négociation

entre les trois contraintes suivantes : faible retard, faible taux d'erreur, et haut débit.

### **Longueur totale : 16 bits**

Le champ "Longueur Totale" est la longueur du datagramme entier y compris en-tête et données, mesurée en octets. Ce champ ne permet de coder qu'une longueur de datagramme d'au plus 65,535 octets. Une telle longueur rendrait les datagrammes impossible à gérer pour la plus grande partie des réseaux. Les hôtes devront au moins pouvoir accepter des datagrammes d'une longueur jusqu'à 576 octets (qu'il s'agisse d'un datagramme unique ou d'un fragment). Il est de même recommandé que des hôtes ne décident d'envoyer des datagrammes de plus de 576 octets que dans la mesure où ils sont sûrs que la destination est capable de les accepter.

Le nombre 576 a été choisi pour permettre à un bloc de données de taille raisonnable d'être transmis dans un datagramme, tenant compte des données à ajouter pour constituer les en-têtes de protocole. Par exemple, cette taille permet la transmission d'un bloc de 512 octets, plus 64 octets d'en-tête dans un datagramme unique. La taille maximale d'un en-tête Internet étant de 60 octets, et sa taille typique étant de 20 octets, ce nombre permet de conserver une bonne marge pour les données protocolaires de plus haut niveau.

### **Identification : 16 bits**

Une valeur d'identification assignée par l'émetteur pour identifier les fragments d'un même datagramme.

### **drapeaux: 3 bits**

Divers commutateurs de contrôle.

Bits 0-2 : Réserve, doit être laissé à zéro

Bit 1 : (AF) 0 = Fragmentation possible      1 = Non fractionnable.

Bit 2 : (DF) 0 = Dernier fragment      1 = Fragment intermédiaire.

### **Offset de fragmentation : 13 bits**

Ces champs indiquent la portion datagramme auquel ce fragment appartient.

**Durée de vie :** 8 bits

Ce champ indique la durée de vie maximale du datagramme dans le système Internet. Une valeur nulle indique que ce datagramme doit être détruit.

**Protocole :** 8 bits

Indique le protocole utilisé dans la portion de données.

**Checksum de l'en-tête :** 16 bits

Vérification de la validité de l'en-tête IP.

**Adresse source :** 32 bits

Adresse Internet de la machine émettrice du datagramme.

**Adresse destination :** 32 bits

Adresse Internet de la machine destinataire du datagramme.

**Options :**

Ce champ a une taille variable. Il n'est pas obligatoire et peut ne pas apparaître dans le datagramme.

**Rembourrage :**

Utilisé en cas d'option pour amener, si besoin, la longueur de l'en-tête Internet à un multiple de 32 bits. Le bourrage se fait par des octets à zéro.

#### I.2.2.6 Les adresses IP [PI]

Sur Internet, les ordinateurs communiquent entre eux grâce au protocole TCP/IP qui utilise des numéros de 32 bits, que l'on écrit sous forme de 4 numéros allant de 0 à 255 (4 fois 8 bits), on les note donc sous la forme xxx.xxx.xxx.xxx où chaque xxx représente un entier de 0 à 255. Ces numéros servent aux ordinateurs du réseau pour se reconnaître, ainsi il ne doit pas exister deux ordinateurs sur le réseau ayant la même adresse IP.

Les adresses IP sont réparties en classes, c'est-à-dire selon le nombre d'octets qui représentent le réseau.

#### I.2.2.6.1 La classe A

Dans une adresse IP de classe A, le premier octet représente le réseau. Le bit de poids fort (le premier bit, celui de gauche) est à zéro, ce qui signifie qu'il y a  $2^7$  (00000000 à 01111111) possibilités de réseaux, c'est-à-dire 128. Toutefois le réseau 0 (00000000) n'existe pas et le nombre 127 est réservé pour désigner votre machine, les réseaux disponibles en classe A sont donc les réseaux allant de **1.0.0.0** à **126.0.0.0** (lorsque les derniers octets sont des zéros cela indique qu'il s'agit d'un réseau et non d'un ordinateur!)

Les trois octets de droite représente les ordinateurs du réseau, le réseau peut donc contenir :

$$2^{24} - 2 = 16777214 \text{ ordinateurs.}$$

Une adresse IP de classe A, en binaire, ressemble à ceci :

1xxxxxxx / xxxxxxxx xxxxxxxx xxxxxxxx

Réseau      Ordinateurs

#### I.2.2.6.2 La classe B

Dans une adresse IP de classe B, les deux premiers octets représente le réseau. Les deux premiers bits sont 1 et 0, ce qui signifie qu'il y a  $2^{14}$  (10 000000 00000000 à 10 111111 11111111) possibilités de réseaux, c'est-à-dire 16384. Les réseaux disponibles en classe B sont donc les réseaux allant de **128.0.0.0** à **191.255.0.0**

L'octet de droite représente les ordinateurs du réseau, le réseau peut donc contenir :

$$2^{16} - 2^1 = 65534 \text{ ordinateurs.}$$

Une adresse IP de classe B, en binaire, ressemble à ceci :

10xxxxxx xxxxxxxx / xxxxxxxx xxxxxxxx

Réseau                      Ordinateurs

#### I.2.2.6.3 La classe C

Dans une adresse IP de classe C, les trois premiers octets représente le réseau. Les deux premiers bits sont 1,1 et 0, ce qui signifie qu'il y a  $2^{21}$  possibilités de réseaux, c'est-à-dire 2097152. Les réseaux disponibles en classe C sont donc les réseaux allant de **192.0.0.0** à **255.255.255.0**

L'octet de droite représente les ordinateurs du réseau, le réseau peut donc contenir :

$$2^8 - 2^1 = 254 \text{ ordinateurs.}$$

Une adresse IP de classe C, en binaire, ressemble à ceci :

110xxxxx xxxxxxxx xxxxxxxx / xxxxxxxx

Réseau

Ordinateurs

Classe	Nombre de réseaux possibles	Nombre d'ordinateurs max. sur chacun
A	126	16777214
B	16384	65535
C	209753	254

Le but de la division des adresses IP en trois classes A, B et C est de faciliter la recherche d'un ordinateur sur le réseau. En effet avec cette notation il est possible de rechercher dans un premier temps le réseau que l'on désire atteindre puis de chercher un ordinateur sur celui-ci. Ainsi l'attribution des adresses IP se fait selon la taille du réseau. Les adresses de classe A sont réservées aux très grands réseaux, tandis que l'on attribuera les adresses de classe C à des petits réseaux d'entreprise par exemple.

### I.3 Les liaisons physiques

#### I.3.1 Introduction

Le réseau Internet est un système de mise en commun de l'information entre plusieurs machines. Il permet de relier, au moyen d'équipements de communication appropriés, des ordinateurs partout dans le monde. Cette connexion nécessite, d'une part un protocole pour uniformiser les transferts ainsi que des supports physiques pour permettre la connexion. Ces supports peuvent être filaires ou non filaires, c'est ce que nous allons essayer d'illustrer dans ce paragraphe.

#### I.3.2.1 La connexion par la ligne téléphonique

Pour établir cette connexion en mode RTC (RTC : Réseau Téléphonique Commuté), l'utilisation d'un modem est nécessaire. Ce modem a pour objectif de convertir le langage informatique (données

numériques) en langage téléphonique (données analogiques), et inversement.

Une ligne téléphonique est conçue pour fonctionner avec un téléphone, c'est pour cela que le modem a besoin d'établir une communication avec un ordinateur distant grâce à un numéro de téléphone avant de pouvoir échanger des informations.

Ce mode d'accès étant cependant techniquement limité par la performance du modem est appelé à être remplacé progressivement par d'autres types de connexions plus rapides et plus performants.

### I.3.2.2 La connexion par le câble [MU]

Ce type de connexion utilise le câble de la télévision. Les données télé et Internet passe sur des canaux différents comme deux chaînes de télévision. Ensuite une rallonge (un câble coaxial blindé) qui va jusqu'au modem spécial (un Motorola : il y a très peu de fabricants pour l'instant) qui est branché à l'ordinateur par l'intermédiaire d'une carte Ethernet.

Le câble permet une connexion à haut débit. Ainsi les échanges avec Internet sont plus fluides, les images ou vidéos s'affichent plus vite, les fichiers se téléchargent plus rapidement mais Le câble est un réseau asymétrique. Autrement dit, recevoir de l'information (débit descendant) va plus vite que d'en envoyer (débit remontant). Ce qui alourdit le nombre des données envoyées (puisqu'elles mettent plus de temps à être expédiées).

### I.3.2.3 La connexion par satellite [TS]

Le satellite est un moyen d'accéder à l'Internet en passant par un prestataire de service spécialisé qui retransmet les demandes des utilisateurs par satellite. Les utilisateurs équipés d'une carte de réception DVB-MPEG2 réseau et d'une parabole reçoivent leurs demandes par satellite à des débits qui peuvent atteindre plus de 2Mb/s. En théorie, il est possible de recevoir un retour de fichier par satellite à des vitesses pouvant aller jusqu'à 48Mb/s.. Dans la pratique, c'est nettement moins.

Bien que des expérimentations soient actuellement menées sur la transmission de données bidirectionnelle entre un PC et un satellite, le matériel le plus fréquemment utilisé dans le cadre d'une utilisation résidentielle reste la carte de réception DVB-MPEG-2 et le modem

### Figure II.2.3 :PSK

La représentation de ces modulations se fait dans le système de coordonnées I, Q. Chaque sommet représente une des valeurs (symboles) possibles. Dans ces exemples un symbole permet de coder respectivement 1, 2 et 3 bits. Les valeurs seront prélevées en synchronisme avec une horloge qu'il faut reconstituer à partir des signaux I, Q. Les lignes reliant les points indiquent l'évolution des composantes I, Q lorsque l'on passe d'un symbole à

l'autre. Dans les exemples ci-dessus toutes les transitions sont permises, cela ne sera pas toujours le cas. Certaines transitions ont une amplitude plus faible que d'autres. Cela pourra conduire à des difficultés pour récupérer l'horloge.

Il existe des variantes à ces types de modulation appelés OQSP (*Offset QSP* – les composantes *I* et *Q* sont transmises avec un bit de décalage) et  $\pi/4$  DQSP (*Differential QSP* – il s'agit de deux constellations décalées de  $\pi/4$ ) qui présentent un certain nombre d'avantages quant à l'efficacité de la transmission.

### II.2.2.3.1 Modulation d'amplitude et de phase

Un des systèmes utilisé, s'appelle nQAM (Quadrature Amplitude modulation). La valeur n indique le nombre de symboles utilisé (par exemple 32QAM correspond à 32 symboles, chaque symbole codant 5 bits). On rencontre couramment les types suivants : 16QAM, 32QAM, 64QAM, 128QAM et 256QAM.

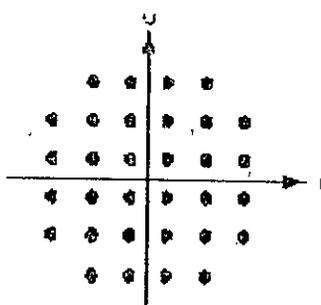


Figure II.2.3.1 :32QAM

L'ensemble des points représentant les symboles forment une constellation. On construit la constellation en prenant les valeurs  $(-n, \dots, -1, 1, \dots, n)$  pour *I* et *Q*. Dans l'exemple ci-dessus  $n=3$  ce qui donne 36 valeurs et l'on enlève les valeurs les plus éloignées de l'origine.

Un des problèmes avec ce système est que les transitions entre symboles voisins sont de faible amplitude et ont un mauvais rapport signal-bruit. C'est pour cette raison que l'on a développé des codes dits convolutionnels.

Avec un codage statique, à chaque valeur binaire correspondra toujours le même symbole.

Dans le cas des codes convolutionnels le symbole est choisi en fonction du code à émettre et des codes précédemment émis. Ce traitement est effectué au moyen d'une opération appelée convolution qui est réalisée par des additionneurs modulo 2 et un registre à décalage.

### II.2.4 Modulation numérique en fréquence

Plus connu sous le nom de F.S.K (Fréquence Shift Keying), ce type de modulation a pour but de changer la fréquence du signal analogique suivant qu'il s'agisse d'un "1" logique ou d'un "0" logique. Si en entrée, on a un "1" logique, alors en sortie, ce bit sera représenté par un signal de fréquence  $2f$ . Et si en entrée, on a un "0" logique, alors en sortie, ce bit sera représenté par un signal de fréquence  $f$  (le double).

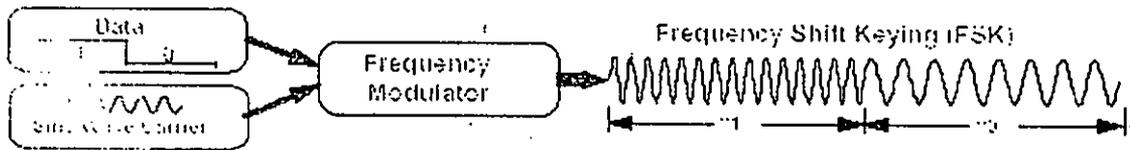


Figure II.2.4 :FSK

La modulation FSK est pour le moment la modulation la plus utilisée dans le domaine de la communication par modem, elle est simple à réaliser (par rapport à la PSK) et les risques d'erreurs restent minimes, donc, c'est la FSK que nous choisissons pour notre application.

## II.3 Fonctionnement de la PLC

### II.3.1 Introduction

La « power line communication » repose sur le principe du courant porteur. En effet, dans le courant alternatif, les électrons sont « tirés » dans un sens puis dans l'autre, créant ainsi une fréquence plus ou moins rapide : 50 cycles par secondes. À ces oscillations, on ajoute des fréquences de cinq à 16 MHz qui codent du son, des images ou du texte. Ces deux types de signaux sont d'abord combinés, à proximité de la maison, par des transformateurs électriques modifiés puis, à l'intérieur même du domicile, séparés à nouveau par un modem branché sur une prise de courant. Puisque chaque transformateur alimente plusieurs maisons, un multiplexage est nécessaire pour envoyer les données aux endroits appropriés.

Le schéma de notre montage se compose de deux parties, la première est l'émetteur/récepteur principal ou « serveur », la seconde est l'émetteur/récepteur terminal ou « client ».

### II.3.2 Schéma synoptique de l'émetteur principal

Le schéma est donné par la figure suivante :

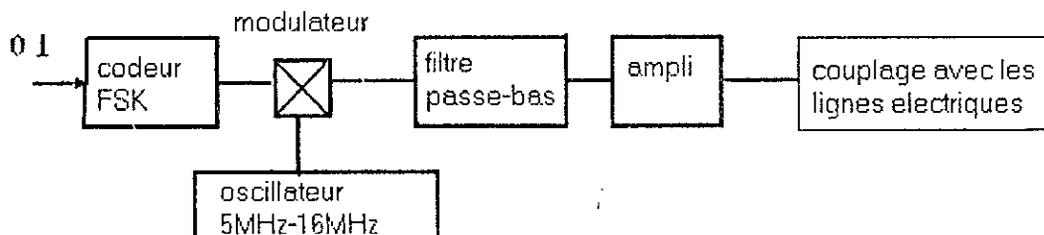


Figure II.3.2 : Schéma synoptique de l'émetteur principal

### II.3.3 principe de fonctionnement de l'émetteur

La séquence binaire à l'entrée du bloc FSK est codée de façon à avoir une fréquence de 250 KHz pour un « 1 », et une fréquence de 500 KHz pour un « 0 ». Ce signal module une porteuse délivrée par l'oscillateur, la fréquence de la porteuse dépend du client, car, on alloue à chaque client une fréquence sur laquelle il peut émettre et recevoir. Le signal composite ainsi constitué sera filtré et amplifié puis couplé au lignes électrique.

L'émetteur du client aura la même configuration, mais la fréquence de la porteuse est fixe et elle dépend de l'adresse IP attribuée à ce client.

### II.3.4 Schéma synoptique du récepteur

Le schéma est donné par la figure suivante :

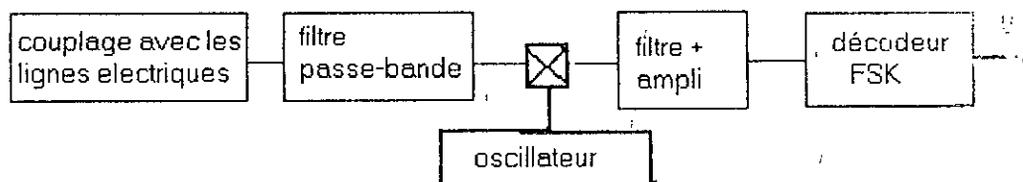


Figure II.3.4 : Schéma synoptique du récepteur

### II.3.5 Principe de fonctionnement du récepteur

Le signal issu du réseau électrique est filtré par un filtre passe-bande centré sur la fréquence propre à l'utilisateur, ce signal est ensuite amplifié puis démodulé pour avoir à la sortie les deux fréquences 250 KHz et 500 KHz du signal message. Le « tone decoder » décode ce signal pour la séquence binaire correspondante.

Du côté du serveur, on aura plusieurs récepteur avec la même configuration précédente, mais chacun est accordé sur la fréquence d'un client différent.

### II.3.6 Aiguillage des données

Chaque utilisateur possède une adresse IP à laquelle nous associons une fréquence porteuse spécifique.

Lorsqu'un utilisateur émis un paquet TCP/IP, celui-ci est modulé puis transmis vers le serveur à travers la prise du courant sur laquelle le modem est branché.

A la réception d'un paquet en provenance du web, le serveur compare l'adresse destination dans son entête avec les adresses IP des clients (qui sont prédéfinis) pour déterminer la fréquence porteuse à moduler par ce paquet de données (donc l'utilisateur vers lequel les données doivent être envoyées).

## II.4 La liaison série

C'est la liaison utilisée pour faire communiquer deux modems.

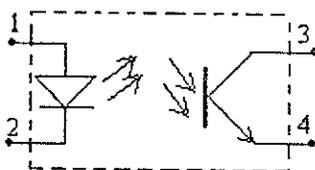
A l'inverse d'une liaison parallèle, les données d'une liaison série sont présentées bit par bit.

Une liaison série permet de faire des liaisons sur de grande distance car elle est moins sensible aux parasites que les liaisons parallèles. Son inconvénient majeur est la lenteur à laquelle sont récupérés les bits. En effet, pour une liaison parallèle à 4 fils, on fait passer 4 bits à chaque cycle d'horloge alors qu'on en fait passer qu'un avec une liaison série, cet inconvénient est compensé par le fait que la liaison série est plus fiable pour des transmissions à grande distance.

## II.5 Interface avec les lignes électriques

Cette partie du modem forme une passerelle entre les lignes de données et les lignes électriques. Elle doit prévoir l'alimentation du modem ainsi que l'isolation. Pour l'alimentation, nous allons utiliser un transformateur suivi d'un pont redresseur et un régulateur. Pour l'isolation, nous allons utiliser un opto-coupleur.

L'opto-coupleur se compose d'une LED et d'un composant photo-sensible. En faisant allumer cette LED, on déclenche le composant. Celui-ci peut être un transistor, un darlington, un thyristor, un triac...



En fait cet ensemble est assimilable à une LED mise face à un photo transistor (ou triac, etc.) dans un milieu totalement isolé. Cela procure une isolation galvanique et électrique très performante.[TI]

## II.6 Conclusion

Nous avons exposés dans ce chapitre les différents type de modulation ainsi que la procédure à suivre pour réaliser la PLC. Nous avons souligné que la modulation FSK multi-porteuses est la plus appropriée pour ce type de communication, que l'utilisation des adresses IP réservées nous permet d'aiguiller les données et que la liaison série est la plus sûre.

# Chapitre III :

## Le PIC16F84

## III Le PIC16F84

### III.1 Introduction

Aujourd'hui, les microcontrôleurs se valent pratiquement tous. Alors pourquoi opter pour un monochip PIC ? Les réponses sont multiples :

- Les performances de ce monochip sont identiques voire même supérieures à ses concurrents.
- Ses prix sont les plus bas du marché
- Les outils de développements sont pratiquement tous gratuits et téléchargeables sur le web.
- Le jeu d'instruction associé au microcontrôleur est à la fois souple, puissant et facile à maîtriser.

Un monochip PIC existe en trois versions. Une version dite reprogrammable destinée aux développements permet de créer, vérifier, modifier à volonté le programme jusqu'au produit final. Le second groupe nommé version OPT (Only Programmable Time) est utilisé uniquement pour la production en série (concrétisation du projet) et enfin, une version mixte.

### II.2 Description générale [MI]

Le PIC16F84 est un microcontrôleur 8 bits de faible coût. Certes, il ne peut se substituer aux autres microcontrôleurs (68HC11 par exemple) mais il permet toutefois de créer de nombreuses applications.

De plus, avec son jeu d'instructions réduit, il est très agréable à utiliser lorsqu'on désire s'initier à l'étude d'un microcontrôleur.

Enfin, ses caractéristiques internes et externes sont très appréciable :

1K de mémoire programme

68 octets de RAM

64 octets D'EEPROM

13 ENTREE/SORTIE

4 SOURCES D'INTERRUPTION

1 TIMER/COMPTEUR

1 CHIEN DE GARDE

MODE SLEEP (pour une faible consommation)

4 SOURCES D'OSCILLATEUR SELECTIONNABLE

PROTECTION DU CODE

PROGRAMMATION PAR ISP (In Serial Programming)

### III.2.1 Structure externe

Le PIC16F84 est logé dans un boîtier 18 broches DIL ou SOIC (version CMS).

La figure ci-dessous nous décrit le brochage du PIC16F84 :

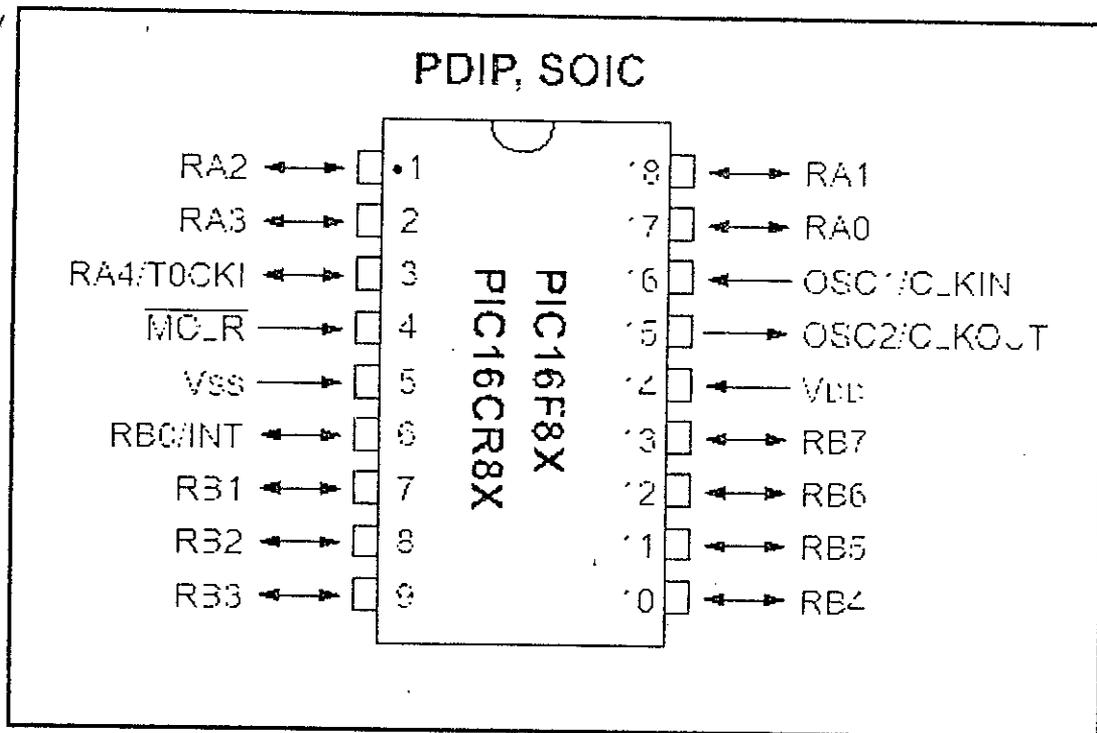


Figure III.2.1 : structure externe

### III.2.2 Structure interne

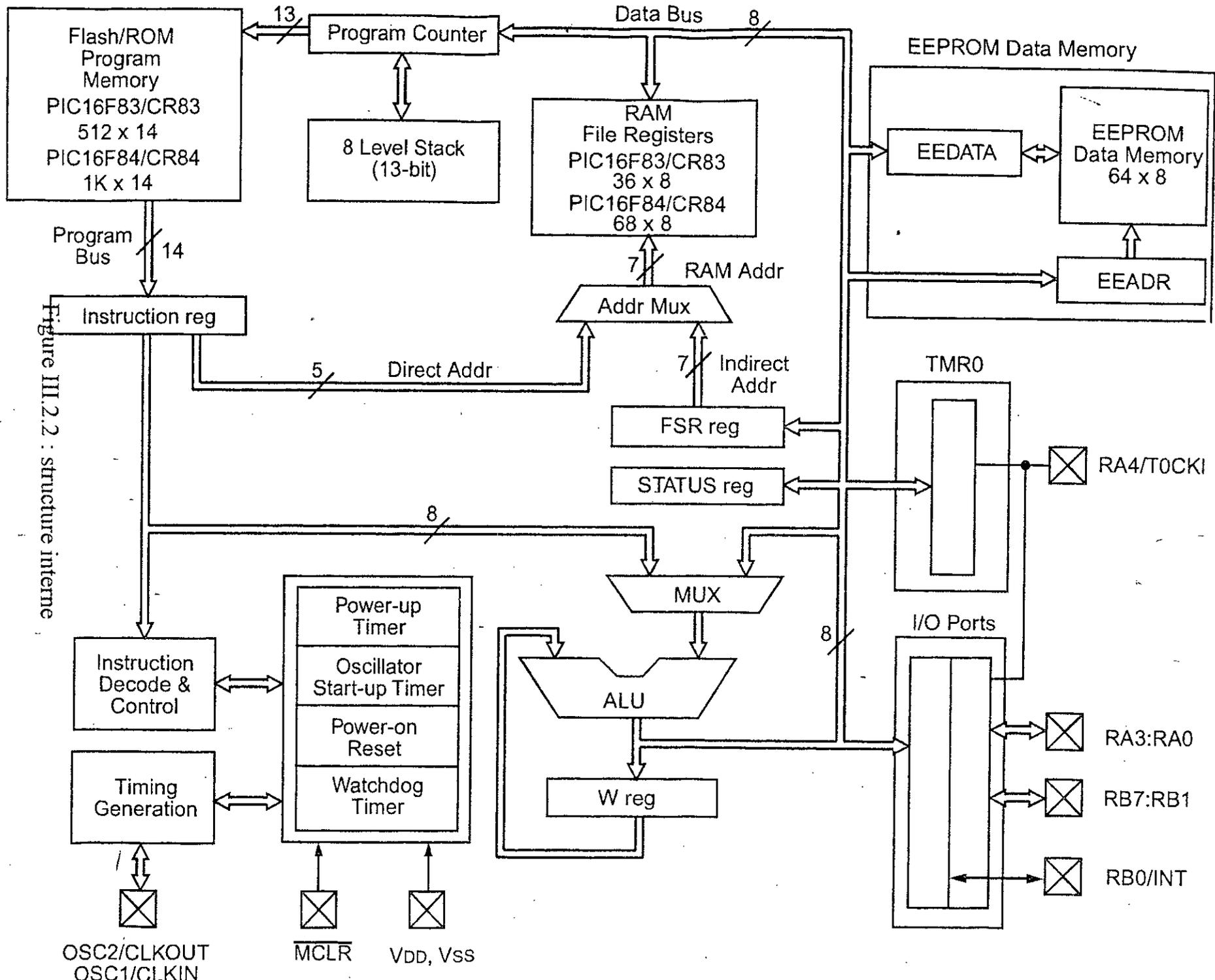


Figure III.2.2 : structure interne

### III.2.3 Organisation de la mémoire

La mémoire du PIC16F84 est divisée en deux parties distinctes : la mémoire programme et la mémoire de donnée.

Le premier contient les instructions du programme ainsi que les vecteurs RESET et INTERRUPTION.

Dans la mémoire de donnée se trouvent tout les registres nécessaires pour la configuration et l'utilisation des périphériques internes ainsi que 68 octets de RAM utilisateurs.

#### III.2.3.1 La mémoire programme

Cette mémoire de 1K stock le programme. L'emplacement de celui-ci peut être à n'importe quel endroit de la mémoire. Cependant, il faut savoir que le PIC16F84, suite à un RESET ou lorsqu'on l'alimente, commence toujours à l'adresse 0000H (Vecteur RESET).

De plus, lorsqu'il y a une interruption, et si celle-ci est validée, le microcontrôleur va à l'adresse 0004H (Vecteur d'INTERRUPTION).

Figure III.2.3.1 : la mémoire programme

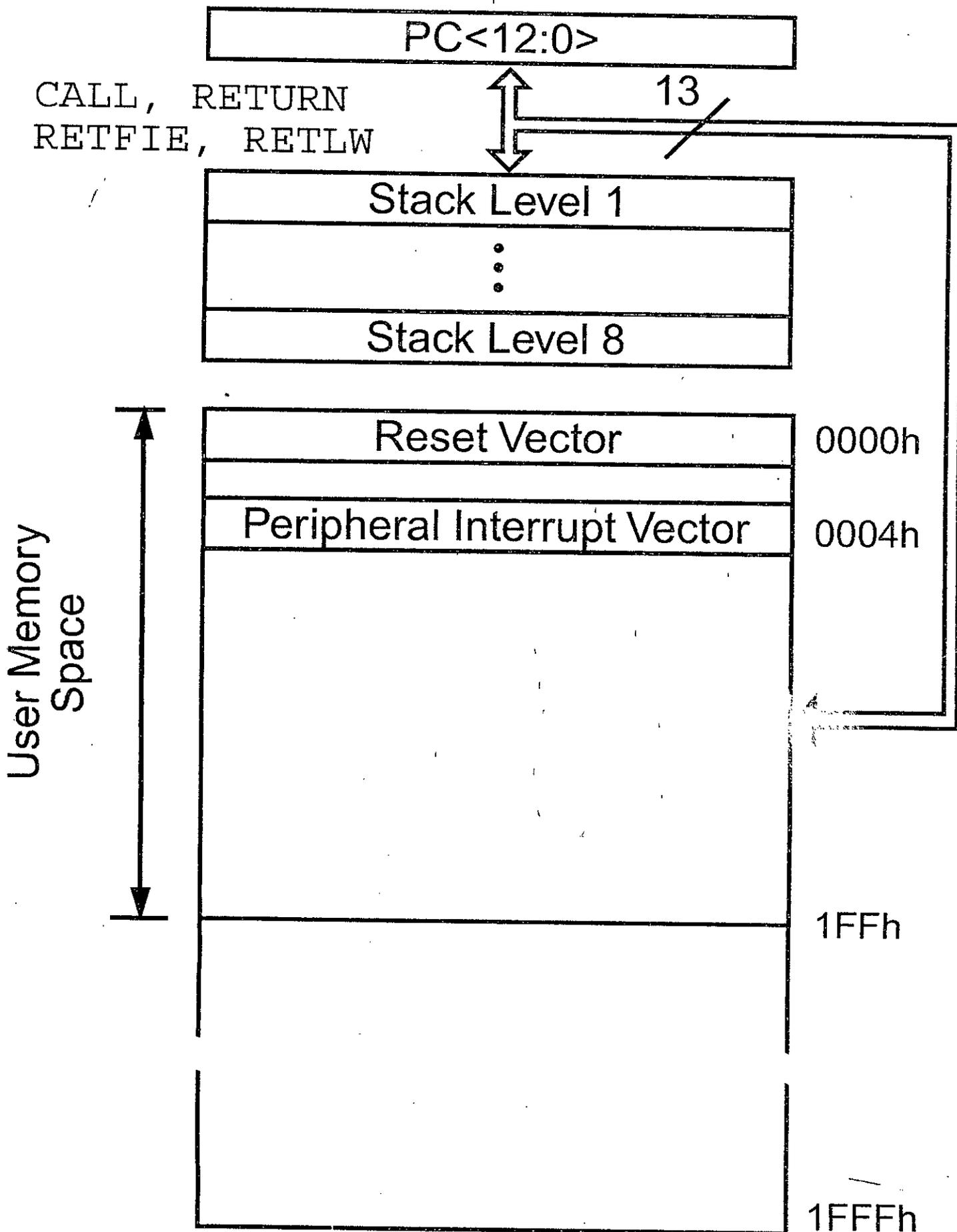


Figure III.2.3.1 : la mémoire programme

### III.2.3.2 La mémoire RAM

La mémoire de donnée est divisée en deux parties. Une partie appelée SFR est encore divisée en deux pages (Bank0 et Bank 1). Toutes les données de la mémoire sont appelées registres y compris les données utilisateurs.

### III.2.3.3 Le SFR (Special Function Registers)

Le SFR permet de configurer tout les modules internes du PIC16F84 (les PORTS, le TIMER, la gestion des interruptions, etc.). Il est de plus utilisé par le CPU.

Pour accéder à l'un de ces registres, il faut d'abord se positionner sur la page (Bank) appropriée. Ensuite, il suffit de lire ou d'écrire dans le registre nommé.

L'adressage direct des pages se fait par deux bits du registre de STATUS (RP0 et RP1). On peut utiliser l'adressage indirect avec le registre FSR.

La figure III.2.3.3 nous montre les différents registres du SFR ainsi que leur adresse associée.

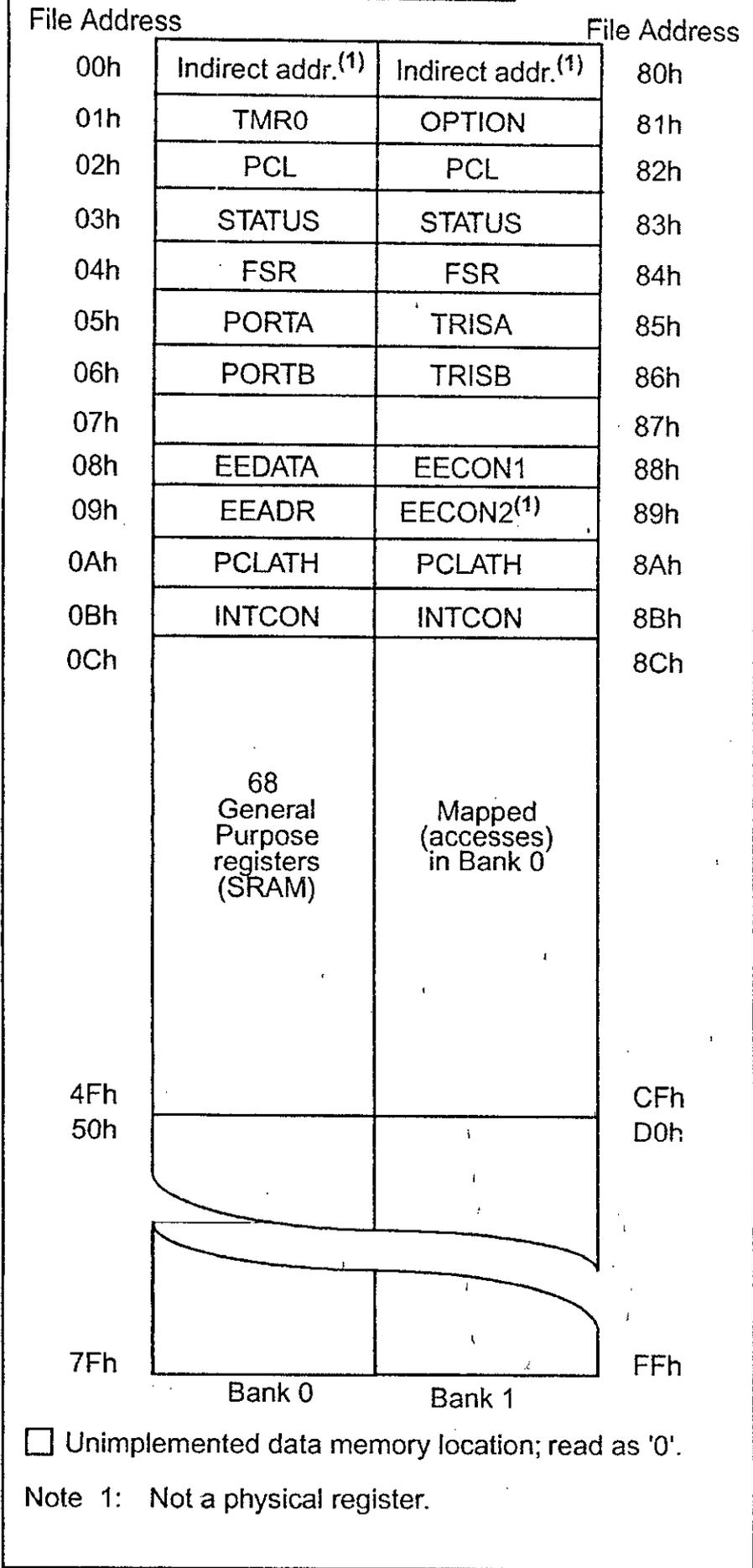


Figure III.2.3.3 : les registres du SFR

### III.2.3.4 Le GPR (general Purpose Register)

Le GPR est en fait la RAM (8bits) utilisateurs. C'est ici que l'on stocke les variables et les données. L'utilisation des pages (Bank ) n'est pas nécessaire puisque le Bank 1 est "mapped" avec le Bank0. Cela signifie qu'écrire une donnée à l'adresse 0CH ou à l'adresse 8CH revient au même.

Enfin, l'adressage peut se faire de manière directe ou indirecte (en utilisant le registre FSR).

### III.2.4 Les registres

#### III.2.4.1 Introduction

Ces registres spéciaux font partie du SFR (Special Function Registers) et configurent le microcontrôleur : Certains registres initialisent les périphériques alors que d'autres sont utilisés par le CPU.

Cette partie décrit uniquement les bits du registre nommé. L'organisation de la mémoire du PIC16F84 a été donnée précédemment.

#### III.2.4.2 Le registre INDF (ADRESSE 00, 80H)

Voir le registre FSR.

#### III.2.4.3 Le registre FSR (ADRESSE 04, 84H)

Le registre INDF ainsi que FSR permettent l'adressage INDIRECT.

Le principe est d'utiliser deux registres intermédiaires pour accéder aux données :

Dans le premier, appelé FSR, on inscrit l'adresse (en fait, il est utilisé comme un pointeur).

L'autre registre (INDF), contient la valeur qui se trouve à l'adresse définie dans le FSR.

exemple:

Admettons la situation suivante :

Adresse	Valeur
10	3
11	4
12	5

Si l'on veut lire successivement les données se trouvant entre l'adresse 10 et 12, une des premières méthodes est de lire une après une, les données.

L'autre méthode, beaucoup plus souple, réalise la même fonction avec l'adressage indirecte :

BOUCLE MOVLW 0X10

MOVWF FSR ; on initialise le registre FSR avec l'adresse 10

MOVF INDF, W ; on retrouve dans W la valeur 3

MOVF VAR ; que l'on met dans VAR

INCF FSR ; on incrémente FSR et a pour adresse maintenant la valeur

11

GOTO BOUCLE ; et on retourne en BOUCLE. Au prochain passage, VAR contiendra la valeur 4.

#### III.2.4.4 Le registre TMRO (ADRESSE 01H)

Ce registre de 8 bits s'incrémente de "1" a chaque impulsion de l'horloge (externe ou interne). Il est associé au module TIMER/COMPTEUR.

#### III.2.4.5 Le registre OPTION (ADRESSE 81H)

Le registre d'OPTION contiens les bits de contrôles du PRESCALER, de l'interruption externe INT, de la sélection TIMER/COMPTEUR et du "tirage au plus" du PORT B.

Bit 7	Bit6	Bit5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RBP U	INTE DG	OCS	TOS E	PSA	PS2	PS1	PS0

##### III.2.4.5.1 Description des BITS

BIT 0,1 et 2 : Sélectionnent le ratio du PRESCALER.

Ces trois bits déterminent le ratio (effectue une division de la fréquence d'horloge) du PRESCALER.

S2	S1	S0	RATIO TMRO	RATIO WDT
			1:2	1:1
			1:4	1:2
			1:8	1:4
			1:16	1:8
			1:32	1:16
			1:64	1:32
			1:128	1:64
			1:256	1:128

BIT 3 : PSA (PreScaler Assignment).

Si PSA=1 alors le prescaler est associé avec le WDT.

Si PSA=0 alors le prescaler est associé avec le TIMER.

BIT 4 : TOSE (tmro Source Edge).

Ce bit détermine sur quel front (montant ou descendant) l'entrée RA4 incrémentera le registre TMRO

TOSE=1 Front descendant

TOSE=0 Front montant.

BIT 5 : TOCS (tmro Clock Source)

Il permet de sélectionner le mode de fonctionnement du TIMER/COMPTEUR.

TOCS=1 sélection de l'horloge externe (broche RA4) qui correspond au COMPTEUR.

TOCS=0 sélection de l'horloge interne et permet au module de travailler en mode TIMER.

BIT 6 : INTEDG (INTerrupt EDGe)

Si INTEDG=1 alors la broche RBO/INT génère une interruption sur un front montant.

Si INTEDG=0 alors la broche RBO/INT génère une interruption sur un front descendant.

BIT 7 : RBPU (port B Pull-Up).

RBPU=1 Le "tirage au plus" interne du PORT B est désactivé.

RBPU=0 Le "tirage au plus" interne du PORT B est activé.

### III.2.4.6 Le registre STATUS (ADRESSE 03 ET 83H)

Divisé en plusieurs blocks, il contient :

Le résultat d'un calcul de l'ALU représenté par les bits C, DC et Z.

Le status d'un RESET par l'intermédiaire des bits PD et TO.

L'accès aux différents bancs de la mémoire par les bits IRP, RP1 et RP0.

bit 7	Bit 6	bit 5	Bit 4	bit 3	bit 2	bit 1	bit 0
R/W	R/W	R	R	R/W	R/W	R/W	R/W
IRP	RP1	RP0	TO	PD	Z	DC	

### III.2.4.6.1 Description des BITS

BIT0: C (Carry).

Il passe à "1" lorsque le résultat d'une opération dépasse la valeur FF ou d'un résultat négatif.

BIT1: DC (Digital Carry).

Il se positionne à "1" lorsqu'une retenue s'est produite entre les bits 3 et 4.

BIT2: Z (Zéro).

Quand il est à "1", il indique que le résultat de l'opération est nul.

BIT3: PD (Power-Down).

PD=1 après un Power-up ou suite à l'instruction CLRWDT.

PD=0 lorsque l'instruction SLEEP est exécutée.

BIT4: TO (Time-Out).

Il passe à "1" après les instructions SLEEP et CLRWDT. Lorsque le WDT (chien de garde) déborde, ce bit passe à "0".

BIT5 et 6 : RP0 et RP1 (bits de sélection de bancs).

Avec ces 2 bits, on accède aux différents bancs de la mémoire.

RP1	RP0	BANC	ADRESSE
0	0	0	00 à 7F
0	1	1	80 à FF
1	0	2	100 à 17F
1	1	3	180 à 1FF

BIT7: IRP.

Ce bit n'est pas utilisé avec le PIC16F84 et doit rester à "0".

### III.2.4.7 Le registre TRISA (ADRESSE 85H)

Le registre TRISA configure chaque E/S du PORTA en ENTREE ou en SORTIE. Après un RESET, toutes les E/S sont en ENTREE. Si le bit

associé à la porte est à "1", alors elle sera configurée en ENTREE. Si le bit est à "0", elle sera en SORTIE.

La broche RA4 est multipléxée avec l'entrée d'horloge du registre TMRO.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
			R/W	R/W	R/W	R/W	R/W
			TRISA4	TRISA3	TRISA2	TRISA1	TRISA0

#### III.2.4.8 Le registre PORTA (ADRESSE 05H)

Le PORTA est un port de 5 bits (RA0 à RA4). Chaque E/S est compatible TTL. La configuration de chaque BIT du port est déterminée avec le registre TRISA.

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	Bit 0
			R/W	R/W	R/W	R/W	R/W
			RA4/TOCK	RA3	RA2	RA1	RA0

#### III.2.4.9 Le registre TRISB (ADRESSE 86H)

Le registre TRISB configure chaque E/S du PORTB en ENTREE ou en SORTIE. Apres un RESET, toutes les E/S sont en ENTREE. Si le bit associé à la porte est à "1", alors elle sera configurée en ENTREE. Si le bit est à "0", elle sera en SORTIE.

La broche RB0 est multipléxée avec l'interruption INT.

Les broches RB4 à RB4, à condition qu'elles soient configurées en ENTREE, peuvent générer une interruption lorsqu'elles changent d'états.

Enfin, toutes les broches du PORTB bénéficient d'un "tirage au plus" interne.

bit 7	bit 6	Bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
R/W							
TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0

#### III.2.4.10 Le registre PORTB (ADRESSE 06H)

Le PORTB est un port bidirectionnel de 8 bits. Toutes les broches sont compatibles TTL. La configuration du PORTB est réalisée avec le registre TRISB.

bit 7	bit 6	bit 5	Bit4	bit 3	Bit 2	bit 1	bit 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0

### III.2.4.11 Le registre EEDATA (ADRESSE 08H)

Ce registre de 8 bits permet de lire ou d'écrire une donnée dans la mémoire non volatile (EE PROM).

### III.2.4.12 Le registre EEADR (ADRESSE 09H)

Registre de 8 bits qui contient l'adresse de la donnée se trouvant dans l'EEPROM.

### III.2.4.13 Le registre EECON1 (ADRESSE 88H)

C'est un registre de contrôle qui permet d'exécuter une lecture ou une écriture dans l'EEPROM.

Seuls les 5 bits de poids faible sont utilisés.

bit 7	Bit 6	bit 5	Bit 4	bit 3	bit 2	bit 1	bit 0
			R/W	R/W	R/W	W	W
			EEIF	WRERR	WREN	WR	RD

#### III.2.4.13.1 Description des BITS

BIT 0 : RD (ReaD EEPROM).

Lorsque ce bit est mis à "1", il indique au microcontrôleur que l'on souhaite une lecture de l'EEPROM.

Après le cycle de lecture, il est mis automatiquement à 0.

BIT 1 : WR (WRite EEPROM).

Lorsque ce bit est mis à "1", il indique au microcontrôleur que l'on souhaite une écriture de l'EEPROM.

Après le cycle d'écriture, il est mis automatiquement à 0.

BIT 2 : WREN (EEPROM WRite ENABLE).

C'est un bit de confirmation d'écriture dans l'EEPROM. En effet, il ne suffit pas de définir un cycle d'écriture uniquement avec le bit WR. Il faut impérativement valider le bit WREN (WREN=1) pour autoriser une écriture.

BIT 3 : WRERR (EEPROM WRite ERROR flag).

Ce drapeau indique qu'une erreur c'est produite lors d'un cycle d'écriture dans l'EEPROM.

WRERR=1 une opération d'écriture a échoué.

WRERR=0 le cycle d'écriture c'est déroulé normalement.

BIT 4 : EEIF (eeprom Interrupt Flag).

EEIF est un drapeau qui génère une interruption lorsqu'un cycle d'écriture c'est déroulé normalement. Il doit être mis à 0 lors de la routine d'interruption.

EEIF=1 l'opération c'est déroulé correctement.

EEIF=0 soit l'opération n'a pas commencé, soit n'est pas terminée.

#### III.2.4.14 Le registre EECON2 (ADRESSE 89H)

Ce registre de 8 bits est exclusivement utilisé pour les séquences d'écritures dans l'EEPROM. Il n'a pas d'adresse physique et la lecture de ce registre retourne une valeur nulle.

#### III.2.4.15 Le registre INTCON (ADRESSE 0B et 8BH)

Le registre INTCON contient tout les bits de validation de chaque source d'interruption ainsi que leur drapeau (Flag).

Les drapeaux doivent être mis à 0 après l'interruption

bit 7	bit 6	Bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
R/W							
GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

##### III.2.4.15.1 Description des BITS

BIT 0 : RBIF (rb port change Interrupt Flag).

C'est un drapeau d'interruption qui indique un changement d'état du PORTB (RB4 à RB7).

RBIF=1 une broche (RB4 à RB7) a changé d'état.

RBIF=0 pas de changement d'état.

BIT 1 : INTF (INT interrupt Flag).

Drapeau d'interruption de l'entrée RB0.

RBIF=1 une interruption est apparue.

RBIF=0 pas d'interruption.

BIT 2 : TOIF (tmro Overflow Interrupt Flag).

Ce drapeau indique un dépassement du registre TMRO (passage de FF à 00).

TOIF=1 dépassement de TMRO.

TOIF=0 pas de dépassement.

BIT 3 : RBIE (RB Interrupt Enable).

Bit de validation qui autorise une interruption lors d'un changement d'état du PORT B (RB4 à RB7).

RBIE=1 Autorise l'interruption.

RBIE=0 Les changements d'état du PORT B (RB4 à RB7) ne générant pas d'interruption.

BIT 4 : INTE (int interrupt Enable).

Valide ou non l'interruption générée par la broche RB0/INT

INTE=1 Valide l'interruption INT.

INTE=0 Pas d'interruption provenant de INT.

BIT 5 : TOIE (TMRO Overflow Interrupt Enable).

Autorise ou non l'interruption provoquée par le dépassement du registre TMRO (passage de FF à 00)

TOIE=1 Valide l'interruption.

TOIE=0 Pas d'interruption provenant du registre TMRO.

BIT 6 : EEIE (EE write Interrupt Enable).

Ce bit autorise ou non une interruption lorsqu'un cycle d'écriture dans L'EEPROM c'est déroulé normalement.

EEIE=1 Génère une interruption lorsqu'un cycle d'écriture c'est déroulé normalement.

EEIE=0 Pas d'interruption.

BIT 7 : GIE (Global Interrupt Enable).

Ce bit autorise ou non toutes les interruptions.

GIE=1 Toutes les interruptions sont prises en compte par le microcontrôleur.

GIE=0 Aucune interruption ne sera validée.

#### III.2.4.16 Le registre PCL (ADRESSE 02, 82H)

Le PC (Program Counter) est un pointeur de 13 bits qui contient l'adresse de la prochaine instruction à exécuter.

Il est composé de deux registres :

Le PCL : C'est un registre de 8 bits accessible en écriture et en lecture et compose la partie basse du PC.

Le PCLATH : C'est un registre de 8 bits dont seuls les 5 bits de poids faible sont utilisés. Non accessible directement, il définit la partie haute du PC. Il permet d'une part, de sélectionner les différentes pages de la mémoire programme et d'autre part, son contenu est transféré lorsque le PC est chargé avec une nouvelle valeur.

REMARQUE: Le PIC16F84 n'a que 1 K de programme mémoire. Par conséquent, les bits 3 et 4 du PCLATH ne sont pas utilisés.

#### III.2.4.17 Le registre PCLATH (ADRESSE 0A, 8AH).

Voir le registre PCL.

### III.2.5 La périphérie

#### III.2.5.1 Le PORTA ET B

Les PORTS A et B sont des ENTREE/SORTIE réalisant le dialogue entre le microcontrôleur et son environnement extérieur. Le PIC16F84 possède 13 entrées/sorties réparties sur deux PORTS (A et B) de la manière suivante :

Le PORT A est un bus de donnée bidirectionnel de 5 bits. Chaque bit peut être configuré en ENTREE ou en SORTIE. Le niveau d'ENTREE/SORTIE est compatible TTL.

Le PORT A est constitué de 5 bits. Les bits RA0 à RA3 ont un niveau d'entrée TTL et une paire de transistors CMOS complémentaires en sortie.

Le bit RA4 est lui aussi compatible TTL en entrée mais sa sortie est en DRAIN OUVERT.

De plus, configuré en ENTREE, il peut être utilisé pour l'incrémementation du compteur interne TMRO.

C'est le registre de direction TRISA qui détermine l'orientation de chaque broche du port. Un "1" dans le registre TRISA positionne la broche associée en ENTREE, alors qu'un "0" la met en SORTIE.

Après un RESET, le PORTA est configuré en ENTREE.

#### REGISTRES ASSOCIES.

ADR	NOM	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
05H	PORT A				RA4/ TOCK	RA3	RA2	RA1	RA0
85H	TRISA				TRISA4	TRISA3	TRISA2	TRISA1	TRISA0

Le port B est un port bidirectionnel de 8bits (RB0 a RB7).

Chaque broche est compatible TTL. L'orientation de chaque bit se fera avec le registre de direction TRISB. On remarquera au passage l'utilisation d'un BUFFER tri-states (3 états) pour l'étage de SORTIE. Celui-ci passe à l'état de HAUTE IMPEDANCE lorsqu'une broche est configurée en ENTREE (un "1" dans le registre TRISB).

Un tirage au plus, "weak pull-up" (WPU), a été intégré sur chaque broche. Il permet ainsi d'avoir un "1" logique lorsqu'une broche est en ENTREE (évitant ainsi des composants externes supplémentaires). Le contrôle s'effectue avec le bit RBPU du registre d'OPTION (actif à l'état bas). Si une broche est configurée en SORTIE ou lors d'un POWER-ON-RESET (POR), l'étage WPU sera alors

automatiquement inhibé. Les 4 broches (RB4 à RB7) peuvent générer une INTERRUPTION (uniquement si elles sont configurées en ENTREE). En regardant la structure interne, on constate que les 2 verrous et la porte logique OU EXCLUSIF forment un ensemble qui compare la dernière valeur lue et la valeur présente sur la broche. S'il y a une différence (un changement d'état), la sortie de la porte sera au niveau haut.

Pour effacer l'interruption, on devra alors, dans la routine d'interruption, lire (écrire) le port ou mettre à zéro le drapeau RBIF du registre INTCON. Cette interruption peut être utilisée pour "réveiller" le PIC lorsqu'il est en mode SLEEP.

La broche RB0 peut-elle aussi générer une INTERRUPTION (drapeau INTF du registre INTCON).

#### REGISTRES ASSOCIES.

ADR	NOM	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
06H	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT
86H	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
81H	OPTION	RBPU							

#### III.2.5.2 Le TIMER

Le TIMER/COMPTEUR est une interface programmable, qui permet de réaliser une large gamme d'applications : compteur d'impulsion, fréquencemètre, générateur de fonction, etc. donc, il convient pour tous les montages nécessitant une base de temps de précision.

La différence entre un TEMPORISATEUR (TIMER) et un COMPTEUR (COUNTER) réside dans le choix de l'horloge qui l'incrémente. Si l'on opte pour une horloge externe, on disposera alors d'un COMPTEUR. Dans ce cas, l'horloge sera synchronisée avec l'horloge interne du microcontrôleur.

Si le choix se porte sur l'horloge interne, l'interface se comportera comme un TIMER.

Le principe est assez simple : il s'agit en effet d'incrémenter le registre TRMO de 1 à chaque impulsion d'horloge interne ou externe. Il est possible d'intercaler un autre compteur (PRESCALER) afin de réaliser une division supplémentaire de la fréquence de l'horloge.

La structure interne nous présente les principaux organes qui composent un TIMER. A part le registre TMR0, chacun d'eux est configuré avec un ou plusieurs BITS du registre d'OPTION.

##### LE REGISTRE TMR0

C'est un registre 8 bits accessible en lecture ou en écriture. Il est incrémenté à chaque cycle d'horloge de PSOUT.

Dans le cas d'un dépassement (lorsque la valeur passe de FF à 00), le

drapeau TOIF passe à "1" et génère une interruption (s'il a été validé dans le registre INTCON).

### III.2.5.3 Le diviseur programmable ou PRESCALER

C'est en fait un compteur de 8 bits qui divise la fréquence PSIN par le ratio défini avec les bits TS0, TS1, TS2 (voir le tableau récapitulatif pour les ratios).

Le prescaler est partagé avec le WDT (WatchDog Timer). Par conséquent, il n'est pas possible d'utiliser le diviseur simultanément avec le WDT et le TIMER/COMPTEUR. Si le PRESCALER est indispensable aux deux, l'alternative est alors la seule solution.

C'est le bit PSA qui détermine l'appartenance du prescaler. Avec un niveau bas ("0") sur PSA, le diviseur est associé au TIMER/COMPTEUR. Si PSA est à "1", il appartient alors au WDT.

#### Sélection de l'horloge

Le choix de l'horloge interne ou externe définit le mode TIMER ou COMPTEUR. La mise à zéro du bit TOCS sélectionne le TIMER et inhibe l'horloge externe.

Si TOCS est à "1", le mode COMPTEUR est pris en compte. Dans ce cas de figure, un autre bit, TOSE, définit le front sur lequel aura lieu l'acquisition (TOSE=0 pour un front montant, TOSE=1 pour un front descendant).

Dans le cas du COMPTEUR, l'horloge externe est synchronisée à l'horloge interne. Il en résulte deux conséquences à ne pas négliger :

Premièrement, il y a un retard entre l'acquisition de l'horloge externe et l'incrémentation (typiquement deux cycles d'horloge interne) du registre TMRO.

Deuxièmement, pour que l'impulsion soit prise en compte, il faut absolument que la période de l'horloge externe soit au minimum égale à 4 cycles d'horloge interne (sans prescaler) ou à 4 cycles d'horloge interne/ N avec l'utilisation du prescaler (n étant le ratio).

#### REGISTRES ASSOCIES

On indique ici que les bits utilisés.

ADR	NOM	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
01H	TMRO								
0BH	INTCON	GIE		TOIE			TOIF		
81H	OPTION			TOCS	TOSE	PSA	PS2	PS1	PS0
85H	TRISA				TRISA4				

### III.2.5.4 Le chien de garde

Chien de garde, "WATCHDOG TIMER" (WDT), permet d'assurer, de manière simple, le bon déroulement d'un programme. Si le programme " plante ", le WDT génère un RESET ou un autre événement permettant, soit à l'utilisateur, soit au logiciel, de réagir en conséquence.

Le WDT du PIC16F84 est composé d'un oscillateur interne à réseau RC indépendant et d'un compteur 8bits. La période de l'oscillateur varie en fonction de la température et de la tension d'alimentation (de 10ms à 60ms dans les cas extrêmes).

#### DEROULEMENT

A chaque cycle d'horloge de l'oscillateur, le compteur s'incrémente de 1. A la fin du comptage, c'est à dire lorsque le compteur passe de FF à 00, le WDT met le bit TO (Time Out) du registre STATUS à 0. Si l'on souhaite augmenter le temps du chien de garde, on peut utiliser le prescaler. C'est un autre compteur binaire programmable qui est mis en série avec le compteur du WDT et permet ainsi d'augmenter le temps (jusqu'à 2.3 secondes). Il est important de noter que le prescaler est partagé avec le TIMER. Il n'est donc pas possible de les utiliser simultanément.

Le WDT est configurable uniquement lors de la programmation du microcontrôleur, c'est à dire dans le "mot de configuration".

#### APPLICATION

Le fonctionnement est assez simple : il suffit d'abord de le valider dans le "mot de configuration" puis de sélectionner la période du WDT (avec ou sans prescaler), et enfin d'effacer le registre WDT périodiquement (avec l'instruction CLRWDT). Si le programme " plante ", il n'y a plus de remise à 0 et le bit TO s'active. Dans le mode SLEEP, le programme continue à l'instruction suivante. En revanche, en mode normal, cela génère un RESET.

#### LE PRESCALER

C'est un compteur binaire qui permet de diviser la fréquence de l'horloge. Suivant le ratio défini par l'utilisateur, la division s'étend de 1 à 128.

#### REGISTRES ASSOCIES

On indique ici que les bits utilisés.

ADR	NOM	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
2007H	CONFIG						WDTE		
81H	OPTION					PSA	PS2	PS1	PS0

### III.2.5.5 L'EEPROM

C'est une mémoire qui permet de stocker jusqu'à 64 octets de données de façon non volatile (les données restent en mémoire même en cas de coupure de l'alimentation).

Les données de cette EEPROM ne sont pas directement accessibles. Il faut alors utiliser l'adressage indirect à travers 4 registres (EECON1, EECON2, EEDATA, EEADR) du registre SFR (Registre de Fonction Spécial) :

#### EEDATA

C'est un registre de 8 bits utilisé pour lire ou écrire une donnée dans l'EEPROM. La donnée de la mémoire EEPROM est envoyée dans ce registre et y est maintenue jusqu'au prochain cycle de lecture ou d'écriture.

#### EEADR

C'est un registre de 8 bits qui définit l'adresse de l'octet à lire ou à écrire. Logiquement, il peut adresser jusqu'à 256 octets, mais le PIC16F84 n'en possède que 64. Aussi la valeur de EEADR sera-t-elle comprise entre 00h et 3Fh.

#### EECON1

C'est un registre de 8 bits dont seuls les 5 premiers, de poids faible, sont utilisés (bit 0 à bit 4).

#### EECON2

C'est un registre de 8 bits particulier qui est réservé uniquement au cycle d'écriture.

### REGISTRES ASSOCIES

On indique ici que les bits utilisés.

ADR	NOM	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
08H	EEDATA								
0BH	INTCON	GIE		TOIE			TOIF		
81H	OPTION			TOCS	TOSE	PSA	PS2	PS1	PS0

### III.2.6 Les interruptions

Le monochip traite des informations provenant de son environnement extérieur. Une première forme de saisie est la « scrutation » : Le microcontrôleur va scruter son environnement par l'intermédiaire des PORTS A et B. Ce principe est correct tant que les informations sont présentes sur les ports. Ce procédé ne convient plus dès lors qu'un événement externe non prévisible se produit : on appelle une interruption cette deuxième manière de traiter des informations.

Une interruption provoque l'arrêt du programme en cours pour permettre le passage au sous-programme d'interruption. Après la fin de cette routine, le microcontrôleur reprend le programme à l'endroit où il s'était arrêté. Comme une interruption est un événement asynchrone, il faut alors sauvegarder le contexte dans lequel se trouvait le microcontrôleur avant l'interruption, afin qu'il puisse reprendre correctement la continuité du programme.

Le PIC16F84 possède 5 sources d'interruption : 3 externes (RB0, RB4 à RB7 du PORTB et évidemment le RESET) et 2 internes (fin de cycle d'écriture de l'EEPROM et dépassement du registre TMR0).

On ne traitera pas ici du RESET, interruption particulière à laquelle un chapitre spécifique est consacré.

Chaque source d'interruption est constituée d'un drapeau et d'un bit de validation. Le premier mémorise l'événement, tandis que le deuxième, comme son nom l'indique, valide ou non la prise en compte de l'événement.

Un bit appelé GIE (Global Interrupt Enable) du registre INTCON autorise ou inhibe toutes les interruptions.

#### III.2.6.1 Déroulement d'une interruption

Lorsqu'un événement apparaît, le drapeau associé le mémorise en passant à l'état haut. Si le bit de validation approprié est à "1", il y a alors une interruption (on suppose que le bit GIE est à "1"). L'adresse contenue dans le PC (Program Counter) est d'abord chargée dans le STACK, puis remplacée par la valeur

0004H (adresse de la routine d'interruption). Le bit GIE passe à "0" pour inhiber toutes les interruptions qui seraient simultanément présentes. A la fin de la routine (instruction RETFIE), le bit GIE passe à l'état haut (autorisant ainsi un autre événement) et le contenu du PC est chargé avec la valeur du STACK.

### III.2.5.2 Les différentes sources d'interruption

#### RBO DU PORTB

Cette interruption appelée INT est gérée par les bits INTF et INTE qui sont respectivement le drapeau et le bit de validation.

Un bit supplémentaire appelé INTEDG du registre d'option détermine le front, montant (INTEDG=1) ou descendant (INTEDG=0), sur lequel sera déclenchée l'interruption.

#### RB4 A RB7 DU PORTB

Lorsque ces 4 broches (ou l'une d'elle) sont configurées en ENTREE, elles entraînent une interruption quand elles changent d'état. Les bits associés sont RBIF (drapeau) et RBIE (bit de validation).

#### TMR0

Quand le registre TMR0 passe de FFH à 00H (dépassement), le drapeau TOIF s'active. L'interruption est validée par le bit TOIE.

#### EEPROM

Après chaque cycle correct d'écriture, le drapeau EEIF (dans le registre EECON1) passe à "1". Le bit de validation EEIE se trouve dans le registre INTCON.

### III.2.7 La configuration de l'oscillateur

L'oscillateur définit la vitesse d'exécution des instructions. Pour des raisons de synchronisation, la fréquence interne de travail est celle de l'oscillateur externe divisé par 4. Par exemple, avec un quartz de 4 MHz, le cycle d'instruction sera d'une microseconde.

Le PIC16F84 peut fonctionner dans 4 configurations d'oscillateurs différents :

- LP (Low Power) : Avec une fréquence maximale de 200 KHz, il permet de réduire la consommation du microcontrôleur.
- XT : C'est le mode d'utilisation le plus utilisé. Sa fréquence de travail peut aller jusqu'à 4 MHz max.
- HS (High Speed) : C'est un mode de configuration pour l'utilisation de fréquence très haute (jusqu'à 10 MHz).
- RC : Dans ce mode, c'est un réseau Résistance/Condensateur qui réalise l'oscillateur.

Le mode de fonctionnement de l'oscillateur se configure lors de la programmation du monochip dans le "mot de configuration".

Le RESET est une interruption particulière qui initialise le microcontrôleur. Le PC (Program Counter) est dans ce cas chargé avec l'adresse 0000H (début du programme).

Un RESET est généré d'une manière soit matérielle, soit logicielle.

Comme le montre le schéma, il existe cinq sources qui génèrent un RESET a savoir :

Le MCLR (Master CLear) assure un RESET général du microcontrôleur quel que soit le mode de fonctionnement du PIC16F84 (normal ou SLEEP). Actif à l'état bas et prioritaire, il est le seul qui peut être provoqué de l'extérieur (broche 4).

Le WDT est un cas particulier. En mode normal, il génère un RESET ; en mode SLEEP, il réveille le microcontrôleur (le programme continue à l'instruction suivante). 2 bits particuliers TO et PD du registre de STATUS indiquent la source du déclenchement et permettent donc de réagir en fonction.

Le POR (Power On Reset) qui n'est actif uniquement lors de la mise sous tension du microcontrôleur. Il évite ainsi le rajout de composant externe (réseau RC) et assure une bonne initialisation.

L'OST (oscillator Start-up Timer) s'assure que l'oscillateur du microcontrôleur est stable.

Le PWRT (PoWeR up Timer) est composé d'un oscillateur interne RC et d'un compteur. Il maintient un RESET pendant 72 ms environ (la durée dépend essentiellement de la tension et de la température de fonctionnement).

#### CHRONOLOGIE D'UN RESET.

Plusieurs modules sont associés ensemble et permettent donc de réaliser une RAZ de bonne qualité avec peu de composants externes.

### III.2.8 Le mode SLEEP

#### III.2.8.1 Introduction

Certaines applications (télécommande, clavier, etc.) fonctionnent avec une

alimentation à pile pour être transportable. Dans ce cas, outre la programmation, le choix d'un microcontrôleur se fera sur sa consommation.

A noter que celle-ci dépend essentiellement de l'alimentation et de la fréquence de travail.

Pour faire face à ce problème d'énergie le PIC16F84 possède un mode de fonctionnement particulier : le mode SLEEP.

### III.2.8.2 Le mode SLEEP

En exécutant l'instruction SLEEP, le microcontrôleur passe à un état statique. En fait, l'oscillateur externe est arrêté. Le PC contient l'adresse de l'instruction suivante. Le WDT est mis à « 0 » (s'il est validé). Le bit PD passe à l'état bas et TO à l'état haut. Enfin, les PORTS conservent l'état dans lequel ils étaient avant l'instruction SLEEP.

### III.2.8.3 Le réveil du microcontrôleur

Le réveil consiste tout simplement à démarrer l'oscillateur. une fois le régime établie, le programme continue.

Lorsqu'il est dans le mode SLEEP, seules 5 sources peuvent relancer le PIC16F84 : MCLR, WDT, Interruption de fin de cycle d'écriture de l'EEPROM, l'Interruption RB0 et l'Interruption de changement d'état RB4 à RB7.

### III.2.9 le jeu d'instruction

Le PIC16F84 a un jeu d'instruction relativement limité mais possède une architecture interne RISC qui permet une programmation efficace et rapide (toutes les instructions, exceptées les sauts, s'exécute en un cycle d'horloge).

Instructions liées aux variables	ADDLW	ANDW	IORLW	MOVLW	RETLW	SUBLW	XORW	
Instructions liées aux données	ADDWF	ANDWF	CLRF	COMF	DECF	DECFSZ	INCF	INCFSZ
	IORWF	MOVF	MOVWF	NOP	RLF	RRF	SUBWF	XORWF
Instruction liées aux bits	BCF	BSF	BTFS	BTFS				
Autres instructions	SWAPF	CALL	GOTO	RETFIE	RETURN	CLRWT	SLEEP	

### III.3 Conclusion

C'est ainsi que nous avons présenté brièvement le PIC16F84 et ses différents registres, nous pensons qu'en utilisant ce microcontrôleur, le travail a été simplifié, car le PIC16F84 possède des ressources innombrables.

# Chapitre IV : Conception d'une PLC

## IV Conception d'une PLC

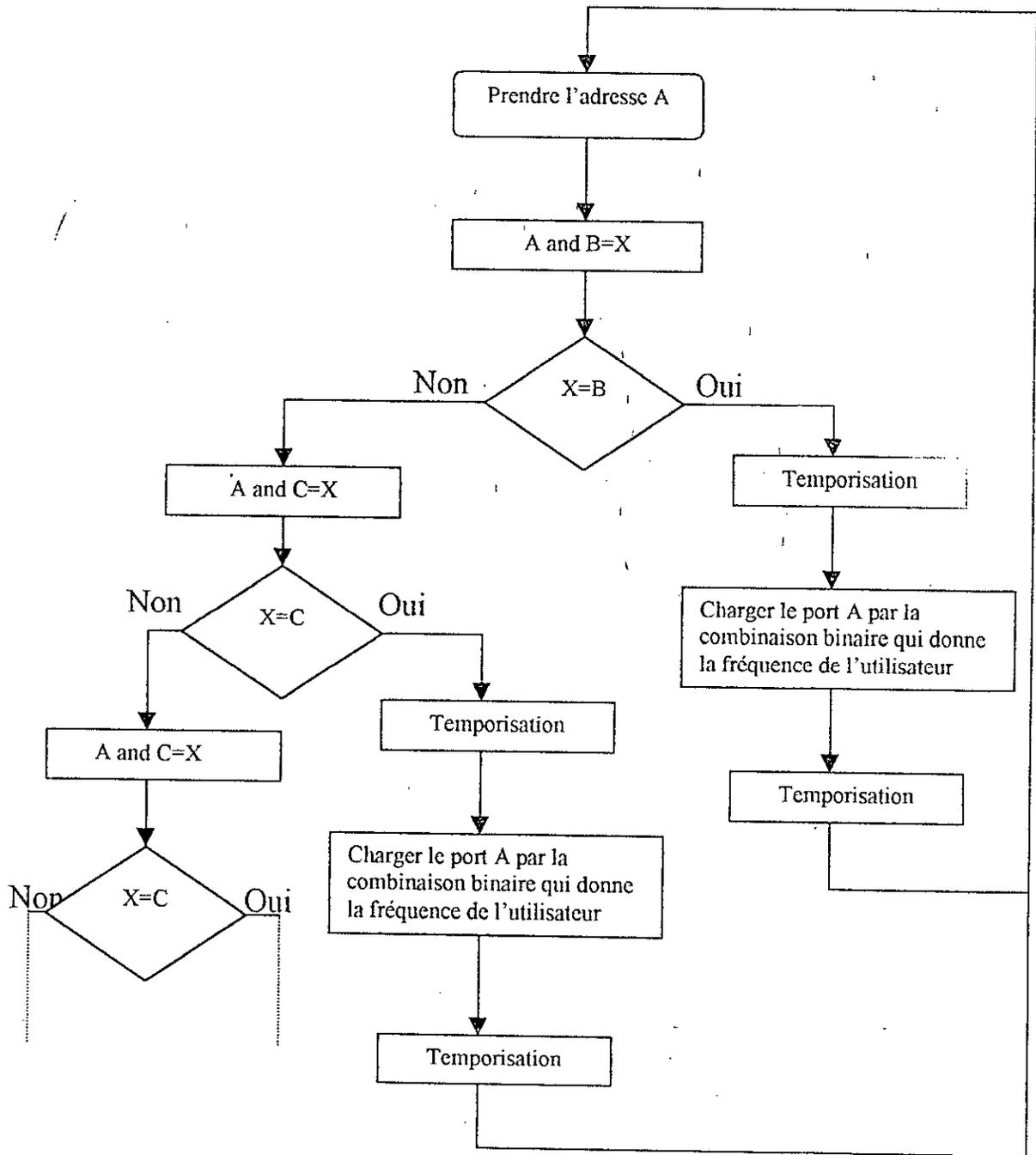
### IV.1 Introduction

Dans ce chapitre nous allons faire la conception d'une PLC, et de se fait /étudier les deux parties que sont le serveur et le client. Le serveur étant branché directement dans un transformateur, pour l'alimentation d'une cité par exemple. Le client quant à lui est branché sur une des trois phases disponibles, et reçoit le signal dans toutes les prises de courant.

### IV.2 Réception des données en provenance du web

Les paquets de données arrivent en série sur l'entrée de notre serveur coté net, ils ont une taille de 567 octets. L'adresse destination se trouve dans les 17ieme, 18ieme, 19ieme et 20ieme octet de ces paquets, donc pour les récupérer, on doit laisser passer les 16 premiers octets, puis récupérer l'adresse (Comme nous utilisons des adresses de classe C, car notre réalisation est plus orientée vers des réseaux avec un nombre restreint d'ordinateurs il nous suffit de tester le 17ieme octet seulement). Ces 16 octets ne doivent pas être perdus, alors on les met dans une pile FIFO(First In First out) pour assurer une temporisation qui permet de récupérer l'adresse sans perdre les données.

L'organigramme explique bien le rôle du microcontrolleur.



Nous allons utiliser une 74F433[NS], une pile FIFO de 64 mots de 4 bits qui se présente en boîtier DIP de 22 broches comme l'indique la figure suivante.

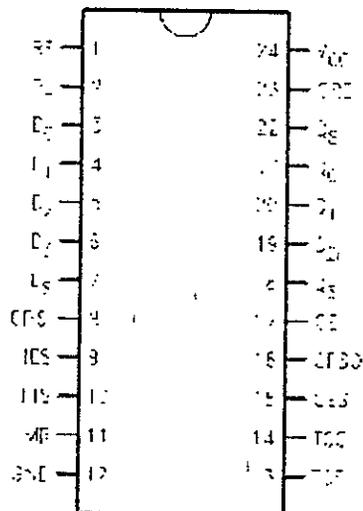


Figure IV.2 : La pile FIFO

Cette pile a une entrée série  $D_s$  et quatre entrées parallèles  $D_0, D_1, D_2, D_3$ . Elle possède aussi une sortie série  $Q_s$  et quatre sorties parallèles  $Q_0, Q_1, Q_2, Q_3$ . Les signaux de commande  $PL, CPSI, TOP, \dots$  permettent de configurer la pile.

Le brochage de la pile dans notre application est celui destiné à une extension de FIFO, donc, la pile que nous utilisons aura l'impression qu'il y a une autre pile reliée à elle, ce qui fait que le signal d'horloge «serial input clock» suffit pour charger les données dans la pile et de les faire sortir après une temporisation  $T$  tel que :

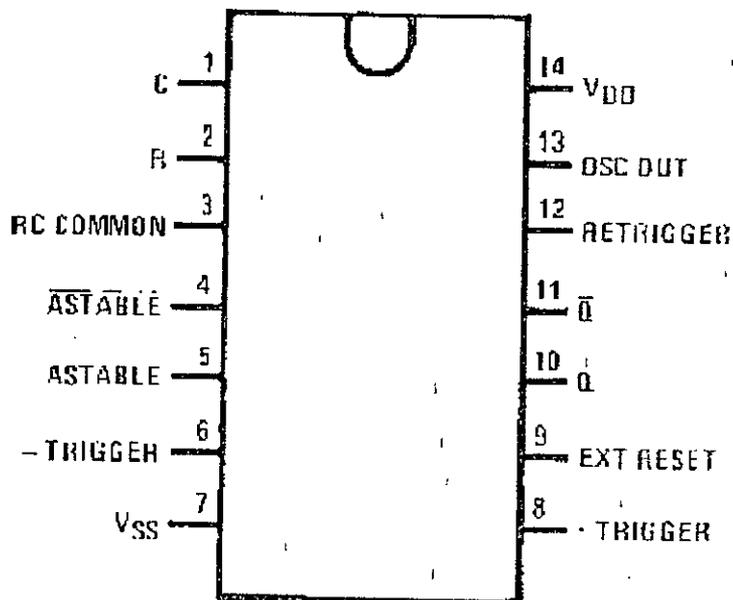
$$T = 64 * 4 / 250000 = 0.001024 \text{ (avec un débit de } 250000 \text{ bits/s)}$$

Le 1<sup>ier</sup> bit du 17<sup>ieme</sup> octet arrive sur l'entrée de la FIFO après 0.000512s alors que le 1<sup>ier</sup> bit du paquet sort de la FIFO après 0.0001024s ce qui nous laisse 0.000512s pour déterminer le destinataire.

Le signal d'horloge est généré par le CD40475[NS] qui est un astable intégré qui se présente en boîtier DIL 14 broche. Pour notre part, nous avons besoin d'une horloge à 250KHz synchronisé. La formule qui donne la fréquence du signal de sortie sur la broche 13 est :

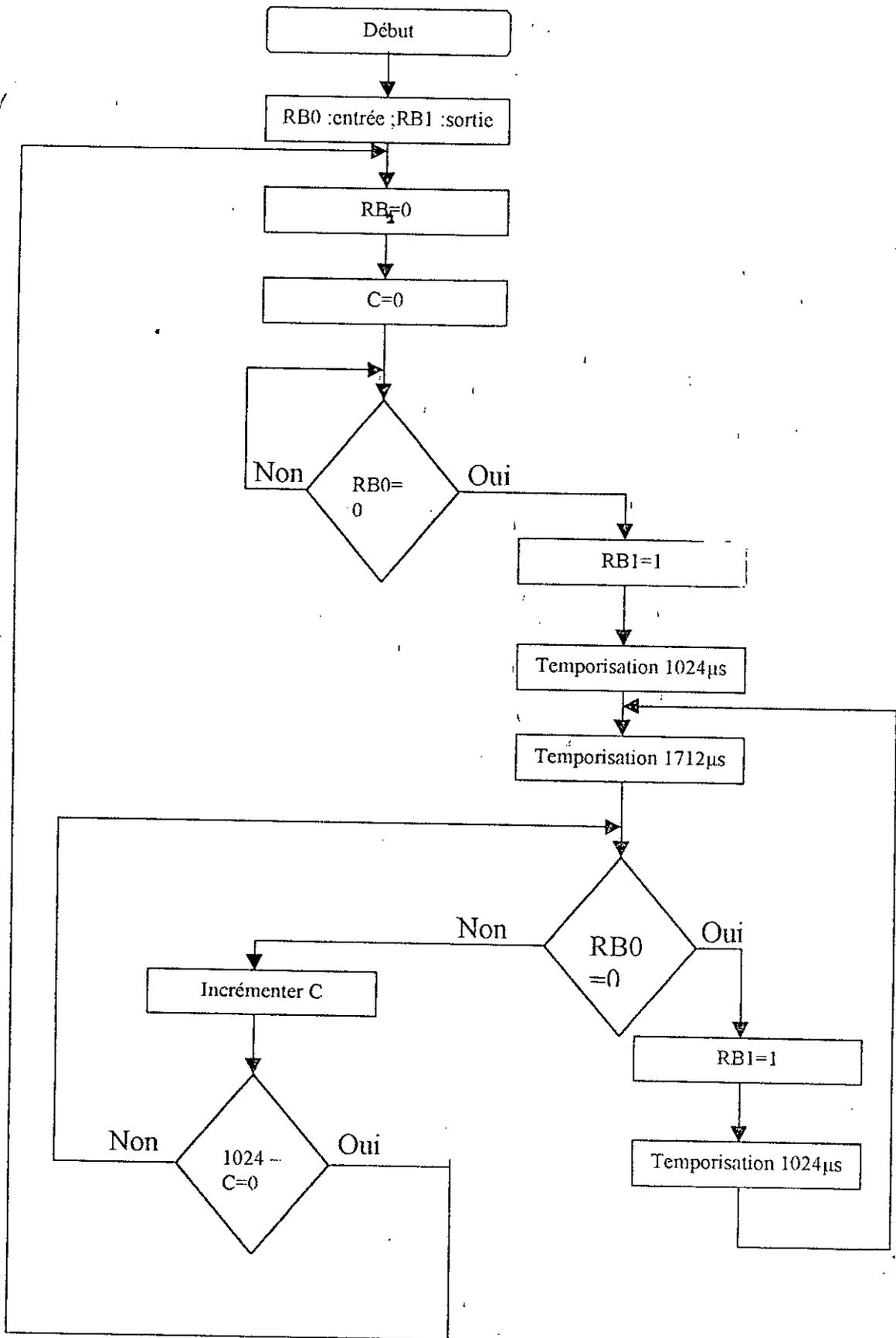
$$f = 1 / (2.2 RC) \quad \text{Avec } C = 200 \text{ pF et } R = 9.1 \text{ K}\Omega$$

Le schéma de brochage du CD4047 est représenté par la figure suivante :



La broche 5 de l'astable (synchro) est commandée par le microcontrôleur qui, détectant un « 0 » sur la broche « Serial Data IN » de la pile (car les 4 premiers bits du paquet TCP/IP représente la version qui est la version 4 donc 0100), envoie un « 1 » sur cette ligne pendant la durée du paquet ( $567 \cdot 8 / 250000 = 0.018144$  s). Après avoir passé les 567 octets, le microcontrôleur met la broche 5 à « 0 » et attend l'arrivée d'un autre paquet de données.

L'organigramme suivant explique le rôle du microcontrôleur.



Le C représente une constante qu'on utilise pour permettre au paquet de sortir entièrement de la pile.

Le schéma retenu pour réaliser cette opération est représenté dans la figure IV.1.

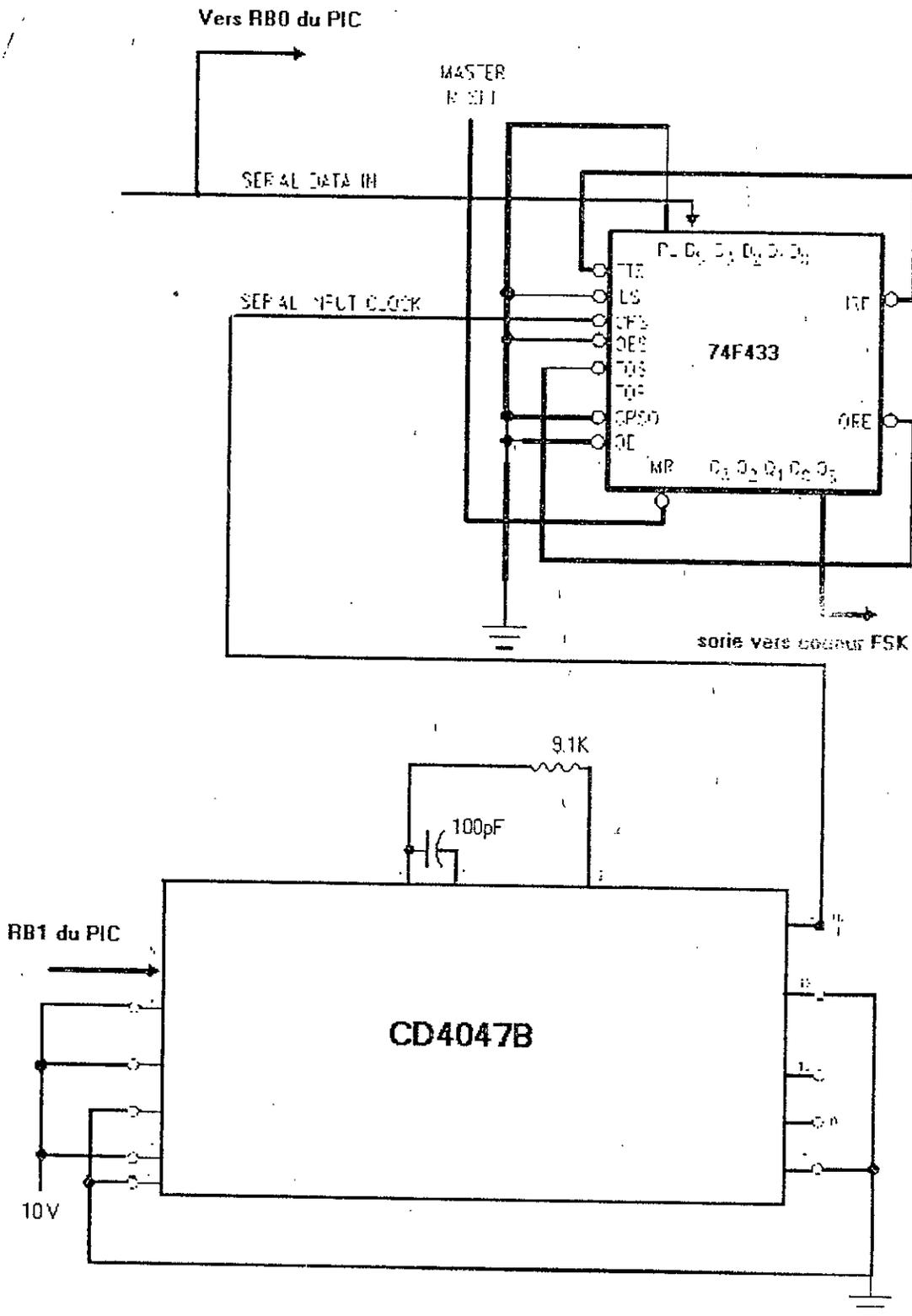


Figure IV.1

### IV.3 Codage du signal issu de la FIFO

A la sortie de la FIFO, un train binaire arrive avec un débit de 500000bits/s, pour transmettre ce signal on utilise la modulation FSK.

Pour réaliser le codage du signal numérique en un signal analogique, on utilise un VCO qui est le XR-2206[EX].

Le XR-2206 est un circuit intégré, en boîtier DIP 16 broches, comme le montre la figure suivante :

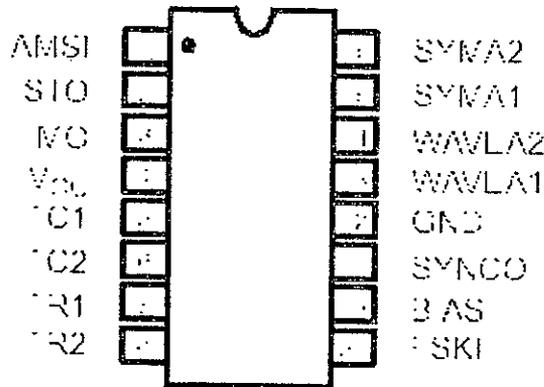


Figure IV.3 a : XR2206

Le XR-2206 travail dans la fréquence : 0.01Hz à 1MHz.

Ce circuit est idéal pour les communications, instrumentations et la génération de fonctions.

Le schéma retenu pour le codage FSK est le suivant :

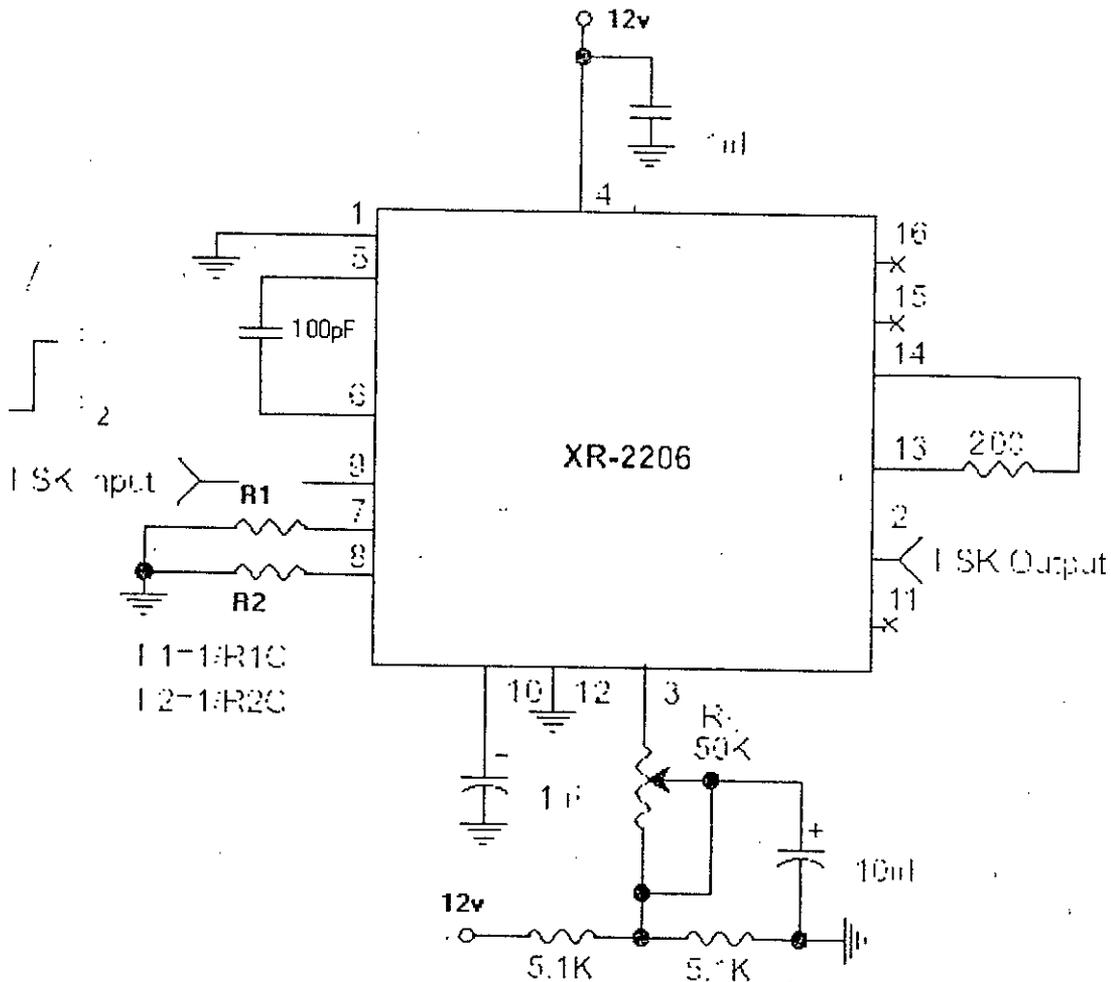


Figure IV.3.b : codeur FSK

Pour le codage d'un « 0 »  $F1=250\text{KHz}$   
 $F1=1/R1C$  avec  $R1=15\text{K}\Omega$  et  $C=270\text{pF}$   
 Pour le codage d'un « 1 »  $F2=500\text{KHz}$  :  
 $F2=1/R2C$  avec  $R2=7.5\text{K}\Omega$  et  $C=270\text{pF}$

A la sortie (FSK Output) on aura un signal composé de deux fréquences (500KHz et 250KHz) suivant le fait qu'il s'agisse d'un « 1 » ou d'un « 0 » à l'entrée.

#### IV.4 l'émetteur

##### IV.4.1 Introduction

Après la sortie des signaux du codeur, ces derniers modulent une porteuse générée par l'oscillateur. Cet oscillateur est commandé par un synthétiseur de fréquence pour constituer une boucle à verrouillage de phase.

#### IV.4.2 la boucle à verrouillage de phase [EP]

La boucle à verrouillage de phase (PLL : Phase Lock Loop) a été créée pour être un élément utile dans de nombreux systèmes de communication. On l'utilise principalement à deux fins différentes :

1. comme démodulateur pour suivre une phase ou une fréquence.
2. comme synchroniseur pour des signaux dont la fréquence peut varier dans le temps.

Le schéma bloc d'un tel asservissement est donné ci après.

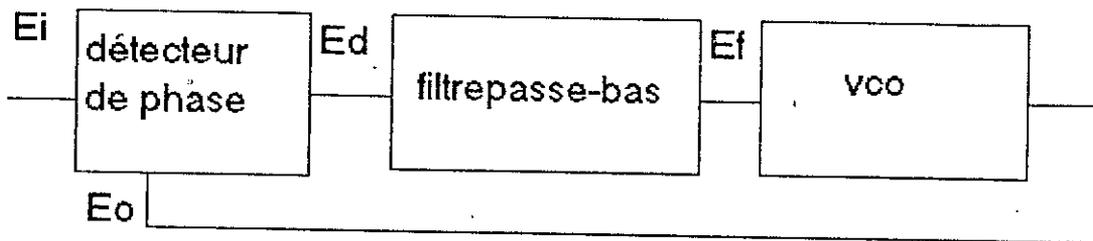


Figure IV.4.2 : boucle à verrouillage de phase

Un système asservi est un système bouclé dont la grandeur de sortie est asservie à celle de l'entrée. Selon que la sortie retenue est celle du filtre passe-bas ou bien celle du VCO, on réalise la fonction 1 ou bien la fonction 2 évoquées plus haut.

Pour mieux comprendre le fonctionnement de la boucle, considérons qu'initialement la boucle n'est pas verrouillée mais que les fréquences des signaux  $E_i$  et  $E_o$  soient très proches. Dans ces conditions,  $E_d$  est un battement dont la fréquence est la différence des fréquences de  $E_i$  et  $E_o$ . Ce signal est appliqué à l'entrée du VCO si sa fréquence est suffisamment faible pour traverser le filtre passe-bas. La fréquence instantanée du VCO varie donc, et, à un certain moment, si elle égale à celle du signal d'entrée, la boucle se verrouille. Si la fréquence de  $E_i$  varie, le déphasage est modifié instantanément ainsi que le niveau continu de  $E_d$ . Ce décalage fait varier d'autant la fréquence du VCO de façon à maintenir le verrouillage.

Dans l'application qui est faite ici, on introduit un diviseur par  $N$  programmable dans la boucle de retour de  $E_o$ . Ainsi, ce sont les signaux  $E_i$  et  $E_o/N$  qui sont verrouillés et on obtient donc un signal  $E_o$  qui est  $N$  fois plus rapide que le signal d'entrée  $E_i$ .

Le circuit MC145151-2 [MC] de MOTOROLA intègre un détecteur de phase ainsi qu'un diviseur de fréquence programmable ce qui nous permet de réaliser une boucle à verrouillage de phase avec à la sortie une fréquence variable.

IV.4.3 Le MC145151-2 [MC]

Le MC145151-2 est un circuit intégré en boîtier DIP de 28 broches.

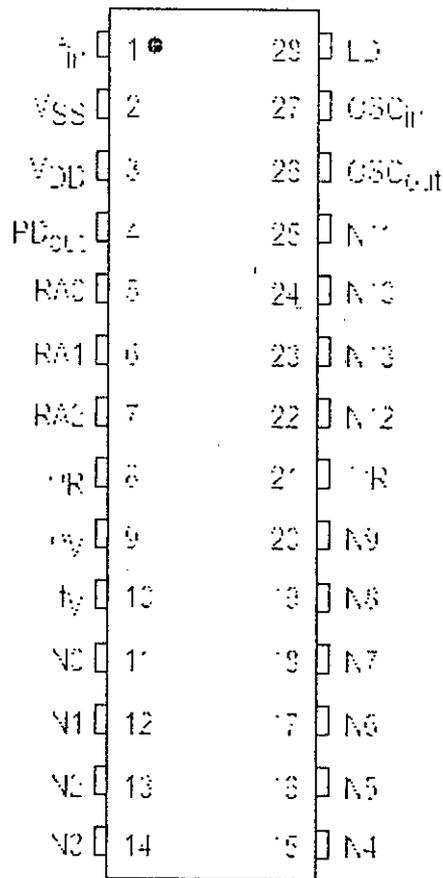


Figure IV.4.3 : MC145151-2

Il intègre un diviseur de fréquence programmable sur les broches N0-N13.

Les broches RA0-RA2 définissent une des huit valeurs possibles pour le diviseur de référence, suivant le tableau ci-dessous :

Reference Address Code			Total Divide Value
RA2	RA1	RA0	
.	.	.	1
.	.	.	1.5
.	.	.	2
.	.	.	3
.	.	.	4
.	.	.	6
.	.	.	12
.	.	.	24

Les broches OSCin et OSCout servent à connecter un quartz.

La broche PDout est la sortie du détecteur de phase.

La broche Fin est connectée à la sortie du VCO pour asservir sa fréquence (boucle de retour).

Le VCO que nous allons utiliser est le MAX038 [MX] de MAXIM, qui génère une des fréquences comprises entre : 0.1Hz et 20MHz.

/ Ce VCO est intégré dans un boîtier DIP à 20 broches.

La sélection de la forme du signal de sortie se fait selon le tableau suivant :

A0	A1	Forme du signal
X	1	Sinusoïdale
0	0	Carré
1	0	Triangulaire

Dans notre cas les broches N9, N10-N11-N12-N13, sont connectées au port A du PIC16F84 qui se charge de délivrer la combinaison binaire qui correspond à l'utilisateur.

Les autres broches, c'est à dire de N0-N8 sont mises à la masse.

La combinaison de RA0RA1RA2=111 (selon le tableau précédent) pour avoir la fréquence maximale divisible égale à 8.192MHz.

Le schéma qui réalise la boucle à verrouillage de phase est tiré du « MAX038 data sheet ».

Les broches A0, A1 sont connectés de façon à avoir un signal sinusoïdal, c'est à dire A0A1=X1.

IV.4.4 La modulation des signaux

Pour moduler les signaux (la porteuse et le signal issu du codeur FSK) on utilise le MC1496[MC] qui fait le produit de ces deux signaux, donc on aura en sortie la différence et la somme de leurs fréquences.

Le schéma suivant tiré du « MC1496 data sheet » donne le brochage du modulateur pour une modulation à porteuse supprimée :

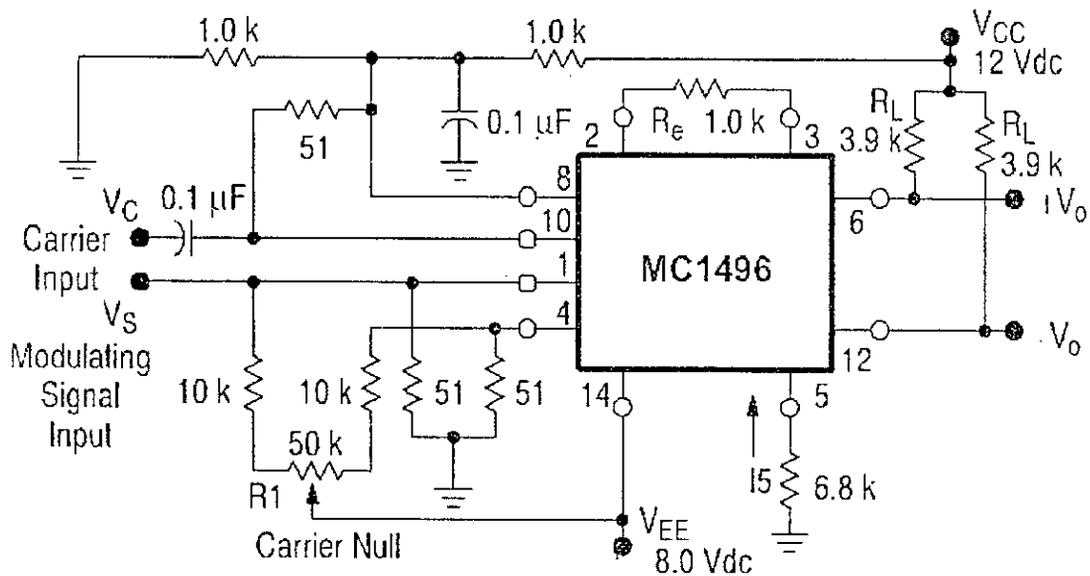


Figure IV.4.4.a : Modulateur

Dans la sortie du modulateur qui correspond à la broche 6 « +V0 » on met un filtre passe-bas de 50 MHz [MX].

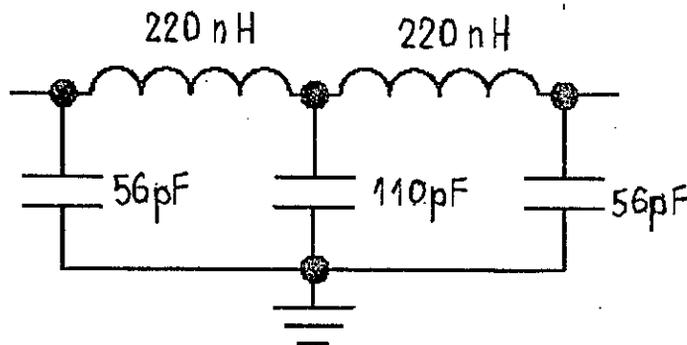


Figure IV.4.4.b : Filtre passe-bas 50MHz

La sortie de ce filtre attaque l'opto-coupleur:

IV.4.5 L'isolation

Le montage est isolé de la ligne électrique par l'opto-coupleur TIL194[TI] qui assure une isolation galvanique et électrique. La sortie de

ce dernier on amplifie le signal par le transistor 2N2222 puis on l'injecte dans le secteur de distribution par l'intermédiaire du condensateur.

Cette opération se fait selon le schéma suivant :

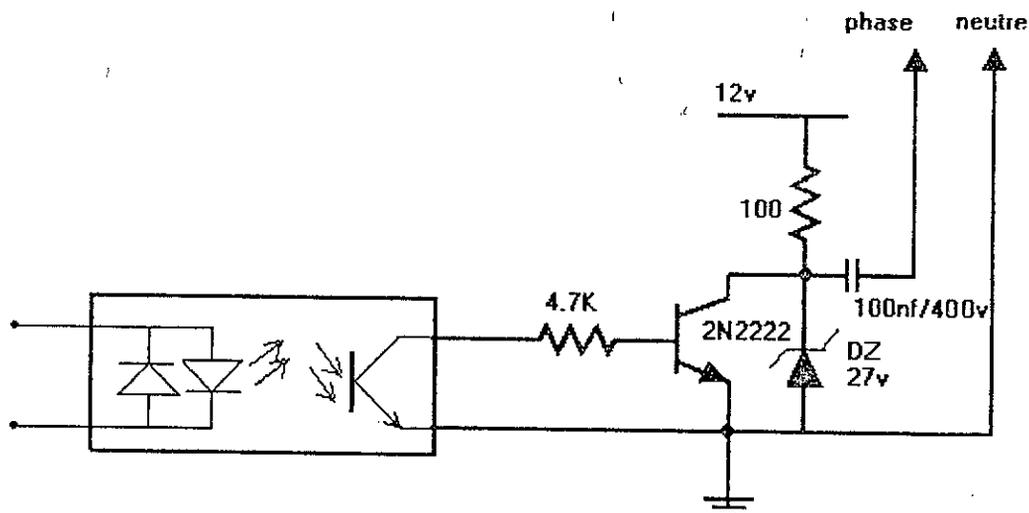


Figure IV.4.5 : L'isolation

### IV.5 Les récepteurs

La réception des données, que se soit coté client ou coté serveur est la même.

Tout d'abord on sépare le signal HF du 220v à l'aide d'un filtre passe-bande qui ne laisse passer que la bande de fréquence voulue, c'est à dire une bande de fréquence centrée sur la fréquence porteuse spécifique au client.

Pour chaque utilisateur on utilise deux filtres en cascade, l'un passe-haut, l'autre passe-bas.

Pour le premier utilisateur par exemple qui a une fréquence porteuse de 5MHz on utilise un filtre passe haut[HA] avec une fréquence de coupure de 4.5MHz, et un filtre passe bas[HA] avec une fréquence de 5.5MHz.

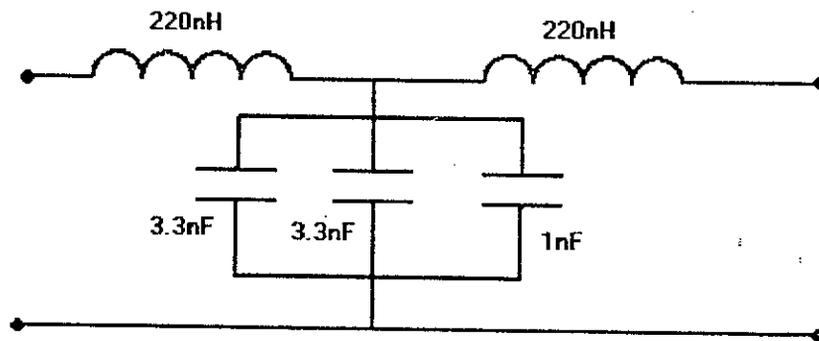


Figure IV.5.a : Filtre passe-bas

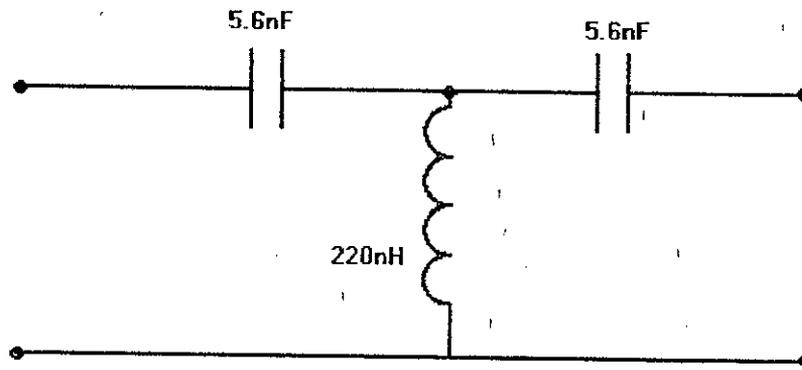


Figure IV.5.b : Filtre passe-haut

Après avoir filtré les données on les amplifie par l'intermédiaire de l'amplificateur opérationnel LF357[NS] qui possède une bande passante supérieure à 20MHz, cette opération est effectuée pour tout les utilisateurs. le schéma réalisant l'amplification est donné en figure IV.5.c.

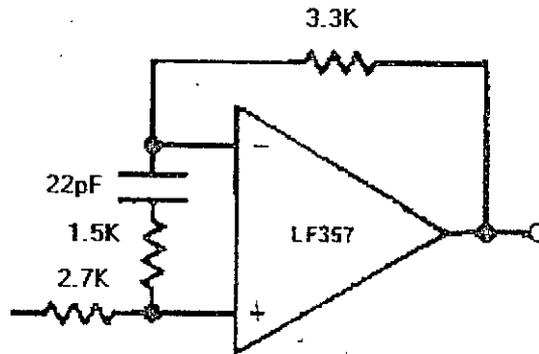


Figure IV.5.c : amplificateur

On démodule par le MC1496, qui nous délivre un signal qui bascule entre deux fréquences selon qu'il s'agira d'un « 1 » ou d'un « 0 ».

L'oscillateur local utilisé dans la démodulation a la même configuration que celui de l'émetteur sauf qu'on remplace le microcontrôleur par des interrupteurs (éventuellement des DIP switch), chaque utilisateur aura une configuration propre à lui qui définit sa fréquence porteuse.

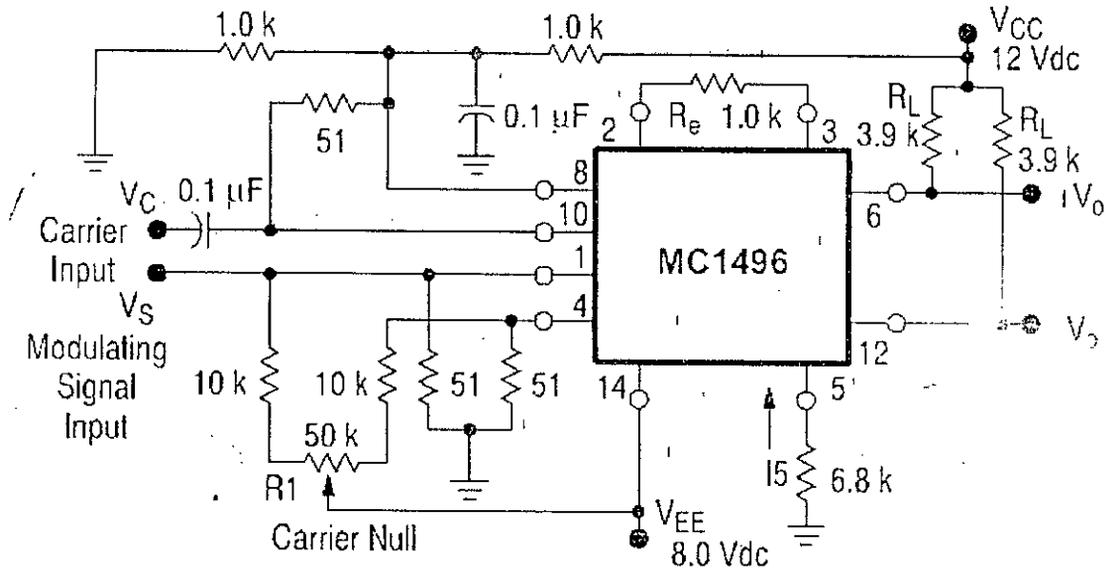


Figure IV.5.d : démodulateur

#### IV.6 Le décodeur FSK [NS]

Pour décoder les signaux FSK, on va utiliser un « tone decoder » qui est le NE567. Ce circuit intégré est branché de façon à avoir sur sa sortie un état bas en ayant un signal de 250KHz à l'entrée.

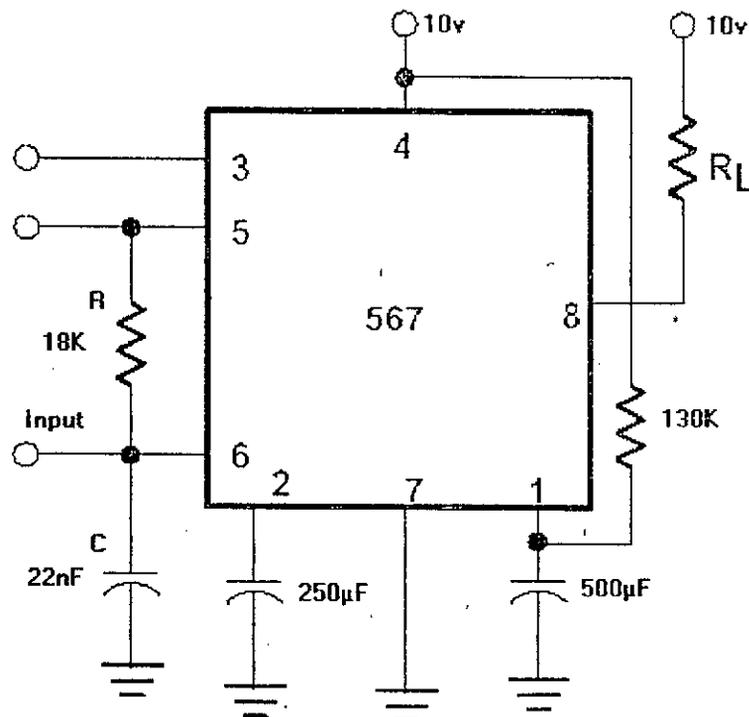


Figure IV.6 : décodeur FSK

La détermination de R et C se fait selon la formule :  $f_0 = 1/RC$

#### IV. 7 L'alimentation du montage

Dans notre montage, on utilise des tensions de : +12v, +10v, +5v, -5v, -8v. L'énergie est fournie par le réseau électrique, par l'intermédiaire d'un transformateur dont l'enroulement secondaire délivre des tensions alternatives de 12v et 5v. Des ponts de diodes redressent les deux demi-alternances, tandis que les capacités réalisent un premier filtrage.

Sur la sortie des régulateurs 7812, LM7912, 78L10, 7905, on relève des tensions continues et stabilisées à 12v, -8v, 10v, -5v, 5v respectivement.

Le choix des régulateurs se justifie par le fait qu'aucun de nos modules ne consomme plus que quelques mA.

Le schéma de cette alimentation apparaît alors dans la figure IV.7.

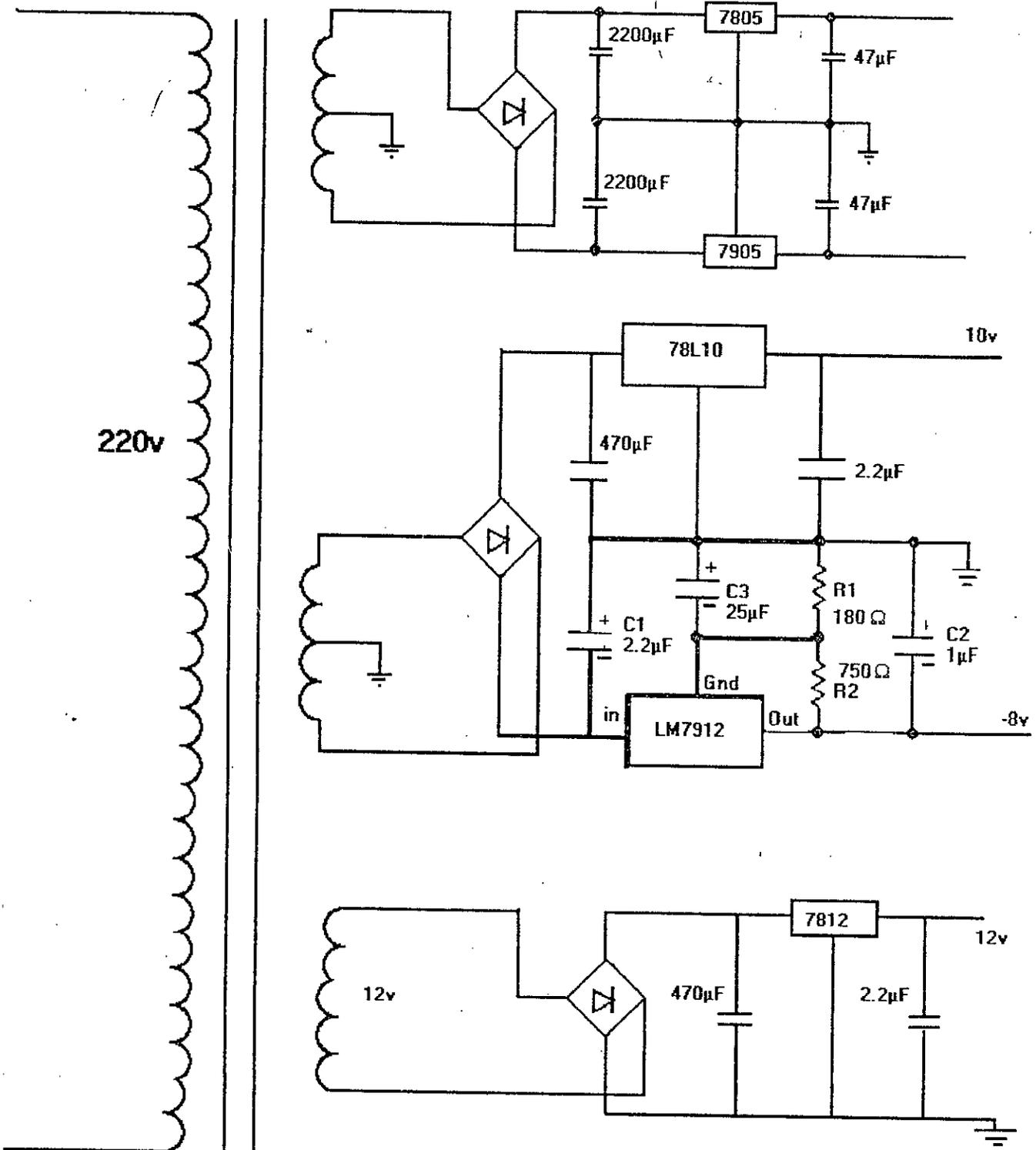


Figure IV.7 : l'alimentation

### IV.8 Conclusion

Nous avons exposé dans ce chapitre une méthode qui permet de réaliser le transfert de données sur le réseau électrique. Ayant été limité par les performances réduites du décodeur FSK qui a une fréquence maximale de 500KHz, le débit de notre montage est de 250000bits/s.

# **Conclusion Générale**

## Conclusion générale

Ce mémoire a présenté un nouveau moyen permettant le transfert de données sur le réseau électrique « PLC ».

L'approche utilisée pour le développement de la PLC est basée sur la notion d'adresses IP.

L'utilisation des microcontrôleurs est très avantageuse dans l'aiguillage et le traitement des données. De plus il rend possible la réduction de la taille de la carte, diminuant le nombre de composants augmentant ainsi la fiabilité.

Nous pensons que notre objectif qui est l'étude et la conception d'une PLC.

Nous tenons à souligner que notre projet a atteint son principal qui est l'étude et la conception d'une PLC.

Nous tenons à souligner, encore, que ce travail a bien enrichi nos connaissances dans plusieurs domaines, tel que la modulation numérique, les opto-coupleurs et le protocole Internet.

De plus, nous nous sommes familiarisés avec les microcontrôleurs PIC qui se sont avérés très performants.

Des améliorations de notre études sont possibles en ouvrant d'autres voies et d'autres perspectives :

La réalisation des cartes.

L'amélioration du débit.

L'interfaçage avec l'ordinateur (utilisation de l'USB).

Références bibliographiques

[EP]

Revue électronique pratique. N0 199-janvier 1996.

[EX]

EXAR [www.exar.com](http://www.exar.com)

[FR]

V.G.Fremaux [www.eisti.fr](http://www.eisti.fr)

[HA]

A.Hambley : an introduction to communications systems  
Computers science presse 1990.isbn 0-7167-8184-0

[HAD]

M.Haddadi cours d 'électronique générale 3ieme année

[MC]

Motorola [www.motorola.com](http://www.motorola.com)

[MI]

Microchip [www.microchip.com](http://www.microchip.com)

[MU]

[web2k.multimania.com](http://web2k.multimania.com)

[MX]

MAXIM [www.maxim.com](http://www.maxim.com)

[NS]

National Semiconductors Book

[PI]

J.F.Pillou [www.commentcamarche.com](http://www.commentcamarche.com)

[TI]

Texas Instrument [www.ti.com](http://www.ti.com)

[TS]

Tele Satellite [www.telesatellite.com](http://www.telesatellite.com)