

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : Electronique  
Option : Télécommunication

Réf:.....

**Mémoire de Fin d'Etudes  
En vue de l'obtention du diplôme:**

**MASTER**

*Thème*

**Etude et réalisation d'un kit de développement  
d'applications à base de pic de la famille 16Fxxx**

Présenté par :  
**BERNAOUI Sif Eddine**  
Soutenu le : 10 Juin 2012

Devant le jury composé de :

MCB. GUESBAYA Taher.  
MCB. BEKHOUCHE Khaled .  
MAA. TERGHINI Ouarda.

MCB  
MCB  
MAA

Président  
Encadreur  
Examineur

**Année universitaire : 2011 / 2012**

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : Electronique  
Option : Télécommunication

Mémoire de Fin d'Etudes  
En vue de l'obtention du diplôme:

**MASTER**

*Thème*

**Etude et réalisation d'un kit de développement d'applications à base  
de pic de la famille 16Fxxx.**

**Présenté par :**

**BERNAOUI Sif Eddine.**

**Avis favorable de l'encadreur :**

**MCB. BEKHOUCHE Khaled.**

**Avis favorable du Président du Jury**

**MCB. GUESBAYA Taher.**

**Cachet et signature**

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : Electronique  
Option : Télécommunication

## *Thème :*

**Etude et réalisation d'un kit de développement d'applications à base de pic de la famille 16Fxxx**

**Proposé par : MCB.BEKHOUCHE Khaled.**

**Dirigé par : MCB.BEKHOUCHE Khaled.**

### **RESUME**

Le microcontrôleur est d'une haute importance aujourd'hui dans notre vie comme des étudiants en électronique ou des électroniciens amateur ou professionnel, car on le trouve dans les différents domaines industriels et il soit une source pour les projets de fin d'étude.

Notre projet a pour but d'étudier et de réaliser un kit de développement d'application à base de pic de la famille 16fxxx et dans notre projet on' a choisit le pic 16F877, et on' a essayer de réaliser et de simuler des applications avec ce pic et d'autres modules et on' a programmée le pic a avec le logiciel Micro C, afin d'obtenir des bons résultats et on 'a programmée le pic en circuit sans l'enlever a partir d'un programmeur de pic en circuit.

## Remerciement

*Au terme de travail je tiens à remercier au premier lieu le dieu qui m'a donné la force d'achever cette étude.*

*Je remercie mes très chers parents pour leurs soutiens et leurs patiences.*

*Je remercie l'encadreur « Mr. BEKHOUCHE Khaled » pour son aide précieux, son engagement total pour réaliser et accomplir mon travail.*

*Je veux, à cette occasion avec le plus grand honneur à remercier sincèrement mes amis qui ont m'a donnée l'aide dans le temps ou j'étais en besoin Et à toutes mes enseignants de d'Université de MOHAMMED Khidher Biskra.*

*Enfin, je remercie vivement les membres de jury.*



# *Dédicace*

---

## *Dédicace*

*Au nom d'Allah, tout Miséricordieux le très Miséricordieux*

*Je viens de dédier*

*Le fruit de modeste effort à :*

*Cette qui était la source intarissable de tendresse, dans le paradis est au dessus  
de ses pieds, cette qui possède tout mon cœur « ma chère mère »*

*Qui je lui souhaite le guérison plus vite inchaalah.*

*Celui qui fait sons mieux pour moi mon suprême idéal*

*« Mon père »*

*A mon frère Adnane et sa petite famille et Walid et Iman et Amira et  
Khouloud et Chahinaz ce qui présentent mon appui dans la vie mes chers frères  
et mes sœurs et*

*Toute ma famille « BERNAOUI »*

*Et mes amis*

*« Hammadi, Mounir, Moussa, Jamel, Oussama, Tarek, Malek, Dassi,  
Salah, Youcef, Bilel, Walid »*

*« A la promotion 2012 »*

*« BERNOUI Sif Eddine »*



# Sommaire:

Introduction générale.....	01
<b>Chapitre I:Présentation de microcontrôleur 16F877:</b>	
I. Introduction.....	02
I.1. Définition de microcontrôleur.....	02
I.2.Les avantages de microcontrôleur.....	02
I.3. Le choix d'un microcontrôleur .....	03
I.4. Définition de pic.....	04
I.4.1. Structure minimale d'un PIC.....	04
I.4.2. Les différentes familles des PIC .....	05
I.4.3. Identification du PIC .....	05
I.5.Pic 16F877.....	06
I.5.1. Identification de Pic 16F877.....	06
I.5.2. Les particularités électrique.....	08
I.5.3. Les PINS de 16F877 .....	10
I.6. Les différentes mémoires de pic 16F877.....	12
I.6.1. La mémoire FLASH (programme) ..	12
I.6.2. La mémoire RAM.....	13
I.6.3. La mémoire EPROM interne .....	13
I.7.Les ports entrée/sortie de Pic 16F877.....	13
I.7.a. Le port A.....	13
I.7.b. le PORT B.....	14
I.7.c. Le PORT C .....	14
I.7.d. Le PORT D .....	15
I.7.e. Le PORT E .....	15
I.8. Les Timers .....	15
I.9. L'unité centrale .....	16
I.9.1. Organisation mémoire.....	16
I.10.Les interruptions.....	16
I.10.1. Présentation.....	16
I.10.2.Fonctionnement.....	17



## Sommaire

I.10.3. Mécanisme d'interruption sur les PICs.....	17
I.11. L'oscillateur .....	18
I.10.Le convertisseur.....	19
I.12.Conclusion.....	20
<b>Chapitre II: Le standard RS-232 .</b>	
II.1.Introduction.....	21
II.2. Les liaisons série .....	22
II.3. Transmission série asynchrone.....	22
II.4. La norme et la liaison RS-232.....	22
II.4.1. Aspect électrique .....	22
II.4.2. Aspect fonctionnel.....	23
II.5. La liaison RS 232 .....	23
II.5.1. Les caractéristiques de la liaison RS232.....	23
II.5.2. Paramètres techniques de la RS232.....	24
II.5.2. a. Liaison unidirectionnelle.....	24
II.5.2. b. Liaison half duplex .....	24
II.5.2. c. Liaison full duplex.....	25
II.6. Le protocole de transmission de la norme RS232.....	25
II.6.1. Handshake .....	26
II.7. La norme fixe des niveaux de tension d'une liaison RS232.....	27
II.8. Brochage d'une liaison RS232.....	27
II.9.Conclusion .....	29
<b>Chapitre III: Etude du kit du développement.</b>	
III.1.Introduction.....	30
III.2. Généralités sur les systèmes et les kits de développements.....	31
III .2.1.Les systèmes de développement.....	31
III.2.2.Base d'un système de développement.....	31
III.2.2.a. Côté logiciel.....	31
III.2.2.b. Côté matériel.....	32
III.2.3.Organisation du développement (matériel et programmation).....	32
III.3.Le kit de développement d'application à base de pic.....	34
III.3.1.Le contenu type d'un kit de développement.....	34
III.3.2.Les KITS de développement à base de Pic les plus connues.....	35



## Sommaire

III.3.2.a. La famille EASY pic de Mikroelektronika.....	35
III.3.2.b. La famille PICDEM.....	36
III.4.L'étude du kit de développement à base de pic 16F877.....	39
III.4.1.Schéma synoptique .....	39
III.4.2.Unité de traitement Pic 16F877.....	40
III.4.3.RESET .....	40
III.4.3.a. Principe de fonctionnement .....	40
III.4.4. Clavier.....	40
III.4.4.a. Principe de fonctionnement.....	41
III.4.4.b. Connexion du clavier sur la carte.....	42
III.4.5.Entrées et sortie simple.....	42
III.4.6.Afficheur LCD .....	43
III.4.6.a. Présentation .....	43
III.4.6.b. Principe de fonctionnement .....	43
III.4.6.c. Connexion de l'afficheur sur la carte .....	45
III.4.7. La liaison RS 232.....	45
III.4.7.a. Présentation .....	45
III.4.7.b.MAX232.....	45
III.4.7.c. Connexion de MAX232 sur la carte.....	47
III.4.8. Programmeur de pic.....	47
III.4.9.Schéma complet du kit du développement .....	48
III.5.Conclusion.....	49
<b>Chapitre IV: Réalisation de kit de développement.</b>	
IV.1.Introduction.....	50
IV.2. Réalisation software.....	51
IV.2.1. Présentation du logiciel « Mikro PRO for PIC » .....	51
IV.2.2. Création du projet en Mikro C PRO for PIC .....	51
IV.3..Présentation de l'Icprog .....	54
IV.3.1. Configuration de l'Icprog.....	54
IV.4. Réalisation du schéma du Kit .....	55
IV.4.1. Logiciel adopté .....	55
IV.4.2Réalisation avec logiciel Proteus .....	55
IV.4.2.a. Généralités sur Isis Proteus .....	55



## *Sommaire*

---

IV.4.2.b. La simulation numérique .....	55
IV.5. Réalisation du schéma de routage .....	57
IV.5.1. Logiciel utilisé .....	57
IV. 5.1. a. Ecran d'accueil.....	57
IV.5.1.b. Insertion des composantes .....	58
IV.6. Schéma de routage.....	58
IV.7. L'algorithme du programme.....	59
IV.7.1. Commande des entrées sorties simples.....	59
IV.7.1.a. commande de boutons poussoirs simple et leds .....	59
IV.7.1.b. commande de bouton poussoir intelligent.....	60
IV.7.2. commande de clavier Lcd .....	61
IV.7.3. commande de RS232.....	62
IV.8. Conclusion.....	63
Conclusion Générale.....	64
<b>Bibliographie.....</b>	<b>65</b>
<b>Annexe.....</b>	<b>67</b>



# *Introduction Générale*

---

## **Introduction Générale:**

Un kit de développement à base du PIC 16Fxxx est un système qui associe un ensemble des fonctions (Prog., RS232) gérées par un microcontrôleur de la famille PIC 16Fxxx.

Alors le but de notre projet est d'étudier et réaliser une Kit de développement, autour d'un PIC, qui permette de manipuler les applications de base et de réaliser des études des systèmes électroniques basées sur le microcontrôleur.

Le kit doit permettre de programmer le microcontrôleur qu'elle l'associe et de programmer d'autres microcontrôleurs en extension.



**CHAPITRE I :**  
**PRESENTATION DE**  
**PIC 16F877**

# **Chapitre 1: Présentation de microcontrôleur 16F877.**

---

## **I. Introduction:**

Ce chapitre est consacré à la présentation du microcontrôleur tout en illustrant ces différentes caractéristiques afin de mieux l'exploiter.

À cause de la difficulté ou la complexité en terme mieux de la réalisation pratique de certaines fonctions de traitement d'informations complexe à base les circuits classiques, tels que les circuits logiques et les circuits analogiques.

Il devient besoin d'un objet électronique qui peut faire ces fonctions rapidement, efficacement et avec le plus petit volume possible, c'est-à-dire un circuit intégré et ce circuit est le microcontrôleur.

### **I.1. Définition de microcontrôleur:[5]**

Un microcontrôleur se présente comme étant une unité de traitement de l'information de type microprocesseur contenant tous les composants d'un système informatique, à savoir microprocesseur, des mémoires et des périphériques (ports, timers, convertisseurs...). Chaque fabricant a sa ou ses familles de microcontrôleurs.

Une famille se caractérise par un noyau commun (le microprocesseur, le jeu d'instruction...). Ainsi les fabricants peuvent présenter un grand nombre de pins qui s'adaptent plus au moins à certaines tâches. Mais un programmeur connaissant une famille n'a pas besoin d'apprendre à utiliser chaque membre, il lui faut connaître juste ces différences par rapport au père de la famille. Ces différences sont souvent, la taille des mémoires, la présence ou l'absence des périphériques et leurs nombres

### **I.2. Les avantages du microcontrôleur:**

L'utilisation des microcontrôleurs pour les circuits programmables à plusieurs points forts et bien réels. Il suffit pour s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants de circuits intégrés en ce domaine depuis quelques années.

Nous allons voir que le nombre d'entre eux découle du simple sens.

- Tout d'abord, un microcontrôleur intègre dans un seul et même boîtier ce qui, avant nécessitait une dizaine d'éléments séparés. Il résulte donc une diminution évidente de l'encombrement de matériel et de circuit imprimé
- Cette intégration a aussi comme conséquence immédiate de simplifier le tracé du circuit imprimé puisqu'il n'est plus nécessaire de véhiculer des bus d'adresses et de donnée d'un composant à un autre.



# **Chapitre 1: Présentation de microcontrôleur 16F877.**

---

- L'augmentation de la fiabilité du système puisque, le nombre des composants diminuant, le nombre des connexions composants/supports ou composants/circuits imprimer diminue .
- Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux
- Moins cher que les autres composants qu'il remplace.
- Diminuer les coûts de main d'œuvre.
- Réalisation des applications non réalisables avec d'autres composants.

## **I.3. Le choix d'un microcontrôleur :[6][8]**

La majorité des grands fabricants de circuits intégrés dispose aujourd'hui de plusieurs gammes de microcontrôleurs qui, si l'on en croit leurs publicités, sont toutes plus performantes les unes que les autres.

On peut donc légitimement se demander quelle famille de circuits choisir et c'est d'ailleurs la question qui taraude généralement un industriel qui doit développer une application.

En ce qui nous concerne, nous ne sommes pas des industriels, ce qui nous simplifie quelque peu le travail. En effet, les seuls critères principaux que nous devons retenir sont les suivants :

- Le ou les circuits de la famille doivent être facilement disponibles sur le marché amateur.
- Le prix des circuits doit être à la portée de toutes les bourses.
- La programmation de la mémoire morte interne (celle qui contient le programme) doit être facile.
- Et enfin, les outils développement (nous verrons dans un instant de quoi il s'agit) doivent être aussi peu coûteux que possible.

A l'heure actuelle, les circuits qui répondent le mieux à ces critères sont les microcontrôleurs de la famille PIC de Microchip.

Comble de chance, ces circuits connaissent actuellement un succès que l'on peut, sans exagérer, qualifier de planétaire et sont très largement utilisés dans l'industrie.

En les choisissant nous bénéficions donc des retombées que cela implique avec, principalement, un très large choix de références, une excellente disponibilité et un très faible prix unitaire.

Dans ce chapitre, on va étudier le microcontrôleur, car il est l'élément de base de notre kit de développement.



# Chapitre 1: Présentation de microcontrôleur 16F877.

---

Peut-être est-il bon de ne pas acheter une voiture avant d'avoir son permis et, avant de traiter de programmation et d'outils de développement, vaut-il mieux définir clairement ce dont on va parler.

Un microcontrôleur est donc un circuit intégré qui contient en interne, c'est-à-dire dans un seul et même boîtier, l'équivalent de la structure complète d'un micro-ordinateur.

Et donc, son architecture interne se contient:

\*UAL (Unité Arithmétique et Logique).

\*Port d'E/S.

\*Interfaces de communications série.

\*Interfaces d'E/S analogique.

\* Timers et horloge temps réels.

## I.4. Définition de pic

Les microcontrôleurs sont aujourd'hui implantés dans la plupart des applications de grand public ou professionnelles, il existe plusieurs familles. La société Américaine Microchip Technologie a mis au point dans les années 90 un microcontrôleur CMOS : le PIC (Periphior Interface). Ce composant encore très utilisé à l'heure actuelle, est un compromis entre simplicité d'emploi, rapidité et prix de revient

### I.4.1. Structure minimale d'un PIC: [9]

Les PICs font partie de la famille des microcontrôleurs. Selon le type de PIC utilisé, on trouvera en interne un certain nombre de périphériques avec des caractéristiques distinctes, ainsi que des capacités mémoire différentes (RAM, EEPROM, mémoire programme FLASH ou EPROM ou OTP).

Un PIC est généralement constitué :

- o Une mémoire programme de type EEPROM flash dont laquelle on trouve le code binaire des instructions que doit réaliser le microcontrôleur .
- o Une RAM donnée de 368 octets (Random Access Memory) pour sauvegarder temporairement des données.
- o Cinq ports d'entrée sortie, A (6 bits), B (8 bits), C (8 bits), D (8 bits) et E(3bits) pour pouvoir dialoguer avec l'extérieur du microcontrôleur.
- o Convertisseur analogique numérique
- o USART, port série universel, mode asynchrone (RS 232) et mode asynchrone
- o SSP, port série synchrone supportant I2C



# Chapitre 1: Présentation de microcontrôleur 16F877.

---

- Trois TIMERS avec leurs prescalers, TMR0, TMR1, TMR2
- Deux modules de comparaison et capture CCP1 et CCP2
- Un chien de garde
- Une unité arithmétique et logique (ALU), chargée d'effectuer toutes les opérations arithmétiques de base (addition, soustraction etc....) ainsi que les opérations logiques de base
- Des registres capteurs de programme (CCP) qui pointe l'adresse mémoire contenant l'instruction courante à réaliser par le microcontrôleur.
- Des registres système
- Des registres utilisateur
- D'une horloge système qui permettra de cadencer tous les échanges internes ou externes au microcontrôleur.

## I.4.2. Les différentes familles des PICs :

Il y en a trois grandes familles de PICs:

- La famille Base Line, qui utilise des mots d'instructions de 12 bits .
- La famille Mid-Range, qui utilise des mots de 14 bits (et dont font partie la 16F84 et 16F876).
- La famille High-End, qui utilise des mots de 16 bits .

Toutes les PICs Mid-Range ont un jeu de 35 instructions, stockent chaque instruction dans un seul mot de programme, et exécutent chaque instruction (sauf les sauts) en un cycle. On atteint donc des très grandes vitesses, et les instructions sont de plus très rapidement assimilées.

## I.4.3. Identification du PIC :[4]

Pour identifier une PIC, on utilise simplement son numéro.

Les 2 premiers chiffres indiquent la catégorie de la PIC, 16 indique une PIC Mid-Range .

Vient ensuite parfois une lettre L: Celle-ci indique que la PIC peut fonctionner avec une plage de tension beaucoup plus tolérante.

Ensuite, on trouve:

- ◆ C indique que la mémoire programme est une EPROM ou plus rarement une EEPROM.
- ◆ CR pour indiquer une mémoire de type ROM.



# Chapitre 1: Présentation de microcontrôleur 16F877.

- ◆ Ou F pour indiquer une mémoire de type FLASH.

La PIC peut recevoir une fréquence d'horloge maximale indiquée sur les boîtiers(XX ) par exemple -04 pour une 4MHz.

Les PICs sont des composants STATIQUES, c'est à dire que la fréquence d'horloge peut être abaissée jusqu'à l'arrêt complet sans perte de données et sans dysfonctionnement.

Ceci par opposition aux composants DYNAMIQUE, donc la fréquence d'horloge doit rester dans des limites précises.

Et voici le tableau I.1. qui montre les différences entre quelques PICs.

	Mem prog en octets	RAM en octets	EEPROM	Fréquence max en MHZ	E/S	Boitier
<b>12C508</b>	512x12	25	-	4	6	8broches
<b>16C72A</b>	2048x14	128	-	20	22	28broches
<b>16F84</b>	1024x14	68	64	20	13	18broches
<b>16F628</b>	2048x14	224	128	20	16	18broches
<b>16F876</b>	8192x14	368	256	20	22	28broches
<b>16F877</b>	8192x14	368	256	20	33	40broches

Tableau I.1. Tableau comparatif.

## I.5. Pic 16F877:[ 4]

Le pic 16F877 est un circuit intègre contenu dans un boîtier nommer « DIL 40 », il présente 40 broches, 20 de chaque côté. Les broches sont virtuellement numérotées de 1 à 40. La 1ere broche est placé dans le coin situé à gauche de l'encoche de repérage.

### I.5.1. Identification de Pic 16F877:

Le choix d'un microcontrôleur est primordial car c'est de que dépendent en grande partie des performances, la taille, la facilité d'utilisation et le prix du montage.

Le PIC 16F877 possède en plus des instructions très puissantes donc un programme à développer réduit, surtout lorsqu'on utilise le logiciel de programmation micro pascal qui possède un nombre important de procédures et fonctions prédéfini et dédié au PIC 16F877.

En fait la cause principale du choix de ce type de microcontrôleur est qu'il dispose de l'option du convertisseur A/D pour satisfaire coté acquisition, aussi la possibilité de l'adaptation au protocole I2C et la liaison RS232 mais aussi le nombre de ports d'entrées /sorties nous convient.

Le PIC 16F877, dispose de 33 lignes d'entrées/sorties reparties en cinq ports:



# Chapitre 1: Présentation de microcontrôleur 16F877.

- Un port A de 6 bits (RA0 à RA5).
- Un port B de 8 bits (RB0 à RB7).
- Un port C de 8 bits (RC0 à RC7).
- Un port D de 8 bits (RD0 à RD7).
- Un port E de 3 bits (RE0 à RE3).

Et d'abord voici (figure I.1) le schéma fonctionnel de pic 16F877.

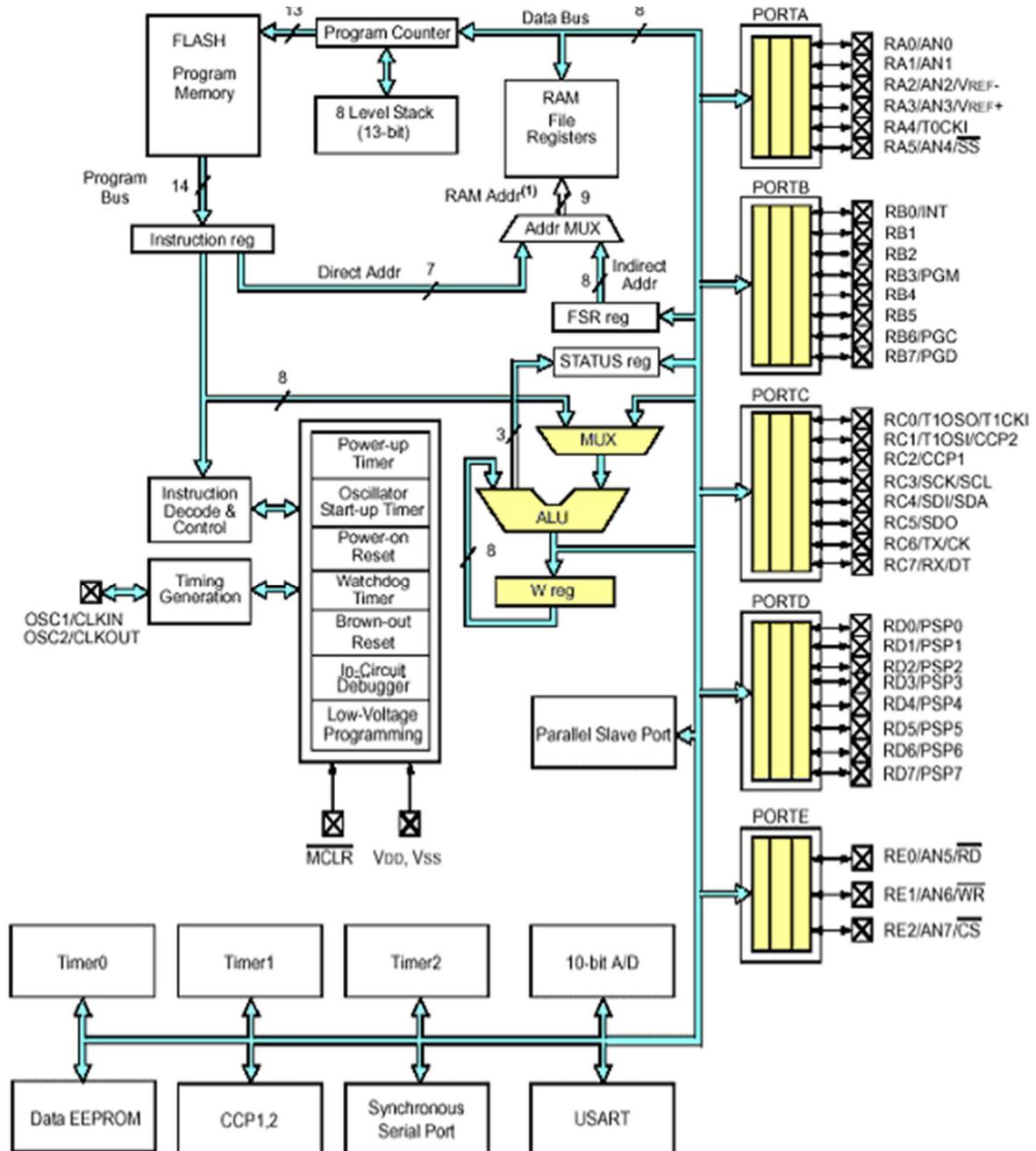


Figure I.1 : Schéma fonctionnel du PIC 16F877.



# Chapitre 1: Présentation de microcontrôleur 16F877.

Ou alors la figure. I.2.dans la page suivante qui nous présente un schéma fonctionnel simplifié de pic 16F877.

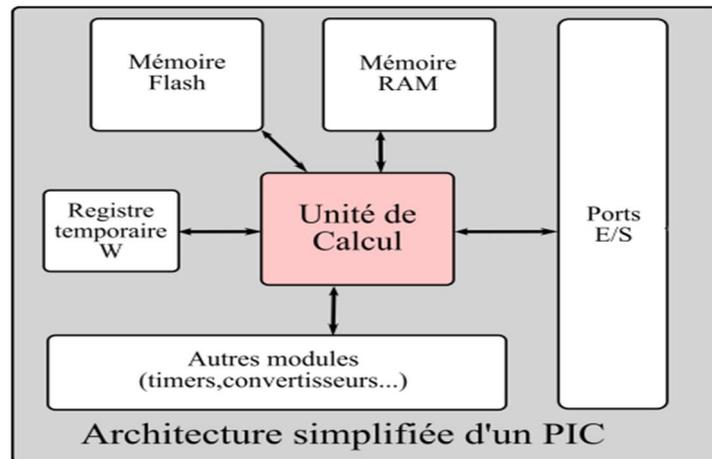


Figure I.2: Schéma fonctionnel simplifier de pic 16F877.

## I.5.2. Les particularités électriques: [4]

Vous constatez que sur le schéma concernant le 16F877, nous avons deux connexions ( $V_{ss}$ ) qui sont reliée à la masse .En fait, en interne, ces pins sont interconnectés. La présence de deux pins s'explique par une raison de dissipation thermique. Les courants véhiculés dans le PIC sont loin d'être négligeables du fait des nombreuses Entrées/sorties disponibles.

Le constructeur a donc décidé de répartir les courants en plaçant deux pins pour l'alimentation  $V_{ss}$ , bien évidemment, pour les mêmes raisons, ces pins sont situés de part et d'autre du PIC, et en positions relativement centrales. Le brochage de pins de pic16F877 est indiqué dans la figure I.3.

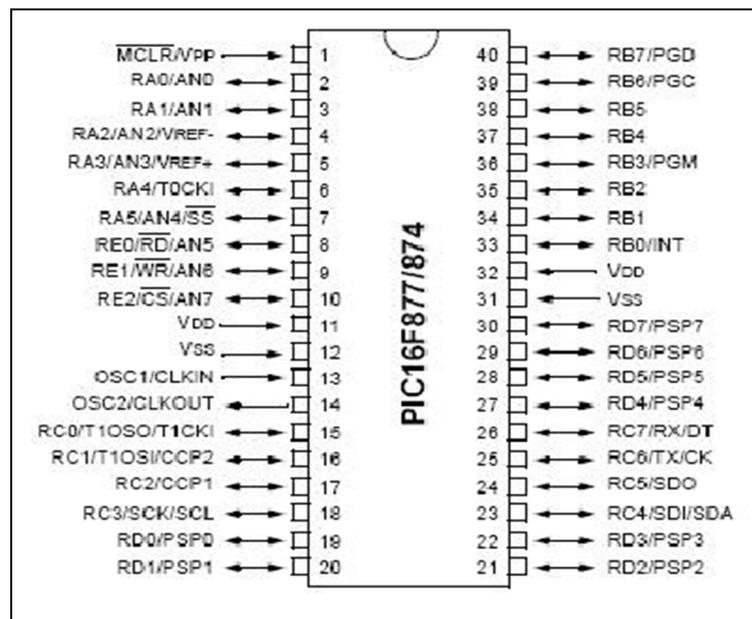


Figure. I. 3: Brochage du PIC16f877.



# Chapitre 1: Présentation de microcontrôleur 16F877.

Ce PIC dispose de 35 instructions de base et de 4 sources d'interruptions:

- Interruption externe commune avec la broche RB0.
- Interruption due au TIMER.
- Interruption sur changement d'état des broches de port RB0 à RB1.
- Interruption de fin d'écriture en EEPROM.

En fait ce choix est imposé dans le cahier de charge .

Nous allons maintenant s'intéresser à la structure interne du PIC 16F877, avec lequel nous avons travaillé.

Le 16F877 est un microcontrôleur de MICROCHIP, fait partie intégrante de la famille des Mid Range (16) dont la mémoire programme est de type flash (F) de type 877 et capable d'accepter une fréquence d'horloge maximale de 4Mhz.

Et voici donc le tableau I.2.

PIC	FLASH	RAM	EPROM	A/D	I/O	PORT//	PORT Série
16F877	8K	368 Octets	258	8	33	PSP	USART/ MSSP

Tableau. I.2. Caractéristique de PIC 16F877.

Nous voyons donc que la mémoire RAM disponible du 16F877 est de 368 octets. Elle est répartie de la manière suivante :

- ✓ 80 octets en banques 0, adresse 0X20 à 0X6F
- ✓ 80 octets en banque 1, adresse 0XA0 à 0XEF
- ✓ 96 octets en banque 2, adresse 0X110 à 0X16F
- ✓ 96 octets en banque 3, adresse 0X190 à 0X1EF
- ✓ 16 octets communs aux 4 banques, soit 0x70 à 0x7F= 0xf0 à 0xFF = 0x 170 à 0x17F == 0x1F à 0x1FF.



## Chapitre 1: Présentation de microcontrôleur 16F877.

### I.5.3. Les PINS de 16F877: [7][4][5]

Voici le tableau. I.3. qui montre la destination des pins de pic 16F877.

<b>BROCHES</b>	<b>DESTINATION</b>
1	MCLR/Vpp/THV
2	RA0/AN0
3	RA1/AN1
4	RA2/AN2VREF-
5	RA3/AN3/VREF+
6	RA4/TOCKI
7	RA5/AN4/SS
8	RE0/AN5/RD
9	RE1/AN6/WR
10	RE2/AN7/CS
11	VCC
12	GND
13	OSC1/CLKIN
14	OSC2/CLKOUT
15	RC0/T1OSO/T1CKI
16	RC1/T1OSI/CCP2
17	RC2/CCP1
18	RC3/SCK/SCL
19	RC4/SDI/SDA
20	RC5/SDO
21	RC6/TX/CK
22	RC7/RX/DT
23	RD0/PSP0
24	RD1/PSP1
25	RD2/PSP2
26	RD3/PSP3
27	RD4/PSP4
28	RD5/PSP5
29	RD6/PSP6
30	RD7/PSP7
31	GND
32	VCC
33	RB0/INT
34	RB1
35	RB2
36	RB3/PGM
37	RB4
38	RB5
39	RB6/PGC
40	RB7/PGD

Tableau. I.3. les pins de PIC16F877.



## ***Chapitre 1: Présentation de microcontrôleur 16F877.***

---

- MCLR:

Cette broche sert à initialiser le microcontrôleur .

Le microcontrôleur dispose de plusieurs sources de RESET :

- ✓ POR .
- ✓ EXTERNAL RESET
- ✓ WDT
- ✓ BOR
- ✓ POR: (POWER ON RESET) Mise sous tension .

Un front montant sur MCLR déclenche l'initialisation du microcontrôleur. Le temps nécessaire est au minimum de 72ms et au maximum de  $72ms + 1024 * T_{osc}$ . Le microcontrôleur dispos en interne d'un circuit de détection de niveau, quand la tension VDD est comprise entre 1.2V et 1.7V, il démarre une procédure d'initialisation .

Cette broche peut être simplement reliée à VDD si on n'a pas besoin de RESET externe. Par contre si on souhaite implanter un bouton de remise à zéro, on pourra câbler un simple réseau RC sur la broche MCLR .

Remarque importante : On peut se passer de circuit RC à la seule condition que le temps de montée de VDD soit suffisamment rapide (au minimum 50mV/ms). Si le temps de montée est inférieur à 50mV/ms, il faut rajouter un réseau RC .

- EXTERNAL RESET: (Mise à l'état bas de MCLR). Remise à zéro extérieure. Il faut appliquer un niveau bas sur l'entrée RESET pendant au moins  $2\mu S$  pour que l'Initialisation soit prise en compte .

- WDT: Chien de garde .

Si le WDT arrive à la fin du temps de garde sans avoir été rafraîchi il y aura alors une initialisation du microcontrôleur .

- BOR: Baisse de l'alimentation .

Si la tension VDD chute en dessous de 4V pendant  $100\mu S$  au moins, le microcontrôleur peut générer un RESET .

- Oscillateur : OSC1 et OSC2 ou CLKIN et CLOUT .Ces broches permettent de faire fonctionner l'oscillateur interne du PIC .

On peut utiliser 3 types d'oscillateurs :

- Un quartz ou résonateur céramique



# Chapitre 1: Présentation de microcontrôleur 16F877.

-Un oscillateur externe

-Un réseau RC

•Alimentation : VDD et VSS .

Ce sont les broches d'alimentation du circuit. Les tensions qui peuvent être appliquées vont :

-De 4,5V à 6V pour la gamme standard F .

-De 2 à 6V pour la gamme étendue LF .

L'intensité du courant consommé peut aller de 1 $\mu$ A à 10mA .

La consommation du microcontrôleur sera fonction de :

-La tension d'alimentation .

-La fréquence interne .

-Le mode de fonctionnement .

De plus ces bornes doivent être découplées par deux condensateurs :

1 $\mu$ F électrolytique .

10nF céramique .

•L'Interruption : RBO/INT .

Cette broche à une double fonction elle peut être utilisée comme une broche standard RBO ou comme une entrée d'interruption INT .

Si cette broche est utilisée comme une entrée d'interruption externe, elle doit être maintenue à un niveau haut par l'intermédiaire de résistances de 10 k $\Omega$  pour ne pas déclencher d'interruptions imprévues, cela permet aussi de relier plusieurs sources d'interruptions sur une même ligne (OU CABLE).

## I.6. Les différentes mémoires de pic 16F877: [5]

Dans le pic 16F877, il y'a trois types de mémoire ce sont les:

### I.6.1. La mémoire FLASH (programme):

La mémoire FLASH appelée aussi (mémoire programme) car c'est une mémoire programme de taille 8ko. Chaque case mémoire unitaire est de taille 13 bits. Cette mémoire est de type mémoires stable, c'est-à-dire qu'on peut réécrire dessus à volonté, car le 16F877 est caractérisé par la possibilité d'écrire des données. La zone mémoire est caractérisée par une adresse de 13 bits, alors ceci nous impose donc pour l'adressage les registres EEAR et EEADRH. De même, nous aurons pour les données, les registres EEDATA et EEDATH.



# **Chapitre 1: Présentation de microcontrôleur 16F877.**

---

## **I.6.2. La mémoire RAM:**

Cette mémoire de taille 368 octets est une mémoire d'accès rapide (Random Access Memory) et elle est volatile (les données seront perdus lorsque elle n'est plus sous tensions).

Elle contient tous les registres de configuration du PIC ainsi que les différents registres de données. Elle contient également les variables utilisées par le programme. La RAM est la mémoire la plus utilisée. Toutes les données qui y sont stockées sont perdues lors d'une coupure de courant .

La RAM est subdivisée de plus en deux parties dans chacune on trouve des « cases mémoire spéciales » appelée REGISTRES SPECIAUX et des cases mémoire « libre » dont on peut se servir provoque un fonctionnement spécial du PIC ou la mise en service d'une fonction particulière.

## **I.6.3. La mémoire EPROM interne : [9]**

Le pic 16F877 contient également la mémoire électriquement effaçable, réécrivable et stable. Ce type de mémoire est d'accès plus lent.

Pour gérer cette EEPROM on a besoin de quatre registres, à savoir EEDR, EEDATA, EECON1 et EECON2.

Le registre EEADR est utilisé pour placer l'adresse relative en EEPROM, tandis que le EEDATA contient la donné à lire ou à écrire.

L'adresse relative de l'accès EEPROM est donc comprise entre 0000 et 00FF ce qui nous permet d'utiliser un registre de huit bit pour définir cette adresse.

## **I.7. Les ports entrée/sortie de Pic 16F877: [4]**

Le pic 16F877 dispose de 5 ports entrée/sortis et voici par la suite les particularités de chaque port :

### **I.7.a. Le port A :**

Dans ce pic ,6 pins I/O numérotées de RA0 à RA5. Nous avons donc 6 bits utiles dans le registre TRIS A. Les bits RA6 et RA7 de ces registres ne sont pas implantés. Ils seront lus comme des « 0 ».

Le PORT A figure I.4, au moment du reset, est configuré comme un ensemble d'entrées analogiques. Donc nous devons forcer une valeur dans le registre ADCON1 dans notre routine d'initialisation pour pouvoir utiliser ce port comme port d'entrée/sortie de type général.



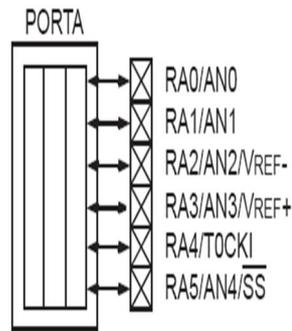


Figure I.4: Le schéma du PORT A.

### I.7.b. le PORT B:

Il contient 8 pins de RB0 à RB7 (figure I.5).

Le pin RB0 qui, en configuration d'entrée, est de type « trigger de Schmitt » quand elle est utilisée en mode interruption « INT ». La lecture simple de RB0 se fait de façon tout à fait classique, en entrée de type TTL.

Au moment de reset on doit forcer une valeur dans le registre ADCON1, pour pouvoir utiliser ce port en entrée/sortie de type générale

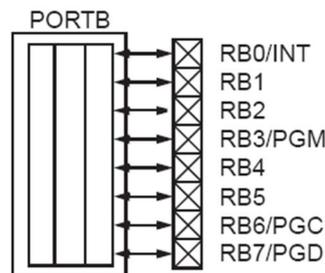


Figure I.5: Le schéma du PORT B.

### I.7.c. Le PORT C :

C'est un port ( voir figure I.6) out ce qu'il a de plus classique, or qu'il a deux pins qu'on utilisera plus tard dans la communication série avec le PC à travers (TX et RX) (pin25et pin 26).

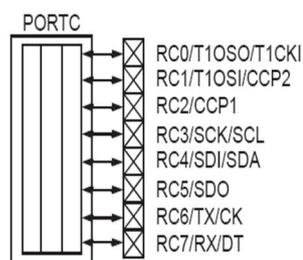


Figure I.6: Le schéma du PORT C.



# Chapitre 1: Présentation de microcontrôleur 16F877.

## I.7.d. Le PORT D :

Ce port (voir figure I.7) fonctionne de façon identique aux autres, dans son mode de fonctionnement général. Le registre TRISD comportera donc les 8bits de direction, pendant que le registre port D correspond aux pins I/O concernés. D'où les 8 pins I/O, en mode entrée, sont du type <trigger de Schmitt>.

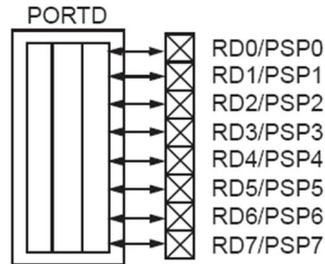


Figure I.7 : schéma du PORT D.

## I.7.e. Le PORT E :

Ce port (figure. I.8) n'est présent que sur les PIC 16F877. Il ne comporte que 3 pins RE0 à RE2, mais, contrairement aux ports, les bits non concernés de TRISE sont, cette fois, implantés pour d'autres fonctions. Les pins REX peuvent également être utilisés comme pins d'entrées analogiques. D'où le registre ADCON1 qui détermine si ce port est utilisé comme port I/O ou comme port analogique.

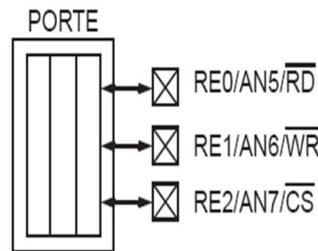


Figure I.8: Le schéma du PORT E.

## I.8. Les Timers :[1]

Notre pic possède 3 timers qui sont :

- Le Timer0 (8bits) : il peut être incrémenté par des impulsions extérieures via la broche (TOCKI/RA4) ou par l'horloge interne (Fosc/4).
  - Le Timer1 (16 bits) : il peut être incrémenté soit par l'horloge interne par des impulsions sur la broche T1CKI/RC0 ou par un oscillateur (RC ou quartz) connecté sur les broches T1OSO/RC0 et T1OSI/RC1.
  - Le Timer2 (8bits) : il est incrémenté par l'horloge interne, celle peut être pré divisée.
- Tous ces timers peuvent déclencher une interruption interne, s'ils ont été autorisés.



# Chapitre 1: Présentation de microcontrôleur 16F877.

## I.9. L'unité centrale :[2]

### I.9.1. Organisation mémoire:

Comme les PICs utilisent un bus pour les instructions et un bus pour les données, il faut considérer deux plans mémoire l'un pour les instructions et l'autre pour les données ainsi que les registres internes (voir figure I.9).

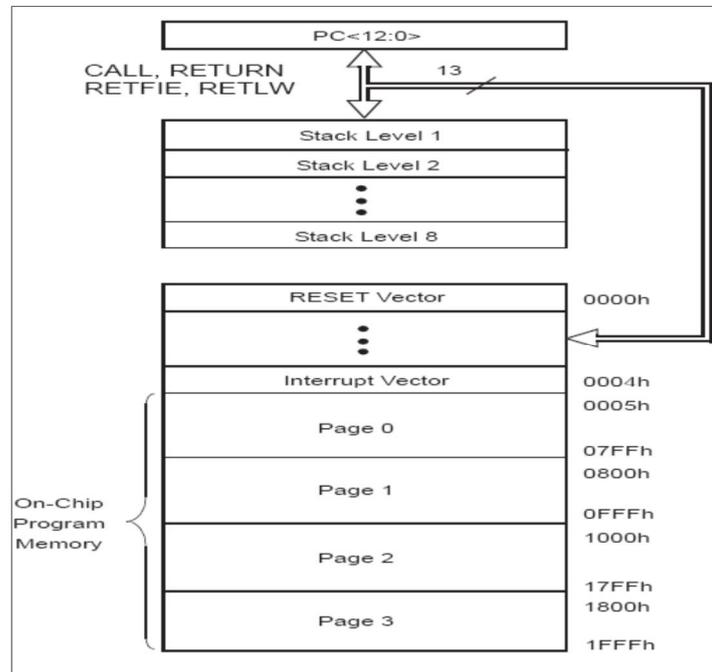


Figure I.9 : Plan Mémoire pour les instructions  
(Code programme)

Le plan mémoire est linéaire les adresses vont de 0000h à 1FFFh (8k mots de 14 bits), par page de 2K mots. On peut remarquer, le vecteur de reset est figé en 0000h.

Les Pics n'ont qu'un seul vecteur d'interruption en 0004h. Lors d'une interruption, le sous programme associé devra déterminer quel périphérique a demandé une interruption.

## I.10. Les interruptions :[5][3]

### I.10.1. Présentation :

Le microcontrôleur dispose de plusieurs sources d'interruptions.

- Une interruption externe, action sur la broche INT/RB0.
- Débordement du TIMER0.
- Changement d'état logique sur une des broches du PORTB (RB4 à RB7).
- Une interruption d'un des périphériques (PEIE).
- Fin de programmation d'une case mémoire de l'EEPROM.
- Changement d'état sur le PORTD (PSPIE).



# Chapitre 1: Présentation de microcontrôleur 16F877.

- Fin de conversion analogique numérique (ADIE).
- Réception d'une information sur la liaison série (RCIE).
- Fin d'émission d'une information sur la liaison série (TXIE).
- Interruption SPI ou I2C du module MSSP (SSPIE).
- Interruption du registre de capture et/ou de comparaison 1 (CCPI1E).
- Interruption du registre de capture et/ou de comparaison 2 (CCPI2E).
- Débordement du TIMER1 (TMR1E).
- Débordement du TIMER2 (TMR2E).
- Collision de BUS (BCLIE)

## I.10.2. Fonctionnement :

Lors d'un événement dans un ou plusieurs des circuits périphériques (ADC, EEPROM, USART-SCI, MSSP-I2C-SPI, TIMER1, TIMER2) comme par exemple : la fin de conversion, la fin de programmation d'un octet dans l'EEPROM, la réception d'une information, la détection d'un front, etc... et si le bit de l'interruption concernée a été autorisée (EEIE, PSPIE, ADIE, RCIE, TXIE, SSPIE, CCP1IE, TMR2IE, TMR1IE, CCP2IE ou BCLIE : Registres PIE1 et PIE2) alors une interruption périphérique est déclenchée. Pour que celle-ci soit prise en compte il faut que le bit d'autorisation des interruptions périphériques soit positionné à 1 (PEIE) ainsi que le bit GIE d'autorisation globale des interruptions du registre INTCON.

Pour qu'une interruption du type TIMER0 ou INT/RB0 ou PORTB soit prise en compte il suffit que le bit local d'autorisation d'interruption soit positionné à 1 (TOIE ou INTE ou RBIE) ainsi que le bit GIE d'autorisation globale des interruptions du registre INTCON. Dans ces conditions le programme en cours d'exécution est interrompu et le microcontrôleur exécute le programme d'interruption à partir de l'adresse 0x0004. Au début de celui-ci il faut que le logiciel vérifie quel périphérique a déclenché l'interruption.

## I.10.3. Mécanisme d'interruption sur les PICs :

Le principe de fonctionnement des interruptions sur les pics, est le même de celui des autres microcontrôleurs, mais les PICs elles ont ses propres spécificités.

- Tout d'abord, l'adresse de début de toute interruption est fixe. Il s'agit toujours de l'adresse 0x04. Toute interruption provoquera le saut du programme vers cette adresse.
- Toutes les sources d'interruption arrivant à cette adresse, si le programmeur utilise plusieurs sources d'interruptions, il lui faudra déterminer lui-même laquelle il est en train de traiter.



## Chapitre 1: Présentation de microcontrôleur 16F877.

- Les PICs en se connectant à cette adresse, ne sauvent rien automatiquement, hormis le contenu du PC, qui servira à connaître l'adresse du retour de l'interruption. C'est donc à l'utilisateur de se charger des sauvegardes.
- Le contenu du PC est sauvé sur la pile interne (8 niveaux). Donc, si vous utilisez des interruptions, vous ne disposez plus que de 7 niveaux d'imbrication pour vos sous-programmes. Moins si vous utilisez des sous-programmes dans vos interruption.
- Le temps de réaction d'une interruption est calculé de la manière suivante : le cycle courant de l'instruction est terminé, le flag d'interruption est lu au début du cycle suivant. Celui-ci est achevé, puis le processeur s'arrête un cycle pour charger l'adresse 0x04 dans PC. Le processeur se connecte alors à l'adresse 0x04 où il lui faudra un cycle supplémentaire pour charger l'instruction à exécuter. Le temps mort total sera donc compris entre 3 et 4 cycles.
- Une interruption ne peut pas être interrompue par une autre interruption. Les interruptions sont donc invalidées automatiquement lors du saut à l'adresse 0x04 par l'effacement du bit GIE (que nous allons voir).
- Les interruptions sont remises en service automatiquement lors du retour de l'interruption. L'instruction RETFIE agit donc exactement comme l'instruction RETURN, mais elle repositionne en même temps le bit GIE.

### I.11. L'oscillateur :[5][6]

L'horloge est un système qui peut être réalisée soit avec un QUARTZ(a), soit avec une horloge extérieur(b), soit avec un circuit RC(c), dans ce dernier la stabilité du montage est limitée. La fréquence maximale d'utilisation va dépendre de Microcontrôleur utilisé. Le suffixe indiqué sur le boîtier donne la nature de l'horloge à utiliser et sa fréquence maximale voir (figure .I.10).

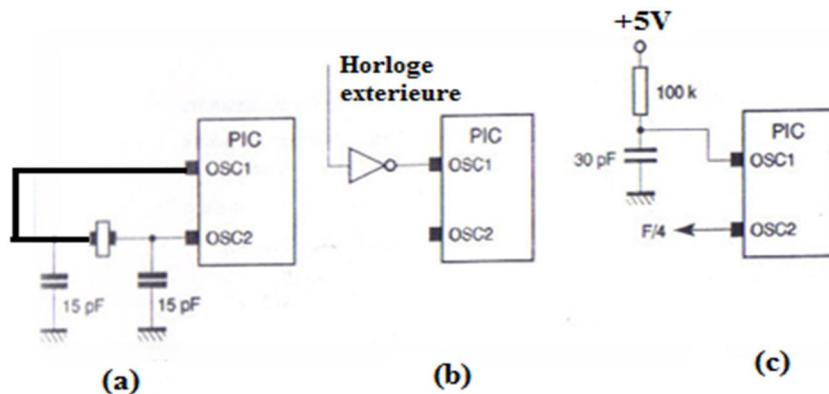


Figure. I.10. Schéma d'oscillateur.



# Chapitre 1: Présentation de microcontrôleur 16F877.

## I.11. Le convertisseur:[5]

Le principe de la conversion suit la séquence est la suivante :

- Le pic connecte le pin sur laquelle se trouve la tension à mesurer à un condensateur interne, qui va se charger via une résistance interne jusque la tension appliquée.

Le pin est déconnecté du condensateur, et ce dernier est connecté sur le convertisseur analogique/numérique interne.

Le temps nécessaire à la conversion est égal au temps nécessaire à la conversion d'un bit multiplié par le nombre de bits désirés pour le résultat. Concernant notre pic, il faut savoir qu'il nécessite, pour la conversion d'un bit, un temps nommé **Tad**.

Ce temps est dérivé par division d'horloge principale. Le diviseur peut prendre une valeur de 2, 8 ou 32. Le temps de conversion **Tad** ne peut descendre, pour des raisons électroniques, en dessous de 1.6 us pour les versions classiques de 16F87x, et en dessous de 6 us pour les versions LC. Donc en fonction des fréquences utilisées pour le quartz du pic, on choisit le diviseur le plus approprié.

Voici le tableau .I.4. qui reprend les valeurs de diviseur à utiliser pour quelques fréquences courantes du quartz et pour les PICs de type classique :

	20Mhz	5Mhz	4Mhz	2Mhz	1,25Mhz	333,3Khz
2	100ns	400ns	500ns	1µs	1,6µs	6µs
8	400ns	1,6µs	2µs	4µs	6,4µs	24µs
32	1,6µs	6,4µs	8µs	16µs	25,6µs	96µs
<b>Osc RC</b>	<b>2-6µs</b>	<b>2-6µs</b>	<b>2-6µs</b>	<b>2-6µs</b>	<b>2-6µs</b>	<b>2-6µs</b>

Tableau. I.4. Valeur de diviseur pour les fréquences du quartz et le pic.

Les valeurs 400ns; 1,6µs; 2µs; 4µs; 6,4; 24µs correspondent au meilleur diviseur en fonction de la fréquence choisie, en ce qui nous concerne puisqu'on utilise une fréquence de 4MHZ, on utilisera la valeur de 2µs dans notre programmation. Il faut à présent préciser que le PIC nécessite un temps Tad avant le démarrage effectif de la conversion, et un temps supplémentaire Tad à la fin de la conversion.

Résumons donc le temps nécessaire pour effectuer l'ensemble des opérations :

- On charge le condensateur interne (nécessite le temps Tacq).
- On effectue la conversion (nécessite le temps 12 \* Tad).
- On doit attendre 2 \* Tad avant de pouvoir recommencer une autre conversion.



# Chapitre 1: Présentation de microcontrôleur 16F877.

Et donc voici la figure I.11. qui montre un cycle de conversion.

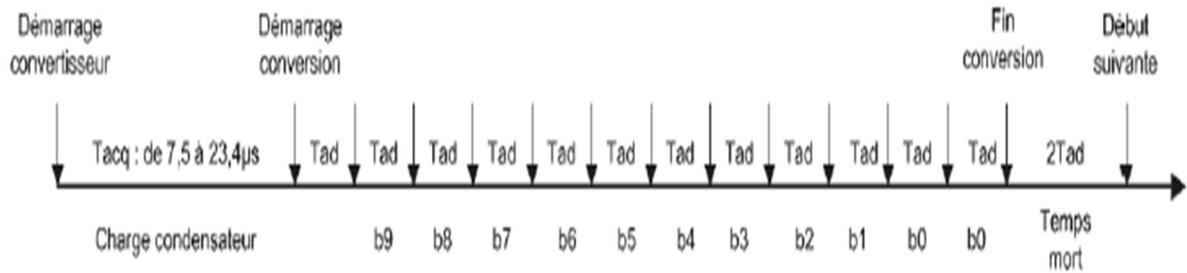


Figure I.11. Cycle de conversion.

Le module de conversion utilise 4 registres disposés comme suit :

- \*Registre de résultat High (ADRESH).
- \*Registre de résultat Low (ADRESL).
- \*Registre0 de contrôle (ADCON0).
- \*Registre1 de contrôle (ADCON1).

Regardons cela de près dans la figure I.12. Ci-dessous :

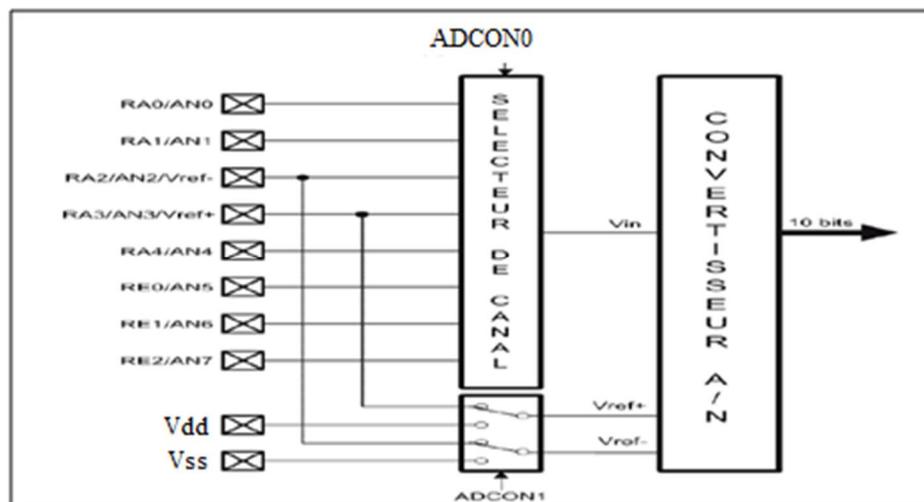


Figure I.12. Module du convertisseur.

## I.12. Conclusion:

Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur: processeur, mémoire, unités périphériques, interfaces d'entrée – sortie. Mais les microcontrôleurs se caractérisent par:

- un plus haut degré d'intégration,
- une plus faible consommation électrique,
- une vitesse de fonctionnement plus faible,
- un coût réduit, par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs.



**CHAPITRE II : LE  
STANDARD RS-232**

## *Chapitre II: Le standard RS-232.*

---

### **II.1.Introduction:**

Le port série est disponible sur presque tous les PC. Le port série est désigné par le nom « COM ». La liaison RS232 est utilisée pour connecter le PC aux appareils électroniques (automate, appareil de mesure, etc.). Dernièrement le port USB commence de plus en plus à remplacer le port série.



## **Chapitre II: Le standard RS-232.**

---

### **II.2. Les liaisons série :[3]**

Les liaisons séries permettent la communication entre deux systèmes numériques en limitant les nombres de fils de transmission.

Les bits constituant l'information à transmettre sont émis successivement, l'un après l'autre. Il peut y avoir, en plus des bits d'information, des bits de contrôle entrelacés. On commence toujours par le bit de poids faible.

Ex: code ASCII (7bits).

Le mot à transmettre est 1001111, les bits transmis sont 1, 1, 1, 1, 0, 0,1.

Pour transmettre en série, il y a deux modes : synchrone et asynchrone.

### **II.3. Transmission série asynchrone :[5]**

On ne transmet pas de signal d'horloge .pour permettre à l'émetteur et au récepteur d'être bien synchronisé, on transmet des blocs de longueur constante, en ajoutant à l'information des bits de contrôle et de synchronisation.

La synchronisation se passe au niveau du mot. Un bloc d'informations est précédé par un bit de départ (Start, toujours à 0) et suivi par un (ou plusieurs) bit (s) d'arrêt (Stop, toujours à 1).

Afin d'éviter les erreurs de synchronisation, les horloges des deux parties (l'émetteur et le récepteur) doivent être très précises, st de vitesse pratiquement égale. Pour éviter une désynchronisation à long terme, la taille du mot encadré des bits START et STOP est toujours fixée faible (7 ou 8 bits).

On doit bien sur limiter la vitesse de transmission afin d'éviter les erreurs de synchronisation. Ce mode de transmission est très économique, puisqu'une seule ligne d'information est nécessaire. Ce mode est adapté aux distances longues et aux transmissions ne demandant pas une vitesse trop grande.

La liaison série la plus connue et donc la plus utilisée notamment en informatique est la liaison RS232.

Il existe d'autres liaisons série asynchrone : RS422, RS485, bus I<sup>2</sup>C...

### **II.4. La norme et la liaison RS-232:[2]**

Le standard RS-232 définit deux aspects d'une liaison série asynchrone : électrique et fonctionnel.

#### **II.4.1. Aspect électrique :**

La norme définit les valeurs suivantes pour les tensions électriques:

Pour l'émetteur:

Le bit 0 doit être transmis par une tension électrique compris entre +5 et +15V.



## Chapitre II: Le standard RS-232.

Le bit 1 doit être transmis par une tension électrique comprise entre -5 et -15V.

Les niveaux classiques utilisés sont +12 et -12V.

Pour le récepteur :

La tension maximale admissible sans dégâts est de +/-25V. La tension minimale (seuil) détectable est de +/-3V.

La définition de ces tensions permet une assez bonne immunité au bruit, et un risque de transformation de bits (0->1 ou 1->0) assez faible.

### II.4.2. Aspect fonctionnel:

La norme définit comment doit être constituée la jonction.

### II.5. La liaison RS 232 :[1]

La liaison série aux normes RS 232 est utilisée dans tous les domaines de l'informatique (ex : port de communication com1 et com2 des PC, permettant la communication avec des périphériques tels que modem et souris). Elle est de type asynchrone, c'est à dire qu'elle ne transmet pas de signal horloge.

Le schéma fonctionnel est montré dans la figure .II.1.suivante:

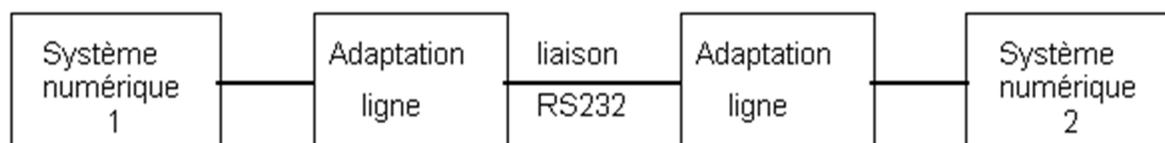


Figure. II.1. Le schéma fonctionnel de la liaison RS232.

La transmission série nécessite un minimum de 2 fils comportant les trames de données en émission (TxD) et en réception (RxD).

L'adaptation des données se fait à l'aide d'un circuit adaptateur de ligne (ex : MAX232), qui transforme les niveaux logiques issus du système numérique en niveaux logiques compatibles avec les normes RS232 et vice versa.

### II.5.1. Les caractéristiques de la liaison RS232

- Mot transmis de 5,6 7 ou 8 bits.
- 1 bits de START (0) et 1,1.5 ou 2 bits de stop (1) indiquent le début et la fin du mot transmis.
- 1 bit de parité (paire ou impaire) est éventuellement ajouté pour détecter les erreurs de transmission.



## Chapitre II: Le standard RS-232.

- Transmission à 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, ..., 115.2K bit/s (appelé aussi Baud)
- Half duplex (émission/réception alternées) ou Full-duplex (émission/réception simultanées).

Pour transmettre 8 bits utiles : de 10 à 12 bits sont envoyés.

Caractéristiques électriques :

'0' → niveau émis +V, '1' → niveau émis -V.

Avec V compris entre 5 et 25 Volts (12Volts pour les ports séries COM1, COM2 sur un PC).

Transmission limitée à une dizaine de mètres maximum.

### II.5.2. Paramètres techniques de la RS232:

#### II.5.2. a. Liaison unidirectionnelle

Il s'agit ici vraiment de la plus simple que l'on puisse trouver.

Cette liaison (figure II.2) est peu utilisée car elle n'offre pas beaucoup de possibilités.

Elle ne permet que la transmission dans un seul sens.

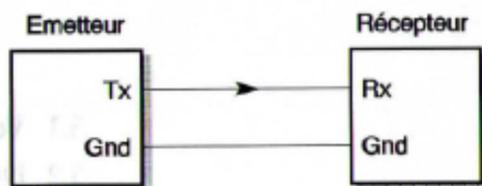


Figure. II.2: La liaison unidirectionnelle.

Et voici aussi le chronogramme d'une liaison unidirectionnelle (figure II.3).

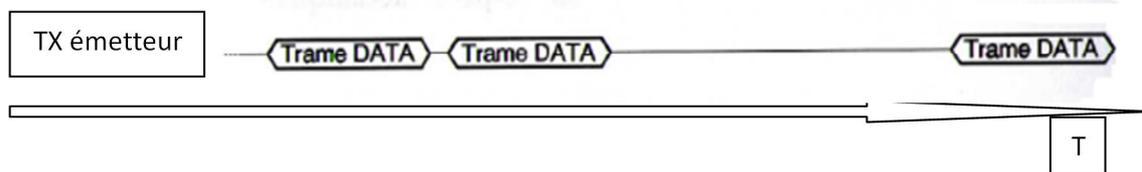


Figure .II.3 : La liaison unidirectionnelle.

#### II.5.2. b. Liaison half duplex :

Ce type de liaison Figure (II.4) permet de dialoguer dans les deux sens mais pas simultanément (donc à chacun son tour). Elle est également peu utilisée. En effet,



## Chapitre II: Le standard RS-232.

seuls certains systèmes, lents ou ayant une unité de contrôle RS232 peu performante, fonctionnent encore uniquement dans ce mode.

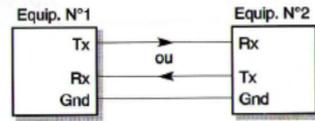


Figure .II.4: La liaison half duplex.

Et voici aussi le chronogramme d'une liaison half duplex (figure II.5)

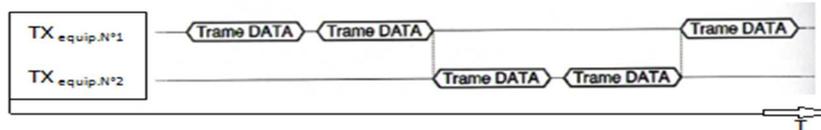


Figure .II.5: Le chronogramme d'une liaison half duplex.

### II.5.2. c. Liaison full duplex :

Cette liaison (Figure II.6) dispose du même câblage que la liaison half duplex mais cette fois les deux interlocuteurs peuvent émettre et recevoir en même temps. Ce mode de transmission est asynchrone, les émissions et les réceptions n'ont pas à être synchronisées.

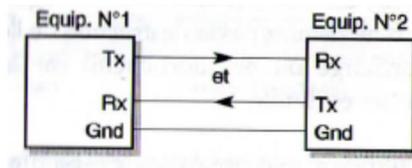


Figure .II.6: La liaison full duplex.

Et voici aussi le chronogramme d'une liaison half duplex (figure II.7)

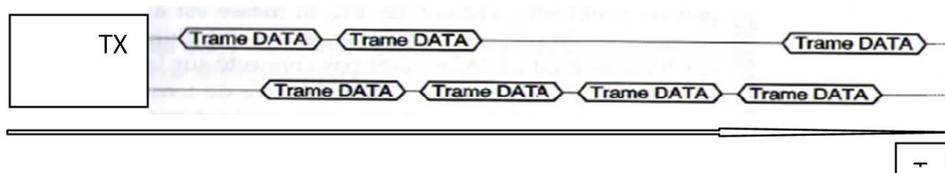


Figure. II.7:Le chronogramme d'une liaison full duplex.

### II.6. Le protocole de transmission de la norme RS232 :

La figure suivante(II.8) montre le chronogramme de protocole de RS232.

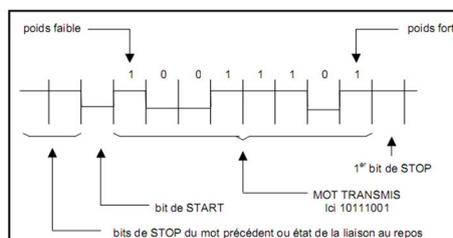


Figure .II.8: le chronogramme de protocole de RS232.



## Chapitre II: Le standard RS-232.

Bit de START : la ligne au repos est à l'état logique 1 pour indiquer qu'un mot va être transmis la ligne passe à l'état bas avant de commencer le transfert. Ce bit permet de synchroniser l'horloge du récepteur.

Bit de stop : après la transmission, la ligne est positionnée au repos pendant 1, 2 ou 1,5 période d'horloge selon le nombre de bits de stop.

Donnée : le bit de poids faible est envoyé en premier. La longueur du mot peut être de 7 bits (ex : caractère ASCII) ou 8 bits.

La vitesse de transmission : les différentes vitesses de transmission sont réglables à partir de 110 bauds (bits par seconde).

Parité : le mot transmis peut être suivi ou non d'un bit de parité qui sert à détecter les erreurs éventuelles de transmission. Il existe deux types de parité. Ce bit de parité est généré automatiquement par l'UART de l'émetteur. A la réception on compare la parité obtenue avec celle reçue.

a) Parité paire : le bit ajouté à la donnée est positionné de telle façon que le nombre des états 1 soit paire sur l'ensemble donnée + bit de parité

Ex : soit la donnée 11001011 contenant 5 états 1, le bit de parité paire est positionné à 1, ramenant ainsi le nombre de 1 à 6.

b) Parité impaire : le bit ajouté à la donnée est positionné de telle façon que le nombre des états 1 soit impaire sur l'ensemble donnée + bit de parité

Ex : soit la donnée 11011001 contenant 5 états 1, le bit de parité impaire est positionné à 0, laissant ainsi un nombre de 1 impaire.

### II.6.1. Handshake :

La méthode la plus courante, pour réguler le flux de données, est l'HandShaking.

Elle utilise une liaison full duplex complète. Ce protocole est géré directement par l'UART. Vous trouverez sur le chronogramme suivant (figure II.9), le mécanisme d'une communication.

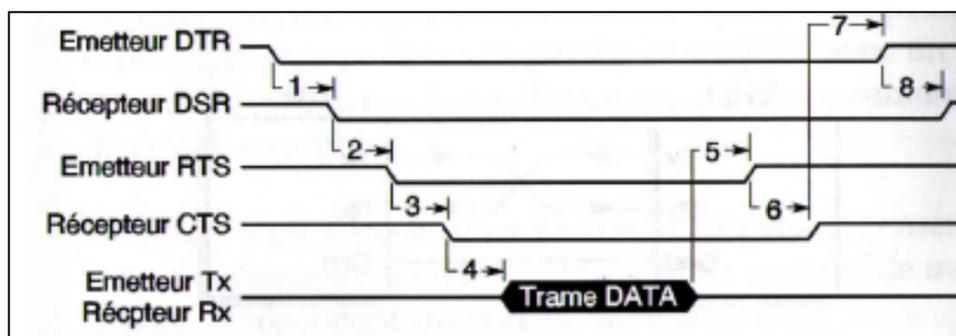


Figure. II.9. Chronogramme de protocole matériel.



## Chapitre II: Le standard RS-232.

### II.7. La norme fixe des niveaux de tension d'une liaison RS232 :

De +3v à +25v : état 0 (niveau logique bas).

De -3v à -25v : état 1 (niveau logique haut).

Dans les systèmes informatiques on utilise souvent le +/- 12v ou le +/- 5v qui sont disponibles au niveau de l'alimentation.

Des circuits intégrés comme le MAX 232 réalise l'adaptation TTL ou CMOS vers RS232 pour des liaisons de type PC vers périphérique par exemple. Ils intègrent deux convertisseurs de tensions capables, avec une seule alimentation +5 volts de produire deux tensions symétriques par rapport à la masse de 10 volts. Le circuit MAX 233 réalise la même fonction avec en plus l'avantage d'intégrer les condensateurs dans son boîtier DIL 20 broches.

<p><b>Avant adaptation :</b> Les niveaux logiques sont les suivants : niveau 0 = 0 V niveau 1 = 5 V</p>	<p><b>Après adaptation :</b> Les niveaux logiques sont les suivants : niveau 0 = +12 V niveau 1 = -12 V</p>
---	---

Tableau .II.1: les niveaux de tension avant et après l'adaptation.

Et voila le chronogramme (figure. II.10) de l'adaptation à base de MAX 232.

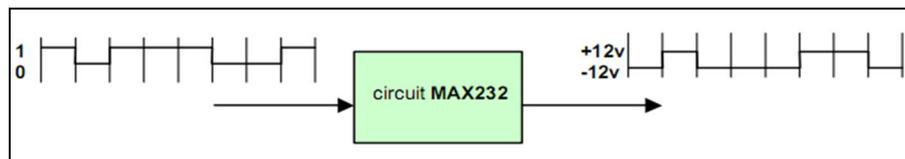


Figure. II.10:Chronogramme de l'adaptation à base de MAX 232.

### II.8. Brochage d'une liaison RS232 :

La liaison RS232 nécessite un minimum de 3 fils, (Tx) pour l'émission, (Rx) pour la réception et une ligne de masse.

La figure ci-dessous (figure II.11) montre le brochage d'un câble RS232 croisé.

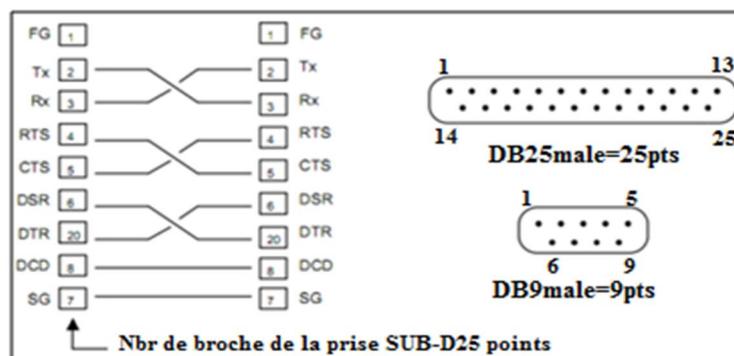


Figure .II.11: le brochage d'un câble RS232 croisé.



## Chapitre II: Le standard RS-232.

Il existe deux types de ports série, la première consistant en 25 pôles qui est le format d'origine définies dans le Protocole et a appelé D25 (très vieux). Et le second, dans lequel un 9 petit pôle, appelé D9 est apparu dans les ordinateurs personnels actuels en raison de la non-utilisation d'autres pôles dans le grand port. La figure suivante(II.12) montre le port série D9 et la distribution des pôles dans lesquels:

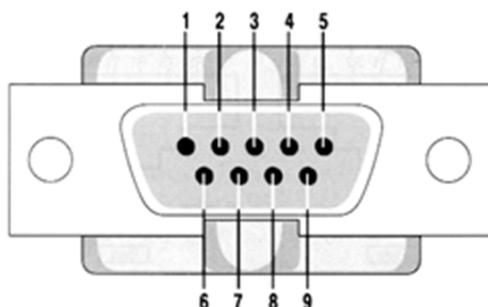


Figure .II.12 : le port série D9 et la distribution des pôles.

Le tableau (II.2). Montre l'affectation physique de chacun des signaux que nous allons décrire brièvement :

Broche	nom	Indiquée comme
1	CD	entrée
2	RX	entrée
3	TX	sortie
4	DTR	sortie
5	GND	masse
6	DSR	entrée
7	RTS	sortie
8	CTS	entrée
9	RI	entrée

Tableau. II.2:les pôles de D9.

- GND ou SG (terre): fournit un terrain d'entente pour la transmission et la réception.
- TxD (Données transmises) les informations envoyées (sortie): les informations sont envoyées à ce pôle. Cette ligne est une sortie. Les données de l'ordinateur vers le correspondant sont véhiculées par son intermédiaire,
- Rxd (données reçues) de recevoir des informations (entrée): L'information est reçue sur ce pôle. Cette ligne est une entrée. C'est ici que transitent les informations du correspondant vers l'ordinateur.
- DTR (Data Terminal Ready) (sortie) serrant la main avec une ligne indique que le terminal de données (informatique) est prêt à envoyer. Cette ligne est une sortie active au niveau logique haut. 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.

## Chapitre II: Le standard RS-232.

---

- DSR (Data Set Ready) (entrée): la ligne de serrer la main indique que le dispositif de données (modem) est prêt à recevoir des informations de votre ordinateur. Cette ligne est une entrée active au niveau logique haut. Elle permet au correspondant de signaler qu'une donnée est prête.
- RTS (Request To Send) (sortie) serrant la main avec la ligne nécessaire pour mener à bien vos données à envoyer. Cette ligne est une sortie active au niveau logique haut. Elle indique au correspondant que l'ordinateur veut lui transmettre des données.
- CTS (Clear To Send) (entrée): indique le terminal de ligne poignée de main (ordinateur) pour la possibilité de commencer le processus d'envoi. Cette ligne est une entrée active au niveau logique haut. Elle indique à l'ordinateur que le correspondant est prêt à recevoir des données,
- CD (Carrier Detected) (entrée): indique à l'ordinateur sur la disponibilité de signal porteur (ton Walton). Cette ligne est une entrée active au niveau logique haut. Elle signale à l'ordinateur qu'une liaison a été établie avec un correspondant.
- RI (*Ring Indicateur*) : cette ligne est une entrée active au niveau logique haut. Elle permet à l'ordinateur de savoir qu'un correspondant veut déclencher une communication avec lui.

Peut atteindre un minimum de contact sans l'aide de trois lignes de handshake sont le seul SG et TxD et Rxd.

### II.9.Conclusion :

Le port série RS232 à l'avantage d'utiliser un nombre réduit de fils (par rapport à la communication parallèle). Deuxième avantage est la possibilité de communiquer sur de grandes distances.



**CHAPITRE III :**  
**ETUDE DE KIT DU**  
**DEVELOPPEMENT**

## ***Chapitre III: Etude du kit du développement.***

---

### **-III.1.Introduction:**

Dans ce chapitre on va présenter une description détaillée du principe des systèmes du développement et le contenu de base ou idéale d'un kit du développement et en abordant la conception détaillée de chaque partie du système afin d'obtenir une schématisation complète et précise.



### **III.2. Généralités sur les systèmes et les kits de développements:[8]**

#### **III .2.1.Les systèmes de développement:**

Pour utiliser un microcontrôleur dans un système plusieurs étapes sont nécessaires :

1. Le choix du microcontrôleur de la carte .Ils sont déterminé par le nombre de ports nécessaires, les fonctions à réaliser et la vitesse souhaitée.
2. L'écriture du logiciel aboutissant à une liste d'octets qui devront être implantés dans la ROM programme.
3. Le test de ce logiciel dans des conditions aussi proches que possible de la réalité dans laquelle travaillera le microcontrôleur.
4. La gravure définitive de la PROM qui sera implantées sur la carte finale. Passons en revue ces diverses étapes.

L'étape 1 est essentiellement un problème d'électronique qui ne peut être détaillé que sur un exemple.

#### **III.2.2.Base d'un système de développement:**

##### **III.2.2.a. Côté logiciel:**

Un système de développement comporte en premier lieu un assembleur et parfois un ou des compilateurs adaptés au langage évolué que l'on souhaite utiliser pour programmer.

L'assembleur traduit les instructions écrites en utilisant les mnémoniques du langage machine en code binaire exécutable par le microcontrôleur.

Le compilateur quant à lui traduit les instructions écrites en langage évolué (BASIC, C, PASCAL, MIKRO C...) qui constituent aussi ce que nous appelons le listing ou code source, en code binaire exécutable par le microcontrôleur qui constitue le code objet.

Dans un système de développement bien conçu, les deux programmes, assembleur et compilateur, peuvent coexister et être utilisables l'un et l'autre sans difficulté. Ces deux programmes, assembleur et/ou compilateur doivent nécessairement fonctionner sur une machine appelée machine hôte. Cette machine peut être un système spécifique du fabricant des microcontrôleurs, une station de travail, un calculateur ou un PC.

Une fois le programme de l'application écrit et assemblé ou compilé sur la machine hôte, nous sommes en possession d'un binaire exécutable.



## Chapitre III: Etude du kit du développement.

### III.2.2.b. Côté matériel:

Pour implanter le binaire exécutable, nous avons besoin d'un programmeur. Cet appareil est en fait un système qui va transférer de la machine hôte, le code objet du programme dans la mémoire du microcontrôleur.

### III.2.3.Organisation du développement (matériel et programmation):

Pour cette solution, on développe conjointement la carte microcontrôleur et le logiciel.

Le développement de la carte et du logiciel associé comprend trois phases:

- . Développement de la carte et du programme simultanément.
- . Association des deux parties matérielle et logicielle.
- . Utilisation d'un émulateur pour tester cet ensemble.

La figure.III.1 suivante montre comment ça marche l'organisation du développement.

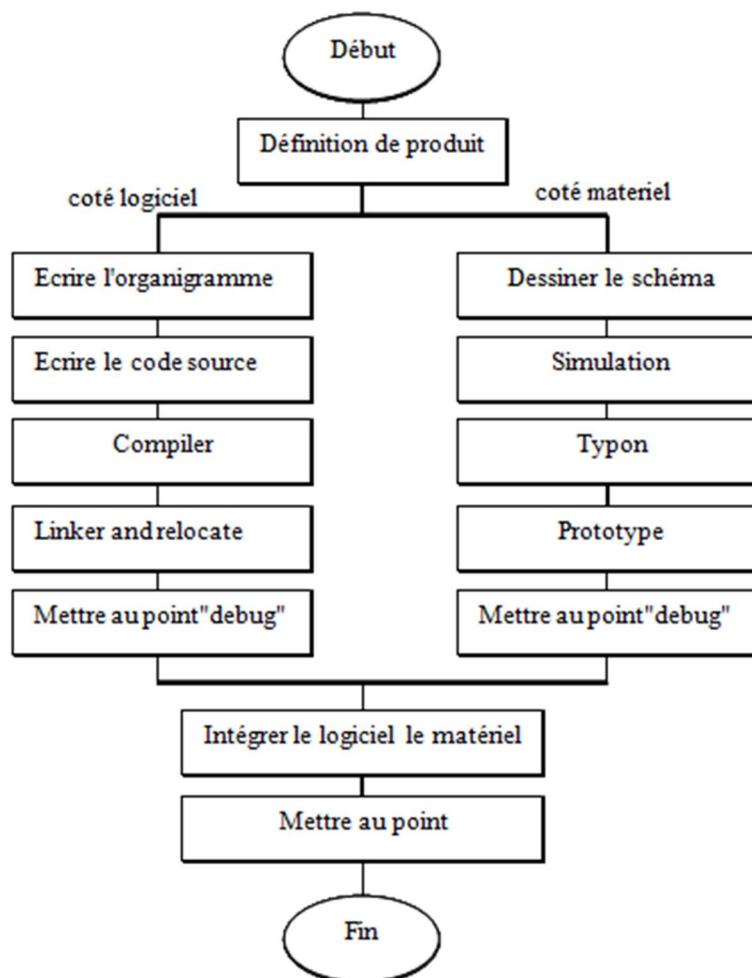


Figure.III.1: l'organisation du développement.

On ne peut pas parler de microcontrôleur sans aborder les logiciels de programmation et les matériels permettant de développer le composant. Pour réaliser les programmes exécutables, on utilisera l'assembleur ou le compilateur C ou le Mikro C ...etc.



### Chapitre III: Etude du kit du développement.

La figure.III.2 suivante montre comment ça marche l'organisation du développement coté de programmation seulement.

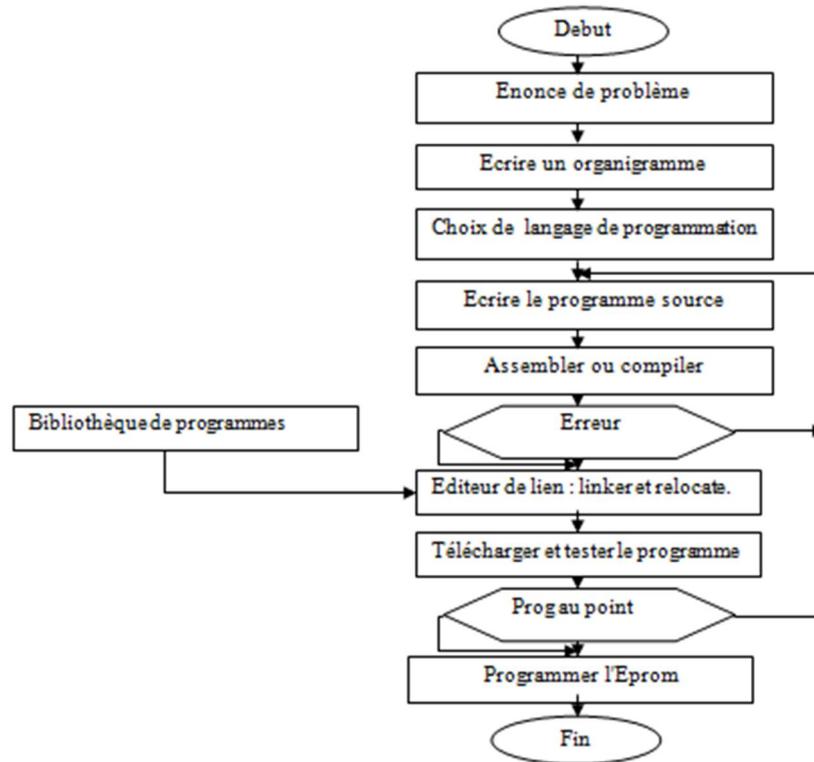


Figure.III.2.Organisation de développement (programmation seule).

Et voici la figure III.3 suivante qui décrit la chaîne de développement d'un système de développement

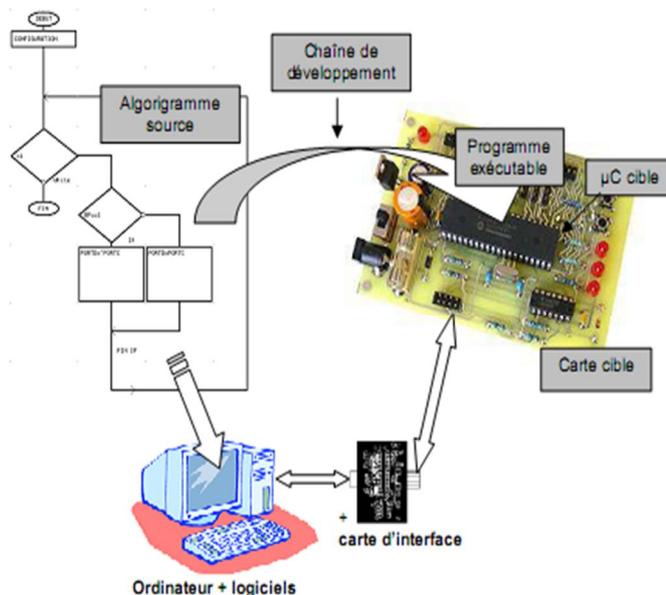


Figure.III.3:chaîne de développement d'un système de développement.



### **III.3. Le kit de développement d'application à base de pic:[10][1]**

#### **III.3.1. Le contenu type d'un kit de développement:**

Idéalement, un kit de développement doit pouvoir supporter un grand nombre de microcontrôleur de la famille choisie.

Pour ce qui est des pic qui nous intéressent aujourd'hui, il faut donc qu'il admette au minimum les 12xxx et les 16xxx et si possible qu'il supporte aussi la modèle "haute gamme" que sont les 18xxx.

Afin de permettre de tester un maximum d'applications, il doit comporter au minimum les éléments suivants:

- Une alimentation stabilisée permettant d'alimenter le kit avec n'importe quelle source de tension non stabilisée telle que, par exemple, un simple bloc secteur "prise de courant".

- Une circuiterie d'horloge dont on doit pouvoir choisir le type si nécessaire (quartz, cellule R-C, etc...)

- Une commande de reset manuelle au cas, très improbable bien sur (!),ou nos premiers programmes "se planteraient".

- Un certain nombre de LED commandés directement par les ports parallèles des PIC.

- Un certain nombre de poussoirs agissant sur ces mêmes ports parallèles.

- Un ou plusieurs afficheurs a LED7 segments, capable de travailler si possible en mode multiplexé.

- Un afficheur alphanumérique à cristaux liquides (LCD) standard de 2 lignes de 16 ou 20 caractères par exemples.

- Un convertisseur de niveau bidirectionnel TTL-RS232 afin de pouvoir mettre en œuvre très simplement asynchrone de ce type.

- Des résistances de rappel au positif de l'alimentation ou à la masse pour les différentes lignes des ports parallèles.

- Au moins un potentiomètre permettant de tester d'éventuelles entrées analogiques.

- Et enfin il doit donner accès, au moins de connecteurs, à tous les ports des microcontrôleurs qu'elle supporte.

Un tel assemblage de fonctions ne présente aucunes difficultés techniques, et pourtant il est très difficile, à l'heure actuelle, de trouver sur le marché un kit satisfasse à toutes ces contraintes.



### **III.3.2. Les Kits de développement à base de Pic les plus connues:**

Aujourd'hui il existe plusieurs exemples industriels de carte de développement à base de microcontrôleur ou de pic dans le marché, parmi les familles les plus connues on a choisi les 2 familles suivantes:

#### **III.3.2.a. La famille EASY pic de Mikroelektronika:[3]**

Une fois n'est pas coutume, ce n'est ni des USA ni d'Extrême-Orient que nous vient cette famille, mais tous simplement d'Europe, et plus précisément, de l'ancienne Yougoslavie.

Cette famille contient plusieurs versions de kits de développements EASY PIC (1, 2,3.4.5..) parmi lesquels on trouve par exemple le kit de développement EASY PIC 2.

Ce kit spécialement conçu pour supporter une multitude de microcontrôleurs PIC 8, 14, 18, 28, et 40 broches. EASY PIC 2 nous permettra de tester d'étudier et de communiquer avec la plupart des périphériques que nous pourrons rencontrer dans les applications industrielles: afficheurs LCD ou 7 segments, claviers de saisie, conversion analogique numérique, etc.

Ses caractéristiques et ses composants sont en effet les suivants:

- alimentation à partir d'un bloc secteur "prise de courant "externe ou par le port USB d'un PC.
- programmeur de PIC intégré à la liaison avec le PC par port USB (ce qui explique la présence de l'option d'alimentation signalée ci-dessus).
- supports pouvant recevoir les Pic à 8, 14, 18, 28 et 40 broches soit quasiment tous les circuits actuels des familles 12xxx, 16xxx, et 18xxx.
- interface série RS232.
- emplacement pour capteur de température de type DS1820 de Dallas.
- deux potentiomètres pour tester les entrées de conversion analogique/numérique et les comparateurs analogique éventuels.
- quatre afficheurs à LED 7 segments.
- affichage de l'état de tous les ports parallèles du PIC via 32 LED pouvant être désactivées si nécessaire.
- tous les ports parallèles du PIC accessibles depuis l'extérieure via des borniers à picots au pas de 2.54 mm pouvant recevoir directement des connecteurs femelles hE10 pour câbles plats.
- résistances de tirage à l'alimentation positive ou à la masse disponibles sur tous les ports.
- poussoirs (pas moins de 24) connectés aux différents ports parallèles du PIC.
- emplacement prévu pour un afficheur à cristaux standard de 2\*16 ou 2\*20 caractères.
- potentiomètre pour réglage du contraste de l'afficheur LCD installés d'origine pré câblés.



Et donc voici la figure .III.4 qui est une photo générale de la carte EASY PIC 2.

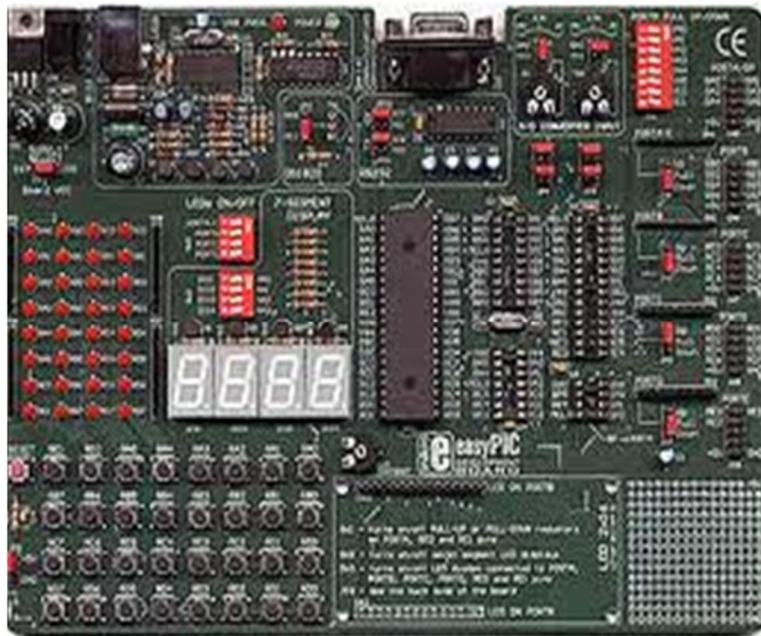


Figure .III.4: photo de la carte EASY PIC2.

### III.3.2.b. La famille PICDEM:

La famille PICDEM est construit par MICROCHIP (le constructeur de microcontrôleur PIC. Et comme la famille précédente est aussi contient plusieurs versions PICDEM (1 , 2 , 3 , ..).Parmi lesquels le kit de développement PICDEM2 PLUS. Mais contrairement a EASY PIC 2 cette carte ne contient pas un programmeur intégré et elle doit utilisée un programmeur universelle comme il est montrée dans la figure suivante (figure III.5):

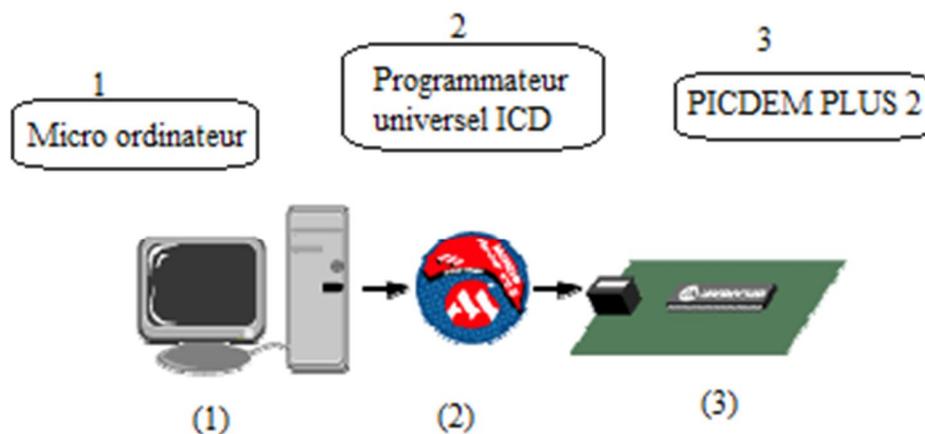


Figure.III.5:L'utilisation de PICDEM PLUS 2 pour programmation.

Pour l'utilisation de la carte PICDEM 2 PLUS pour déboguer

- On peut connecter ICD2 sur le port USB ou série (COM) de l'ordinateur.
- La carte PICDEM 2 Plus ne pourra fonctionner sans ICD2 et l'ordinateur.

### ***Chapitre III: Etude du kit du développement.***

---

- Il faudra configurer MPLAB IDE en pour déboguer.

Les caractéristiques et les composants de PICDEM S PLUS 2 sont en effet les suivants:

- Des supports des microcontrôleurs : contiennent des Supports de 18,28 et 40 broches.

-Les Diodes Electroluminescentes

Une LED verte qui représente la mise sous tension de la carte. Quatre LEDs rouges qui sont connectées au PORTB par l'intermédiaire du jumper.

-L'alimentation de la carte Par une pile de 9V ou par un bloc transformateur AC/DC 9V 100mA.

-La liaison série RS-232

Le connecteur (DB9) permet une communication (série RS232) avec les microcontrôleurs 28 et 40 broches. Les PIC16 et PIC18 qui sont équipés d'un UART, auront leurs broches RX et TX reliées au MAX232.

-Les boutons poussoirs : 3 boutons poussoirs :

S1 permet un RESET du microcontrôleur, actif au niveau bas (broche MCLR).S2 met un niveau bas sur l'entrée RA4/TOCKI S3 met un niveau bas sur l'entrée RB0/INT.

-Des différents oscillateurs.

-Les entrées analogiques : un potentiomètre de 5K $\Omega$  reliée avec une résistance en série à RA0, crée une variation de potentiel sur l'entrée AN0 du Convertisseur Analogique Numérique.

-La connexion avec l'ICD 2 :

Un connecteur relié à ICD2 permet de déboguer ou de programmer les Microcontrôleurs (cela se fait sur les broches RB6 et RB7).

-Capteur de température : un capteur de température TC74 est connecté aux supports 28 et 40 Broches.

- La communication se fait par le protocole I2C sur 2 fils. L'adresse du composant est b'0100 1101'.

-EEPROM externe série: une EEPROM série externe (256Kx8) est connecté aux supports 28 et 40 broches. La communication se fait par le protocole I2C. L'adresse du composant est b'1010 0000'.

- Afficheur LCD : un afficheur LCD 2 lignes x 16 caractères est connecté au support 40 broches.



## Chapitre III: Etude du kit du développement.

Et voici la figure .III.6 qui représente le schéma structurel complet de PICDEM PLUS 2 avec les supports de 18,28 et 40 broches et la figure .III.7 qui représente une photo générale de la carte.

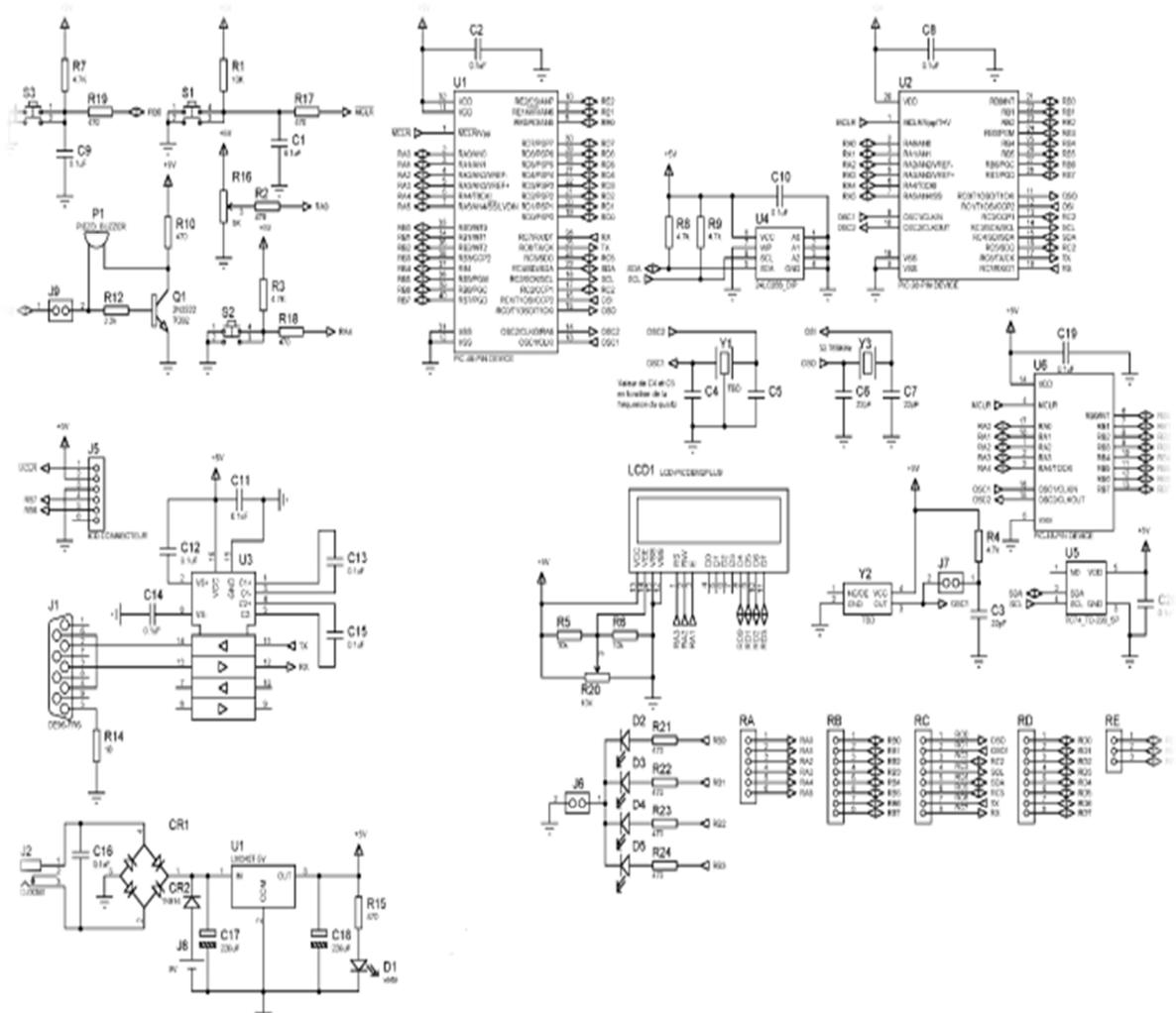


Figure.III.6.: le schéma structurel complet de PICDEM PLUS 2.

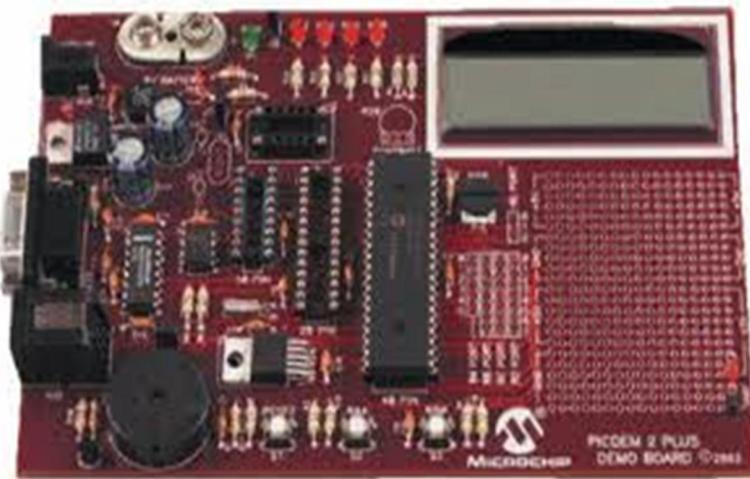


Figure .III.7: Une photo générale de la carte.

### III.4.L'étude du kit de développement à base de pic 16F877:

#### III.4.1.Schéma synoptique : [4][5]

Le schéma synoptique de notre kit du développement est représenté dans la figure III.8.

Notre kit de développement composé de l'unité centrale qui est le pic 16F877 et un clavier de 12 touches pour le traitement et un afficheur Lcd pour l'affichage et de liaison série RS232 pour la communication série et des leds comme des sorties simple et des boutons poussoir comme des entrées simple un programmeur de pic intégré dans le kit.

Le kit est alimenté avec une alimentation stabilisée de 5V et de12V.

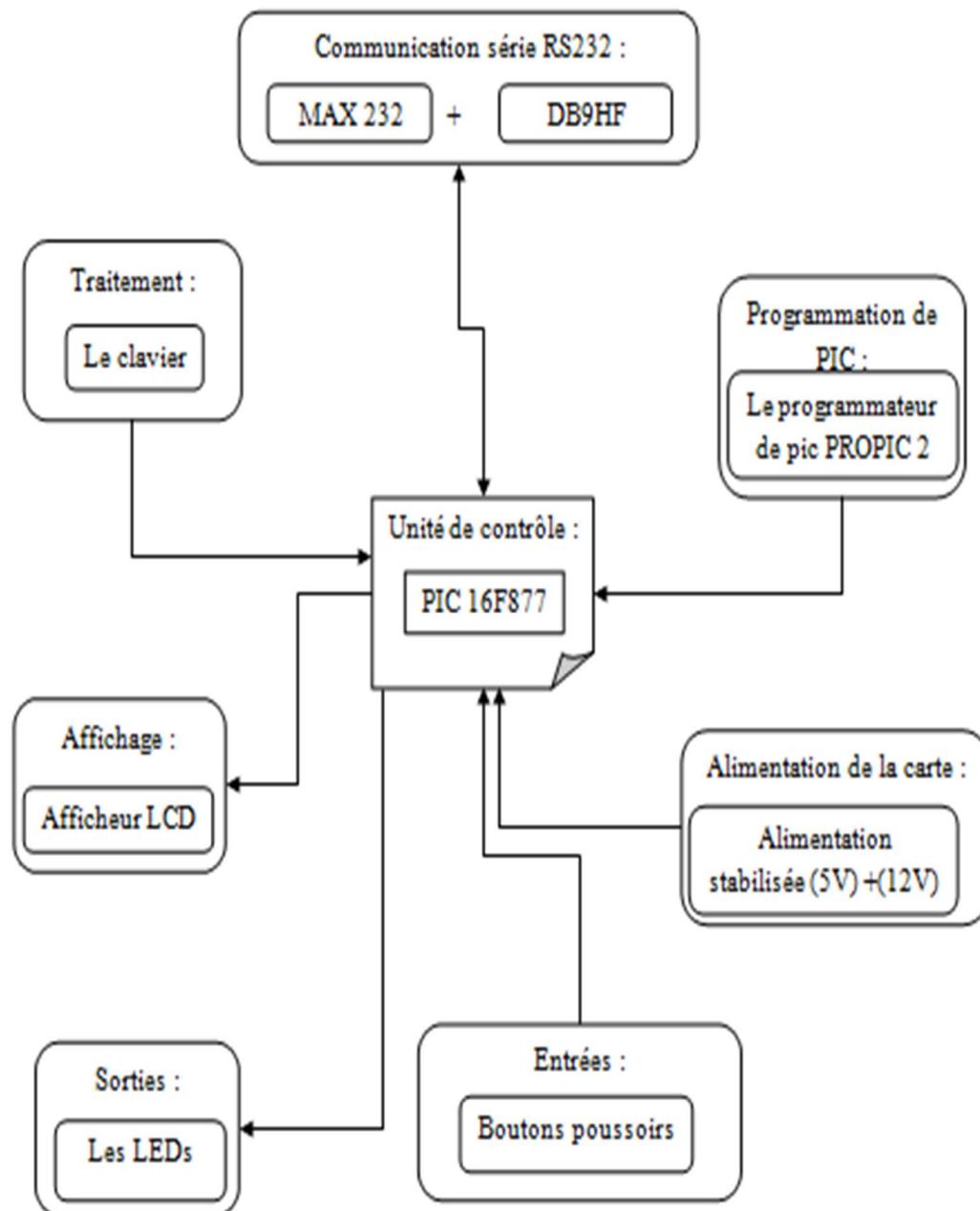


Figure III. 8:Schéma synoptique de kit.



### III.4.2. Unité de traitement Pic 16F877:

Dans nos jours, les mini projets à besoin d'un circuit intégré à pour rôle le traitement de l'information, rapide et de prix moins chère alors que la solution c'est le pic, avec cette condition on à utiliser dans notre projet le pic 16F877 comme l'indique la figure(III.9) suivante :

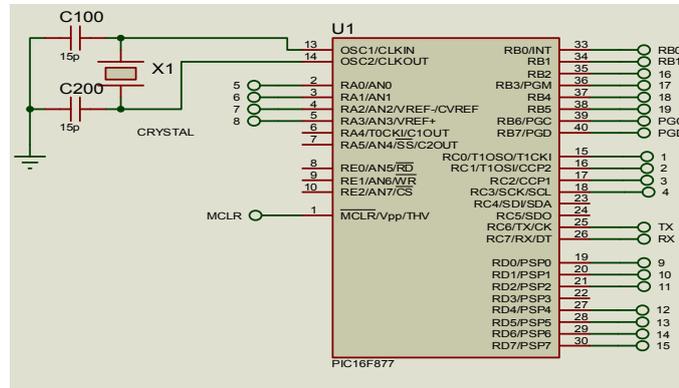


Figure.III.9: connexion de pic 16f877 sur le kit.

### III.4.3. RESET :

#### III.4.3.a. Principe de fonctionnement :

Le RESET est relié au pin MCLR du PIC c'est une entrée de remise à zéro, lorsque cette entrée est mise à l'état bas, le microcontrôleur est réinitialisé : il va exécuter l'instruction se trouvant à l'adresse 00 H.

Mais aussi lorsque le microcontrôleur est mis sous tension il est préférable que RESET soit à l'état logique 0 pendant un temps très court c'est le rôle du circuit RC.

Et la figure.III.10 montre la connexion de RESET sur le circuit.

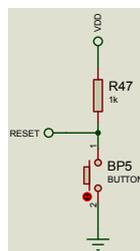


Figure.III.10: la connexion de RESET sur le circuit.

### III.4.4. Clavier:

Le clavier est le périphérique le plus commode pour saisir du texte, mais dans notre kit on va utiliser un clavier numérique à 12 touches matricées pour saisir des numéros.

Le clavier se compose de 12 touches réparties sur une matrice de 4 lignes (chaque ligne contient 4 touches) 4 pistes sont disposées horizontalement (elles correspondant aux 4 lignes de touches) et 3 autres pistes sont disposées verticalement (elles correspondant aux 3 colonnes

## Chapitre III: Etude du kit du développement.

de touches). Chaque touche agit comme un bouton poussoir qui établit le contact entre une des 4 pistes horizontales et une des 3 pistes verticales.

### III.4.4.a. Principe de fonctionnement:

Les claviers sont des commutateurs, qui sont connectés à des lignes et des colonnes. Si la touche "1" est pressé, la colonne 1 est connectée électriquement à la ligne 1 (Row1), si la touche "2" est pressé la colonne 2 (Col 2) avec la ligne 1, et ainsi de suite ...

Ce model est appelée matriciel, c-à-d les lignes et les colonnes forment une matrice de 3\*4.

Comme nous voyons dans la figure III.11.suivante :

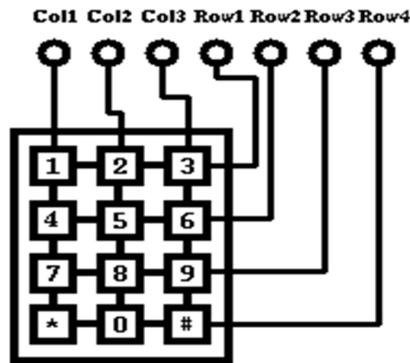


Figure III.11:Les colonnes et les rangées d'un clavier 12 touches.

Pour détecter, dont l'un des 12 touches est pressée, déroulez les trois lignes de colonnes, un par un à la masse (les deux autres lignes de colonnes à plus) et de lire les quatre résultantes rangée lignes.

Si l'une des quatre lignes des lignes est faible, arrêtez de lire plus loin et identifier le code de la clé de la colonne et la rangée d'info.

Comme le montre le tableau III.1 dans la page suivante:



## Chapitre III: Etude du kit du développement.

Colonne			Rangée				Key	
Col 1	Col 2	Col 3	Ran 1	Ran 2	Ran 3	Ran 4	Caractère	Code binaire
0	0	0	1	1	1	1	(aucun)	1111
0	1	1	0	1	1	1	1	0001
1	0	1	0	1	1	1	2	0010
1	1	0	0	1	1	1	3	0011
0	1	1	1	0	1	1	4	0100
1	0	1	1	0	1	1	5	0101
1	1	0	1	0	1	1	6	0110
0	1	1	1	1	0	1	7	0111
1	0	1	1	1	0	1	8	1000
1	1	0	1	1	0	1	9	1001
0	1	1	1	1	1	0	*	1010
1	0	1	1	1	1	0	0	0000
1	1	0	1	1	1	0	#	1011

Tableau III.1:détection des pins de clavier

### III.4.4.b. Connexion du clavier sur la carte :

Le clavier est branché dans le kit comme indique la figure.III.12 suivante:

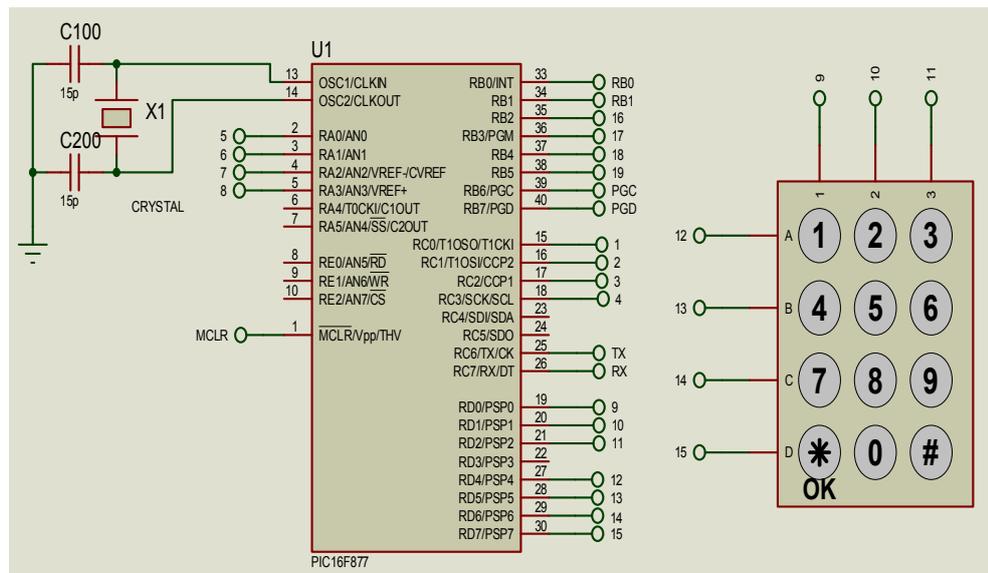


Figure.III.12:connexion de clavier dans le kit.

### III.4.5.Entrées et sortie simple:

La figure III.13 décrit un ensemble de boutons poussoirs relié avec le port C de pic comme entrées et des LEDs relié avec le port A comme sorties.



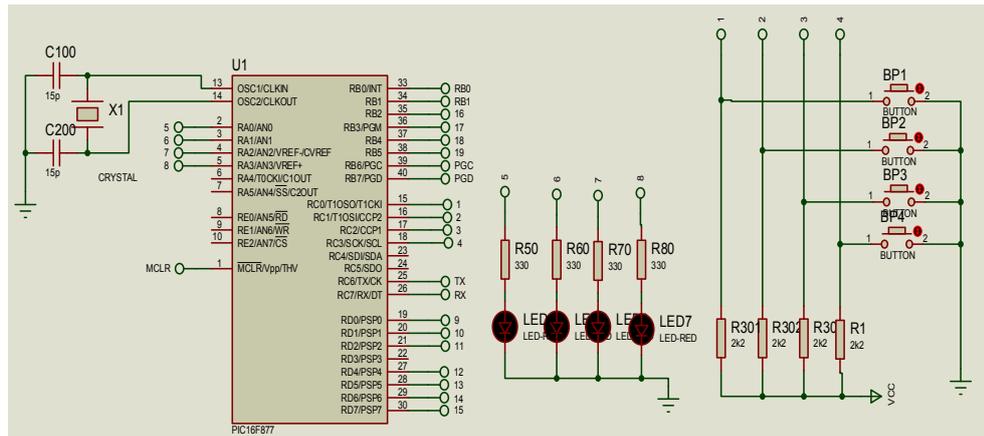


Figure.III.13:connexion du boutons et LED sur le kit.

### III.4.6.Afficheur LCD :

#### III.4.6.a. Présentation :

Dans notre carte on est besoin d'un afficheur LCD, autrement appelé un afficheur à cristaux liquide. Il consomme relativement de 1 à 5 mA et constitué de deux lames de verre, distante de 20  $\mu\text{m}$  environ, sur lesquelles sont dessinées les mers nantisses formant les caractères. L'application entre les deux faces d'une tension alternative basse fréquence de quelques volts (3 à 5V) le rend absorbant. Un afficheur à cristaux liquide ne peut être utilisé qu'avec un bon éclairage ambiant. Son lisibilité augmente avec l'éclairage.

#### III.4.6.b. Principe de fonctionnement : [4]

Dans notre projet on va utiliser le mode 4 bits de l'afficheur LCD. Dans ce mode, seul les 4 bits de poids fort (D4 à D7) de l'afficheur sont utilisés pour transmettre les données et les lires. Les 4 bits de poids faible (D0 à D3) sont alors connectés à la masse, on a donc besoin hors alimentation de sept fils pour commander l'afficheur. Les données sont écrites séquentiellement les quatre bits de poids fort suivi des quatre bits de poids faible. Une impulsion positive d'au moins 450ns doit être envoyée sur la ligne E pour valider chaque demi-octet.

Qu'il soit à une ou deux lignes, un afficheur LCD se présente sous la forme suivante :



Figure.III.14 : photo d'un afficheur LCD.

### Chapitre III: Etude du kit du développement.

Au-dessus de l'écran à cristaux liquides proprement dit, on trouve une série de 14 broches aux rôles indiquées au tableau III.2 suivant :

N° de pastille	Appellation	Fonction
1	Vss	Masse
2	Vdd	Alimentation +5V
3	V0	Contraste afficheur
4	Rs	Sélection commande/donnée
5	R/W	Lecture/écriture
6	E	Validation des données
7	D0	Donnée D0 [poids faible]
8	D1	Donnée D1
9	D2	Donnée D2
10	D3	Donnée D3
11	D4	Donnée D4
12	D5	Donnée D5
13	D6	Donnée D6
14	D7	Donnée D7 [poids fort]

Tableau III.2.l'appellation et fonction des branche de LCD

- Broche 1 : masse.
- Broche 2 : Vdd alimentation +5V.
- Broche 3 : luminosité.
- Broche 4 : sélection commande/donnée.

Sert à dire au module dans quel registre il doit écrire les données présentes sur les broches des données.

Si RS=0 ; le module sait que c'est une instruction et va donc l'écrire dans le registre d'instruction.

Si RS=1 ; le module sait que c'est un code caractère et va donc l'écrire dans le registre de donnée.

- Broche 5, R/W : sélection du mode lecture ou écriture :

Sert à dire au module, si les broches des données sont en <<entrée>> ou en <<sortie>>, autrement dit si les données sont <<écrite>> vers le module ou <<lues>> depuis le module.

0 → écriture.

1 → lecture.

- Broche 6, E : validation des données.

Sert à valider les données ou les instructions écrites sur les broches des données de module. La validation se fait sur le front descendant.



## Chapitre III: Etude du kit du développement.

- Broches 7 à 14 : utilisées pour le transfert des données ou des instructions. Le transfert peut se faire sur 8 bits, toutes les broches sont alors utilisées, ou sur 4 bits, dans ce cas, seules les broches 11 à 14 sont utilisées.

### III.4.6.c. Connexion de l'afficheur sur la carte :

L'afficheur de notre kit se connecte comme le montre la figure.III.15 suivante:

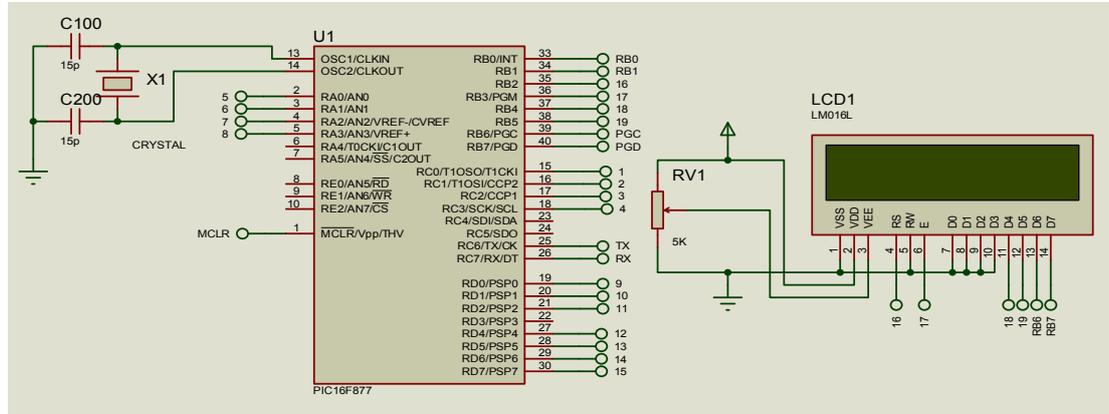


Figure.III.15: connexion du LCD dans la carte.

### III.4.7. La liaison RS 232 :

#### III.4.7.a. Présentation :

Les liaisons séries permettant la communication entre deux systèmes numériques en limitant le nombre de Fils de transmission.

La liaison série à la norme RS 232 est utilisée dans tous les domaines de l'informatique (exp. port de communication com1 et com2 des PC permettant la communication avec des périphériques tels que modem et souris). Elle est de type asynchrone, c'est à dire qu'elle ne transmet pas de signal d'horloge. Et tout ça a été présenté en détail dans le chapitre II précédent.

#### III.4.7.b.MAX232 : [5]

Le MAX232 est un standard depuis longtemps, il permet de réaliser des liaisons RS232 et des interfaces de communication, il amplifie et met en forme deux entrées et deux sorties TTL/CMOS vers deux entrées et deux sorties RS232, la connexion est réalisée avec un DB9. Le pic 16F877 utilise les niveaux 0v et 5v pour définir respectivement les bits : 0 et 1. La norme RS 232 définit des niveaux de +12v et -12v pour établir ces mêmes niveaux Nous avons donc besoin d'un circuit (driver de bus) chargé de convertir les niveaux des signaux entre PIC et PC alors que la solution c'est le circuit MAX232.

Le montage de MAX232 est indiquée la figure III.16.



## Chapitre III: Etude du kit du développement.

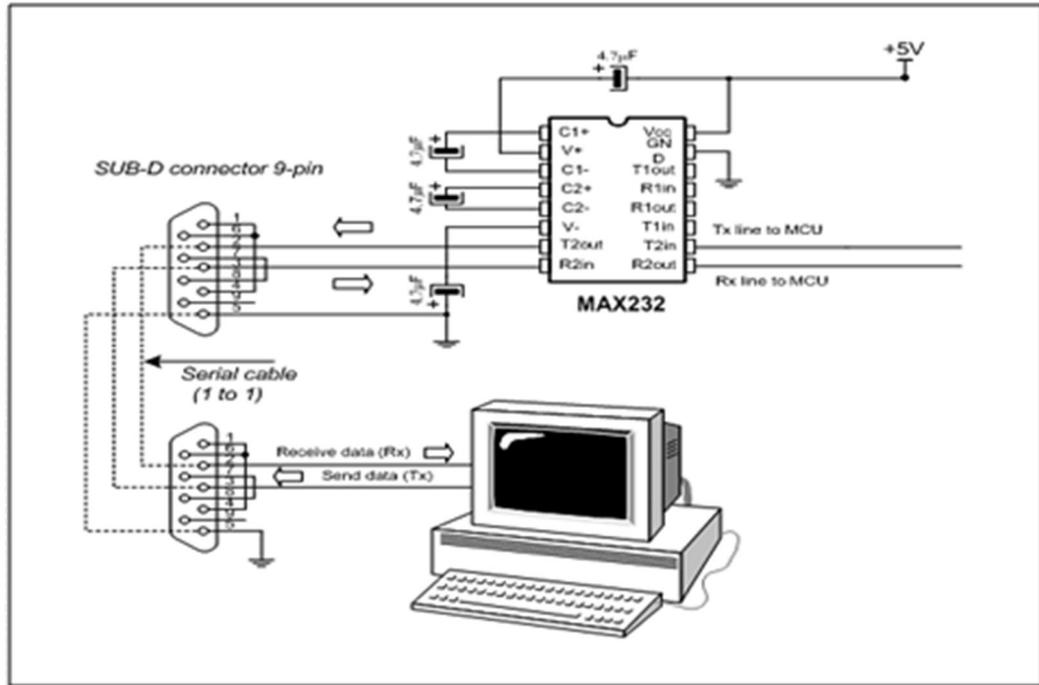


Figure .III.16: montage de MAX232.

Ce circuit dispose de deux blocs dénommés T1 et T2 qui convertissent les niveaux entrés en 0v et 5v en signaux sortis sous +12v et -12v. Les entrées de ces blocs sont donc dirigés vers le microcontrôleur pic et les sorties sont connectées sur le port RS 232.

Deux blocs dénommés R1 et R2, qui convertissent les niveaux entrés en +12v /-12v en signaux sortis sous 0v/5v.

Les entrées de ces blocs sont donc connectées sur le port RS232, les sorties sur le PC.

Et voici donc l'architecture interne de MAX232 indiquée dans la figure III.17 suivante

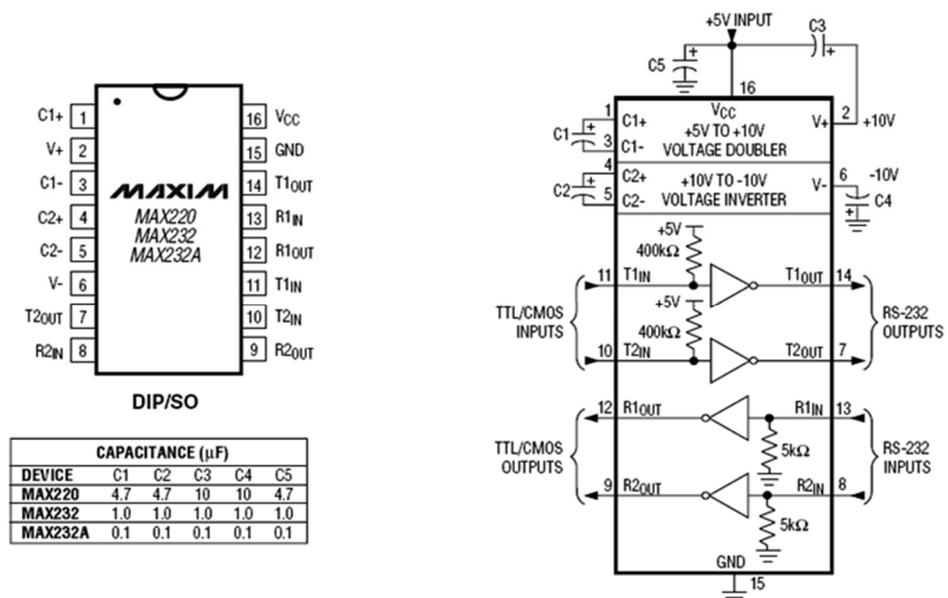


Figure.III.17:l'architecture interne de MAX232.



## Chapitre III: Etude du kit du développement.

### III.4.7.c. Connexion de MAX232 sur la carte :

Ce schéma (figure.III.18) montre que le MAX232 relie avec le DB9 (utilisé les pines T2OUT et R2IN) et le Pic 16F877 (utilisé les pines T2IN et R2OUT).

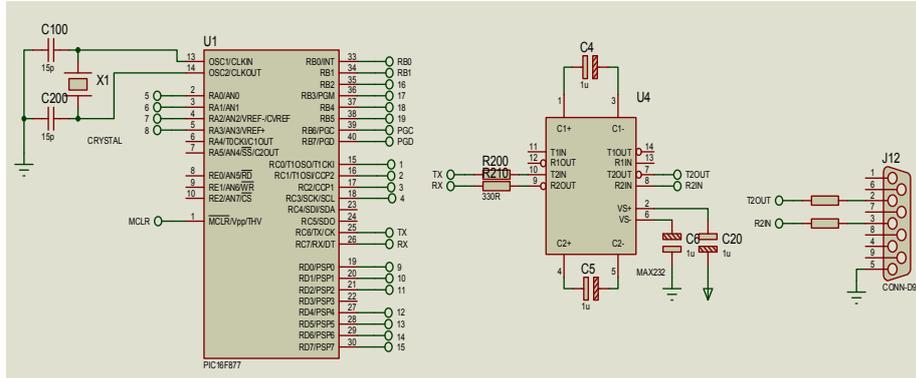


Figure.III.18:connexion de liaison RS232 dans la carte.

### III.4.8. Programmateur de pic :[8][5]

Notre kit est à base de microcontrôleur PIC, et pour qu'il sera fonctionnelle il faut programmer le microcontrôleur PIC, c'est pour cette raison qu'on a réalisé un programmeur.

On peut réaliser un programmeur isolé à la carte de développement, mais pour faciliter la tâche à l'utilisateur et pour permettre un gain de temps dans le développement, on intègre ce programmeur dans le kit comme le montre la figure III.19 suivante:

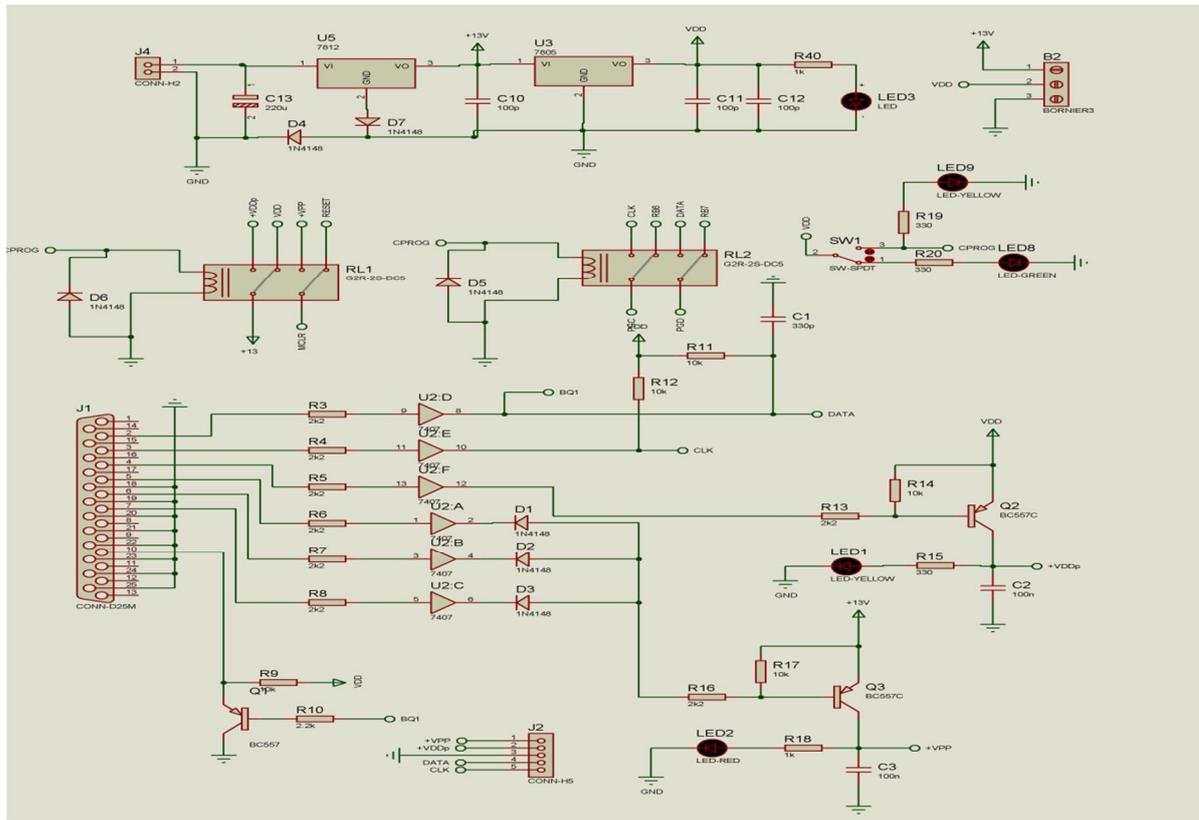


Figure III.19: connexion du programmeur dans la carte.



### **Chapitre III: Etude du kit du développement.**

Ce programmeur est un programmeur série en circuit ou s'appelle aussi modèle ICSP (In Circuit Serial Programming) de modèle PROPIC2 de Microchip.

Le programmeur PROPIC2 commandé par le port parallèle, opère une programmation en série comme l'imposent les microcontrôleurs PIC. Sept lignes de l'interface Centronics sont mises à contribution: six sorties (D0 à D5), et une entrée (ACK). Les sorties, protégées par les résistances R3 à R8, sont tamponnées par les six amplificateurs A à F de U2. Les données sont transmises en série sur la ligne D0 pour l'écriture, la résistance R11 force la broche « DATA » à l'état haut au repos. Afin de pouvoir lire le contenu des mémoires, les signaux « DATA » polarisent le transistor Q1 à travers la résistance R10. Lors d'une tension positive, Q1 reste bloqué, et la résistance R9 porte « ACK » à l'état haut ; une information de niveau bas bloque Q1 qui force « ACK » à la masse. Le signal d'horloge « CLK » est véhiculé par la sortie D1, et forcé à l'état haut au repos via la résistance R12.

La ligne D2 du port parallèle se charge de l'alimentation du composant à programmer. Le transistor Q2 est commandé à travers sa résistance de base R13 et bloqué au repos par la résistance R14 reliée au positif. Dans ce cas de figure, aucune tension d'alimentation ne circule sur les supports de circuits à programmer. En portant sa base au niveau bas, Q2 devient passant, et alimente positivement la broche « +VDDp » découplée par le condensateur C2. La LED jaune « LED1 » atteste de cet état, elle est limitée en courant par sa résistance R15. Les lignes D3, D4 et D5 du port parallèle ont pour mission de commuter la tension de programmation sur la broche « +VPP ». Les diodes anti-retour D1, D2 et D3 jouent le rôle d'une fonction 'OU' à trois entrées afin de commander le transistor Q3 via sa résistance de base R8. Le principe de fonctionnement est identique à celui de Q2, la LED rouge LED3 visualise la présence de la tension de programmation sur les supports. Remarque : Pour faciliter le passage de la mode programmation vers le fonctionnement normal on a ajouté le SW1 qui est aussi connecté avec les deux relais RL1 et RL2.

#### **III.4.9. Schéma complet du kit du développement :**

Après qu'on a vu précédemment les différents blocs et composants en détail et séparément voici donc la figure .III.20 qui montre le schéma général de notre kit du développement à base de pic 16F877.



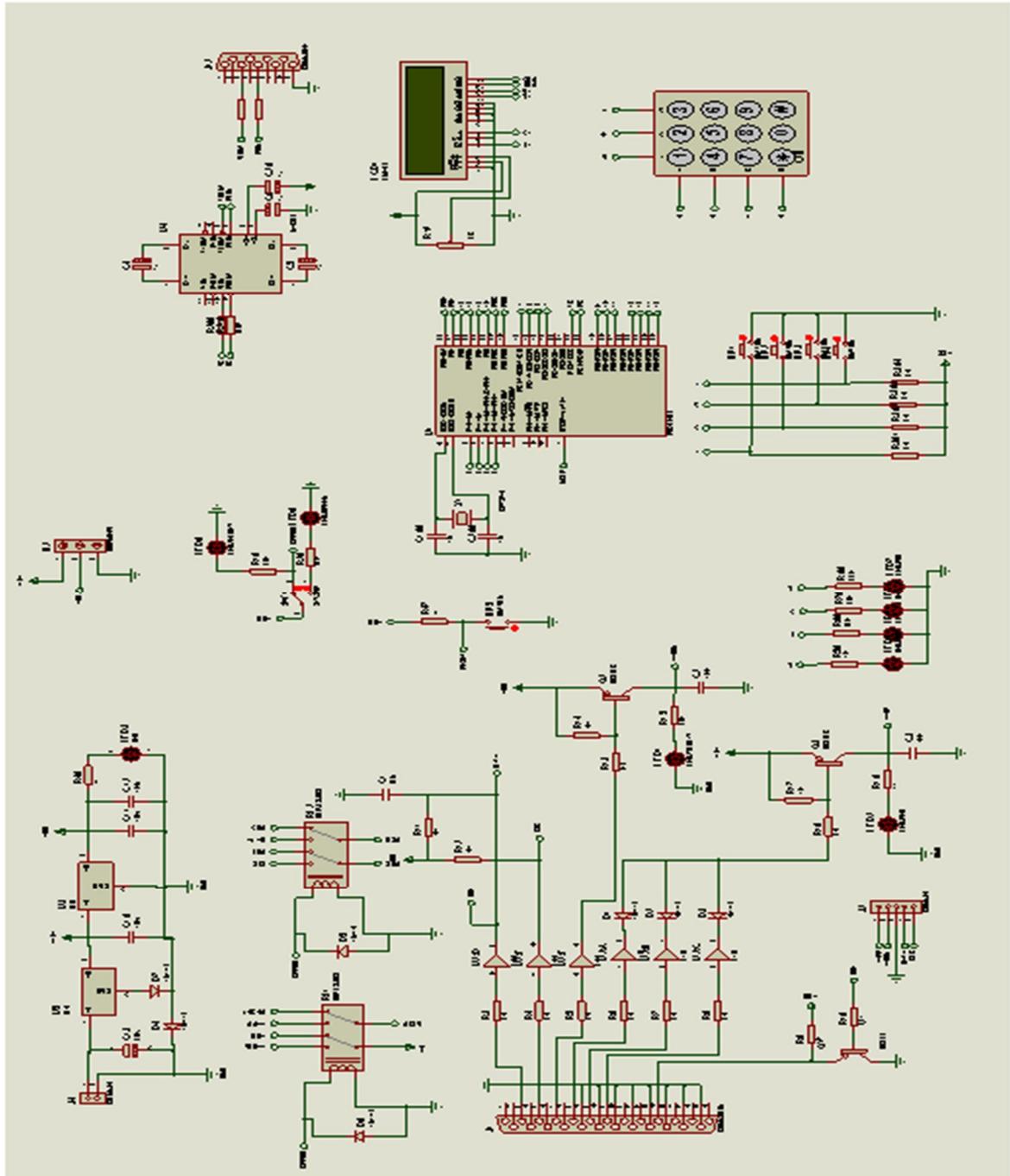


Figure. III.20:Schéma complet de kit du développement à base de pic 16F877.

**III.5.Conclusion:**

Dans ce chapitre on peut constater le rôle principal de l'étude théorique,

Le kit de développement est basé sur le microcontrôleur PIC16f877 comme élément principal, qui va permettre de développer des différentes applications avec bien sur des d'autres périphériques.

Et après cette étude théorique, alors maintenant on peut passer à la réalisation de Kit du développement.



CHAPITRE IV :  
RÉALISATION DE  
KIT DU  
DEVELOPPEMENT

## ***CHAPITRE IV: Réalisation du kit du développement.***

---

### **IV.1.Introduction:**

Après une étude générale des différents éléments constituant notre carte électronique on passe maintenant à la simulation et la réalisation physique de notre projet.

Dans cette partie la programmation du PIC nécessite à comprendre le logiciel de programmation. Alors on va présenter le logiciel, puis on va introduire les organigrammes de calcul. En plus on va utiliser d'autres logiciels et outils pour réaliser notre Kit du développement.



## CHAPITRE IV: Réalisation du kit du développement.

### IV.2. Réalisation software:

La programmation du PIC nécessite à comprendre le logiciel de programmation -dans notre cas c'est le « MicroC », Alors on a présenté le logiciel, puis on va introduire les organigrammes de calcul.

#### IV.2.1. Présentation du logiciel « Mikro PRO for PIC » : [1]

Mikro PRO pour PIC est un plein de fonctionnalités compilateur C ansi pour les appareils PIC de Microchip. Il est la meilleure solution pour développer du code pour les appareils PIC. Il dispose d'IDE intuitive, puissant compilateur avec des optimisations qui nous aideront dans notre travail, et il est fourni avec le fichier d'aide complet et beaucoup de prêt-à-utiliser des exemples destinés à nous aider à démarrer en un rien de temps.

#### IV.2.2. Création du projet en Mikro C PRO for PIC : [3]

Pour créer un nouveau projet, et de saisir et de compiler un programme on passe par les étapes suivantes :

- Lancement du programme Mikro PRO for PIC : double cliquer sur l'icône de logiciel.
- Création d'un nouveau projet : aller à Project »New Project »Next. Voir figure (IV.1).

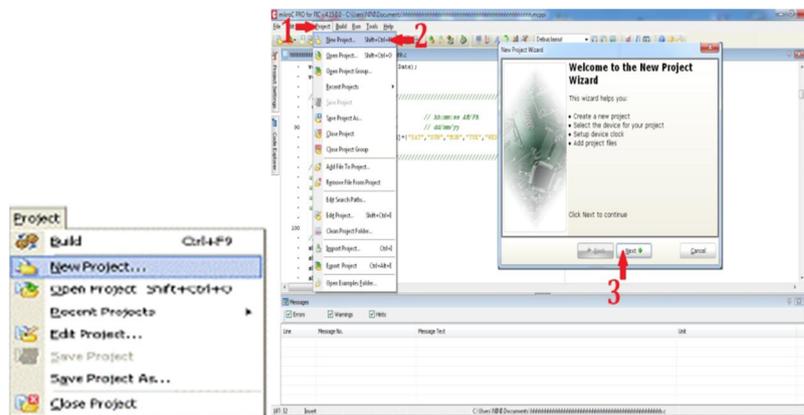


Figure. IV.1.Méthode de création d'un nouveau projet.

- Choix de pic : Voir figure .IV.2.

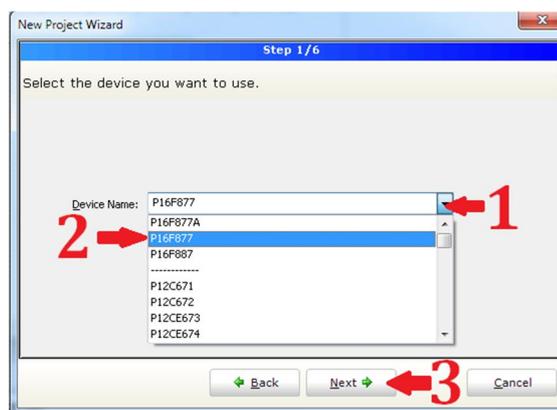


Figure. IV.2.Choix de PIC.



## CHAPITRE IV: Réalisation du kit du développement.

- Choix de fréquence : Voir figure .IV.3.

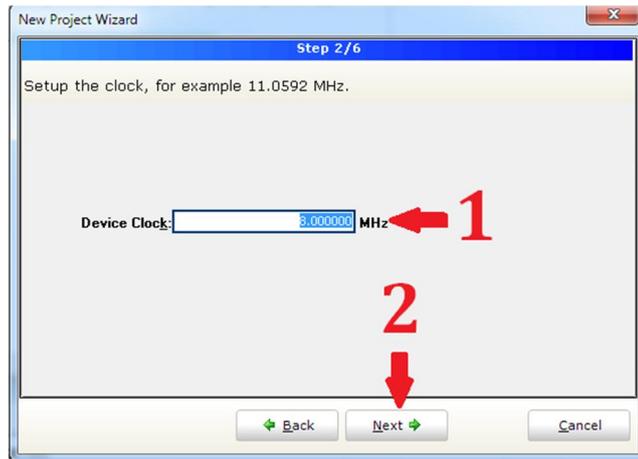


Figure. IV.3.Choix de fréquence.

- Nommer le projet et enregistré : Voir figure.IV.4.

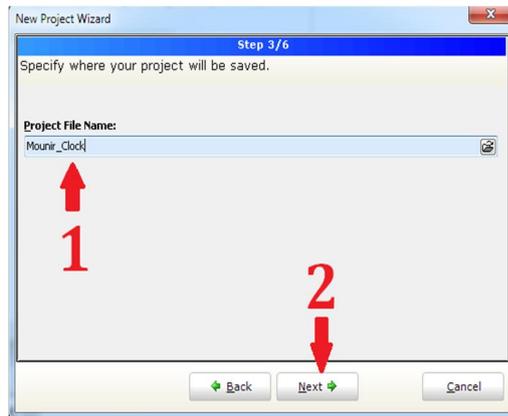


Figure .IV.4.Nommer le projet.

- L'ajoute des fichiers supplémentaires : Voir figure. IV.5.

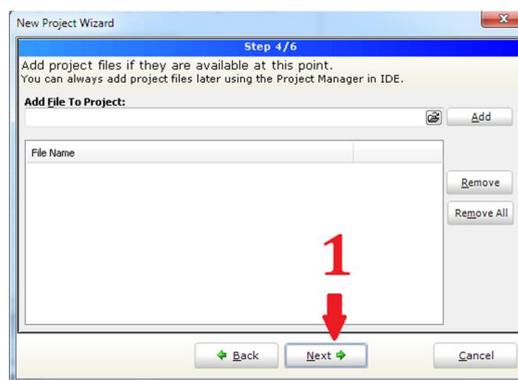


Figure .IV.5.Fenêtre d'ajoute des fichiers supplémentaires.

- La sélectionne de la bibliothèque : Voir figure. IV.6.



## CHAPITRE IV: Réalisation du kit du développement.



Figure .I.V.6.La sélectionne de la bibliothèque.

- La fin de la création d'un projet : Voir figure. IV.7.

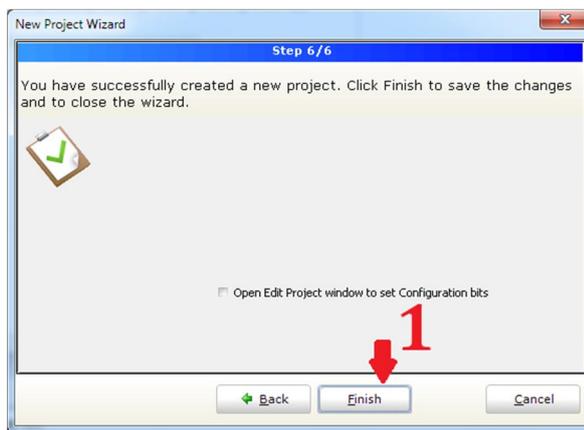


Figure. IV.7.La fin de la création d'un projet avec succès.

- Saisie du programme: Voir figure. IV.8.

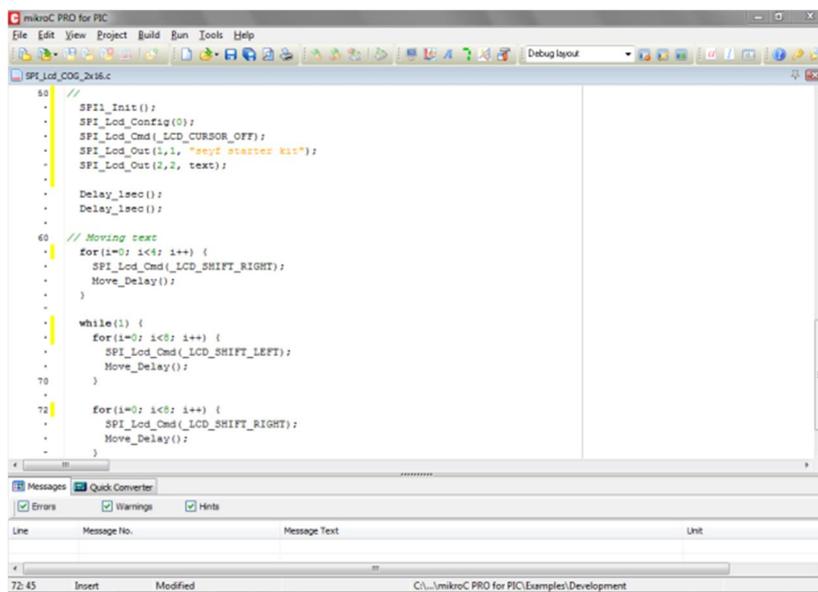


Figure. IV.8.Exemple d'un programme.



# CHAPITRE IV: Réalisation du kit du développement.

- Compilation: Voir figure. IV.9.

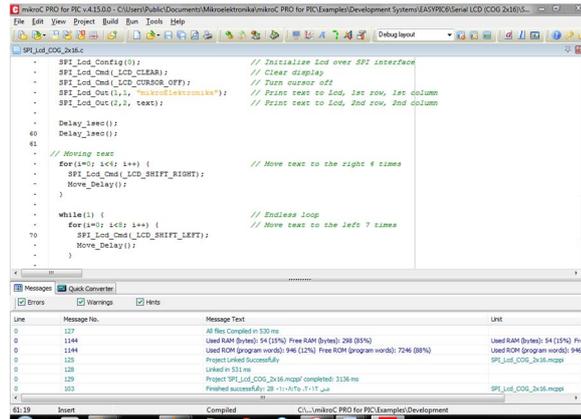


Figure. IV.9.Compilation avec succès.

## IV.3.Présentation de l'ICprog :

ICprog est un programme permet de transférer un fichier compilé (hex) vers un pic ou un mémoire de type EEPROM.fenêtre principale de l'ICprog dans la figure IV.10.suivante:

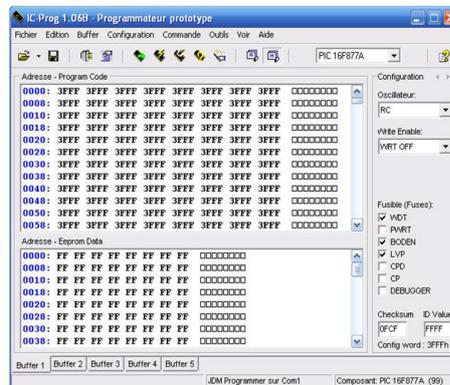


Figure. IV.10.fenêtre principale de l'ICprog.

### IV.3.2. Configuration de l'ICprog:

Depuis le menu « configuration » puis « hardware » sélectionner selon le programmeur (dans notre cas PROPIC2) une des options dans la figure IV.11

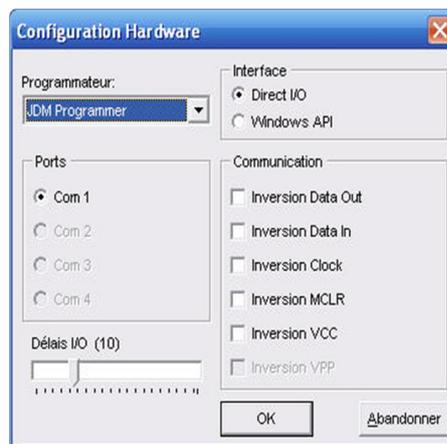


Figure. IV.11.Fenêtre de configuration hardware.



## CHAPITRE IV: Réalisation du kit du développement.

- Pour sélectionner un composant aller dans « configuration » puis « composant » .voir figure. IV.12.

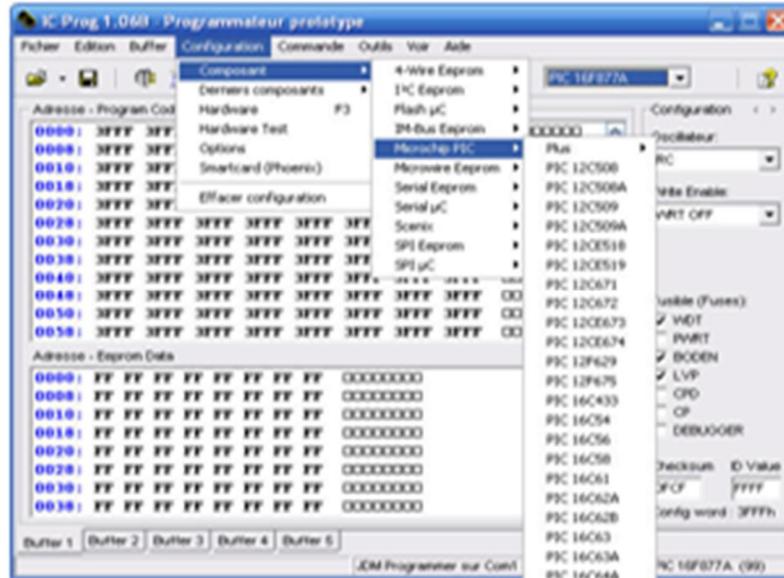


Figure. IV.12.fenêtre sélection de composant.

### IV.4. Réalisation du schéma du Kit :

#### IV.4.1. Logiciel adopté :

Pour la création du schéma de la carte on utilisé le logiciel ISIS de la société LABCENTER ELECTRONICS.

#### IV.4.2Réalisation avec logiciel Proteus : [2]

##### IV.4.2.a. Généralités sur Isis Proteus :

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.

##### IV.4.2.b. La simulation numérique :

Il faut bien évidemment commencer par la saisie du schéma de la carte qu'on souhaite réaliser :

Pour saisir le schéma, il faut créer un nouveau projet puis placer les composants qui doivent être sélectionnés à partir de la bibliothèque des composants : Voir la figure I.13.



## CHAPITRE IV: Réalisation du kit du développement.

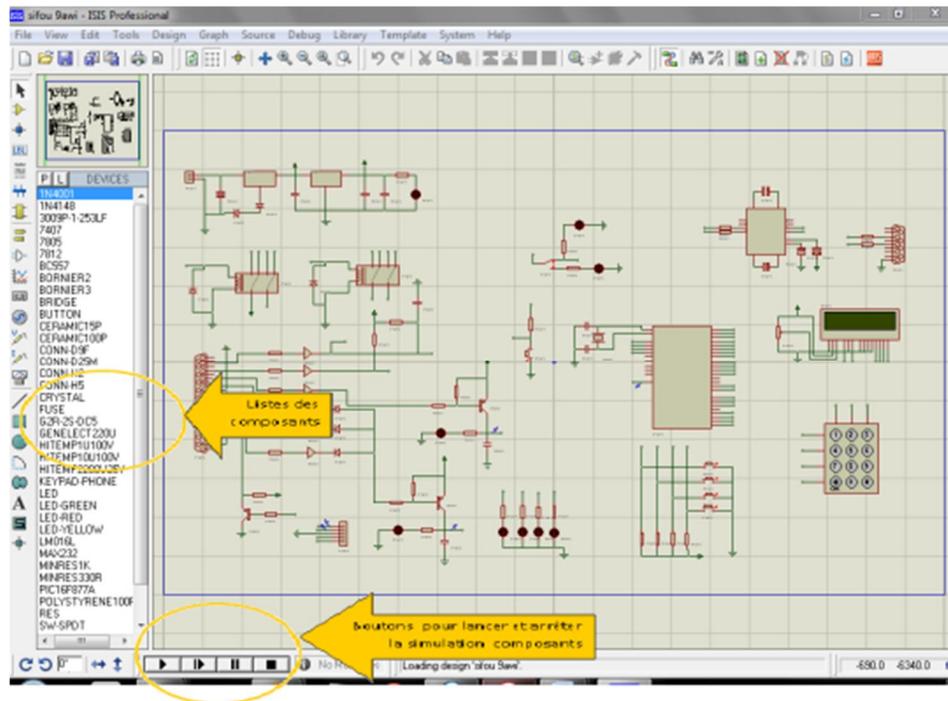


Figure. IV.13. Ecran d'accueil d'ISIS.

Dans le menu « source », sélectionnez la commande « Ajout/suppression fichiers source » puis sélectionnez le fichier ASM du programme à utiliser dans la simulation.

Puis choisissez « Outil de génération de code » pour générer un fichier HEX.

Editez le PIC16F877 et ajoutez le fichier HEX dans le champ « programme file ».

Appuyez sur le bouton jouer du « magnétoscope ». La barre des messages doit indiquer le temps écoulé depuis le lancement de l'animation.

Afficher les fenêtres « Registers » « Source Code » « Data memory » par l'intermédiaire du menu « Debug » pour contrôler le fonctionnement du programme, détecter et corriger les erreurs.

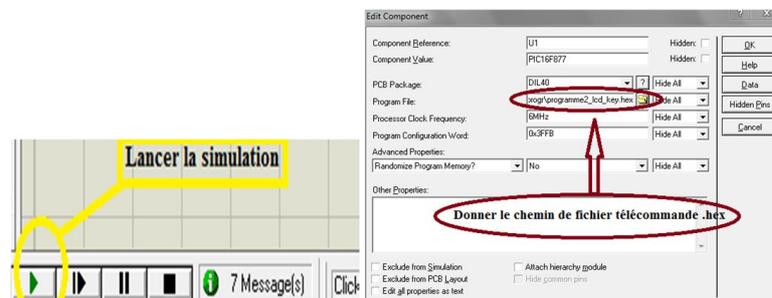


Figure. IV.14. : Saisie d'un schéma sous ISIS

En second lieu, on doit associer le programme de simulation au PIC.



## CHAPITRE IV: Réalisation du kit du développement.

Et voici le schéma de kit avec Isis. Voir figure .IV.15.

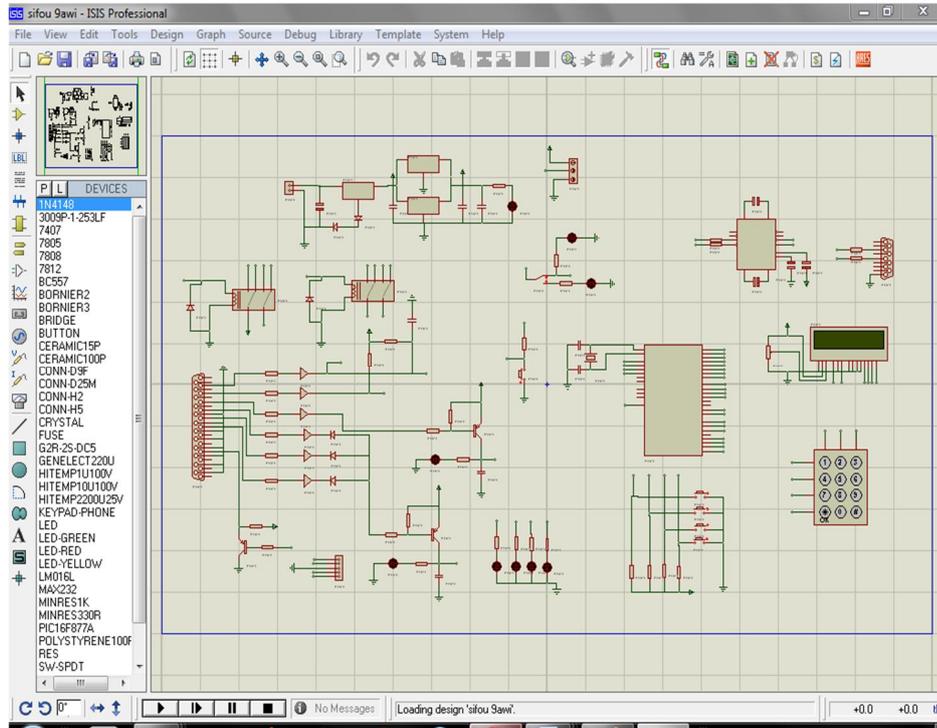


Figure .IV.15.Schéma de kit (Isis).

### IV.5. Réalisation du schéma de routage :

#### IV.5.1. Logiciel utilisé :

Pour le schéma de routage on a choisi d'utiliser le logiciel ARES de la même compagnie LABCENTER ELECTRONICS. C'est un logiciel très pratique à manipuler.

#### IV. 5.1. a. Ecran d'accueil :

En exécutant notre logiciel on obtient l'affichage (figure. IV.16) suivant :

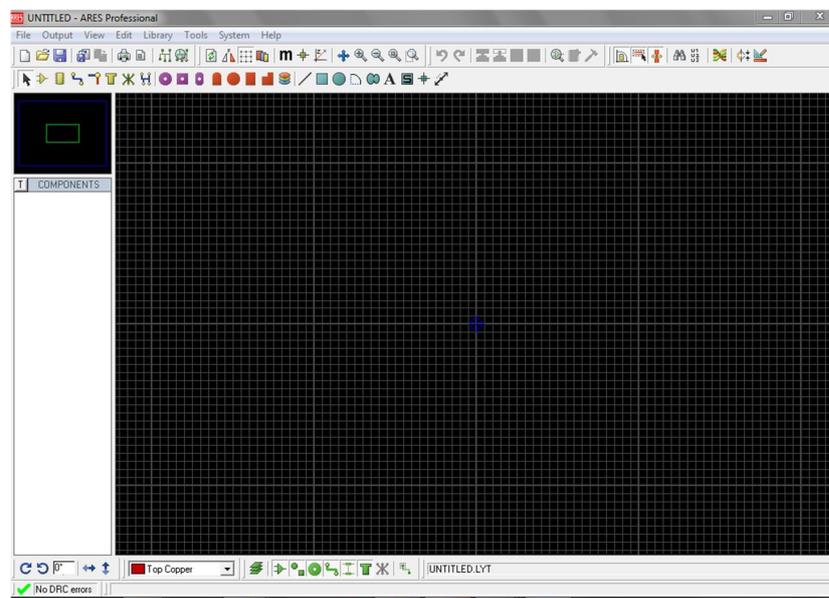


Figure. IV.16: Ecran d'accueil d' Ares.



## CHAPITRE IV: Réalisation du kit du développement.

### IV.5.1.b. Insertion des composantes :

Pour insérer les composantes de notre carte il suffit de sélectionner l'élément voulu et de cliquer sur l'écran noir par le bouton droit de la souris, on choisi la commande « PLACE » puis « Component » et ainsi de suite.

Et voila, le schéma de kit avec Ares. Voir la figure IV.17.

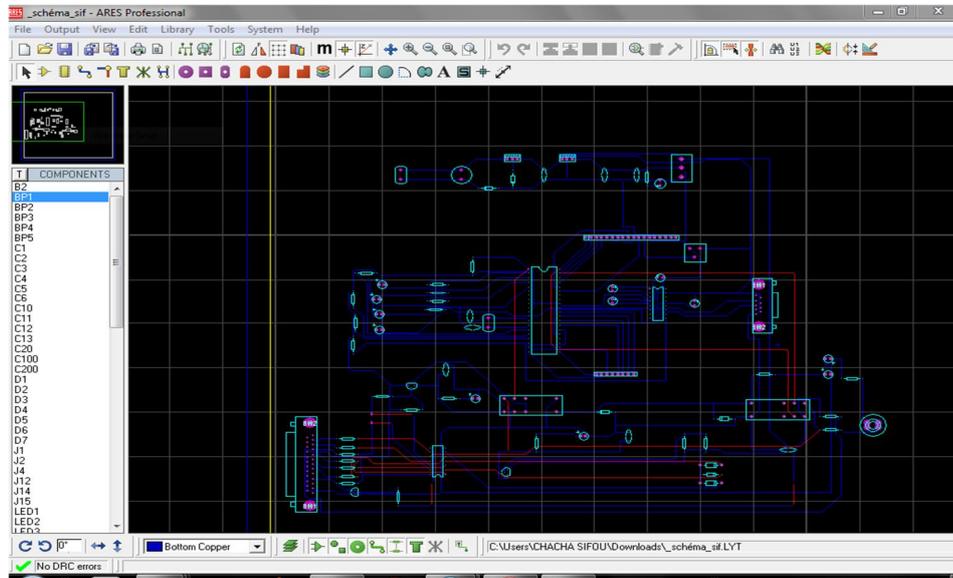


Figure IV.17:schéma de kit (Ares).

### IV.6.Schéma de routage:

Voici maintenant le schéma de routage de notre kit représentée dans la figure.IV.18

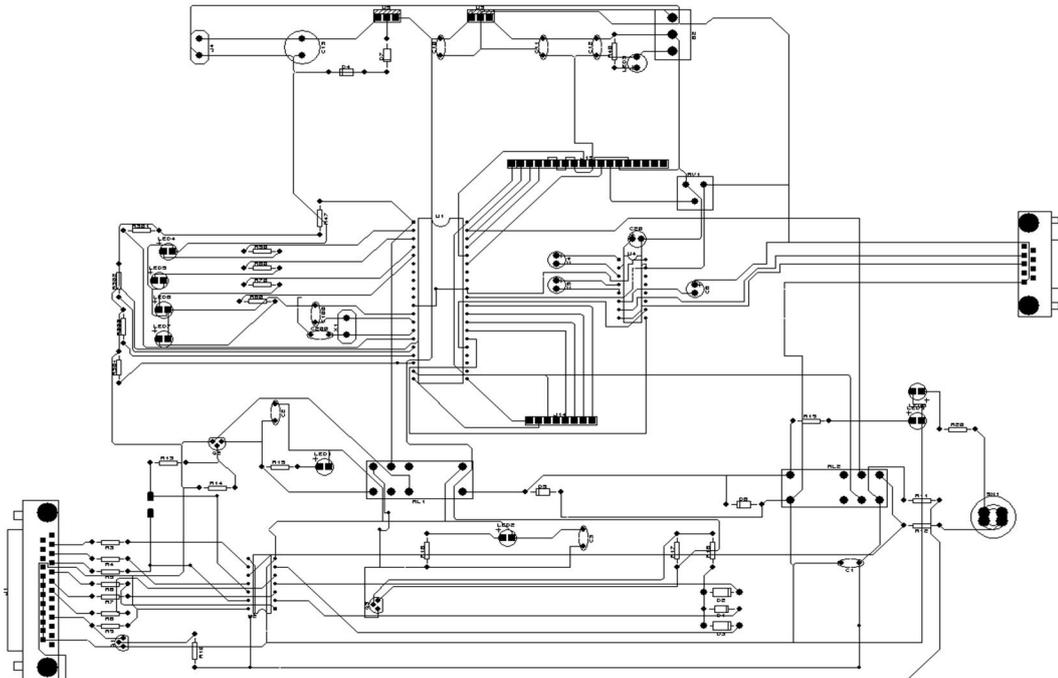


Figure IV.18: Schéma de routage de kit(cuivre et composants).



## CHAPITRE IV: Réalisation du kit du développement.

### IV.7. L'algorithmes des programmes :

#### IV.7.1. Commande des entrées sorties simples:

##### IV.7.1.a. Commande de boutons poussoirs simple et leds :

Le but est de réaliser la commande d'un éclairage à partir des simples boutons poussoir. En appuyant sur un bouton poussoir, on allume l'éclairage. Et s'éteindre automatiquement après 5s. Voici l'organigramme de l'application (figure. IV.19).

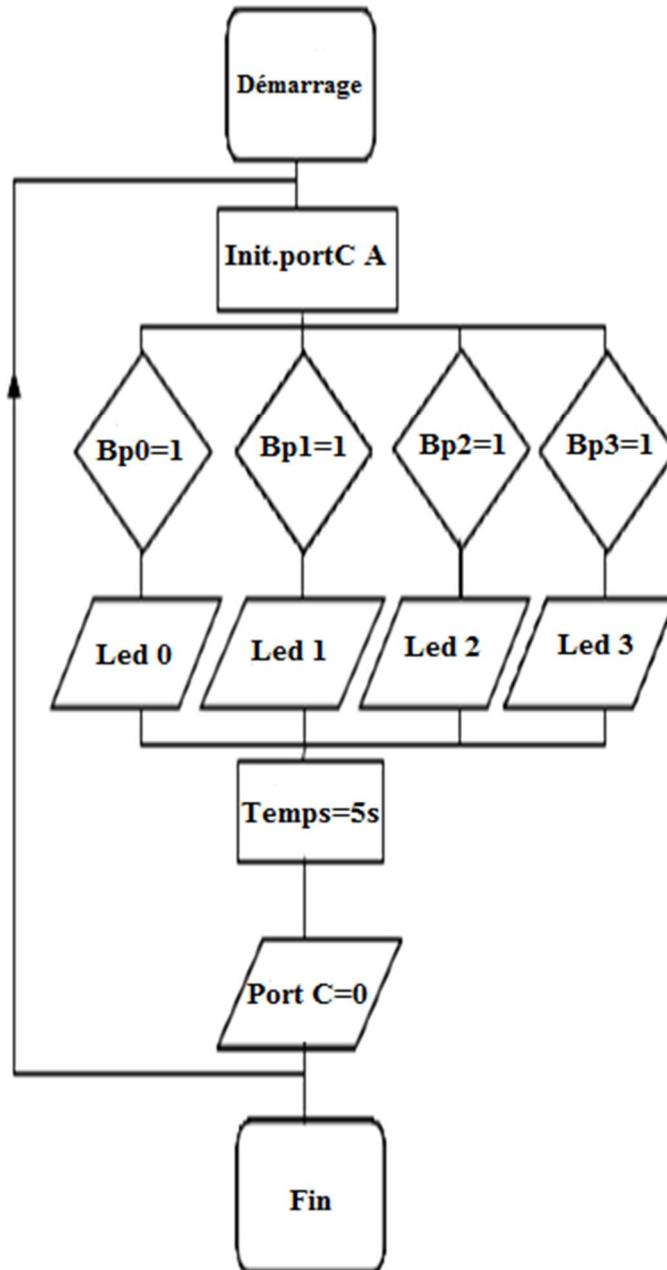


Figure. IV.19. Commande de boutons poussoirs simple et leds.

Bp0.Bp1.Bp2.Bp3 : boutons poussoir.



## CHAPITRE IV: Réalisation du kit du développement.

### IV.7.1.b.commande de bouton poussoir intelligent:

Le but est de réaliser la commande d'un éclairage à partir d'un simple bouton poussoir. En appuyant une première fois sur le bouton poussoir, on allume l'éclairage. Une seconde pression l'éteindra. Cependant, si on ne pense pas à éteindre les lampes, une minuterie les éteindra automatiquement. Voici l'organigramme de l'application (figure. IV.20).

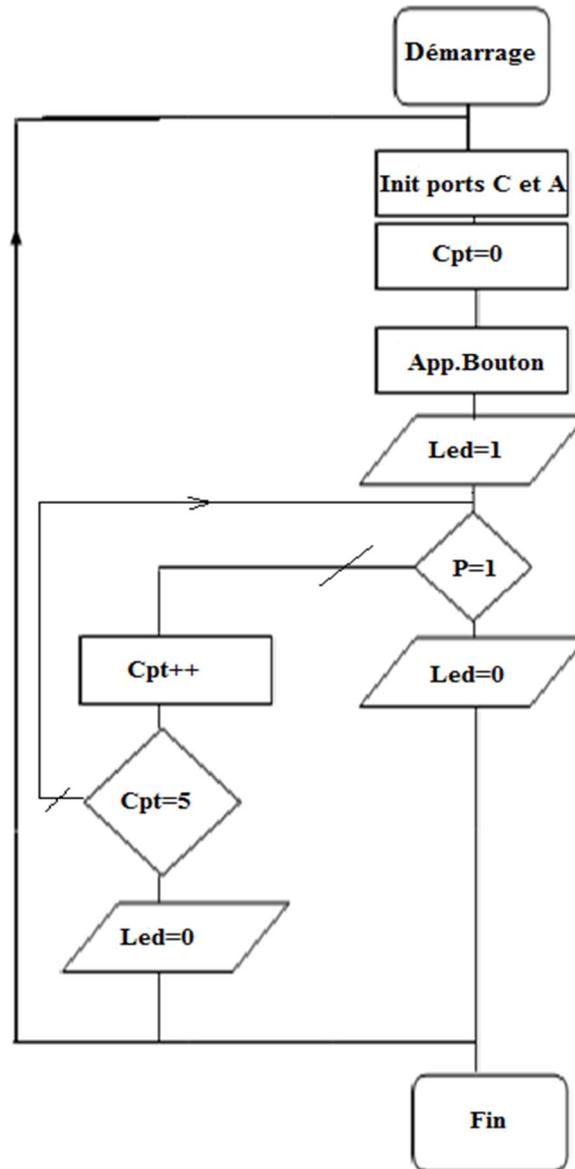


Figure. IV.20. commande de bouton poussoir intelligent.

Cpt : compteur.

P : bouton poussoir.



## CHAPITRE IV: Réalisation du kit du développement.

### IV.7.2.Utilisation de clavier et LCD:

Le but est d'utiliser le clavier avec LCD.

En appuyant sur une touche de clavier et afficher la valeur du touche dans LCD. Voici l'organigramme de l'application (figure. IV.21).

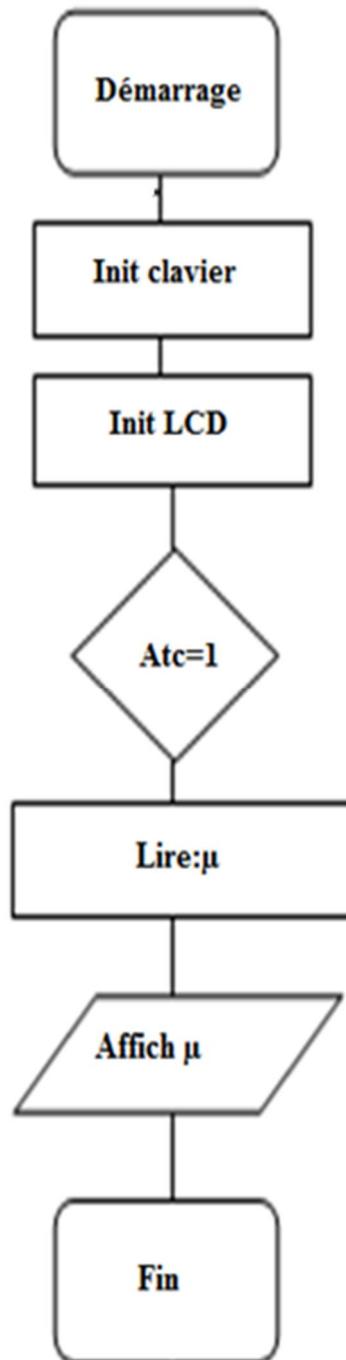


Figure. IV.21. Utilisation de clavier et LCD.

Atc : appuyez sur une touche sur le clavier.

μ : numéro de touche sur le clavier



## CHAPITRE IV: Réalisation du kit du développement.

### IV.7.3.Utilisation de RS232:

Le but est de réaliser la communication du PIC avec le PC via une liaison RS232.

Il s'agit d'envoyer un message du PC vers l'afficheur LCD à travers le PIC en utilisant un programme qui s'exécute sur le PC pour la configuration des COM.

- Utilisation d'une liaison RS232 pour établir la connexion entre les deux périphériques.

Voici l'organigramme de l'application (figure IV.22).

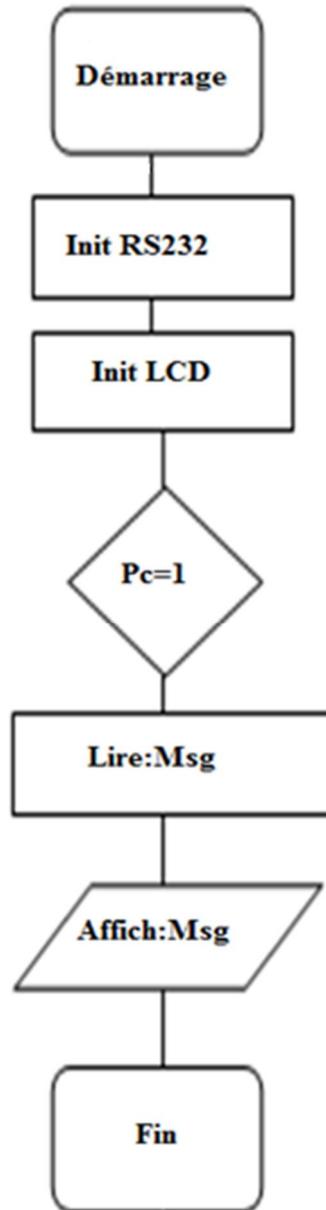


Figure. IV.22. Utilisation de RS232.

Msg : message.



## ***CHAPITRE IV: Réalisation du kit du développement.***

---

### **IV.8. Conclusion**

Dans ce chapitre on peut constater le rôle principal de système pratique, Le kit de développement est basé sur le microcontrôleur PIC16f877 comme élément essentiel, nous avons programmé le pic et simuler le circuit.

La réalisation pratique de montage était pour nous une expérience très enrichissante du fait que nous avons vécu un cas réel de conception.

Les expériences que nous avons menées durant ce chapitre nous ont montré que l'étude théorique et l'étude par simulation étaient très proches de la réalité pratique.



## *Conclusion générale :*

---

### **Conclusion Générale :**

Les microcontrôleurs permettent de réaliser des applications diverses qui peuvent servir dans différents domaines industriels ou pour les projets de fin d'étude de les étudiants.

Dans notre projet nous avons traité les différentes interfaces qu'on nous pouvons de les réaliser a base du pic et les simulées dans plusieurs application sur le kit de développement de pic 16f877 qu'on à réalisé.

Ce projet nous a beaucoup appris sur les microcontrôleurs et leurs programmations. Il a été mené d'une recherche bibliographique bien détaillée.

L'élaboration de ce travail, nous a permis d'approfondir nos connaissances théoriques et d'acquérir une bonne expérience au niveau de la réalisation pratique.

Lors de cette manipulation, nous avons réalisé une Kit de développement à base de PIC. Ce projet nous a permis d'enrichir nos connaissances dans le domaine électronique :

- ❖ La conception des cartes et des circuits imprimés.
- ❖ La réalisation des cartes d'interface.
- ❖ La programmation des microcontrôleurs PIC.
- ❖ La mise en pratique des connaissances théoriques en électronique et en programmation en Mikro C.
- ❖ La manipulation des différents modules .

Et tous ceci dans le but d'étudier les microcontrôleurs Pic et leur applications en général avec un pic de la famille 16Fxxx et de réaliser un objet physique qui peut nous aidons à réaliser ces applications.

Et ce kit (ou un autre kit de ce genre) est très outil pour les étudiants qui ont un projet de fin d'étude qui basé sur un pic 16F877 parce qu'il va de diminuer plusieurs difficulté a propos leur travail.



## ***Bibliographie:***

---

### **Bibliographie:**

#### **Chapitre I: Présentation de microcontrôleur 16F877**

[1]CHRISTIAN Tavernier\_: «Les microcontrôleurs PIC ‘recueil d’application’ » 3eme edition, DUNOD, Collection TE ,2005.

[2] CHRISTIAN Tavernier : «Microcontrôleurs PIC 10,12,16 ‘description et mise en œuvre’ » DUNOD, Collection TE,2007.

[3]GERARD Samblancat: « Progressez avec les microcontrôleurs PIC » DUNOD, Collection ESTF,2006.

[4]ABIDI.H:«Carte de développement pour micro contrôleur de la famille PIC »2006.

[5]Rjeb.B, Waz .R : « Projet de réalisation d’une maquette didactique a base de pic 16F877 »2007.

[6]<http://www.abcelectronique.com/bigonoff/>

[7] [www.didel.com/pic/Architecture.pdf](http://www.didel.com/pic/Architecture.pdf)

[8][www.technologuepro.com/](http://www.technologuepro.com/)

[9]Datasheet du 16F877.pdf

#### **Chapitre II: Le standard RS-232**

[1]Alain, Ursula Bouteveille-Sanders« La liaison RS232» 2002.

[2]Etude de la liaison série RS232.pdf

[3]Description d'une liaison.pdf

[4] Liaison série RS232.pdf

[5] Liaison série asynchrone.pdf

[6] Les liaison série.pdf

#### **Chapitre III: Étude de kit de développement.**

[1] <http://www.rennes.supelec.fr/ren/fi/elec/mcu/pic/maq877/maq877.html>

[2]<http://www.technologuepro.com/>

[3]<http://www.mikroelektronika.com/>

[4] ABIDI.H:«Carte de développement pour micro contrôleur de la famille PIC »2006.

[5] Rjeb.B, Waz .R : « Projet de réalisation d’une maquette didactique a base de pic 16F877 »2007.



## ***Bibliographie:***

---

[6]Le système de développement.pdf

[7] Electronique numérique programmée: « Mise en œuvre et programmation des microcontrôleurs». Pdf

[8]Programmeur pic ICSP.pdf

[9]CHRISTIAN Tavernier\_: «Les microcontrôleurs PIC ‘recueil d’application’ » 3eme édition, DUNOD, Collection TE ,2005.

### **Chapitre VI : Réalisation du kit de développement.**

[1]<http://www.mikroe.com/>.

[2]Saisie de schéma ISIS 6.doc.

[3]Creating The first Project in microC for PIC.pdf



# Liste de figures

---

## La liste des figures par chapitre:

### Chapitre I

Figure I.1 : Schéma fonctionnel du PIC 16F877.....	7
Figure I.2: Schéma fonctionnel simplifier de pic 16F877.....	8
Figure. I. 3: Brochage du PIC16f877.....	8
Figure I.4: Le schéma du PORT A.....	14
Figure I.5: Le schéma du PORT B.....	14
Figure I.6: Le schéma du PORT C.....	14
Figure I.7 : schéma du PORT D.....	15
Figure I.8: Le schéma du PORT E.....	15
Figure I.9 : Plan Mémoire pour les instructions (Code programme).....	16
Figure. I.10. Schéma d'oscillateur.....	18
Figure I.11. Cycle de conversion.....	20
Figure .I.12. Module du convertisseur.....	20

### Chapitre II

Figure. II.1. Le schéma fonctionnel de la liaison RS232.....	23
Figure. II.2: La liaison unidirectionnelle.....	24
Figure .II.3 : La liaison unidirectionnelle.....	24
Figure .II.4: La liaison half duplex.....	25
Figure .II.5: Le chronogramme d'une liaison half duplex.....	25
Figure .II.6: La liaison full duplex.....	25
Figure. II.7: Le chronogramme d'une liaison full duplex.....	25
Figure .II.8: le chronogramme de protocole de RS232.....	25
Figure. II.9. Chronogramme de protocole matériel.....	26
Figure. II.10:Chronogramme de l'adaptation à base de MAX 232.....	27
Figure .II.11 : le brochage d'un câble RS232 croisé.....	27
Figure .II.12 : le port série D9 et la distribution des pôles.....	28



## Liste de figures

---

### Chapitre III:

Figure.III.1: l'organisation du développement.....	32
Figure.III.2.Organisation de développement (programmation seule).....	33
Figure.III.3:chaine de développement d'un système de développement.....	33
Figure .III.4: photo de la carte EASY PIC2.....	36
Figure.III.5:L'utilisation de PICDEM PLUS 2 pour programmation.....	36
Figure.III.6.: le schéma structurel complet de PICDEM PLUS 2 .....	38
Figure .III.7: Une photo générale de la carte.....	38
Figure .III.8 : Schéma synoptique .....	39
Figure.III.9:connexion de pic 16f877 sur le kit .....	39
Figure.III.10: la connexion de RESET sur le circuit.....	40
Figure III.11:Les colonnes et les rangées d'un clavier 12 touches.....	40
Figure.III.12:connexion de clavier dans le kit.....	41
Figure.III.13:connexion du boutons et LED sur le kit.....	42
Figure.III.14 : photo d'un afficheur LCD.....	42
Figure.III.15: connexion du LCD dans la carte.....	44
Figure .III.16: montage de MAX232.....	45
Figure.III.18:connexion de liaison RS232 dans la carte.....	46
Figure III.19: connexion du programmeur dans la carte.....	46
Figure. III.20:schéma complet de kit du développement à base de pic 16F877.....	48

### Chapitre IV:

Figure. IV.1.Méthode de création d'un nouveau projet.....	50
Figure. IV.2.Choix de PIC.....	51
Figure. IV.3.Choix de fréquence.....	52
Figure .IV.4.Nommer le projet.....	52
Figure .IV.5.Fenêtre d'ajoute des fichiers supplémentaires.....	52
Figure .I.V.6.La sélectionne de la bibliothèque .....	53



## Liste de figures

---

Figure. IV.7.La fin de la création d'un projet avec succès.....	53
Figure. IV.8.Exemple d'un programme.....	53
Figure. IV.9.Compilation avec succès.....	54
Figure. IV.10.fenêtre principale de l'Icprog.....	54
Figure. IV.11.Fenêtre de configuration hardware.....	54
Figure. IV.12.fenêtre sélection de composant.....	55
Figure. IV.13. Ecran d'accueil d'ISIS.....	56
Figure. IV.14. : Saisie d'un schéma sous ISIS.....	56
Figure .IV.15. Figure .IV.15.Schéma de kit (Isis).....	57
Figure. IV.16: Ecran d'accueil d'Ares.....	57
Figure IV.17:schéma de kit (Ares).....	58
Figure IV.18: Schéma de routage de kit(cuivre et composants). ....	58
Figure. IV.19. Commande de boutons poussoirs simple et leds.....	59
Figure. IV.20. commande de bouton poussoir intelligent.....	60
Figure. IV.21. Utilisation de clavier et LCD.....	61
Figure. IV.22. Utilisation de RS232.....	62



## *Liste des tableaux:*

### **Chapitre I:Présentation de microcontrôleur.**

Tableau I.1. Tableau comparatif.....	6
Tableau. I.2. Caractéristique de PIC 16F877.....	9
Tableau. I.3. les pins de PIC16F877.....	10
Tableau. I.4. Valeur de diviseur pour les fréquences du quartz et le pic.....	19

### **Chapitre II: Le standard RS-232.**

Tableau .II.1: les niveaux de tension avant et après l adaptation.....	27
Tableau. II.2:les pôles de D9.....	28

### **Chapitre III : Etude de kit de développement.**

Tableau III.1:détection des pins de clavier.....	42
Tableau III.2.l'appellation et fonction des branche de LCD.....	44



# Annexe

---

## Annexe :

Hyper terminale est un interface de communication série a partir le port com avec les peripherique externe

Voila les etapes de l utilisation de l interface.

1.Double clic pour cette interface aparaisse



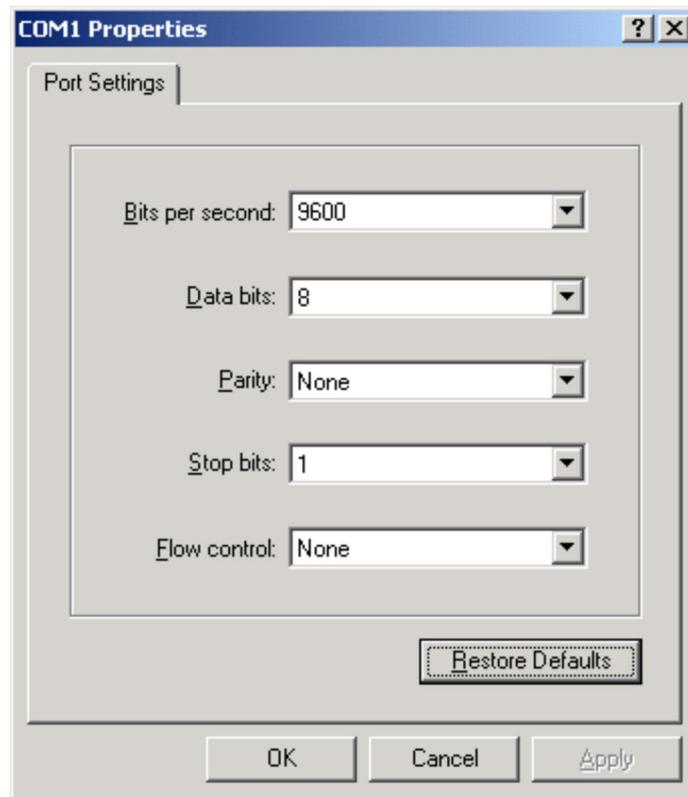
Choix de com.



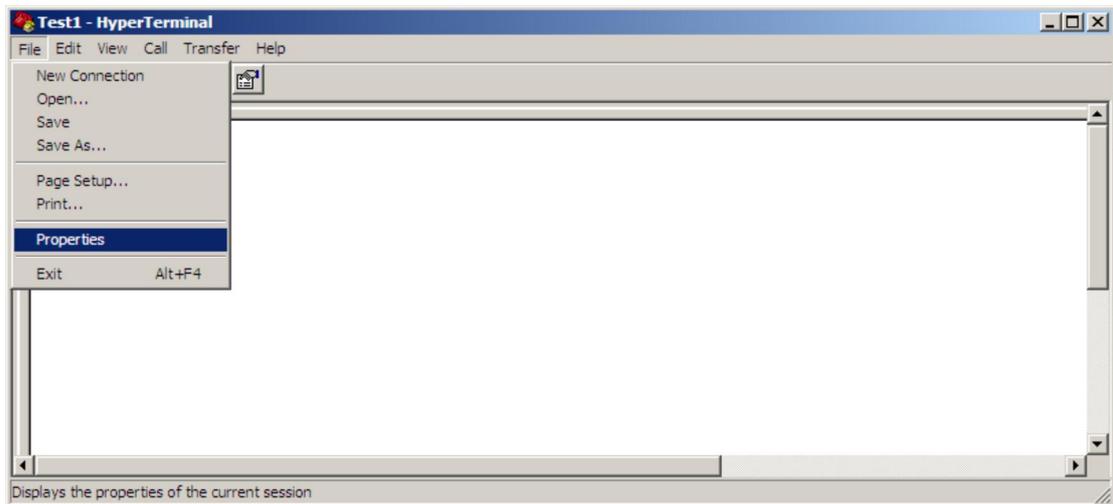
# Annexe

---

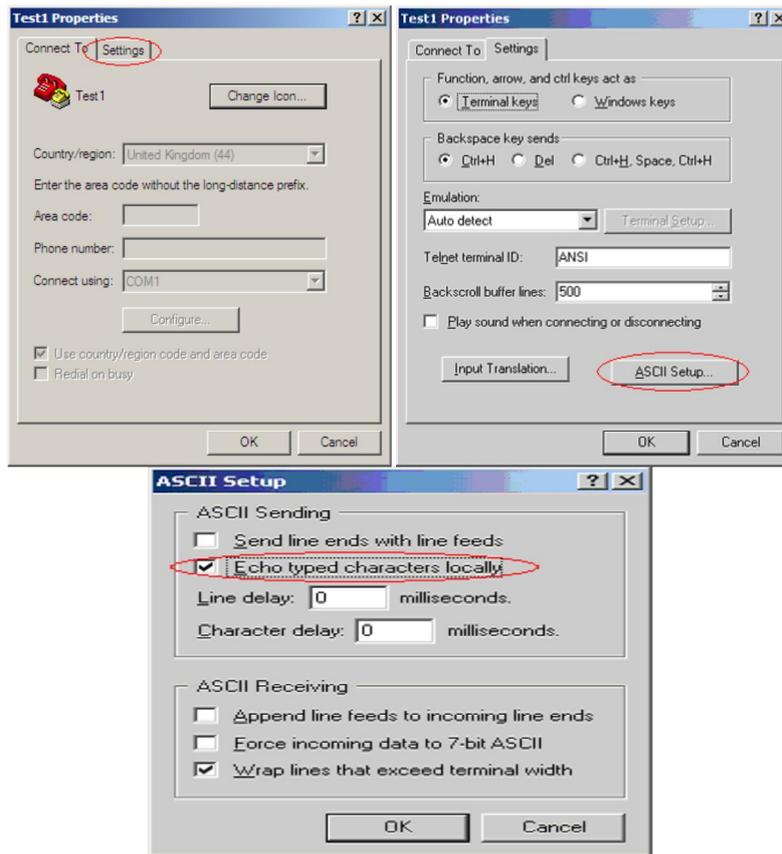
Configuration des paramètres de ports



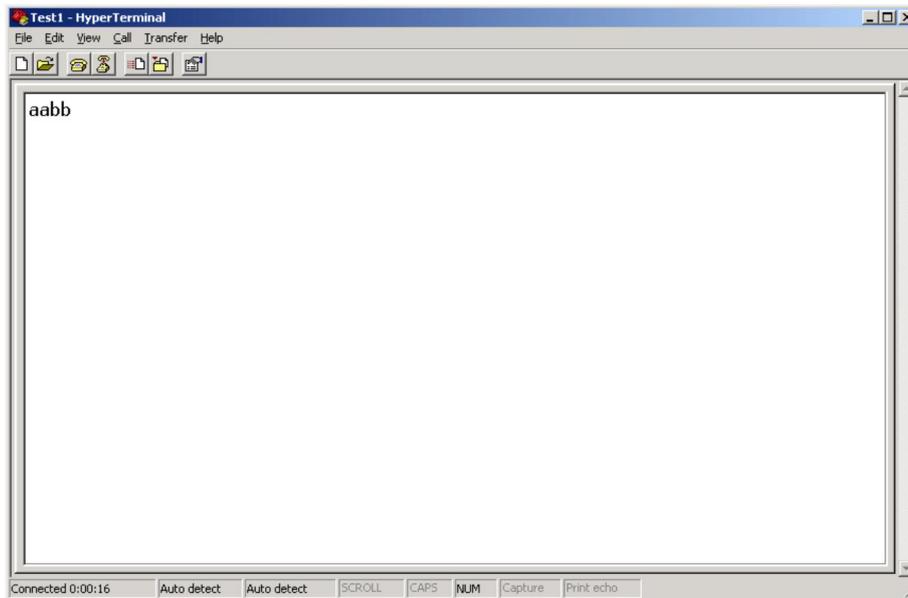
Choix propriétés.



# Annexe



Suives ces étapes pour afficher les caractères entrés instantanément



Maintenant l'interface est prêt pour le communication communication.