

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 : électronique
Option : télécommunication

Réf:

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

MASTER

Thème

Mise en œuvre du protocole I2C
Dans un environnement à
microcontrôleur Microchip (PIC16F877)

Présenté par :
Aberkane Mounir
Soutenu le : 05 Juin 2013

Devant le jury composé de :
Mr. Guesbaya.T
Mr. Debilou.A
Mlle.Handaoui.M

MCA
MCA
MAA

Président
Encadreur
Examineur

Année universitaire : 2012 / 2013

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 : électronique
Option : télécommunication

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

MASTER

Thème

Mise en œuvre du protocole I2C
Dans un environnement à microcontrôleur
Microchip (PIC16F877)

Présenté par :

Aberkane Mounir

Avis favorable de l'encadreur :

Debilou Abderrazak

signature

Avis favorable du Président du Jury

Dr. Guesbaya.T

Signature

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 : électronique
Option : télécommunication

Thème :

Mise en œuvre du protocole I2C Dans un environnement à microcontrôleur Microchip (PIC16F877)

Proposé par : **Debilou Abderrazak**
Dirigé par : **Debilou Abderrazak**

RESUMES (Français et Arabe)

Les protocoles de communication sont dévers et varient entre les appareils électroniques et se diffère selon leur utilisation.

Dans notre projet, nous avons utilisé le protocole I2C dans un environnement à microcontrôleur PIC 16F877 pour contrôler les circuits séries-parallèle est changer de commander une matrice graphique à diode LED extensible.

تتعدد بروتوكولات الاتصال بين الاجهزة الالكترونية وتختلف حسب استعمالاتها .
في مشروعنا هذا استخدمنا البروتوكول I2C في بيئة المتحكم (PIC 16F877) من اجل مراقبة الدارة
المتكاملة (PCF8574) للتحكم في مصفوفة من الصمامات الثنائية التي ينبعث منها الضوء (LED) قابلة
للتمدد .

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Au nom de dieu clément et miséricordieux

Dédicace

*Avant tout, je tiens à remercier le bon dieu, et l'unique qui m'offre le courage
et la volonté nécessaire pour affronter les différentes difficultés de la vie,
Je dédie ce modeste travail*

A ma mère.

A mon père.

A mes frères et mes sœurs.

A toute ma famille.

*A mes amis surtout Ismail Sofiane et mes collègues de la
promotion 2012-2013*

« Electronique ».

A mes féaux amis (chaque un à son nom).

A tous les musulmans dans le monde entier.

M.ABERKANE

Remerciements

Nous rendons nos profondes gratitudee à dieu qui nous a aidés à réaliser ce modeste travail.

Nous exprimons nos profondes gratitudee à nos parents pour leurs encouragements, leurs soutiens et pour les sacrifices qu'ils ont enduré.

Nous remercions l'encadreur Dr. Dehilou Abderrazak pour les efforts qu'il a déployés, pour nous aider, conseiller, encourager et corriger.

Nous tenons à remercier les membres de jury d'avoir accepté d'examiner notre travail.

Nous remercions aussi tout le corps enseignant et administratif qui a contribué à notre formation universitaire.

Sans oublier tous nos amis.

M.ABERKANE

LISTE DES TABLAUX

Tableau 1 : Comparaison des technologies RISC et CISC.....	16
Tableau 2 : Les caractéristiques générales de PIC16F877	22
Tableau 3 : Bibliothèque du MikroC.....	40
Tableau 4 : Les éléments d'un PCF8574.....	47

LISTE DES FIGURES

Figure.I.1: Architecture matérielle d'un système embarqué.....	3
Figure.I.2: La loi G. Moore pour les processeurs Intel.....	6
Figure.I.3: Radar ACC 77Ghz.....	9
Figure.I.4: Fonctionnalité de radar ACC.....	11
Figure.II.1 : Architecture de Von Neumann.....	13
Figure.II.2 : Architecture de Harvard.....	13
Figure.II.3 : Structure de Harvard modifiée.....	14
Figure.II.4: Organisation Processeur pipeline.....	17
Figure.II.5: Les différentes familles de PIC.....	18
Figure.II.6: exemple de l'identification PIC16F877.....	19
Figure.II.7: Architecture interne du PIC16F877	20
Figure.II.8: Brochage du PIC16F877.....	23
Figure.II.9: Schéma d'horloge externe.....	24
Figure.II.10: Le reset et l'horloge HS(Crystal).....	24
Figure.III.1: Structure d'E/S d'un module I2C.....	29
Figure.III.2: Exemple de condition de départ et d'arrêt.....	30
Figure.III.4: Exemple de transmission réussie.....	31
Figure.III.5: Exemple d'octet d'adresse.....	31
Figure.III.6: Ecriture d'une donnée (ds1307).....	32
Figure.III.7: Le chronogramme correspondant des lignes SDA et SCL.....	33
Figure.III.8 : Lecture d'une donnée (ds1307).....	34
Figure .III.9: Interface du logiciel MikroC.....	37
Figure .III.10: Création d'un projet en MikroC.....	37
Figure .III.11: Les configurations de projet.....	38
Figure .III.12: Fenêtre de saisie de programme.....	39
Figure.IV.1 : Schéma synoptique du projet.....	42
Figure.IV.2: Interface du logiciel ISIS.....	43
Figure.IV.3: Organisation de la boîte à outils ISIS.....	44

Figure.IV.4: Le logiciel ISIS 7 en mode simulation du projet.....	45
Figure.IV.5: Configuration le port c (SDA et SCL).....	46
Figure.IV.6: Brochage du PCF8574.....	47
Figure.IV.7: La structure de matrice de points LED 8x8.....	48
Figure.IV.8: Le niveau logique de la matrice 8x8.....	48

LISTE DES ABRÉVIATIONS

- A/N** : Convertisseur analogique – numérique
- N/A** : Convertisseurs Numériques – Analogiques
- FPGA** : Circuit logique programmable
- E/S** : Entré Sortie
- CPU** : Unité centrale de traitement
- ACC** : Adaptive Cruise Control
- DSP** : Digital Signal Processor
- RISC** : Reduced Instruction Set Computer
- CISC** : Complex Instruction Set Computer
- PIC** : Programmable Interface Contrôler
- ALU** : Unité arithmétique et logique
- USART** : Port série universel mode asynchrone
- SSP** : Port série synchrone
- ICSP** : In Circuit Serial Programming
- MCLR** : Reset
- I2C** : (IIC) Inter Integrated Circuit
- SDA** : Signal de donnée
- SCL** : Signal d'horloge
- ACK** : Acknowledge (reconnaître)
- ISIS** : Logicielle de simulation
- LED** : Light emitting diode (Diode électroluminescente)

RÉSUMÉ

Les protocoles de communication sont dévers et varient entre les appareils électroniques et se diffère selon leur utilisation.

Dans notre projet, nous avons utilisé le protocole I2C dans un environnement à microcontrôleur PIC 16F877 pour contrôler les circuits séries-parallèle est changer de commander une matrice graphique à diode LED extensible.

ملخص

تتعدد بروتوكولات الاتصال بين الاجهزة الالكترونية وتختلف حسب استعمالاتها .
في مشروعنا هذا استخدمنا البروتوكول I2C في بيئة المتحكم (PIC 16F877) من اجل مراقبة الدارة
المتكاملة (PCF8574) للتحكم في مصفوفة من الصمامات الثنائية التي ينبعث منها الضوء (LED) قابلة
للتمدد .

SOMMAIRE

Chapitre I : Généralité sur le système embarqué

I.1 Introduction.....	2
I.2 Système embarqué typique.....	3
I.3 Circuit logique programmable« FPGA ».....	5
I.4 Caractéristiques principales d'un système embarqué.....	5
I.5 Le langage C pour l'embarqué.....	8
I.6 Exemple un système embarqué (radar ACC 77Ghz).....	9
I.6.1 Organigramme de radar ACC.....	10
I.6.2 Fonctionnalité de radar ACC.....	11
I.7 Conclusion.....	11

Chapitre II : Les microcontrôleurs dans un systèmes embarqués

II.1 Introduction.....	12
II.2 Organisation du plan mémoire.....	12
II.2.1 Architecture de Von Neumann et architecture de Harvard.....	13
II.2.2 Architecture de Harvard modifiée DSP (Digital Signal Processor).....	14
II.3 Les type de processeur.....	15
II.3.1 Architecture RISC (Reduced Instruction Set Computer).....	15
II.3.2 Caractéristiques des machines RISC.....	15
II.3.3 Architecture CISC (Complex Instruction Set Computer).....	15
II.3.4 Comparaison des technologies RISC et CISC.....	16
II.4 Processeur pipeline.....	17
II.5 Le microcontrôleur pic.....	18

II.5.1	Definition d'un PIC.....	18
II.5.2	Les différentes familles de PIC.....	18
II.5.3	Identification des PIC.....	19
II.6	Architecture interne du PIC16F877.....	20
II.6.1	Caractéristiques principales du pic 16F877.....	21
II.7	Brochage du PIC16F877.....	23
II.7.1	L'horloge.....	23
II.7.2	Le Reset.....	24
II.7.3	Les ports d'E/S.....	25
II.8	le bus I2C et le PIC 16F877.....	26
II.8 .1	Organisation de la mémoire.....	27
II.9	Conclusion.....	27

Chapitre III : Le protocole bus I2C

III.1	Introduction.....	28
III.2	Présentation et caractéristiques du bus I2C.....	28
III.3	Le protocole I2C.....	30
III.3.1	La prise contrôle du bus.....	30
III.3.2	La transmission d'un octet.....	30
III.3.3	La transmission d'une adresse.....	31
III.3.4	Écriture d'une donnée.....	32
III.3.5	Lecture d'une donnée.....	33
III.4	Présentation du MikroC.....	36
III.4.1	Création d'un projet.....	37
III.4.2	Les bibliothèques du MikroC.....	39
III.5	Bus I2C et MikroC.....	40
III.5.1	Fonction d'écriture.....	40
III.5.2	Fonction de lecture.....	41

III.6 Conclusion.....41

Chapitre IV : Conception de la carte et simulation

IV.1 Introduction.....42

IV.2 Schéma synoptique du projet.....42

IV.3 Présentation du logiciel ISIS.....43

IV.3.1 Fenêtre d'ensemble (Vue d'ensemble).....43

IV.3.2 Fenêtre d'édition.....44

IV.3.3 La boîte à outils.....44

IV.3.4 Organisation de la boîte à outils.....44

IV.4 Simulation par ISIS.....45

IV.4.1 descriptif des composants utilisés.....45

IV.4.1.1 Alimentation.....45

IV.4.1.2 Unité de commande.....45

IV.4.1.3 Trois ports séries - parallèles PCF8574.....46

IV.4.1.4 Deux matrices LED.....48

IV.5 Fonctionnement de système.....49

IV.6 organigramme de système.....50

IV.7 Conclusion.....52

Introduction générale

Les nouvelles applications industrielles de contrôle, de commande et de communication sont devenues des systèmes embarqués qui reposent sur l'utilisation de circuits intégrés complexes incluant matériel et logiciels (systèmes d'exploitation et logiciel d'application) implanté dans une même puce électronique, FPGA, ASIC etc.

Que ce soit en génie électriques, dans les entraînements industriels, dans les systèmes de télécommunication ou même dans le domaine de l'instrumentation médicale les systèmes embarqués sont fortement présents.

Notre projet a pour objectif la conception et la mise en œuvre d'un système électronique embarqué capable de piloté via un bus I2C l'affichage d'information sur matrice a diode LED. Le système a concevoir est flexible, extensible et programmable en langage C . L'ensemble du système est piloté par le microcontrôleur pic.Ce dernier est conçu par Microchip pour les applications similaires.

Dans le premier chapitre nous avons étudié le système embarqué en générale. Ces élément sont rassemblés dans un circuit intelligent le microcontrôleur qui est programmé en un langage c. En fin nous avons présenté un exemple de système embarqué.

Dans le deuxième chapitre nous avons vu une étude détaillée sur les microcontrôleurs et leur architecteurs et l'organisation de leur mémoire après comparaison des technologies CISC et RISC on s'est intéressé au microcontrôleur PIC 16F877 conçu en technologie RISC.

Dans le troisième chapitre nous avons étudié le protocole de bus I2C qui un protocole de communication rapide et facile à mettre en œuvre, Il permet de créer des bus contenant plusieurs maîtres et esclaves. Le logiciel de programmation mikroC et utilisé pour l'implantation de ce protocole.

Dans le dernier chapitre nous présentons les modèles et résultats de simulation en environnement à microcontrôleur PIC 16F877 pour la mise en œuvre de notre application.

Chapitre I : Généralité sur le système embarqué

I.1 Introduction.....	2
I.2 Système embarqué typique.....	3
I.3 Circuit logique programmable« FPGA ».....	5
I.4 Caractéristiques principales d'un système embarqué.....	5
I.5 Le langage C pour l'embarqué.....	8
I.6 Exemple un système embarqué (radar ACC 77Ghz).....	9
I.6.1 Organigramme de radar ACC.....	10
I.6.2 Fonctionnalité de radar ACC.....	11
I.7 Conclusion.....	11

I.1 Introduction

Un système embarqué est un système électronique et informatique autonome ne possédant pas des entrées/sorties standards comme un clavier ou un écran d'ordinateur. Il est piloté dans la majorité des cas par un logiciel, qui est complètement intégré au système qu'il contrôle. On peut aussi définir un système embarqué comme un système électronique soumis à diverses contraintes.

Les concepteurs des systèmes électroniques sont aujourd'hui confrontés à la complexité croissante des algorithmes mis en œuvre et à la variété des cibles potentielles FPGAs et/ou DSPs. Actuellement, il n'est pas rare que ces systèmes intègrent plusieurs douzaines voire des centaines de processeurs. A l'origine, ce sont des systèmes matériels et logiciels intégrés dans des avions militaires ou des missiles. Ensuite dans le civil: avions, voitures, machine à laver..., La machine et le logiciel sont intimement liés et noyés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type PC. On mentionne les contraintes physiques fortes: dimensions, poids, taille, autonomie, consommation, fiabilité, contraintes temporelles (temps réels).

Un système embarqué peut être défini comme un système électronique et informatique autonome, qui est dédié à une tâche bien précise.

I.2 Système embarqué typique

Dans la figure (I.1) au-dessous architecture matérielle d'un système embarqué :

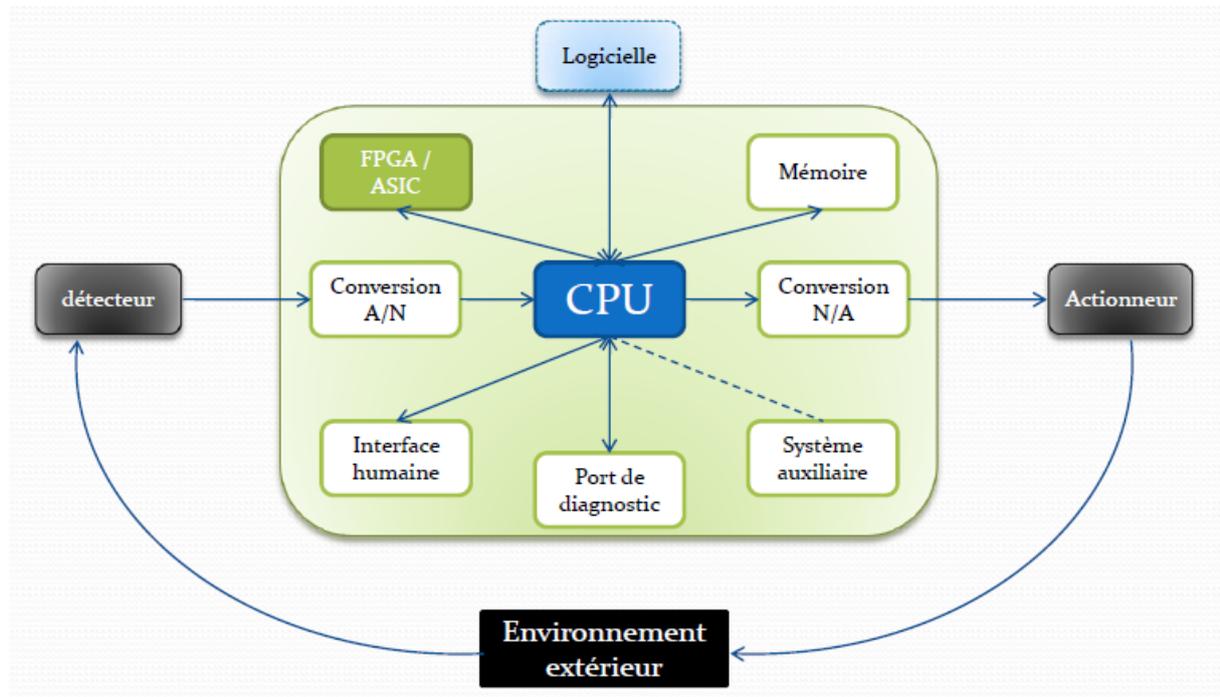


Figure.I.1 : Architecture matérielle d'un système embarqué

On retrouve en entrée des capteurs généralement analogiques couplés à des convertisseurs A/N. On retrouve en sortie des actionneurs généralement analogiques couplés à des convertisseurs N/A. Au milieu, on trouve le calculateur mettant en œuvre un processeur embarqué et ses périphériques d'E/S. Il est à noter qu'il est complété généralement d'un circuit FPGA jouant le rôle de coprocesseur afin de proposer des accélérations matérielles au processeur.

On retrouve en fait un système d'asservissement entre les entrées et les sorties, Il est à noter que l'expression la plus simple de cette figure (I.1) est de considérer comme capteurs des interrupteurs et comme actionneurs des LED.

Sur ce schéma théorique se greffe un paramètre important : le rôle de l'environnement extérieur. Un système embarqué doit faire face à des environnements plus hostiles. Il doit faire face à un ensemble de paramètres agressifs :

- Variations de la température.
- Vibrations, chocs.
- Variations des alimentations.
- Corrosion.
- Eau, feu, radiations.

L'environnement dans lequel opère le système embarqué n'est pas contrôlé ou contrôlable. Cela suppose donc de prendre en compte ce paramètre lors de sa conception. On doit par exemple prendre en compte les évolutions des caractéristiques électriques des composants en fonction de la température, des radiations..., les systèmes embarqués sont aujourd'hui fortement communicants. Cela est possible grâce aux puissances de calcul offertes par les processeurs pour l'embarqué (32 bits en particulier) et grâce aussi à l'explosion de l'usage de la connectivité Internet. La connectivité internet permet fondamentalement de contrôler à distance un système embarqué par Internet. Ce n'est en fait que l'aboutissement du contrôle à distance d'un système électronique par des liaisons de tout type : liaisons RS.232, RS.485, bus de terrain...

Cela permet l'emploi des technologies modernes d'internet pour ce contrôle à distance par l'utilisateur : il suffit d'embarquer un serveur internet dans son équipement électronique pour pouvoir le contrôler ensuite à distance, de n'importe où, à l'aide d'un simple navigateur. Il n'y a plus d'interface humaine spécifique à concevoir pour cela, ce rôle étant rempli par le navigateur Internet. Cela est une réalité : les chauffagistes proposent maintenant des chaudières pouvant être pilotées par l'internet, Il faut aussi noter la montée en puissance des communications sans fil dans l'embarqué au détriment des communications filaires pour limiter le câblage et faciliter la mise en place du système embarqué. Le wifi et toutes les normes de réseaux sans fil IEEE 802.15 comme ZigBee dans l'embarqué et surtout en domotique (réseaux de capteurs sans fil par exemple). Mais ne nous méprenons pas sur ces facilités et commodités, cela a bien sûr un revers : la sécurité du système embarqué, puisque connecté à Internet. [1]

Un système embarqué composé de :

- Unité centrale de traitement « CPU ».
- Les Convertisseurs analogiques – numériques (CAN).
- Les Convertisseurs Numériques – Analogiques (CNA).
- Les mémoires.
- les Circuits logiques programmables « FPGA ».

I.3 Circuit logique programmable « FPGA »

Un circuit logique programmable, ou réseau logique programmable, est un circuit intégré logique qui peut être reprogrammé après sa fabrication. Il est composé de nombreuses cellules logiques élémentaires librement assemblables.

La plupart des grands FPGA modernes sont basés sur des cellules SRAM aussi bien pour le routage du circuit que pour les blocs logiques à interconnecter. FPGA jouant le rôle de coprocesseur afin de proposer des accélérations matérielles au processeur. [1]

Les circuits spécifiques sont spécialement conçu au spécialement adapté au système embarqué les microcontrôleurs.

I.4 Caractéristiques principales d'un système embarqué

Les principales caractéristiques d'un système embarqué sont les suivantes :

- C'est un système principalement numérique.
- Il met en œuvre généralement un processeur.
- Il exécute une application logicielle dédiée pour réaliser une fonctionnalité précise et n'exécute donc pas une application scientifique ou grand public traditionnelle.
- Il n'a pas réellement de clavier standard (Bouton Poussoir, clavier matriciel...). L'affichage est limité (écran LCD...) ou n'existe pas du tout.
- Ce n'est pas un PC en général mais des architectures similaires (x86) basse consommation sont de plus en plus utilisées pour certaines applications embarquées.

De ce constat, on peut voir :

- Qu'un PC standard peut exécuter tout type d'applications car il est généraliste alors qu'un système embarqué n'exécute qu'une application dédiée.
- Que des circuits numériques ou des circuits analogiques sont utilisés en plus pour augmenter les performances du système embarqué ou sa fiabilité.

L'omniprésence des systèmes embarqués dans notre vie est liée à la révolution numérique opérée dans les années 1970 avec l'avènement des processeurs. Les processeurs, de plus en plus rapides, puissants et bon marché ont permis cette révolution et aussi le boom du marché de l'embarqué. Ceci se confirme au travers de la loi empirique de Gordon Moore, cofondateur d'Intel, qui stipule que pour une surface de silicium donnée, on double le nombre de transistors intégrés tous les 18 mois, La Figure.I.2 montre cette évolution inexorable. [1]

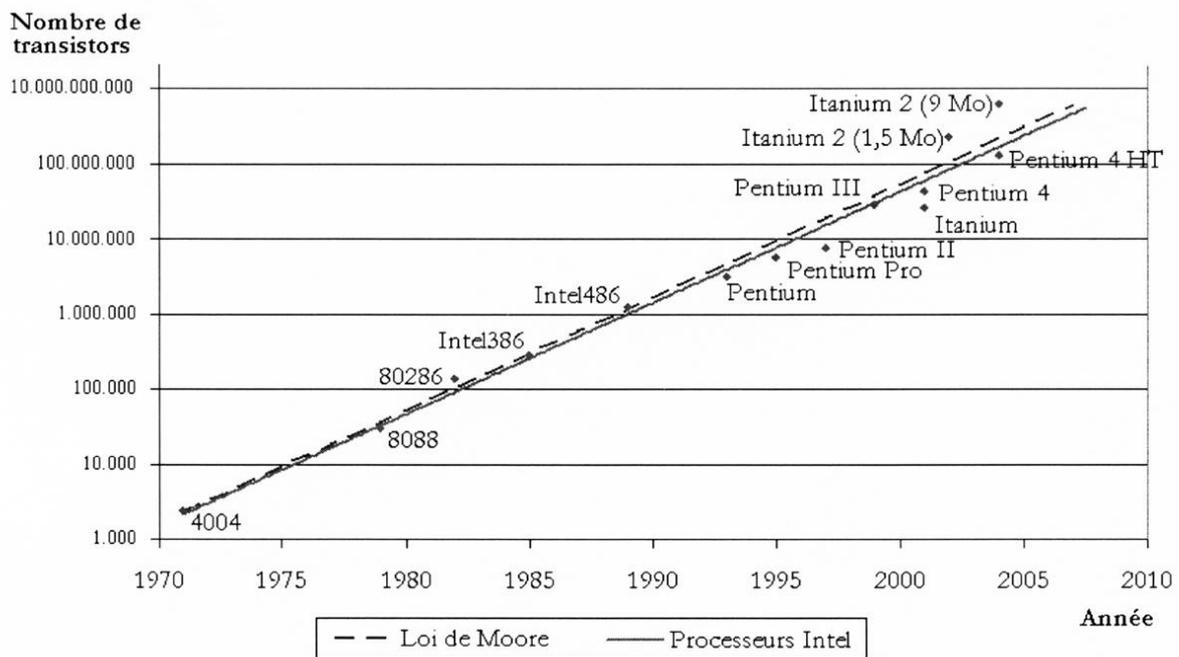


Figure.I.2 : La loi G. Moore pour les processeurs Intel

En 1999, il a été vendu pour le marché de l'embarqué :

- 1,3 milliard de processeurs 4 bits.
- 1,4 milliard de processeurs 8 bits.
- 375 millions de processeurs 16 bits.
- 127 millions de processeurs 32 bits.
- 3,2 millions de processeurs 64 bits.

A côté de cela, à cette époque, il a été vendu seulement 108 millions de processeurs (famille x86) pour le marché du PC grand public. Pour 2004, il a été vendu environ 260 millions des processeurs pour le marché du PC grand public à comparer aux 14 milliards des processeurs tout type confondu (microprocesseur, microcontrôleur, DSP) pour le marché de l'embarqué.

Le marché des processeurs pour l'embarqué selon Electronics.Ca Réseau de recherche devrait croître de 6 % en 2005 pour un chiffre d'affaire mondial de 18 milliards USD, Les chiffres parlent d'eux-mêmes. On peut aussi tirer le constat actuel suivant :

- Moins de 2 % des processeurs vendus sont pour le marché du PC contre 98 % pour l'embarqué. On utilise massivement pour le PC grand public le système d'exploitation commercial bien connu.

- Pour les 98 % autres processeurs vendus, on utilisera généralement un autre système d'exploitation. On trouvera ici dans 60 % des cas un système d'exploitation commercial spécialisé pour l'embarqué [Vx Works (système d'exploitation temps réel multitâche), QNX (système d'exploitation UNIX)]. Pour le reste, il s'agit d'un système d'exploitation home made mais de plus en plus optent pour des systèmes d'exploitation libres comme Linux pour l'embarqué.

- Moins de 10 % des processeurs vendus sont des processeurs 32 bits pour près de 31 % du chiffre d'affaire sur les processeurs. Cette part du chiffre d'affaire est estimée à près de 48 % pour 2008 : cela montre la migration rapide vers ces processeurs 32 bits dans l'embarqué, condition nécessaire pour pouvoir mettre en œuvre Linux.

- Si l'on regarde le prix moyen d'un processeur tout type confondu, on arrive à 6 USD par unité à comparer au prix moyen de 300 USD par unité pour un processeur pour PC. Le marché du processeur pour PC est très faible en volume mais extrêmement lucratif, Ce quelques chiffres permettent bien de prendre conscience de l'importance du marché de l'embarqué et de mettre fin.

Les grands secteurs de l'embarqué concernent les domaines suivants :

- Jeux et calcul général : application similaire à une application de bureau mais empaquetée dans un système embarqué : jeux vidéo, set top box...

- Contrôle de systèmes : automobile, processus chimique, processus nucléaire, système de navigation...

- Traitement du signal : radar, sonar, compression vidéo...
- Communication et réseaux : transmission d'information et commutation, Internet... [1]

I.5 Le langage C pour l'embarqué

Le langage de programmation C est la référence pour programmer les microcontrôleurs avec des exigences temps réel. En effet, ce dernier présente l'avantage d'être un langage de programmation orienté matériel. Le code machine généré par le compilateur est presque optimal en ce qui concerne d'une part la taille de la mémoire requise et d'autre part la vitesse d'exécution du programme. Par conséquent, environ 65% des applications embarquées sont programmées en C.

Le langage C est un langage :

1- Universel : C n'est pas orienté vers un domaine d'applications spéciales comme SQL par exemple.

2- Compact : C'est basé sur un noyau de fonctions et d'opérateurs limités qui permet la formulation d'expressions simples mais efficaces.

3- Moderne : C'est un langage structuré, déclaratif et récursif. Il offre des structures de contrôle et de déclaration comme le langage Pascal.

4- Près de la machine : C offre des opérateurs qui sont très proches de ceux du langage machine (Manipulations de bits, pointeurs...). C'est un atout essentiel pour la programmation des systèmes embarqués.

5- Rapide : C permet de développer des programmes consistants et rapides.

6- Indépendant de la machine : C'est un langage près de la machine (microprocesseur) mais il peut être utilisé sur n'importe quel système ayant un compilateur C.

Au début C'était surtout le langage des systèmes UNIX mais on le retrouve comme langage de développement aujourd'hui de tous les systèmes d'exploitation.

7- portable: En respectant le standard ANSI-C, il est possible d'utiliser (théoriquement) le même programme sur tout autre système (autre hardware, autre système d'exploitation), simplement en Le recompilant. Il convient néanmoins de faire attention dans le cas d'un système embarqué qui n'inclut pas toujours un système d'exploitation. [2]

I.6 Exemple un système embarqué (radar ACC 77Ghz)

ACC : Initiales pour Adaptive Cruise Control, ce terme désigne le régulateur de vitesse gérant l'inter-distance avec le véhicule qui précède. L'usage d'un radar à ondes millimétriques permet d'évaluer la position et la vitesse relatives du véhicule précédent. Une régulation intervient par action sur l'accélérateur et les freins pour maintenir entre les deux véhicules une distance qui respecte les contraintes de sécurité définies dans la stratégie implémentée dans le dispositif.

Il y a plusieurs applications pour les systèmes radars automobile comme indiqué sur la(Figure.I.3). Pour le système 77 GHz qui est adapté jusqu'à 200 m, le système radar est utilisé comme: radar de prospection, régulateur de vitesse et d'espacement (ACC), [3]

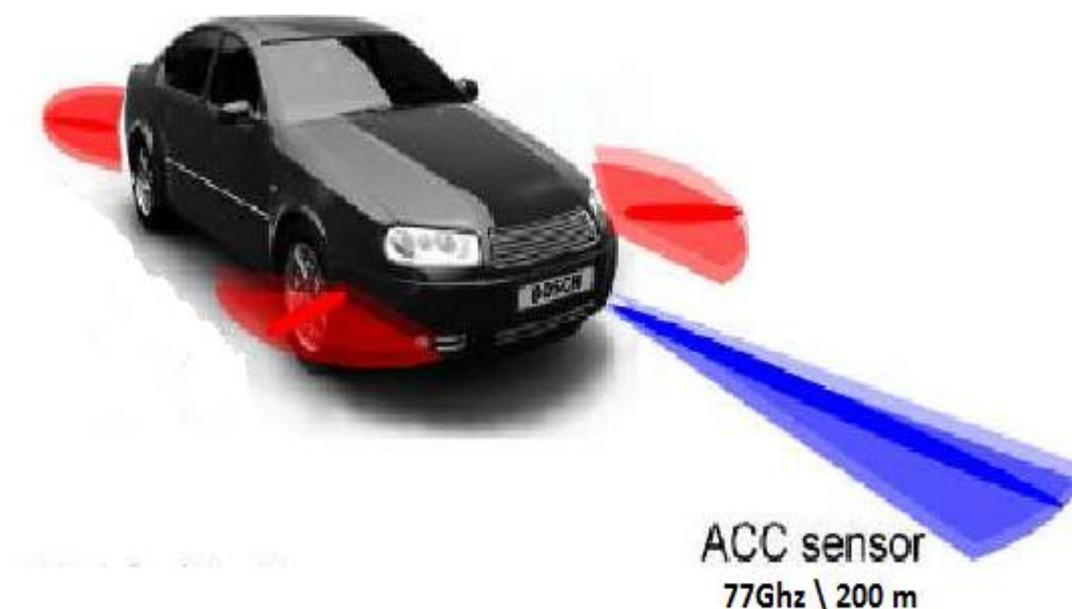


Figure.I.3 : radar ACC 77Ghz

I.6.1 Organigramme de radar ACC

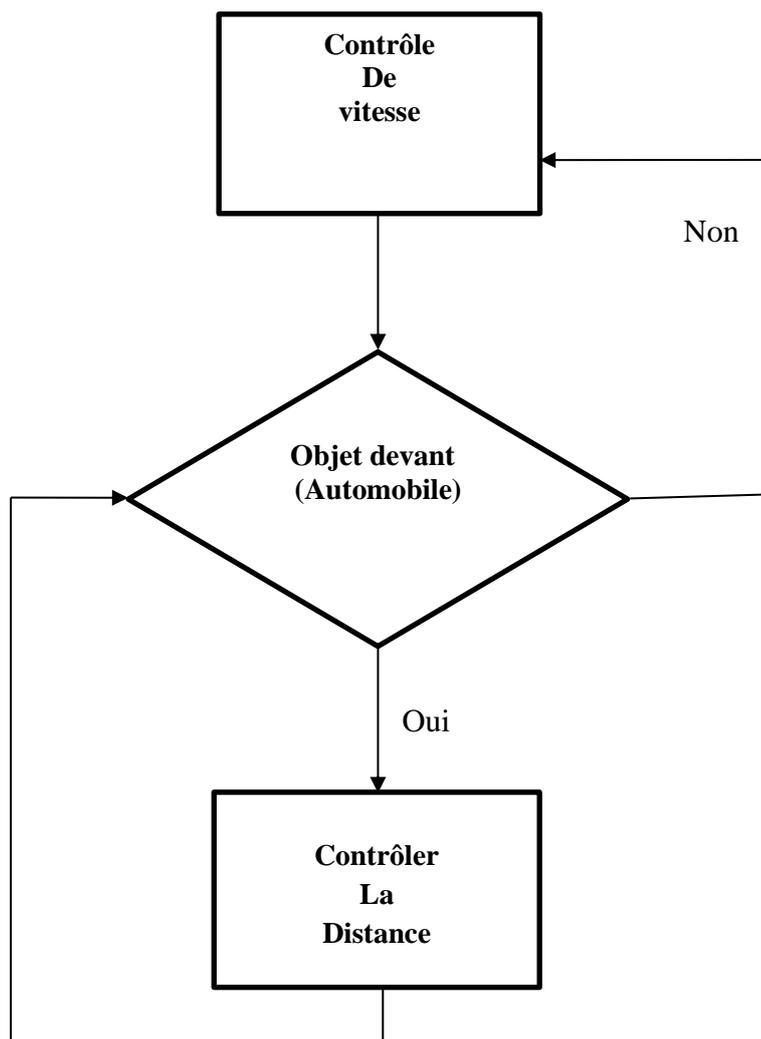


Figure.I.3.1 : Organigramme de radar ACC

I.6.2 Fonctionnalité de radar ACC

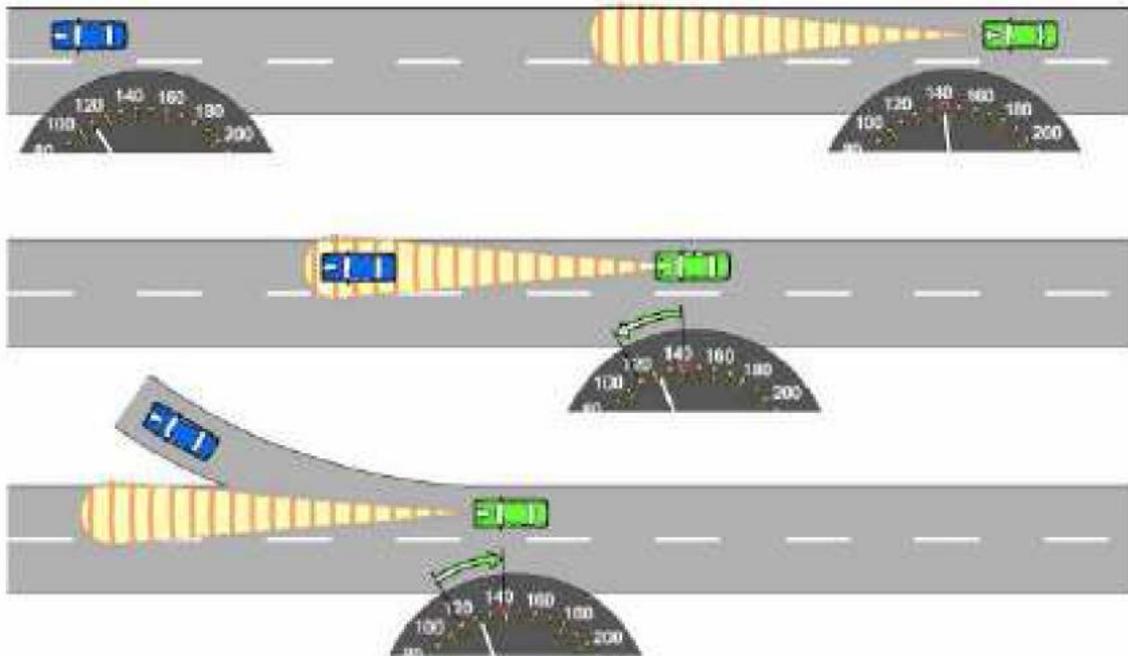


Figure.I.4 : Fonctionnalité de radar ACC [3]

I.7 Conclusion

Dans ce chapitre, nous avons présenté d'une manière les systèmes embarqués, leur principes de fonctionnement leur architecture ainsi que l'outil principal de développement logiciel, le langage C dédié à ce système.

Chapitre II : Les microcontrôleurs dans un systèmes embarqués

II.1 Introduction.....	12
II.2 Organisation du plan mémoire.....	12
II.2.1 Architecture de Von Neumann et architecture de Harvard.....	13
II.2.2 Architecture de Harvard modifiée DSP (Digital Signal Processor).....	14
II.3 Les types de processeur.....	15
II.3.1 Architecture RISC (Reduced Instruction Set Computer).....	15
II.3.2 Caractéristiques des machines RISC.....	15
II.3.3 Architecture CISC (Complex Instruction Set Computer).....	15
II.3.4 Comparaison des technologies RISC et CISC.....	16
II.4 Processeur pipeline.....	17
II.5 Le microcontrôleur pic.....	18
II.5.1 Definition d'un PIC.....	18
II.5.2 Les différentes familles de PIC.....	18
II.5.3 Identification des PIC.....	19
II.6 Architecture interne du PIC16F877.....	20
II.6.1 Caractéristiques principales du pic 16F877.....	21
II.7 Brochage du PIC16F877.....	23
II.7.1 L'horloge.....	23
II.7.2 Le Reset.....	24
II.7.3 Les ports d'E/S.....	25
II.8 le bus I2C et le PIC 16F877.....	26
II.8 .1 Organisation de la mémoire.....	27
II.9 Conclusion.....	27

II.1 Introduction

Le processeur constitue le noyau principale d'un système embarqué. Il doit avoir un certain nombre de caractéristique lui permettent d'accomplir sa tâche convenablement : rapidité, consommation, puissance....

La notion de puissance est la capacité de traiter un grand nombre d'opérations par seconde sur de grands nombres et en grande quantité. Intrinsèquement la puissance se joue donc sur les trois critères suivants:

- La longueur des mots : données et instructions (on parle de largeur du bus des données).
- Le nombre d'octets que le microprocesseur peut adressé (on parle de largeur du bus des adresses).
- La vitesse d'exécution des instructions liée à la fréquence de fonctionnement de l'horloge de synchronisation exprimée en MHZ.

Il ya diffèrent architecture et technologie d'un système de processeur dédié.

Les classifications des microcontrôleurs à deux niveaux :

- Au niveau du processeur:
 - RISC : Reduced Instruction Set Computer.
 - CISC : Complex Instruction Set Computer.
- Au niveau de l'organisation de la mémoire.
 - Architecture Von Neumann : une mémoire unique et pour le programme et pour les données.
 - Architecture Harvard : le programme et les données sont stockés dans des mémoires physiquement séparées.

II.2 Organisation du plan mémoire

Il existe deux types de plan mémoire :

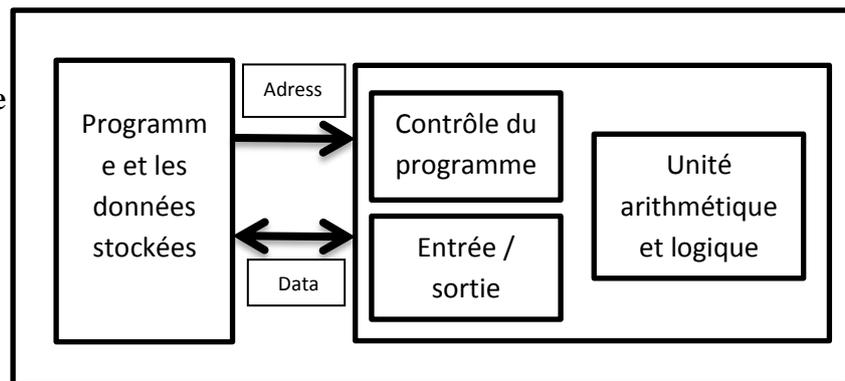
- architecture de type Von Neumann.
- architecture de type Harvard.

II.2.1 Architecture de Von Neumann et architecture de Harvard

À chaque cycle d'horloge, le processeur sait par le compteur de programme l'instruction qu'il doit faire exécuter. Nous avons vu qu'il va chercher chaque instruction en mémoire, l'exécute avec les données correspondantes et retourne les données résultantes en mémoire.

Dans l'architecture de la machine de Von Neumann, le programme et les données sont enregistrés sur la même mémoire. Chaque instruction contient la commande de l'opération à effectuer et l'adresse de la donnée à utiliser, il faut donc souvent plusieurs cycles d'horloge pour exécuter une instruction. La Figure.II.1 indique une architecture simple de Von Neumann, constituée d'un bus de données et de programme et d'un bus d'adresses.

Figure.II.1 : Architecture de Von Neumann.



On voit que les échanges s'effectuent de manière simple entre l'unité arithmétique et logique (ALU), c'est-à-dire l'unité centrale et la mémoire unique, par un bus transitant les codes de programme et les données. On a ainsi des données « collées » aux instructions. Les microprocesseurs et beaucoup de microcontrôleurs utilisent cette architecture car elle est très souple pour la programmation. [4]

Dans l'architecture dite de Harvard (car mise au point dans cette université américaine en 1930), on sépare systématiquement la mémoire de programme de la mémoire des données : l'adressage de ces mémoires est indépendant. La Figure.II.2 indique une architecture simple de Harvard constituée d'un bus de données, d'un bus de programme et de deux bus d'adresse.

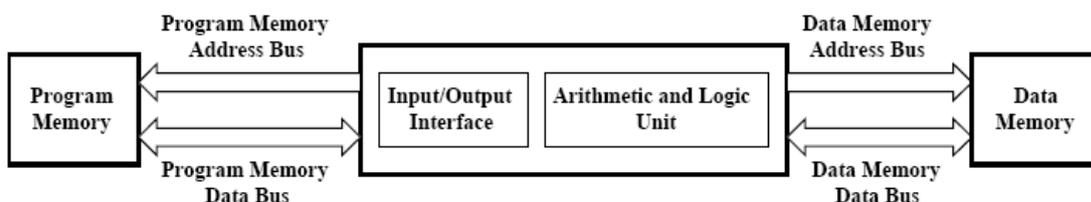


Figure.II.2 : Architecture de Harvard

On voit que les échanges s'effectuent de manière double entre l'unité centrale et les deux mémoires, ce qui permet une grande souplesse pour l'enregistrement et l'utilisation des données. D'ailleurs, la mémoire de programme est également utilisée en partie comme mémoire de données pour obtenir encore plus de possibilités de traitement avec des algorithmes complexes. [4]

L'architecture Harvard est souvent utilisée dans :

- les processeurs numériques de signal (DSP) ;
- les microcontrôleurs, notamment les PIC de Micro chip et les AVR d'Atmel.

II.2.2 Architecture de Harvard modifiée DSP (Digital Signal Processor)

La structure de HARVARD est cependant pénalisante par le nombre de broches utilisées et le nombre de boîtiers mémoires utilisés comme de nombreux autres constructeurs (INTEL avec le MCS 51, MICROCHIP avec les microcontrôleurs PIC) utilise une structure de HARVARD modifiée, qui consiste à n'utiliser qu'un bus commun pour les accès en mémoire externe (architecture de VON NEUMAN) et à utiliser une structure de HARVARD pour les accès en mémoire interne. Charge ensuite au programmeur, si la taille de la mémoire programme ou de la mémoire données est trop importante, de placer en mémoire interne le bout de programme et de données qui demande une rapidité optimale d'exécution. [5]

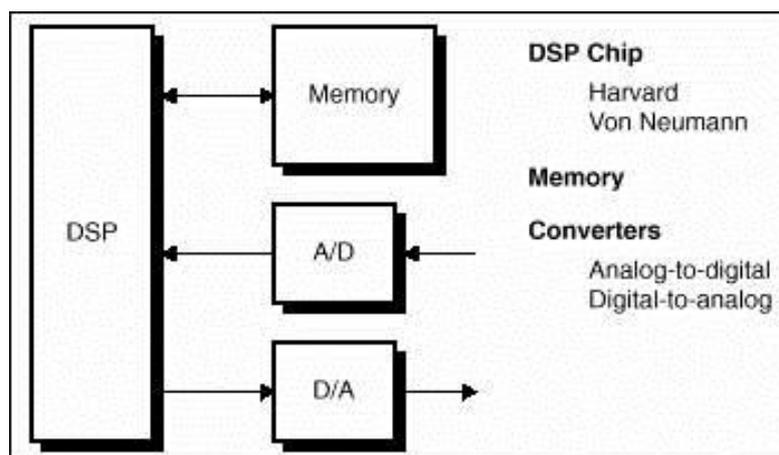


Figure.II.3 : Structure de Harvard modifiée

II.3 Les type de processeur

Il existe sur le marché deux types de processeurs:

II.3.1 Architecture RISC (Reduced Instruction Set Computer)

Architecture dans laquelle les instructions sont en nombre réduit (chargement, branchement, appel sous-programme) et elles sont fréquemment utilisées. Le but est d'éliminer les instructions rarement employées et de consacrer les ressources matérielles à exécuter les instructions relativement simples en un cycle d'horloge et à émuler les autres instructions à l'aide des séquences basées sur les instructions élémentaires. On trouve donc une meilleure performance à une vitesse donnée (le gain en performance envisageable est important mais dépend de la qualité du compilateur). [6]

II.3.2 Caractéristiques des machines RISC

- 1- un cycle d'horloge au maximum pour l'exécution d'une instruction.
 - 2- une dépendance plus forte par rapport aux compilateurs.
 - 3- toutes les instructions ont la même longueur.
 - 4- une utilisation systématique des registres (généralement plus nombreux que sur les processeurs CISC).
 - 5- séquenceur câblé plutôt que programmé.
 - 6- une limitation des accès mémoire aux deux opérations LOAD et STORE. Stations de travail équipées de processeurs à architecture RISC : SUN, HP, IBM.
- Processeurs RISC : Alpha (DEC), Power RISC (IBM), SPARC (SUN), PA-RISC (HP). [6]

II.3.3 Architecture CISC (Complex Instruction Set Computer)

C'est une architecture avec un grand nombre d'instructions. Le processeur doit exécuter des tâches complexes par instruction unique. Donc, pour une tâche donnée, une machine CISC exécute un petit nombre d'instructions mais chacun nécessite un plus grand nombre de cycles d'horloge (Pentium et PowerPC). Actuellement les deux technologies convergent : les processeurs CISC (Pentium par exemple) utilisent des instructions de plus en plus simples et exécutent parfois plusieurs instructions en un cycle d'horloge. [6]

II.3.4 Comparaison des technologies RISC et CISC

	Cisc	Risc
Signification	Complexe Instructions Set Computer.	Reduced Instructions Set Computer.
Traduction	Processeur à jeu d'instructions complexe.	Processeur à jeu d'instructions réduit.
Apparition	Technologie plus ancienne.	Technologie plus récente.
Principes fondamentaux	Complexité des instructions disponibles. Le jeu d'instructions s'est enrichi au fil du temps en gardant une compatibilité ascendante. Le décodage assez lent de chaque instruction.	- Report de la complexité matérielle sur le logiciel (compilateur). - Effectuer des accès à la mémoire. Uniquement au travers de deux instructions : Load et Store. Être adapté aux langages de haut niveau.
Jeu d'instruions	Environ 350 instructions.	- Environ 150 instructions. Utilisation d'un jeu d'instructions de base minimal, simulation des autres.
Format des instructions	Variable selon J'instruction.	Unique ou presque.
Durée d'exécution d'une instruction	instructions simples ne prenant qu'un seul cycle.	Instructions complexes prenant plusieurs cycles.
Hierarchisation de la mémoire	Moins de 10 registres internes spécialisés les nombreux accès à la mémoire.	- 64 registres internes et polyvalents.
Fonctionnement	- séquentiel - Logique de traitement et décodage qui occupe 50% de la surface du circuit Séquenceur microprogramme.	- pipeline - Logique de traitement et décodage qui occupe 5 à 10% de la surface du circuit. Séquenceur câblé. il reste de la place pour de nombreuses fonctions supplémentaires qui augmentent les performances.
Comparatif	Technologie moins puissante. Plus complexe, moins performante et plus chère.	- Inverse. - L'architecture plus adaptée à l'évolution l'informatique.

Tableau 1 : Comparaison des technologies RISC et CISC [7]

II.4 Processeur pipeline

L'idée est inspirée de l'organisation du travail à la chaîne : cette technique permet d'effectuer davantage de travail par unité de temps. Chaque étape de l'exécution de l'instruction (chargement, décodage, recherche des données) se fait dans une unité séparée :

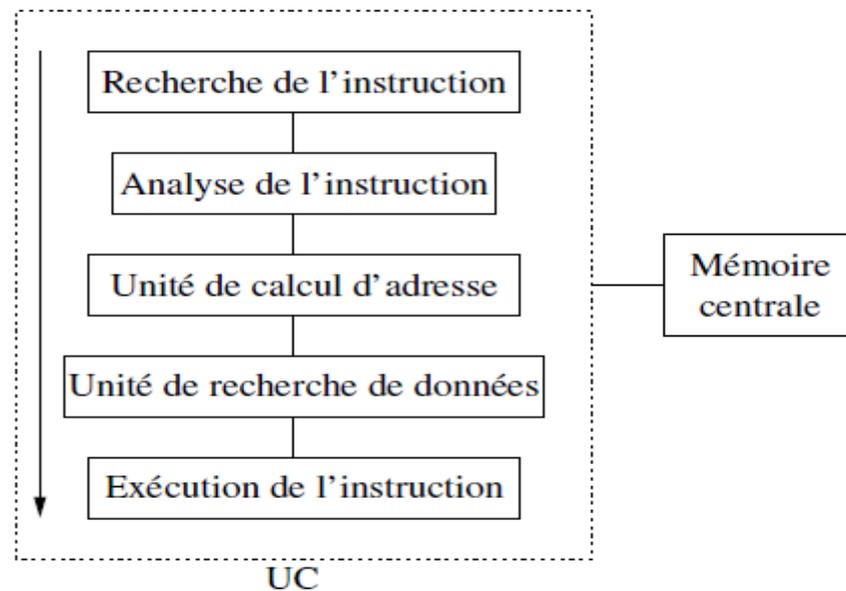


Figure.II.4: Organisation processeur pipeline

- la première est chargée dans la première unité.
- la deuxième unité commence à décoder l'instruction pendant que la première travaille à charger la deuxième instruction.
- un peu plus tard, la troisième unité travaille sur le calcul d'adresse pendant que la deuxième unité décode la deuxième instruction, et que la première unité charge l'instruction suivante (exemple : IBM 360/195, CDC STAR). [8]

II.4.1 Définition d'un microcontrôleur

Un microcontrôleur, est un composant électronique qui rassemble tous les éléments d'un "mini-ordinateur" et qui se présente sous la forme d'un circuit intégré. Un microcontrôleur permet de réaliser des systèmes et montages électroniques programmés. Cela veut dire que l'on pourra, avec le même montage, réaliser des fonctions très différentes qui dépendront du programme qui aura été programmé dans le microprocesseur. [9]

II.5 Le microcontrôleur pic

II.5.1 Définition d'un PIC

Le PIC (Programmable Interface Contrôler) est une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de faciliter l'interfaçage avec le monde extérieur sans nécessiter l'ajout de composants externes. Les PICs sont des composant dits RISC (Reduced Instructions Set Computer), ou encore composant à jeu d'instruction réduit. Le microcontrôleur ce trouve, dans plusieurs appareils telle que: les téléphones portables, machines à laver, télévisions vidéos Etc. [10]

II.5.2 Les différentes familles de PIC

Les PICs sont subdivisés en 3 grandes familles :

- 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 les 16F876 et 16F877)
- la famille High-End, qui utilise des mots de 16 bits. [10]



Figure.II.5: Les différentes familles de PIC

II.5.3 Identification des PIC

Un PIC est généralement identifié par une référence de la forme suivante : xx(L)XXyy-zz

Xx : famille du composant, actuellement « 12, 14, 16, 17 et 18 ».

L : tolérance plus importante de la plage de tension.

XX : type de mémoire programme

C : EPROM ou EEPROM

CR : PROM

F : Flash

Yy : Identificateur.

Zz : vitesse maximale du quartz de pilotage. [11]

Exemple :

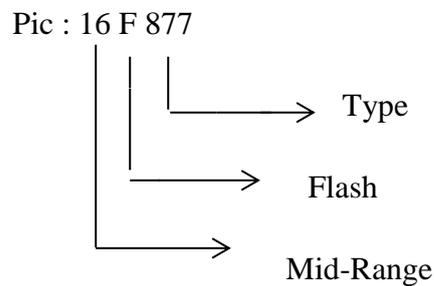


Figure.II.6 : Exemple de l'identification PIC16F877

II.6 Architecture interne du PIC16F877

Ce PIC 16F877 est un microcontrôleur de MICROCHIP, le schéma architecture interne sont présentés dans le la figure (2.4). Il fait partie intégrante de la famille des Mid-Range (16) dont la mémoire programme est de type flash (F). Il est constitué des éléments suivants:

- Quatre ports d'entrées/sorties.
- Une unité arithmétique et logique (ALU).
- trois compteurs (Timers) Timer0,Timer1,Timer2.
- Un compteur de programme (program conter).
- Une mémoire RAM, 368 octets.
- Une mémoire EEPROM de 256 octets de données.
- Un registre contenant le code de l'instruction à exécuter.
- Convertisseur analogique numérique 5 canaux 10bit [12].

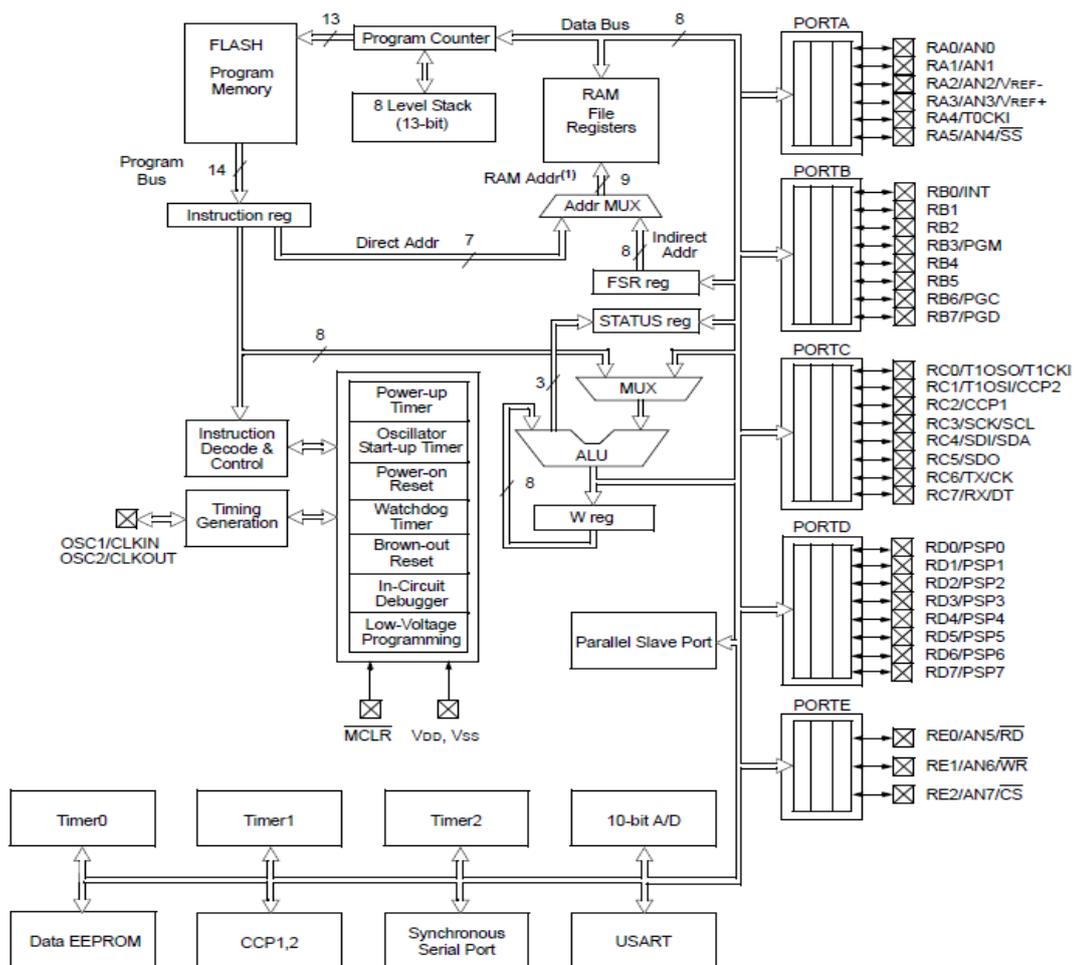


Figure.II.7: Architecture interne du PIC16F877

II.6.1 Caractéristiques principales du pic 16F877

Les caractéristiques principales du 16F877 sont:

- Une mémoire programme de type EEPROM flash de 8K mots de 14 bits.
- Une RAM donnée de 368 octets.
- Une mémoire EEPROM de 256 octets.

- Cinq ports d'entrée sortie, A (6 bits), B (8 bits), C (8 bits), D(8 bits), E(3 bits) :
 - Port A : 6 pins I/O numérotées de RA0 à RA5.
 - Port B : 8 pins I/O numérotées de RB0 à RB7.
 - Port C : 8 pins I/O numérotées de RC0 à RC7.
 - Port D : 8 pins I/O numérotées de RD0 à RD7.
 - Port E : 3 pins I/O numérotées de RE0 à RE2.
- Convertisseur Analogiques numériques 10 bits a 5 canaux.
- USART, Port série universel, mode asynchrone (RS232) et mode synchrone SSP(supportant le bus I2C).

- Trois TIMERS avec leurs Prescalers, TMR0, TMR1, TMR2.
- Deux modules de comparaison et Capture CCP1 et CCP2.
- Un chien de garde.
- 13 sources d'interruption.
- Générateur d'horloge, a quartz (jusqu' à 20 MHz) ou a Oscillateur RC.
- Protection de code.
- Fonctionnement en mode sleep pour réduction de la consommation.
- Programmation par mode ICSP (In Circuit Serial Programming) 12V ou 5V.
- Tension de fonctionnement de 2 à 5V. **[12]**

Le tableau suivant présente les caractéristiques générales de PIC16F877 :

La caractéristique	Valeur
Fréquence Horloge MHz	20MHz
Mémoire programme FLASH	8Ko
Mémoire données (RAM)	368 octets
Mémoire EEPROM	256 Octets
Interruptions	13
Ports parallèles (Nombre des lignes)	A(6), B(8) C(8), D(8), E(3)
Timers	3
CAN 10-bit	5
Vitesse du CPU (MIPS)	12
Périphériques de communication numérique	1- USART, 1-SSP (support I2C)
Capture / Comparateur	2 CCP
Comparateurs	2
Plage de tension de fonctionnement (V)	de 2 à 5 v
Température (C)	-40 à 85

Tableau 2 : Les caractéristiques générales de PIC16F877

II.7 Brochage du PIC16F877 :

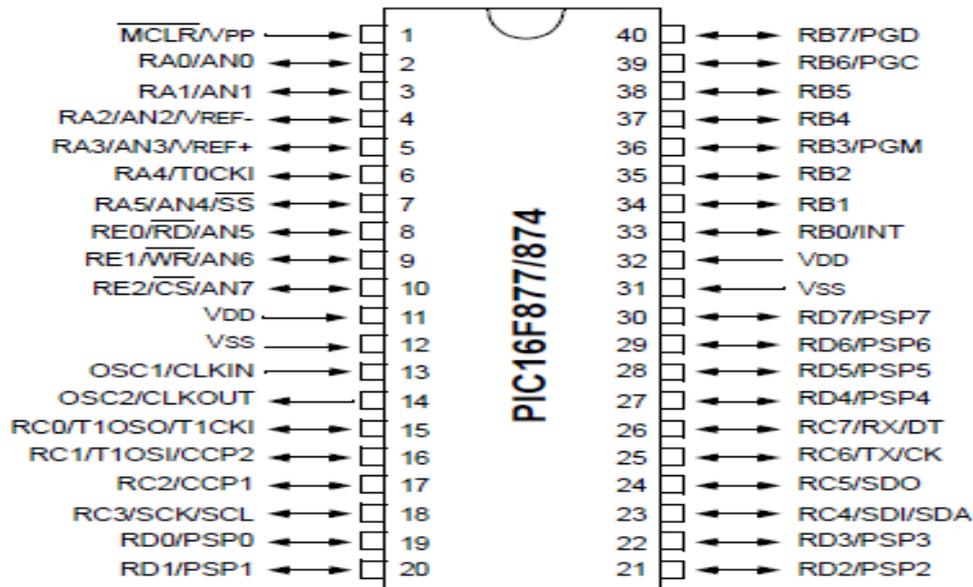


Figure.II.8: Brochage du PIC16F877

On peut distinguer sur ce schéma :

L'alimentation : VDD (+5V) et Vss (0V).

Les bornes du quartz (oscillateur a quartz) : OSC1 et OSC2.

L'entrée RESET (MCLR : Master clear).

Les différents ports d'Entrées/Sorties : RAx, RBx, RCx, RDx, Rex. [12]

II.7.1 L'horloge

Le rôle d'horloge est de cadencer le rythme d'exécution des instructions. Il y a 4 modes possibles pour réaliser l'horloge :

- LP Low-Power Crystal
- XT Crystal/Résonateur
- HS High-Speed Crystal/Résonateur
- RC résistance /Capacité

On utilise plus souvent un quartz (de 1MHZ jusqu'à 20MHZ) relié avec deux condensateurs de filtrage. [12]

II.7.3 Les ports d'E/S

Le PIC 16F877 dispose de 33 broches d'entrée sortie regroupées dans 5 ports PORTA, PORTB, PORTC, PORTD et PORTE. Chaque broche d'un port peut être configurée soit en entrée soit en sortie à l'aide des registres de direction TRISA, TRISB, TRISC et TRISD et TRISE:

Bit k de TRISx = 0 _ broche k de PORTx = SORTIE.

Bit k de TRISx = 1 _ broche k de PORTx = ENTREE.

Certains ports ont quelques particularités que nous allons détailler ci-dessous, [12]

II.7.3.1 Le port d'E/S PORTA

Le port A désigné par "PORTA" est un port de 6 bits (RA0 à RA5). RA6 et RA7 ne sont pas accessibles.

La configuration de direction se fait à l'aide du registre TRISA, positionner un bit de TRISA à 1 configure la broche correspondante de PORTA en entrée et inversement. Au départ toutes les broches sont configurées en entrée. [12]

II.7.3.2 Le port d'E/S PORTB

Le port B désigné par PORTB est un port bidirectionnel de 8 bits (RB0 à RB7). Toutes les broches sont compatibles TTL.

La configuration de direction se fait à l'aide du registre TRISB, positionner un bit de TRISB à 1 configure la broche correspondante de PORTB en entrée et inversement. Au départ toutes les broches sont configurées en entrée..

En entrée, la ligne RB0 appelée aussi INT peut déclencher l'interruption externe INT.

En entrée, une quelconque des lignes RB4 à RB7 peut déclencher l'interruption RBI. [12]

II.7.3.3 Le port d'E/S PORTC

Le port C désigné par PORTC est un port bidirectionnel de 8 bits (RC0 à RC7). Toutes les broches sont compatibles TTL.

La configuration de direction se fait à l'aide du registre TRISC, positionner un bit de TRISC à 1 configure la broche correspondante de PORTC en entrée et inversement. Au départ toutes les broches sont configurées en entrée.

Toutes les broches du port C peuvent être utilisées soit comme E/S normales soit comme broches d'accès à différents modules comme le timer 1, les modules de comparaison et de capture CCP1/2, le timer 2, le port I2C ou le port série. [12]

Pour l'utilisation d'une broche du port C comme E/S normale, il faut s'assurer qu'elle n'a pas été affectée à un de ces modules. Par exemple, si TIMER1 est valide, il peut utiliser les broches RC0 et RC1 selon sa configuration. [12]

II.7.3.4 Le port d'E/S PORTD

Le port D désigné par PORTD est un port bidirectionnel de 8 bits (RD0 à RD7). Toutes les broches sont compatibles TTL et ont la fonction trigger de Schmitt en entrée.

Chaque broche est configurable en entrée ou en sortie à l'aide du registre TRISD. Pour configurer une broche en entrée, on positionne le bit correspondant dans TRISD à 1 et inversement.

PORTD peut être utilisé dans un mode particulier appelé parallèle slave port, pour cela il faut placer le bit PSPMODE (bit 4) de TRISE à 1. Dans ce cas les 3 bits de PORTE deviennent les entrées de contrôle de ce port (RE, WE et CS). [12]

II.7.3.5 Le port d'E/S PORTE

PORTE contient seulement 3 bits RE0, RE1 et RE2. Les 3 sont configurables en entrée ou en sortie à l'aide des bits 0, 1 ou 2 du registre TRISE.

Les 3 bits de PORTE peuvent être utilisés soit comme E/S numérique soit comme entrées analogiques du CAN. La configuration se fait à l'aide du registre ADCON1.

Si le bit PSPMODE de TRISE est placé à 1, Les trois bits de PORTE deviennent les entrées de contrôle du PORTD qui (dans ce cas) fonctionne en mode parallèle slave mode.

À la mise sous tension (RESET), les 3 broches de PORTE sont configurées comme entrées analogiques. [12]

II.8 le bus I2C et le PIC 16F877

Le PIC16F87x possède des entrées/sorties spécialement prévues pour la communication I2C. Ce sont les broches nommées **RC3 (SCL)** et **RC4 (SDA)**. Elles sont directement reliées à un contrôleur I2C.

Ce contrôleur prévoit la gestion de tout le protocole I2C (Prise de parole, collision, ligne occupée...). Il permet une gestion facile de la liaison, étudier en détail au chapitre trois.

II.8 .1 Organisation de la mémoire

Le microcontrôleur PIC16F877 possède une architecture de type Harvard avec 2 plans mémoires :

Une mémoire de programme de 8 ko (mémoire FLASH).

Une mémoire de données de 368 octets.

II.9 Conclusion

Nous avons vu une étude détaillée sur microcontrôleurs et leur architecture et organisation de leur mémoire. Après comparaison des technologies CISC et RISC on s'est intéressé au microcontrôleur PIC 16F877 conçu en technologie la plus récente, avec des composants RISC et une architecture HARVARD modifiée.

Chapitre III : Le protocole bus I2C

III.1 Introduction.....	28
III.2 Présentation et caractéristiques du bus I2C.....	28
III.3 Le protocole I2C.....	30
III.3.1 La prise contrôle du bus.....	30
III.3.2 La transmission d'un octet.....	30
III.3.3 La transmission d'une adresse.....	31
III.3.4 Écriture d'une donnée.....	32
III.3.5 Lecture d'une donnée.....	33
III.4 Présentation du MikroC.....	36
III.4.1 Création d'un projet.....	37
III.4.2 Les bibliothèques du MikroC.....	39
III.5 Bus I2C et MikroC.....	40
III.5.1 Fonction d'écriture.....	40
III.5.2 Fonction de lecture.....	41
III.6 Conclusion.....	41

III.1 Introduction :

Le bus I2C (Inter Integrated Circuit) a été développé au début des années 80 par Philips semi-conducteurs pour permettre de relier facilement à un microprocesseur les différents circuits d'un téléviseur moderne et contrôlé le bus I2C avec microcontrôleur PIC En utilisant la logiciel et choisissez le programme MikroC.

III.2 Présentation et caractéristiques du bus I2C :

Le bus I2C a été inventé pour simplifier la communication entre des composants divers dont le sigle signifie Inter Integrated Circuit ce qui donne IIC et par contraction I2C. Le protocole est initialement proposé par Philips mais adopté de nos jours par de très nombreux fabricants. C'est un bus de communication de type série synchrone.

Ce bus est parfois nommé sous le nom de TWI (Two Wire Interface). Il permet de communiquer sur trois fils :

- Signal de donnée: SDA (Serial Data)
- Signal d'horloge: SCL (Serial Clock)
- Signal de référence électrique : masse de périphériques.

Ceci permet de réaliser des équipements ayant des fonctionnalités très puissantes (En apportant toute la puissance des systèmes microprogrammes) et conservant un circuit imprimé très simple, par rapport un schéma classique (8bits de données, 16 bits d'adresse + Les bits de contrôle).

Les données sont transmises en série à 100Kbits/s en mode standard et jusqu'à 400Kbits/s en mode rapide. Ce qui ouvre la porte de cette technologie à toutes les applications où la vitesse n'est pas primordiale.

Le nombre de composants qu'il est ainsi possible de relier est essentiellement limité par la charge capacitive des lignes SDA et SCL : 400 pF. [13]

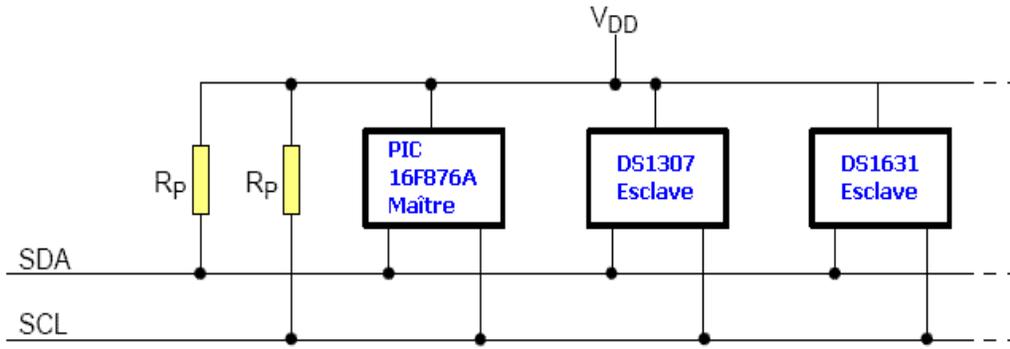


Figure.III : Le bus I2C

Le bus I2C est contrôlé par un maître (c'est généralement un microcontrôleur, par exemple un PIC 16F876A). Afin d'éviter les conflits électriques les Entrées/Sorties SDA et SCL sont de type «Collecteur Ouvert" Voir la (figure.III.1):

- Deux résistances de pull-up sont bien sûr placées entre les lignes SDA et SCL et l'alimentation (VDD).
- Quand le bus n'est pas utilisé, SDA et SCL sont au niveau haut (niveau de repos).
- Quand une ligne (SDA ou SCL) est au repos (niveau 1), on peut la forcer à 0.
- Quand une ligne (SDA ou SCL) est au niveau 0, on ne peut pas la forcer à 1.

Dans l'exemple ci-dessous, l'horloge SCL est au niveau 0 car c'est le PIC 16F877A (maître) qui force cette ligne à 0. [14]

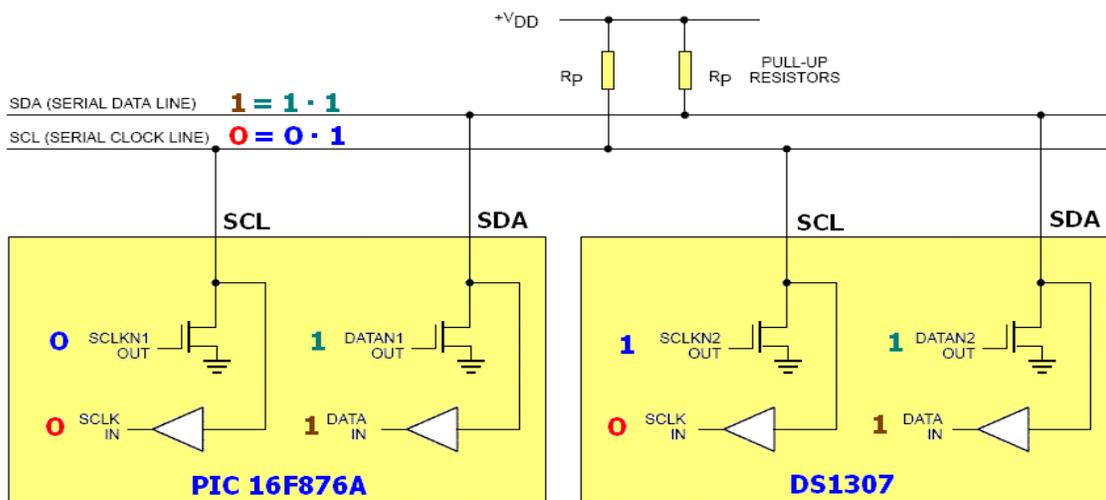


Figure.III.1 : Structure d'E/S d'un module I2C

III.3 Le protocole I2C :

Le protocole I2C définit des états logiques possibles sur SDA et SCL, et la façon dont doivent réagir les circuits en cas de conflits.

III.3.1 La prise de contrôle du bus :

Pour prendre le contrôle du bus, il faut que celui-ci soit au repos (SDA et SCL à '1'). Pour transmettre des données sur le bus, il faut donc surveiller deux conditions particulières :

- La condition de départ. (SDA passe à '0' alors que SCL reste à '1')
- La condition d'arrêt. (SDA passe à '1' alors que SCL reste à '1')

Lorsqu'un circuit, après avoir vérifié que le bus est libre, prend le contrôle de celui-ci, il en devient le **maître**. C'est lui qui génère le signal d'horloge.

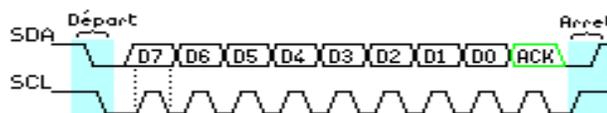


Figure.III.2: Exemple de condition de départ et d'arrêt

III.3.2 La transmission d'un octet :

Après avoir imposé la condition de départ, le maître applique sur SDA le bit de Poids fort D7. Il valide ensuite la donnée en appliquant pendant un instant un niveau '1' sur la ligne SCL. Lorsque SCL revient à '0', il recommence l'opération jusqu'à ce que l'octet complet soit transmis. Il envoie alors un bit ACK à '1' tout en scrutant l'état réel de SDA. L'esclave doit alors imposer un niveau '0' pour signaler au maître que la transmission s'est effectuée correctement. Les sorties de chacun étant à collecteurs ouverts, le maître voie le '0' et peut alors passer à la suite.

Exemple de transmission réussie.

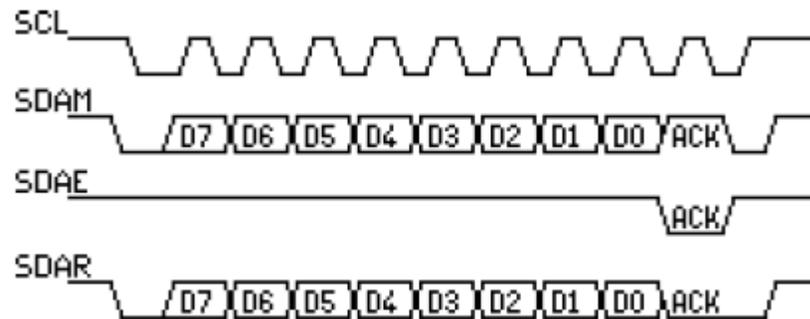


Figure.III.3: Exemple de transmission réussie

Dans cet exemple :

- SCL : Horloge imposée par le maître.
- SDAM : Niveaux de SDA imposés par le maître.
- SDAE : Niveaux de SDA imposés par l'esclave.
- SDAR : Niveaux de SDA réels résultants.

III.3.3 La transmission d'une adresse :

Le nombre de composants qu'il est possible de connecter sur un bus I2C étant largement supérieur à deux, il est nécessaire de définir pour chacun une adresse unique. L'adresse d'un circuit codée sur sept bits, est définie d'une part par son type et d'autre part par l'état appliqué à un certain nombre de ces broches. Cette adresse est transmise sous la forme d'un octet au format particulier.



Figure.III.5: Exemple d'octet d'adresse

On remarque ici que les bits D7 à D1 représentent les adresse A6 à A0, et que le bit D0 et remplacé par le bit de R/W qui permet au maître de signaler s'il veut lire ou écrire une donnée. [13]

III.3.4 Écriture d'une donnée :

- Exemple d'écriture du DS1307

L'émetteur est le maître et le récepteur est l'esclave.

Le registre d'adresse 0x04 du DS1307 contient la date.

Pour régler le calendrier au 27 du mois, il faut écrire la donnée 0x27 (codage BCD) dans le registre d'adresse 0x04 du DS1307.

Le bus I2C utilise le protocole suivant :

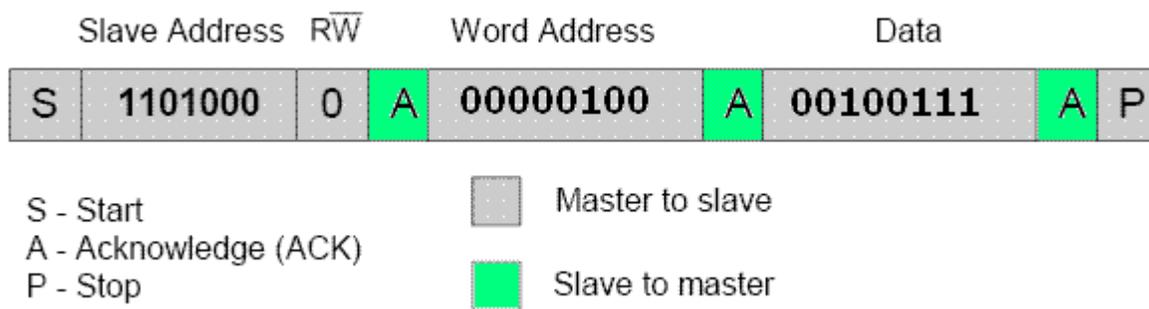


Figure.III.6 : Ecriture d'une donnée (ds1307)

- 1) Pour initier le dialogue, le maître crée une condition **Start**.
- 2) Le maître envoie l'adresse de l'esclave (11001000) suivi du bit 0 (bit Write).
- 3) L'esclave répond (accusé de réception : bit Acknowledge).
- 4) Le maître envoie l'adresse du registre (0x04) à écrire.
- 5) L'esclave répond (accusé de réception : bit Acknowledge).
- 6) Le maître envoie la donnée (0x27) à écrire.

7) L'esclave écrit la donnée puis envoie un accusé de réception (bit Acknowledge).

8) Le maître termine le dialogue avec une condition **Stop** .

Le bus I2C est maintenant libre (SCL = 1, SDA = 1 : niveaux de repos).

Un peu moins évident, le chronogramme correspondant des lignes SDA et SCL :

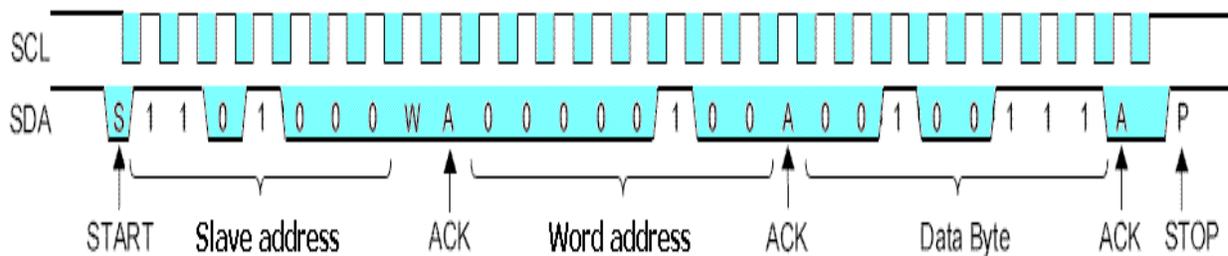


Figure.III.7 : le chronogramme correspondant des lignes SDA et SCL [14]

III.3.5 Lecture d'une donnée :

- Exemple de lecture du DS1307

L'émetteur est l'esclave et le récepteur est le maître.

Les registres d'adresses 0x00 à 0x06 du DS1307 contiennent respectivement les secondes, minutes, heures, jours, dates, mois et années.

Voici comment lire, le contenu des registres d'adresses 0x00 à 0x06 du DS1307

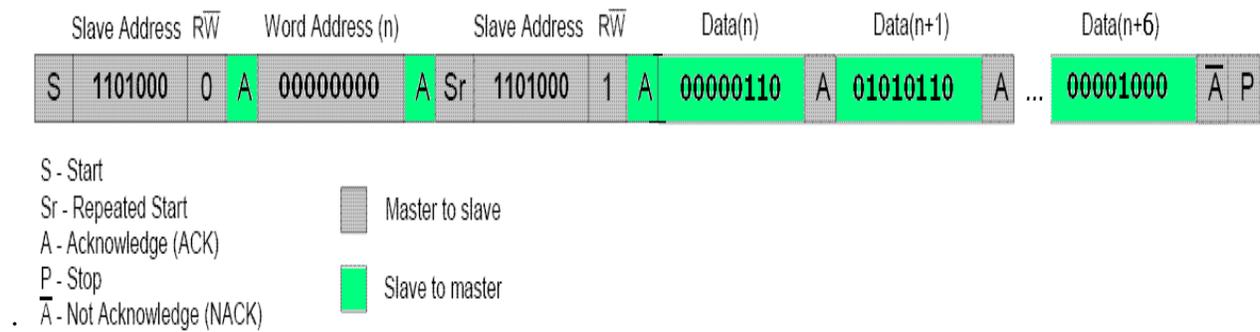


Figure.III.8: Lecture d'une donnée (ds1307)

- 1) Pour initier le dialogue, le maître crée une condition **Start**.
- 2) Le maître envoie l'adresse de l'esclave (1010100) suivi du bit 0 (bit Write).
- 3) L'esclave répond (accusé de réception : bit Acknowledge).
- 4) Le maître envoie l'adresse du registre (0x00) à lire.
- 5) L'esclave répond (accusé de réception : bit Acknowledge).
- 6) Le maître émet une condition **Repeated Start**.
- 7) Le maître envoie l'adresse de l'esclave (1010100) suivi du bit 1 (**bit Read**).
- 8) L'esclave répond (accusé de réception : bit Acknowledge).
- 9) L'esclave envoie le contenu du registre d'adresse 0x00 au maître.
- 10) **Le maître répond** (accusé de réception : bit Acknowledge).
- 11) L'esclave envoie le contenu du registre d'adresse 0x01 (automatiquement incrémenté) au maître.
- 12) Le maître répond (accusé de réception : bit Acknowledge)

13) L'esclave envoie le contenu du registre d'adresse 0x02 (automatiquement incrémenté) au maître

14) Le maître répond (accusé de réception : bit Acknowledge)

21) L'esclave envoie le contenu du registre d'adresse 0x06 (automatiquement incrémenté) au maître

22) Le maître répond (accusé de réception : bit **Not Acknowledge**)

23) Le maître termine le dialogue avec une condition Stop

Le bus I2C est maintenant libre (SCL = 1, SDA = 1 : niveaux de repos).

Le contenu du registre d'adresse 0x00 du DS1307 est 0x06 (codage BCD : 06 secondes).

Le contenu du registre d'adresse 0x01 est 0x56 (c'est-à-dire 56 minutes).

Le contenu du registre d'adresse 0x02 est 0x09 (c'est-à-dire 09 heures).

Le contenu du registre d'adresse 0x03 est 0x03 (c'est-à-dire Mardi).

Le contenu du registre d'adresse 0x04 est 0x07 (c'est-à-dire 7 ème jour du mois).

Le contenu du registre d'adresse 0x05 est 0x05 (c'est-à-dire mois de mai).

Le contenu du registre d'adresse 0x06 est 0x0D (c'est-à-dire année 2013).

(Mardi 07 mai 2013, 9 heures 56 minutes et 6 secondes). [14]

III.3.6 langage de programmation:

La programmation des microcontrôleurs PIC est supportée par plusieurs langages de programmation tel que : MikroC for pic, MPLAB, Mikrobasic, HI-TECH C for PIC, flow code, etc... Dans le chapitre présent nous présentons la procédure de programmation des microcontrôleurs... Dans notre projet nous avons opté pour le compilateur de MIKRO C qui est un compilateur en langage C (langage évolué) et qui permet d'intégrer certaines routines en assembleur. Ce choix est à la fois un choix personnel et un choix technologique parce que il est basé sur le langage C.

III.4 Présentation du MikroC :

Le « MikroC » est un compilateur pour PIC Conçu par la société « Mikroelektronika », le compilateur C nouvelle génération "MikroC" pour microcontrôleurs PIC bénéficie d'une prise en main très facile. Il comporte plusieurs outils intégrés (mode simulateur, terminal de communication, gestionnaire 7 segments, analyseur statistique, correcteur d'erreur, explorateur de code...), Il a une capacité à pouvoir gérer la plupart des périphériques rencontrés dans l'industrie (Bus I2C, 1Wire, SPI, RS485, Bus CAN, cartes compact Flash, signaux PWM, afficheurs LCD et 7 segments...), de ce fait il est un des outils de développement incontournable et puissant.

Il est conçu pour fournir les solutions les plus faciles que possibles pour des applications se développant pour les systèmes à microcontrôleur. Il contient un large ensemble de bibliothèques de matériel, de composant et la documentation complète. [15]

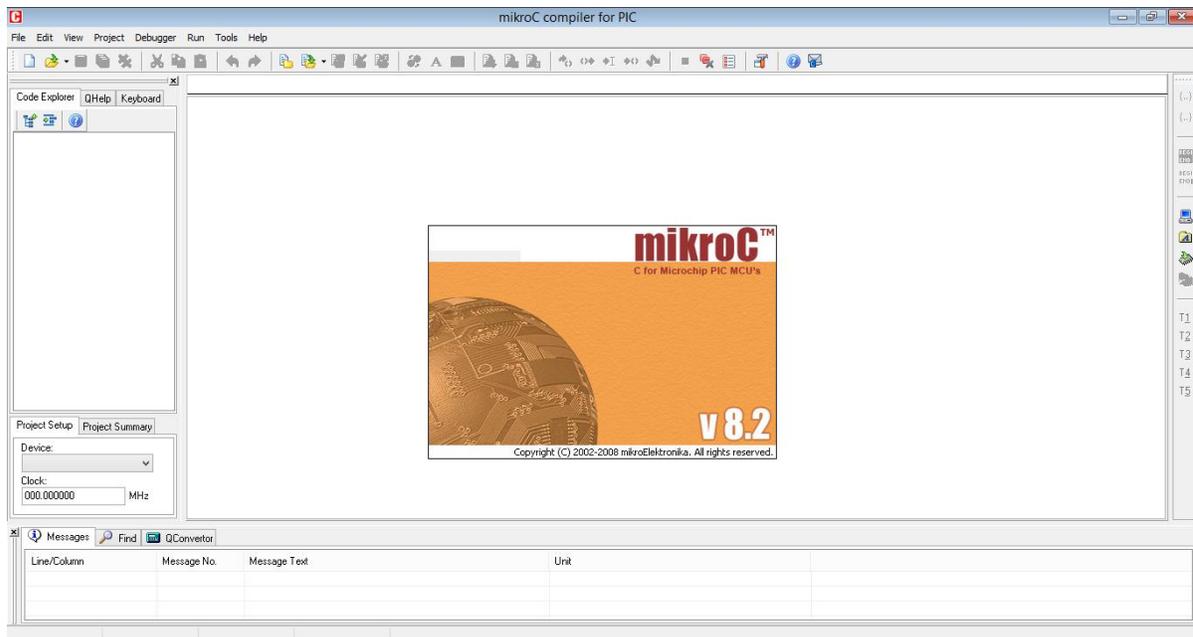


Figure.III.9: Interface du logiciel MikroC

III.4.1 Création d'un projet :

Le processus de création d'un nouveau projet est vraiment très simple. Sélectionnez New Project (Nouveau Projet) de puis le menu Project (Projet), comme indiqué sur la (figure.III.10).

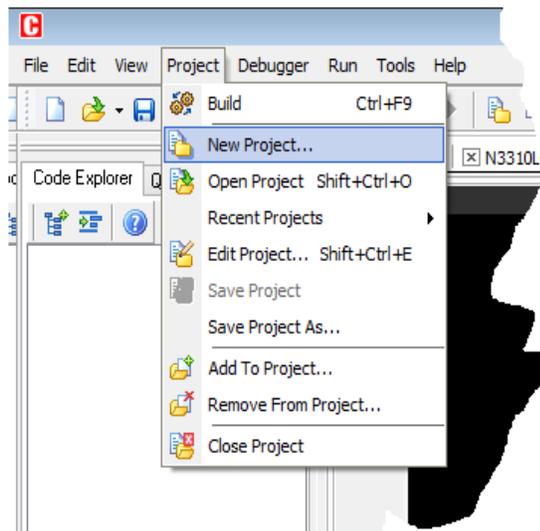
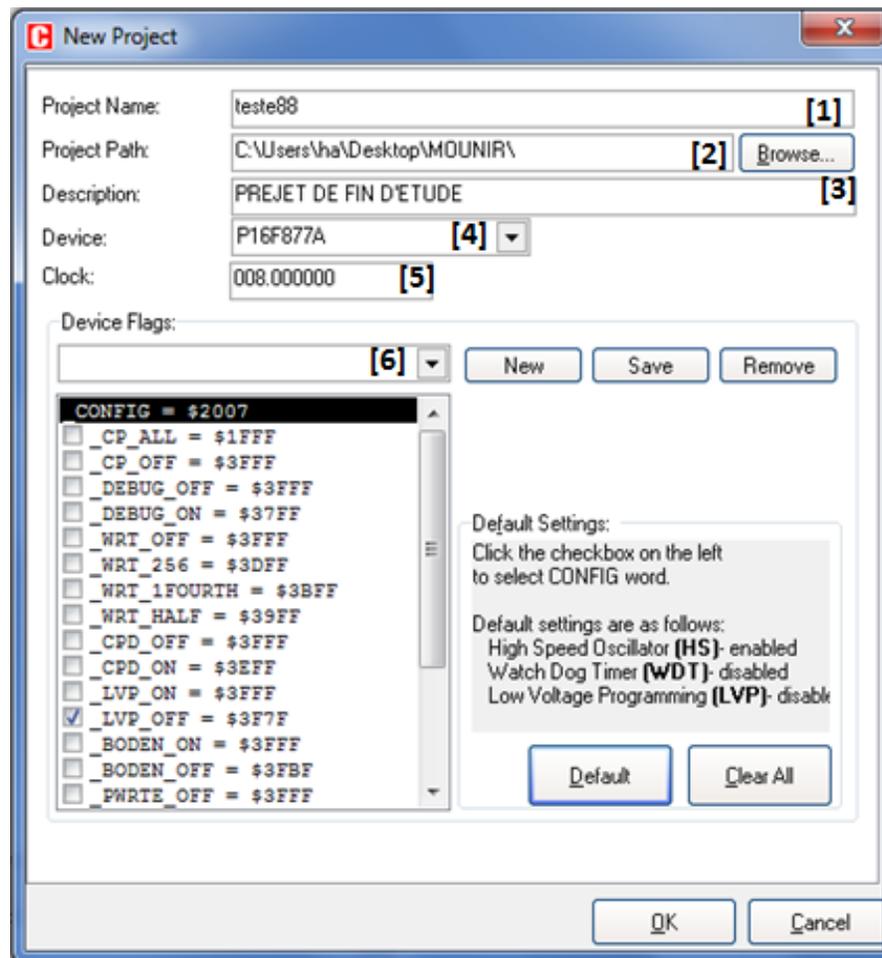


Figure.III.10 : Création d'un projet en MikroC

Une nouvelle fenêtre apparaîtra. Comme indiqué sur (figure.III.11), il y a plusieurs champs à renseigner comme le nom du projet, l'emplacement du projet, sa description, l'horloge et les options du composant. Le tableau device flags (options composant) est utilisé pour la configuration des paramètres du microcontrôleur.



- [1] : Nom de projet.
- [2] : Nom de l'emplacement du projet.
- [3] : La description du projet.
- [4] : Le choix du PIC.
- [5] : Le choix de la fréquence d'horloge interne du PIC.
- [6] : Les paramètres du PIC.

Figure.III.11 : Les configurations de projet

Après ça, une nouvelle fenêtre (figure.III.12) vide s'affiche pour écrire notre programme

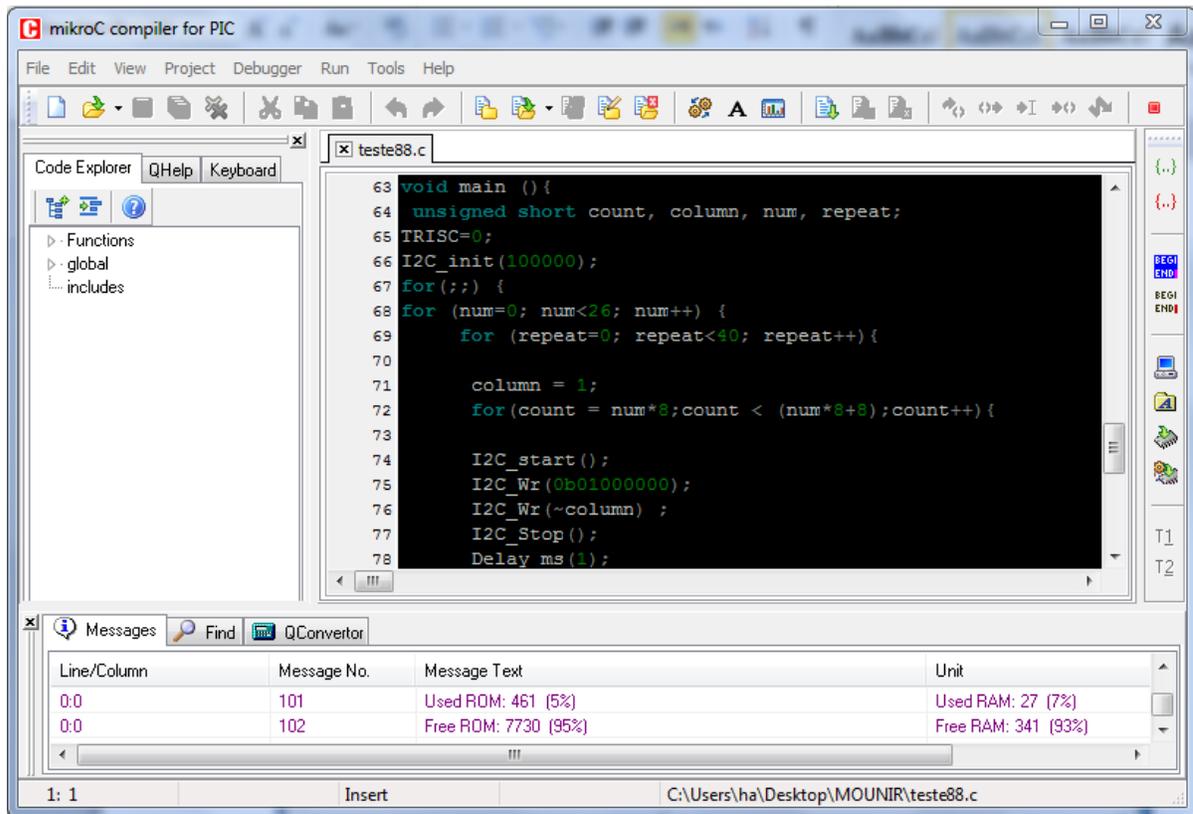


Figure.III.12: Fenêtre de saisie de programme

III.4.2 Les bibliothèques du MikroC :

MikroC fournit quelques bibliothèques et nous avons utilisé uniquement les bibliothèques suivantes :

Bibliothèque	Les fonctions	La description
I2C	I2C_Init	Initialise I ² C avec horloge souhaitée Exemple : I2C_Init(100000);
	I2C_Start	Détermine si I ² C bus est libre et lancer un signal

		START. Exemple : I2C_Start ();
	I2C_Repeated_Start	Répéter un signal START. Exemple : I2C_Repeated_Start();
	I2C_Rd	Lit un octet de l'esclave, et envoie pas signal d'acquittement ACK si le paramètre est 0, sinon il envoie reconnaître. Exemple : take = I2C_Rd(0);
	I2C_Wr	Envoie octet de données (données de paramétrage) par l'intermédiaire du bus I2C. Exemple : I2C_Write(0xA3);
	I2C_Stop	Lancer un signal STOP. Exemple : I2C_Stop();

Tableau 3: Bibliothèque du MikroC

III.5 Bus I2C et MikroC :

MikroC fournit une bibliothèque prête du bus I2C (Tableaux 3), nous utilisons cette bibliothèque pour communiquer entre le maître (PIC16F877A) et l'esclave (PCF8574). L'exemple suivant montre comment écrire en l'esclave PCF8574 avec les commentaires.

III.5.1 Fonction d'écriture :

```
void write_pcf8574 (unsigned short column)
{
I2C_Start (); // lancer un signal de départ (le bit de start)
I2C_Wr (0x40); // envoyer un octet via I2C (adresse de composant + un bit de mode action (R/W))
I2C_Wr (column); // envoyer des données (Les données qui seront écrites)
I2C_Stop (); // envoyer un signal de stop (bit de stop) }
```

Un exemple de l'utilisation de cette fonction :

```
write_pcf8574 (column); // afficher le colon de led matrice
```

III.5.2 Fonction de lecture :

```
void read_ (unsigned short address)
{
I2C_Start (); // lancer un signal de départ (le bit de Start)
I2C_Wr (0xD0); // envoyer un octet via I2C (adresse de composant + un bit de mode action (R/W))
I2C_Wr (address); // envoyer un octet (envoyer l'adresse du registre à lire)
I2C_Wr (0xD1); // envoyer un octet via I2C (adresse de composant + un bit de mode action (R/W))
I2C_Stop (); // envoyer un signal de stop (bit de stop)}
```

III.6 Conclusion

Dans ce chapitre nous avons étudié le protocole de bus I2C c'est un protocole de communication rapide et facile à mettre en œuvre. Nous avons présenté le principe de lecture et écriture d'une donnée. Il permet de créer des bus contenant plusieurs maîtres et esclaves. Le Logiciel de programmation mikroC a été utilisé l'implantation de ce protocole.

Chapitre IV : Conception de la carte et simulation

IV.1 Introduction.....	42
IV.2 Schéma synoptique du projet.....	42
IV.3 Présentation du logiciel ISIS.....	43
IV.3.1 Fenêtre d'ensemble (Vue d'ensemble).....	43
IV.3.2 Fenêtre d'édition.....	44
IV.3.3 La boîte à outils.....	44
IV.3.4 Organisation de la boîte à outils.....	44
IV.4 Simulation par ISIS.....	45
IV.4.1 descriptif des composants utilisés.....	45
IV.4.1.1 Alimentation.....	45
IV.4.1.2 Unité de commande.....	45
IV.4.1.3 Trois ports séries - parallèles PCF8574.....	46
IV.4.1.4 Deux matrices LED.....	48
IV.5 Fonctionnement de système.....	49
IV.6 organigramme de système.....	50
IV.7 Conclusion.....	52

IV.1 Introduction

D'après ce que nous avons vu dans les chapitres précédents sur les l'étude théorique des microcontrôleurs et logicielle de programmation. L'objectif de ce projet est de mettre en œuvre une application à microcontrôleur qui permet d'afficher des caractères sur un afficheur graphique à diode LED composé de deux matrices (8*8) extensibles. Les LED sont commandées via des circuits série parrèle utilisant le protocole I2C. Nous utilisons logiciel de simulation proteus 7.

IV.2 Schéma synoptique du projet

La figure ci-dessous donner le schéma synoptique de certain système d'affichage à élément en trouver :

- Le microcontrôleur pic 16F877
- Trois circuits série parrèle à bus I2C
- Matrice à LED

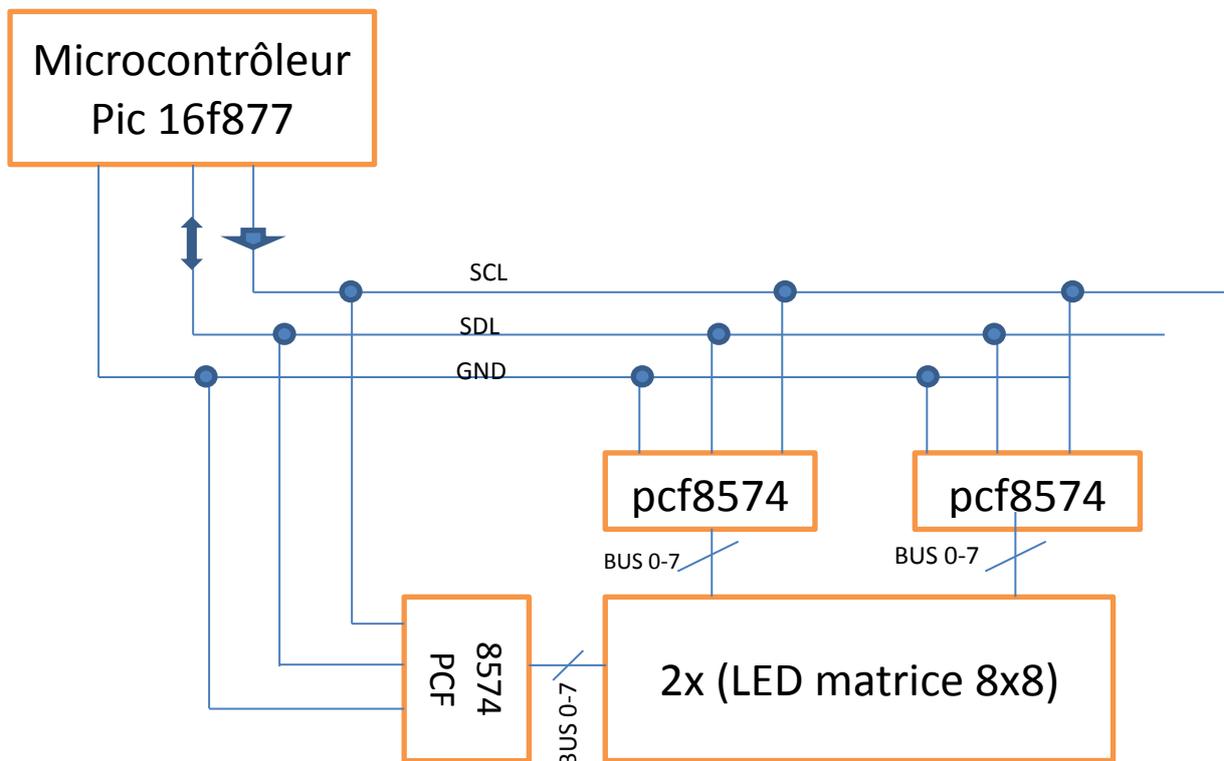


Figure.IV.1 : Schéma synoptique du projet

IV.3 Présentation du logiciel ISIS

Isis est un éditeur de schémas qui intègre un simulateur analogique / logique ou mixte. Toutes les opérations se passent dans cet environnement, aussi bien la configuration des différentes sources que le placement des sondes et le tracé des courbes.

Dans cette section nous allons commencer par la présentation de la fenêtre du logiciel ISIS (Figure.IV.2). [16]

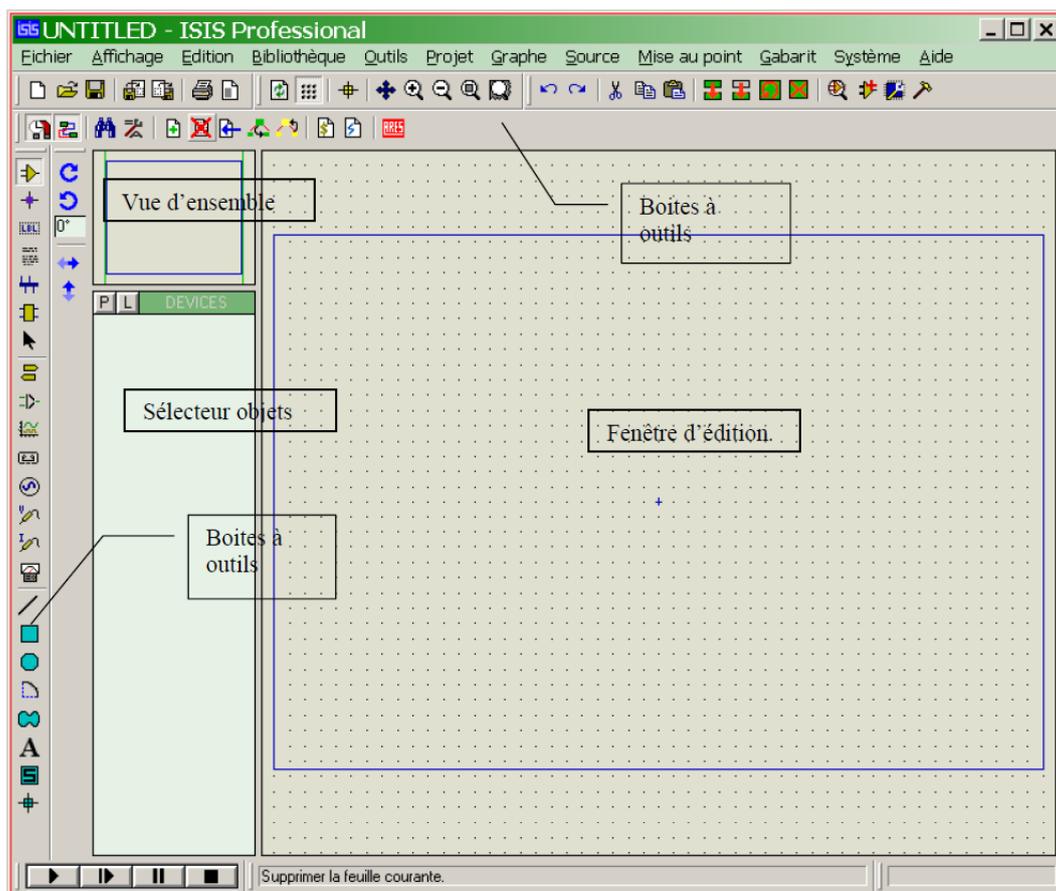


Figure.IV.2: Interface du logiciel ISIS

IV.3.1 Fenêtre d'ensemble (Vue d'ensemble)

Le cadre en bleu délimite l'espace de travail tel qu'il a été défini par la commande '*Définir taille des feuilles*' du menu '*système*'.

Le cadre en vert délimite La zone de travail, c'est à dire la partie du schéma visible dans la fenêtre principale.

- Vous pouvez déplacer cette zone de travail en pointant la souris sur la zone désirée de la fenêtre d'ensemble et en effectuant un clic gauche.
- Vous pouvez redéfinir la zone de travail dans la fenêtre d'ensemble en appuyant sur la touche majuscule 'shift' du clavier, associée au déplacement de la souris en maintenant appuyé le bouton gauche. [17]

IV.3.2 Fenêtre d'édition

La surface la plus grande de l'écran s'appelle "Fenêtre d'édition" et se comporte comme une fenêtre de dessin. C'est là que vous placez et câblez les composants.

IV.3.3 La boîte à outils

Elle est composée d'un ensemble d'icônes dont les fonctions seront détaillées ultérieurement et d'un sélecteur d'objet utilisé pour choisir les boîtiers, le style des pastilles, des traces, des traversées, etc. [17]

IV.3.4 Organisation de la boîte à outils



Figure.IV.3: Organisation de la boîte à outils ISIS

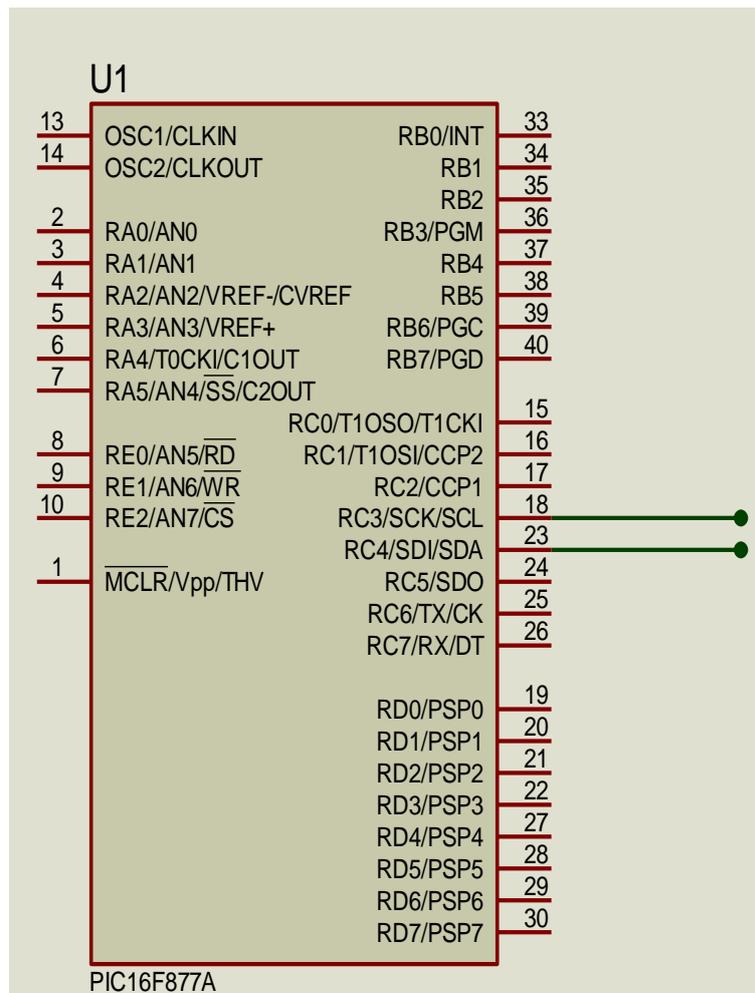


Figure.IV.5: configuration le port c (SDA et SCL)

IV.4.1.3 Trois ports séries - parallèles PCF8574

Le PCF8574 est un épandeur 8-bit E/ S pour le bus I2C produite par Philips Semi-conducteurs. Il est conçu pour fournir un bus I2C interface avec jusqu'à 8 appareils différents.

Le circuit permet de gérer 8 lignes d'entrée / sorties parallèles via le bus I2C. ADRESSE : 0 1 0 0 A2 A1 A0 R/W : où A2, A1 et A0 sont les niveaux logiques appliqués sur les broches du même nom. Lorsque le composant reçoit son adresse et que le bit R/W est positionné à 0, alors l'octet suivant l'adresse dans la trame est mémorisé et présenté sur les 8 lignes d'entrée / sorties.

Lorsque le composant reçoit son adresse et que le bit R/W est positionné à 1, alors le 8574 vient lire l'état réel qui est présent sur les 8 lignes d'entrée / sorties et le renvoi sous forme d'un octet placé à la suite de l'adresse dans la trame.

IV.4.1.3.1 Brochage du PCF8574

Port parallèle PCF8574 est commercialisé dans boîtier 16 broches classiques, comme illustré dans la (figure.IV.6). [18]

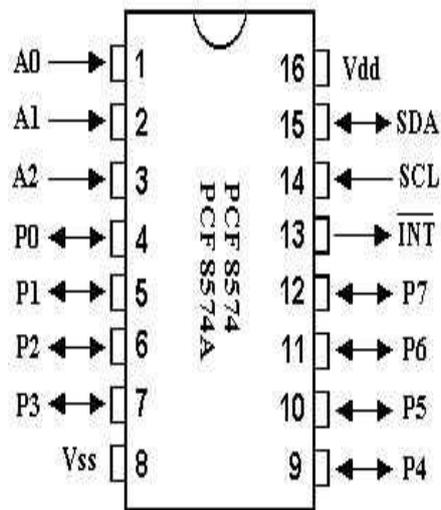


Figure.IV.6: Brochage du PCF8574

SYMBOLE	broche	Description
A0	1	adresse entrée 0
A1	2	adresse entrée 1
A2	3	adresse entrée 2
P0	4	quasi-bidirectionnel E/S 0
P1	5	quasi-bidirectionnel E/S 1
P2	6	quasi-bidirectionnel E/S 2
P3	7	quasi-bidirectionnel E/S 3
VSS	8	masse de l'alimentation
P4	9	quasi-bidirectionnel E/S 4
P5	10	quasi-bidirectionnel E/S 5
P6	11	quasi-bidirectionnel E/S 6
P7	12	quasi-bidirectionnel E/S 7
INT	13	sortie d'interruption
SCL	14	La ligne d'horloge en série
SDA	15	La ligne de données en série
Vdd	16	la tension d'alimentation

Tableau 4: Les éléments d'un PCF8574

IV.4.1.4 Deux matrices led

Dans un dispositif d'affichage LED point matrice, plusieurs diodes sont connectées ensemble à les lignes et les colonnes. Cette connexion est établie afin de réduire le nombre de broches nécessaires pour les manipuler, nous allons utiliser une matrice de LED 8x8, ce qui exigerait 64 broches d'E / S, un pour chaque pixel.

Raccordement de toutes les anodes d'ensemble (R1 à R8) et des cathodes à partir de la colonne (C1 à C8), le nombre requis de broches d'E / S est réduit à 16. Chaque LED est accessible par son numéro de ligne et de colonne.

Dans la figure ci-dessous, si R5 est à 1 logique et C3 est à 0 logique, la LED sur la ligne 5 colonne 3 sera allumé.

Les caractères sont affichés par un balayage rapide de chaque ligne et colonne. [19]

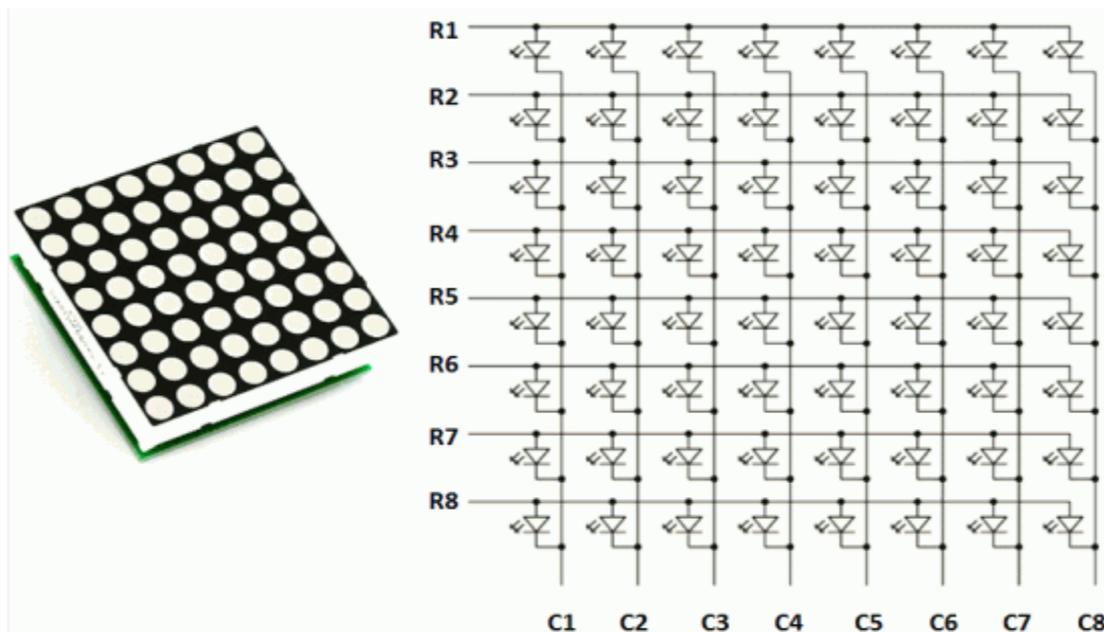


Figure.IV.7: La structure de matrice de points LED 8x8

Le tableau ci-dessous montre les niveaux logiques, qui sont appliquées à des lignes et des colonnes, pour afficher le caractère "X". Au bas du tableau le code en hexadécimal. [19]

R1	0	1	0	0	0	0	1	0
R2	0	1	0	0	0	0	1	0
R3	0	0	1	0	0	1	0	0
R4	0	0	0	1	1	0	0	0
R5	0	0	0	1	1	0	0	0
R6	0	0	1	0	0	1	0	0
R7	0	1	0	0	0	0	1	0
R8	0	1	0	0	0	0	1	0
	C1	C2	C3	C4	C5	C6	C7	C8
HEX	00	C3	24	18	18	24	C3	00

Figure.IV.8: Le niveau logique de la matrice 8x8

IV.5 Fonctionnement du système

Dans notre système de commande il ya un seul mode fonctionnement de base alors le microcontrôleur pic communique avec les 3 portes parallèles pcf8574 à partir de ce circuit.

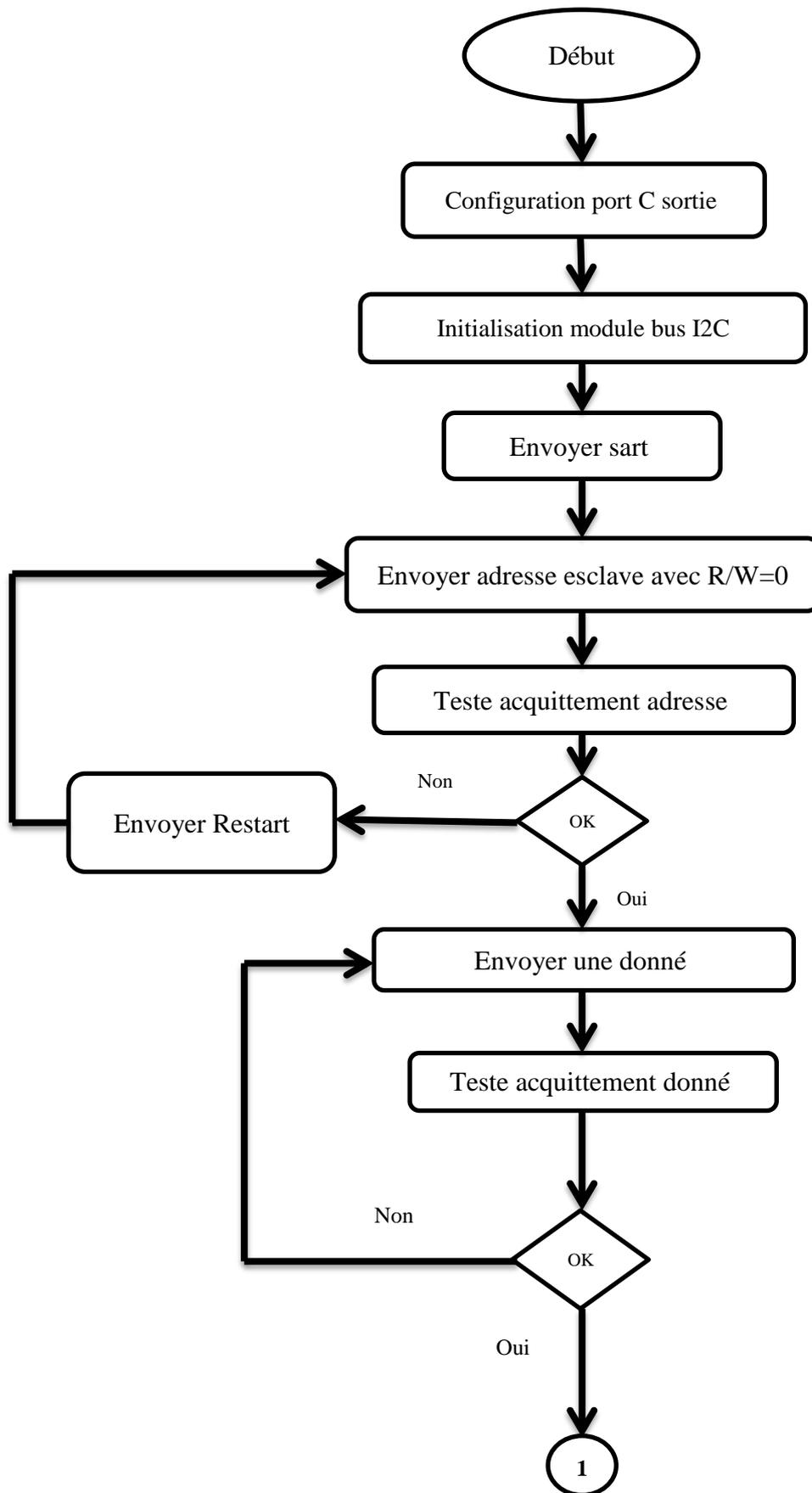
Le PIC émit les données en bus I2C pour pcf8574 en début envoyer Start, envoyer adresse avec R/W=0, teste ack adresse, envoyer une donnée, teste ack, allumer les colonnes de deux matrices, envoyer stop.

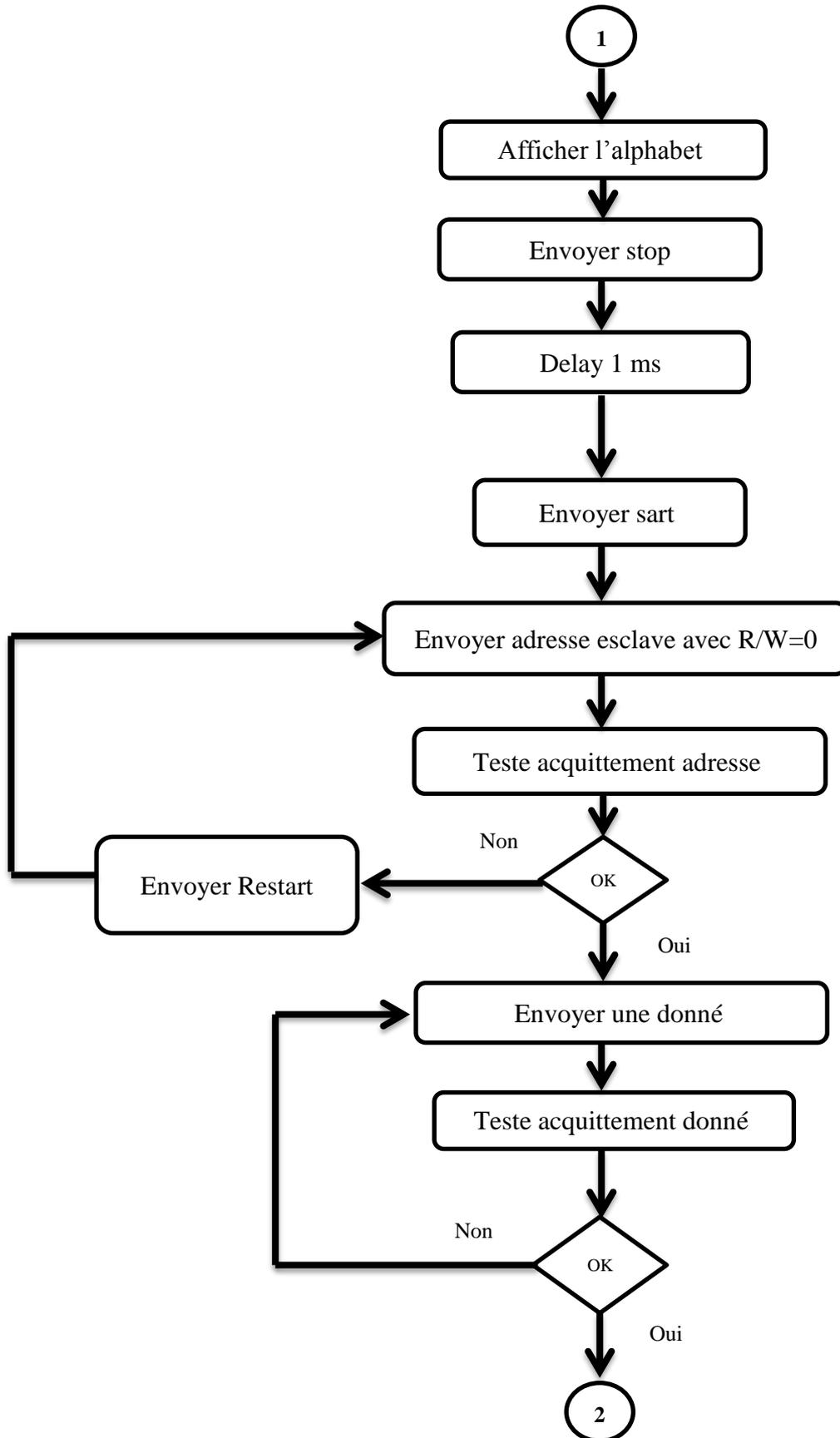
Le deuxième pcf8574 en début envoyer Start, envoyer adresse avec R/W=0, teste ack adresse, envoyer une donnée, teste ack, afficher l'alphabet, envoyer stop.

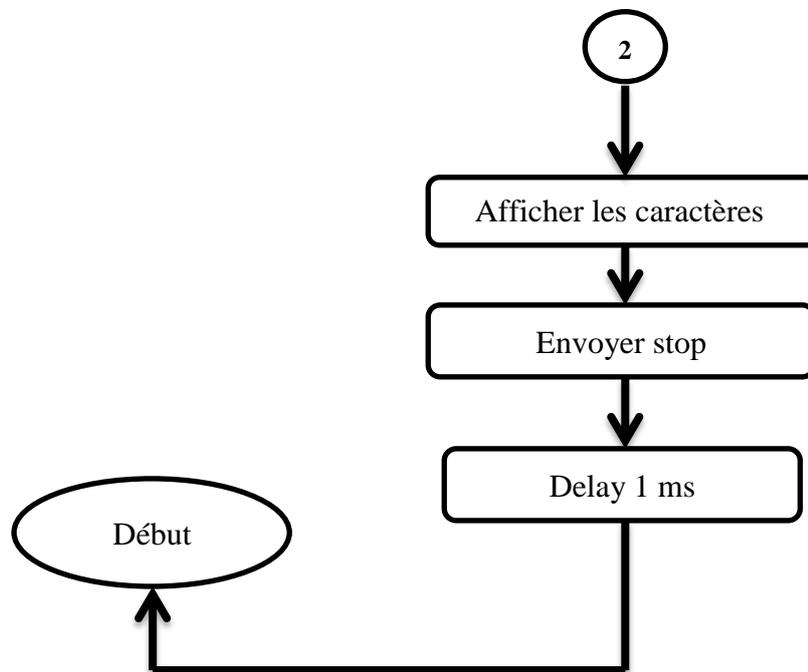
Le troisième pcf8574 en début envoyer Start, envoyer adresse avec R/W=0, teste ack adresse, envoyer une donner, teste ack, afficher les caractères, envoyer stop.

Comme ça, il se répète ce système jusqu'a d'afficher l'alphabet et les caractères.

IV.6 organigramme de système :







IV.7 Conclusion:

Dans le dernier chapitre nous présentons les modèles et résultats de simulation en environnement à microcontrôleur PIC 16F877 pour la mise en œuvre de notre application.

Conclusion générale

L'objectif de notre projet utilisé « le protocole I2C dans environnement à microcontrôleur pic 16F877 » est d'étudier et développer un système d'affichage graphique à diode LED .Les diode LED sous forme matrice de ' ' 8x8' ' et sont aussi extensible pour matrice.

Dans l'objectif de simplifier et d'alléger la connectique entre matrices et CPU, le bus I2C est utilisé.

Ce bus a nécessité l'emploi de registre minus d'entrées séries compatible bus I2C et de sorties parallèles pour la commande des lignes et des colonnes des matrices à diode LED, nous avons utilisés les registres PCF8574 qui sont initialement conçu cet objectif.

En conclusion ce projet a été bénéfique pour notre formation il nous a permis de :

- Comprendre la philosophie des systèmes embarquées, leur architecture des systèmes électroniques standards en termes d'utilisation de processeur.
- comprendre également les différentes architectures de processeur, les performances, les limites d'utilisation pour chaque type d'applications souhaitée.
- Programmer modestement en langage C la famille des microcontrôleurs PIC d' MICROCHIP
- Connaitre la famille des microcontrôleurs MICROCHIP, connue par leur richesse en terme d'élément d' E / S et possibilités de programmabilité et souplesse de développement et de mise en œuvre pratique.

BIBLIOGRAPHIE

- [1] : Article: " Les systèmes embarqués " –
(http://www.enseirb-matmeca.fr/cosynux/grand_public/embedded_intro_2006) - 06/03/2013
- [2] : " le langage C pour l'embarqué" –
(<http://87.98.130.175/kadionik/systeme/langage-c-embarque>) - 07/03/2013
- [3] : thèse : Najat Demirel; Co-design d'un bloc PA-Antenne en technologie silicium pour application radar l'université bordeaux 1 École doctorale des Sciences Physiques et de l'Ingénieur; - (http://tel.archives-ouvertes.fr/docs/00/58/60/71/PDF/Demirel_These)
2009/2010, P42
- [4] : Article: " Architecture des DSP " - (<http://www.technologuepro.com/cours-dsp/chapitre-3-Architecture-des-DSP>) - 09/03/2013
- [5]: Architecture de Von Neumann et architecture de Harvard technique -
(<http://www.scripgroup.com/limba/franceza/159/Architecture-de-Von-Neuman-et-41539.php>)
- 12/03/2013
- [6]: "Architecture des ordinateurs" –
(<http://www.elektronique.fr/documents-pour-le-site/Cours-II.pdf>) - 15/03/2013
- [7] : " Le micro-ordinateur" –
(<http://www.math-info.univ-paris5.fr/~le-t/CoursHTML1/COURS/L'ORDI/L'ORDI%20annexes.pdf>) - 16/03/2013
- [8] : "Cours de processeur"
(<http://perso.ibcp.fr/gilbert.deleage//Cours/processeur.pdf>) - 16/03/2013
- [9] : "Qu'est-ce qu'un microcontrôleur ?" (http://www.mon-club-elec.fr/pmwiki_mon_club_elec/pmwiki.php?n=MAIN.COMPRENDREQuezakoMicrocontrol eur)- 17/03/2013
- [10] : Alibi El Mehdi/Jawadi Sami Rapport de projet de fin d'études Conception et réalisation d'un enregistreur de données -
(http://pfmh.uvt.rnu.tn/491/1/Conception_et_réalisation__d'un_enregistreur_de_données_.pd)
- 2010/2011 - 24/03/2013 - p43

- [11] : "Les microcontrôleurs PIC"-
(<http://www.technologuepro.com/microcontroleur/Chap1.htm>) - 26/03/2013
- [12] : "Le microcontrôleur PIC 16F877 /876 "-
(<http://www.oumnad.123.fr/Microcontroleurs/PIC16F877.pdf>) - 29/03/2013
- [13] : "Le bus I2C" -
(http://artic.ac.besancon.fr/genie_electronique/telech/sonde/doc_bus_i2c.pdf) - 02/04/2013
- [14] : "Le bus I2C" -
(<file:///C:/Users/ha/Desktop/i2c%20&Mise%20en%20oeuvre%20du%20bus%20I2C%20avec%20le%20PIC%2016F876A.htm>) - 03/04/2013
- [15] : "Alibi El Mehdi/Jawadi Sami Rapport de projet de fin d'études Conception et réalisation d'un enregistreur de données"-
(http://pf-mh.uvt.rnu.tn/491/1/Conception_et_réalisation_d'un_enregistreur_de_données_.pdf) - 2010/2011 - 14/04/2013 - p43
- [16] : "L'EDITEUR ISIS Version 6.5 SP5" - (www.gesi.asso.fr/web/ressources-ressources-dl.php?fichier=46) - 18/04/2013
- [17] : " SAISIE DE SCHEMA ISIS 6 ?" - (http://www.louis-armand-paris.fr/elec/claroline/document/goto/?url=%2FTP_N%B01%2F1_Cours_ISIS.doc) - 23/04/2013
- [18] : "Data sheet PCF8574 " - (http://www.electronika.fr/blog/wp-content/uploads/2009/10/doc_pcf8574.pdfhtml) - 27/04/2013
- [19] : "Matrix Led 8x8"- (<http://electronicexperiments.blogspot.com/2011/07/dot-matrix-led-8x8-max-7219.html>) - 05/05/2013

ANNEXE

Remote 8-bit I/O expander for I²C-bus BLOCK DIAGRAM

PCF8574

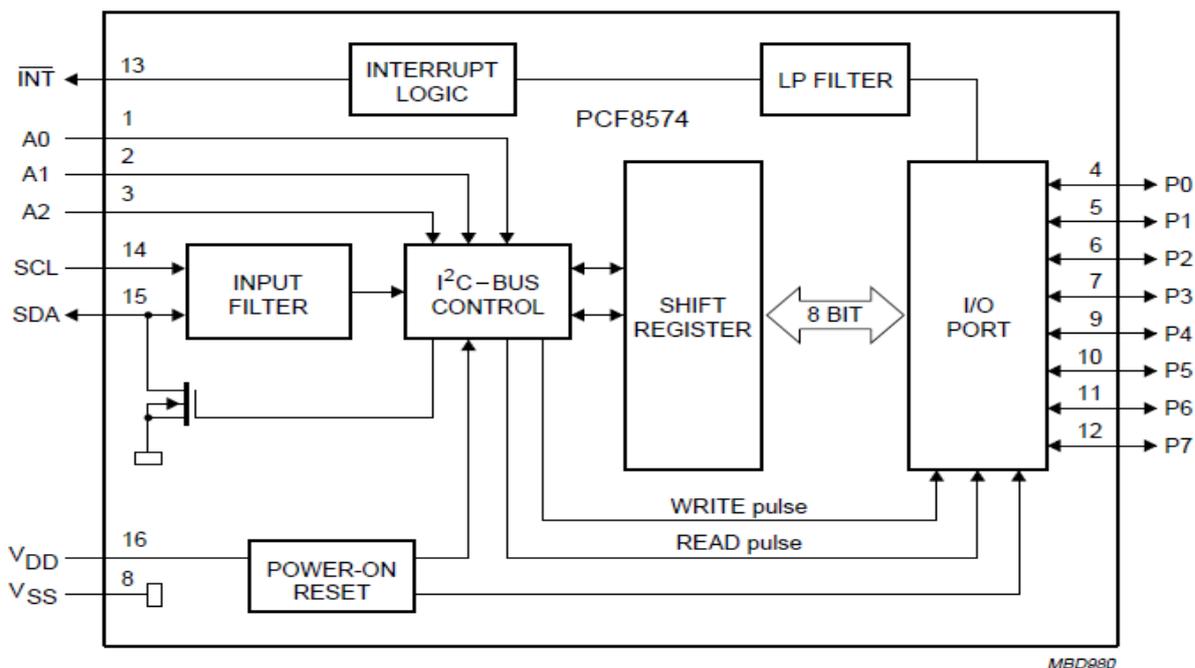


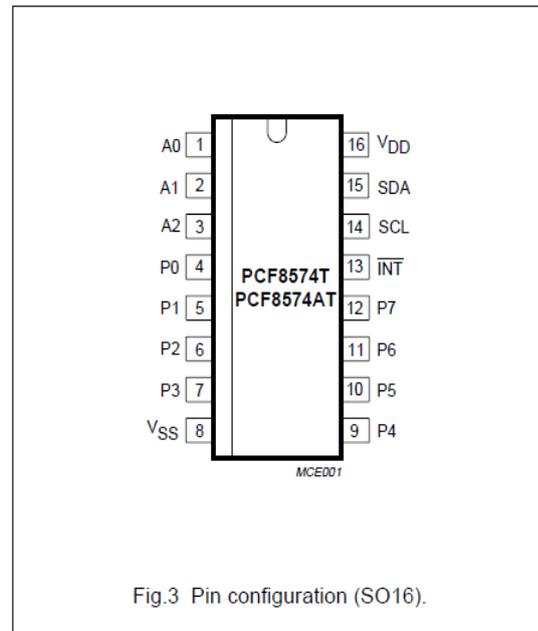
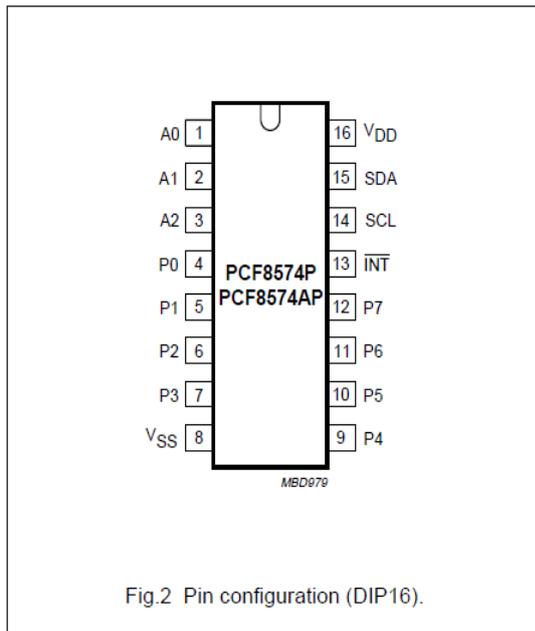
fig.1 : Block diagram (pin numbers apply to DIP16 and SO16 packages).

ADDRESS REFERENCE

INPUTS			I ² C-BUS SLAVE ADDRESS
A2	A1	A0	
L	L	L	32 (decimal), 20 (hexadecimal)
L	L	H	33 (decimal), 21 (hexadecimal)
L	H	L	34 (decimal), 22 (hexadecimal)
L	H	H	35 (decimal), 23 (hexadecimal)
H	L	L	36 (decimal), 24 (hexadecimal)
H	L	H	37 (decimal), 25 (hexadecimal)
H	H	L	38 (decimal), 26 (hexadecimal)
H	H	H	39 (decimal), 27 (hexadecimal)

Tableau.1 : adresse référence

SYMBOL	PIN	DESCRIPTION
A0	1	address input 0
A1	2	address input 1
A2	3	address input 2
P0	4	quasi-bidirectional I/O 0
P1	5	quasi-bidirectional I/O 1
P2	6	quasi-bidirectional I/O 2
P3	7	quasi-bidirectional I/O 3
V _{SS}	8	supply ground
P4	9	quasi-bidirectional I/O 4
P5	10	quasi-bidirectional I/O 5
P6	11	quasi-bidirectional I/O 6
P7	12	quasi-bidirectional I/O 7
$\overline{\text{INT}}$	13	interrupt output (active LOW)
SCL	14	serial clock line
SDA	15	serial data line
V _{DD}	16	supply voltage



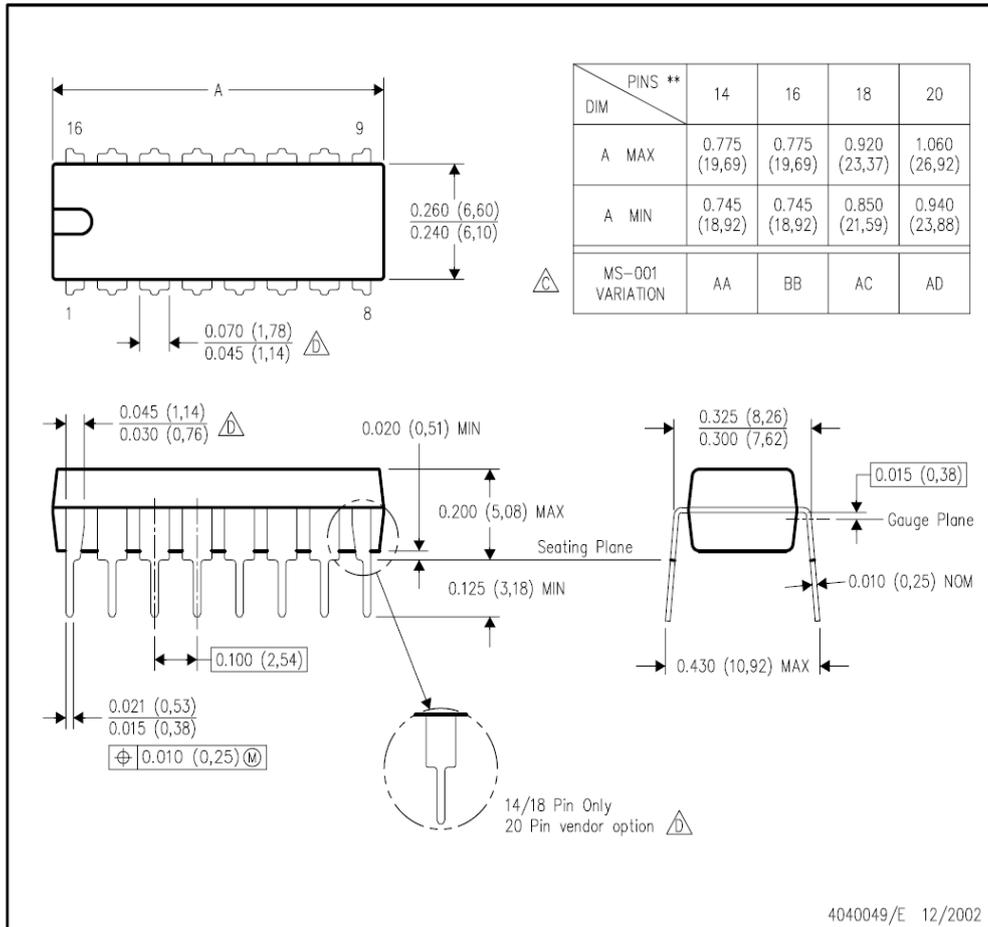


fig.4 : schéma interne du pcf8574

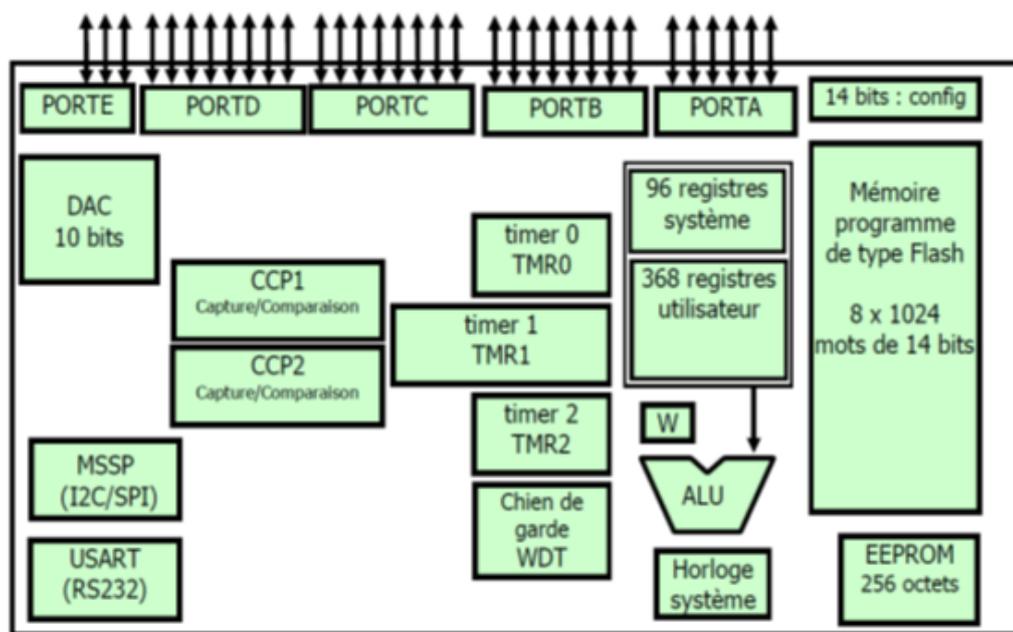


fig.5 : Les éléments constitutifs du PIC 16F877