

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Djilali Bounaama Khemis Miliana



Faculté des Sciences et de la Technologie
Département de Technologie

Mémoire du Projet de Fin d'Etudes
Pour l'obtention du diplôme de

Master

En

« Génie électrique »

Option :

«Electrotechnique»

Titre :

***Etude et réalisation d'une couveuse de
simulation à base du PIC 16F876A on
utilisant le bus I2C***

Réalisé par :

HAMLAOUI Abdelkader.

DJEBABERI Ali.

Encadré par :

MAHDAB Salim.

Année Universitaire : 2015/2016

Remerciement

Nous remercions vivement Mr. Mahdab pour sa disponibilité tout au long de ce travail, nous le remercions encore pour son aide si précieuse sur les plans conseils et orientations.

Nous tenons tout particulièrement à remercier les enseignants du département de génie électrique sans oublier l'ingénieur Rédhouane et AEK Youcefi pour leurs disponibilité et encouragement.

Nos reconnaissances vont aussi à messieurs les membres de jury, pour l'honneur qu'ils auront fait en acceptant de juger ce travail.



Dédicace

Louange à dieu le tout puissant, prière et salut sur le prophète Mohamed que le salut sur lui.

Je remercie dieu et grâce à lui que je suis arrivée à ce niveau.

Je dédie ce modeste travail à :

Mon père et ma mère qui m'encouragent beaucoup.

Mes frères et sœurs.

Tous les membres de ma grande famille.

MON ami ABDELKADER.

A Tous mes amis de la 2^{ème} année ET.

Toute personne ayant participé de loin ou de près pour la réalisation de ce travail.

ACI

Dédicace

*Je dédie ce modeste travail et ma profonde
gratitude à mes chers parents pour leurs éducation
qu'il m'ont prodigué , avec tous les moyens et au prix
de toutes les sacrifices qu'ils ont consentis à mon
égard , pour le sens de devoir qu'ils m'ont enseigné
depuis mon enfance ,*

A mes très chers frères.

A tous mes amis

Abdelkader

Résumé :

les systèmes de régulation de température sont divers et varient selon les appareils électronique et dépendent surtout de leurs domaines d'application.

Dans notre projet , nous avons utilisé un microcontrôleur Pic 16f876A , un bus de communication I2c . Pour mesurer la température un capteur DS1621 sera mis en œuvre et les résultat seront afficher par la suite dans un écran LCD.

Tous ces éléments cité constitue la carte électronique d'une couveuse d'un prématuré.

Abstract:

Systems of temprature regulation are differents and vary according to electronique cartes and their domain of application.

In our project ,we have used a microcontroler Pic 16f876A , a bus of communication I2c , a capter DS1621. In order to show results we have also used an LCD screen.

All these elements constitute an electrique board of an incubator.

ملخص:

تختلف أنظمة ضبط الحرارة باختلاف طبيعة البطاقات الالكترونية و ميادين استعمالها.

استعملنا في مشروعنا هذا المتحكم بيك16ف876أ , الناقل ايس , اللقط دي اس1621 بالاضافة الى شاشة ال سي دي

كل هذه العناصر المذكورة تشكل البطاقة الأم للحاضنة الكهربائية.

Table de Matière

Remerciement

Résumé

Introduction générale..... 1

Chapitre I :Le microcontrôleur16f876A

I.1. Introduction:..... 2

I.2.Définition de microcontrôleur:..... 2

I.3.Les avantages du microcontrôleur: 2

I.4.Le choix d'un microcontrôleur: 3

I.5.Les deux architecture des microcontrôleurs: 4

I.5.1.L'architecture de VON NEUMANN: 4

I.5.2.L'architecture de HARVARD:..... 5

I.5.3.Avantages et inconvénients des deux architectures: 5

I.6.Définition de Pic : 6

I.6.1.Structure minimale d'un PIC: 6

I.6.2.Les différentes familles des PICs : 6

I.6.3.Identification du PIC : 7

I.7.Le microcontrôleur PIC 16F876. 8

I.7.1Structure interne: 8

I.7.2.Unité Arithmétique et logique ALU: 10

I.7.3.Registre d'état : 10

I.7.4.Identification du PIC 16F876A: 11

I.7.5.Les caractéristiques du PIC 16F876A..... 11

I.7.6.Brochage..... 11

I.7.6.1.MCLR..... 12

I.7.6.2. Alimentation : VDD et VSS:	13
I.7.6.3. L'Interruption RBO/INT:	13
I.7.6.4. Etude des ports d'entrée sortie:	14
I.7.6.4.1. Le port A:	14
I.7.6.4.2. Le port B:	15
I.7.6.4.3. Le port C:	16
I.7.7. Les registres spéciaux:	17
I.7.7.1. Les registres TRISA, TRISB, TRISC.	17
I.7.7.1.1. Le port A et le registre de direction TRISA :	17
I.7.7.1.2. Le port B et le registre de direction TRISB :	17
I.7.7.1.3. Le port C et le registre de direction TRISC :	18
I.7.7.2. Le registre OPTION_REG:	18
I.7.7.3. Le registre INTCON:	19
I.7.7.4. Le registre TMR0:	19
I.7.8. Les Interruptions :	19
I.8. Les afficheurs LCD:	21
I.8.1. Brochage d'un LCD:	21
I.8.2. La mémoire d'affichage (DD RAM):	22
I.8.3. La mémoire du générateur de caractères (CG RAM):	23
I.8.4. Schéma fonctionnel d'un LCD:	23
I.8.5. Commande d'un afficheur LCD :	24
I.8.5.1. Mode 8 bits :	24
I.8.5.2. Mode 4 bits :	24
I.9. Conclusion :	25

Chapitre II : Le bus I2C et capteur DS1621.

II.1. Introduction :	26
II.2. Présentation :	26
II.3. caractéristiques du bus I2C :	27
II.4. Codage de l'information :	28
II.5. Le protocole I2C :	29
II.6. L'échange entre le maître et l'esclave :	30

II.6.2.La condition « Pause » :	32
II.6.3.La condition « Stop » :	32
II.7.La prise contrôle du bus :	33
II.8.La transmission d'un octet :	33
II.8.1.La transmission d'une adresse :	34
II.9.Le Capteur de Température DS 1621:	35
II.9.1. Définition:	35
II.9.2. Application:	36
II.9.3. Adressage :	36
II.9.4. Description des pins:	37
II.9.5. Mot de contrôle:	37
II.9.6. Commande:	37
II.9.7.DS1621 bloc-diagramme fonctionnel :	38
II.10.Commande du thermostat:	38
II.10.2.DQ (sortie Thermostat, actif = élevé) :	39
II.11.Fonctionnement et contrôle :	39
II.11.1.Le registre de configuration est défini comme suit :	39
II.12.Le DS1621 peut fonctionner dans les deux modes suivants :	41
II.13.Ensemble de commandes :	41
II.13.1.Lire la température [AAh]:	41
II.13.2.Accéder à TH [A1h]:	42
II.13.3.Accéder aux TL [A2h]:	42
II.13.4.Accéder à Config [ACh]:	42
II.13.5.Lire compteur [A8h]:	42
II.13.6.Lire la pente [A9h]:	43
II.13.7.Commencer à convertir T [EEh]:	43
II.13.8.Arrêter convertir T [22h]:	43
II.14.Avantage et inconvénient de ce thermostat :	43
II. 15.Conclusion :	44

Chapitre III : Etude technique

III.1.Introduction :	45
III.2. schéma synoptique :	45

III.3. simulation du scéma globale :	46
III.4.Caractéristiques techniques :	47
III.4.1.Condition ambiantes :	47
III.5. Explication de schéma synoptique :	47
III.5.1.Bloc d'Alimentation :	47
III.5.2. Bloc de traitement :	48
III.5.3: bloc d'entrée :	48
III.5.3.1: Les boutons:	48
III.5.3.2. Le capteur DS1621:	49
III.5.3.3. Bus i2c :	49
III.6. Bloc de sortie :	51
III.6.1. Sortie commande de l'afficheur :	51
III.6.2. Sortie relais (chauffage + ventilation) :	51
III.7. Conclusion :	53

Chapitre IV : Programmation en miKroC.

IV. 1.Introduction :	54
IV. 2. Avantages du langage C :	54
IV. 3. Inconvénients du langage C :	55
IV. 4. langage et compilateur mikroC pour les microcontrôleurs PIC :	55
IV. 4.1. Compilateur mikroC PRO pour PIC :	56
IV. 5. Introduction au langage de programmation mikroC :	57
IV. 5.1. Structure d'un programme en miKroC :	57
IV. 5.2. Règles générale d'écriture en micro C :	57
IV. 5.3. les commentaires:	57
IV. 5.4. Début et fin d'un programme :	58
IV. 6. Les bibliothèques du MikroC :	59
IV. 6.1. Fonction d'écriture :	60
IV. 6.2. Fonction de lecture :	61
IV. 7. Affichage dans LCD :	61
IV. 8. Porteuse ISIS professional:	63
IV. 9. programme générale de la carte:	65
IV. 10.Conclusion:	68

Chapitre V : Réalisation du circuit.

V.1.Introduction:	69
V.2.Circuit imprimé:	69
V.3.Principe:.....	70
V.4.Tracée du circuits:	70
V.5.Utilisation d'un typon:.....	71
V.6.La conception sur ordinateur :	71
V.6.1.Présentation du logiciel TCI:	72
V.6.2.les deux grandes parties du TCI:	72
V.6.3.Créer un nouveau schéma électronique en TCI:.....	73
V.6.4.Etapes de tracé schéma avec logiciel TCI :	73
V.6.5.Le tracé de notre circuit en TCI:.....	75
V.6.6.Transformation du TCI vers le typon:	75
V.6.7.Typon:.....	76
V.7.Transfert du circuit sur la plaque du cuivre:.....	77
V.7.1.Coupe:.....	77
V.7.2.Préparation:.....	78
V.7.3.Insolation:	78
V.7.4.Révélation:	79
V.7.5.Rinçage:	80
V.7.6.Gravure:	80
V.8.Implantation des composants et soudure:.....	82
V.8.1.Perçage:.....	82
V.9.Conclusion:.....	84
Conclusion générale.....	85

Bibliographie

Annexe

Liste des figures

Fig. I.1: Architecture de Von Newman.....	5
Fig. I.2: Architecture de Harvard.....	5
Fig. I.3: Structure minimale du PIC.....	7
Fig. I.4: Architecture interne du PIC 16F876.....	10
Fig. I.5: Structure de l' ALU.....	11
Fig. I.6: Registre d'état.....	11
Fig. I.7: Brochage du PIC16F876a.....	13
Fig. I.8: Les trois ports du pic 16f876a.....	15
Fig. I.9: Afficheur LCD.....	23
Fig. II.1 : la topologie esclave.....	27
Fig.II. 2 : schéma d'une alimentation extérieure.....	28
Fig. II.3: Codage de l'information.....	29
Fig. II.4: les conditions Start et Stop.....	30
Fig. II.5: L'échange entre le maitre et l'esclave.....	31
Fig. II.6: « acquittement et non acquittement ».....	31
Fig. II.7: La condition « Pause ».....	32
Fig. II.8: La condition « stop ».....	32
Fig. II.9 : Exemple de condition de départ et d'arrêt.....	33
Fig. II.10: La transmission d'un octet.....	34
Fig. II.11 : Exemple d'octet d'adresse.....	35
Fig. II.12 : Brochage du capteur DS 1621.....	36
Fig. II.13 : Adressage du DS1621.....	36
Fig. II.14: Mot de contrôle.....	37
Fig. II.15: Brochage du capteur DS 1621.....	38
Fig. III.16 : Sortie thermostat.....	39
Fig. III.17 : Le registre de configuration.....	39
Fig. III.1: schéma synoptique.....	46
Fig. III.2: simulation du schéma globale.....	47
Fig. III.3 : schéma de bloc d'alimentation.....	48
Fig. III.4 : schéma de bloc de traitement.....	48

Fig. III .5: schéma de bloc d'entrée (clavier).....	49
Fig. III .6: schéma de capteur DS1621.....	50
Fig. III .7: Schéma de sortie afficheur.....	51
Fig. III. 8 : Schéma de bloc sorti (relais).....	52
Fig. IV.1: Interface du logiciel MikroC.....	56
Fig. IV.2: Fenêtre de saisie de programme.....	59
Fig. IV.3: Fonction d'écriture.....	60
Fig. IV.4: Fonction de lecture.....	61
Fig. IV.5: interface de logiciel Isis porteuse.....	64
Fig. IV.6: Interface de sélection de composants de isis porteuse.....	63
Fig. V.1:exemple d'un circuit imprimé.....	70
Fig. V.2:exemple d'un typon.	71
Fig. V.3:interface du logiciel TCI.	72
Fig. V.4: les deux partie du TCI.....	72
Fig. V.5:la création d'un nouveau schéma.....	73
Fig. V.6: schéma en TCI de notre PCB.....	75
Fig. V.7:Transformation vers le typon.	75
Fig. V.8:Typon.	76
Fig. V.9:cisaille coupante.....	77
Fig. V.10:Placement des différents éléments pour l'insolation.....	78
Fig. V.11:Le tracé des pistes sur la résine.....	79
Fig. V.12:révélation.....	79
Fig. V.13:gravure.	80
Fig. V.14:Circuit imprimé.....	81
Fig. V.15:perceuse.....	82
Fig. V.18:Circuit imprimé vu du côté soudures.....	83
Fig. V.19:Circuit imprimé vu du côté composants.....	83

Liste des tableaux

Tableau. I.1: Avantages et inconvénients des architectures	6
Tableau I.2 :Tableau comparatif.	8
Tableau. I.3: Rapport de la pré division.	20
Tableau. I.4: Exemple d'adressage d'un LCD 2*16.	24
Tableau. IV.1 : Bibliothèque du MikroC.	60

Liste des abréviations

E/S: Entrée sortie.

UAL: Unité Arithmétique et Logique.

CISC : Complex Instruction Set Computer.

CPU : Unité centrale de traitement.

RISC : Reduced Instruction Set Computer.

PIC : Periphériol Interface.

RAM : Random Access Memory.

ROM : Read Only Memory.

EEPROM : Electrical Erasable Programing Only Memory.

MHz : Méga Hertz.

Kmots :Kilo Mots.

CAN : Convertisseur Analogique Numérique.

DIL : Dual In Line.

MCLR: master clear. Broche de réinitialisation.

WDT : Watch Dog Timer.

POR : Power On Reset.

VDD : Tension d'alimentation.

VSS : Neutre.

VPP :Tension de programmation.

INT: Interruptions.

LCD : Cristaux Liquide Display.

DS : Dallas.

I2C : (IIC) Inter Integrated Circuit.

SDA : Signal de donnée

SCL : Signal d'horloge

ACK : Acknowledge (reconnaître).

Introduction générale

Introduction générale :

Actuellement, l'électronique est au centre de toutes les réalisations et au cœur de plusieurs développements et perfectionnements. Les dispositifs électroniques sont aussi bien retrouvés dans le domaine médical, industriel mais aussi au niveau des systèmes de contrôle et de régulation automatique.

La protection thermique du nouveau-né comporte une série de mesures prises à la naissance et pendant les premiers jours de la vie afin d'assurer que la température corporelle de l'enfant ne s'abaisse pas trop ou ne s'élève pas trop, mais se maintient dans la normale entre 36.5 et 37.5 °C. la couveuse sert donc à maintenir cette température.

Deux objectifs sont assignés à ce travail. Le premier objectif c'est d'affûter nos connaissances théoriques et pratiques de par les circuits qui vont être manipulés au cours de ce travail. Le deuxième objectif, c'est de la familiarisation avec les constituants d'une couveuse ainsi que son principe de fonctionnement.

Notre travail est basé sur l'exploitation d'un capteur de température DS1621, un microcontrôleur, un bloc d'alimentation, un bloc de chauffage et une enceinte en verre.

Le travail décrit dans ce présent document est structuré en une introduction, cinq chapitres, une conclusion ainsi que des références bibliographiques.

Le premier chapitre comportera principalement la présentation du pic 16f876A.

Le second chapitre sera consacré à l'étude théorique du dispositif DS1621 et le bus I2C, portant sur son principe de fonctionnement.

Le troisième chapitre il est axé sur le rôle des différents blocs le constituant et leur intérêt.

La partie software avec les organigrammes et les procédures sont abordés dans **le quatrième chapitre**.

Le cinquième et le dernier **chapitre** est consacré à la phase de la réalisation pratique, ainsi le logiciel utilisé dans cette conception.

Finalement le travail sera fini par une conclusion générale.

Chapitre I:

Le microcontrôleur

PIC 16f876A

I.1. 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 des 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 [2].

I.2. Définition de microcontrôleur:

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 [2].

I.3. 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 [2].

- 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.
- 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.4.Le choix d'un microcontrôleur:

La majorité des grands fabricants de circuits intégrés disposent 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.

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 [10].

Son architecture interne 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.5. Les deux architecture des microcontrôleurs:

Les microcontrôleurs ont été conçus sur une architecture dite HARVARD (RISC) et non sur un modèle VON NEUMANN (CISC) [10].

I.5.1. L'architecture de VON NEUMANN:

Employée par la plupart des microcontrôleurs actuels (INTEL80XX, MOTOROLA HC05, HC08 et HC12, ou ZILOG Z80) est basée sur un bus de données unique.

Celui-ci véhicule les instructions (du programme) et les données.

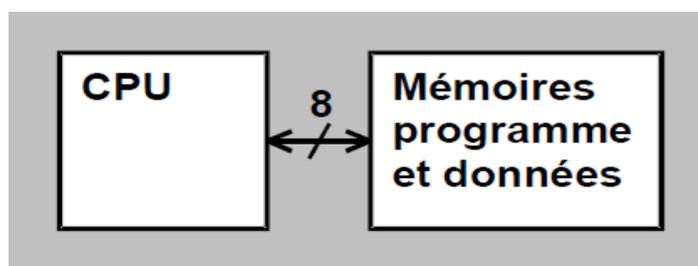


Fig. I.1: Architecture de Von Newman.

I.5.2.L'architecture de HARVARD:

Utilisée par les microcontrôleurs PIC est basée sur deux bus de données.
 Un bus est utilisé pour les données et un autre pour les instructions (du programme).

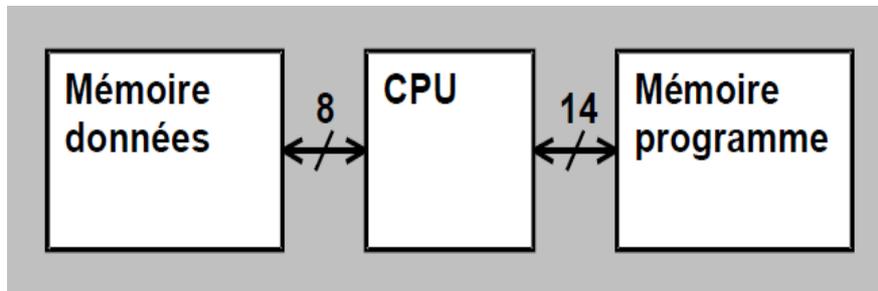


Fig. I.2: Architecture de Harvard.

I.5.3.Avantages et inconvénients des deux architectures:

	Architecture VON NEUMANN (MOTOROLA, INTEL, ZILOG, ..)	Architecture HARVARD (RISC) (MICROCHIP PIC)
Avantages	<ul style="list-style-type: none"> ▪ Jeu d'instructions riches. ▪ Accès à la mémoire facile. 	<ul style="list-style-type: none"> ▪ Jeu d'instructions pauvre, mais facile à mémoriser. ▪ Le codage des instructions est facile, chaque instruction est codée sur un mot et dure un cycle machine. ▪ Le code est plus compact.
Inconvénients	<ul style="list-style-type: none"> ▪ Le temps pour exécuter une instruction est variable. ▪ Le codage des instructions se fait sur plusieurs octets. 	<ul style="list-style-type: none"> ▪ Le jeu d'instructions est très pauvre, par exemple pour effectuer une comparaison il faut faire une soustraction. ▪ Les accès aux registres internes et la mémoire sont très délicats.

Tableau. I.1: Avantages et inconvénients des architecture.

Remarque :

Le format des instructions du programme spécifié pour un PICs s'exprime en **kilo Mots (14bits** pour la famille **16F87X)** et non en **kilo octets**. Comme chaque instruction est codée par un mot de **14 bits**, comparées aux microcontrôleurs classiques (**1, 2 ou 3** octets par instruction), les **PICs** ont des codes instruction plus compacts et utilisent moins de mémoire.

I.6.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 (Periphriol Interface). Ce composant encore très utilisé à l'heure actuelle, est un compromis entre simplicité d'emploi, rapidité et prix de revient [11].

I.6.1.Structure minimale d'un PIC:

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) [11].

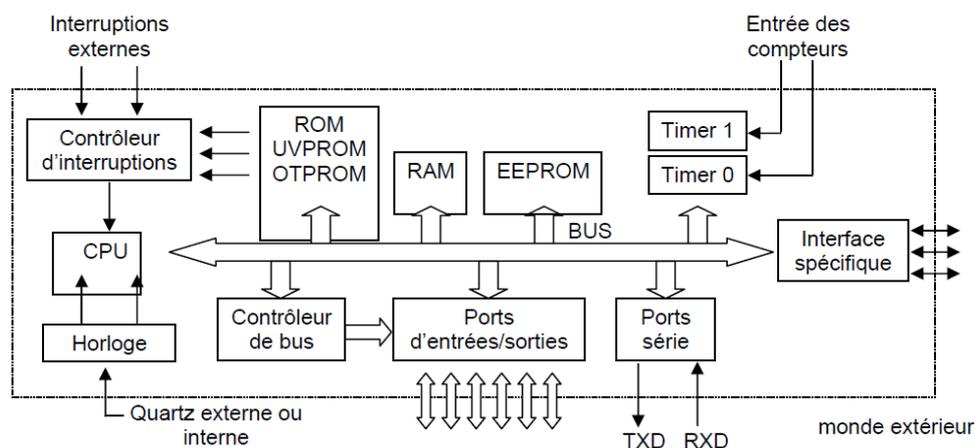


Fig. I.3: Structure minimale du PIC.

I.6.2.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 [11].

I.6.3. Identification du PIC :

La dénomination PIC est sous copyright de Micro Chip, les autres fabricants sont dans l'impossibilité d'utiliser ce terme.

Pour identifier une PIC, on utilise simplement son numéro. Les 2 premiers chiffres indiquent la catégorie du PIC, 16 indique une PIC MidRange . Vient ensuite parfois une lettre L: Celle-ci indique que le 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.
- ◆ 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 disfonctionnement.

Ceci par opposition aux composants DYNAMIQUE, donc la fréquence d'horloge doit rester dans des limites précises [11].

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

Tableau I.2.Tableau comparatif.

I.7. Le microcontrôleur PIC 16F876.

La PIC 16F876A est un circuit intégré de type CMOS. Le choix d'un PIC est lié directement à l'application envisagée, et aux différents avantages qu'il procure :

- Peu coûteux, il permet de réaliser des applications avec très peu de composants.
- Disponibilité des différents outils de développement.

Notre application utilise un **PIC 16F876-20**, soit :

- 16 : Famille Mid-Range.
- F : Mémoire FLASH.
- Type 876.
- 20 : Quartz à 20MHz au maximum.
- Un jeu d'instruction réduit à 35 instructions.
- Une mémoire vive de 368 octets.
- Une mémoire EEPROM pour sauvegarder des paramètres de 256 octets.
- Une mémoire morte de type FLASH de 8 Kmot (1mot = 14 bits), réinscriptible à volonté.
- 22 Entrées/sorties réparties sur 4 ports.
- 3 Temporisateurs : TIMER0 (8 bits avec pré diviseur), TIMER1 (16 bits avec pré diviseur avec possibilité d'utiliser une horloge externe réseau RC ou QUARTZ) et TIMER2 (8bits avec pré diviseur et post diviseur).
- Une tension d'alimentation entre 2 et 5,5 V.

I.7.1 Structure interne:

Le microcontrôleur est un dérivé du microprocesseur . Sa structure est celle des systèmes à base de microprocesseurs . Il est donc composé d'une unité centrale de traitement et de commande(équivalent au microprocesseur),et de ports d'entrées /sorties.

Les microcontrôleurs ont en supplément des systèmes de comptage (TIMERS). Quelques uns sont dotés d'un convertisseur analogique numérique (CAN)intégré [1].

I.7.2. Unité Arithmétique et logique ALU:

Comme son nom l'indique, l'unité arithmétique et logique (ALU) effectue au sein du microcontrôleur PIC toutes les opérations arithmétiques et logiques dans un format de données 8 bits (addition, soustraction, comparaison logique, ET logique, etc...).

Ces opérations sont effectuées entre le registre de travail W et un opérande provenant d'un registre de la RAM.

Les trois bits de poids faible du registre STATUS sont affectés par le résultat de l'opération effectuée par l' ALU [1].

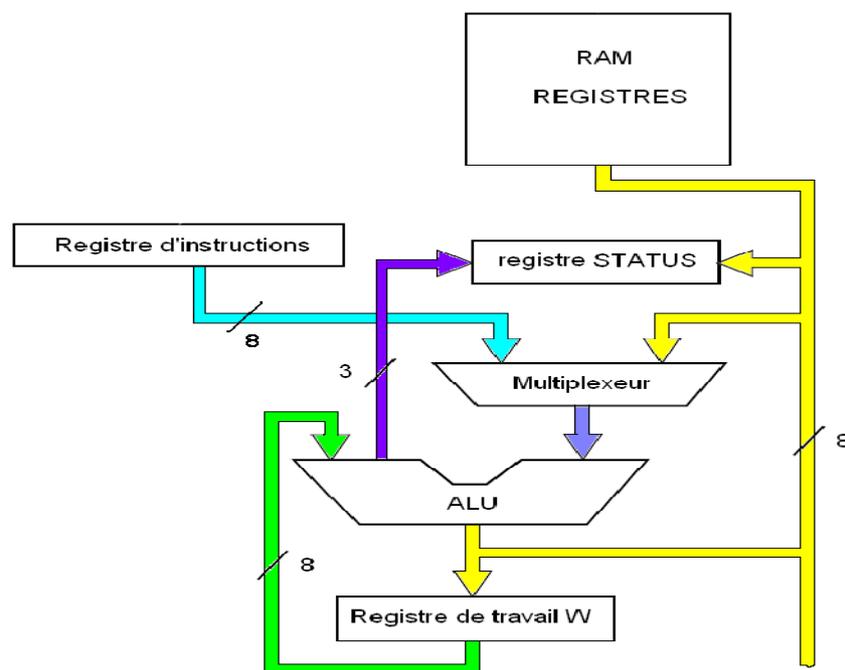


Fig. I.5: Structure de l' ALU.

I.7.3. Registre d'état :

Ce registre est constitué de 8 bits et chaque bit nous renseigne sur l'état du résultat obtenu par l'UAL.

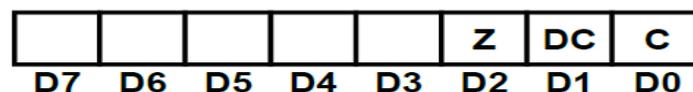


Fig. I.6: Registre d'état.

Les lettres Z,DC,C signifie ce qui suis:

- Z (Zéro): ce bit est mis à l'état logique "1" si le résultat de l'opération effectuée est nul.
- DC (Digit Carry): ce bit de demi-retenu est mis à l'état logique "1" si l'opération effectuée entraîne la mise à l'état logique "1" du bit n°4 du résultat.
- C (Carry): ce bit de retenue est mis à l'état logique "1" si l'opération effectuée entraîne un résultat dont le format est supérieur à 8 bits.

I.7.4. Identification du PIC 16F876A:

Il est de la famille MID-RANGE et la mémoire programme est de type FLASH (F). Le 16 indique que la PIC utilise des mots de 14 bits pour coder une instruction.

Il est capable de fonctionner à des fréquences d'horloge allant de 0 a 20MHz.

I.7.5. Les caractéristiques du PIC 16F876A.

- Une mémoire morte de type FLASH de 8K mots (1mot = 14 bits), elle est réinscriptible à volonté.
- Une Fréquence de fonctionnement qui va de 0 à 20 MHz.
- 3 Temporisateurs : TIMER0 (8 bits avec pré diviseur), TIMER1 (16 bits avec pré diviseur avec possibilité d'utiliser une horloge externe réseau RC ou QUARTZ et TIMER2 (8 bits avec pré diviseur et post diviseur).
- 13 sources d'interruption.
- 3 ports de communication.

I.7.6. Brochage.

Le 16F876A est un circuit intégré de 28 broches, que l'on peut trouver dans un boîtier DIL (Dual In Line) de 2x14pattes comme indiqué à la figure ci-dessous. A chacune de ses broches, il est associé une ou plusieurs fonctions. Chaque broche peut donc jouer plusieurs rôles selon les configurations effectuées lors de la programmation du PIC [12].

Le boîtier du PIC 16F876 décrit par la Figure I-7 comprend donc 28 pins : 22 pins d'entrées/sorties, 1 pin pour l'alimentation, 2 pins pour l'oscillateur et une pin pour le reset (MCLR).

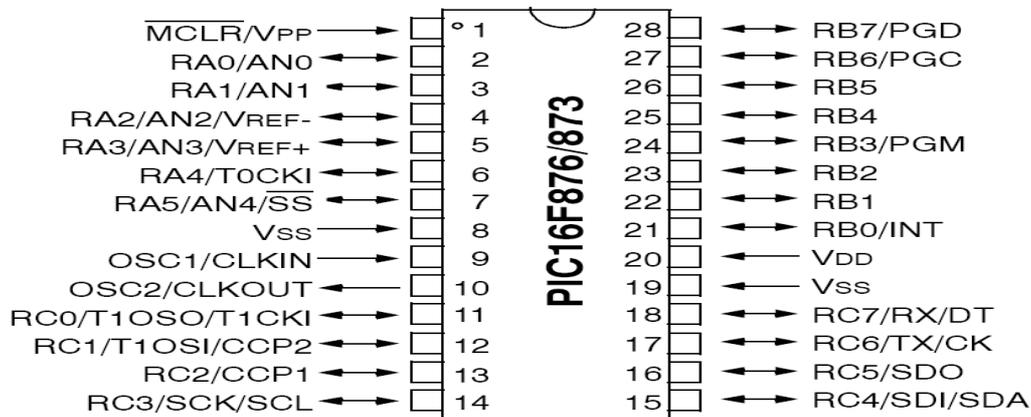


Fig. I.7: Brochage du PIC16F876a.

I.7.6.1.MCLR.

MCLR : master clear. Broche de réinitialisation.

Cette broche sert à initialiser le μ C.

Le μ C 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 μ C. Le temps nécessaire est au minimum de 72mS et au maximum de $72mS + 1024 * T_{osc}$. Le μ C dispose 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 [12].

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 μ 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 μ C.

_ BOR: Baisse de l'alimentation.

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

I.7.6.2. 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 μ A à 10mA.

La consommation du μ C 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 μ F électrolytique.

- 10nF céramique.

I.7.6.3. L'Interruption RBO/INT:

Cette broche a 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_Ω pour ne pas déclencher d'interruptions imprévues, cela permet aussi de relier plusieurs sources d'interruptions sur une même ligne (OU CABLE) [12].

VPP : Tension de programmation ≈ 13V.

I.7.6.4. Etude des ports d'entrée sortie:

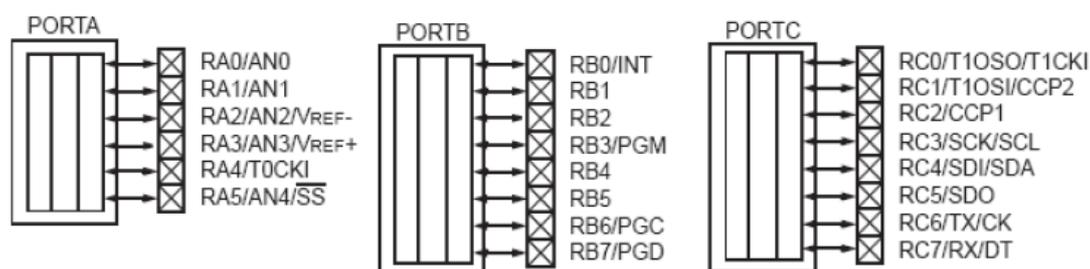


Fig. I.8: Les trois ports du pic 16f876a.

Le 16F876A possède jusqu'à 22 entrées/sorties :

- 6 sur le port A (RA0 à RA5) ;
- 8 sur le port B (RB0 à RB7) ;
- 8 sur le port C (RC0 à RC7) ;

Notez qu'il y a deux broches de masse (broches 8 et 19).

Certaines broches sont multiplexées à d'autres fonctions comme indiqué ci-dessous :

I.7.6.4.1. Le port A.

Les abréviations et configurations relatives au port A sont données ci-dessous.

- **RA0/AN0** :
- RA0 : Entrée Sortie numérique.
- AN0 : Entrée analogique.
- **RA1/AN1** :
- RA1 : Entrée Sortie numérique.
- AN1 : Entrée analogique.

- RA2/AN2 :

- RA2 : Entrée Sortie numérique.

- AN2 : Entrée analogique.

- RA3/AN3/VREF :

- RA3 : Entrée Sortie numérique.

- AN3 : Entrée analogique.

- VREF : Tension de référence.

- RA4/T0CKI :

- RA4 : Entrée Sortie numérique.

- T0CKI : Entrée d'horloge du TMR0.

- RA5/S \bar{S} /AN4 :

- RA5 : Entrée Sortie numérique.

- S \bar{S} : Entrée de sélection esclave pour le port série synchrone.

- AN4 : Entrée analogique.

I.7.6.4.2. Le port B:

Les abréviations et configurations relatives au port B sont données ci-dessous.

- RB0/INT :

- RB0 : Entrée Sortie numérique.

- INT : Broche d'interruption externe.

- RB1 :

- RB1 : Entrée Sortie numérique.

- RB2 :

- RB2 : Entrée Sortie numérique.

- RB3/PGM :

- RB3 : Entrée Sortie numérique.

- PGM : Entrée de la tension de programmation basse tension.

- RB4 :

- RB4 : Entrée Sortie numérique.

- RB5 :

- RB5 : Entrée Sortie numérique.

- RB6/PGC :

- RB6 : Entrée Sortie numérique.
- PGC : Entrée d'horloge en mode programmation.

- RB7/PGD :

- RB7 : Entrée Sortie numérique.
- PGD : Entrée de donnée en mode programmation.

I.7.6.4.3. Le port C:

Les abréviations et configurations relatives au port C sont données ci-dessous.

- RC0/T1OS0/T1CKI :

- RC0 : Entrée Sortie numérique.
- T1OSO : Sortie d'oscillateur du TMR1.
- T1CKI : Entrée d'horloge du TMR1.

- RC1/T1OSI/CCP2 :

- RC0 : Entrée Sortie numérique.
- T1OSI : Entrée de l'oscillateur du TMR1.
- CCP2 : Entrée/Sortie du module CCP2.

- RC2/CCP1 :

- RC2 : Entrée Sortie numérique.
- CCP1 : Entrée/Sortie du module CCP1.

- RC3/SCK/SCL :

- RC3 : Entrée Sortie numérique.
- SCK : Entrée d'horloge en mode SPI.
- SCL : Entrée d'horloge en mode I2C.

- RC4/SDI/SDA :

- RC4 : Entrée Sortie numérique.
- SDI : Entrée de données en mode SPI.
- SDA : Entrée/Sortie de données en mode I2C.

- RC5/SDO :

- RC5 : Entrée Sortie numérique.
- SDO : Sortie de données en mode SSP.

- RC6/TX/CK :

- RC6 : Entrée Sortie numérique.
- TX : Broche de transmission en mode USART Asynchrone.
- CK : Entrée d'horloge en mode USART synchrone.

- RC7/RX/DT :

- RC7 : Entrée Sortie numérique.
- RX : Broche de réception en mode USART Asynchrone.
- DT : Entrée/Sortie en mode USART synchrone.

I.7.7. Les registres spéciaux:**I.7.7.1. Les registres TRISA, TRISB, TRISC.**

La plupart de ces lignes ont une double fonction suivant leur programmation. Le sens de direction des données (entrée ou sortie) est mémorisé dans des registres TRIS (TRAnsfert Input Set). TRISA est dédié au port A, TRISB au port B et TRISC au port C. Chaque ligne peut être configurés individuellement en entrée ou en sortie.

La mise à « 1 » d'un bit du registre TRIS configure la ligne correspondante en entrée, un « 0 » configure la ligne en sortie.

Au RESET, toutes les lignes sont configurées en entrées.

I.7.7.1.1. Le PORTA et le registre de direction TRISA:

Le port A est un port bidirectionnel à 6 fils. Il cumule les fonctions d'E/S numériques, d'entrées analogiques et entrée du Timer0. Au reset, ces lignes sont en mode analogique.

I.7.7.1.2. Le PORTB et le registre de direction TRISB:

Le port B est un port bidirectionnel à 8 fils. Les lignes RB3, RB6 et RB7 peuvent être utilisées en E/S ou pour la programmation in-situ. Toutes les lignes du port B peuvent être reliées à une résistance de tirage vers le haut (voir registre Option_Reg). Les lignes RB4 à RB7 peuvent déclencher une interruption.

I.7.7.1.3. Le PORTC et le registre de direction TRISC:

Le port C est un port bidirectionnel à 8 fils. Toutes les entrées du port C sont des entrées trigger de schmitt. Ce port peut aussi servir pour le Timer1, la réalisation de signaux PWM (Pulse Width Modulation : modulation à largeur d'impulsions), la communication avec des périphériques I2C et enfin à dialoguer par voie série [12].

I.7.7.2. Le registre OPTION_REG:

Il permet la configuration des paramètres du TMR0, du chien de garde et des résistances de tirage comme indiqué ci-dessous.

OPTION_REG.7 = RB \bar{U} : Permet d'activer les résistances de tirage (0).

OPTION_REG.6 = INTEDG : Permet de sélectionner le front, montant (1) ou descendant, du signal d'interruption de la broche RB0.

OPTION-REG.5. = T0CS : Permet de sélectionner la source d'horloge, interne (0) ou externe (1), du TMR0.

OPTION-REG.4. = T0SE : Permet de sélectionner le front d'horloge, montant (0) ou descendant (1), du TMR0.

OPTION-REG.3. = PSA : Permet d'appliquer le pré-diviseur soit sur le TMR0 (0), soit sur le chien de garde (1).

OPTION-REG.2 : OPTION-REG.0. = PS2 : PS0 : permettent de fixer le rapport de la pré-division selon le tableau ci-dessous.

Valeur binaire	Rapport de la pré-division	
	TMR0	Chien de garde
000	1/2	1/1
001	1/4	1/2
010	1/8	1/4
011	1/16	1/8
100	1/32	1/16
101	1/64	1/32
110	1/128	1/64
111	1/256	1/128

Tableau. I.3: Rapport de la pré division.

I.7.7.3. Le registre INTCON:

Ce registre permet d'effectuer les interruptions et de stocker les indicateurs d'interruptions comme indiqué ci-dessous.

INTCON.7 = GIE : Permet d'activer ou de désactiver (0) les interruptions de façon globale.

INTCON.6 = PEIE : Permet d'activer ou de désactiver (0) toutes les interruptions externes.

INTCON.5 = TOIE : Permet d'activer ou de désactiver (0) l'interruption liée au TMR0.

INTCON.4 = INTE : Permet d'activer ou de désactiver (0) l'interruption liée à la broche RB0.

INTCON.3 = RBIE : Permet d'activer ou de désactiver (0) l'interruption liée au changement d'état sur l'une des broches de RB7 à RB4.

INTCON.2 = TOIF : C'est l'indicateur (1) d'interruption du TR0.

INTCON.1 = INTF : C'est l'indicateur (1) d'interruption sur la broche RB0.

INTCON.0 = RBIF : C'est l'indicateur (1) d'interruption sur l'une des broches de RB7 à RB4.

I.7.7.4. Le registre TMR0:

C'est un compteur/temporisateur dont les caractéristiques sont les suivantes :

- Compteur sur 8 bits.
- Pré-diviseur 8 bits programmable.
- Choix de l'horloge : interne (mode temporisateur) ou externe (mode compteur).
- Interruption au débordement (passage de FF à 00).
- Choix du front de l'horloge en mode horloge externe.

Tous les bits de configuration du TMR0 sont dans le registre OPTION.

I.7.8. Les Interruptions :

Interruption par définition est un programme qui provoque une rupture d'un autre programme de sorte que ce programme puisse reprendre là où il était après exécution du premier.

Le fonctionnement par interruptions sur un microcontrôleur permet à celui-ci d'exécuter une tâche (programme principal) qui peut être interrompue par un événement. Le processeur doit alors exécuter une tâche (sous-programme) associée à cette source d'interruption. Quand la tâche est exécutée, le processeur revient à sa tâche principale.

Le traitement de ces évènements peut lui-même être interrompu par un événement jugé plus important.

De la même façon, avec un système ordonné, on doit souvent définir des priorités dans les interruptions. Ainsi le traitement d'une interruption peut être interrompu par une interruption qui lui est prioritaire.

Le PIC16F876A comporte 13 sources d'interruption dont entre autres :

1. Débordement du TMR0.
2. Transition sur RB0.
3. Changement d'état sur l'une des broches de RB7 à RB4.

les circuits d'interfaces peuvent piloter des matériels très différents, (moteur pas à pas, afficheur LCD, thermistance, communication avec des pc ou d'autres microcontrôleurs ,etc..) [5].

I.8. Les afficheurs LCD:

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts, intelligents qui nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA), sont relativement bon marché et s'utilisent avec beaucoup de facilité.

Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, non seulement par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétro éclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module [13].

Un exemple de module LCD est donné à la Figure I-9.

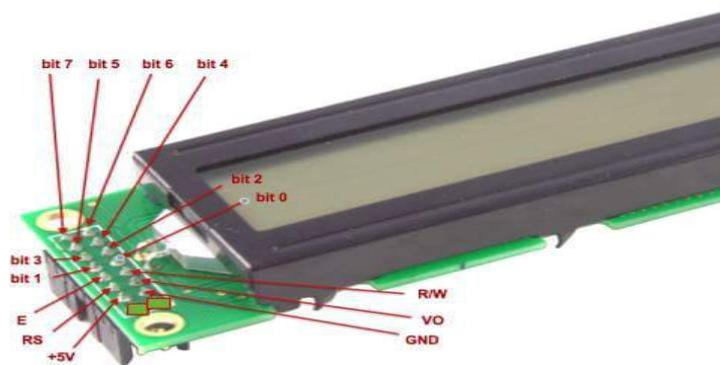


Fig. I.9: Afficheur LCD.

I.8.1. Brochage d'un LCD:

Un circuit intégré spécialisé est chargé de la gestion du module. Il remplit une double fonction: d'une part il commande l'affichage et de l'autre il se charge de la communication avec l'extérieur.

Avec l'écran à cristaux liquides proprement dit, on trouve généralement une série de broches aux rôles suivants :

- Deux broches pour l'alimentation du module
- Une broche pour le réglage du contraste de l'afficheur.

- Une broche (RS) : TTL pour la Sélection du registre (Register Select). Grâce à cette broche, l'afficheur est capable de faire la différence entre une commande et une donnée : Un niveau bas indique une commande et un niveau haut indique une donnée.
- Une broche (R/W) : TTL qui permet la sélection du mode lecture ou écriture
- Une broche E : TTL pour la commande des opérations d'écriture ou de lecture.
- Un ensemble de huit Broches 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 quatre broches sont utilisées.
- Deux broches supplémentaires sont disponibles sur les afficheurs LCD avec retro éclairage.

L'afficheur possède deux types de mémoire, la DD RAM et la CG RAM, respectivement mémoire d'affichage et mémoire du générateur de caractères [13].

I.8.2.La mémoire d'affichage (DD RAM):

La DD RAM est la mémoire où sont stockés les caractères actuellement affichés à l'écran. Pour un afficheur de 2 lignes de 16 caractères, les adresses sont définies de la façon suivante :

Ligne	Visible	Invisible
Haut	00H.....0FH	10H.....27H
Bas	40H.....4FH	50H.....67H

Tableau. I.4: Exemple d'adressage d'un LCD 2*16.

L'adresse 00H correspond à la ligne du haut à gauche, 0FH à droite. L'adresse 40H correspond à la ligne du bas à gauche, 4FH à droite. La zone invisible correspond à la mémoire de l'afficheur (48 caractères). Lorsqu'un caractère est inscrit à l'adresse 27H, le caractère suivant apparaît à la ligne suivante [13].

I.8.3. La mémoire du générateur de caractères (CG RAM):

Le générateur de caractère permet la création d'un maximum de 8 caractères ou symboles 5x7. Une fois les nouveaux caractères chargés en mémoire, il est possible d'y accéder comme s'il s'agissait de caractères classiques stockés en ROM.

La CG RAM utilise des mots de 8 bits, mais seuls les 5 bits de poids faible apparaissent sur le LCD. Ainsi D4 représente le point le plus à gauche et D0 le point le plus à droite. Par exemple, charger un octet de la CG RAM à 1Fh fait apparaître tous les points de cette rangée ; charger un octet à 00h éteint tous ces points. Les 8 lignes d'un caractère doivent être chargées dans la CG RAM.

La CG RAM peut être utilisée pour créer des caractères en vidéo inversée, des caractères avec des accents, etc. La limitation d'un total de 8 caractères peut être contournée en utilisant une bibliothèque de 8 symboles résidant dans le système hôte. Un maximum de 8 caractères peut être affichés à la fois. La CG RAM peut être rechargée périodiquement en fonction des besoins. Si un caractère actuellement sur l'afficheur est changé, alors le changement est immédiatement apparent sur l'afficheur [13].

I.8.4. Schéma fonctionnel d'un LCD:

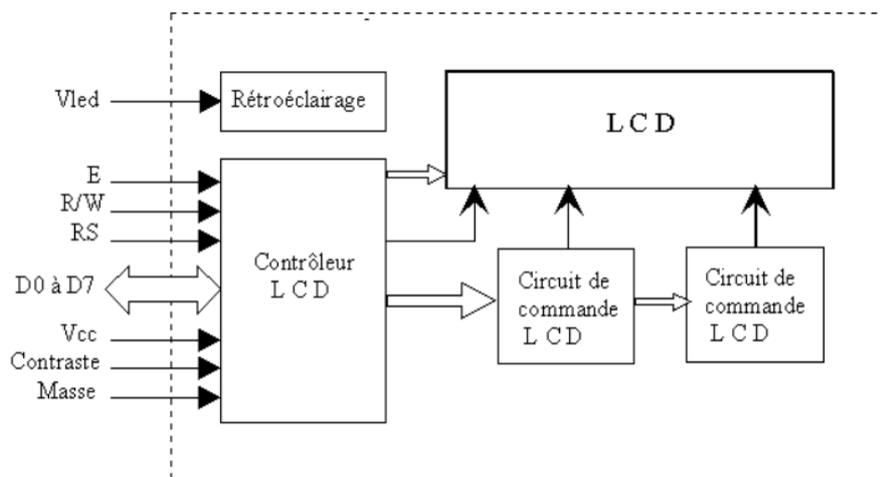


Fig. I.10: Schéma fonctionnel de l'LCD.

I.8.5. Commande d'un afficheur LCD :

Deux modes de fonctionnement de l'afficheur sont disponibles, le mode 4 bits et le mode 8 bits, modes que l'on choisira à l'initialisation de l'afficheur.

I.8.5.1. Mode 8 bits :

Dans ce mode 8 bits, les données sont envoyées à l'afficheur sur les broches D0 à D7. On place la ligne RS à 0 ou à 1 selon que l'on désire transmettre une commande ou une donnée. Il faut aussi placer la ligne R/W à 0 pour indiquer à l'afficheur que l'on désire effectuer une écriture. Il reste à envoyer une impulsion d'au moins 450 ns sur l'entrée E, pour indiquer que des données valides sont présentes sur les broches D0 à D7. L'afficheur lira la donnée sur le front descendant de cette entrée. Si on désire au contraire effectuer une lecture, la procédure est identique, mais on place cette fois la ligne R/W à 1 pour demander une lecture. Les données seront valide sur les lignes D0 à D7 lors de l'état haut de la ligne E [14].

I.8.5.2. Mode 4 bits :

Il peut, dans certains cas, être nécessaire de diminuer le nombre de fils utilisés pour commander l'afficheur, comme, par exemple lorsqu'on dispose de très peu de broches d'entrées sorties disponibles sur un microcontrôleur. Dans ce cas, on peut utiliser le mode quatre bits de l'afficheur. Les données sont alors écrites ou lues en envoyant séquentiellement les quatre bits de poids fort suivi des quatre bits de poids faible. Une impulsion positive d'au moins 450 ns doit être envoyée sur la ligne E pour valider chaque demi-octet ou nibble. Dans les deux modes, on peut, après chaque action sur l'afficheur, vérifier que celui-ci est en mesure de traiter l'information suivante. Pour cela, il faut demander une lecture en mode commande, et tester le flag Busy BF. Lorsque BF=0, l'afficheur est prêt à recevoir une nouvelle commande ou donnée. Il se peut qu'on dispose encore de moins de broches disponibles dans l'application envisagée. Dans ce cas, on peut alors relier la ligne R/W à la masse de façon à forcer l'afficheur en écriture.

On a alors besoin, hors alimentation de seulement six fils en mode 4 bits, et dix fils en mode 8 bits, pour commander l'afficheur, mais on ne peut alors plus relire l'afficheur. Ceci n'est pas gênant dans la mesure où on sait ce qu'on a écrit sur l'afficheur, mais on ne peut alors plus relire le flag Busy. Il faut alors utiliser des temporisations après chaque écriture sur l'afficheur. On perd alors un peu en temps d'affichage, mais on gagne une broche d'entrée /sortie [14].

I.9.Conclusion :

Partant d'une présentation générale sur les microcontrôleurs, nous avons ensuite défini la famille des PICs et plus particulièrement la PIC 16F876. En conclusion dans ce chapitre nous pouvons dire que le microcontrôleur peut bien jouer le rôle d'une unité de contrôle pour notre carte. Pour faire fonctionner cette unité de contrôle, il faut la programmer et l'adapter à un compilateur de programmation.

Chapitre II :

Le bus I2C et capteur DS1621

II.1 .Introduction :

Dans de nombreux domaines ayant trait au quotidien et à la technique, la mesure de température est la tâche de mesure la plus importante. En concurrence avec les processus analogiques classiques, les capteurs numériques « intelligents » capables de faire plus que de mesurer uniquement la température, prennent une place de plus en plus importante. Le DS1621 de Dallas Semi-conducteur en est l'exemple-type. La tâche première d'un capteur de température numérique moderne est, et reste, bien entendu, toujours de mesurer la température instantanée. Pour ce faire, la puce mesure sa propre température qui est pratiquement celle de son environnement direct, la convertit en une valeur analogique et l'envoie, sériellement, vers un microcontrôleur.

II.2.Présentation :

Dans les années 1982, une véritable « guerre des standards » éclate dans le monde de l'électronique. Cette guerre touche tous les représentants du monde des composants, c'est à cette occasion, que PHILIPS, un des leaders mondiaux de l'électroménager et des composants électroniques, lance le bus I2C (Inter Integrated Circuit) [13].

Ce bus est destiné, entre autre, à des applications dans le domaine de la domotique. Ce qui le différencie, par rapport à d'autre bus comme le bus CAN par exemple, c'est sa diversité aux niveaux de ses utilisations. En effet, il existe une multitude de périphériques utilisant ce bus, et même utilisable avec n'importe qu'elle microcontrôleur, de manière software (à travers de logiciels) [14].

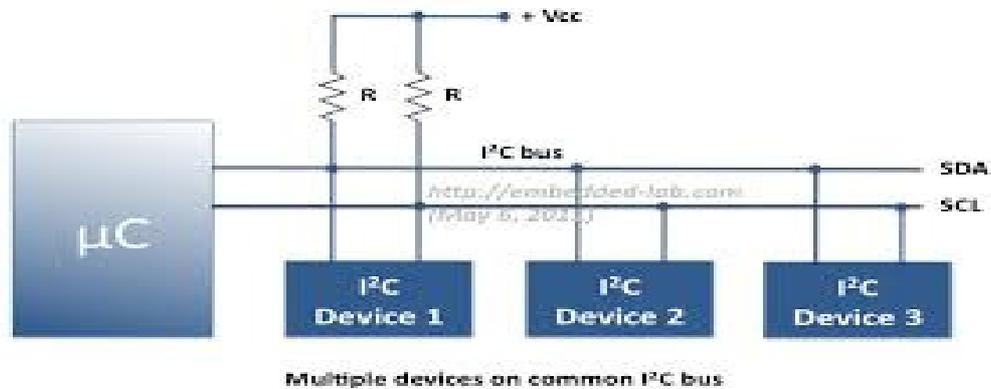


Fig. II.1 : la topologie esclave.

II.3. caractéristiques du bus I2C :

Le bus I2C a été inventé pour simplifier la communication entre des composants divers dont le signal 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 3 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 à un schéma classique (8bits de données, 16 bits d'adresse+ Les bits de contrôle) [15].

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 [15].

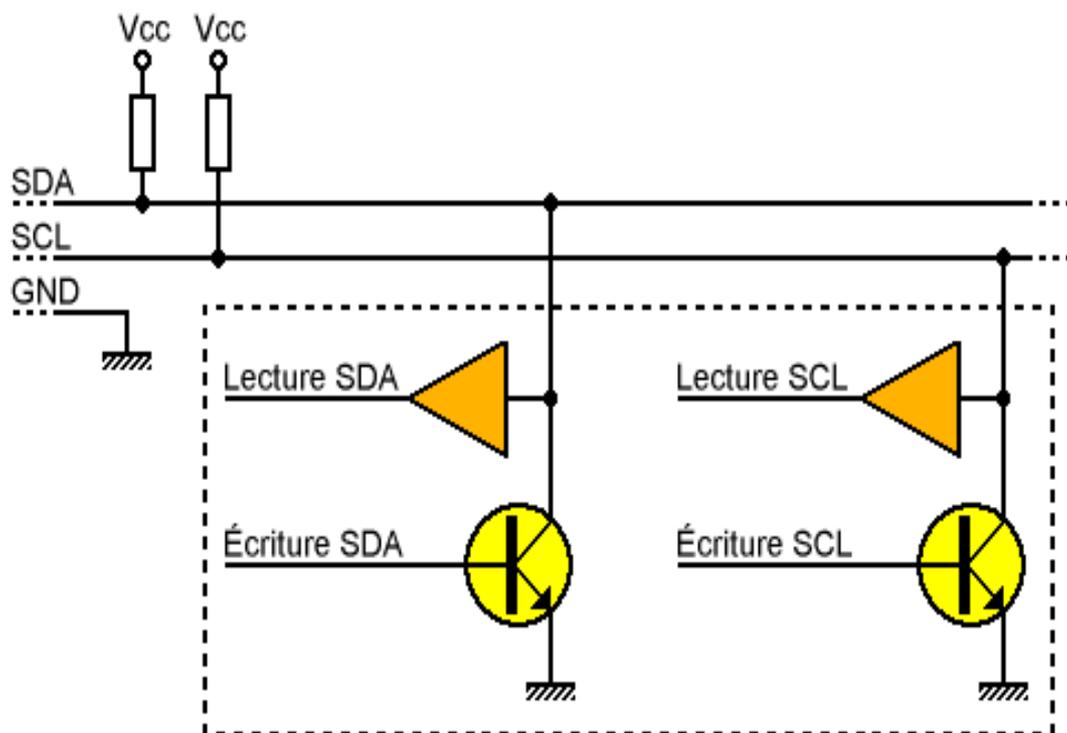


Fig. II. 2 : schéma d'une alimentation extérieure.

II.4.Codage de l'information :

L'I2C est basé sur le code NRZ (Non Retour à Zéro), c'est un codage de type binaire (0 ou 1), qui ne connaît pas de valeur intermédiaire que ces deux états. Les valeurs de tensions qui correspondent à 0 ou à 1 sont symétriques par rapport à 0V. On en déduit donc que si on inverse le branchement des deux fils, le sens sera aussi inversé [13].

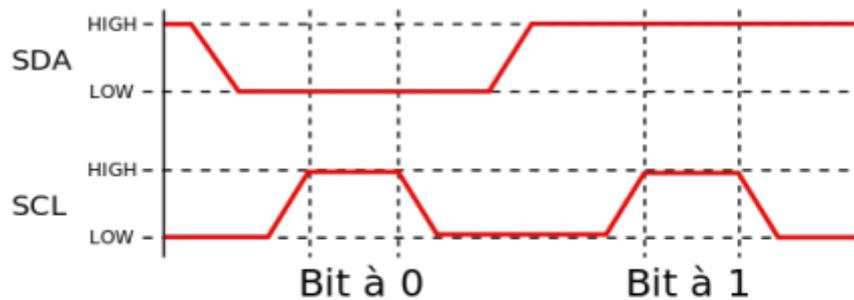


Fig. II.3: Codage de l'information.

Sur le graphique ci-dessus, on voit qu'il faut que le niveau High ou Low de la ligne des données(SDA) soit à un niveau stable au moment du passage au niveau High sur la ligne d'horloge(SCL) pour que le bit soit lu. De plus, on peut aussi voir que l'état logique 0, représenté par l'état LOW sur le graphique correspond à l'état dit « dominant » pour la ligne d'horloge et que l'état logique 1, représenté par l'état HIGH correspond à l'état dit « récessif ». Ce bus utilise des résistances de type « pull up » qui sont des résistances de tirage vers le 5V, c'est pourquoi, lorsque le bus n'est pas utilisé, SDA et SCL sont aux niveaux hauts [13].

II.5.Le protocole I2C :

Il y a une condition appelée « Start » qui est toujours la condition du début des échanges de données, cela se traduit par le passage de l'état logique 1(niveau haut) à l'état 0(niveau bas) pour SDA avec le niveau haut pour SCL. La condition «Stop » détermine la fin des échanges avec le passage de SDA de l'état 0(niveau bas) à l'état logique 1(niveau haut) avec la ligne SCL à 1.

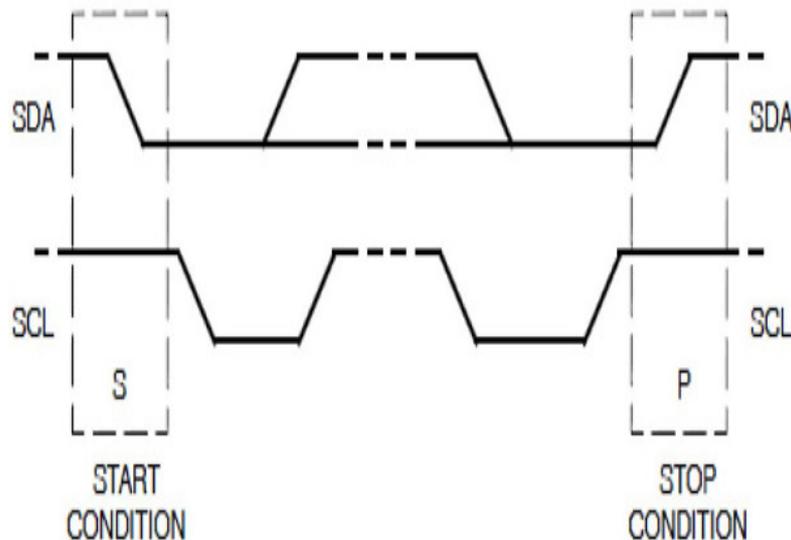


Fig. II.4: les conditions Start et Stop.

II.6.L'échange entre le maitre et l'esclave :

Un exemple type d'échange entre le maitre et l'esclave :

Cela commence, comme on l'a vu précédemment par un bit de Start, puis l'adresse du composant esclave, suivi du bit de lecture/écriture mis à zéro ($R/W=0$). L'esclave répond au maitre par un bit d'acquiescement (ACK) puis peut demander une pause. Une fois la pause terminée, le maitre envoie les données puis reçoit un bit d'acquiescement ou de non acquiescement envoyé par l'esclave, il peut demander alors une pause. Le maitre envoie un bit de restart, puis désigne le même esclave avec cette fois le bit d'écriture/lecture à 1, le maitre devient alors récepteur et l'esclave émetteur. L'émetteur envoie alors une série de données, toujours suivi d'un bit d'acquiescement envoyé par le maitre. Une fois le dialogue terminé, le maitre envoie un bit de non acquiescement pour terminer l'échange ainsi qu'un bit de stop à tous les esclaves [14].



Fig. II.5: L'échange entre le maître et l'esclave.

Schématiquement, voici à quoi ressemble la condition d'acquiescement ou de non-acquiescement :

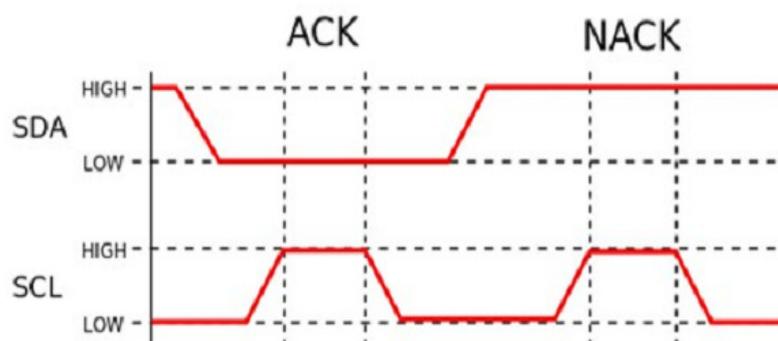


Fig. II.6: « acquiescement et non acquiescement ».

II.6.2. La condition « Pause » :

Schématiquement, voici à quoi ressemble la condition de Pause :

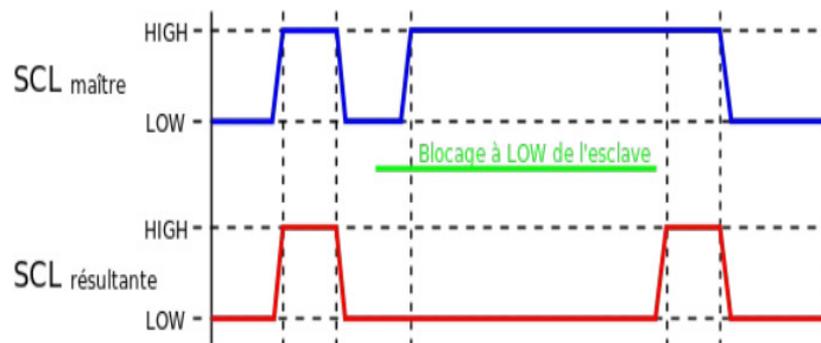


Fig. II.7: La condition « Pause ».

II.6.3. La condition « Stop » :

Schématiquement, voici la condition Stop :

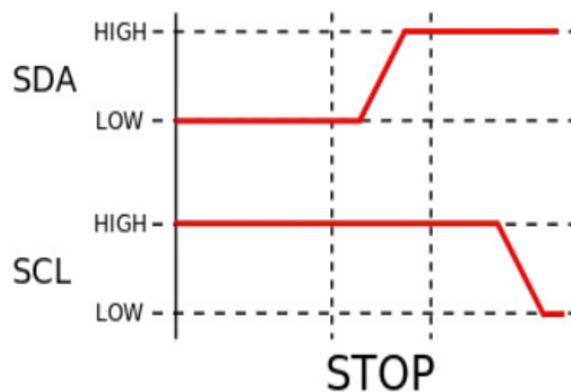


Fig. II.8: La condition « stop ».

II.7.La prise 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 [16].

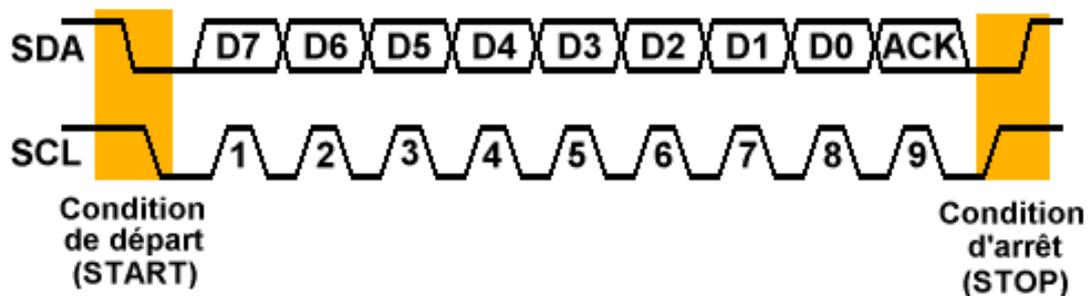


Fig. II.9 : Exemple de condition de départ et d'arrêt.

II.8.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 voit le '0' et peut alors passer à la suite [17].

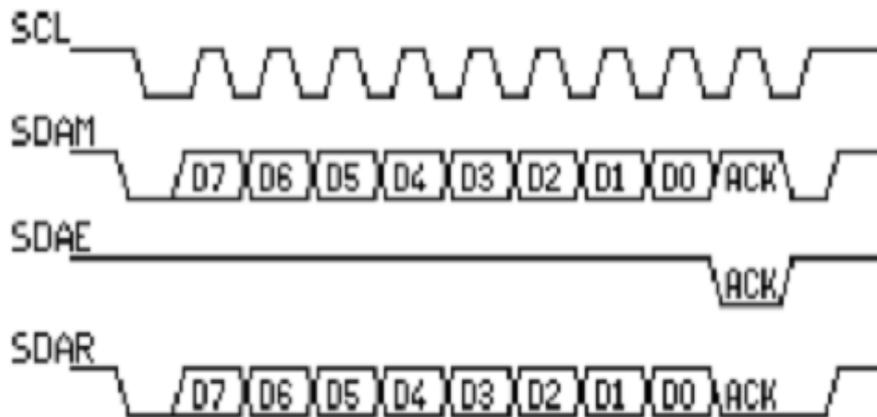


Fig. II.10: La transmission d'un octet.

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.

II.8.1. 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 [17].

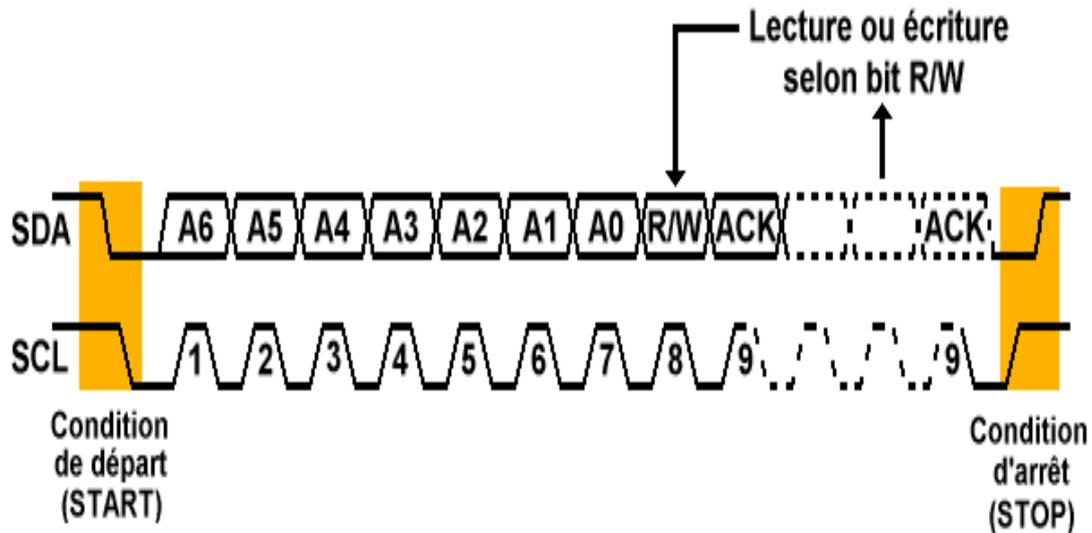


Fig. II.11 : Exemple d'octet d'adresse.

II.9. Le Capteur de Température DS 1621:

II.9.1. Définition:

Le capteur rend une lecture numérique directe de la température mesurée possible, sans qu'un convertisseur analogique/numérique soit requis et il a déjà été calibré par le fabricant. Le capteur contient un thermomètre numérique et dispose en outre de fonctions comme une mémoire ou un thermostat numérique programmable [14].

Le capteur de température DS1621 permet de mesurer des températures comprises entre -55°C et $+125^{\circ}\text{C}$, avec une résolution de 0.5°C . La donnée mesurée est transmise en I2C sur 9 bits (un octet contenant les 8 MSB et un octet contenant le LSB).

Le composant compte 8 broches, peut être alimenté entre 2.7V et 5.5V et communique par bus I2C au choix à 100kHz ou à 400kHz. Son brochage, extrait de la datasheet du composant.

II.9.2. Application:

- Commandes thermostatiques.
- Systèmes industriels.
- Produits de consommation.
- Thermomètres.

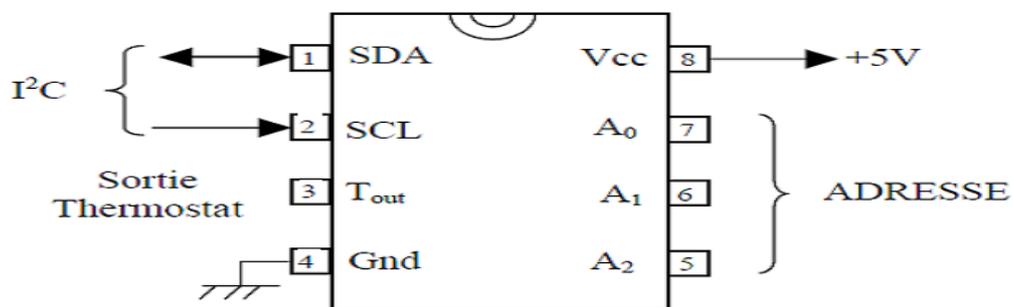


Fig. II.12 : Brochage du capteur DS 1621.

Tout dialogue du Maître avec un DS1621 esclave doit débiter par l'envoi d'un mot de contrôle [6].

II.9.3. Adressage :

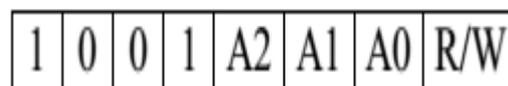


Fig. II.13 : Adressage du DS1621.

Adresse de la mémoire : 0x90 en écriture et 0x91 en lecture, avec les trois lignes d'adresse A2, A1 et A0 à 0.

II.9.4. Description des pins:

SDA - 2-Wire Serial Data Input/Output.

SCL - 2-Wire Serial Clock.

GND - Ground.

TOUT - Thermostat Output Signal.

A0 - Chip Address Input.

A1 - Chip Address Input.

A2 - Chip Address Input.

VDD - Power Supply Voltage.

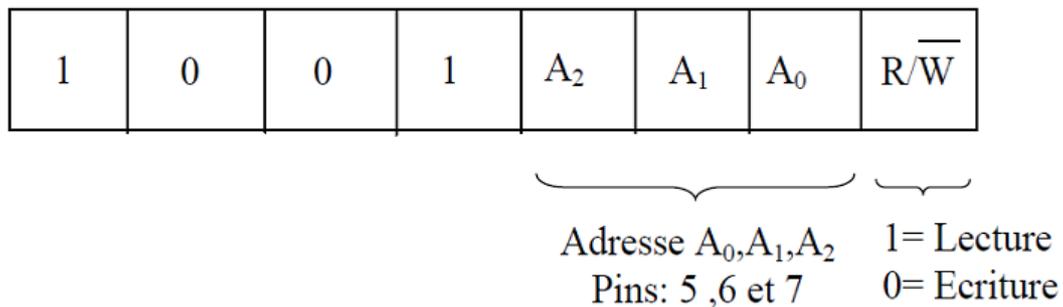
II.9.5. Mot de contrôle:

Fig. II.14: Mot de contrôle.

II.9.6. Commande:

Le maître envoie ensuite une commande, suivie ou non de données.

Les commandes AC, EE et 22 ne sont pas suivies de l'envoi de données. Par contre la commande AA est suivie de l'envoi par le DS1621 de 2 octets représentant la température. Les commandes A1 et A2 sont suivies elles aussi de 2 octets envoyés par le maître dans le cas d'une écriture des seuils du thermostat ou bien retournés par le DS1621 dans le cas d'une lecture [6].

II.9.7.DS1621 bloc-diagramme fonctionnel :

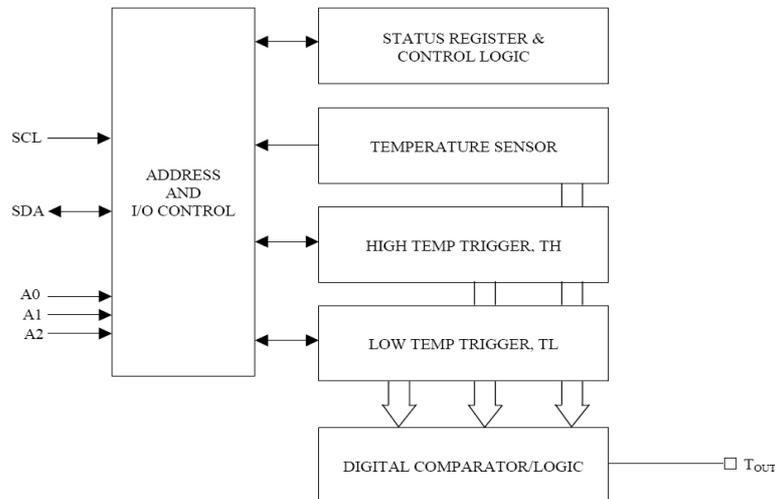


Fig. II.15: Brochage du capteur DS 1621.

II.10.Commande du thermostat:

Dans son mode de fonctionnement, le DS1621 fonctionne comme un thermostat avec hystérésis programmable, comme illustré à la Figure 3. Le thermostat sortie des mises à jour dès qu'une conversion de la température est terminée.

Lorsque la température de la DS1621 satisfait ou dépasse la valeur stockée dans le voyage de haute température s'inscrire (TH), la sortie devenue active et restera active jusqu'à ce que la température tombe en dessous de la température enregistrée dans le registre de déclencheur de basse température (TL). De cette manière, une quantité quelconque d'hystérésis peut être obtenue.

L'état actif pour la sortie est programmable par l'utilisateur afin que l'état actif peut être une logique "1" (VDD) ou une logique "0" (0V). Cela se fait à l'aide de la POL bit dans le registre de configuration comme expliqué dans la section de fonctionnement et le contrôle de cette feuille de données [7].

II.10.1.Fonctionnement du thermostat en sortie :

II.10.2.DQ (sortie Thermostat, actif = élevé) :

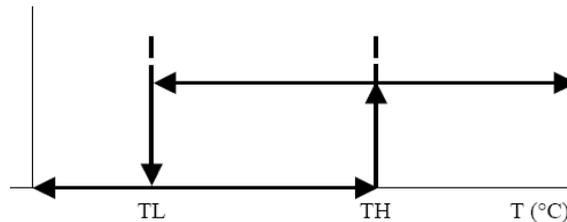


Fig. II.16 : Sortie thermostat.

II.11.Fonctionnement et contrôle :

Le DS1621 doit avoir les réglages de température résidents dans les registres de TH et TL pour un fonctionnement thermostatique. Un registre d'État/configuration détermine également la méthode de fonctionnement que le DS1621 utilisera dans une application particulière, comme indiquant l'état de l'opération de conversion de température [7].

II.11.1.Le registre de configuration est défini comme suit :

MSb	Bit 6	Bit5	Bit 4	Bit 3	Bit 2	Bit 1	LSb
DONE	THF	TLF	NVB	X	X	POL	1SHOT

Fig. II.17 : Le registre de configuration.

FAIT = Conversion faite peu. « 1 » = conversion complète, « 0 » = Conversion en cours.

THF = drapeau haute température. Ce bit est défini à « 1 » lorsque la température est supérieure ou égale à la valeur de Th Il restera « 1 » jusqu'à ce que reset en écrivant « 0 » dans cet endroit ou pouvoir retirer de l'appareil.

Cette fonction fournit une méthode pour déterminer si le DS1621 a jamais été soumis à des températures supérieures à TH tandis que l'alimentation est activée.

TLF = indicateur de basse température. Ce bit est défini à « 1 » lorsque la température est inférieure ou égale à la valeur de TL. Il restera « 1 » jusqu'à ce que reset en écrivant « 0 » dans cet endroit ou pouvoir retirer de l'appareil. Cette fonction fournit une méthode pour déterminer si le DS1621 ne jamais a été soumis à des températures de TL, tandis que l'alimentation est activée.

VAB = indicateur occupé à mémoire non volatile. « 1 » = écriture vers une cellule de mémoire E2 en cours, « 0 » = non volatile de mémoire n'est pas occupé. Une copie à E2 peut prendre jusqu'à 10 ms.

POL = Bit de polarité de sortie. « 1 » = actif élevé, « 0 » = basse active. Ce bit est non volatile.

1Shot = un Mode de tir. Si 1SHOT est « 1 », le DS1621 effectuera une conversion de la température à la réception du protocole commencé à convertir T. Si 1SHOT est « 0 », le DS1621 en continu effectue les conversions de température. Ce bit est non volatile.

X = réservés.

Pour thermostat typique fonctionnement le DS1621 fonctionnera en mode continu. Toutefois, pour les applications nécessitant qu'une seule lecture à certains moments ou pour économiser l'énergie, le mode One-Shot peut servir. Notez que la sortie du thermostat (TOUT) restera dans l'État où qu'elle était après la conversion de température valide dernier cycle en mode One-Shot.

II.12. Les deux modes de fonctionnement du DS1621 :

1. esclave de récepteur : données sérielles et horloge sont reçus par le biais de SDA et SCL. Après que chaque octet est reçu un bit acknowledge est transmis. DÉMARRAGE et arrêt des conditions sont reconnus comme le début et la fin d'un transfert série. Reconnaissance de l'adresse est interprétée par matériel après réception de l'embout de direction et adresse esclave [4].

2. esclave émetteur : le premier octet est reçu et traité comme en mode esclave récepteur. Toutefois, dans ce mode le bit de direction indique que le sens de transfert est inversé. Données en série sont diffusées le SDA le DS1621 tandis que l'horloge série est entrée sur SCL. DÉMARRAGE et arrêt des conditions sont reconnus comme le début et la fin d'un transfert série [4].

II.13. Ensemble de commandes :

Pour écrire dans le DS1621, le capitaine attribuera l'adresse de l'esclave de la DS1621 et le bit R/W est défini sur « 0 ». Après avoir reçu un accusé de réception, le maître de bus fournit un protocole de commande. Après avoir reçu ce protocole, le DS1621 émettra un accusé de réception et ensuite le maître peut envoyer des données vers le DS1621. Si le DS1621 doit être lue, le capitaine doit envoyer le protocole de commande comme avant et puis émettre une répétées

Condition de départ et l'octet de contrôle à nouveau, cette fois avec le bit R/W valeur « 1 » pour permettre la lecture des données depuis le DS1621. La commande défini pour le DS1621 comme indiqué dans le tableau 3 est la suivante :

II.13.1. Lire la température [AAh]:

Cette commande lit le dernier résultat de conversion de température. Le DS1621 enverra 2 octets, dans le format décrit précédemment, qui sont le contenu de ce registre.

II.13.2. Accéder à TH [A1h]:

Si R/W est « 0 » cette commande écrit dans le registre de TH (haute température). Après l'émission de cette commande, les prochaines 2 octets écrits dans le DS1621, dans le même format tel que décrit pour la lecture de la température, définira le seuil de température élevée pour l'utilisation de la sortie TOUT. Si R/W est « 1 » la valeur stockée dans ce registre est collationnée.

II.13.3. Accéder aux TL [A2h]:

Si R/W est « 0 » cette commande écrit dans le registre de TL (basse température). Après l'émission de cette commande, les prochaines 2 octets écrits dans le DS1621, dans le même format tel que décrit pour la lecture de la température, définira le seuil de température élevée utilisation de la sortie TOUT. Si R/W est « 1 » la valeur stockée dans ce registre est collationnée.

II.13.4. Accéder à Config [ACh]:

Si R/W est « 0 » cette commande écrit dans le registre de configuration. Après l'émission de cette commande, l'octet de données suivant est la valeur à écrire dans le registre de configuration. Si R/W est « 1 » le prochain octet de données lire est la valeur stockée dans le registre de configuration.

II.13.5. Lire compteur [A8h]:

Cette commande lit la valeur Count_Remain. Cette commande n'est valable que si R/W est « 1 ».

II.13.6.Lire la pente [A9h]:

Cette commande lit la valeur Count_Per_C. Cette commande n'est valable que si R/W est « 1 ».

II.13.7.Commencer à convertir T [EEh]:

Cette commande commence une conversion de la température. Aucune donnée supplémentaire n'est nécessaire. En mode One-Shot effectuera la conversion de la température, et puis le DS1621 restera inactif. En mode continu cette commande va donc entreprendre des conversions continues.

II.13.8.Arrêter convertir T [22h]:

Cette commande arrête la conversion de la température. Aucune donnée supplémentaire n'est nécessaire. Cette commande peut être utilisée pour stopper un DS1621 en mode de conversion continue. Après l'émission de cette commande, la mesure de la température actuelle sera terminée et le DS1621, restera inactif jusqu'à ce qu'un T de commencer à convertir est émis pour rétablir son fonctionnement continu.

II.14.Avantage et inconvénient de ce thermostat :

L'avantage de ce circuit est ce qui est très facile à réaliser et qui demande peu de composants avec un coût vraiment réduit autre avantage, est que ce circuit peut commander le démarrage ou l'arrêt d'un refroidisseur, si la température est haute le circuit va allumer un refroidisseur si la température est basse.

Son inconvénient est qu'en cas de coupure d'électricité, il faut reconfigurer les valeurs maximale et minimale

II. 15. Conclusion :

Le bus I2C est un protocole de communication rapide et facile à mettre en œuvre. Il permet de créer des bus contenant plusieurs maîtres et esclaves. Ce protocole prévoit la gestion automatique des conflits entre composants. (Ce sont les composants eux-mêmes qui s'en occupent).

Cette étude se limite au fonctionnement en mode mono-maître. En revanche le protocole ne prévoit aucune vérification des données transmises. Il est alors difficile de savoir si les valeurs reçues sont exactes.

Chapitre III :

Etude technique

III.1.Introduction :

L'incubateur néonatal, appelé aussi « couveuse » dans le langage courant, est un appareil qui permet de maintenir des nouveau-nés, malade ou prématurés, dans des conditions climatiques favorables à une évolution positive de leur état. Il est constitué d'un habitacle transparent qui permet la surveillance visuelle du nouveau-né.

Notre projet se compose de capteurs de température numérique pour contrôler la température plus précis dans diverses industries. Notre système permet de surmonter les inconvénients des systèmes de thermostat/analogique en termes de précision.

L'étude qui suit est divisés en deux parties, la première partie est consacrée à tout ce qu'est développement de la carte et la deuxième partie est relative à la programmation de la carte.

III.2. schéma synoptique :

Notre carte est basée sur une carte mère au centre de laquelle se situe un microcontrôleur 16f876a, elle sert pour contrôler la température. Aussi prend une entrée provenant d'un capteur de température DS1621 avec le protocole de communication I2C, et a une sortie qui est reliée à un élément de commande, tel qu'un dispositif de chauffage et de ventilation plus un système de réglage de la consigne.

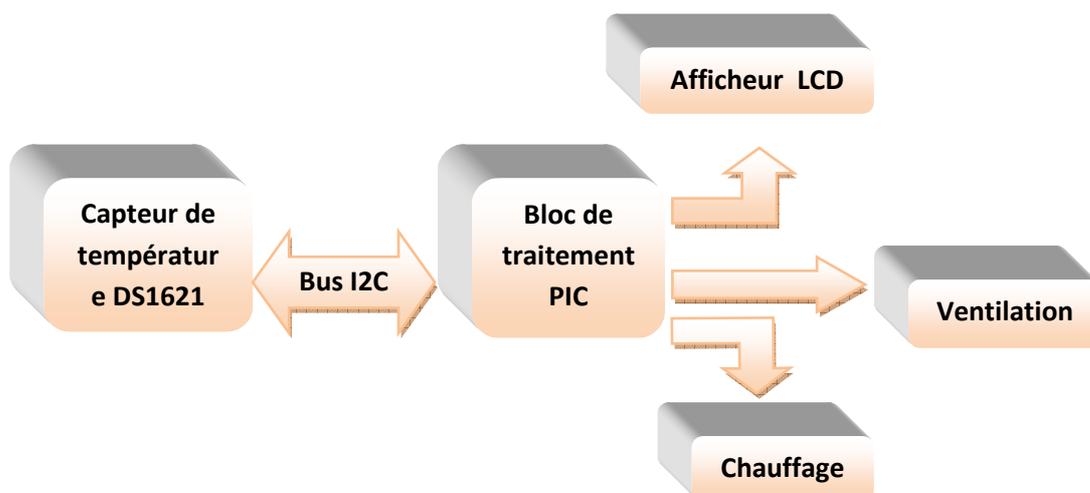


Fig. III.1: schéma synoptique.

III.3. simulation du schéma globale :

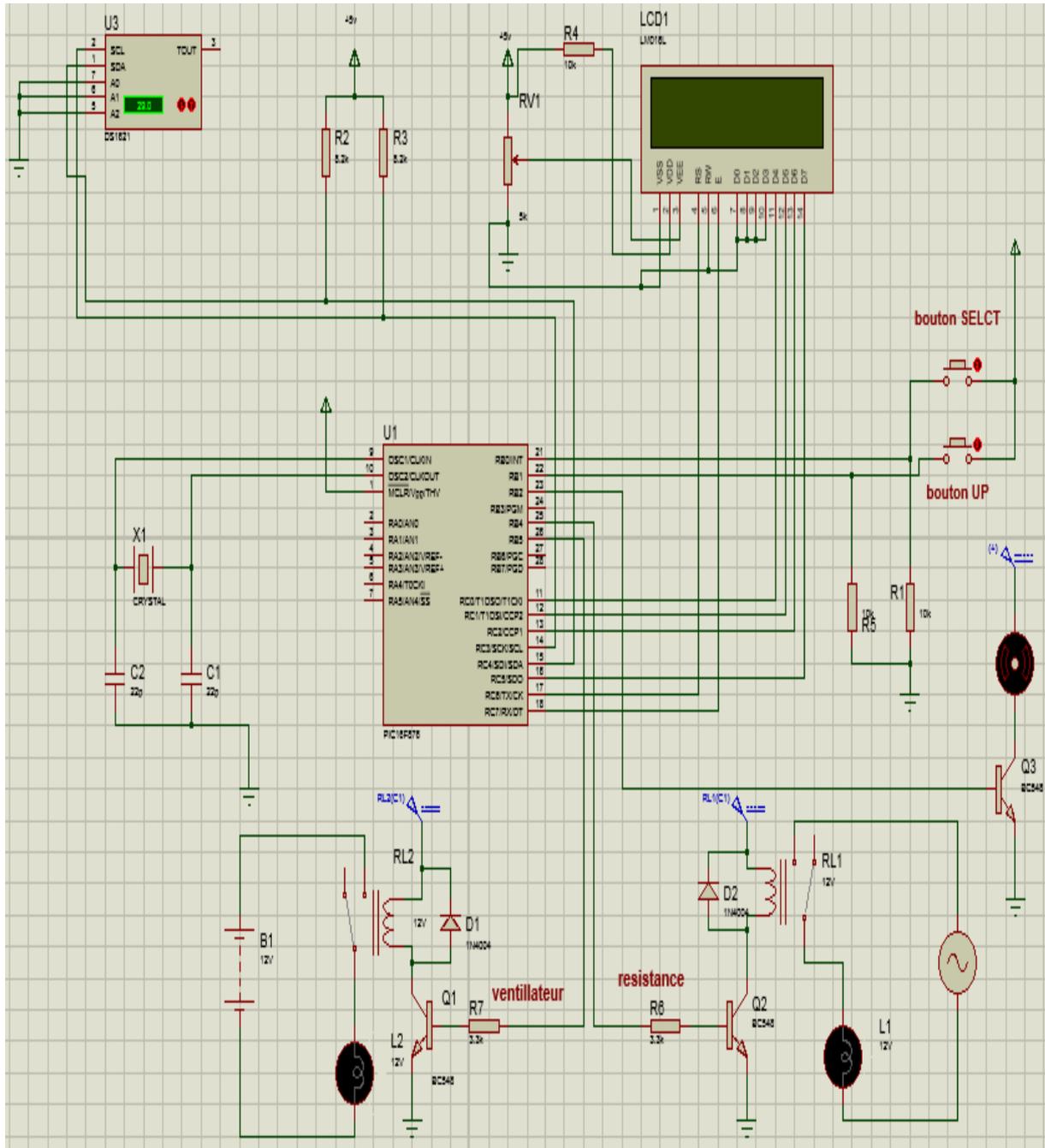


Fig. III.2: simulation du schéma globale.

III.4. Caractéristiques techniques :

III.4.1. Condition ambiantes :

Avec température ambiante de la pièce à 25 °C et température interne de la couveuse 36°C.

III.5. Explication de schéma de la carte :

III.5.1. Bloc d’Alimentation :

L'alimentation du montage est assurée soit par une tension stabilisée de 12 V, soit par une pile de 9 V. Un régulateur de tension de type 7805 fournit 5 V au montage. Deux diodes de protection IN4007 permettent d'éviter toute destruction de composant dans le cas de l'inversion des fils de raccordement d'alimentation [13].

Les deux condensateurs C2 et C3 sont des découplages qui évitent une éventuelle entrée en oscillation du régulateur. C1 est le condensateur « réservoir » qui filtre la tension continue issue d'un redresseur. On prévoit souvent un autre condensateur « réservoir » en sortie C4 [13].

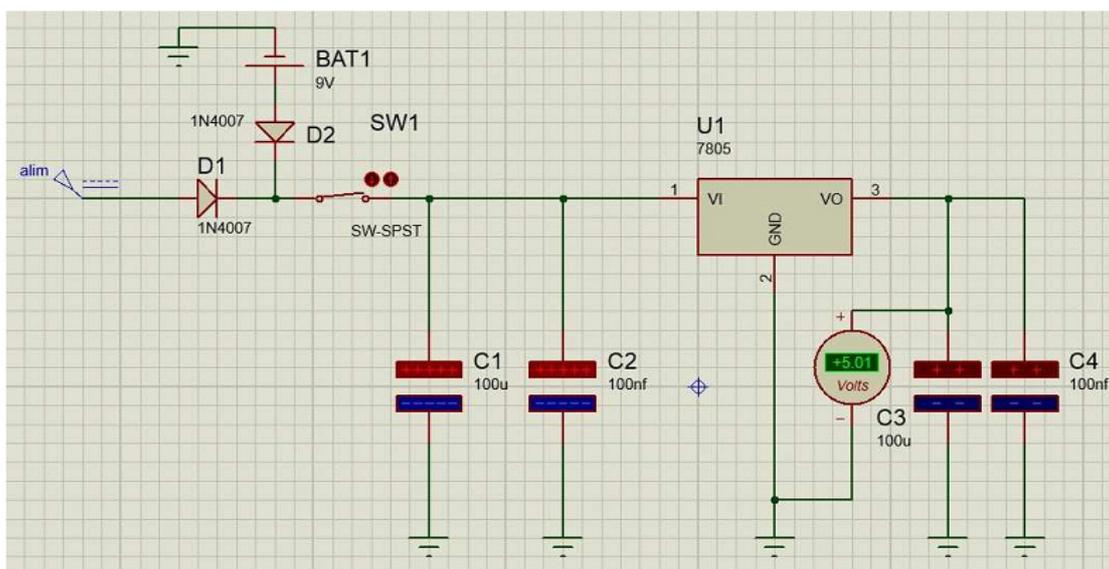


Fig. III.3 : schéma de bloc d'alimentation.

III.5.2. Bloc de traitement :

Le cœur du montage est le μC d'un PIC 16F876A, qui est cadencé par un quartz de 4MHz. Les broches OSC1 et OSC2 ou CLKIN et CLOUT permettent de faire fonctionner L'oscillateur interne du PIC qui peut être un quartz, un résonateur céramique, un oscillateur externe ou un réseau RC dont le rôle est de créer des impulsions de fréquences élevées.

La broche MCLR sert à initialiser le μC en cas de la mise sous tension, de remise à zéro externe, de chien de garde et en cas de la baisse de tension d'alimentation [12].

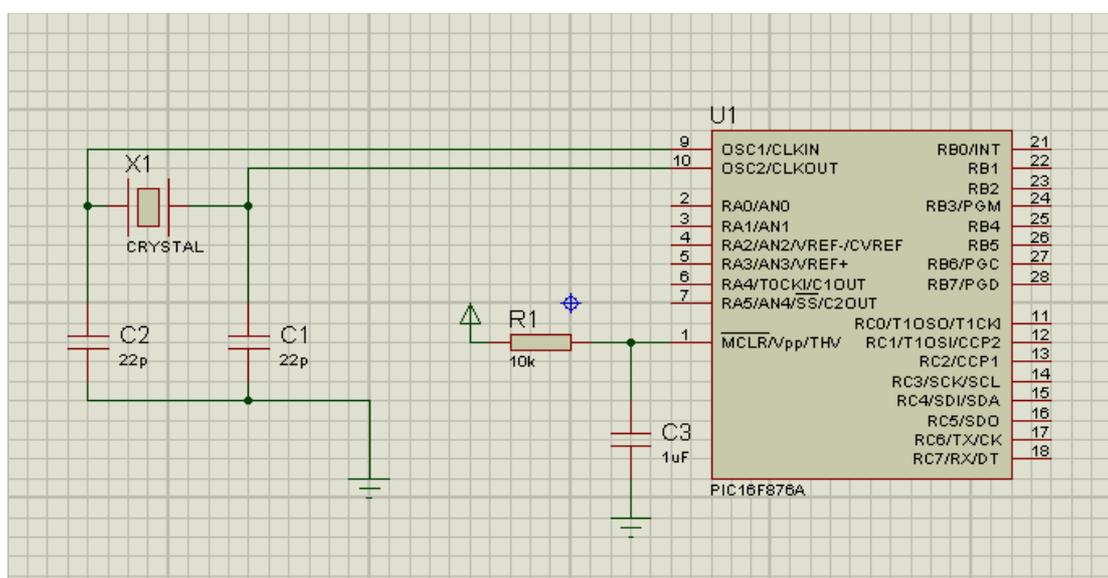


Fig. III.4 : schéma de bloc de traitement.

III.5.3: bloc d'entrée :

III.5.3.1: Les boutons:

Comme l'on a explicité auparavant, le clavier est constitué de 2 boutons poussoirs qui servent dans notre montage à manipuler le menu de gestion de la manière suivante :

Bouton select : permet de changer le fonctionnement du montage en mode consigne ou bien température référence, il permet aussi de sélectionner la température.

Bouton up: c'est pour incrémenter et décrémenter la consigne de température.

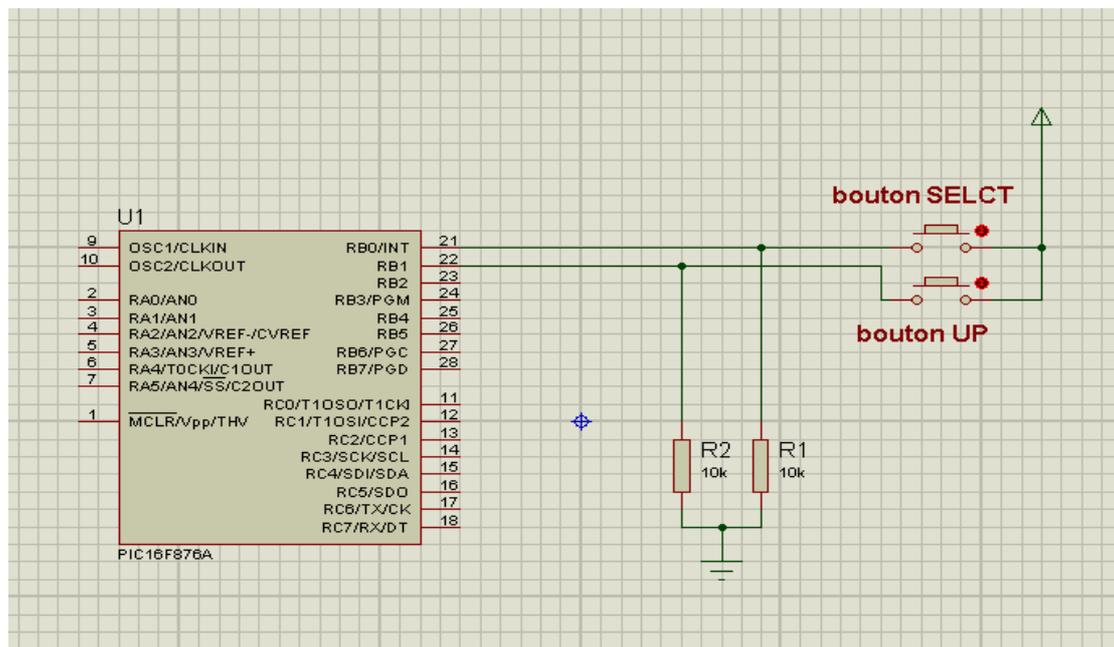


Fig. III .5: schéma de bloc d'entrée (clavier).

III.5.3.2. Le capteur DS1621:

A présent, comme tous les régulateurs de température, nous avons installé un capteur de température numérique le DS1621. Il est utilisé pour contrôler la température avec précision dans diverses industries. Ce capteur communique avec le microcontrôleur grâce au protocole I2C [13].

- Mesure de température (-55 °C à +125 °C par incrément de 0,5 °C), conversion de la température en mot de 16 bits en moins d'une seconde.
- Configuration possible en thermostat (configuration des seuils de température haut et bas non volatile).
- Version en boîtier DIP 8 broches.

III.5.3.3. Bus i2c :

Le protocole permet de mettre en communication un composant **maître** (généralement le microprocesseur) et plusieurs périphériques **esclaves**. Plusieurs maîtres peuvent partager le même bus, et un même composant peut passer du statut d'esclave à celui de maître, ou inversement. Toutefois, la communication n'a lieu qu'entre un seul maître et un seul esclave.

Notons également que le maître peut également envoyer un ordre à tous ses esclaves simultanément (par exemple, une mise en sommeil ou une demande de réinitialisation) [13].

Il permet de communiquer sur 3 fils :

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

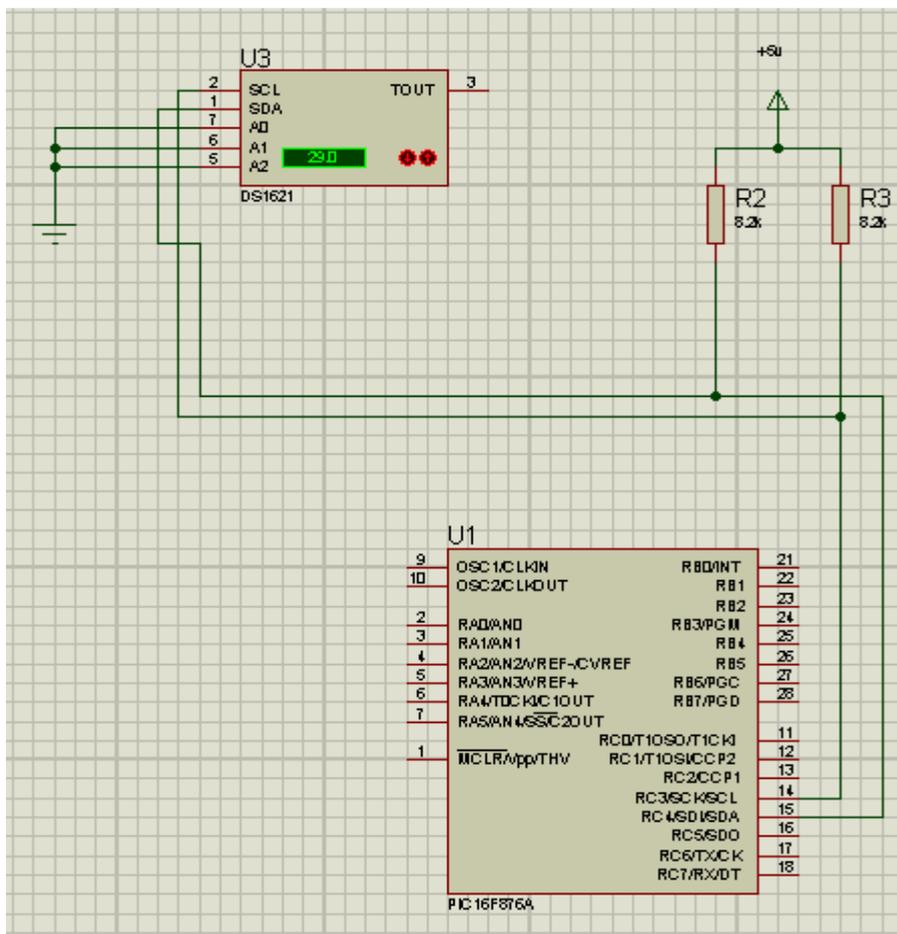


Fig. III .6: schéma de capteur DS1621.

III.6. Bloc de sortie :

III.6.1. Sortie commande de l'afficheur :

On a choisi le mode 4 bits, on a consacré le PORT C qui est configuré en sortie, les premières lignes du port sont reliées aux lignes 4D-7D et sont décidées au transfert des données commandes, les lignes D0-D3 sont reliées à la masse.

Les lignes E-RS sont reliées à RC6 et RC7, car on utilise l'afficheur que pour l'écriture la ligne RW est reliée à la masse pour valider que l'écriture. La résistance variable RVI règle le contraste (luminosité) de LCD [3].

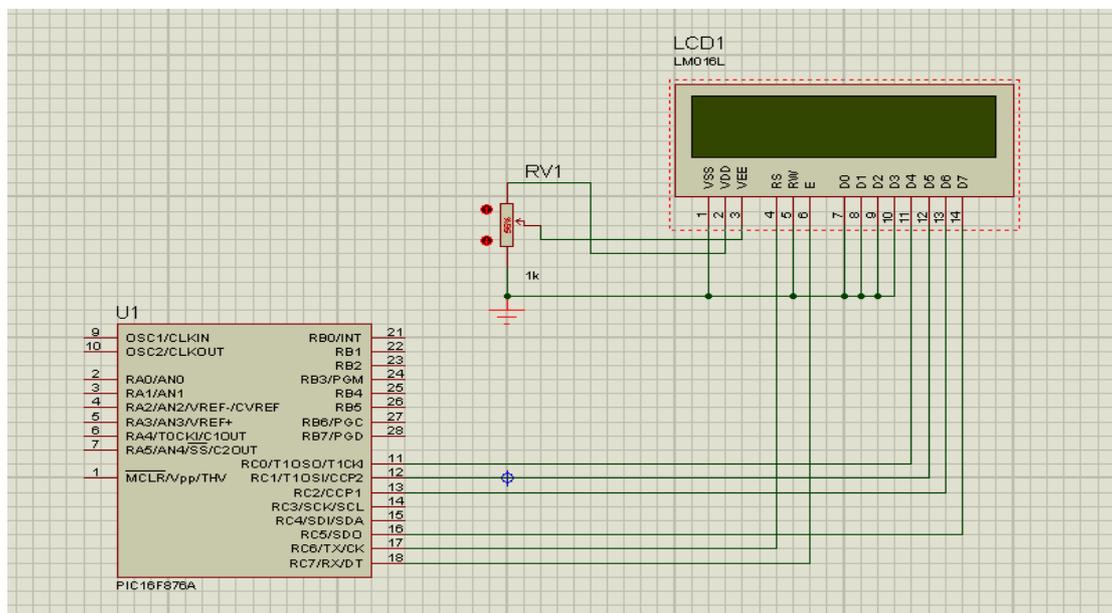


Figure. III .7: Schéma de sortie afficheur.

III.6.2. Sortie relais (chauffage + ventilation) :

Un relais électromécanique est un organe électrotechnique permettant la commutation de liaisons électriques, Le plus important est que le relais un conducteur magnétique. Il est chargé de transmettre un ordre de la partie commande à la partie puissance d'un appareil électrique et permet, entre autres, une isolation galvanique entre les deux parties.

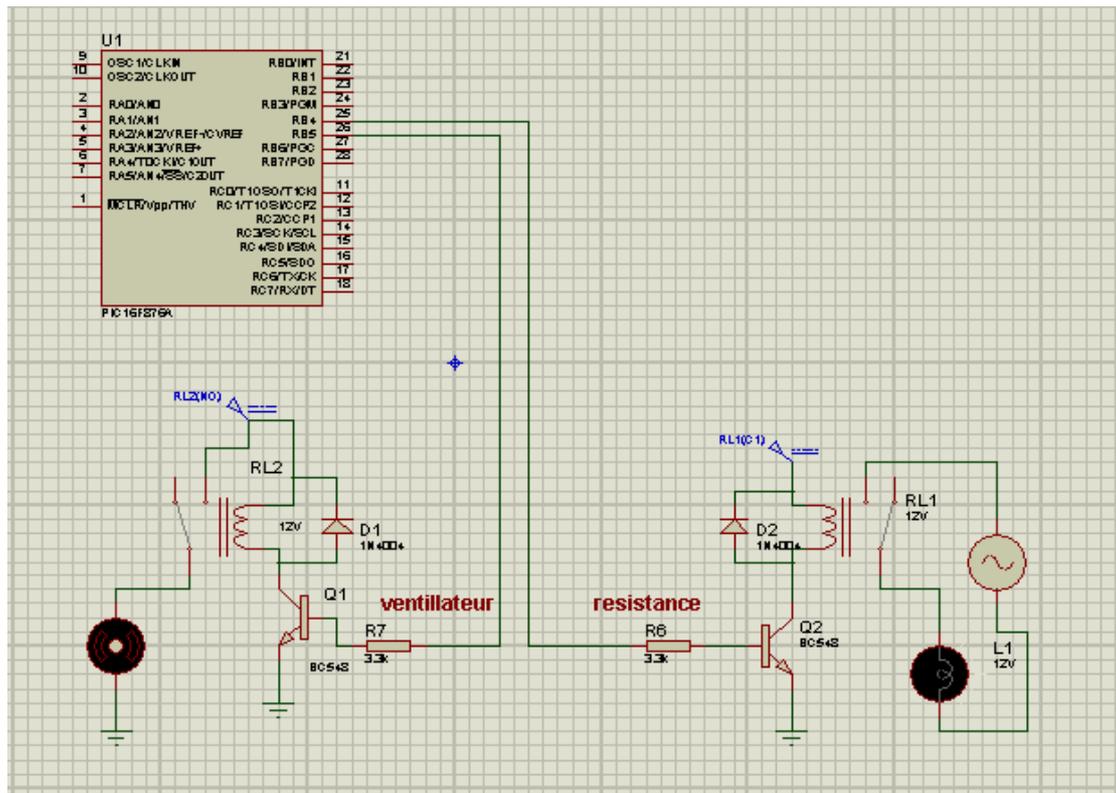


Fig. III.8 : Schéma de bloc sorti (relais).

Les commandes issues des pins RB4 et RB5 servent pour l'actionnement des lampes de chauffage et le ventilateur. Ainsi, la commande RB4 sert à activer une lampe et le ventilateur qui sont le premier niveau de chauffage, alors que la commande RB5 sert à activer le deuxième niveau de chauffage. A cet effet, un relais (12v) est actionné (ventilateur et une lampe) à travers un transistor en commutation relié à RB4, puis un deuxième relais (12v) est activé à travers un transistor en commutation et également relié à RB5.

Le système utilise un capteur de température numérique afin de détecter la température et de transmettre les données au microcontrôleur. Le microcontrôleur PIC 16F876A traite les données et envoie la température à afficher sur l'écran LCD. L'affichage se compose de l'unité d'affichage.

III.7. Conclusion :

Dans ce chapitre nous avons passé en revue toutes les connexions du pic 16f876A avec les différents composants travaillant avec ce pic , comme un pic ne peut fonctionner sans programme ce qui nous emmènera dans le prochain chapitre à aborder la partie soft de cette conception.

Chapitre IV :

Programmation en

miKroC

IV.1. Introduction :

La programmation des PIC se fait par le langage assembleur qui est un langage de bas niveau qui représente le langage machine sous une forme lisible par un humain, Les combinaisons de bits du langage machine sont représentées par des symboles dits « mnémoniques » (du grec mnêmonikos, relatif à la mémoire), c'est-à-dire faciles à retenir.

Le programme assembleur convertit ces mnémoniques en langage machine en vue de créer par exemple un fichier exécutable.

Le développement des environnements de programmation , nous a permis de voir naitre de nouveaux compilateurs qui permettent de programmer avec les langages haut niveau tels que le C , PASCAL,BASIC etc.... Ces environnements comportent aussi des bibliothèques qui permettent de faciliter le développement.

Il existe plusieurs outils de développement, les uns sont gratuits, les autres sont payants. Dans notre recherche de l'outil que nous allons utiliser pour programmer notre PIC , nous avons optés pour le langage C.

Ce choix est à la fois un choix personnel et un choix technologique. D'une part le langage C est utilisé dans différents systèmes et domaines de développement, ce qui nous permettra une évolution future, d'autre part le langage C est l'un des Langues les plus puissants.

IV.2. Avantages du langage C :

Pour de la programmation de base, le C est intéressant. Il permet rapidement, sans gros effort, de développer des programmes fonctionnels. Il permet aussi de s'affranchir de connaissances complexes sur l'architecture des PIC. Il a l'avantage de gérer facilement les boucles, les choix, ainsi que l'affichage [5].

IV.3. Inconvénients du langage C :

Le C n'est pas le langage naturel du microcontrôleur. Il permet de programmer plus intuitivement. Les logiciels de programmation en C transforment alors les lignes en C en lignes assembleurs directement compréhensibles par le microcontrôleur [5].

Pour programmer efficacement, il est souvent nécessaire d'aller voir le code assembleur, il est donc conseillé d'avoir des bases solides en assembleur. Nous avons fait le choix d'utiliser l'environnement de développement MikroC de Mikroelektronika [5].

IV.4. langage et compilateur mikroC pour les microcontrôleurs PIC :

Le langage micro C pour PIC a trouvé une large application pour le développement de systèmes embarqués sur la base de microcontrôleur. Il assure une combinaison de l'environnement de programmation avancée IDE (Integrated Développement Environnement), et d'un vaste ensemble de bibliothèques pour le matériel, de la documentation complète et d'un grand nombre des exemples [4].

Le compilateur micro C pour PIC bénéficie d'une prise en main très intuitive et d'une ergonomie sans faille. Ses très nombreux outils intégrés (mode simulateur, terminal de communication Ethernet, terminal de communication USB, gestionnaire pour afficheurs 7 segments, analyseur statistique, correcteur d'erreur, explorateur de code, mode Débug, ICD...) associé à sa capacité à pouvoir gérer la plupart des périphériques rencontrés dans l'industrie (Bus I2C™, 1Wire™, SPI™, RS485, Bus CAN™, USB, gestion de cartes compact Flash et SD™/MMC™, génération de signaux PWM, afficheurs LCD alphanumériques et graphiques, afficheur LED à 7 segments, etc..) en font un outil de développement incontournable pour les systèmes embarqués, sans aucun compromis entre la performance et la facilité de débogage [4].

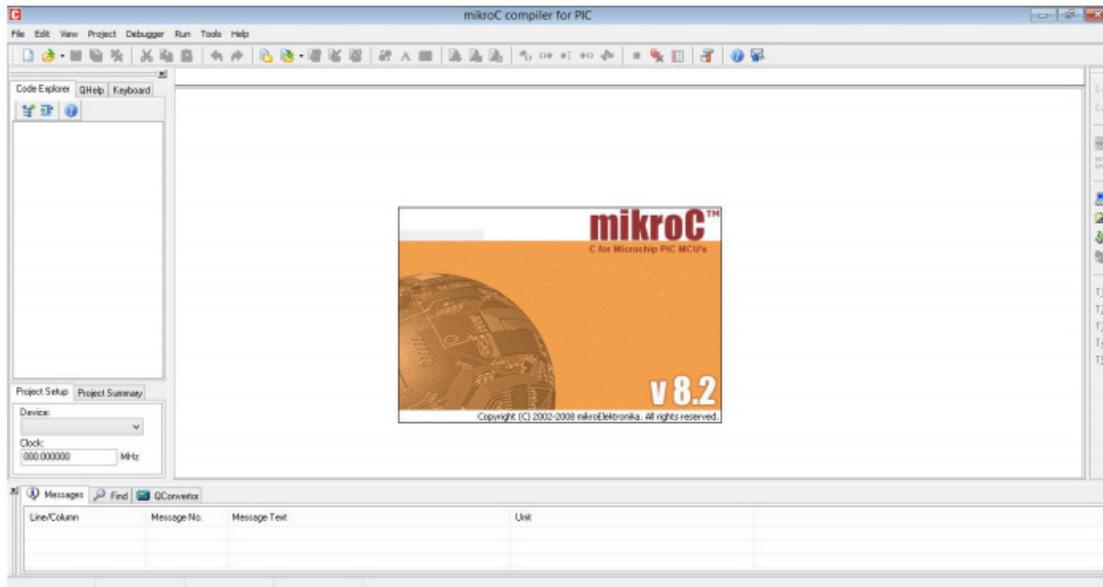


Fig. IV.1: Interface du logiciel MikroC.

IV.4.1. Compilateur mikroC PRO pour PIC :

La nouvelle version appelée mikroC PRO dispose de très nombreuses améliorations du compilateur mikroC : nouvelles variables utilisables, nouvelle interface IDE, amélioration des performances du linker et de l'optimisateur, cycle de compilation plus rapide, code machine généré plus compact (jusqu'à 40 % suivant les cas), nouveaux PIC supportés, environnement de développement encore plus ergonomique, nouveaux exemples d'applications, etc [5].

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.

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.

IV.5. Introduction au langage de programmation mikroC :

IV.5.1. Structure d'un programme en mikroC :

La structure la plus simple d'un programme en micro C, c'est le programme représenté dans le code-source qui nous permettra de faire clignoter une LED connectée au PORTB (par exemple bit 0 du PORTB) du microcontrôleur PIC avec une période de 2 secondes (1 seconde allumée et une seconde éteinte) [4].

IV.5.2. Règles générale d'écriture en micro C :

- Les instructions propres au langage micro C doivent être écrites en minuscule (void main ((void)) .
- Les instructions particulières aux microcontrôleurs doivent être écrites en majuscule (TRISB).
- Les retours à la ligne et les espaces servent uniquement à aérer le code.
- Toutes instructions ou actions se terminent par un point virgule «;».

IV.5.3. les commentaires:

Les phrase après les caractères (//) nommée les commentaires sont utilisés pour préciser le fonctionnement du programme et pour une annotation du programme. Les lignes de commentaires sont ignorées et non compilé par le compilateur. En micro C, les commentaires de programmes peuvent être de deux types:

- **Longs commentaires** : s'étendant sur plusieurs lignes.
- **Courts commentaires** : occupant une seule ligne.

Un commentaire au début d'un programme peut décrire brièvement le fonctionnement du programme et de fournir le nom de l'auteur, le nom du fichier programme, la date à laquelle le programme a été écrit.

Et une liste des numéros de version, ainsi que les modifications dans chaque version. Comme montre le Code-source 2.1 les longs commentaires commencent par le caractère « /* » et se termine par le caractère « */».

De même, de courts commentaires commencent par le caractère « // » et il n'a pas besoin d'un caractère de terminaison.

IV.5.4. Début et fin d'un programme :

En langage micro C, un programme commence avec les mots-clés suivants:

- **void main ()** : Après cela, une accolade ouvrante est utilisée pour indiquer le début du corps de programme. Le programme se termine par une accolade fermante. Ainsi, comme indiqué dans le Code-source 2.1, le programme a la structure suivante :

```
Void main ()  
  
{ /  
  
/ Votre code ici  
  
}
```

- **Fin d'une instruction** : Le point virgule « ; » indique la fin d'une instruction, sinon une erreur du compilateur sera générée.

Exemple :

```
j = 5; // correcte
```

```
j = 5 // erreur
```

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

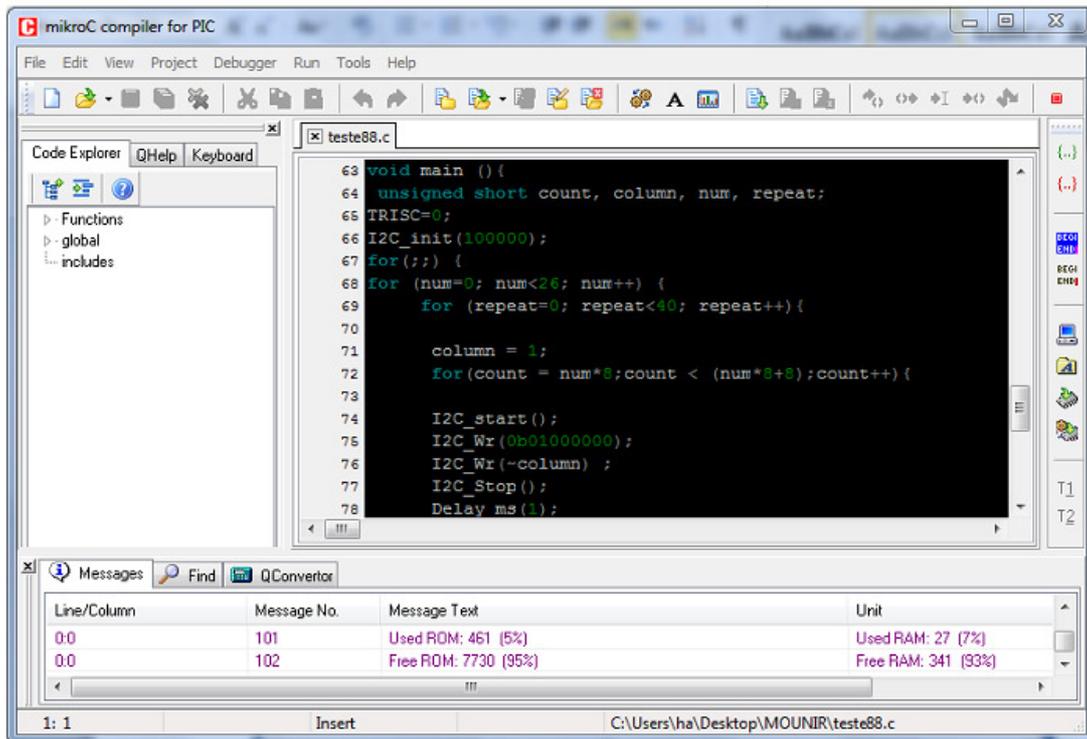


Fig. IV.2: Fenêtre de saisie de programme.

IV.6. 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 I ² C. Exemple : I2C_Write(0xA3);
	I2C_Stop	Lancer un signal STOP. Exemple : I2C_Stop();

Tableau. IV.1 : Bibliothèque du MikroC.

IV.6.1. Fonction d'écriture :

```

void write_pic 16f876a (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) }

```

Fig. IV.3: Fonction d'écriture.

IV.6.2. 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)}
```

Fig. IV.4: Fonction de lecture.

IV.7. Affichage dans LCD :

La bibliothèque de mikroC LCD fournit un grand nombre de fonctions pour contrôler du texte LCD avec 4-bits et les interfaces de données 8-bit, et pour les graphiques LCD. La plus courante est les 4-bit interfaces en mode texte sur écran LCD. Ce qui suit sont les fonctions disponibles pour LCD 4-bit interface en mode texte LCD :

- Lcd_Init
- Lcd_Out
- Lcd_Out_Cp
- Lcd_Chr
- Lcd_Chr_Cp
- Lcd_Cmd

La fonction `Lcd_Init` est appelée pour configurer l'interface entre le microcontrôleur et l'écran LCD lorsque les connexions sont faites par défaut que vient d'être illustré. Le nom du port doit être spécifié en passant son adresse. Par exemple, en supposant que LCD est reliée à `PORTB` et les connexions par défaut précédentes sont utilisées, la fonction doit être appelée en tant que :

- `Lcd_Init()`;
- `Lcd_Out`

La fonction `Lcd_Out` affiche du texte à la ligne indiquée et la position des colonnes de l'écran LCD. La fonction doit être appelée avec des paramètres dans l'ordre suivant :

`Ligne, colonne, texte`

Par exemple, pour afficher du texte "Ordinateur" à la ligne 1 et la colonne 2 de l'écran LCD nous devrions appeler la fonction comme suit :

```
Lcd_Out (1, 2, "Ordinateur");
```

```
Lcd_Out_Cp
```

La fonction `Lcd_Out_Cp` affiche le texte à la position actuelle du curseur. Par exemple, pour afficher le texte "Ordinateur" à la position actuelle du curseur, la fonction devrait être appelé en tant que :

```
Lcd_Out_Cp ("Ordinateur"); Lcd_Chr
```

La fonction `Lcd_Chr` affiche un caractère à la ligne indiquée et la colonne la position du curseur. La fonction doit être appelée avec les paramètres dans le l'ordre suivant :

`Ligne, colonne, caractère`

Par exemple, pour afficher caractère "K" à la ligne 2 et la colonne 4 de l'écran LCD nous devrions appeler la fonction de :

```
Lcd_Chr (2, 4, 'K');
```

```
Lcd_Chr_Cp
```

IV.8. Porteuse ISIS Professional:



Fig. IV.5: interface de logiciel Isis porteuse.

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 [1].

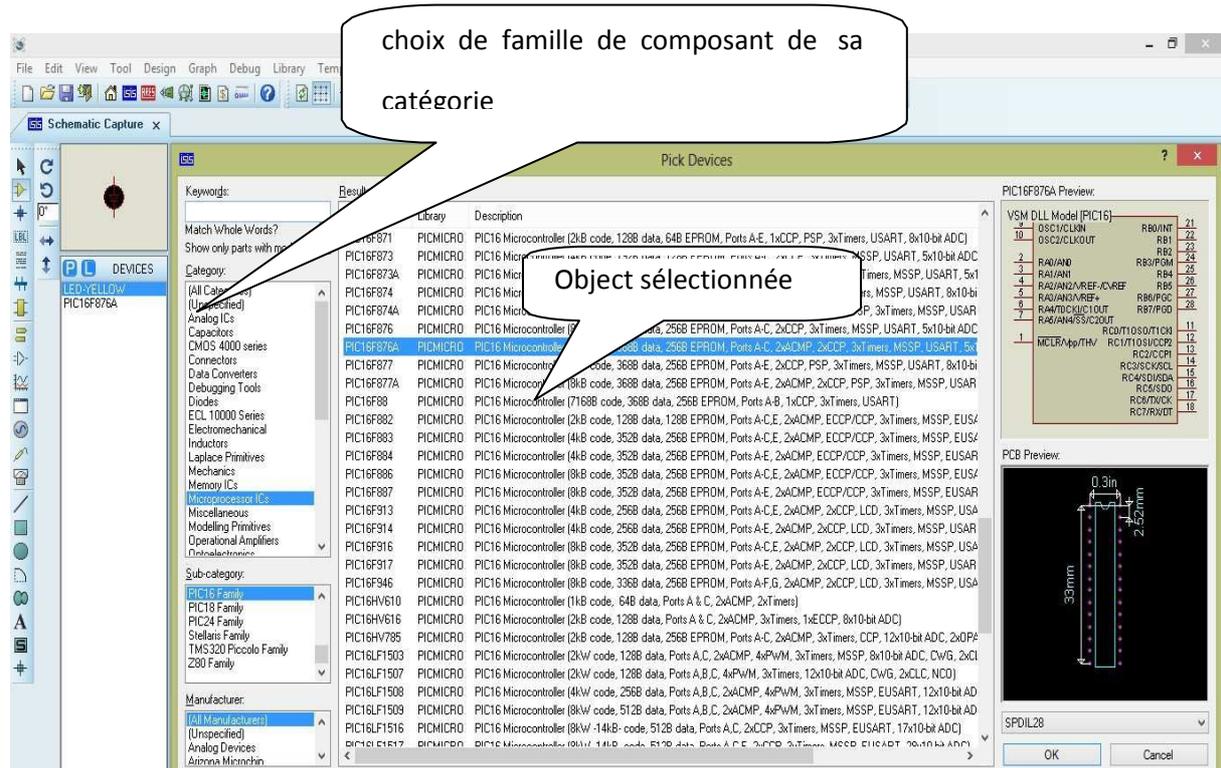


Fig. IV.6: Interface de sélection de composants de isis porteuse.

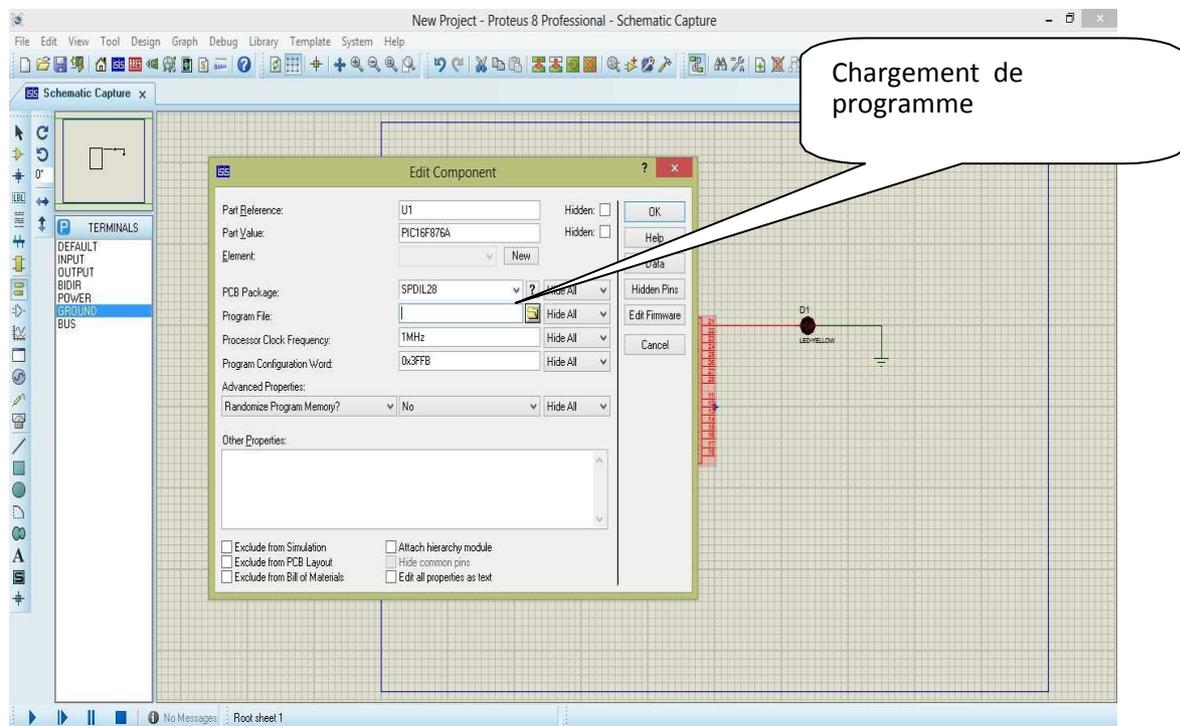
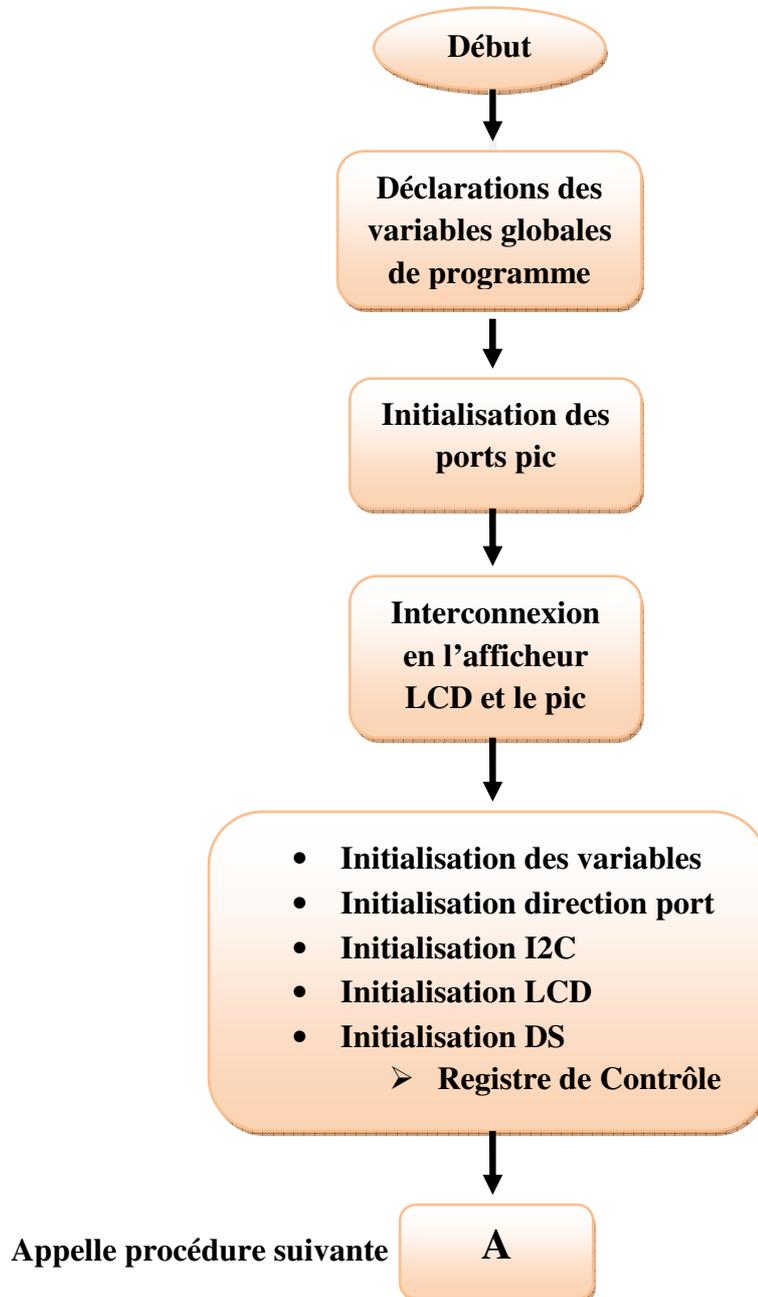
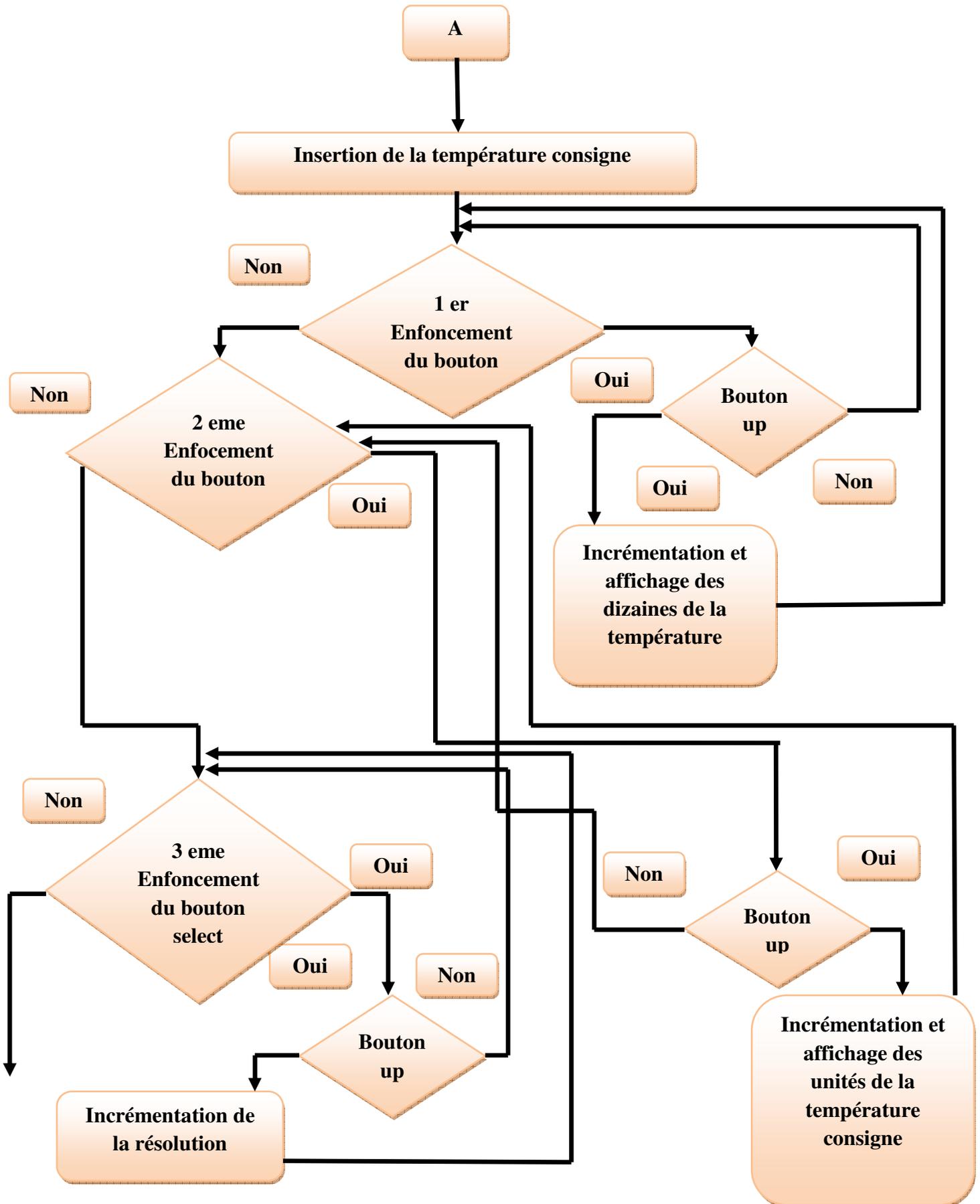
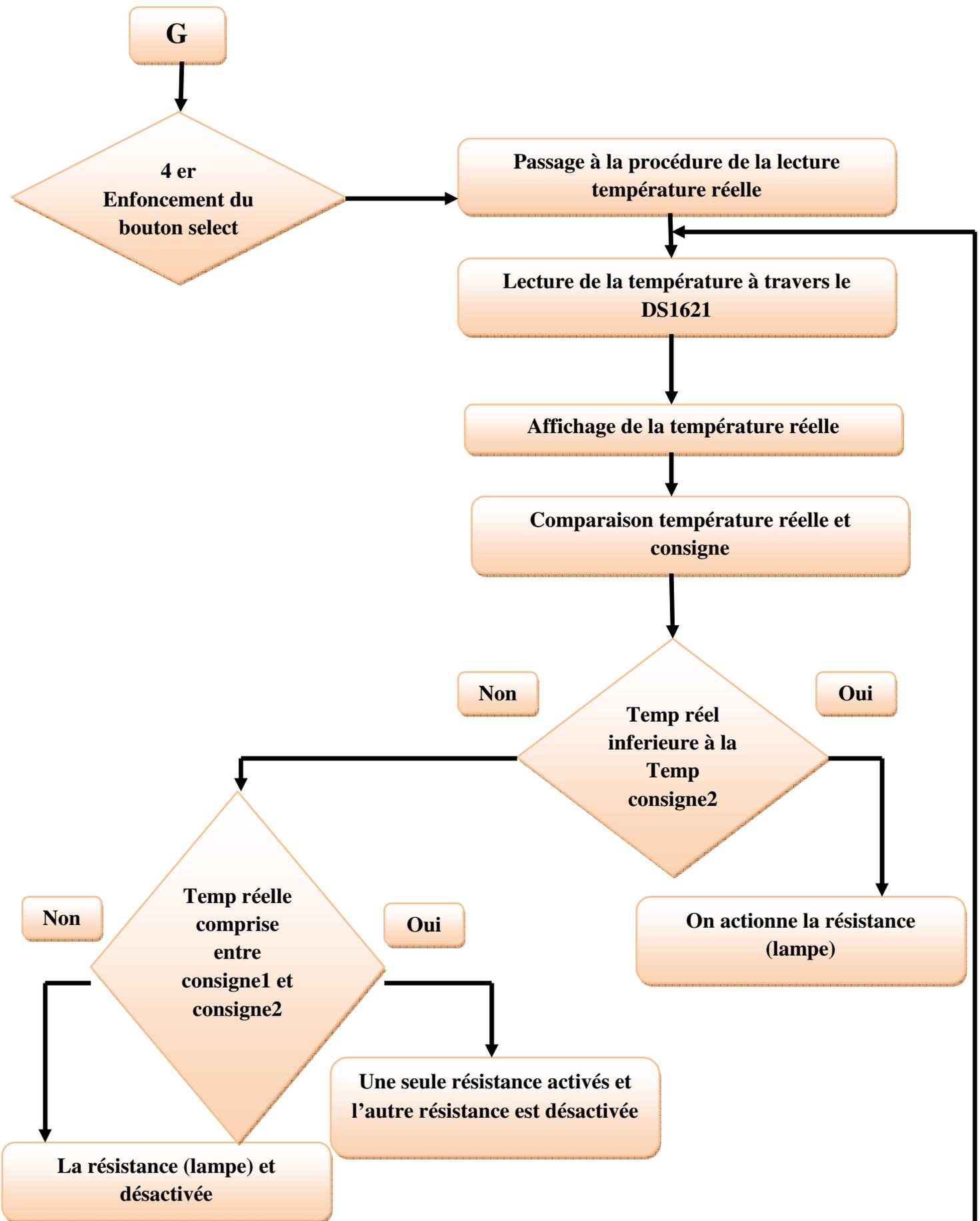


Fig. IV.7: Interface de charger programme.

IV.9: Organigramme générale:







IV.10.Conclusion:

A Travers ce chapitre nous avons montré comment programmer le pic avec langage c et éditeur mikroC et aussi la simulation avec logiciel proteus isis ,puis nous avons donner les organigrammes des différents procédures qui a permis de nous mettre en bonne vue pour le dernier étape de notre travail.

Chapitre V :

La réalisation du

circuit

V.1.Introduction:

Une carte électronique est un ensemble de composants tel que des résistances, condensateurs ou circuits intégrés réunis sur une plaque de manière à former un circuit destiné à un usage précis. Cela nous amène donc à nous demander : Quels sont les différentes étapes, de la conception à la fabrication, dans la réalisation d'une carte électronique. Nous étudierons tout d'abord la conception par ordinateur du circuit électronique, puis la préparation du circuit imprimé et pour terminer la mise en place et la soudure des composants.

Les PCB en abrégé "Printed Circuit Board" servent au montage de pratiquement tous les circuits électroniques : périphériques d'ordinateurs, téléphones portables, APN et même les cartes électroniques dans les machines équipées de composants électroniques telles que dans le domaine de l'électroménager ou de l'automobile [13].

Notre circuit sera dédié à la réalisation d'une couveuse électrique à base du pic 16F876a et un capteur de température DS1621.

V.2.Circuit imprimé:

Un circuit imprimé ou PCB est un support, en général une plaque, permettant de maintenir et de relier électriquement un ensemble de composants électroniques entre eux, dans le but de réaliser un circuit électronique complexe. On le désigne aussi par le terme carte électronique.

Il est constitué d'un assemblage d'une ou plusieurs fines couches de cuivre séparées par un matériau isolant. Les couches de cuivre sont gravées par un procédé chimique pour obtenir un ensemble de pistes, terminées par des pastilles. Le circuit imprimé est souvent recouvert d'une couche de vernis coloré qui protège les pistes de l'oxydation et d'éventuels courts-circuits.

Les pistes relient électriquement différentes zones du circuit imprimé. Les pastilles, une fois perforées, établissent une liaison électrique, soit entre les composants soudés à travers le circuit imprimé, soit entre les différentes couches de cuivre. Dans certains cas, des pastilles non perforées servent à souder des composants montés en surface [13].

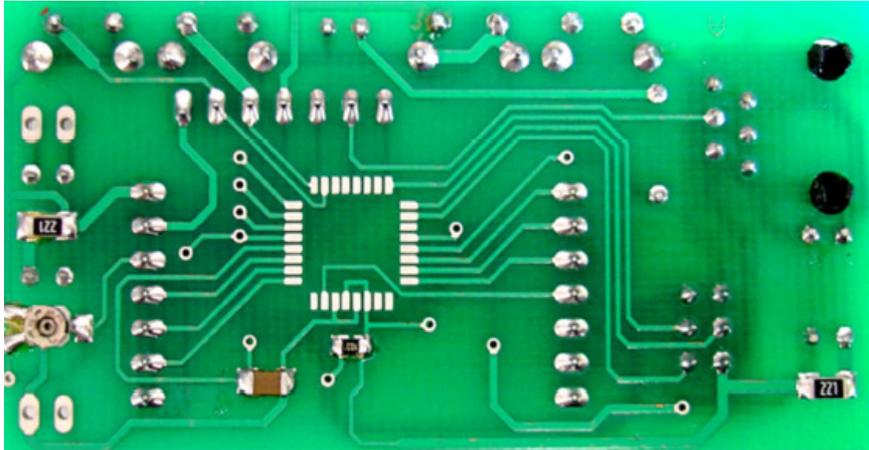


Fig. V.1: exemple d'un circuit imprimé.

V.3.Principe:

Les montages électroniques se réalisent toujours sur un circuit imprimé, c'est-à-dire un support isolant sur lequel des pistes de cuivre réalisent la quasi totalité des connexions entre les composants qu'il supporte. Le matériau utilisé est presque toujours du verre époxy c'est-à-dire un isolant à base de fibres de verre compressées et imprégnées de résine époxy. Une pellicule de cuivre très fine ($< 100 \mu\text{m}$) est fortement collée sur le support [14].

V.4.Tracée du circuit:

Le traitement consiste à supprimer le cuivre là où il n'y a pas de liaison à assurer et à le laisser au contraire là où on doit relier les composants.

Le cuivre est supprimé par attaque chimique, généralement au perchlorure de fer. L'action de supprimer le cuivre par attaque chimique s'appelle la gravure. Toute la difficulté réside dans le fait qu'il faut protéger le cuivre là où il doit rester. On peut utiliser un feutre spécial avec une encre résistante mais il faut alors tracer le dessin des pistes à la main, ce qui ne peut pas convenir pour les circuits imprimés compliqués [6].

V.5.Utilisation d'un typon:

La méthode la plus utilisée aujourd'hui pour réaliser un circuit imprimé consiste à utiliser un typon représentant le tracé des pistes que l'on voudra faire apparaître sur le circuit.

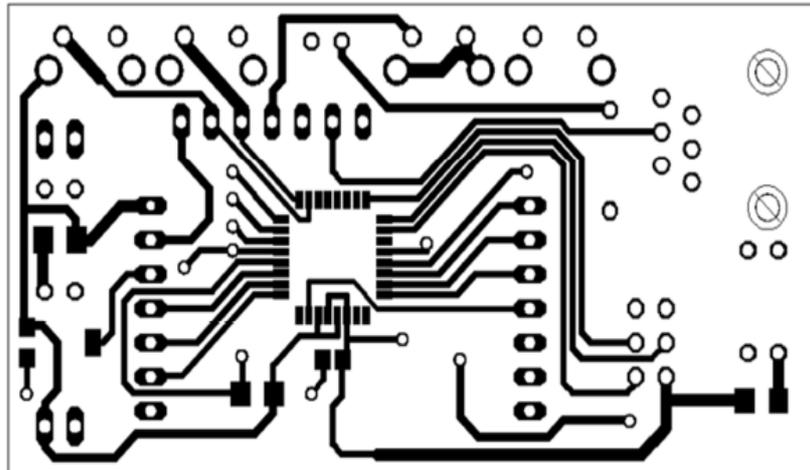


Fig. V.2:exemple d'un typon.

Ce typon pourra être dessiné à la main sur une feuille de papier calque, ou tracé sur ordinateur avec un logiciel spécialisé puis imprimé sur un transparent. L'avantage d'utiliser un typon est double : on pourra facilement fabriquer plusieurs circuits imprimés identiques « à la chaîne », et il est possible de réaliser des circuits imprimés complexes, possédant beaucoup de composants, et demandant une grande précision [6].

V.6.La conception sur ordinateur :

Une fois le besoin analysé et le cahier des charges validé la première grande étape dans la réalisation d'une carte électronique est la conception et la simulation des différentes fonctions de celle-ci. Il existe de nombreux logiciels de CAO qui nous permettent de réaliser ces simulations facilement. On utilise d'abord des outils de simulations fonctionnelles et électriques. Comme on peut le voir ci-dessus on dessine avec le logiciel tracé de ci le schéma électrique en utilisant les bibliothèques de composants incluses dans ceux-ci. Ainsi nous pouvons tester le comportement du circuit grâce aux modes de simulations proposé par ces différents outils de CAO [6].

V.6.1.Présentation du logiciel TCI:

Ce logiciel est conçu pour dessiner manuellement des circuits imprimés, et sortir le dessin sur imprimante, sur papier ou sur transparent. (Fini les rubans et pastilles à coller) La dimension du circuit est donc limitée à une feuille A4 soit 270x195mm (zone utile) [7].



Fig. V.3:interface du logiciel TCI.

V.6.2.les deux grandes parties du TCI:

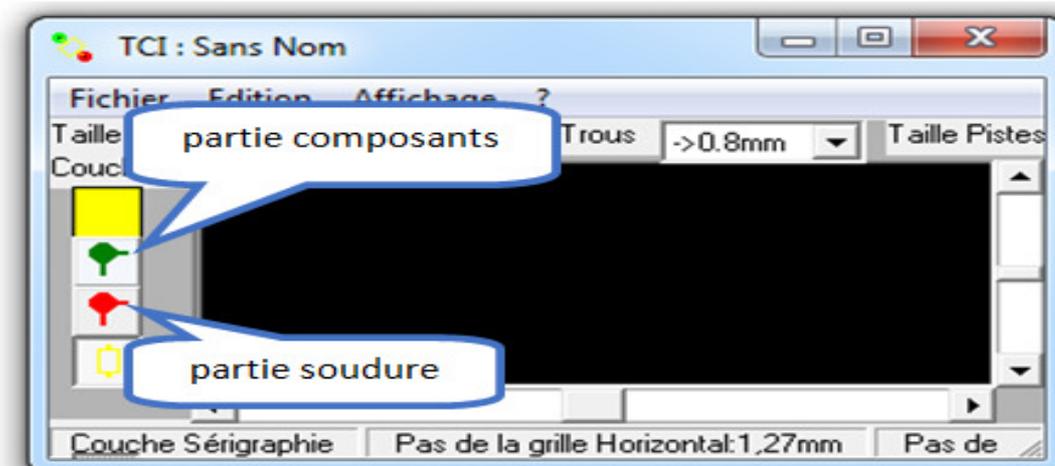


Fig. V.4: les deux parties du TCI.

V.6.3. Créer un nouveau schéma électronique en TCI:

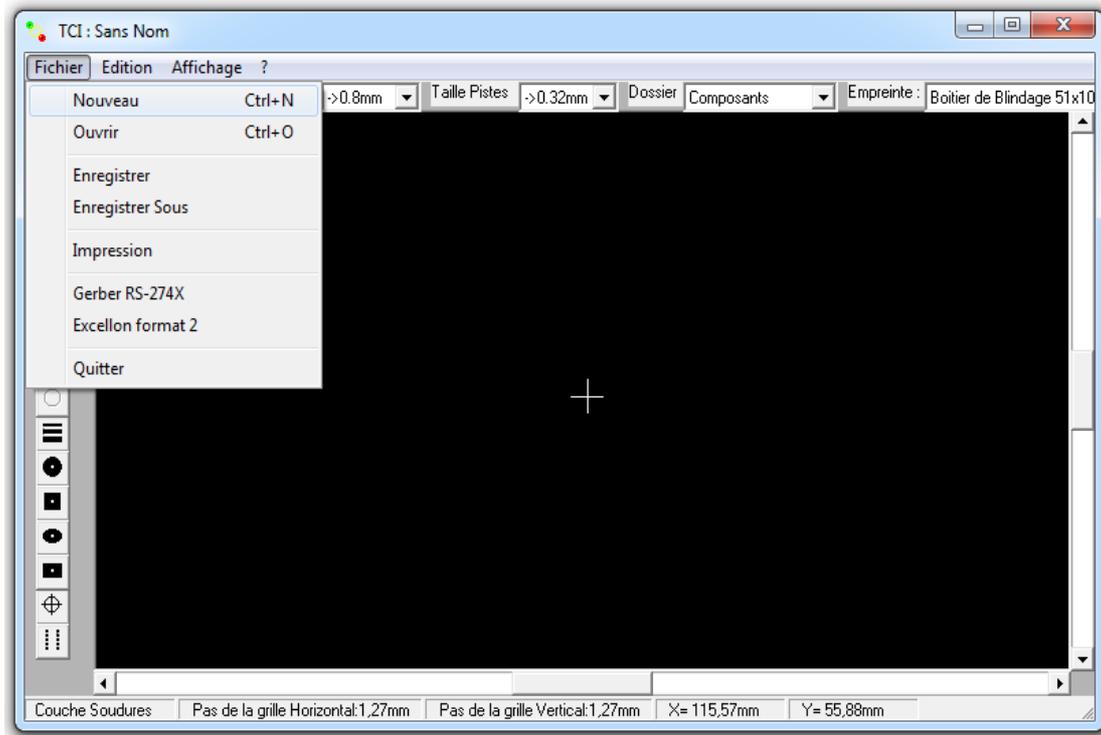


Fig. V.5: la création d'un nouveau schéma.

V.6.4. Etapes de tracé schéma avec logiciel TCI :

Le carré sur la gauche de l'écran doit rester rouge (nous restons sur la couche soudure) en limitant également les dimensions de notre carte. et nous choisissons par la suite la taille des pistes pastilles et trous.

Les liaisons entre les différents composants sont les traits rouges (pistes), les pastilles représentent l'emplacement où seront soudées les différentes pattes des composants

L'objectif est donc d'obtenir le chemin de ces pistes grâce aux fonctions de routage des logiciels.

Nous observons clairement le chemin emprunté par les pistes sur la plaque. Le routage est régi par des règles rendant sa réalisation parfois plus complexe qu'il n'y paraît. Par exemple la largeur des pistes ou la distance entre celles-ci [7].

Une fois le schéma saisi, il faut générer une Netlist : Liste des composants utilisés et de leurs connections.

- Potentiomètre 10k Ω .
- Résistance 1k Ω (3).
- Résistance 2.2k Ω (2).
- Résistance 3.2k Ω (2).
- Resistance 8.2 Ω (3).
- Condensateurs 100 μ f (2).
- Condensateurs 10 μ f.
- Condensateurs 100 nf (2).
- Condensateur 22 pf (2).
- Un microcontrôleur pic 16F876A.
- Transistor (2N2219) (2).
- Led (vert).
- Diode (1N4007) (3).
- Régulateur 5v:7805.
- DS1621.
- Quartz 4 MHZ.
- Buzzer 5V (speaker).
- Relais type nais ds2e-s-dc12v.
- Bouton poussoir(2).
- Afficheur alphanumérique 2x20.
- Support dil 28 broche.
- Jak alimentation femelle.
- Bornier(3 entrée) (2).
- Fusible 1A.

V.6.5. Le tracé de notre circuit en TCI:

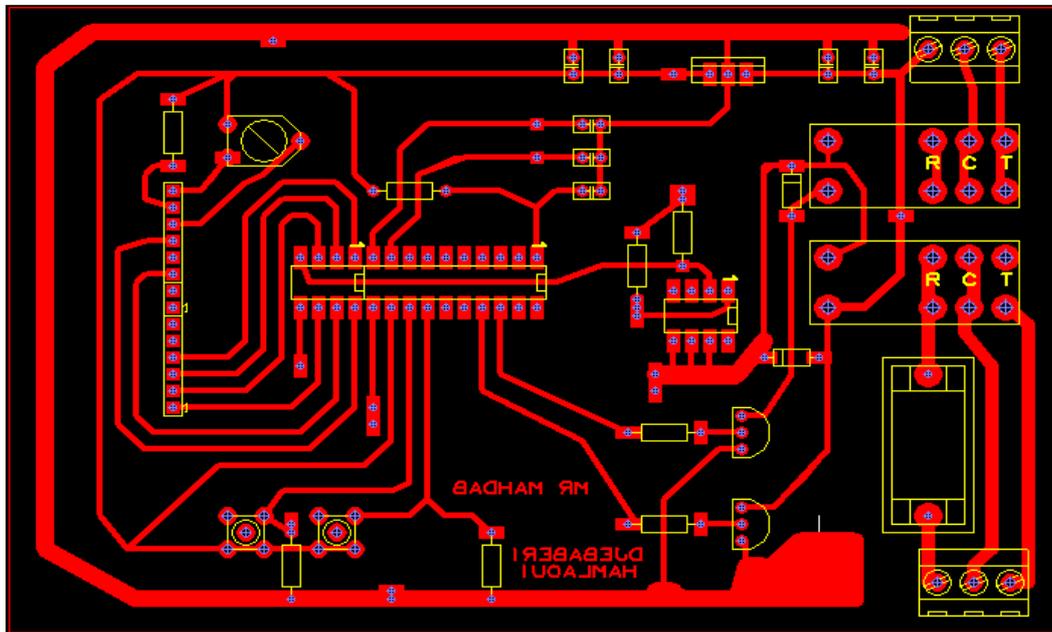


Fig. V.6: schéma en TCI de notre PCB.

V.6.6. Transformation du TCI vers le typon:

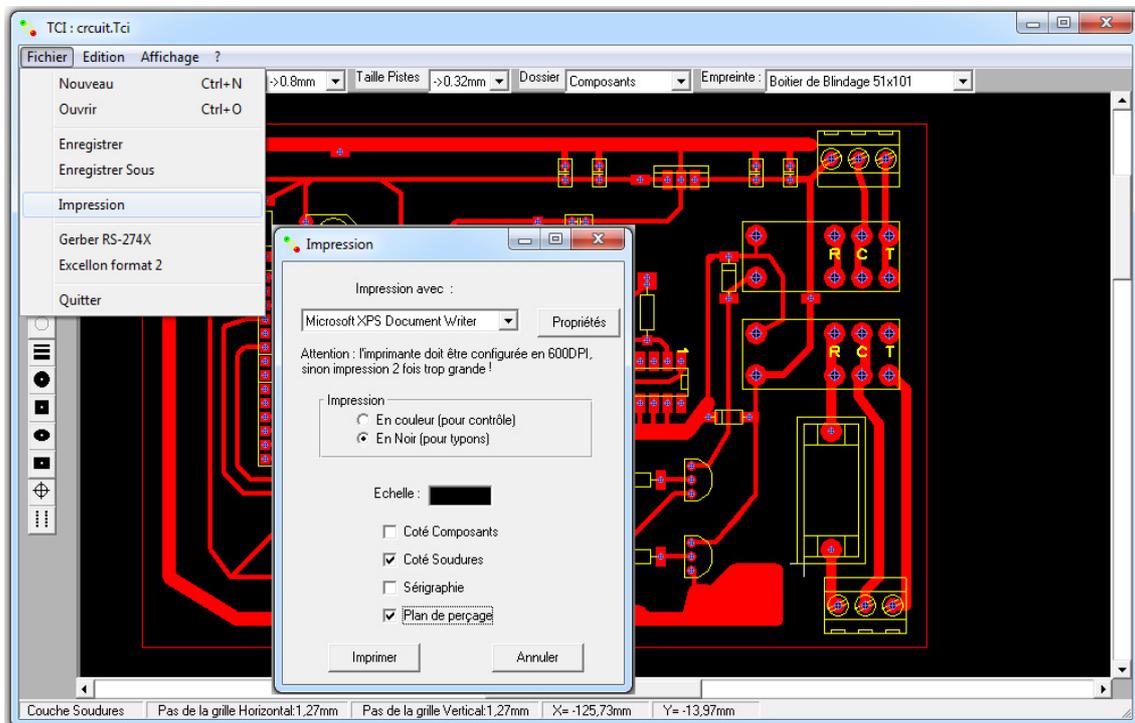


Fig. V.7: Transformation vers le typon.

V.6.7.Typon:

Et voilà notre typon qui ne laisse apparaître que les pistes que l'on veut imprimer sur notre support transparent (calque, transparent...).

Le plus souvent, le typon est imprimé à l'aide d'une imprimante Laser, sur du papier calque ou sur transparent. Plus le support est transparent et plus l'encre est opaque, meilleur sera le résultat.

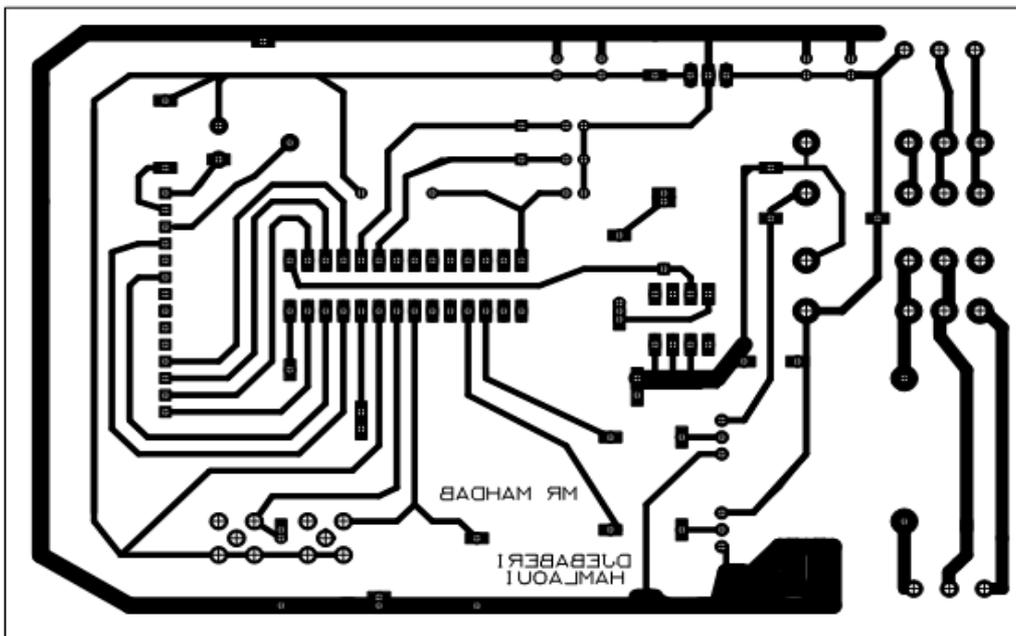


Fig. V.8:Typon.

Pour « transférer » le tracé du typon sur la plaque du circuit imprimé, on utilise des plaques de cuivre photosensibles. Les plaques photosensibles sont recouvertes d'une couche protectrice photosensible dure et très adhérente, de couleur bleu-violet et d'une épaisseur de 2,5 microns. Cette couche protectrice protégera le cuivre lors de la gravure. La couche protectrice est photosensible, ce qui signifie qu'elle est détruite au contact des rayons ultraviolets. Pour protéger le cuivre seulement aux endroits contenant des pistes, on va exposer la plaque photosensible aux rayons ultraviolets, après l'avoir recouverte avec le typon : ainsi, la couche protectrice photosensible sera brûlée et éliminée par les rayons ultra-violet à l'extérieur des pistes (là où le typon est transparent) et demeurera intacte sur les pistes (là où le typon est opaque) [6].

V.7. Transfert du circuit sur la plaque de cuivre:

Le support des cartes électroniques est un circuit imprimé. Celui-ci est une plaque en époxy à la surface de laquelle des pistes en cuivre sont gravées. Il ne faut pas confondre « circuit imprimé » qui désigne la plaque et les pistes sur lesquelles seront soudés les composants et « circuit électronique » qui désigne l'ensemble de la carte électronique.

Voici les étapes que nous avons suivies pour fabriquer notre circuit imprimé en utilisant un typon:

V.7.1. Coupe:

Avant de pouvoir imprimer le schéma sur notre plaque de cuivre, nous allons devoir nous occuper un peu de celle-ci. Premièrement, on utilise le schéma pour tracer un rectangle de la taille voulue sur notre plaque de cuivre. Nous pouvons tracer ce rectangle à l'aide d'un marqueur indélébile.

Nous découpons ensuite notre plaque de cuivre suivant ces lignes à l'aide d'une cisaille, la plaque résultante aura les dimensions du futur circuit imprimé.



Fig. V.9: cisaille coupante.

V.7.2.Préparation:

Pour que l'insolation puisse s'effectuer, il faut retirer l'adhésif opaque qui protège la couche photosensible des rayons ultra-violets de la lumière ambiante. Une fois l'adhésif retiré, la plaque pré-sensibilisée sera placée dans l'insoleuse, avec le typon côté cuivre (en vérifiant bien son sens).

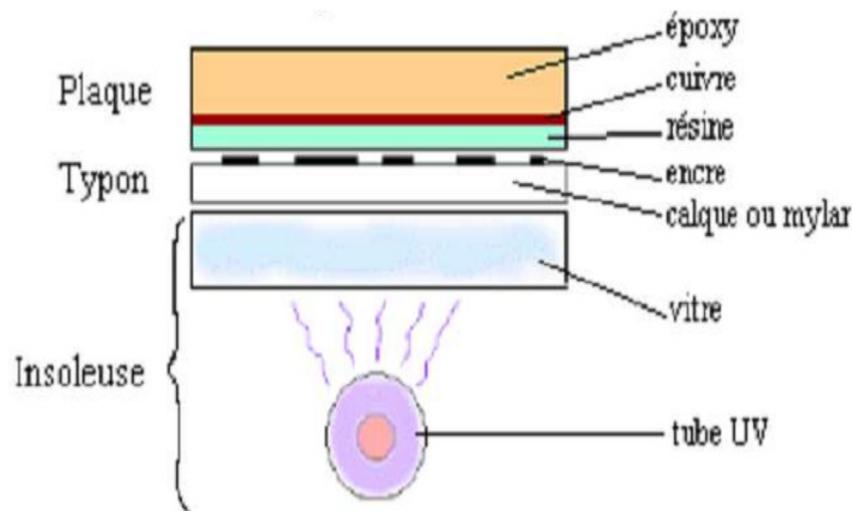


Fig. V.10: Placement des différents éléments pour l'insolation.

Une insoleuse est principalement constituée de puissants tubes néon UV et d'une vitre totalement transparente sur laquelle on déposera la plaque.

V.7.3.Insolation:

Cette opération, appelée insolation, sert à transférer le dessin sur la plaque. Les tracés à l'encre opaque sur le typon empêchent les ultraviolets d'atteindre la laque photosensible là où le cuivre devra être conservé.

Nous fermons le capot protecteur de l'insoleuse et nous réglons aussi la minuterie sur 2 min 30 sec. Les ultra-violets émis par les tubes de l'insoleuse vont détruire la couche photosensible aux endroits non protégés par le tracé du typon. Lorsque la minuterie s'arrête, et après avoir ouvert le capot protecteur de l'insoleuse et sortir la plaque pré-sensibilisée. Nous devons pouvoir observer le tracé des pistes imprimé sur la résine.



Fig. V.11:Le tracé des pistes sur la résine.

V.7.4.Révélation:

Nous trempons par la suite la plaque pré-sensibilisée dans le bain révélateur en suivant bien les consignes de sécurité. Si le révélateur est récent ou très actif, l'opération risque d'être très courte. Et la nous observons bien l'action du produit qui désagrège la couche photosensible brûlée par les ultra-violets, ne laissant que le tracé des futures pistes sur la couche de cuivre.

Le révélateur est un produit chimique que l'on peut le fabriquer soi-même, puisqu'il s'agit d'une simple solution de soude caustique à 7g/l comme le Destop (produit pour déboucher les canalisations).



Fig. V.12:révélation.

V.7.5.Rinçage:

La couche de cuivre va progressivement apparaître autour des pistes qui sont encore protégées par la résine. Une fois la plaque révélée elle est sortie du bac et rincée à l'eau. Maintenant que nous avons fait apparaître la couche de cuivre autour des pistes protégées par la résine, il faut la détruire [5].

V.7.6.Gravure:

Notre plaque est plongée dans un bac à graver qui contient un produit acide : le perchlorure de fer. Cet acide va dissoudre le cuivre autour des pistes protégées par la résine. Le Perchlorure de Fer suractivé est un liquide de couleur marron très foncé. On l'utilise pour graver les circuits imprimés car il a la particularité de détruire (par réaction chimique) tout le cuivre qui n'est pas recouvert de résine photosensible. Cela a pour conséquence de ne laisser sur la platine que les pistes qui nous intéressent.

Dans cette phase nous plaçons la plaque pré-sensibilisée dans la graveuse en suivant bien les consignes de sécurité et en Réglant la minuterie sur 20 min.

Le perchlorure de fer décompose le cuivre qui n'est pas protégé par la couche photosensible : le cuivre sera donc éliminé sur toute la plaque sauf sur les pistes tracées sur le typon.



Fig. V.13:gravure.

Le perchlorure est encore actif, toxique et salissant. Il faut donc rincer le circuit imprimé à l'eau.

A l'aide d'un chiffon imbibé d'alcool (dissolvant ou bien acétone), nous nettoyons le circuit imprimé jusqu'à faire disparaître la couche photosensible encore présente sur les piste, et devenue maintenant inutile.

En respectant le temps de chaque opération (insolation ,révélation et gravure) et en s'adaptant avec les conditions et le matériel disponible nous aurons sûrement notre circuit imprimé prêt à l'utilisation.

Bien que ce ne soit pas très voyant sur la photo, notre plaque est maintenant criblé de trous à travers lesquels nous allons faire passer nos composants [6].

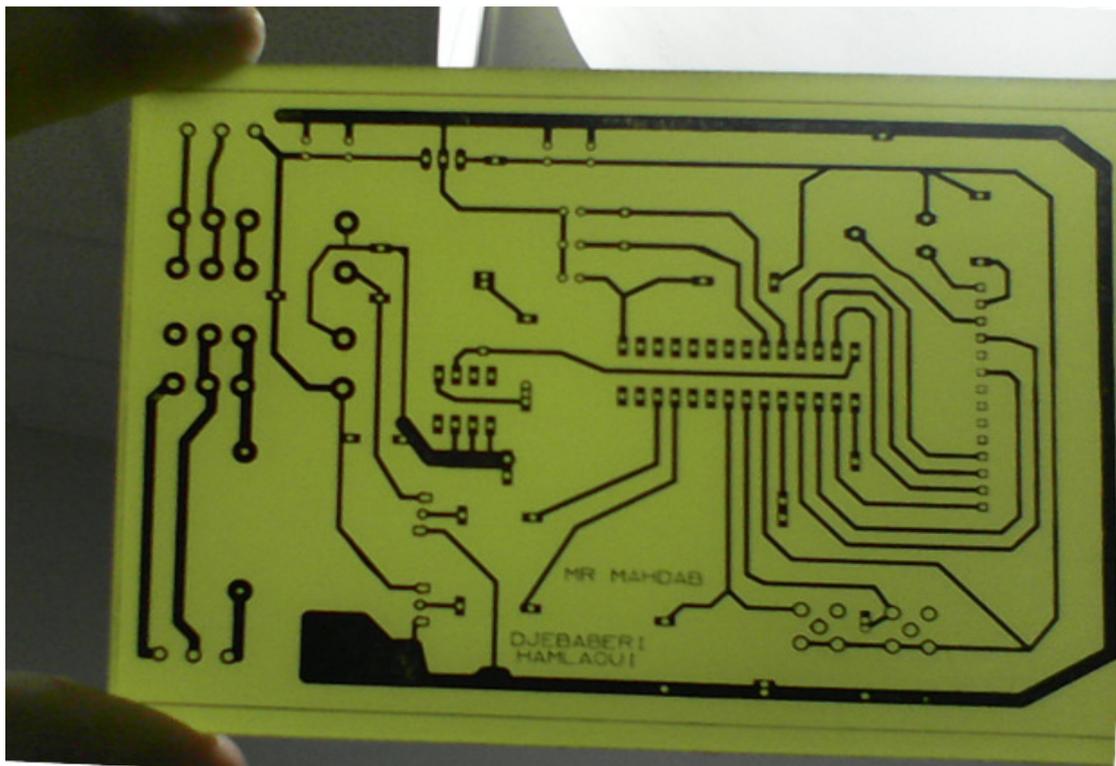


Fig. V.14:Circuit imprimé.

V.8. Implantation des composants et soudure:

V.8.1. Perçage:

Maintenant que notre circuit est imprimé sur la plaque d'époxy, nous allons devoir le percer pour pouvoir y insérer les composants afin de les souder.

Pour le perçage nous utilisons une micro-perceuse montée sur bâti, avec des forets de 0,8 ou 1 mm suivant le diamètre des fils de sortie des composants.

Certains trous pourront nécessiter des forets de diamètre 3 mm pour la fixation du circuit par exemple [6].

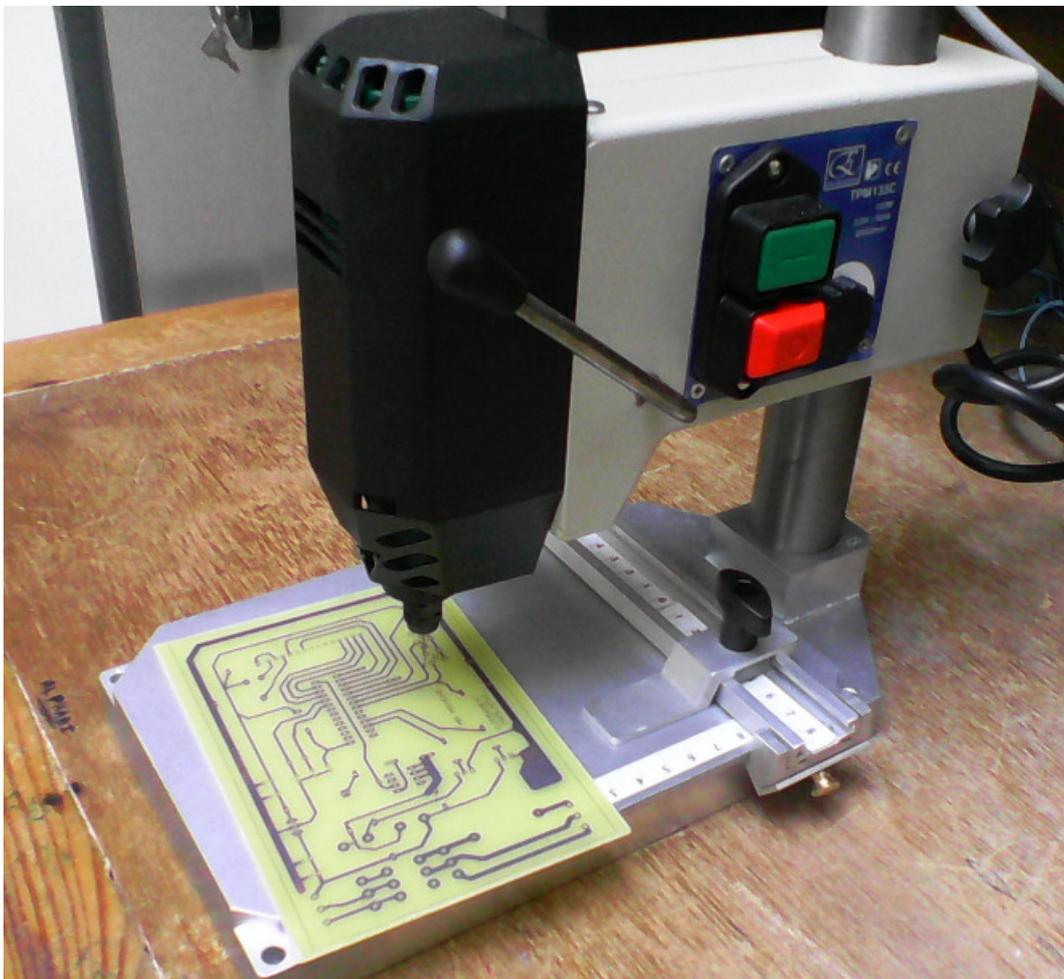


Fig. V.15:perceuse.

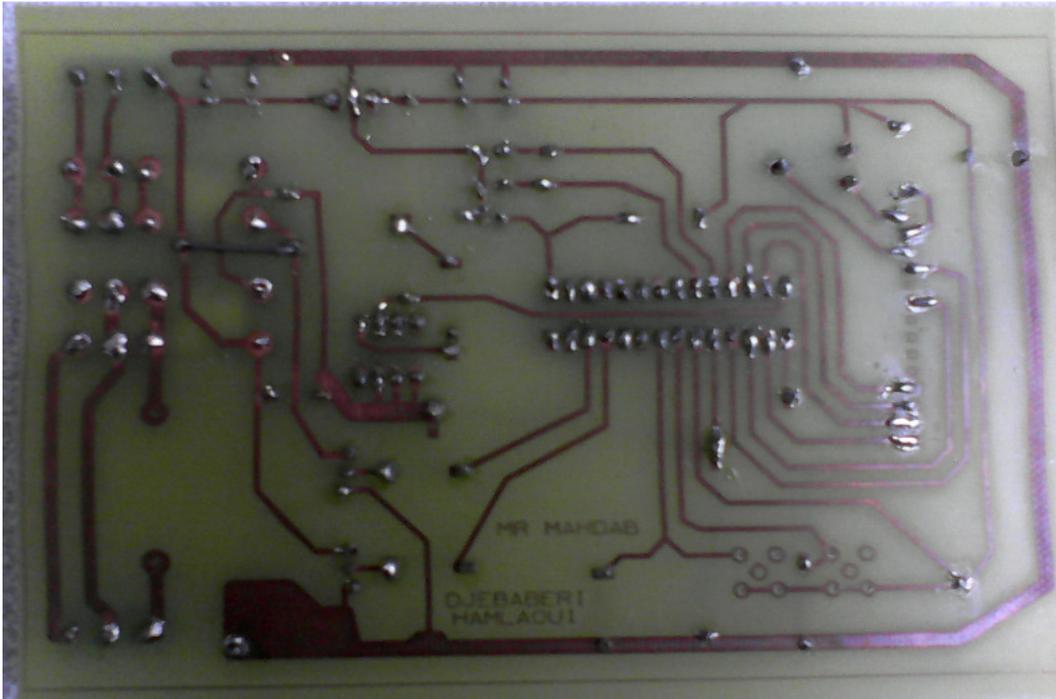


Fig. V.18:Circuit imprimé vu du côté soudures.

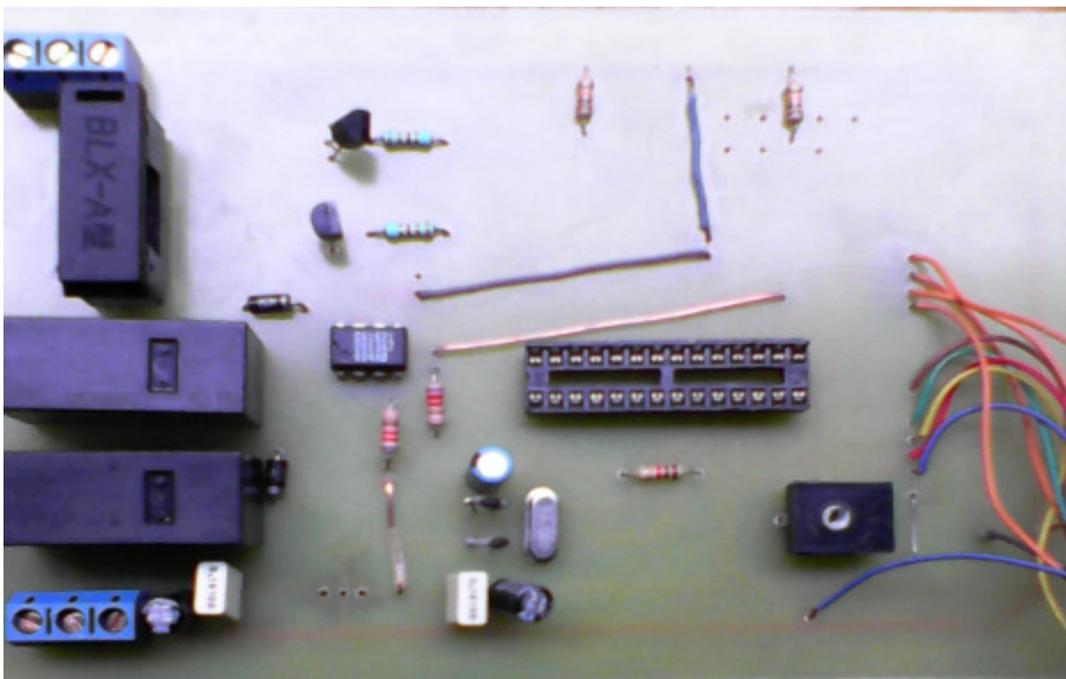


Fig. V.19:Circuit imprimé vu du côté composants.

V.9.Conclusion:

En conclusion nous avons étudié toutes les étapes dans la réalisation d'une carte électronique de sa conception jusqu'à sa fabrication. Pour cela nous avons étudié l'étape de conception grâce à des logiciels spécialisés permettant de réaliser facilement des schémas fonctionnels et électriques puis de gérer le routage du circuit. Ensuite nous avons détaillé les différentes étapes dans la fabrication du circuit imprimé, à savoir l'impression du typon, l'insolation, la révélation puis la gravure ainsi que les différentes techniques pour mettre en œuvre cette procédure de fabrication.

Conclusion générale

Conclusion générale :

Notre projet a porté sur l'étude et la réalisation d'une couveuse. Il nous a mis dans une vraie situation de réalisation pratique, qui nous a permis de comprendre que plusieurs étapes sont nécessaires à savoir.

Même si sa finalité peut paraître simple à réaliser, la mise en œuvre de cette conception nous a demandé beaucoup d'attention et de soins. Comme cette conception est basée sur l'acquisition de la température à partir d'un capteur de température fonctionnant avec le bus I2C, ceci a rendu la conception du programme un peu pénible, ajouté à cela le côté comparaison de la température réelle lue à partir du capteur a deux consignes.

En générale nous dirons que par le biais de ce travail nous avons eu également l'opportunité de se familiariser avec des logiciels tels que PROTEUS (logiciel de simulation) mikroC (Editeur/ compilateur du code C pour les pics) TCI pour le circuit imprimé.

En fin, nous estimons que ce travail a été d'une grande utilité pour nous et qu'il sera bénéfique pour les promotions à venir. On espère que cette étude servira d'appui pour une éventuelle amélioration et qu'elle sera caractérisée comme une approche expérimentale.

Bibliographies

- [1] V. TOURTCHINE, mémoire de fin d'étude Université de Boumerdes : Programmation. en mikro C Application pour les microcontrôleurs de la famille PIC, promotion 2012.
- [2] Jamart J-F, Advanced PIC microcontroller projects in C-Dogan Ibrahim-newnes.
- [3] H.E.L. – I.S.E.T. Laboratoire électronique.
- [4] PASKAL MAYEUX apprendre la programmation des pic .dunod.
- [5] CHRISTIAN TAVERNIER programmation en c des pics.
- [6] ROGER BOURCERON schéma et circuit électronique.
- [7] ALBERT PAUL MALVIND principes de l'électronique.
- [8] ALIBI ELMEHDI Jawadi Sami, Conception et réalisation d'un enregistreur de données.
- [9] apprendre l'électronique en pratiquant zéro niveau 1
- [10] M. BERNARD Le microcontrôleur PIC 16F876A Lycée LACHENAL édité le 02/11/2008.
- [11] PIC 16F87XA Data Sheet 28/40/44-Pin Enhanced Flash Microcontrollers.
- [12] PIC 16F87XA Data Sheet 28/40/44-Pin.
- [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

Webiographie:

- [15] www.lelectronique.com
- [16] www.techno-flash.com
- [17] www.abcelectronique.com
- [18] www.lewebelectronique.free.f

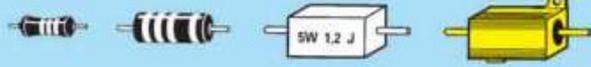
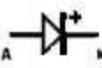
SYMBOLE	ABR.	DESCRIPTION	COMMENT ILS SE PRÉSENTENT
	F	FUSIBLE	
	R	RÉSISTANCE	
	P ou POT.	POTENTIOMÈTRE	
	C	CONDENSATEUR CERAMIQUE ou POLYESTER	
	C	CONDENSATEUR CHIMIQUE	
	D	DIODE SILICIUM	
	LED	DIODE LED	
	T	TRANSISTOR NPN	
	TR ou TRIAC	TRIAC	
	S	INTERRUPTEUR	
	RL	RELAIS 1 CIRCUIT	
	QZ ou XTAL	QUARTZ	
	BZ	BUZZER	

Fig.1 : Tableau de quelques composants utilisés.

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 reference.

PIC	FLASH	RAM	EEPROM	I/O	A/D	Port //	Port Série
16F870	2K	128	64	22	5	NON	USART
16F871	2K	128	64	33	8	PSP	USART
16F872	2K	128	64	22	5	NON	MSSP
16F873	4K	192	128	22	5	NON	USART/MSSP
16F874	4K	192	128	33	8	PSP	USART/MSSP
16F876	8K	368	256	22	5	NON	USART/MSSP
16F877	8K	368	256	33	8	PSP	USART/MSSP

Tableau.2: Different circuit de la famille 16f8XX.



PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

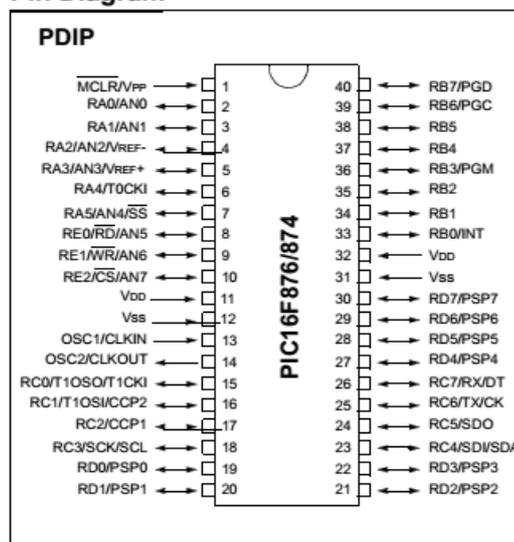
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature
ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 μ A typical @ 3V, 32 kHz
 - < 1 μ A typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during SLEEP via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master
mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) 8-bits wide, with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

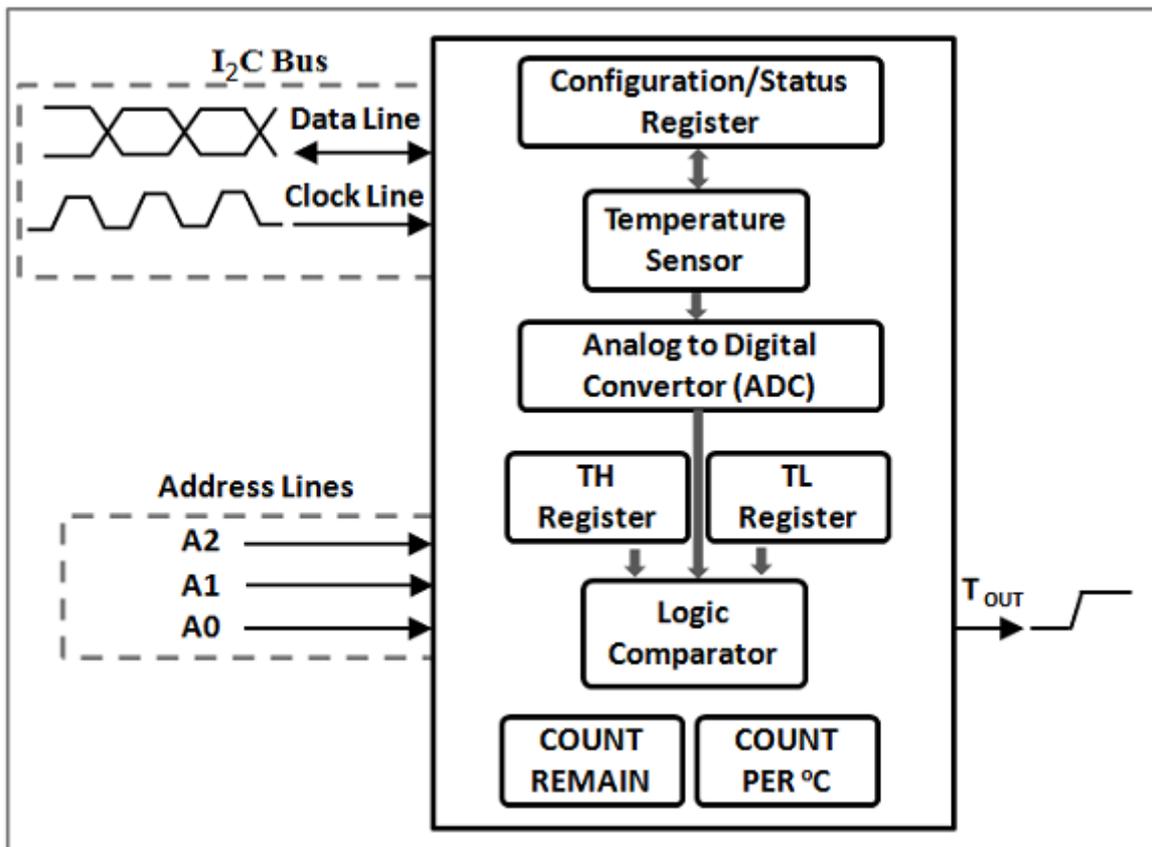


Fig.1: Functional diagram of DS1621 temperature sensor.

APPLICATIONS

1) EXEMPLE de LECTURE I2C :

Le DALLAS 1621 :

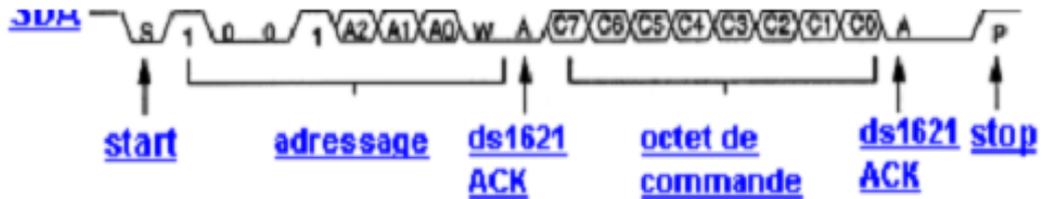
L'utilisation du thermomètre digital Dallas 1621 donne un exemple de lecture de plusieurs octets en I2C.

La trame envoyée doit commencer par l'adressage du boîtier avec un accès en écriture, suivi du code de début de conversion.

Il faut laisser un bref délai pour que le thermomètre effectue une première conversion.

Ensuite il faut commencer une nouvelle communication en s'adressant au boîtier toujours en écriture. L'octet suivant contient le code de la fonction de lecture.

Une fois ces opérations effectuées on crée un "restart" (condition de stop immédiatement suivie d'une condition de start), on s'adresse à l'esclave en mode lecture et on attend la réponse.



ds1621 ACK: acquittement de l'esclave

Cette réponse est sur deux octets. On effectuera donc un acquittement après réception du premier octet et après réception du deuxième octet un non acquittement suivi d'une condition de stop.

