République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane MIRA- Bejaia

Faculté de Technologie

Département de Génie Electrique

Mémoire de fin d'étude

En vue de l'obtention du diplôme MASTER en Electronique

Option : *Instrumentation*

Thème

Etude et réalisation d'un calendrier numérique réglable à base d'un microcontroleur PIC16F877

Réalisé par :

IGUI Lydia
REMILA Hadjer

Encadré par :

Mr HANFOUG Salah

Examiné par :

Mr TAFININE

Mme IDJDARENE

PROMOTION
2019/2020

Remerciement

Tout d'abord nous tenons à remercier dieu, le tout puissant, de nous avoir guidé dans la réalisation de notre travail.

Nous tenons à exprimer notre profonde gratitude et notre sincère remerciement à notre promoteur monsieur S.HANFOUG, pour son aide ces critique constructives, ses explications et suggestions pertinentes et pour la qualité de ses orientations tout au long de ce travail et pour avoir apporté tant de soins a la réalisation de ce mémoire.

Nous remercions les membres du jury qui nous font honneur en acceptant d'examiner et de juger notre travail.

Nous remercions particulièrement nos parents pour leurs soutiens inconditionnels tout au long de ces longues années d'études.

Enfin une pensée à tous nos anciens enseignant ainsi toute personne qui a contribué de prés ou de loin a la concrétisation de notre travail.

Dédicace

C'est avec une grande émotion que je dédie ce modeste travail de fin d'étude mes êtres les plus chers ;

A la mémoire de mon grand père.

A mes très chers parents qui ont fait de moi ce que je suis aujourd'hui.

A mes trois petites frères Mehdi ,Youcef et Rayane.

Au reste des membre de ma famille qui m'on vraiment soutenu durant le parcours de mon mémoire .

A ma chère binôme lydia et set son adorable petite famille.

A tout qui m'on aidé de prés ou de loin pour la réalisation de ce travail.

A tout mes amis.

Hadjer .R

Dédicaces

Je dédié ce modeste travail :
A mes chers parents qui m'ont toujours soutenu, encouragé et
Donné la force d'aller au bout de ce parcours.
A la mémoire de mes grands parents que la paix soit avec eux.
Je le dédie également à mon cher frère LAMINE et mes deux chères sœurs KATIA et TEREZA qui m'ont aidé à bien surmonter des obstacles.
Sans oublier ma binôme HADJAR et toute sa famille, son esprit d'équipe qui nous à aidé pour bien mener notre modeste
Travail.
Et sans oublier mes deux chères amies BAYA, Naoual et surtout mon cher ami AISSA qui a été toujours à mes cotés Tout au long de l'année avec ses encouragements inlassable.
A tous ceux qui me connaissent et qui ont contribué de prés ou de loin à la réalisation de ce mémoire.

LYDIA.I

Liste des abréviations

LCD **:** Liquid Crystal Display

µC **:** Microcontrôleur

PIC **:** Programmable Interface Controller.

RISC **:** Reduced Instructions Set Computer

CISC **:** Complex Instruction Set Computer

CPU **:** Central Processor Unit

EPROM **:** Erasable Programmable Read-Only Memory

EEPROM **:**Electrically-Erasable Programmable Read-Only Memory ou mémoire morte effaçable électriquement et programmable

RAM **:** Random Access Memory ou mémoire à accès direct

VDD **:** Voltage Drain Drain

VSS **:** Voltage Source Source

MCLR **:** Master Clear

CAN **:** convertisseur analogique / numérique

ALU **:** Arithmetic Logic United

TMR0 **:** Timer 0

WDT **:** Watchdog

OSC : oscillateur

Liste des figures

Liste des figures

Chapitre I :

Chapitre II :

Chapitre III:

Liste des tableaux

Chapitre I :

Chapitre II :

Chapitre III :

Sommaire

Sommaire

Chapitre II : Mise en œuvre du microcontrôleur PIC 16F877

Chapitre III : Conception et réalisation pratique

Introduction générale

Introduction générale

L'électronique de nos jours est tellement développé que les gens n'accordent plus beaucoup d'importance aux petits appareils quotidiens comme le calendrier, cet appareil peut paraître anodin et basique mais il reste tout de même un outil pour mesurer la grandeur physique la plus importante voire la plus chère sur terre : le temps.

L'idée de l'utilisation des microcontrôleurs est née de la nécessité de disposer pour certaines applications d'une commande avec des performances assez élevées. De plus, les microcontrôleurs possèdent un indéniable avantage sur la logique câblée. En effet pour modifier le fonctionnement d'une application, il suffit de modifier le programme sans refaire de câblage [22]. Les microcontrôleurs possèdent également la puissance d'un microprocesseur mais ils ont un atout en plus, du fait qu'ils possèdent des périphériques intégrés et des mémoires dans le même boitier [31].

Dans notre projet, le Microcontrôleur PIC 16F877 va gérer toutes les opérations, les différents traitements et d'assurer par conséquence les liaisons entre les différentes parties électroniques du calendrier numérique. On a choisi ce microcontrôleur parce qu'il dispose d'une mémoire flash pour stocker le programme (donc il est possible de le reprogrammer en cas de bug ou d'évolution) et d'une mémoire RAM de taille suffisante.

L'objectif de notre projet est d'étudier et réaliser un calendrier numérique réglable à travers le pin RB0 autour d'un PIC 16F877 à affichage numérique via des écrans LCD permettant d'afficher la date.

Afin de répondre aux objectifs cités précédemment, Ce manuscrit est structuré en trois chapitres :

Le premier chapitre : intitulé << Présentation et description des outils du projet >> nous avons présenté les éléments jugés importants pour la réalisation de notre travail. A cet effet nous avons présenté d'une manière générale les différents blocs nécessaires à la réalisation pratique de notre calendrier électronique, ainsi que la description des outils de conception et de simulation (logiciel proteus, mikroC) avec lesquels on a travaillé.

Le deuxième chapitre : intitulé << Mise en œuvre du microcontrôleur PIC 16F877 >> est consacré à la description de l'unité de commande et de traitement de notre système **:** Le microcontrôleur PIC 16F877, ainsi ses différentes ressources internes, et ses principales caractéristiques.

Le troisième chapitre : intitulé << Conception et Réalisation pratique >>concerne la conception et la réalisation pratique des différents blocs de notre système. On effet nous avons décrit toutes les étapes de réalisation du prototype, et décrivant la partie informatique on programmant le PIC16F877 avec le logiciel MikroC, et la simulation des différents montages par le simulateur ISIS Professionnel pour s'assurer de leur bon fonctionnement électronique et par conséquent du montage électronique du calendrier numérique, ce chapitre s'achève sur la réalisation de la partie électronique sur plaque d'essai.

Nous terminons par une conclusion générale et des perspectives.

Chapitre I

Présentation et description des outils du projet

I.1. Introduction

Avec l'évènement de ce que l'on appelle les "nouvelles technologies", l'objectif premier est de réaliser des traitements de plus en plus complexes le plus rapidement possible, en particulier le domaine d'électronique, cependant il n'est plus rare de nos jours de voir les appareils usuels remplacés par des appareils électroniques, nous pouvons penser entre autres aux calendriers numériques.

Ce projet nous permettra donc de réaliser cet appareil. La conception et la réalisation de ce système dépend de microcontrôleur PIC16F877 afin que ce dernier va gérer tous les processus dans le circuit de notre calendrier. La simulation est réalisée à l'aide d'un programme élaboré sous ISIS et la programmation sous logiciel micro C.

I.2. Description matériel

I.2.1. Afficheur LCD

Les études sur la polarisation de la lumière ont trouvé de nombreuses applications depuis longtemps mais ce sont les plus récentes qui sont entrées dans notre environnement quotidien, il s'agit des afficheurs à cristaux liquides. Ces derniers sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement, ils sont devenus indispensables dans les systèmes techniques qui nécessitent l'affichage de paramètres de fonctionnement. Grâce à la commande par un microcontrôleur ces afficheurs permettent de réaliser un affichage de messages aisés. Ils permettent également de créer ses propres caractères [1].

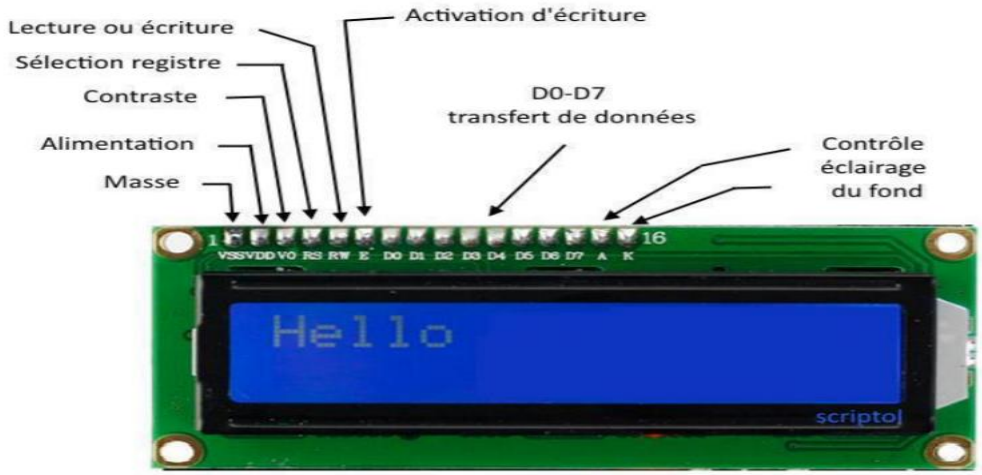


Figure I.1: Afficheur a cristaux liquide.

I.2.1.1. Définition de l'afficheur LCD

Un cristal liquide est produit de la chimie organique, qui possède les propriétés optiques des cristaux solides alors qu'il est lui-même liquide. L'afficheur LCD utilise la polarisation de la lumière, grâce à des filtres polarisants, et à la biréfringence de certains cristaux liquides en phase nématique (phase intermédiaire entre liquide et solide). Du point de vue optique, l'afficheur à cristaux liquides est un dispositif passif (il n'émet pas de la lumière) ; il doit donc être éclairé. D'abord disponible en monochrome et en petite taille, il est utilisé dans les calculatrices et les montres, du fait de sa faible consommation électrique. Il permet actuellement d'afficher en couleurs dans des dimensions dépassant le mètre de diagonale. Il a pu remplacer le tube cathodique dans la plupart des applications, sauf en très haute définition, lorsque la palette de couleurs doit être précise et fidèle [1].

Il existe deux types d'affichage :

- Par pixels
- Par segments

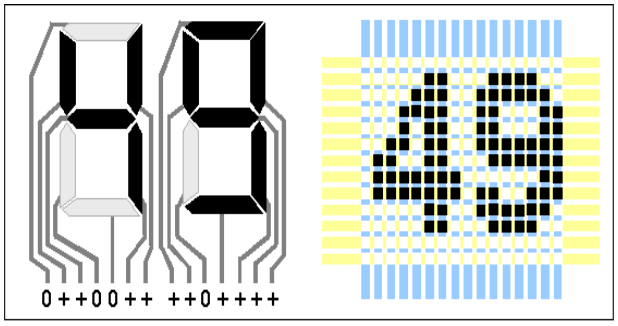


Figure I.2 : Affichage par segments et par pixels.

I.2.1.2. Les bases des afficheurs LCD

LCD est l'acronyme de Liquid Crystal Display (en anglais), ce qui signifie en français écran à cristaux liquides [1]. Par opposition à un afficheur à LED (comme les afficheurs 7 segments par exemple) où il suffit d'allumer une LED pour créer des caractères, l'affichage d'un message textuel sur un afficheur LCD n'est jamais direct. Il faut envoyer une série de commandes à l'afficheur, qui les interprètent et qui réalise en fonction certaines actions dont l'affichage des caractères. On distingue 2 types de commandes : les instructions (pour configurer l'afficheur) et les données (pour afficher un caractère à partir de son code ASCII), le protocole d'envoi des commandes à l'afficheur est très précis et doit être respecté si on veut

que la réaction de l'afficheur soit le résultat attendu : afficher un message [2]. Un afficheur LCD contient :

- Une entrée de contrôle RS (Register Select).
- Une autre entrée RW.
- Une entrée de validation E (Enable).
- 8 entrées de données D0 à D7.
- 3 entrées d'alimentation VSS, VDD et VEE.

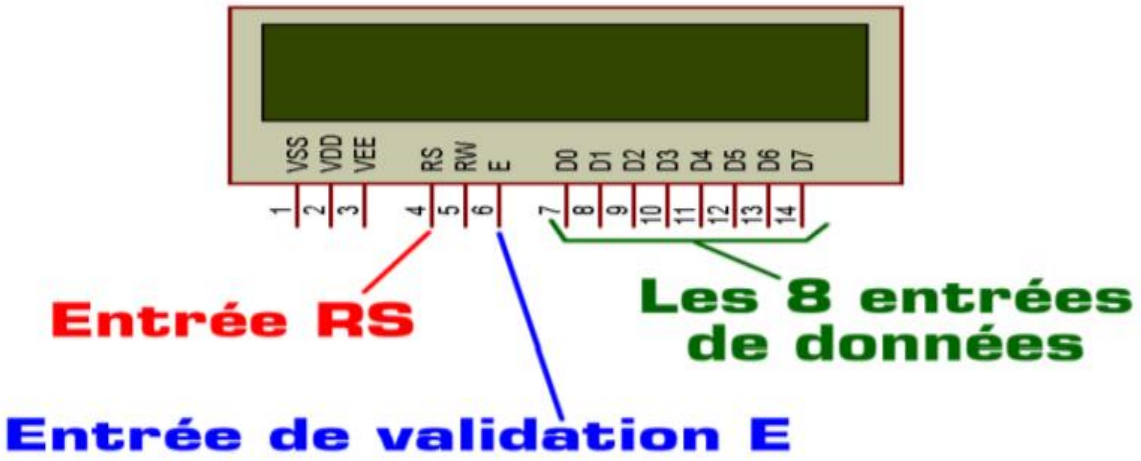


Figure I.3 : Les entres d'un afficheur LCD.

- RS permet de préciser si la commande présente sur les entrées D0 à D7 est une instruction ou une donnée, et E permet de valider cette commande.
- Une commande est une valeur numérique présente sur les entrées D0 à D7 et validée par une impulsion sur E.
- Le protocole d'envoie des commandes précise la liste des instructions à envoyer pour configurer l'afficheur (RS=0) suivie des données à envoyer (RS=1).
- L'entrée RW sera mise à zéro (connectée à la masse) et sera inutilisée ici.
- Les 3 entrées d'alimentation VSS VDD et VEE n'ayant pas besoin d'être obligatoirement alimentées dans ISIS Proteus, elles resteront non connectées dans les exemples ci-dessous.

I.2.1.3. Fonctionnement d'un afficheur LCD

- **Le registre d'instruction IR :** (Instruction Register), C'est le registre de contrôle, suivant la valeur que l'on met dedans, l'afficheur exécute des opérations de configurations, Exemple : "effacement de l'écran" [24].

❖ **Le registre de données DR :** (Data Register), Suivant la valeur que l'on met dedans l'afficheur peut :

- Afficher un caractère (Code ASCII ou spécifiques).

  Ce registre est bidirectionnel. Il peut recevoir les caractères que l'on désire afficher.

Les échanges d'informations sont synchronisés par des signaux de commandes :

- **R/W** (Lecture/écriture).
- **RS** (Register Select : Registre de sélection).
- **E** (Enable : Mémorisation).

RS	RW	Registre sélectionné
0	**0**	**Ecriture dans IR :** registre de contrôle
0	**1**	**Lecture dans IR :** registre de contrôle
1	**0**	**Ecriture dans DR :** registre de données
1	**1**	**Lecture dans DR :** registre de données

Tableau I.1 : Fonctionnement de l'afficheur LCD 2* 16.

I.2.1.4. Le brochage d'un afficheur LCD

L'afficheur LCD a 14 broches en standard et souvent 16, les broches 15 et 16 servent au rétro-éclairage [3], voir le tableau ci-dessous :

Bronche	Nom	Description
1	Vss	Masse
2	Vdd	Alimentation 5v
3	CO	variable de 0 à 5v permet de modifier le contraste de l'afficheur
4	RS	Indique une commande ou une donnée à afficher (0 :Commande / 1 : Donnée)
5	R/W	Indique une écriture ou une lecture (0 : écriture / 1 : lecture)
6	E	Indique une Validation
7-14	D0-D7	Bus de données bidirectionnels
15	A	Anode rétro éclairage
16	K	Cathode rétro éclairage

Tableau I.2 : Le brochage d'afficheur LCD .

I.2.1.5. Commandes d'un afficheur LCD

Deux modes de fonctionnement de l'afficheur sont disponibles :

- **Mode 8 bits :** En mode de commande 8 bits on utilise plus de broches du microcontrôleur. Il faut utiliser 11 broches des ports d'entrées/sorties du microcontrôleur (configurées en sorties) de manière à commander l'afficheur [25].

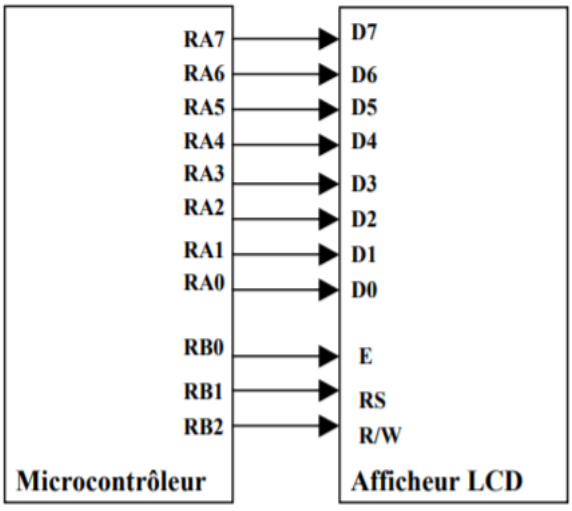


Figure I.4 : Le fonctionnement de l'afficheur LCD en mode 8 bits.

- **Mode 4 bits :** En mode de commande 4 bits l'intérêt est de limiter le nombre de broches du microcontrôleur. Il faut utiliser 7 broches des ports d'entrées/sorties du microcontrôleur (configurées en sorties) de manière à commander l'afficheur [25].

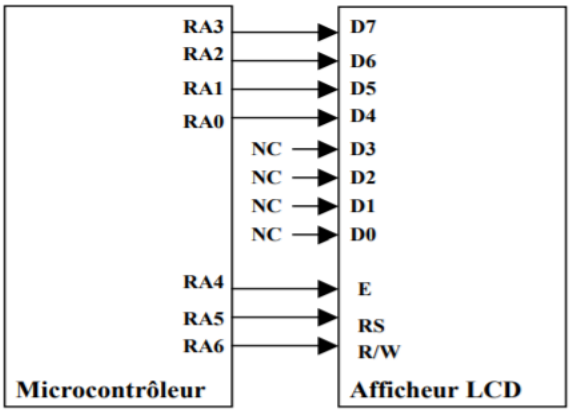


Figure I.5 : Le fonctionnement de l'afficheur LCD en mode 4 bits.

I.2.1.6. Branchements afficheur LCD- microcontrôleurs

Les fils des données (D0, D1, D2, D3) et R/W (Read Write) sont branchées à la masse tandis que (D4, D5, D6, D7) et RS (Registre Select) et E (Enable) sont branchées aux sorties du microcontrôleur [15].

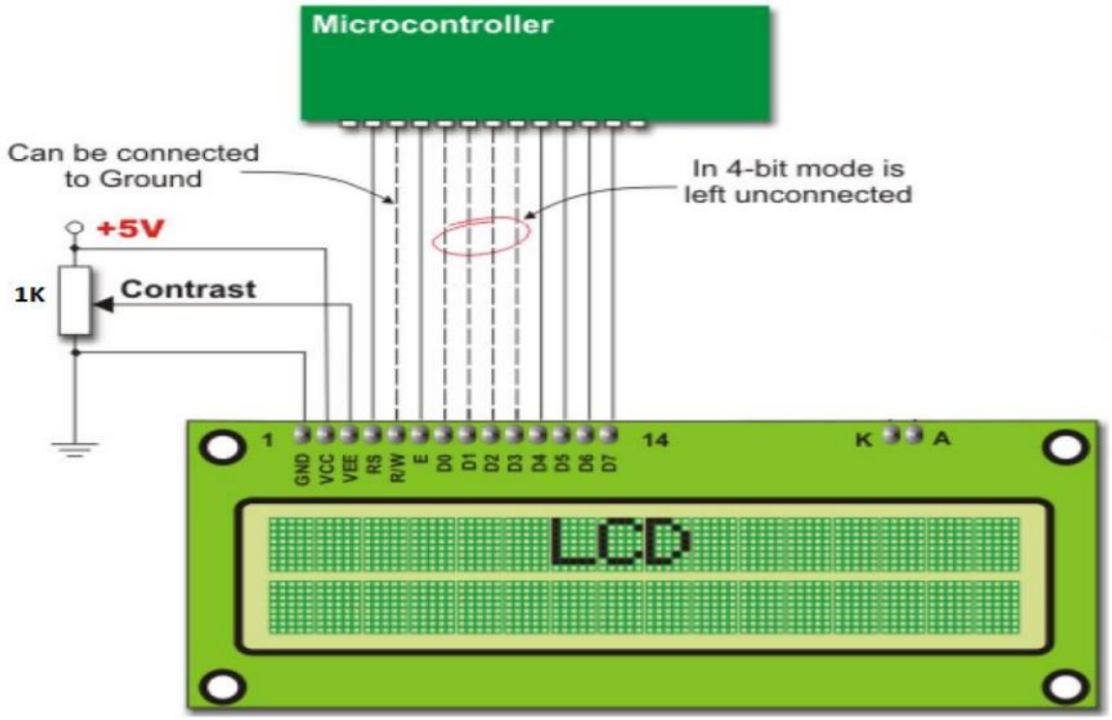


Figure I.6 : Branchement de L'afficheur LCD avec le PIC 16F877.

I.2.1.7. Principes de fonctionnement afficheur- microcontrôleur

La configuration du type de l'information reçue par le microcontrôleur, qui peut être donnée ou commende et cela via le RS [4].

- L'envoi de l'information aux pins D4, D5, D6, D7 de l'afficheur LCD, ce dernier saura le type de l'information puisque le RS est déjà configuré.
- La validation de l'information par le microcontrôleur a l'afficheur via le pin E.
- Si l'information est de type commende, l'afficheur exécutera cette commande, par exemple : saut de ligne, effacer l'afficheur**,** si l'information est de type donné en caractère correspondant et cela en utilisant la table ASCII.
- Le caractère adéquat est affiché.

I.2.1.8. Les mémoires

L'afficheur possède deux types de mémoire, la DD RAM et la CG RAM. La DD RAM est la mémoire d'affichage et la CG RAM est la mémoire du générateur de caractères.

- **La mémoire d'affichage (DD RAM) :** La DD RAM est la mémoire qui stocke les caractères actuellement affiché à l'écran, responsable du positionnement du curseur [3].
- **La mémoire du générateur de caractères (CG RAM) :** Le générateur de caractère est quelque chose de très utile. Il 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 accéd er comme s'il s'agissait de caractères classiques stockés en ROM [3].

I.2.2. Le PIC 16F877

PIC16F877 est le microcontrôleur que nous avons utilisé dans notre projets. Il permet le contrôle et l'affichage des éléments sur l'afficheur LCD. Ce pic est un circuit intègre contenu dans un boîtier nommer « DIL 40 », il présente 40 broches, 20 de chaque côté. Les broches sont virtuellement numérotées de 1 à 40. La 1ere broche est placé dans le coin situé à gauche de l'encoche de repérage. Certaines fonctionnalités de ce microcontrôleur ainsi que ça sa structure interne et externe sont décrites et détaillées dans le chapitre suivant [5].

Figure I.7 : PIC 16F877.

I.3. Description logiciels

I.3.1. Le logiciel ISIS PROTEUS

I.3.1.1. Présentation générale

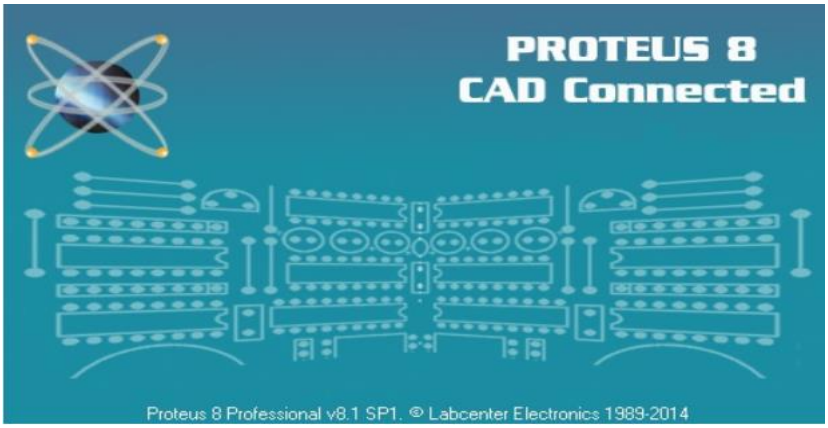


Figure I.8 : Le logiciel proteus.

Le logiciel PROTEUS est un logiciel composé de nombreux modules. En général, on utilise souvent les deux principaux logiciels qui permettent de faire une Conception Assistée par Ordinateur ou Electronic Design Automation (EDA). II est édité par la société Labcenter Electronics. Ses deux logiciels principaux sont :

- L'éditeur de schéma ISIS.
- L'outil de conception de circuit imprimé ARES.

Dans l'environnement électronique, PROTEUS est très apprécié. Grace à sa capacité de faire combiner plusieurs circuits, de nombreux entreprises et organismes de formation et beaucoup de chercheur utilisent et travaillent avec ce logiciel. En plus de la popularité de ses outils, PROTEUS possède aussi d'autre avantages très intéressent. II est un outil de création de prototype virtuel permettant de réduire les coûts matériels et logiciels lors de la conception d'un projet. De plus, il a des supports techniques très performants et il contient aussi des packs logiciels qui sont faciles et rapides à manipuler [6].

Les deux logiciels principaux qui nous intéressent sont :

- Le logiciel ISIS (Intelligent Schematic Input System) est principalement connue pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui nous permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits conçus grâce à ce logiciel peuvent être utilisé dans des

documentations car il permet de contrôler la majorité de l'aspect graphique des circuits.

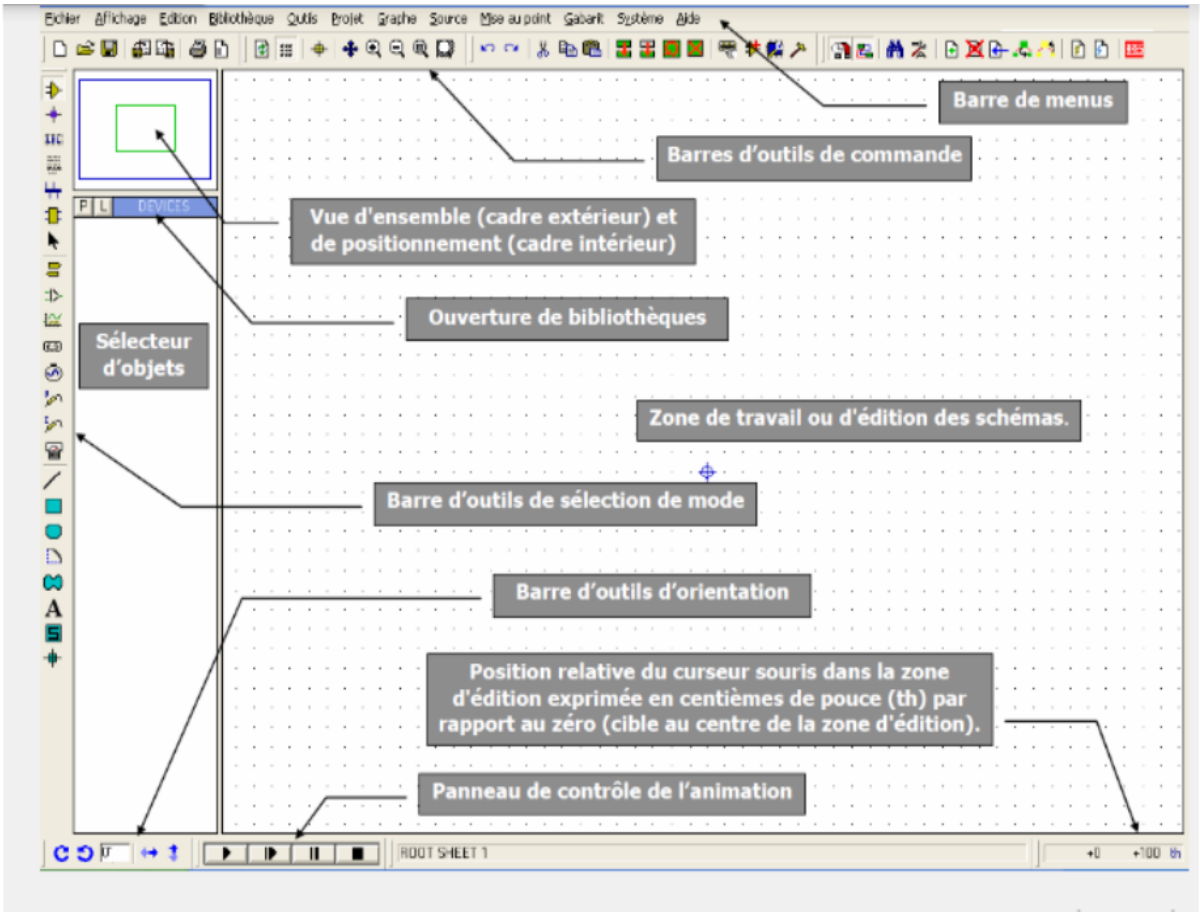


Figure I.9 : Présentation d'ISIS de PROTEUS.

- Le logiciel ARES (Automatic Rotage Equipment System) est un outil d'édition et de routage qui complète parfaitement ISIS. Un schéma électrique réalisé sur ISIS peut alors être importé facilement sur ARES pour réaliser le PCB (Printed Circuit Board qui veut dire circuit imprime) de la carte électronique. Bien que l'édition d'un circuit imprimé soit plus efficace lorsqu'elle est réalisée manuellement par routage, ce logiciel permet de placer automatiquement les composants et de réaliser le routage automatiquement. Cette partie du travail est la plus importante car elle demande de la concentration et un savoir-faire qu'est obtenu avec le temps et surtout avec la pratique [6], voir la figure ci-dessous :

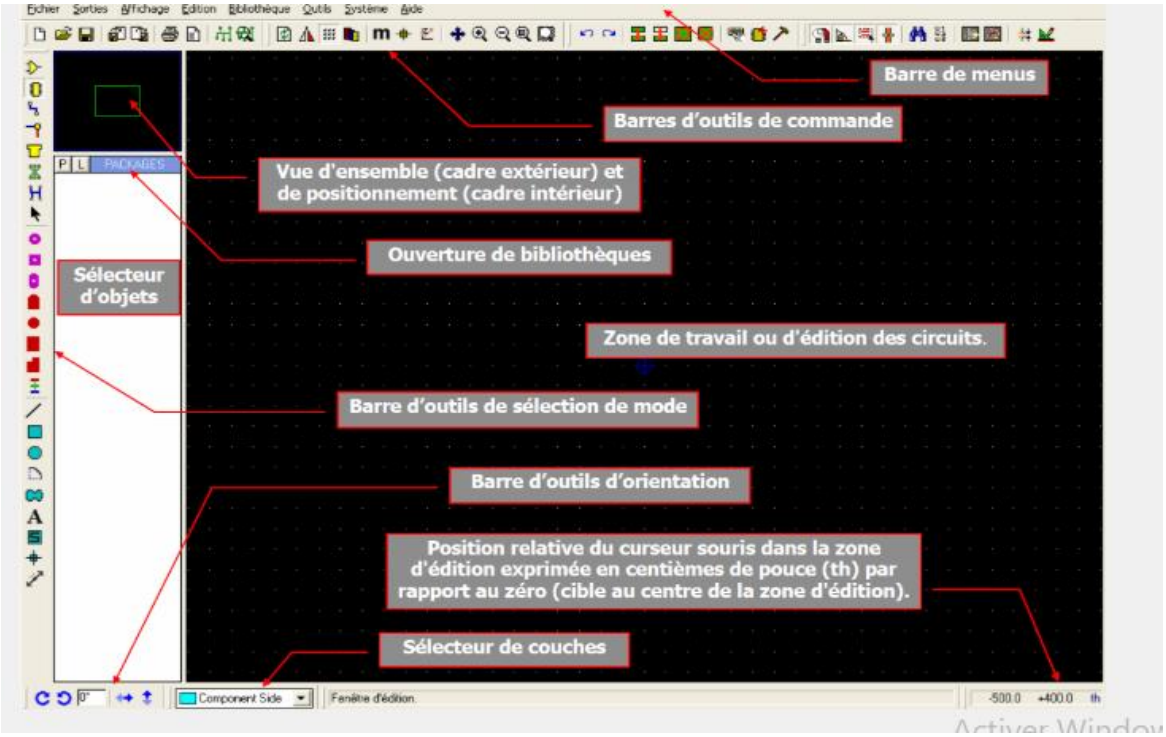


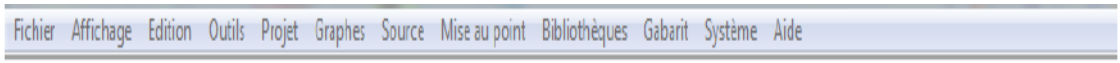
Figure I.10 Présentation d'ARES de PROTEUS.

I.3.1.2. Commencer avec ISIS Proteus

Pour créer un nouveau projet, on clique sur «file-new design» puis on clique sur «Save design» et on donne le nom et l'emplacement de fichier sur PC. Une fois le nouveau projet est créé, on peut commencer la conception de schéma électrique de projet en ajoutant les composantes directement de la bibliothèque de composantes. Afin de tester le montage que nous avons fait, il suffit juste de démarrer la simulation en cliquant sur «Run the simulation », Proteus dispose aussi d'une boite à outils pour la mesure (Ampèremètre, Voltmètre, Oscilloscope, …). Pour passer de schéma électrique au design PCB on clique sur le bouton « ARES » dans la barre d'options, une autre fenêtre apparaître contenant les composantes que nous avons utilisées dans le schéma avec toutes les connexions que nous avons fait. Il suffit juste de commencer le routage et la structuration de la carte PCB de la façon que nous souhaitons [6].

I.3.1.3. Les barres d'outils

- **Barre de menus**



Elle permet de gérer les travaux (ouverture, sauvegarde…) sur vos fichiers.

- **Barre des outils de commande :**

Elle reprend ce qui est accessible par les menus.

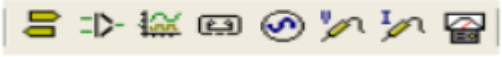
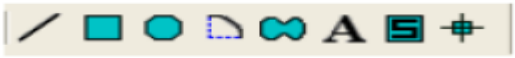
 Commande sur les fichier (nouveau, ouvrir.….)

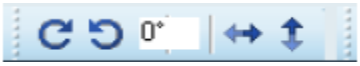
 Commandes d'affichage (grille, zoom…..)

 Commande /Edition/Object / Bibliothèque….

 Commande Outils /Project …

- **Barre d'outils de sélection des modes :**
  Cette **barre** permet de sélectionner un outil parmi les modes d'édition disponible .

1. **Modes principaux** 
2. **Modes gadget** 
3. **Mode graphique** 

- **Barre d'outils d'orirntation *:***
   Cette barre permet d'afficher et de contrôler la rotation d'un objet placé ou à placer.

- **Zone de travail :**
   zone rectangulaire ou on dépose les composant pour dessiner le schéma structurel du modèle à simulé ou de la carte à router.

- **Sélecteur et d'objet :**
  

  zone rectangulaire ou on trouve tous les composants présents dans le dessin

I.3.2. Le logiciel MikroC

I.3.2.1. Présentation du MikroC

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

Il est conçu pour fournir les solutions les plus faciles que possibles pour des applications se développant pour les systèmes à microcontrôleur. Il contient un large ensemble de bibliothèques de matériel, de composant et la documentation complète [7].

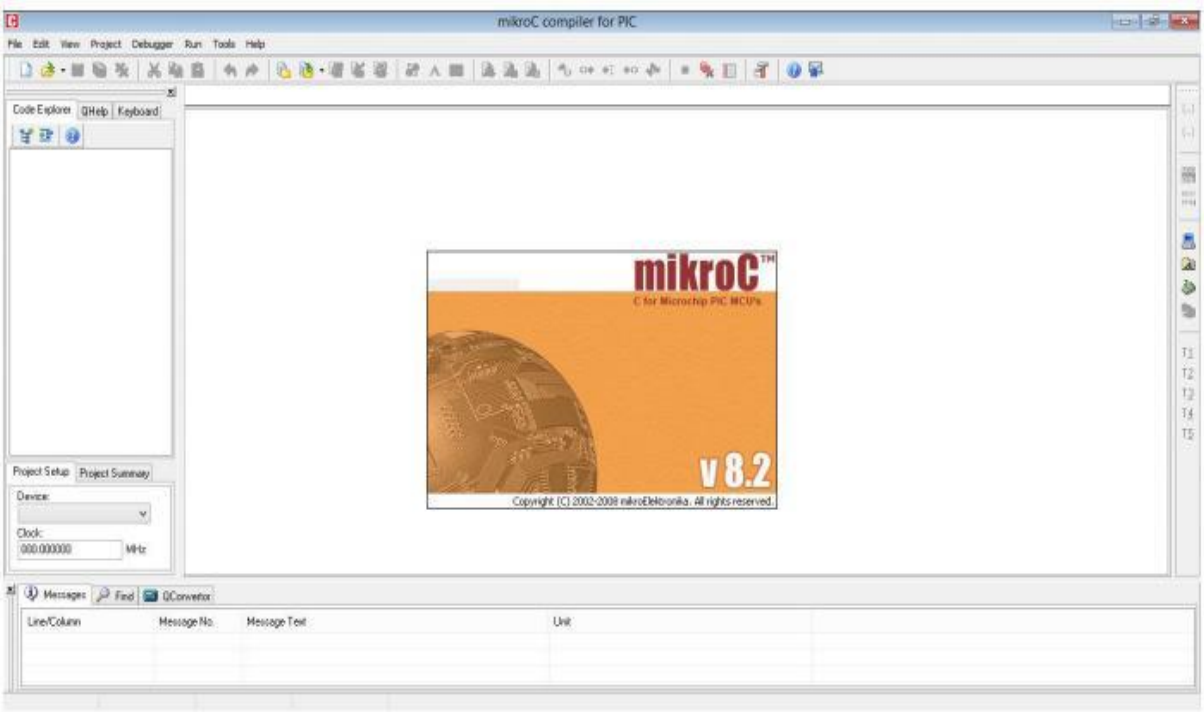


Figure I.11: Interface du logiciel MikroC.

Le compilateur MikroC nous permet de développer rapidement des applications complexes [7].

I.3.2. 2. Création d'un projet

Le mikroC PRO pour PIC organise des applications dans des projets, composé d'un seul fichier de projet (extension. mcppi) et un ou plusieurs fichiers sources (extension).

Les fichiers source peuvent être compilés que si elles font partie d'un projet. Le fichier projet contient les informations suivantes : [7]

- Nom du projet et une description facultative.
- Composant cible.
- Option du composant.
- Fréquence d'horloge du composant.
- La liste des fichiers source du projet avec les chemins.
- Fichiers d'image.
- Fichiers binaires (* mcl.).
- Autres fichiers.

Le processus de création d'un nouveau projet est vraiment très simple. Sélectionnez New Project (Nouveau Projet) de puis le menu Project (Projet), comme indiqué sur (Fig I.12).

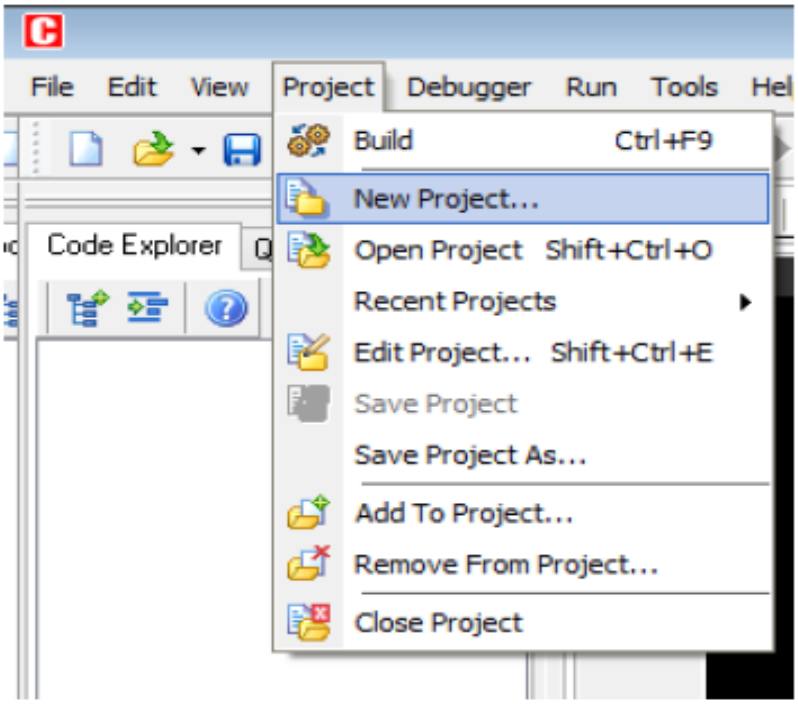


Figure I.12: Création d'un projet.

Nouvelles étapes de l'Assistant de projet :

Commencez à créer votre nouveau projet, en cliquant sur le bouton Next :

Premier et deuxième étapes Sélectionnez le périphérique dans le périphérique dans la liste déroulante et Saisir la valeur de fréquence de l'oscillateur.

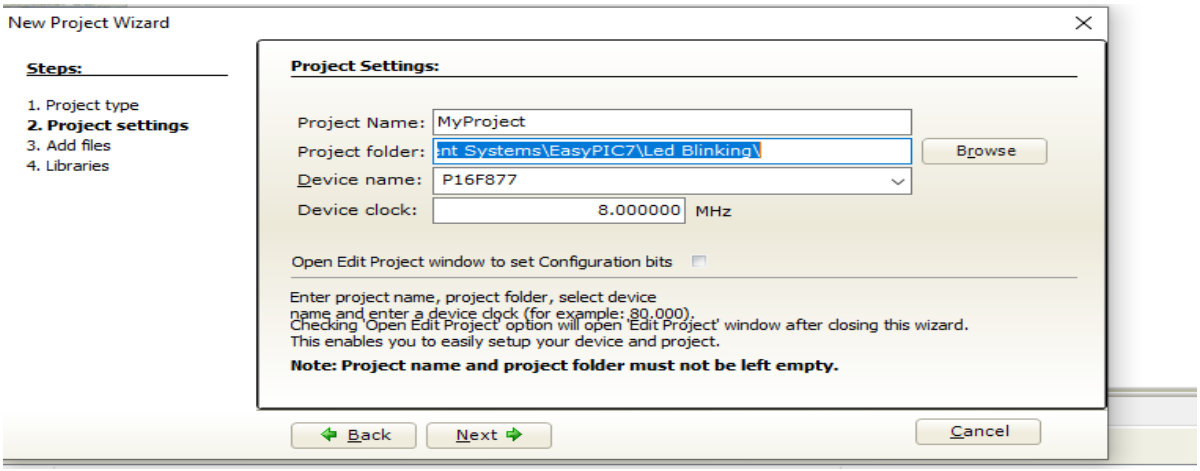


Figure I.13 : Fréquence d'oscillateur.

Troisième étape : Spécifiez l'emplacement ou votre projet sera enregistré.

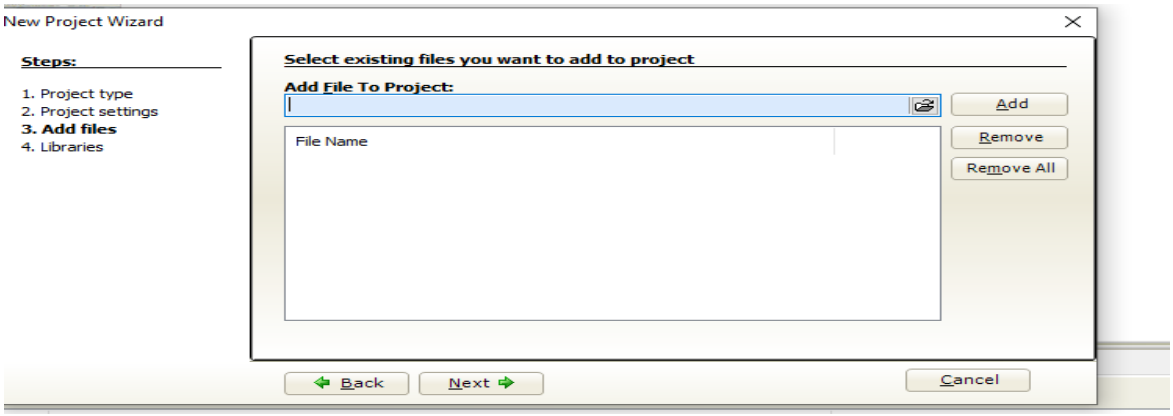


Figure I.14 : L'emplacement du projet.

Quatrième étape : Cliquez sur finish pour créer votre projet. À ce stade, une nouvelle fenêtre vide s'afficher afin de saisir votre programme.

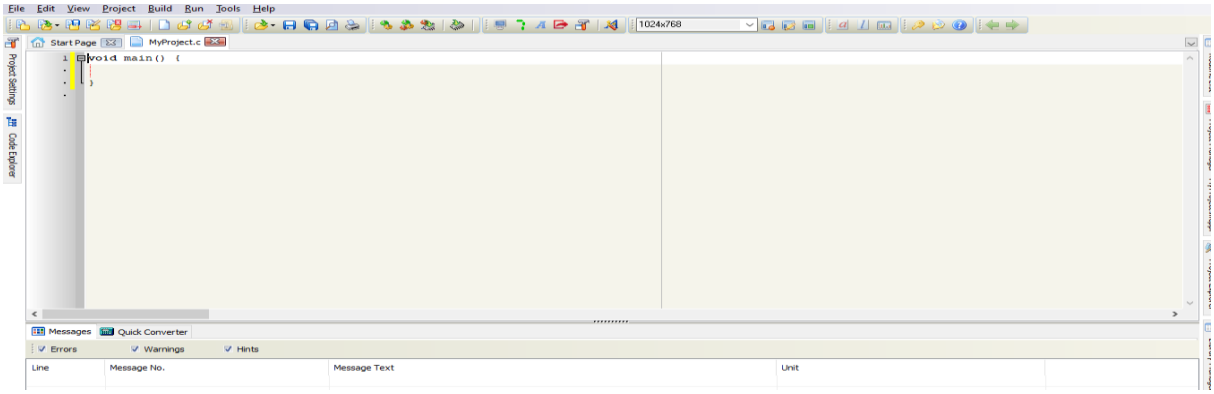


Figure I.15 : Fenêtre de la saisie du programme.

I.3.2.3 Compilation

Lorsque vous avez créé le projet et écrit le code source, il est temps de le compiler. Sélectionnez **ProjectBuild** à partir du menu déroulant ou cliquez sur l'icône **Build** dans la barre d'outils du projet.Si plus d'un projet est ouvert, vous pouvez compiler tous ouverts projets en sélectionnant Project >Build All dans le menu déroulant, ou cliquez sur l'icône  de la barre d'outils du projet. Barre de progression s'affiche pour vous informer sur l'état de la compilation. S'il y a des quelques erreurs, vous en serez informé dans la fenêtre d'erreur (figure I.16) [7].

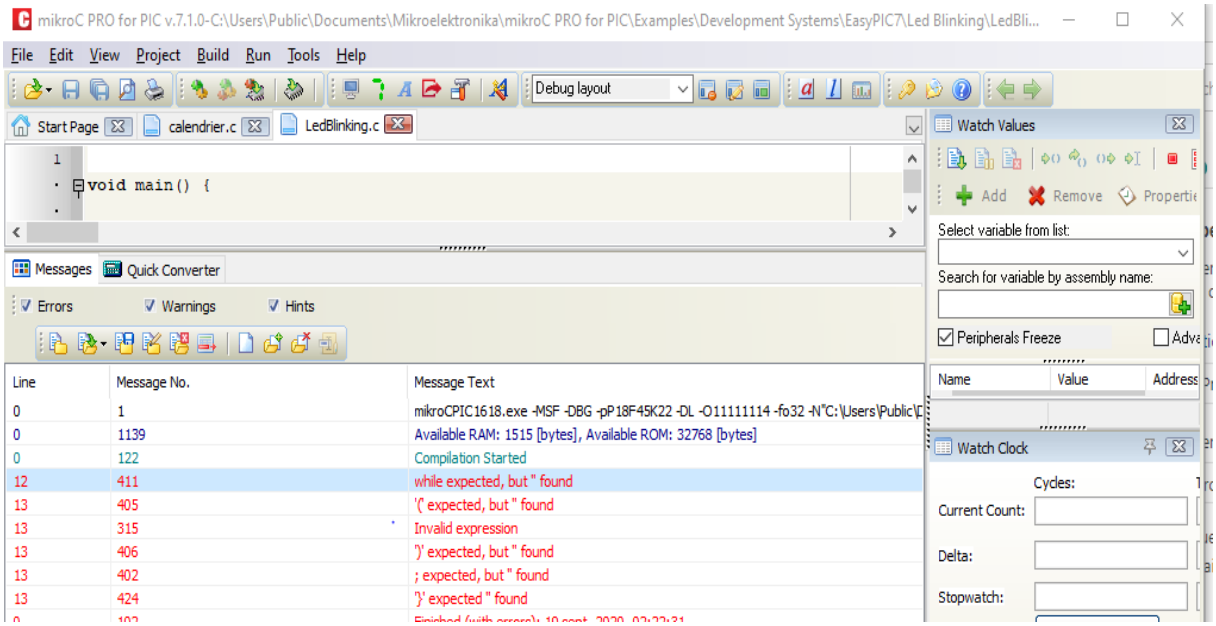


Figure I.16 : Avertissement des erreurs.

Après la compilation réussie, le compilateur mikroC PRO pour PIC génère des fichiers de sortie dans le dossier du projet (dossier qui contient le fichier projet. mcppi). Les fichiers de sortie sont résumés dans le tableau ci-dessous:

Format	Description	Type de fichier
Intel HEX	Code hexadécimal dans le format Intel. Fichier est utilisé pour programmer PIC	.hex
Binary	Fichier compilé pour la bibliothèque mikroC. Les distribution binaires sont des routines qui susceptible d'être inscrits dans d'autre projets.	.mcl
List File	L'image globale de l'allocation de mémoire du PIC pour : Adresses et les étiquettes du programme.	.lst
Assembler file	Le fichier en assembleur avec des noms symboliques. Obtenus à partir de liste des fichiers.	.asm

Tableau I.3 : Compilation (mikroC).

I.3.3. Introduction au langage de programmation mikroc

I.3.3.1. Règles générales d'écriture en mikroC

- Les instructions propres au langage mikroC 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 « ; ».

I.3.3.2. Début et fin d'un programme

En langage mikroC, un programme commence avec les mots-clés :

```
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. le programme a la structure suivante :

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

I.3.3.3. Fin d'une instruction

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

Example :

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

I.3.3.4. Espaces blancs

Les espaces blancs sont des espaces, des flans, les tabulations et les caractères de nouvelle ligne. Le compilateur *mikroC* ne tient pas compte de tous les espaces blancs.
Ainsi, les trois séquences suivantes sont identiques :

```
int i; char j;
ou
int i;
char j;
ou
int i;
char j;
```

I.3.3.5. Instructions d'itération for, while, do, goto, continue et break

Les instructions d'itération nous permettent d'effectuer des boucles dans un programme, où une partie d'un code doit être répété un certain nombre de fois. Dans mikroC l'itération peut être effectuée de quatre façons. Nous se penchera sur chacun des exemples :

- Utilisation de for
- Utilisation de while
- Utilisation de do
- Utilisation de goto, continue et break

ADC Utilis**é** pour conversion analogique/numérique
PWM Utilisé pour les opérations avec le module de PWM
LCD_CLEAR Effacer l'affichage

LCD Utilisé pour le fonctionnement avec LCD standard

Delay_ms Crée le retard constant dans les unités de millisecondes

ADC	Utilisé pour conversion analogique/numérique
PWM	Utilisé pour les opérations avec le moule de PWM
LCD_CLEAR	Effacer l'affichage
LCD	Utilisé pour le fonctionnement avec LCD standard
Delay_ms	Crée le retard constant dans les unités de millisecondes

Tableau I.4 : Instructions d'itération.

I.4. Conclusion

Dans ce chapitre nous avons présenté les outils utilisés dans notre projet, qui sont très important pour accomplir notre travaille en général la partie matérielle (afficheur LCD) et spécifiquement les logiciels d'assimilation et de programmation. Le chapitre suivant sera consacré à l'étude des généralités sur les microcontrôleurs en générale et spécifiquement le PIC16F877.

Chapitre II

Mise en œuvre du microcontrôleur pic 16F877

II.1. Introduction

L'évolution des systèmes électroniques amène de plus en plus souvent les concepteurs à remplacer l'électronique câblée à base de nombreux circuits intégrés par un circuit programmable qui remplit à lui seul toutes les fonctions, les microcontrôleurs appartiennent à cette famille de circuits.

Le développement des applications à base des microcontrôleurs PIC est assez courant, ceci est dû à plusieurs causes : réalisation d'un ensemble important de traitements d'informations, beaucoup de ressources internes, mémoires embarquées de plus en plus grande, vitesse de calcul accrue…

Les microcontrôleurs sont très utilisés dans le monde de l'industrie, notamment dans les systèmes embarqués. On pourra donc les retrouver dans l'aéronautique, l'aérospatial, l'automobile, l'électronique grand publique. Leur polyvalence, et leur taille les rendent intéressants pour des modules de traitement de données numériques et aussi analogiques.

Au niveau de ce chapitre on va essayer de mieux connaître le PIC16F877 (PIC choisie pour ce projet), de savoir manipuler ces instructions internes.

II.2. Généralités sur les microcontrôleurs

Le microcontrôleur (µC) est un dérivé du microprocesseur. Sa structure est celle des systèmes à base de microprocesseurs. La partie la plus « noble » du microcontrôleur est ce qu'on appelle la CPU (Central Processing Unit, unité centrale de traitement). C'est dans cette unité que se trouvent plusieurs éléments tels que l'ALU (Arithmetic and Logical Unit, unité arithmétique et logique) qui exécute une par une l'ensemble des instructions écrites dans le programme principal.

Le microcontrôleur est aussi composé d'une mémoire (mémoire vive RAM et mémoire morte ROM), une (ou plusieurs) interface de communication avec l'extérieur matérialisé par les ports d'entrée/sortie.

En plus de cette configuration minimale, les microcontrôleurs sont dotés d'autres circuits d'interface qui vont dépendre du microcontrôleur choisi à savoir les systèmes de comptage (TIMER), les convertisseurs analogique/numérique (CAN) intégré, gestion d'une liaison série ou parallèle, un Watchdog (surveillance du programme), une sortie PWM (modulation d'impulsion), … [8] [9], comme le montre la figure suivante :

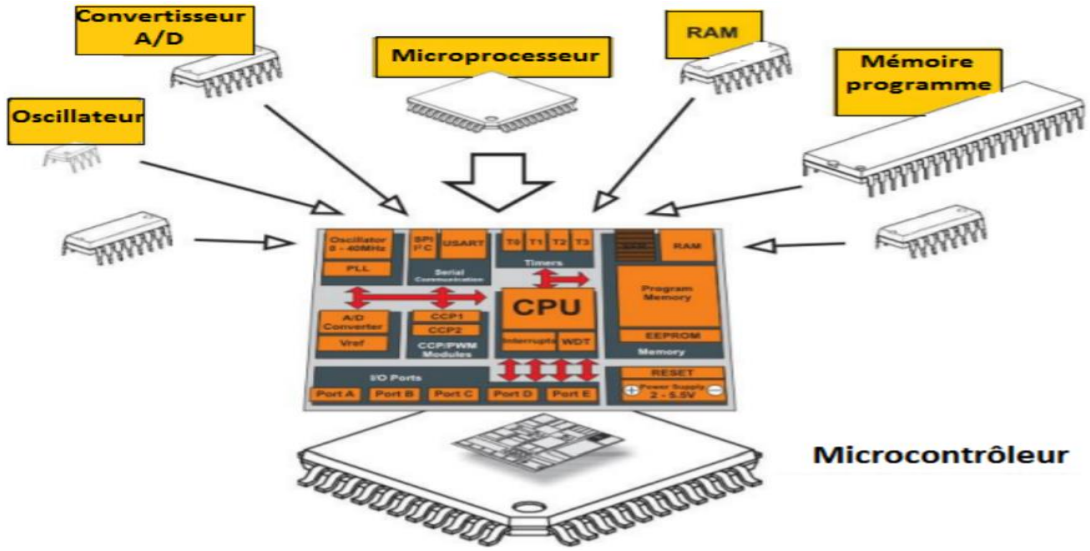


Figure II.1 : Contenu type d'un microcontrôleur.

Les microcontrôleurs sont plutôt dédiés aux applications qui ne nécessitent pas une grande quantité de calculs complexes, mais qui demandent beaucoup de manipulations d'entrées/sorties. C'est le cas de contrôle de processus. Les systèmes à microprocesseurs sont plutôt réservés pour les applications demandant beaucoup de traitement de l'information et assez peu de gestion d'entrées / sorties. Les ordinateurs sont réalisés avec des systèmes à microprocesseur [8].

Il existe plusieurs familles de microcontrôleurs dont les plus connues sont :

- **Atmel** : AT; familles AT89Sxxxx, AT90xxxx, …
- **Motorolla** : famille 68HCxxx, …
- **Microship** : PIC ; familles 12Cxxx, 16Cxxx, 16Fxxx, 18Fxxx, …
- **Intel** : famille 80C186XX.
- **STMicroelectronics** : famille STX.
- **AnalogDevices** : famille ADuC.

Nous allons nous intéresser dans le cadre de ce cours à la famille Microchip PIC (Programmable Integrated Circuit) de moyenne gamme (MIDRANGE).

II.3. Présentation d'un microcontrôleur PIC

Ils sont des composants dits RISC (Réduced Instructions Construction Set), ou encore composant à jeu d'instructions réduit. Chaque instruction complexe peut être programmée par plusieurs instructions simples. Sachant que plus on réduit le nombre d'instructions, plus facile et plus rapide qu'en est le décodage, et plus vite le composant fonctionne [8].

II.3.1. Les différentes familles des PIC

La famille des PIC à processeur 8 bits est subdivisée à l'heure actuelle en 3 grandes Catégories : [8]

- ✓ **Base-Line** : ils utilisent des mots d'instruction de 12 bits
- ✓ **Mid-Range** : ils utilisent des mots d'instruction de 14 bits.
- ✓ **High-End** : ils utilisent des mots d'instruction de 16 bits.

Nous nous limiterons dans notre travail à la famille « **Mid-Range** » et particulièrement au PIC16F877 qu'on le trouve en boîtier DIL (Dual In Line) de 2x20 broches.

II.3.2. Identification des PIC

Pour identifier un PIC, on utilise simplement son appellation du type : wwlxxyyy-zz

- WW : Représente la catégorie du composant (12, 14, 16, 17, 18).
- L : Tolérance plus importante de la plage de tension.
- XX : Type de mémoire de programme :
  - ✓ C: EPROM ou EEPROM.
  - ✓ CR: PROM.
  - ✓ F : FLASH.
- YYY: indique le modèle du PIC (son identité)-
-ZZ: la fréquence d'horloge maximale supportée par le PIC.

Concernant le PIC 16F877-A

- 16 : indique la famille du PIC (Mid-Rang).
- F : indique le type de la mémoire utilisée (Flash).
- 877 : identité du PIC.
- A : fréquence maximale d'horloge qui est de 20MHZ.

Alors Le 16F877A est un microcontrôleur de MICROCHIP, fait partie intégrante de la famille des Mid- Range (16) dont la mémoire programme est de type flash (F) de type 877 et capable d'accepter une fréquence d'horloge maximale de 20Mhz.

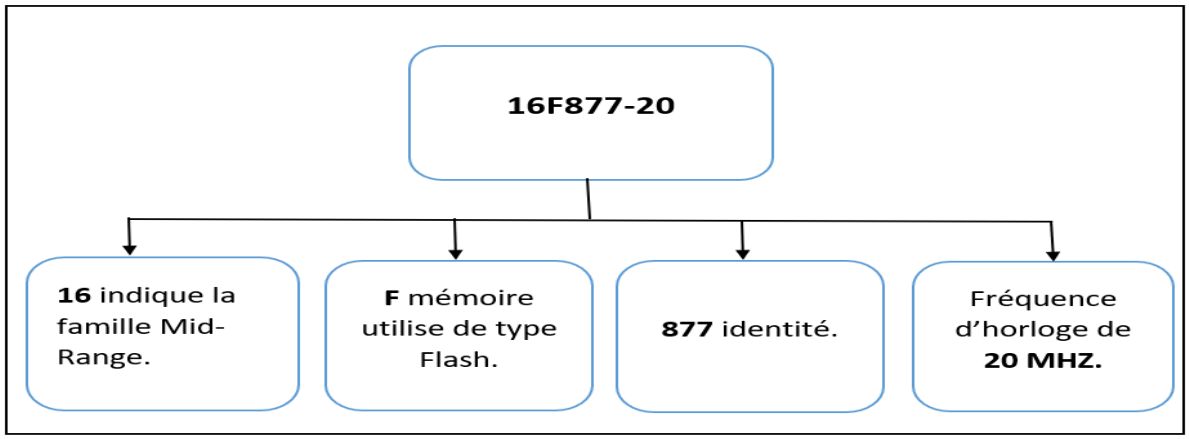


Figure II.2 : Description de la configuration du pic 16F877A.

II.4. Les différentes architectures des Pics

II.4.1. Architecture en matière de représentation

Deux types d'architectures sont conventionnels dans les microcontrôleurs :

A. Architecture Von –Neumann

L'architecture VON NEUMANN employée par la plupart des microcontrôleurs actuels (INTEL80XX, Motorola HC05, HC08 et HC11, ou ZILOG Z80) est basée sur un bus de données unique. Celui-ci véhicule les instructions et les données. 1 bloc mémoire et 1 bus de données sur 8 bits (1 octet). Toutes les données sont échangées sur ce bus qui, surchargé, rend la communication très lente [10].

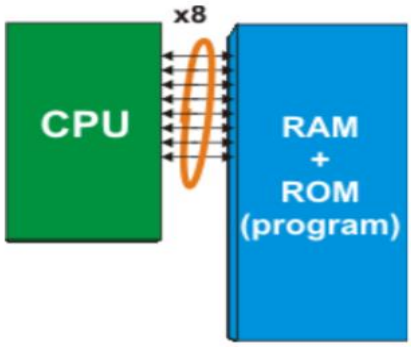


Figure II.3 : Architecture Von –Neumann.

B. Architecture Harvard

L'architecture HARVARD utilisée par les microcontrôleurs PIC est basée sur deux bus de données. Un bus est utilisé pour les données et un autre pour l'instruction. 2 blocs mémoire distincts et 2 bus différents : 1 bus 8 bits pour communiquer avec la RAM, 1 bus 14 bits pour communiquer avec la ROM, qui contient le programme. La CPU peut lire une instruction (en

ROM) et accéder à la mémoire de données (en RAM) en même temps résultantes ainsi une amélioration des performances [10].

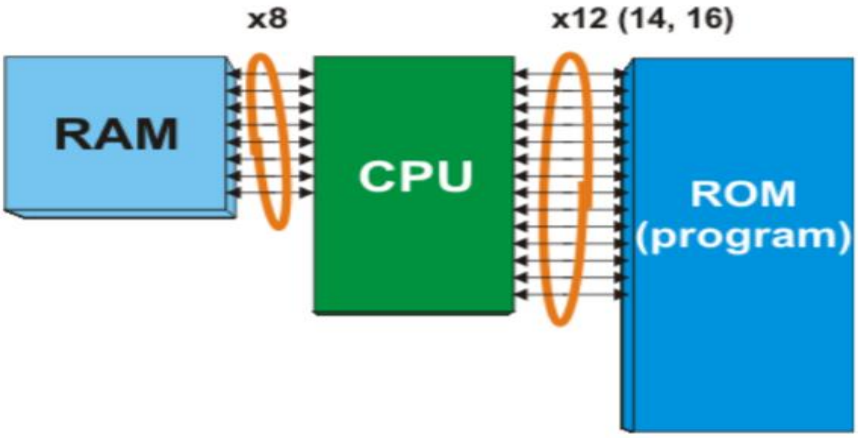


Figure II.4 : Architecture Harvard.

Remarque : La différence se situe au niveau de la séparation ou non des mémoires programmes et données. La structure de Harvard permet de transférer données et instruction Simultanément, ce qui permet un gain de performances [21].

II.4.2 Architecture en matière de traitement des instructions

Il existe 2 catégories de microprocesseur : les CISC et les RISC.

A-CISC (Complex Instruction Set Computer) : Ce microprocesseur possède un nombre important d'instructions. Chacune d'elles s'exécute en plusieurs périodes d'horloges [11].

B-RISC (Reduced Instruction Set Computer) : Ce microprocesseur possède un nombre réduit d'instructions. Chacune d'elles s'exécute en une période d'horloge. (Cas du PIC) [11].

II.5. Le microcontrôleur PIC 16F877

II.5.1. Schéma simplifié

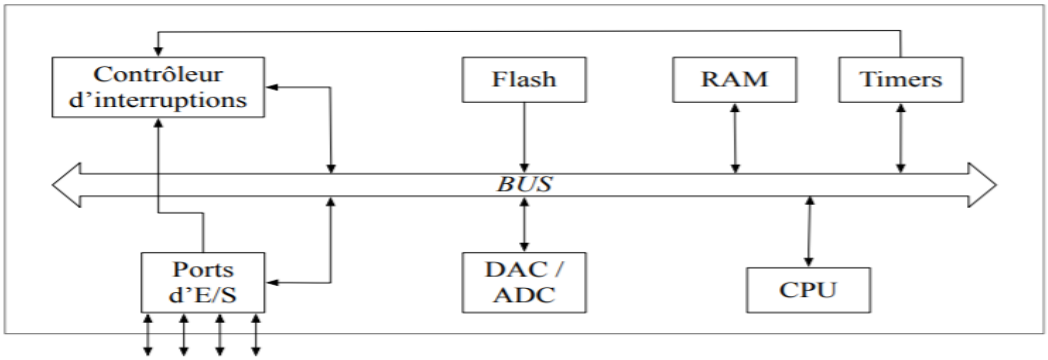


Figure II.5 : Schéma simplifié du PIC16F877 [12].

Le schéma précédent montre qu'un microcontrôleur est constitué de plusieurs éléments lui permettant d'effectuer les calculs nécessaires à l'exécution de programmes qui lui sont dédiés, et à l'interaction avec le circuit électronique qui constitue son environnement [12]. Les principaux composants d'un microcontrôleur sont les suivants :

- Une unité centrale de calcul, ou CPU (Central Processing Unit), dont le rôle est de réaliser les calculs arithmétiques et logiques nécessaires à l'exécution d'un programme.
- Une mémoire vive, ou RAM (Random-Access Memory) qui contient les données dynamiques, volatiles, générées lors de l'exécution du programme.
- Une mémoire flash, non-volatile, qui est destinée à stocker le code du programme. Bien que la mémoire flash soit techniquement accessible en écriture au cours de l'exécution d'un programme, il est habituel de la considérer comme une mémoire morte (ou ROM : Read-Only Memory).
- Des compteurs (ou timers) qui permettent de mesurer des durées, afin de synchroniser les divers composants électroniques d'un montage entre eux. Ces compteurs sont incrémentés à intervalles de temps réguliers.
- Un mécanisme d'interruptions permettant de déclencher, dès l'apparition d'un stimulus interne ou externe particulier, l'exécution immédiate d'une routine spécialisée pour le traiter.
- Des ports d'entrées/sorties permettent de communiquer avec les composants électroniques connectés au microcontrôleur. Cette communication est réalisée en échangeant des signaux électriques de tension variable (par exemple 0 ou 5 Volts), permettant de représenter la transmission de signaux binaires entre le microcontrôleur et son environnement.
- Certains ports d'entrée/sortie supportent par ailleurs la lecture de valeurs analogiques par l'intermédiaire de convertisseurs analogique-numérique (ou ADC : Analog-to-Digital Converters).

Le tableau suivant représente les caractéristiques générales du PIC 16F877 :

Caractéristiques	16F877
Broche	40
E/S max	33
µy flash	8 KO
µy E²PROM	256 O
CAN	7
PWM	2 DE 10 BITS
TIMER r	3
Comparateurs	2
Interruption	13
Oscillateur	20 MHz MAX
Port série	USART/SSP

Tableau. II.1 : Caractéristiques du PIC 16F877.

II.6. Structure du PIC 16F877

La diversité des domaines d'utilisation des Pics, a fait que ces produits se présentent sous différentes formes et structures, pour quelles soit adaptées pour chaque domaine d'utilisation. Nous allons présenter dans ce qui suit les structures susceptibles des PIC à savoir la structure externe et interne.

II. 6.1 Structure externe

Le schéma de la figure représente la structure externe du PIC 16F877, comportant [13]

- 33 pins d'entrées/sorties.
- 4 pins pour l'alimentation : deux connections VDD et deux connections VSS.
- 2 pins pour l'oscillateur.
- 1 pin pour le RESET : MCLR.

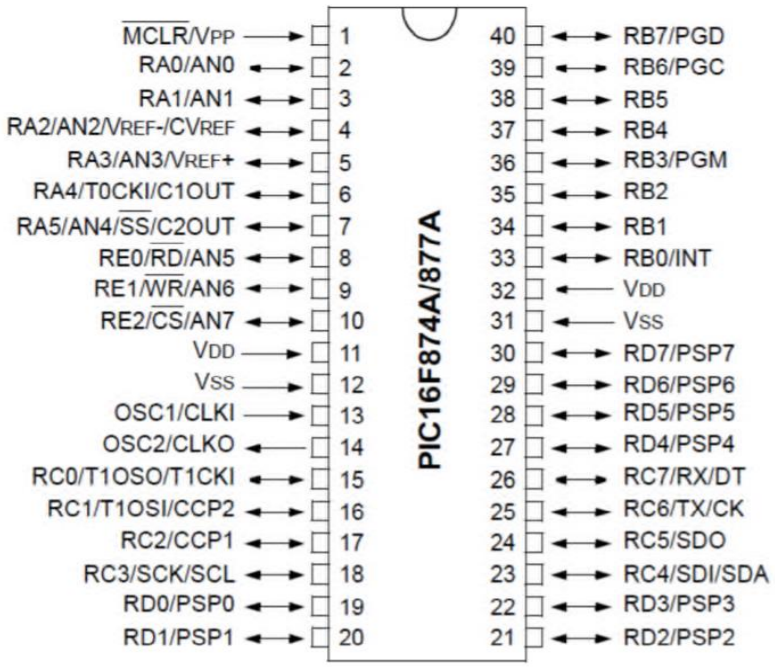


Figure II.6 : Les broches du PIC 16F877 [14].

II.6.1.1. Les broches du PIC16F877

A- Les broches de fonctionnement

Les broches de « fonctionnement » sont les broches qui permettent au microprocesseur de fonctionner. Ces broches doivent obligatoirement être connectées pour que le 16F877 fonctionne d'où leur nom : [5]

❖ **Les broches d'alimentation :**

Comme tout CI, le 16F877 a des broches d'alimentation : **VDD** (Broche 11 et 32) et **VSS** (Broche 12 et 31). Il suffit de connecter une de chaque à l'alimentation pour que le CI fonctionne. On remarque qu'on a 2 connections « VDD » et 2 connections « VSS ». La présence de ces 2 pins s'explique pour une raison de dissipation thermique. Le courant porté dans le PIC n'est pas négligeable parce qu'il existe de nombreuses lignes d'entrées/sorties disponibles.

❖ **Les broches du Quartz :**

Deux pates sont présentées sur le PIC16F877. La première nommée **OSC1** (Broche 13) et et la seconde **OSC2** (Broche 14). Ces deux broches peuvent être utilisées de plusieurs manières en fonction d'oscillateurs ou du quartz employés

❖ **La broche de réinitialisation :**

La broche **MCLR** sert à initialiser le pic qui dispose de plusieurs sources de RESET

(POR: la mise sous tension, EXTERNAL RESET: remise à zéro externe, WDT: chien de garde, BOR : baisse de tension d'alimentation).[5]

B- Les Ports du 16F877

Toutes les autres broches du 16F877 sont des broches de port pour la communication avec l'extérieur.

II .6.1.2. Les ports du PIC 16F877

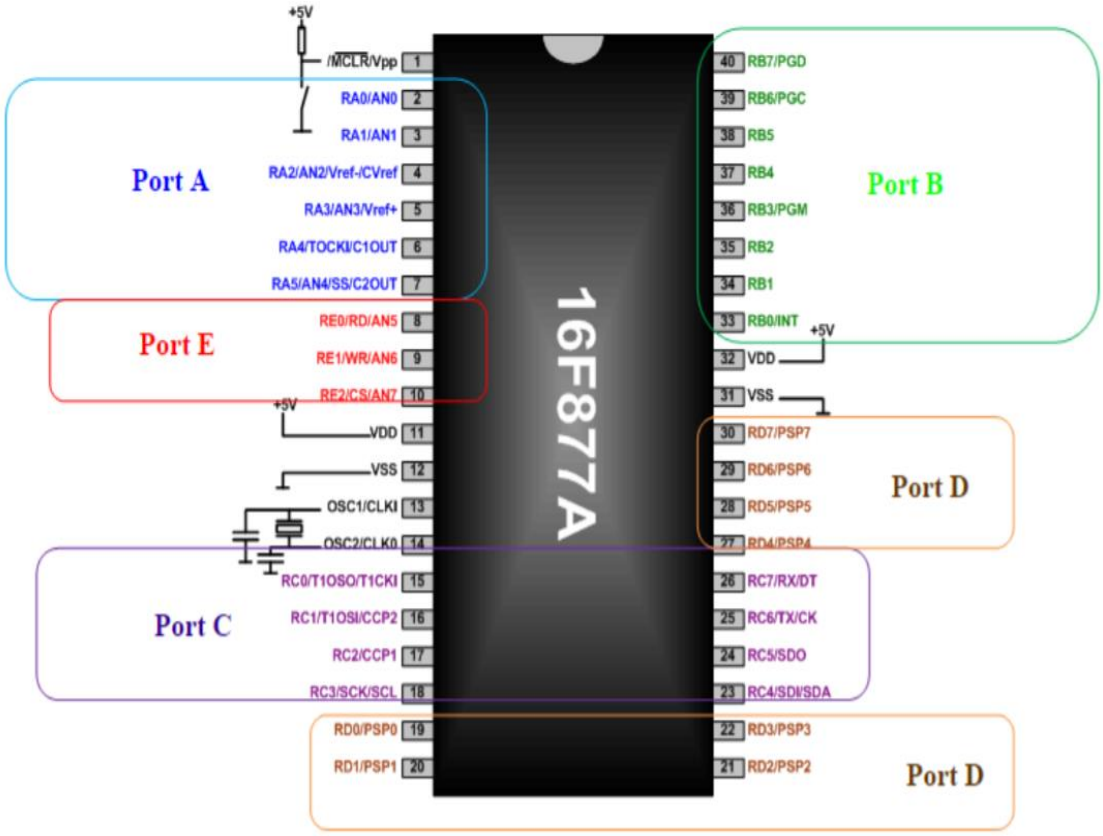


Figure II.7 : Les ports du PIC 16F877.

Pour communiquer avec l'extérieur Le PIC 16F877 dispose de 5 ports qui sont bidirectionnels, ce qui signifie qu'ils peuvent êtres configure s et utilises comme étant des sorties ou entrés : [10]

- Port A : 6 pins I/O numérotées de RA0 à RA5.
- Port B : 8 pins I/O numérotées de RB0 à RB7.
- Port C : 8 pins I/O numérotées de RC0 à RC7.
- Port D : 8 pins I/O numérotées de RD0 à RD7.
- Port E : 3 pins I/O numérotées de RE0 à RE2.

Remarque : Tout comme pour la numérotation des bits d'un registre, la 5 éme broche du port B porte le numéro 4 (et non pas 5). Toutes les broches de ces ports sont des broches E/S, c'est-à-dire que l'on va pouvoir les configurer en entrée ou en sortie. Avec ses broches, on va donc pouvoir soit « sortir » des données du 16F877 sous forme de 1 et de 0 correspondant respectivement à une mise des broches au niveau HAUT ou BAS. On va également pouvoir « lire » des données sous forme de 1 et de 0 correspondant respectivement à un niveau HAUT ou BAS présent sur les broches. La structure de chaque port, à quelques variations près, est grosso modo la même ; chaque port possède assez logiquement un registre de données associé. Pour chaque broche configurée en sortie, chaque bit du registre de donnée détermine le niveau BAS ou HAUT présent sur cette broche. Pour chaque broche configurée en entrée, à chaque bit du registre de donnée correspond le niveau BAS ou HAUT présent sur cette broche [11].

En résumé :

- le sens E/S des broches est configuré dans un registre appelé TRIS. Ce dernier permet de définir si les différentes pattes du port fonctionnent en entrée ou en sortie.
  - ✓ Un « 1 » dans un bit du registre TRISA met la sortie correspondante en haute impédance, elle peut ainsi servir d'entrée.
  - ✓ Un « 0 » dans un bit de ce registre transfert le contenu de la sortie de la bascule D sur la sortie correspondante.
- un registre de données est attaché à chaque port.
- Certaines pattes ont plusieurs fonctions : On dit que les fonctions sont multiplexées [28].

➢ **Port A**

Les broches port A, excepté RA4, sont multiplexées, avec les entrées du convertisseur analogique numérique (AN0 .. AN4). La broche RA4 est multiplexé avec l'entrée d'horloge externe du timer0 (RA4/T0CKI) [11].

➢ **Port B**

Le port B peut être programmé pour un tirage à 5V (pull up) de toutes ses lignes que l'on peut mettre ou non en service en mode entrée uniquement. Elles sont automatiquement désactivées quand le port est configuré en sortie. En mode entrée, chaque broche du PORTB doit être maintenue à un niveau haut par l'intermédiaire de résistances de 10 k pour ne pas déclencher d'interruptions imprévues. Cette possibilité d'interruption sur un changement d'état associé à la fonction de tirage configurable sur ces 4 broches, permet l'interfaçage facile avec

un clavier. Cela rend possible le réveil du PIC en mode SLEEP par un appui sur une touche du clavier [11].

- **Port C**

Le port C est partagé avec liaisons, les timers 1 et 2 et les modules CCP [11].

- **Port D et E**

En plus de leur utilisation comme PORTS E/S ; les ports D et E, permettent au microcontrôleur de travailler en mode PSP (Parallel Slave Port) c'est-à-dire, qu'il peut être interfacé avec un autre microprocesseur. Dans ce cas le PORTD représente le bus de données et le PORTE les signaux de contrôle (RD\, WR\ et CS\). Le PORTE peut être aussi, configuré en mode analogique pour former avec le PORTA les 8 entrées du convertisseur analogique numérique. Par défaut, le PORTE est configuré comme port analogique, et donc, comme pour le PORTA.

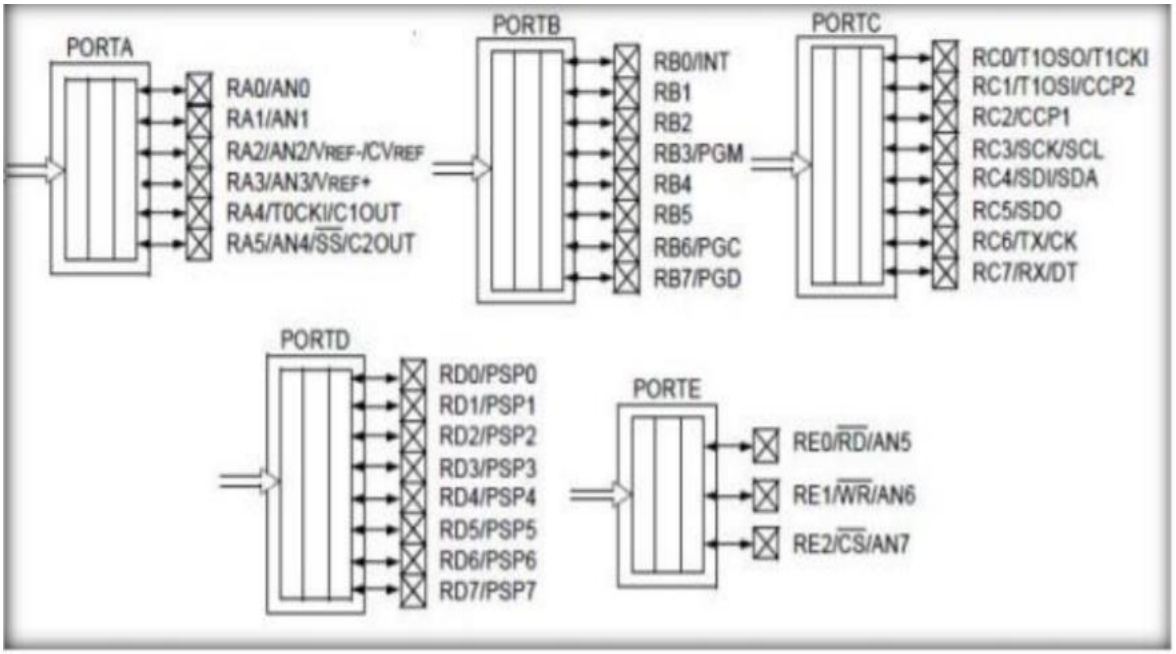


Figure II.8 : Les différents PORT de PIC16F877 [14].

II.6.2 Structure interne du PIC 16F877

On trouve à l'intérieur d'un microcontrôleur des éléments d'une structure à base d'un microprocesseur qui sont réunis dans un seul circuit intégré et ces éléments sont : [15]

- ✓ Une mémoire de donnée (RAM et EEPROM).
- ✓ Une mémoire de programme (FLASH, ROM, OTPROM, UVPROM ou EEPROM).
- ✓ Un chien de garde.
- ✓ 13 sources d'interruptions.
- ✓ Générateur d'horloge.
- ✓ Interface parallèle pour la connexion des entrées/sorties
- ✓ Interfaces séries (synchrone ou asynchrone).
- ✓ Les bus de données et d'adresses.
- ✓ D'autres périphériques (compteur/timer, convertisseurs analogiques/numériques (10 bits) à 8 canaux, etc).

Le schéma bloc de la figure ci-dessous, représente la structure interne du PIC 16F877

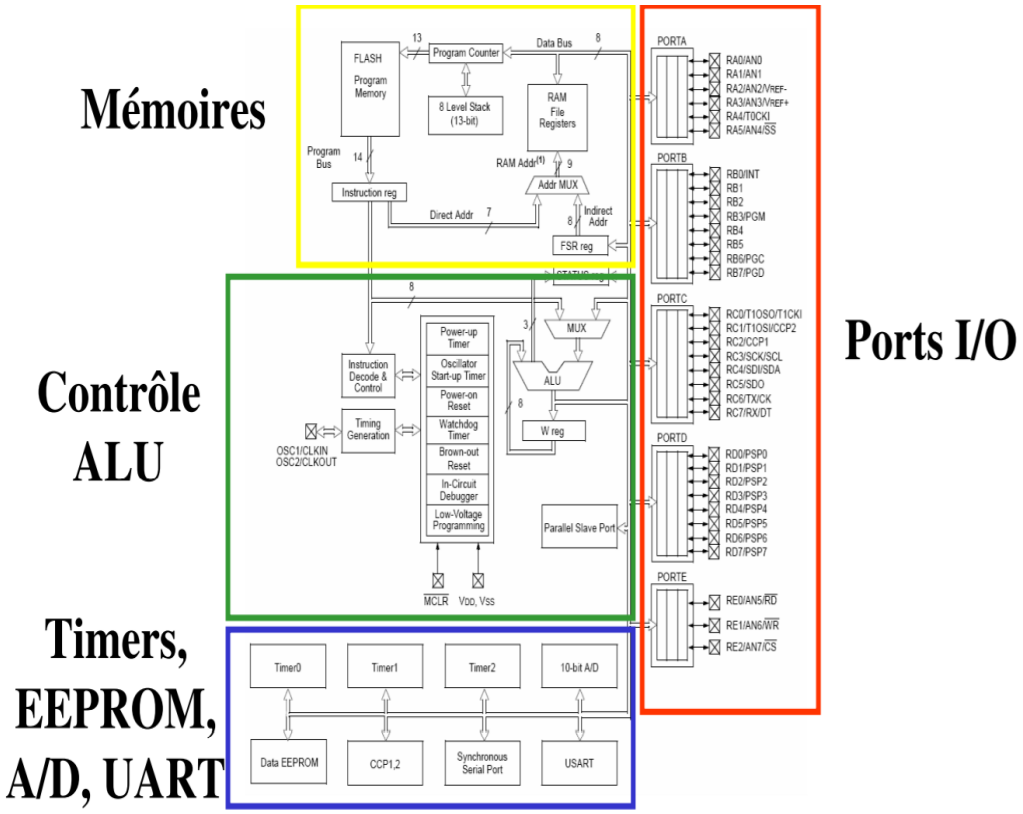


Figure II.9 : Structure interne du PIC 16F877.

II.6.2.1. Caractéristiques CPU

Un microcontrôleur, c'est avant tout un microprocesseur, c'est l'endroit où toutes les opérations arithmétiques et logiques contenant dans un microprogramme stocké en mémoire programme sont effectuées séquentiellement. L'unité centrale de traitement ou processeur contrôle les opérations internes du microprocesseur et envoie des signaux de contrôle à d'autres parties du microprocesseur pour exécuter les instructions requises [16].

- **ALU (**Unité arithmétique et logique) **:** C'est le cerveau du microcontrôleur, permettant d'effectuer des opérations entre l'accumulateur et une opérande, son fonctionnement dépend de la fréquence d'horloge.

II.6.2.2. les mémoires du PIC 16F877

La mémoire est une partie très importante du système de microcontrôleur, qui peut être classifié en deux différents types : mémoire de données et mémoire programme [15].

A- La mémoire programme :

C'est dans cette zone que nous allons écrire notre programme (contient les instructions du programme que doit exécuter le microprocesseur). Le PIC exécute une à une les instructions logées dans la mémoire de programme. Ce type de mémoires (appelé mémoire morte), est uniquement accessible en lecture. On retrouve :

i.EEPROM programme (flash) :

Cette mémoire de 8 x 1024 mots de 14 bits sert à stocker le programme, mais elle est accessible par programme et peut donc être utilisée comme une extension de la mémoire EEPROM de données. Elle est non volatile (c-à-d que les données sont conservées en cas de coupure de l'alimentation ; **flash**) et reprogrammable à souhait.

C'est une mémoire qui stocke le programme du microcontrôleur. Après compilation du fichier source, le compilateur génère un fichier **« .hex »** qui est transféré ensuite dans la mémoire programme du PIC à l'aide d'une interface appropriée (carte de développement ou de programmation). Cette mémoire n'est pas reliée au bus de données (DATA Bus), elle sert à stocker le programme du PIC mais pas les variables du programme.

B- La mémoire de donnée :

C'est une mémoire similaire à la mémoire programme. On s'en sert surtout pour stocker des constantes. Elle permet de mémoriser temporairement les données générées par le microprocesseur pendant les différentes phases du traitement numérique (résultats

d'opérations, états des capteurs...). Ces mémoires sont accessibles en écriture et en lecture. On en trouve 2 types :

i. De la mémoire morte (EEPROM):

Non-volatile, ayant un temps d'écriture assez élevé (quelques ms) par rapport au temps de lecture qui est assez faible (quelques ns). Elle est constitués de 256 octets, ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres semi-permanents.

ii. De la mémoire vive (RAM):

La mémoire RAM est celle qui est souvent utilisée. Elle est Volatile (données perdues en cas de coupure de l'alimentation) ayant un temps de lecture et écriture assez court (quelques ns), elle comprend tous les registres spéciaux permettant de contrôler le PIC ainsi que ses périphériques. Les variables des programmes pourront être stockées dans des cases mémoires à usage commun. La mémoire RAM disponible du 16F877 est de 368 octets. Le plan mémoire des données et des registres internes est découpé en 4 zones ou Bank, chaque zone est un ensemble de cases qui peuvent stocker une information sur 8 bits et une zone RAM utilisateur de 360 registres.

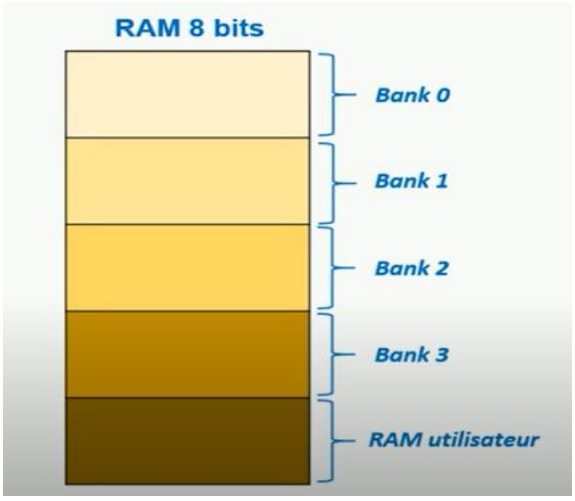


Figure II.10 : La configuration de la RAM.

Les registres contenants dans les Banks sont configurés comme suit :

Bank 0	
00h	INDF
01h	TMR0
02h	PCL
03h	STATUS
04h	FSR
05h	PORTA
06h	PORTB
07h	PORTC
08h	PORD (*)
09h	PORT (*)
0Ah	PCLATH
0Bh	INTCON
0Ch	PIR1
0Dh	PIR2
0Eh	TMR1L
0Fh	TMR1H
10h	T1CON
11h	TMR2
12h	T2CON
13h	SSPBUF
14h	SSPCON
15h	CCPR1L
16h	CCPR1H
17h	CCP1CON
18h	RCSTA
19h	TXREG
1Ah	RCREG
1Bh	CCPR2L
1Ch	CCPR2H
1Dh	CCP2CON
1Eh	ADRESH
1Fh	ADCON0

Bank 1	
80h	INDF
81h	OPTION_REG
82h	PCL
83h	STATUS
84h	FSR
85h	TRISA
86h	TRISB
87h	TRISC
88h	TRISD (*)
89h	TRISE (*)
8Ah	PCLATH
8Bh	INTCON
8Ch	PIE1
8Dh	PIE2
8Eh	PCON
8Fh	
90h	
91h	SSPCON2
92h	PR2
93h	SSPADD
94h	SSPSTAT
95h	
96h	
97h	
98h	TXSTA
99h	SPBRG
9Ah	
9Bh	
9Ch	
9Dh	
9Eh	ADRESL
9Fh	ADCON1

Bank 2	
100h	INDF
101h	TMR0
102h	PCL
103h	STATUS
104h	FSR
105h	
106h	PORTB
107h	
108h	
109h	
10Ah	PCLATH
10Bh	INTCON
10Ch	EEDATA
10Dh	EEADR
10Eh	EEDATH
10Fh	EEADRH

Bank 3	
180h	INDF
181h	OPTION_REG
182h	PCL
183h	STATUS
184h	FSR
185h	
186h	TRISB
187h	
188h	
189h	
18Ah	PCLATH
18Bh	INTCON
18Ch	EECON1
18Dh	EECON2
18Eh	
18Fh	

Figure II.11 : Configuration des registres dans la RAM [28].

II.6.2.3. Les timers du pic 16F877

Les Timers/compteurs sont des périphériques de gestion de temps. Ils permettent de réaliser les fonctions suivantes : [17]

- comptage des évènements.

- synchronisation des signaux.

- fixer le débit d'une liaison série synchrone ou asynchrone.

- génération des signaux périodiques (carré, MLI …).

- mesure de temps…

Les timers sont des compteurs formés généralement d'un pré-diviseur suivi d'un registre compteur de 8 ou 16 bits. L'entrée d'horloge peut être interne (mode timer) ou externe (mode compteur d'événements). Lorsque le registre compteur atteint sa valeur maximale et repasse à 0, un bit indicateur (flag) sera positionné et une interruption pourra être générée, informant ainsi la CPU du débordement du timer. Il faut bien noter que le programmeur devra remettre à zéro cet indicateur après chaque débordement. Le microcontrôleur PIC16F877 dispose de trois timers appelés Timer0, Timer1 et Timer2.

1-Timer0 :

Le timer 0 est un compteur 8 bits qui peut compter (de 0 à 255) :

- soit les impulsions de l'horloge via un pré diviseur

- soit des impulsions externes, via la broche PA4 Le débordement (over flow) du compteur, qui a lieu lorsque le compteur passe de 255 à 0, provoque une interruption. [11]

Le module TMR0 (timer0)possède deux modes de fonctionnement :

- Mode compteur : Compter les impulsions reçues sur le pin RA4/TOK1.
- Mode timer : Compter les cycles d'horloges du PIC. Dans ce cas, comme l'horloge est fixe, nous compterons donc en réalité du temps.

La sélection d'un ou l'autre de ces deux modes de fonctionnement s'effectue par le bit 5 du registre OPTION : T0CS [28]

- TOCS = 1 :Compteur.

Timer 0 s'incrémente à chaque front montant ou descendant du signal connecté à la patte RA4/TOCKI.

- TOCS = 0 : Timer.

Timer 0 s'incrémente à chaque cycle s'il n'y a pas de prédiviseur. En cas d'opération d'écriture dans le timer, l'incrémentation est inhibée pour les 2 cycles suivants.

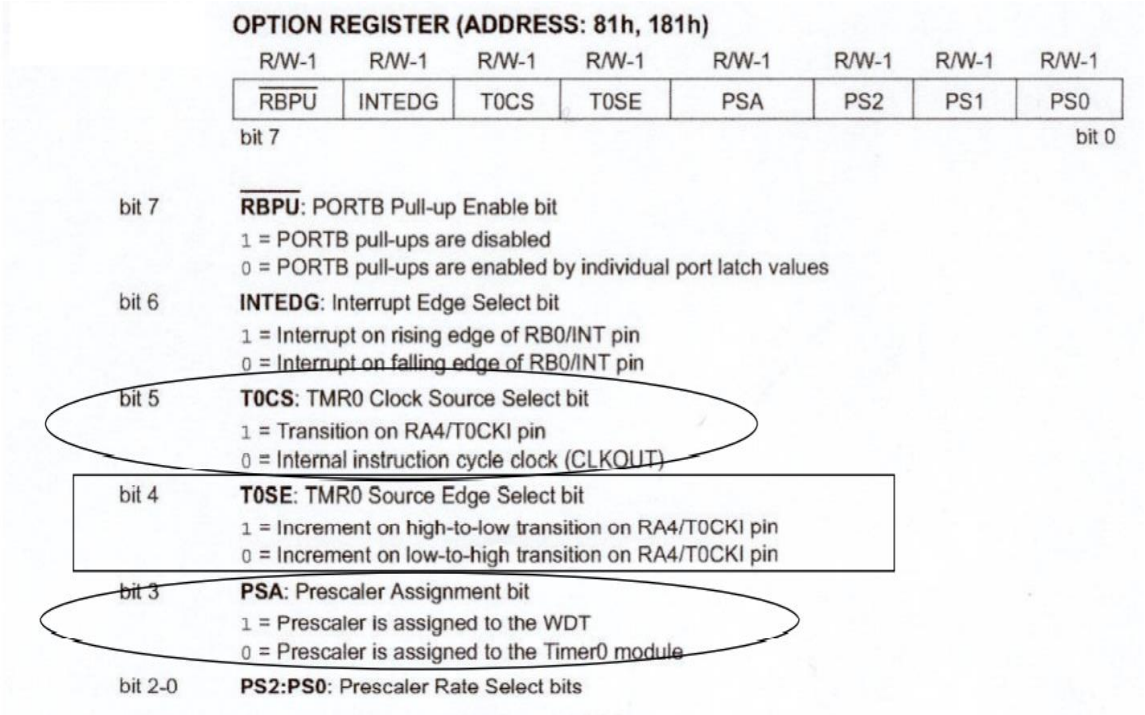


Figure II.12 : Registre option [28].

2-Timer1 :

Le timer 1 est un compteur 16 bits qui peut compter (de 0 à 65535) : - soit les impulsions de l'horloge - soit les impulsions externes, et en particulier les impulsions d'un quartz externe. Là aussi, le débordement provoque une interruption. [11]

3-Timer2 :

Le timer 2 est un timer couplé au module dit CCP. Il peut compter de 0 à 65535 comme le timer1. C'est ce timer que nous utilisons pour compter les durées entre l'émission et la réception des signaux venant de l'émetteur [11].

II.6.2.4. Watchdog

De nombreux microcontrôleurs disposent d'au moins un dispositif de surveillance, également appelé "watchdog tuner" (ou WDT). Un WDT est généralement un temporisateur de 8 bits avec une option de prédécalage et est synchronisé à partir d'un oscillateur à puce fonctionnant librement. Le chien de garde est généralement rafraîchi par le programme utilisateur à intervalles réguliers et une réinitialisation se produit si le programme ne parvient pas à le rafraîchir. Les dispositifs de surveillance sont généralement utilisés dans les systèmes en temps réel où il est nécessaire de vérifier la bonne fin d'une ou plusieurs activités. Tous les microcontrôleurs PIC sont équipés d'un WDT [18].

II.6.2.5. L'oscillateur

Tous les microcontrôleurs nécessitent une horloge (ou un oscillateur) pour fixer la vitesse d'exécution des instructions.

On utilise pour ce faire un quartz dont le rôle est de créer une impulsion de fréquences élevées. Dans le cas du 16F877, le quartz utilisé est typiquement un quartz de 8Mhz, c'est-à-dire qu'il va fournir 8 millions d'impulsions par secondes. Le temps qui s'écoule entre 2 impulsions s'appelle un cycle d'horloge .Un cycle d'horloge dure donc 125 nanosecondes (c'est-à-dire 125 milliardièmes de seconde) [11].

L'horloge peut être soit interne soit externe. L'horloge interne est constituée d'un oscillateur à quartz ou d'un oscillateur RC. [8]

- **RC :** une résistance et un condensateur pour une utilisation normale. Elle peut varier légèrement d'un circuit à l'autre, et dont n'est pas trop précis.

$$F=1/\tau,\ \tau=RC$$

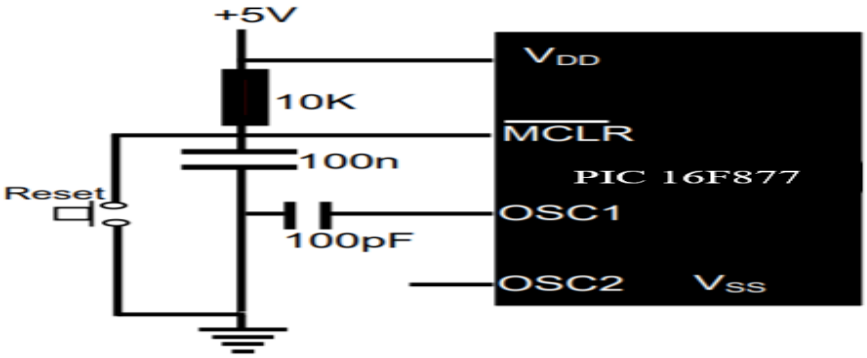


Figure II.13 : Pilotage par RC [20]

- **Systèmes à quartz :** Un quartz et deux condensateurs pour un pilotage précis, on peut avoir des fréquences allant jusqu'à 20 MHz, [8].

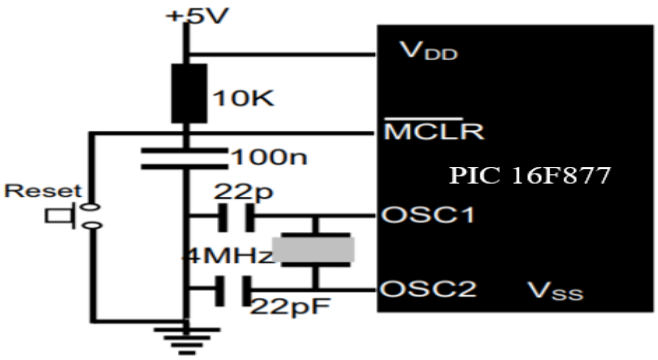


Figure II.14 : Pilotage par Quartz [20].

Le PIC 16F877 peut fonctionner en 4 modes d'oscillateurs, la sélection de l'un de ces modes est obtenue par la configuration des bits FOSC1 et FOSC0 [29].

Fosci1 : Fosc0	**MODE**
00	LP
01	XT
10	HS
11	RS

Tableau II.2 : Fonctionnement de l'horloge.

Pour le 16F877, nous allons considérer plusieurs modes d'oscillateurs :

Mode	Frequency	C1,C2 (pF)
LP	32 KHz	33
	200 KHz	15
XT	200 KHz	22–68
	1.0 MHz	15
	4.0 MHz	15
HS	4.0 MHz	15
	8.0 MHz	15–33
	20.0 MHz	15–33
	25.0 MHz	15–33

Tableau II.3 : Les modes d'oscillateurs [18].

Remarque : L'Unité de Calcul est le cœur PIC. Ici se déroulent toutes les opérations à une vitesse définie par la fréquence d'horloge (fréquence d'oscillation divisée par 4) [11].

$$F_{horloge} = \frac{F_{oscillateur}}{4}$$

Un microprocesseur exécute séquentiellement les instructions stockées dans la mémoire programme. Il est capable d'opérer sur des mots binaires dont la taille, en bits, est celle du bus des données (parfois le double pour certains microcontrôleurs) [11].

II.6.2.6. Les interruptions

A-Définition :

Une interruption est un signal déclenché par un événement interne à la machine ou externe, qui provoque l'arrêt immédiat d'un programme en cours d'exécution, et cela juste à la fin de l'opération courante, plus particulièrement dans un point que l'on appelle point observable ou interruptible du processeur (c'est un point entre l'exécution de deux opérations élémentaires), et cela au profit d'un programme plus prioritaire appelé programme d'interruption (ou Traitant de l'interruption).

Ensuite, le programme interrompu reprend son exécution à l'endroit où il avait été interrompu [24].

B-Mécanisme général :

Physiquement, l'interruption se traduit par un signal envoyé au processeur. Elle permet de forcer le processeur à suspendre l'exécution du programme en cours, et à déclencher l'exécution d'un autre programme prédéfini, spécifique à l'événement (appelé la cause de l'interruption) et qui permet d'exécuter le traitement de l'interruption ; ce dernier peut avoir plusieurs appellations: Traitant de l'interruption (Interrupt-Handler) ou Rit (Routine d'interruption), ou encore ISR (Interrupt Service Routine). Le traitant de l'interruption fait d'abord une sauvegarde du contexte du processus interrompu avant de réaliser son propre traitement. A la fin de celui-ci, le contexte du processus interrompu est restauré ce qui lui permet de continuer son exécution convenablement à l'endroit où il a été interrompu [24].

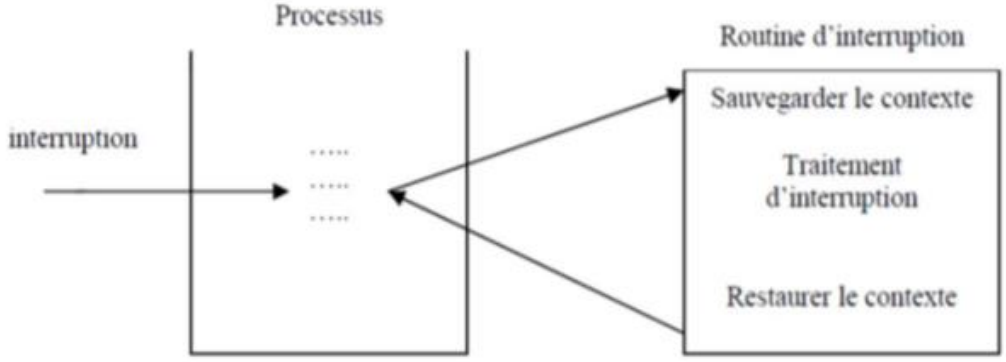


Figure II.15 : Déroulement d'une routine d'interruption.

C-Catégories d'interruptions :

Les interruptions peuvent être des origines diverses, mais on peut les classer en deux grandes classes (ou catégories) : [24]

− Les interruptions externes ou matérielles.

− Les interruptions internes ou logicielles.

- **Interruptions externes :** (indépendantes du processus) : Ce sont les interruptions causées par des organes externes au processeur central, comme les horloges de temps, les périphériques d'E/S, …etc [23].
- **Interruptions internes :** Ce sont des interruptions causées par des anomalies dans l'exécution du programme en cours protection du système et des processus, appelées par une instruction à l'intérieur d'un programme (erreur d'adressage, code opération inexistant, problème de parité…) [23].

Le PIC16F877A 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.

D-Le registre INTCON :

Ce registre permet d'effectuer les interruptions et de stocker les indicateurs d'interruptions comme indiquer ci-dessous [30].

- INTCON.7 = GIE : Permet d'activer ou de désactiver (0) les interruptions de façon globale.
- INTCON.6 = PEIE : Permet d'activer ou de désactive (0) toutes les interruptions externes.
- INTCON.5 = T0IE : Permet d'activer ou de désactive (0) l'interruption liée au TMR0.
- INTCON.4 = INTE : Permet d'activer ou de désactive (0) l'interruption liée à la broche RB0.
- INTCON.3 = RBIE : Permet d'activer ou de désactive (0) l'interruption liée au changement d'état sur l'une des broches de RB7 à RB4.
- INTCON.2 = T0IF : 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.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GIE	PEIE	TMROIE	INTE	RBIE	TMROIE	INTF	RBIF

Tableau II.4 : Le registre INTCON.

II. 6.2.7 Convertisseur analogique-numérique

La fonction conversion analogique-numérique consiste à transformer une grandeur électrique en une grandeur numérique exprimée sur N bits qui pourra être utilisé par un programme [19].

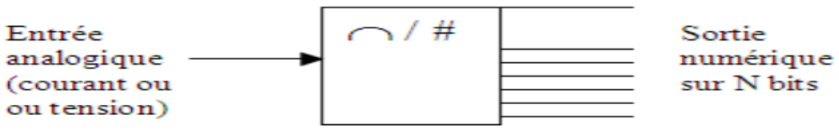


Figure II.16 : Schéma convertisse analogique/numérique.

Notre 16F877 travaille avec un convertisseur analogique/numérique, Ce module est constitué d'un CAN 10 bits dont l'entrée analogique peut être connectée sur l'une des 8 entrées analogiques externes (RA0-RA5, RE0-RE2). On dit qu'on a un CAN à 8 canaux. Les entrées analogiques doivent être configurées en entrée à l'aide des registres TRISA et/ou TRISE.

Le PIC dispose d'un échantillonneur bloqueur intégré constitué d'un interrupteur S, d'une capacité de maintien C=120 pF et d'un CAN 10 bits. Pendant la conversion, la tension Ve à l'entrée du convertisseur A/N doit être maintenue constante [8].

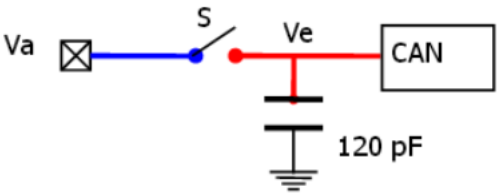


Figure II.17 : Constitution du CAN.

Remarque : La gestion de la conversion se fait grâce à 4 registres 8 bits : [28]

- ADRESH et ADRESL (Analog to Digital result High and Low), contiennent le résultat de la conversion sur 10 bits.

- ADCON0 et ADCON1 (Analog to Digital Control 0, 1), permettent de configurer la conversion.

II 6.2.8. Le reset

Il existe différentes situations pouvant entraîner un reset du circuit : [22]

- **Power on reset :** la tension d'alimentation descend en dessous du seuil de 1,2 à 1,7v.
- **Brown on reset :** la tension d'alimentation reste vers un niveau de 4 v pendant un temps assez long ·
- **Watcdog reset :** action du chien de garde ·
- **MCLR/ activé :** action sur l'entré e MCLR.

Ceci provoque l'arrêt du programme qui va recommencer à la première instruction. Pour que le microprocesseur fonctionne, il faut donc que cette broche soit connectée au +5V en permanence.

II.7. Conclusion

Nous avons traité tout au long de ce chapitre l'élément que nous avons jugés importants pour notre travail à savoir ses caractéristiques, son architecture interne et externe, …etc. Partant de cette étude, on peut déduire que le microcontrôleur 16F877 peut bien jouer le rôle d'une unité de contrôle pour notre montage et maintenant nous pouvons passer à la conception et la réalisation, puisque le composant le plus important dans notre système nous est déjà familier.

Chapitre III

Conception et réalisation Pratique

III.1. Introduction

L'étude théorique que nous avons donnée dans les chapitres précédents, nous facilitera la réalisation de notre dispositif expérimental intitulé « conception et réalisation d'un calendrier numérique réglable à base de PIC 16F877 ».

Le système proposé dans ce projet, a pour but d'afficher la date et l'heur et pouvoir les régler, on note que pour la réalisation de ce calendrier numérique on s'est reposé sur le périphérique TMR0 et le système interruption sur la première broche du port B RB0.

Dans ce chapitre on va expliquer d'une manière simple le fonctionnement d'horloge numérique réglable à base de PIC 16F877 de façon clair et méthodique. On va décrire brièvement les différentes étapes de la réalisation du projet jusqu'à la réalisation du circuit imprimé.

Notre réalisation pratique a été répartie en trois parties :

Partie I : Description du système

- Schéma synoptique.
- Présentation des blocs.
- Fonctionnement.

Partie II : Simulation du circuit

- Schéma électronique.
- Programmation du PIC.
- Routage et Création du circuit imprimé.

Partie III : Réalisation pratique

- Liste des composants du circuit.
- La fabrication du circuit imprimé .
- Gravure du programme sur le pic.

III.2. Description du système

III.2.1. Schéma synoptique

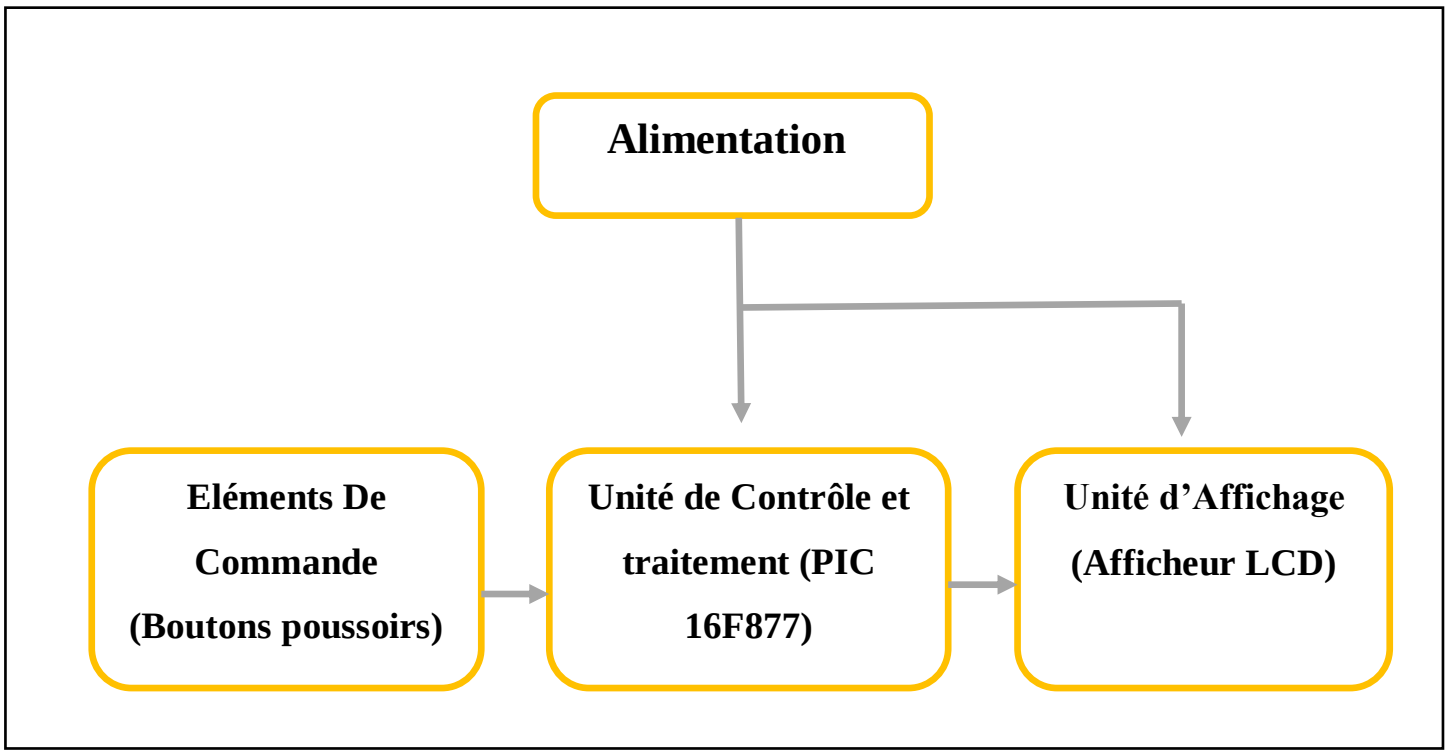


Figure III.1 : Schéma synoptique du circuit.

III.2.2. Présentation des blocs

Le schéma de la figure ci-dessus donne une vue générale sur le principe de fonctionnement de notre montage, il est articulé par quatre principales parties fonctionnelles :

- **Alimentation :** stabilisée de 5V, a pour but l'alimentation des différents blocs du circuit électronique.

- **Eléments de commande :** boutons poussoirs.

- **Unité de contrôle de traitement :** PIC16F877.

- **Unité d'affichage :** Afficheur LCD.

III.2.3. Fonctionnement

- **Alimentation :**

Les montages de base nécessaires pour la mise en marche de la réalisation, sont les alimentations à faible puissance pour des circuits électroniques. Le fonctionnement de notre pic et ses périphériques exige une tension d'alimentation de 5V, pour cela on a réalisé une alimentation par une pile 9V suivie d'un régulateur de tension du type LM7805.

La figure suivante représente les pins du régulateur LM7805.

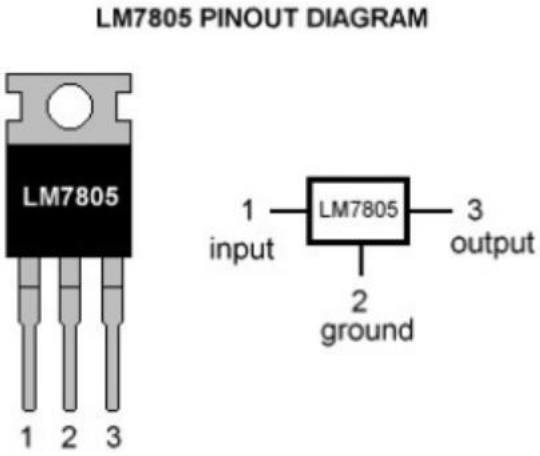


Figure III. 2: Régulateur LM7805.

Le régulateur LM7805 permet de réguler et stabiliser la tension d'entrée à une valeur fixe pour n'importe quel courant de sortie ou charge, résultant ainsi en sortie la tension désiré (5v).

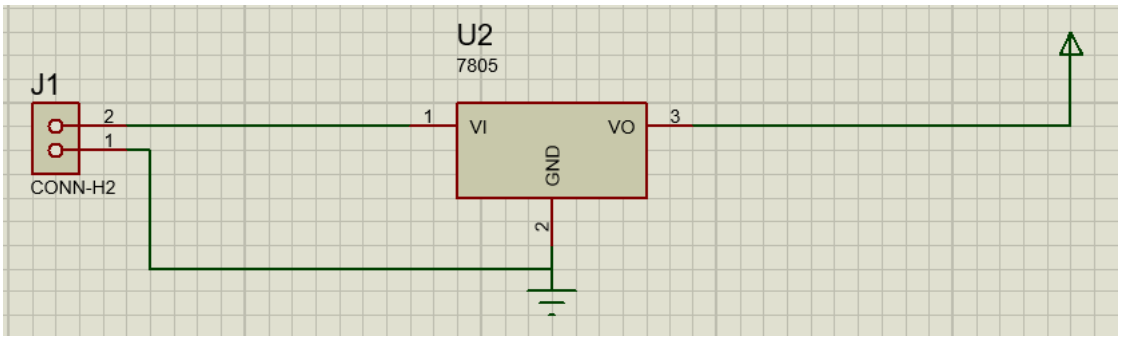


Figure III.3: Schéma électrique du circuit d'alimentation 5V sous PROTEUS ISIS.

- **Unité de contrôle et traitement :**

Le choix d'un microcontrôleur est primordial car c'est de lui que dépendent en grande partie les performances grâce à ca taille et ca facilité d'utilisation en programmation et simulation, il possède en plus des instructions très puissantes donc un programme à développer réduit et une programmation simple.

A cet effet, on a utilisé le PIC 16F877 pour la gestion de la communication et de traitement de données dans nôtres mémoire.

Pour le fonctionnement du pic on a besoin de :

- une alimentation de 5 Volts.

- un quartz et deux condensateurs.
- un bouton poussoir et une résistance, pour la mise en place d'une commande de Reset.

✓ **Alimentation :**

L'alimentation est assurée par les pattes VDD et VSS. Elles permettent à l'ensemble des composants électroniques du PIC de fonctionner. Pour cela on relie VSS (Broche 12 ou 31) à la masse et VDD (Broche 11 ou 32) à la borne positive de l'alimentation qui doit délivrer une tension continue de 5V.

✓ **Cadencement du PIC 16F877 :**

L'oscillateur du microcontrôleur cadence le déroulement du programme. Le PIC 16F877 est Piloté par une horloge qui permet de générer des fréquences allant jusqu'à 20 MHz. Cette dernière est constituée d'un oscillateur à quartz et de deux capacités C1 et C2 qui limitent les harmoniques dus à l'écrêtage et réduit l'amplitude de l'oscillation.

Pour le 16F877, nous allons considérer plusieurs types d'oscillateurs :

- **LP** : ce sont les oscillateurs de basse fréquence, en dessous de 200KHz.
- **XT** : les oscillateurs de fréquence moyenne, entre 200KHz et 4MHz.
- **HS** : les oscillateurs de haute fréquence, entre 4MHz et 25MHz.

En mode LP, XT ou HS, un quartz ou un résonateur en céramique est connecté aux pins OSC1/CLKIN et OSC2/CLKOUT pour établir l'oscillation.

Dans le cas de notre application on a utiliser l'oscillateur de type HS avec un quartz de 8MHz connecté aux pattes OSC1/CLKIN (patte 13) et OSC2/CLKOUT (patte 14).

L'oscillateur étant divisé par 4, on aura une horloge cycle maximum de 2 MHz soit un temps de cycle de 500 ns.

D'où notre oscillateur comporte :

- 1 Quartz 8Mhz.
- 2 Capacités de 22 pF.

La figue suivante représente le schéma oscillateur sous le logiciel ISIS.

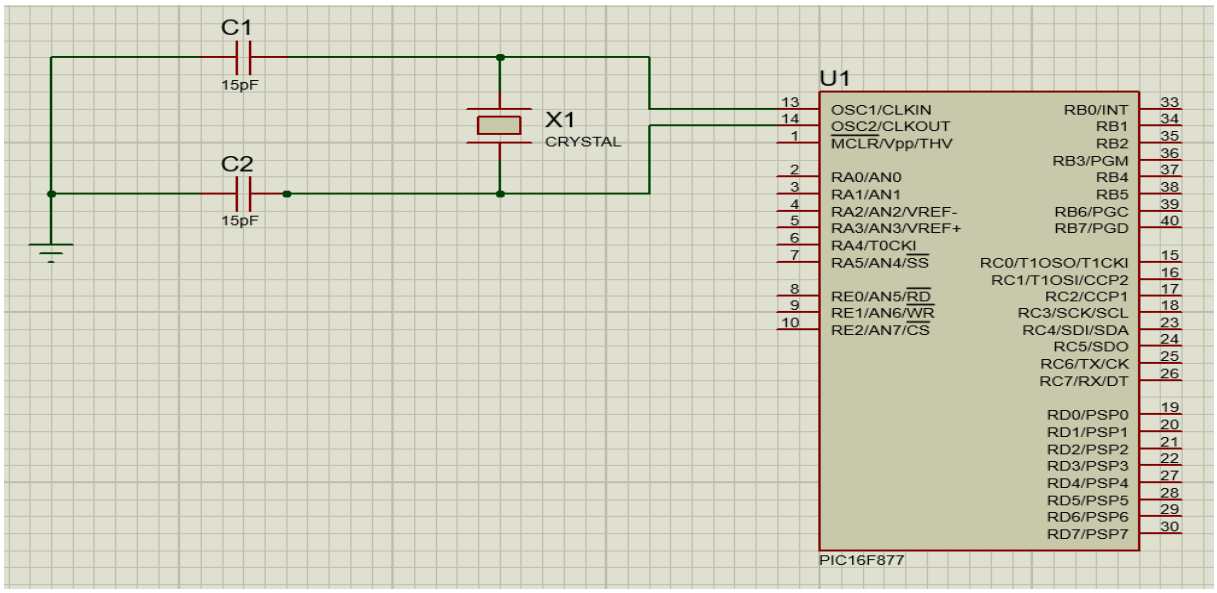


Figure III.4 : Le Circuit d'Oscillateur sous PROTEUS ISIS.

➢ **Circuit Reset MCLR :**

Le PIC16F877 possède une broche MCLR (Master Clear) ayant le rôle de provoquer une réinitialisation du CPU quand elle est connectée à zéro dans ce cas-là, un signal RESTE est activé.

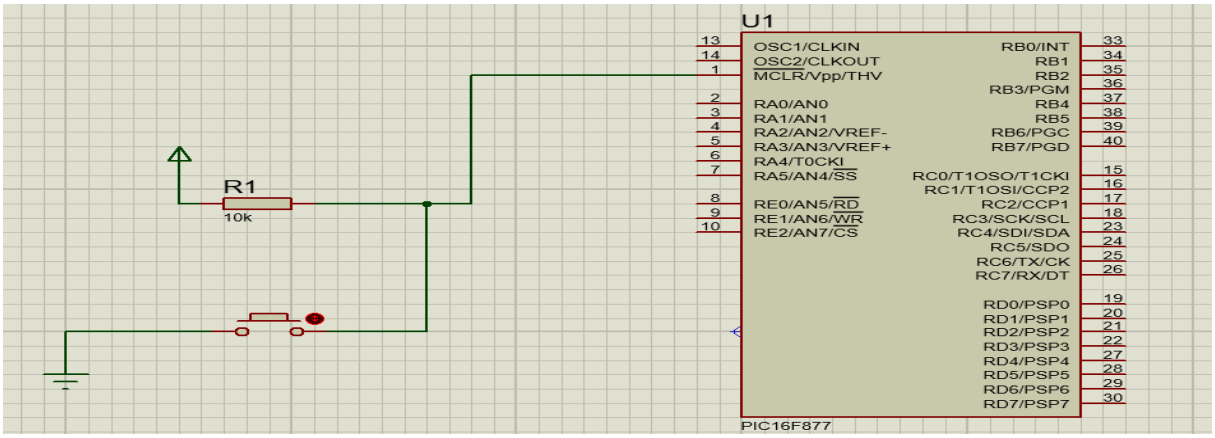


Figure III.5 : Circuit Reset MCLR sous PROUTEUS ISIS.

➢ **Branchement des ports sorties / entrées :**

PIC16F877 permet le contrôle et l'affichage des éléments sur l'afficheur LCD. Il présente 40 broches, 20 de chaque côté. Les broches sont virtuellement numérotées de 1 à 40, et dispos 5 ports.

Le logiciel MikroC possède une bibliothèque qui gère l'afficheur LCD, cette bibliothèque fonctionne en mode 4 bits et dispose de plusieurs fonctions qui manipulent l'initialisation, l'écriture et la transmission des données.

Nous avons interfacé le PIC 16F877 sur le PORT B possédant 8 broches (nommées RB0 à RB7).

- la broche **RB0** peut également servir comme interruption éventuelle.
- La broche **RB1** est une sortie destinée à commander une entrée de contrôle RS (Registre Select).
- La broche **RB2** est une sortie destinée à la validation (EN), qui est utilisée pour initier le transfert de commandes ou de données entre le module et le microcontrôleur.
- Les broches **RB3** à **RB6** sont des lignes de bus de données (D4 à D7). Pour communiquer avec l'écran LCD. Les données peuvent être transférées entre le microcontrôleur et le module LCD.

Nous avons branché l'afficheur LCD sur le port B de notre microcontrôleur comme suit :

16F877A	RB1	RB2	RB3	RB4	RB5	RB6
LCD	**RS**	**E**	**D4**	**D5**	**D6**	**D7**

Tableaux III.1 : Bronchement du PORT B du PIC 16F877.

➢ **Programmation du pic 16F877 :**

La programmation des microcontrôleurs PIC est supportée par plusieurs langages de programmation tel que : MikroC for PIC, MPLAB, MikroBasic PRO for PIC, HI-TECH C for PIC, flow code...etc. Dans notre projet nous avons opté pour le compilateur de MikroC qui est un compilateur en langage C (langage évolué).

Le MikroC est un puissant outil pour les microcontrôleurs PIC. Il est conçu pour fournir au programmeur la solution le plus simple possible pour développer des applications pour les systèmes embarqués, sans compromettre les performances ou le contrôle. Le MikroC permet de développer et déployer des applications complexes.

❖ **Programme principal :**

Le programme principal est constitué d'un sous-programme destiné à gérer le processus du réglage. Pour faciliter la compréhension et le déroulement du programme, nous avons choisi

de le représenter dans un organigramme qui ne représente que les actions principales effectuées par le microcontrôleur.

L'organigramme (figure III.6) ci-dessous indique une représentation générale du programme d'affichage et du réglage.

Figure III.6 : Organigramme principale.

❖ **Unité d'Affichage (Afficheur LCD) :**

Nous avons utilisé un afficheur LCD de type accès parallèle (transmet les informations à afficher sous forme parallèles avec une possibilité de transmission sur quatre ou huit bits).

L'écran est alimenté en 5 v par la carte d'alimentation, le LCD utilisé de 2x16 caractères.

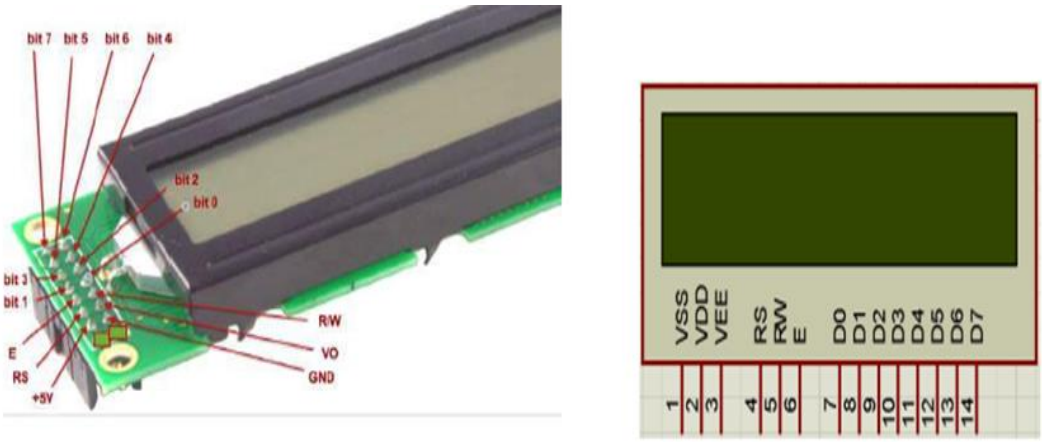


Figure III. 7: Les broches de l'afficheur LCD.

➢ **Branchement Afficheur LCD :**

Pour commander l'afficheur LCD, il existe deux modes de branchement : le mode 8 bits et le mode demi-octet (4bits). Pour notre application nous allons utiliser le mode 4 bits.

- **Mode 4 bit** :

Ce mode ne demande que 4 broches, ce qui permet de diminuer le nombre de fils utilisés pour commander l'afficheur et de libérer des pins sur le microcontrôleur, dans ce mode seuls les 4 bits de poids fort (D4 àd7) de l'afficheur sont utilisées pour transmettre les données et les lire, les 4 bits de poids faible sont alors non connectés.

Voici les broches présentes sur le module :

- **La patte VSS** est reliée à **0**V.
- **La patte VDD** doit être connectée à la borne positive d'alimentation. Bien que les fabricants spécifient une alimentation 5**V** DC.
- **La patte 3 VEE** est désignée pour le réglage du contraste de l'affichage et doit être reliée à une alimentation en courant continu.
- **La patte 4** est le registre de sélection **(RS)** et lorsque cette patte à **0**V, les données sont transférées à l'affichage. Lorsque **RS** est a **+5** V, les données de caractères peuvent être transférées à partir du module LCD.

- **La patte 5** est le registre de sélection de lecture / écriture (**R** / **W**). Cette patte est reliée avec la masse (état logique bas) afin d'écrire des données de caractères au module LCD.
- **La patte 6** est la validation **(EN)**, qui est utilisée pour initier le transfert de commandes ou de données entre le module et le microcontrôleur.
- **Les pattes 7 à 14** sont les huit lignes de bus de données (**D0** à **D7**). Les données peuvent être transférées entre le microcontrôleur et le module LCD à l'aide soit d'un seul octet de 8 bits soit de deux 4-bits. Dans notre cas, seules les quatre lignes de données (**D4** à **D7**) sont utilisées. Le 4-bits mode a l'avantage de nécessiter moins de lignes d'E/ S pour communiquer avec l'écran LCD.

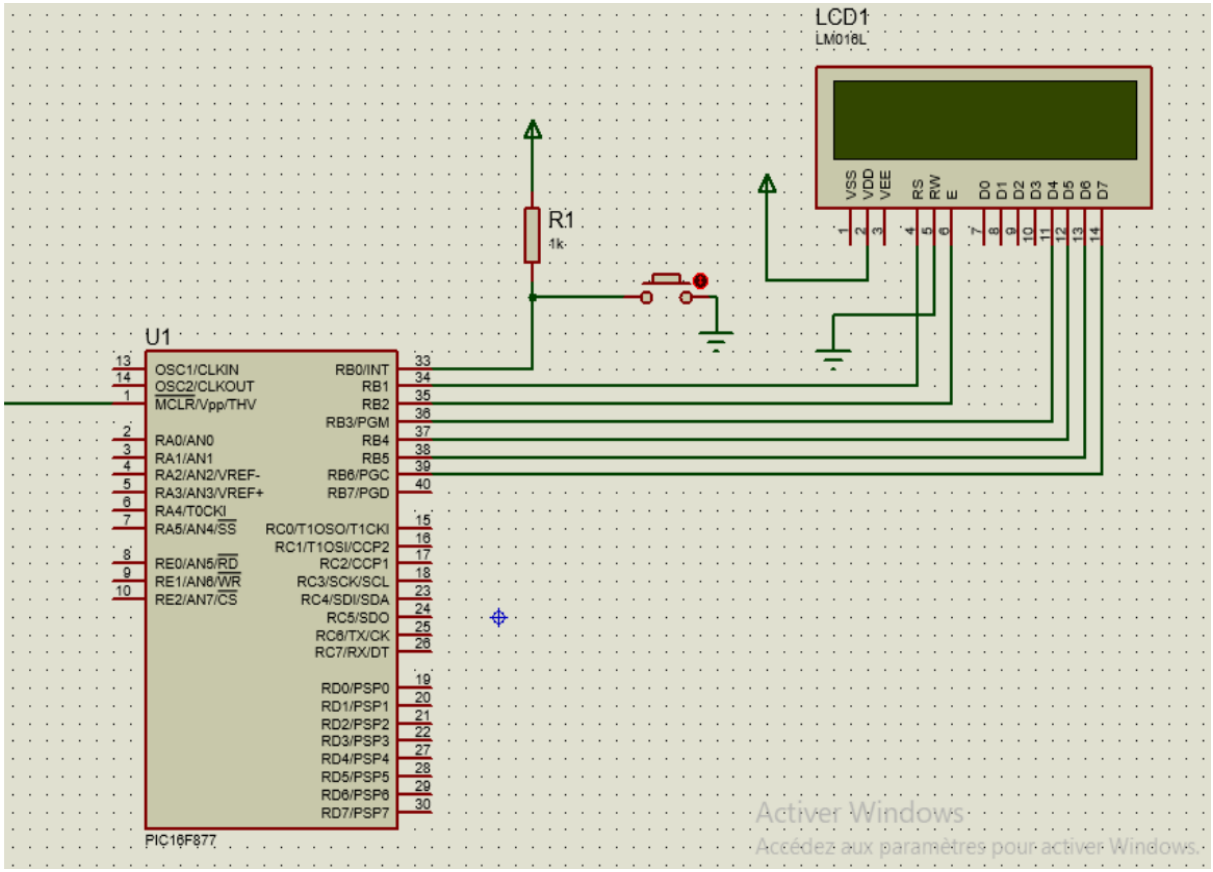


Figure III.8 : Schéma du branchement afficheur LCD sous PROTEUS ISIS.

III.3. Simulation du montage

Le Logiciel de simulation des circuits électroniques est un logiciel qui dessine un circuit par les composants et les circuits intégrés et permet de voir les résultats de la réalisation pratique, il existe plusieurs simulateurs : Work bench, Multi Sim, PROTEUS, Tina…etc. Pour notre cas on a utilisé PROTEUS. Le logiciel PROTEUS se compose de deux parties, le

logiciel ISIS pour la simulation des circuits électroniques, et le logiciel ARES pour dessiner les circuits imprimés.

Le but du logiciel ISIS est de dessiner, simuler des circuits électroniques et tracer les courbes. La deuxième partie le logiciel ARES ; est un logiciel permettant le routage de cartes électroniques en mode automatique ou manuel. Il est possible d'utiliser ARES sans créer du schéma électronique dans l'ISIS. Cette fonctionnalité permet de réaliser très rapidement des circuits de faible complexité en plaçant les composants et traçant les pistes directement dans ARES. Une fois les connections établies il est possible d'effectuer un routage automatique des pistes.

III.3.1. Schéma électronique

Pour la création du schéma électronique de notre montage on a utilisé le logiciel ISIS Proteus qui doit être sélectionnés à partir de la bibliothèque des composants sur la zone de travail.

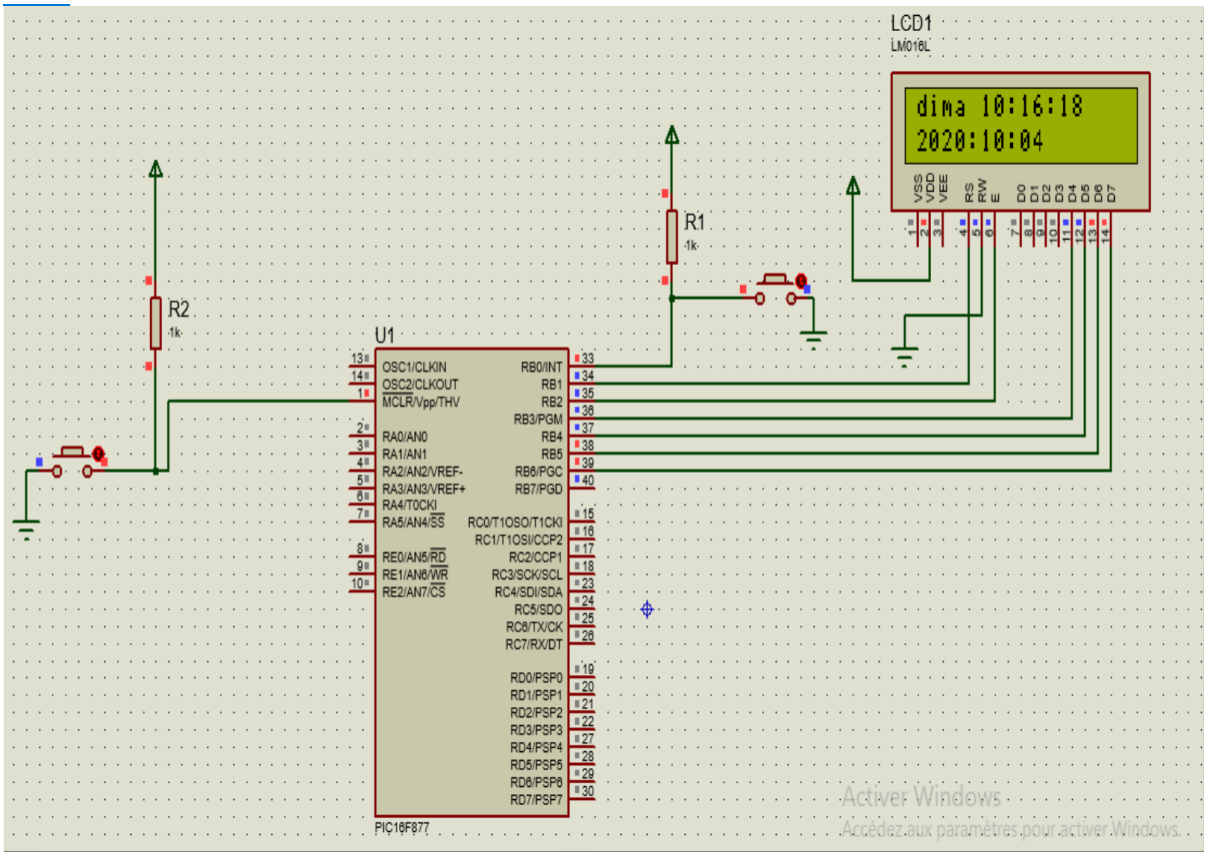


Figure III.9 : Schéma électronique du calendrier numérique réglable.

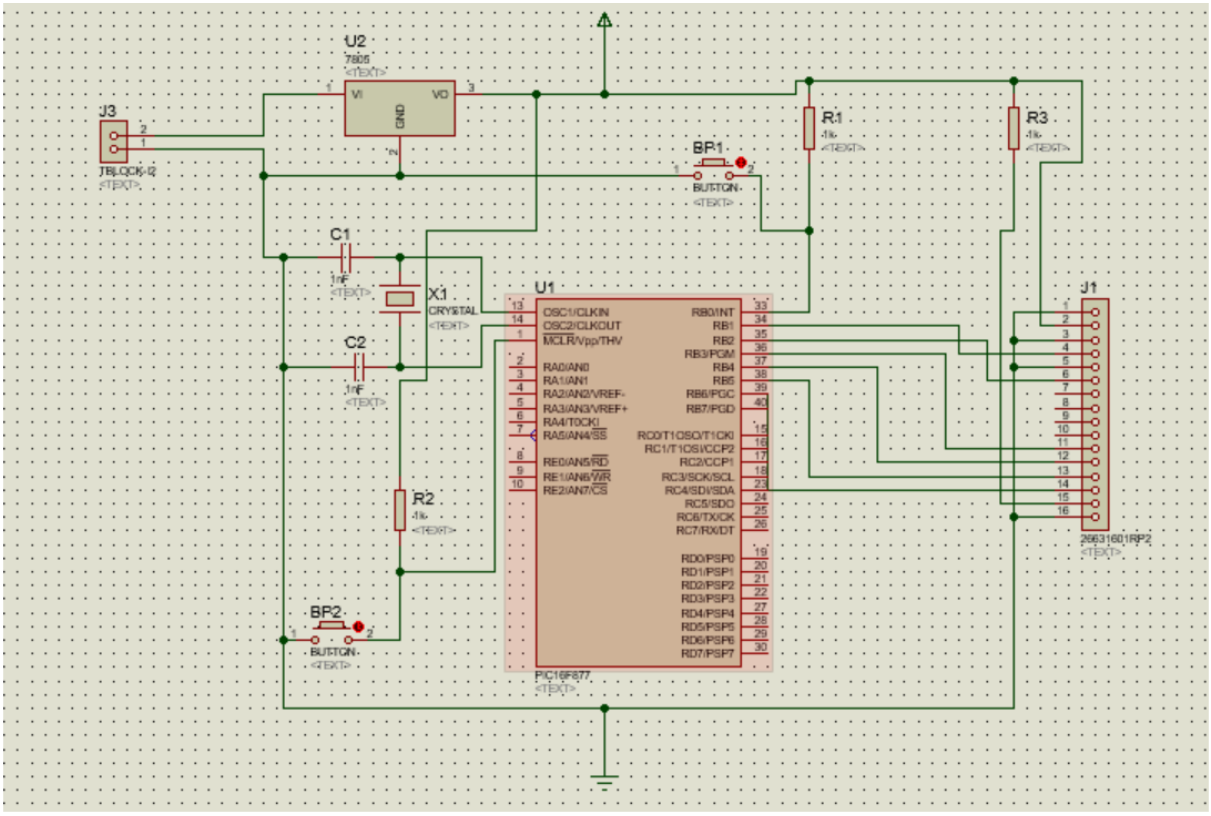


Figure III.10 : Schéma électronique globale du calendrier numérique réglable.

III.3.2. Programmation du PIC 16F877

Pour la simplicité et la facilité de la programmation, plusieurs langages ont été évolués dans le temps. En cherchant le compilateur le plus adapté aux microcontrôleurs PIC, on trouve le MikroBasic, MikroC et MikroPascal, et pour la simulation on trouve le logiciel PROTEUS ISIS.

On a utilisé le compilateur MikroC pro, Pour créer un nouveau projet, saisir et compiler un programme on suit les étapes suivantes :

- Lancement le compilateur mikroC PRO for PIC en cliquant sur l'icône de logiciel.

Création d'un nouveau projet :

- Création d'un nouveau projet on cliquant « New Project ».

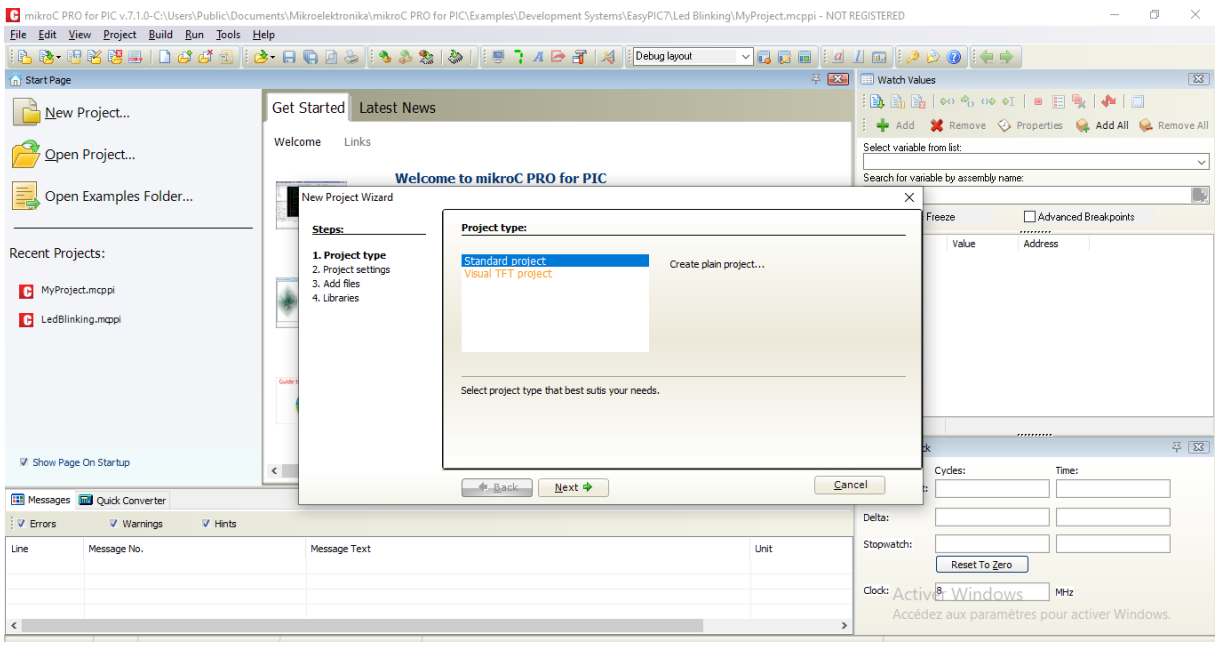


Figure III.11 : création d'un nouveau projet.

Commencez à créer nouveau projet, en cliquant sur le bouton **Next.** Par la suite remplir le '**Project Name'**, puis choisir le pic '**Device Name**'; puis sélectionner la fréquence du signal d'horloge '**Device Clock'**.

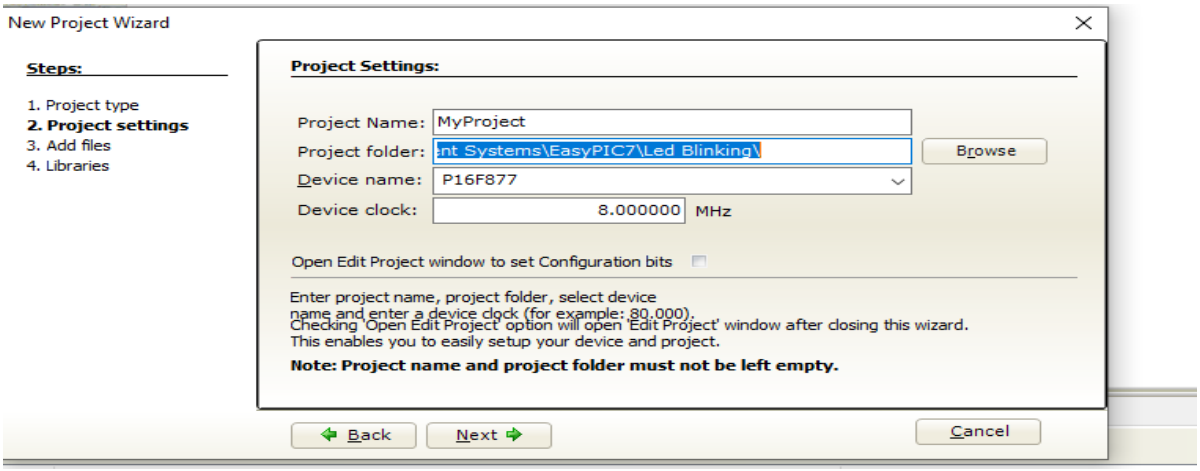


Figure III.12 : choix de fréquence et le pic.

L'option suivante permet de définir le répertoire où le développeur enregistrera le projet, dans ce répertoire le programme enregistrera tous les fichiers nécessaires, parmi lesquels le code source qui sera archivé avec l'extension *.c*, et l'exécutable du PIC avec l'extension (*hex).*

Enfin, la configuration est terminée et le projet est créé, à la fin la fenêtre doit apparaître ou on peut écrire le programme tout simplement comme suit :

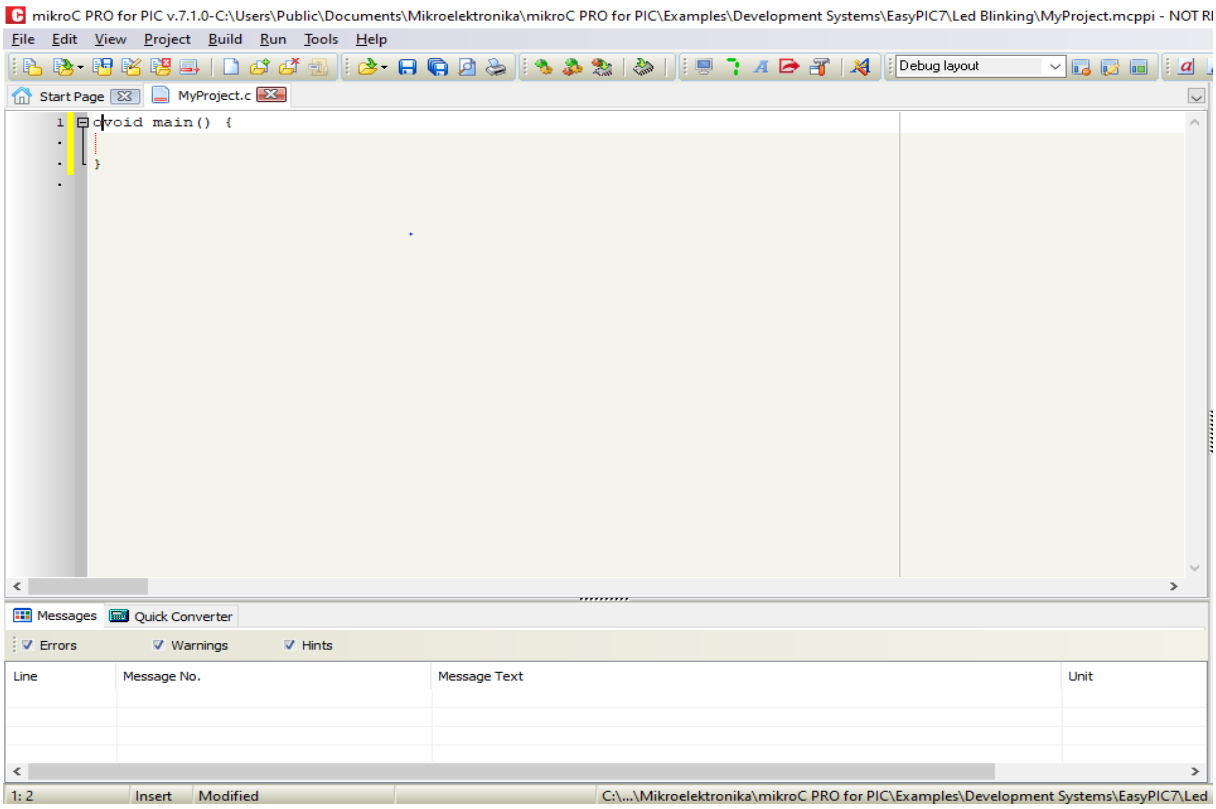


Figure III.13 : Exemple d'un programme.

Après la création du programme, on doit associer ce programme au pic en passant par les étapes suivantes :

- Aller au logiciel ISIS.
- Double clique sur pic 16F877

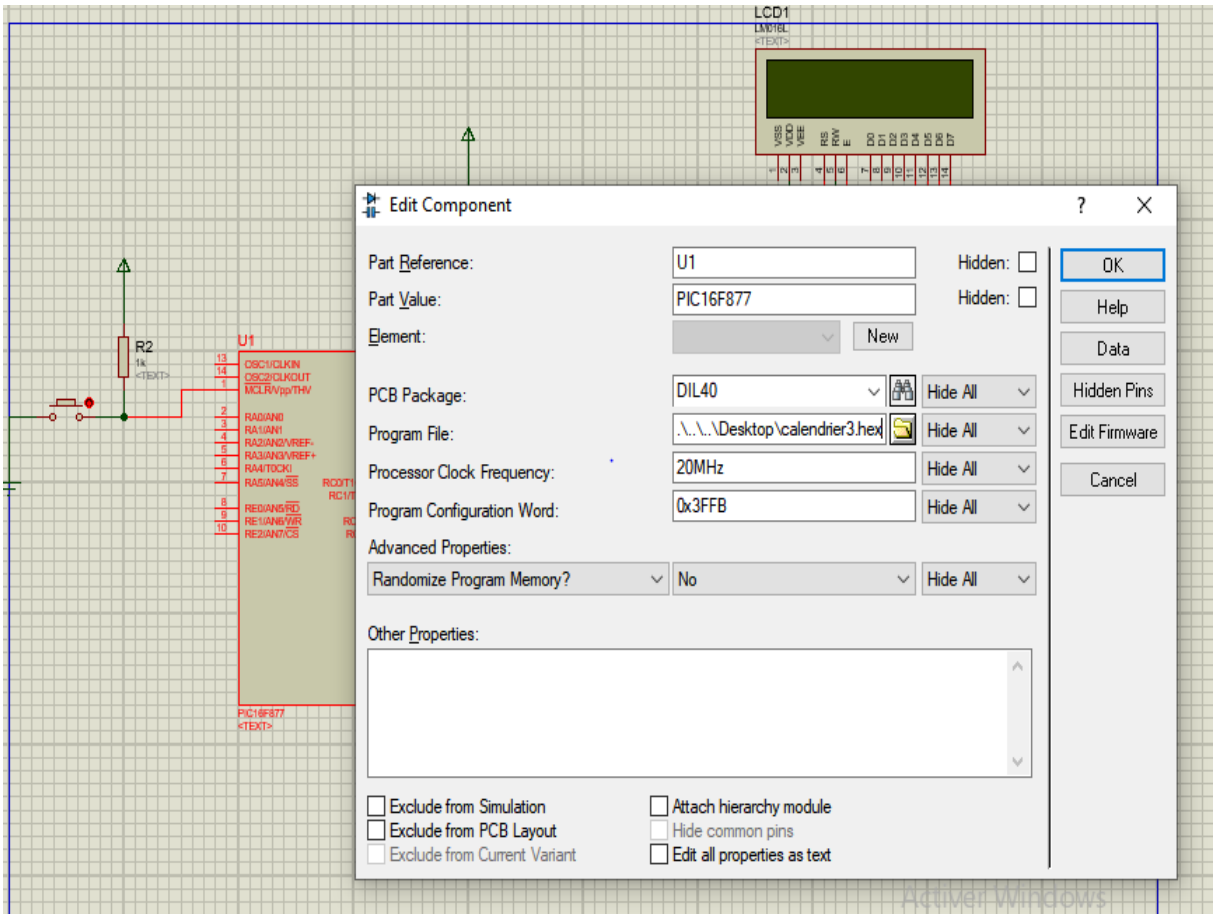


Figure III.14 : Saisie d'un programme sur ISIS Proteus.

- Entrez sur (program file) et apportez le fichier (file.HEX) et appuyez sur ok
- Le programme est chargé dans le PIC16F877.
- Assurez-vous que tous les fils sont connectés.
- Cliquez sur le bouton (Play) pour commencer la simulation.

III.3.3 : Routage et création du circuit imprimé :

Avant la réalisation matérielle de notre carte, nous devons réaliser le schéma en forme de circuit imprimé. Pour se faire, nous avons utilisé l'ARES sous PROTEUS pour simuler les connexions de tous les composants implantés sur notre carte.

Les figures III.15, III.16 et III.17 représentent respectivement la page de démarrage de l'ARES et le dessin du montage de notre appareil sur ce logiciel et le circuit du calendrier numérique réalisé sous ARES en visualisation 3D.

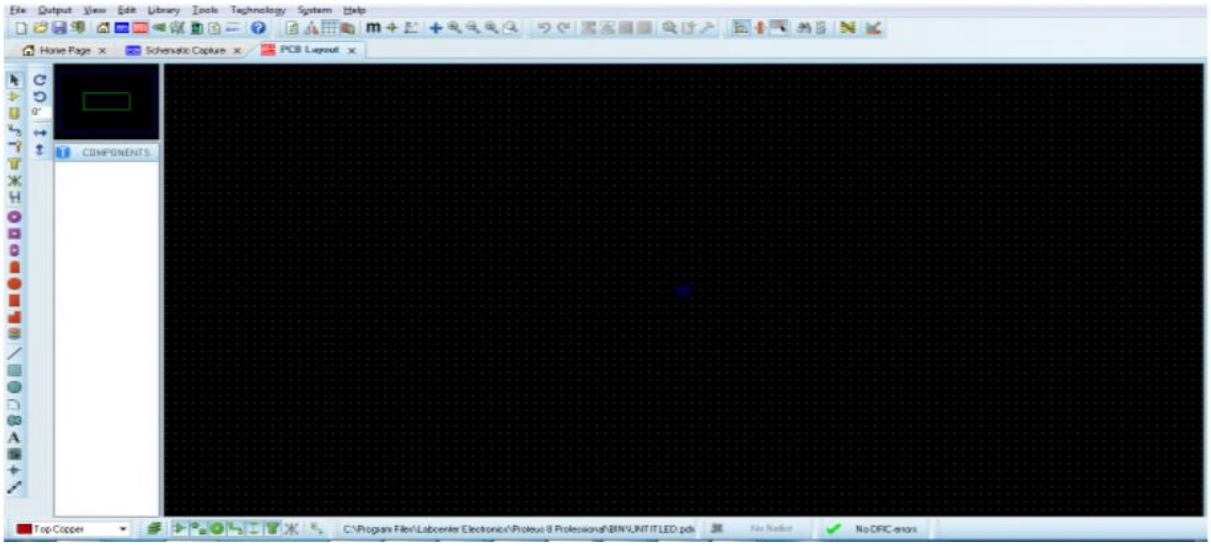


Figure III.15 : Illustration du démarrage de l'ARES.

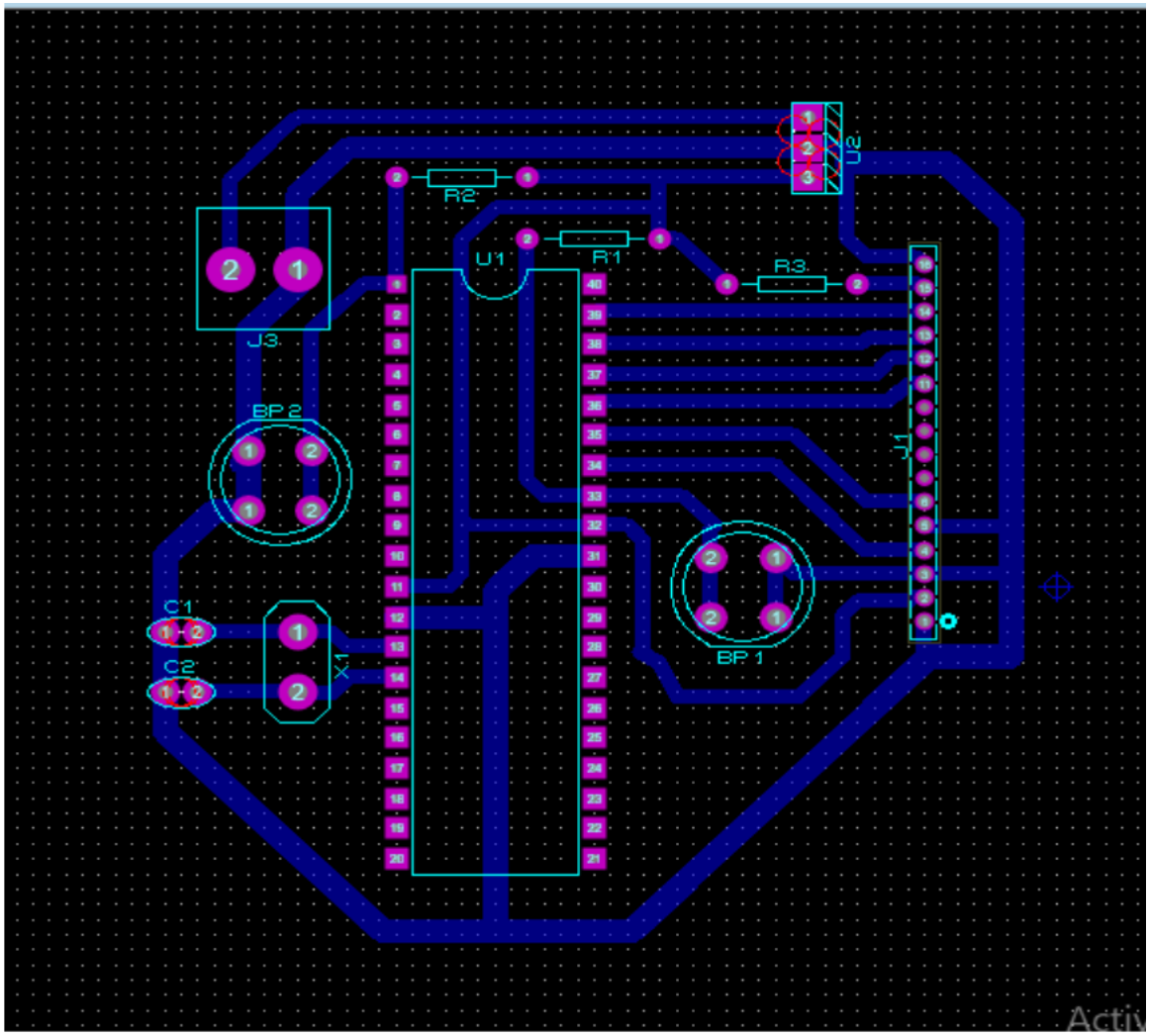


Figure III.16 : Schéma du circuit imprimé final sous logiciel ARES.

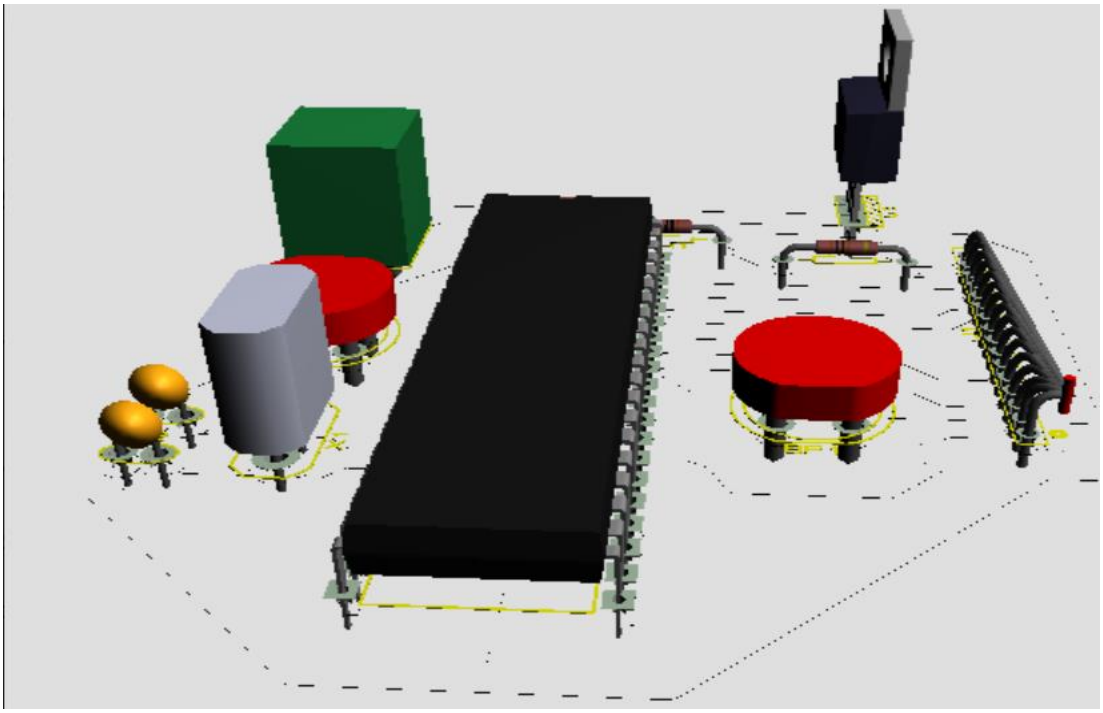


Figure III.17 : Le circuit du calendrier numérique réalisé sous ARES en visualisation 3D.

III 4. Réalisation pratique

Dans cette partie finale, nous allons réaliser les circuits des différents blocs présentés qu'on a testés sur une plaque d'essai.

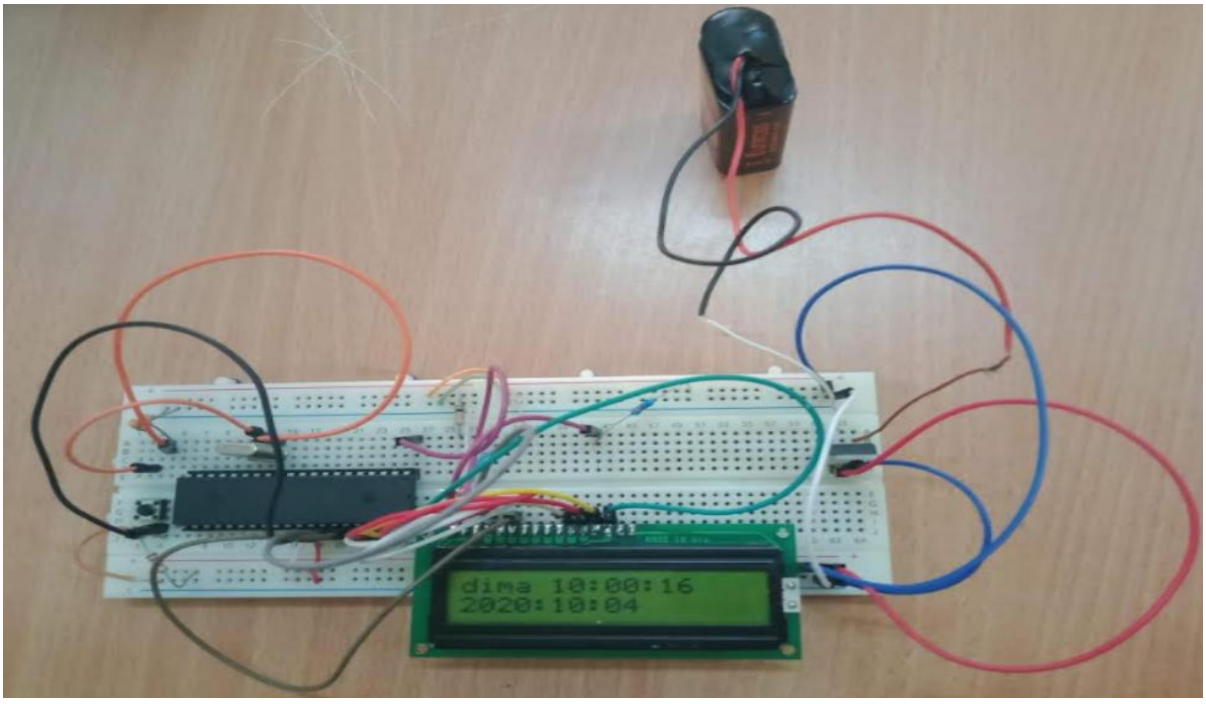
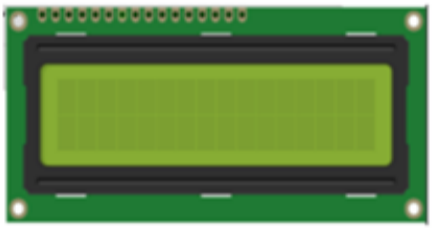


Figure III.18: Montage réalisé sur la plaque d'essai

III.4.1. La liste des composants

❖ **Les circuits intégrés :**

PIC 16F877 : **Afficheur LCD 2*16:** **Régulateur de tension 7805 :**




❖ **Résistances /condensateurs :**

- **2 condensateurs 22PF :** - **Une résistance 1KΩ :** - **Une résistance 10K Ω :**


❖ **Alimentation / Oscillateurs :**

- **Quartz F = 8 MHZ :** - **Une pile (9v) :**

- **2 Boutons poussoirs :**

III.4.2. La fabrication du circuit imprimé

La fabrication du circuit imprimé se passe par plusieurs étapes, Il suffit d'imprimer le typon sur un transparent que montre la figure III.16 pour pouvoir réaliser l'impression directement sur la carte grâce à une machine spécialement conçue pour imprimer les pistes sur un côté, c'est l'imprimante laser.

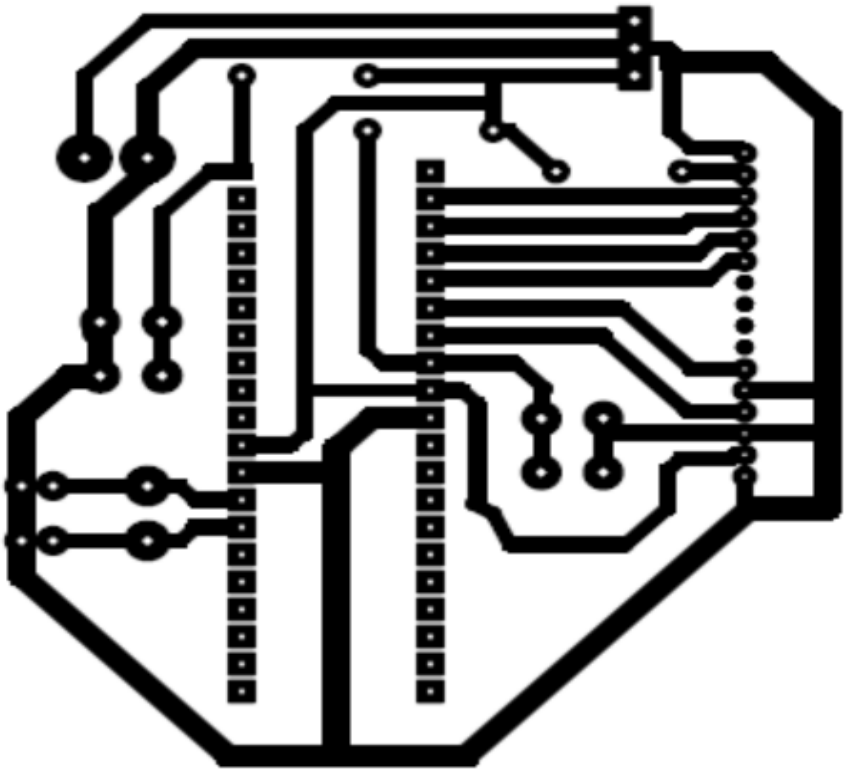


Figure III.19 : Le typon du calendrier numérique.

Pour transférer le tracé du typon sur la plaque du circuit imprimé, on utilise une plaque du cuivre photosensible appelé la carte époxy. Le but après la réalisation de ces étapes, on obtient le circuit imprimé représenté dans la figure III.20

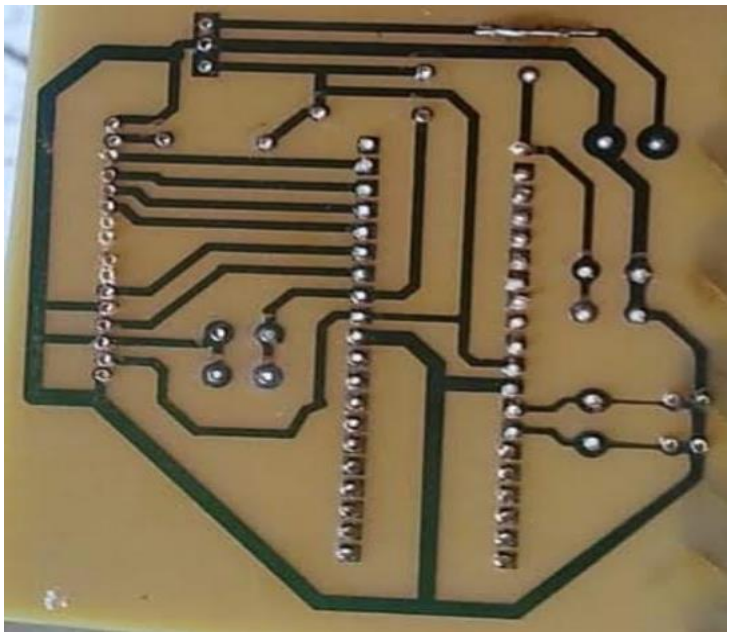


Figure III.20: Image réelle du circuit imprime.

III.4.3. Gravure du programme sur le pic

Les PIC sont des composants qui ne sont capables de rien tant qu'on ne leur a pas fait ingérer un programme. Alors après le teste de ce programme par simulation on l'a transféré vers le microcontrôleur pour procéduraux pratiques, cela à été fait par le chargement du programme dans le pic 16 F877 via un programmateur PICkit2. Cet appareil permet en particulier de transférer le fichier compilé (.hex) vers le PIC, après avoir été détecté par le logiciel puis nous procédant à une vérification (verify) qui indique la succession du transfert.

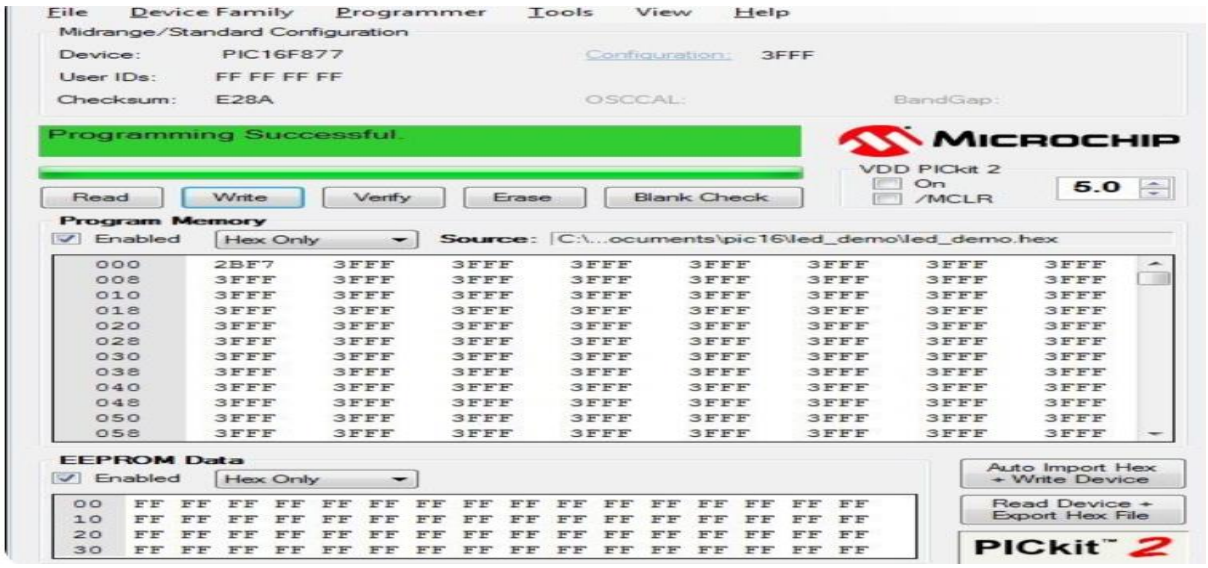


Figure III.21 : Vu du logiciel PICkit2.

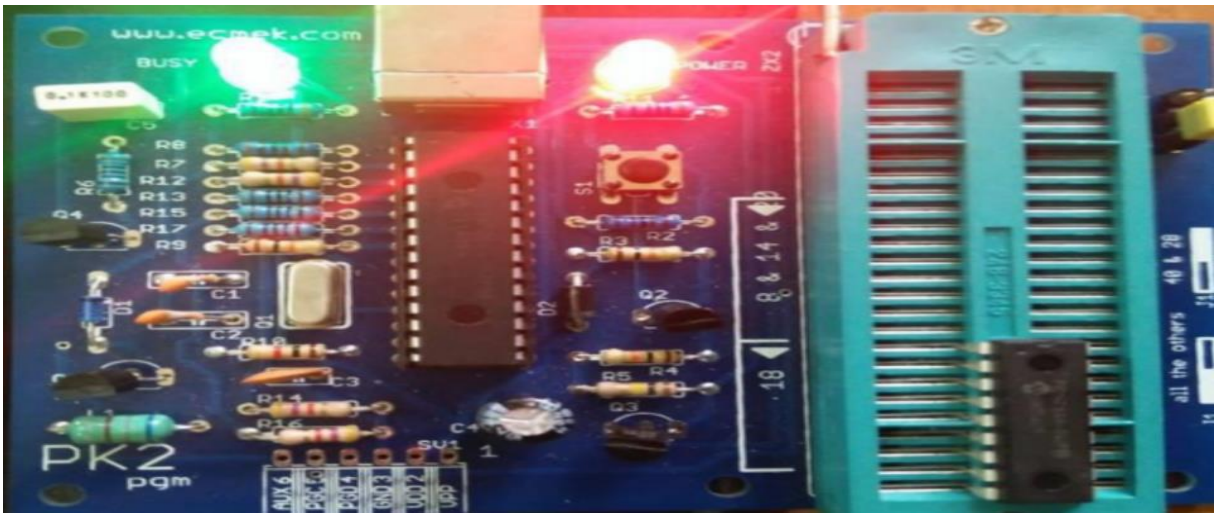


Figure III.22 : Gravure du programme sur le PIC 16F877.

III.5. Conclusion

Les chapitres précédents constituent une base de connaissances théoriques du projet réalisé. Nous avons présentés au niveau de ce chapitre final les différentes étapes de conception et réalisation des différentes parties de notre système. Nous avons aussi présenté l'organigramme pour bien comprendre le principe de fonctionnement de ce dernier. On peut conclure que notre objectif est accompli et les résultats obtenus sont très satisfaisantes sur la plaque d'essai et nous n'avions pas assez de possibilité pour la développer en circuit imprimé.

Conclusion générale

Conclusion générale

La réalisation d'un projet en fin d'études est d'une importance capitale car cela nous a permet de mettre en œuvre nos connaissances théoriques acquises afin d'améliorer nôtres savoir-faire.

L'étude du thème conception et réalisation d'un calendrier numérique réglable déroulée en trois parties (présentation et description des outils du projet, mise en œuvre du microcontrôleur PIC16F877, et la conception et réalisation pratique) nous a montré que tout étude de système technique suit une démarche méthodique constituée de différentes parties aussi importantes les unes que les autres.

Ce projet est à n'en point douter un thème purement électronique qui nous a permis de découvrir la puissance du traitement des microcontrôleurs PIC16F877 et leurs richesses en termes de modules de communication, ou capacité des mémoires RAM / ROM surtout le rapport performances/Prix qu'ils présentent. II nous a permis également d'approfondir nos connaissances en programmation des systèmes à microcontrôleurs pic en langage C ainsi la maîtrise de la gestion des périphériques d'entrées et sorties, il nous a aussi permet la métrise d'utilisation du logiciel PROTEUS, et à affronter les difficultés liées à la conception et la réalisation d'un appareillage électronique.

Au terme de notre travail, ce projet dont la fin est satisfaisante pour nous s'est déroulé avec peu de difficultés rencontrées au terme logiciel et pratique, nos solutions ont été largement inspirées des conseils avisés de notre professeur suiveur qui a su nous guider.

Malheureusement, on n'a pas ou accomplir notre réalisation électronique a terme et l'implantation des composants électronique sur la carte du circuit imprimé qu'a été d'effectué lors de ca création et la non disponibilité d'une autre carte époxy.

Ce travail reste, comme toute œuvre humaine, incomplète et perfectible, nous recommandons d'en améliorer la conception et pour cela nous proposons ci-dessous des améliorations pour les futurs développements :

- Utilisation d'une carte arduino au lieu du microcontrôleur 16F877.
- Ajouter deux condensateur aux bornes du régulateur de tension .
- Ajouter un système de sauvegarde avec un Buzzer.
- Ajouter dans le calendrier un autre affichage du paramètre de la température ambiante via le capteur LM35.
- Synchronisation.

Bibliographie

Bibliographie

[1]**N. NASRI,**cour surles afficheurs LCD :http://blogmatlab.blogspot.com**,** consulté le 13 avril 2020.

[2]www.fractale.gecif.net**,**consulté le 13avril 2020

[3]www.aurel32.net**,**consulté en avril 2020

[4]www.sin.ledantec-numerique.fr**,**consulté en avril 2020

[5]www.memoireonline.comconsulté**,** le 20 avril 2020

[6] **M. TOURE Mohamed Lamine** Ingénieur en Instrumentation et Mesure Physique, cour de proteus professionnel (ISIS & ARES) : http://www.magoe.net**,**consulté le 10 mai 2020.

[7] **V. TOURTCHINE**. Programmation en mikroC. Applications pour les microcontrôleurs de la famille PIC. Manuscrit élaboré selon le programme officiellement agrée et confirmé par le Conseil Scientifique de la Faculté des Sciences. BOUMERDES –2012 :http://www.univ-bouira.dz**,**consulté en mai 2020 .

[8] **Y. Rkhissi Kammoun ,**cours microcontrôleurs **,**www.technologuepro.com consulte le 15 mai 2020.

[9] **R.MALLARD,** les microcontrôleurs PIC pour les débutants qui veulent programmer sans patauger avec mikroPascal : intermédiaire idéal entre BASIC (trop facile) et C (trop difficile), Edition Elektor, 448 pages, 2013.

[10] **N. TOUJENI,** cour microcontrôleurs :www.academia.edu**,**consulté en juin 2020.

[11] **R. RAZAFIARISERA,** RESISTIVIMETRE A BASE DE PIC 16F877, mémoire pour l'obtention du diplôme d'études approfondies en physique,université D'ANTANANARIVO, Madagascar ,67 pages , Juin 2007.

[12] **S.VAROUMAS,** Modèles de programmation de haut niveau pour microcontrôleurs à faibles ressources. Informatique et langage [cs.CL], Thèse de doctorat, Sorbonne Université, France ,253 pages,2009.

[13] **A. OUMNAD**, MICROCONTROLEURS Famille Mid-Range de Microchip (LE PIC 16F876/877).docplayer.fr consulté en 25 juin 2020.

[14] www.microchip.com**,**Consulté en juin 2020.

[15] **I. DOUCHI et L.BELKACMI,** Réalisation d'un prototype permettant le contrôle des deux modes Starting\Generation a bord de l' ATR 72-500,Wilaya de Blida ,projet fin d'études

en vu d'obtention du diplôme master, institut d'aéronautique et des études spaciales ,Blida ,Algerie,76 pages,2017.

[16] **I. DOGAN,** Microcontroller-based Temperature Monitoring and Control, 263 pages , 2002.

[17] **A. HMIDENE**, Agrégé en Génie Support de cours microcontrôleur PIC16F877 Electrique Technologue à l'ISET de Sousse **:**www.espacetechnologue.com**,** consulté en juillet 2020 .

[18] **I. DOGAN,** Advanced PIC Microcontroller Projects in C: from USB to ZIGBEE with the PIC 18F series,2008.

[19] **M. LAGHROUCHE,** Elaboration de la maquette de développement EASYPIC6, Wilaya de TIZI –OUZOU ,Mémoire de fin d'études présenté en vue de l'obtention du Diplôme d'Ingénieur d'état en électronique, université Mouloud Mammeri de Tizi-Ouzou, Algérie,67 pages,2011.

[20] **E. AGOURIANE,** coursmicrocontrôleurs PIC**,** Université Sultan Moulay Slimane, LST Ingénierie Electronique et Télécommunication, Département de Physiquehttp://f2school.com consulté le 10 juillet 2020.

[21] **M. Allain et J. Marot,** Cours Master SIS, Micro-contrôleursMicrochip, universite Paul Cezanne :http://www.lsis.org**,**consulté le 23 jullet 2020.

[22] Traitement programme de l'information, Génie Electronique 2ième année de formation partie 6, le microcontrôleur PIC 16F877 :www.electronique-mixte.fr**,**consulté le le 02 aout 2020.

[23] **Y. PAGNOTTE,** Systèmes d'exploitation et programmation système, Chapitre 9 :Fr.screabd .com **,**consulté le 5 aout 2020.

[24] **B. H, support de cours** LES INTERRUPTIONS Chapitre 3 ,université Constantine 2 ,2020 :elearning .univ-constantine2 .dz**,** consulté le 5 aout 2020.

[25] **Ph. LETENNEUR** - STS GRANVILLE –2004 :www.academia.edu**,**consulté le 10 aout 2020.

[26] **G.BERTHOME**, Lycée Mireille GRENET – COMPIEGNE L'afficheur LCD (Light Control Display) :http://gilles.berthome.free.fr/**,** consulté le 20 AOUT 2020

[27] www.researchgate.net/profile/Islam_Aoufi/publication/269162058_pic_16f**,**Consult le 30 aout 2020.

[28] **V. Chollet** , MICROCONTROLEURS PIC PROGRAMMATION EN C , 09/12/2012.

[29] **S. Zaamta**: « Réalisation d'un régulateur solaire à base de microcontrôleur pour le

contrôle de l'état de charge et la protection des accumulateurs » mémoire de magistère Université Larbi Ben M'hidi Oum ElBouaghi.2008

[30] **Ph. Hoppenot ,** (juin 2004) Département GEII : http://lsc.univ-evry.fr/**,**consulté le 03 septembre 2020.

[31] www.wikipedia.com consulte le 01 octobre 2020.

Annexes

MICROCHIP

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

Devices Included in this Data Sheet:

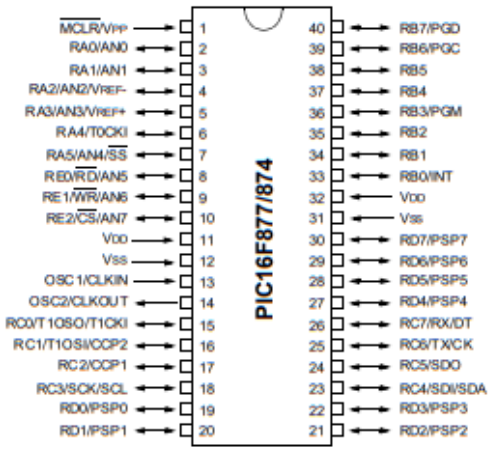
- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

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 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

Pin Diagram

PDIP



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 I^2C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

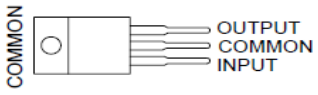
Annexe 2 : Régulateur 7805 Data Sheet

µA7800 SERIES
POSITIVE-VOLTAGE REGULATORS

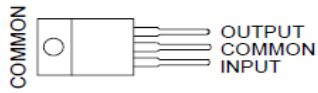
SLVS056J – MAY 1976 – REVISED MAY 2003

- 3-Terminal Regulators
- Output Current up to 1.5 A
- Internal Thermal-Overload Protection
- High Power-Dissipation Capability
- Internal Short-Circuit Current Limiting
- Output Transistor Safe-Area Compensation

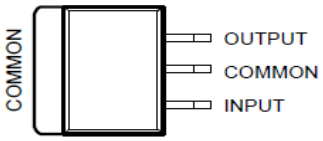
KC (TO-220) PACKAGE
(TOP VIEW)



KCS (TO-220) PACKAGE
(TOP VIEW)



KTE PACKAGE
(TOP VIEW)



description/ordering information

This series of fixed-voltage integrated-circuit voltage regulators is designed for a wide range of applications. These applications include on-card regulation for elimination of noise and distribution problems associated with single-point regulation. Each of these regulators can deliver up to 1.5 A of output current. The internal current-limiting and thermal-shutdown features of these regulators essentially make them immune to overload. In addition to use as fixed-voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents, and also can be used as the power-pass element in precision regulators.

ORDERING INFORMATION

T_J	$V_{O(NOM)}$ (V)	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 125°C	5	POWER-FLEX (KTE)	Reel of 2000	µA7805CKTER	µA7805C
		TO-220 (KC)	Tube of 50	µA7805CKC	µA7805C
		TO-220, short shoulder (KCS)	Tube of 20	µA7805CKCS	
	8	POWER-FLEX (KTE)	Reel of 2000	µA7808CKTER	µA7808C
		TO-220 (KC)	Tube of 50	µA7808CKC	µA7808C
		TO-220, short shoulder (KCS)	Tube of 20	µA7808CKCS	
	10	POWER-FLEX (KTE)	Reel of 2000	µA7810CKTER	µA7810C
		TO-220 (KC)	Tube of 50	µA7810CKC	µA7810C
	12	POWER-FLEX (KTE)	Reel of 2000	µA7812CKTER	µA7812C
		TO-220 (KC)	Tube of 50	µA7812CKC	µA7812C
		TO-220, short shoulder (KCS)	Tube of 20	µA7812CKCS	
	15	POWER-FLEX (KTE)	Reel of 2000	µA7815CKTER	µA7815C
		TO-220 (KC)	Tube of 50	µA7815CKC	µA7815C
		TO-220, short shoulder (KCS)	Tube of 20	µA7815CKCS	
	24	POWER-FLEX (KTE)	Reel of 2000	µA7824CKTER	µA7824C
		TO-220 (KC)	Tube of 50	µA7824CKC	µA7824C

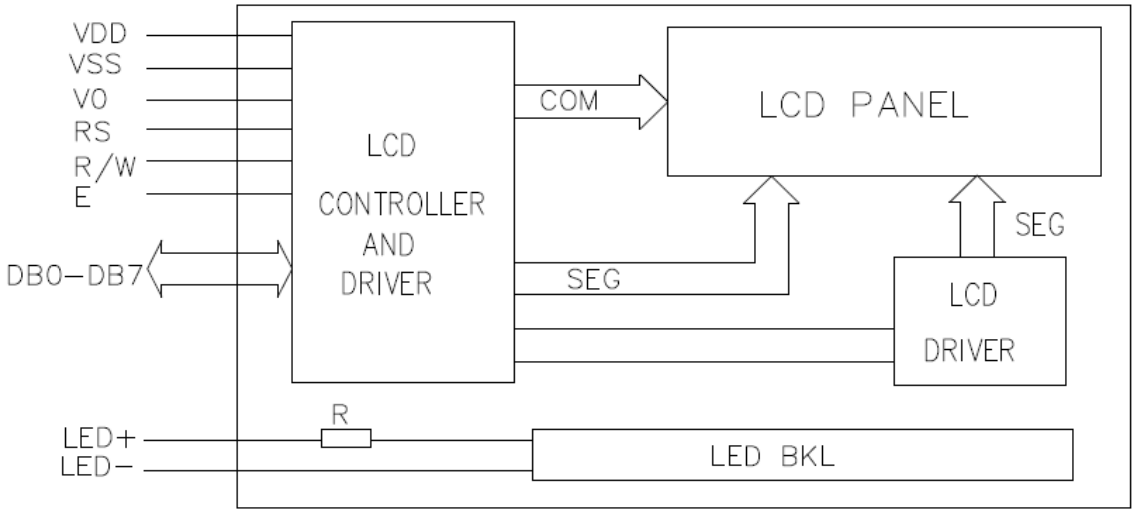
† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

4. Absolute maximum ratings

Item	Symbol	Standard			Unit
Power voltage	V_{DD}-V_{SS}	0	-	7.0	V
Input voltage	V_{IN}	VSS	-	VDD	
Operating temperature range	V_{OP}	0	-	+50	℃
Storage temperature range	V_{ST}	-10	-	+60	

5. Block diagram



6. Interface pin description

Pin no.	Symbol	External connection	Function
1	Vss	Power supply	Signal ground for LCM
2	V_{DD}		Power supply for logic for LCM
3	V_0		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL
16	LED-		Power supply for BKL

Annexe4 : Code source du microcontrôleur pic 16F877

```
//Module de connexion de l'Afficheur LCD//
sbit LCD_RS at  RB1_bit;
sbit LCD_EN at  RB2_bit;
sbit LCD_D4 at  RB3_bit;
sbit LCD_D5 at  RB4_bit;
sbit LCD_D6 at  RB5_bit;
sbit LCD_D7 at  RB6_bit;

sbit LCD_RS_Direction at TRISB1_bit;
sbit LCD_EN_Direction at TRISB2_bit;
sbit LCD_D4_Direction at TRISB3_bit;
sbit LCD_D5_Direction at TRISB4_bit;
sbit LCD_D6_Direction at TRISB5_bit;
sbit LCD_D7_Direction at TRISB6_bit;
// Fin module de connexion de l'Afficheur LCD//
// Déclarations des variables
 char A1,A2,A3,A4,D1,D2,J1,J2,j,t,s2,s1,m2,m1,h2,h1,A6,i;
 bit g;
// Affichage initial de l'heur et jour et date  //
 void affichage()
{  Lcd_chr(1,6,h1); Lcd_chr(1,7,h2);Lcd_chr(1,8,A6);Lcd_chr(1,9,m1);
Lcd_chr(1,10,m2);Lcd_chr(1,11,A6);Lcd_chr(1,12,s1);Lcd_chr(1,13,s2);
Lcd_chr(2,1,A1); Lcd_chr(2,2,A2);Lcd_chr(2,3,A3);Lcd_chr(2,4,A4);
Lcd_chr(2,5,A6);Lcd_chr(2,6,D1); Lcd_chr(2,7,D2);Lcd_chr(2,8,A6);
Lcd_chr(2,9,J1);Lcd_chr(2,10,J2);
if(j==0){ Lcd_out(1,1,"sam");}
}
 void affichagej()
{
        if(j==0) Lcd_out(1,1,"samd");
```

```
            if(j==1) Lcd_out(1,1,"dima");
            if(j==2) Lcd_out(1,1,"lund");
            if(j==3) Lcd_out(1,1,"mard");
            if(j==4) Lcd_out(1,1,"merc");
            if(j==5) Lcd_out(1,1,"jeud");
            if(j==6) Lcd_out(1,1,"vend");

            if(j==7) j=0;
}
// Sous programme du Timer0//
void interrupt()
 {
if(INTCON.T0IF)
      { t++;

       if(t==76){s2++;t=0;
       if(s2==57){s1++;s2=48;}
       if(s1==54){m2++;s1=s2=48;}
       if(m2==57){m1++;s2=s1=m2=48;}
       if(m1==54){h2++;s1=s2=m2=m1=48;}
       if(h2==57){h1++;h2=48;}
      if(h1==50&& h2==51){if(m1==53&& m2==57){if(s1==53&&
s2==57){s2=s1=m2=m1=h2=h1=48;j++;J2++;}}}
         if(J2==57){J1++;J2=48;}
         if(J1==51){D2++;J1=J2=48;}
         if(D2==57){D1++;J2=J1=D2=48;}
         if((D1==49)&&(D2==50)){A4++;D2=J2=49;D1=J1=48;}
         if(A4==57){A3++;D2=J2=49;A4=D1=J1=48;}
         if(A3==57){A2++;D2=J2=49;A3=A4=D1=J1=48;}
         if(A2==57){A1++;D2=J2=49;A2=A3=A4=D1=J1=48;}
          if(A1==57){D2=J2=49;A1=A2=A3=A4=D1=J1=48;}
```

```
        if(s1==53&& s2==57){s2=s1=m2=m1=h2=h1=48;j++;}
          }
       INTCON.T0IF=0;
    }
//Routine d'interruption //

      if(INTCON.INTF)
 {g=~g;}
 INTCON.INTF=0;
     }
// programme principal //

void main()
//déclaration et initialisation des variables //
 {  A1=A3=50;
    A2=A4=D2=J1=48;
    J2=52;
    D1=49;
    s2=s1=m2=m1=h2=48,h1=49,A6=58;
   g=0;
   i= t=0;
    j=1;

TRISB=0B00000001;

  OPTION_REG=0b01000111;
  INTCON=0b10110000;
   TMR0=0;
   Lcd_Init();
   Lcd_cmd(_LCD_CLEAR);
```

```
  Lcd_cmd(_LCD_CURSOR_OFF);
affichage();
   for(;;)
    {
      affichagej();
     affichage();
      while(g==1)
     {
       Lcd_cmd(_LCD_CLEAR);delay_ms(400);affichage();delay_ms(400);
       Lcd_cmd(_LCD_CLEAR);delay_ms(400);affichage();
       while(g==1)
          {
       delay_ms(1000);m2++;
       if(m2==58){m2=48;m1++;}
       if(m1==54){m2=m1=48;}
 affichage();
          }
        delay_ms(2000);
       Lcd_cmd(_LCD_CLEAR);delay_ms(400);affichage();delay_ms(400);
       Lcd_cmd(_LCD_CLEAR);delay_ms(400);affichage();
       while(g==1)
          {
       delay_ms(1000);h2++;
       if(h2==58){h2=48;h1++;}
       if(h1==50&&h2==52){h2=h1=48;}
  affichage();
          }
        delay_ms(2000);
       Lcd_cmd(_LCD_CLEAR);delay_ms(400);affichage();delay_ms(400);
       Lcd_cmd(_LCD_CLEAR);delay_ms(400);affichage();
```

```
 while(g==1)
   {
delay_ms(1000);J2++;
if(J2==58){h2=49;J1++;}
if(J1==50&&J2==48){J2=49;J1=48;}
affichage();
   }
   delay_ms(2000);
Lcd_cmd(_LCD_CLEAR);delay_ms(400);affichage();delay_ms(400);
Lcd_cmd(_LCD_CLEAR);delay_ms(400);affichage();
while(g==1)
   {
delay_ms(1000);D2++;
if(D2==58){D2=49;D1++;}
if(D1==49&&D2==57){D2=49;D1=48;}
affichage();
   }
   delay_ms(2000);
Lcd_cmd(_LCD_CLEAR);delay_ms(400);affichage();delay_ms(400);
Lcd_cmd(_LCD_CLEAR);delay_ms(400);affichage();
while(g==1)
   {
delay_ms(1000);A4++;
if(A4==58){A4=48;A3++;}
 if(A3==57&& A4==57){A4=A3=48;A2++;}
 if(A2==57&& A3==57&& A4==57){A2=A3=A4=48;A1++;}
  if(A1==58&& A2==57&& A3==58&& A4==58){A2=A3=A4=A1=48;}
affichage();
   }
   delay_ms(2000);
```

```
        Lcd_cmd(_LCD_CLEAR);delay_ms(400);affichage();delay_ms(400);
        Lcd_cmd(_LCD_CLEAR);delay_ms(400);affichage();
    while(g==1)
         {     j++;
           affichagej();
affichage();
           delay_ms(1000);
         }
         g=0;
      }
     }
  }
```

Résumé :

Pendant de nombreux siècles, l'homme a compris toute l'utilité de la mesure du temps qui rythmait sa vie quotidienne, c'est ainsi que la maîtrise de cette mesure s'est accrue en mobilisant de nombreux chercheurs, utilisant des techniques de plus en plus complexes, afin de devenir quasiment parfaites de nos jours.

L'objectif principal de ce travail est de concevoir et réaliser un appareil électronique, permettant d'afficher la date et l'heur, il s'agit d'un calendrier à affichage sur un module LCD alphanumérique, basé sur le microcontrôleur PIC 16F877 programmé en langage C à l'aide d'un compilateur MicroC.

Mots clés : Afficheur LCD, PIC 16F877, Langage c, compilateur microC, logiciel Isis Proteus, programme, Organigramme, Interruption, Registre.

Abstract :

For many centuries, man has understood the usefulness of the measurement of time which punctuated his daily life, thus the mastery of this measurement has increased by mobilizing many researchers, using techniques more and more complex, in order to become almost perfect nowadays.

The main objective of this work is to design and produce an electronic device, allowing to display the date, it's about a calendar display on an alphanumeric LCD module, based on the PIC 16F877 microcontroller programmed in C language at using a MicroC compiler.

Keywords : LCD display, PIC 16F877, c language, microC compiler, Isis Proteus software, program, flowchart, interrupt, register.