

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ IBN-KHALDOUN DE TIARET

FACULTÉ DES SCIENCES APPLIQUEES
DÉPARTEMENT DE GENIE ELECTRIQUE



Mémoire de fin d'études

Pour l'obtention du diplôme de Master

Domaine : Sciences et Technologie

Filière : Génie Electrique

Spécialité : Informatique Industrielle

THÈME

**Commande d'un éclairage public et mesure de température à base de pic
16f877A**

Préparé par :

AMRANI Ayoub

HARBOUCHE Nour El Houda

Devant le Jury :

Nom et prénoms	Grade	Qualité
Mr BELARBIM	MCB	Président
Mr NASRI.DJ	MCA	Examineur
Mr GHELLAB.AEK	MCB	Examineur
Mr SAHLI.B	MCA	Encadreur

PROMOTION 2016 /2017

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Au nom de dieu clément et miséricordieux

Dédicace

Je dédie ce projet

A ma chère mère,

A mon cher père,

*Qui n'ont jamais cessé, de me soutenir
et de m'épauler pour que je puisse atteindre mes objectifs.*

A mes sœurs Marwa, Meriem et Ritedj

Pour son soutien moral

A ma Binôme HOUDA

*Pour son entente et sa sympathie, et sur tout l'effort qu'elle a
Donnée durant la préparation de notre projet.*

A mes chers ami(e)s sans exception

A Toute ma Famille.

A Tous ceux que j'aime et ceux qui m'aime

AMRANI AYDUB

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Au nom de dieu clément et miséricordieux

Dédicace

Je dédie ce modeste travail :

A ma chère mère,

A mon cher père,

Qui m'ont poussé toujours vers le mieux

Et qui me guide avec leur soutien

A mes sœurs Romaiassa, Nani et lilia

Ma tante Nadjia

Mon ange Shahde

A mes cousines Fatima, Meriem, Djamila

A Toute ma Famille

Sans oublier mon Binôme Ayoub

A mes amis et mes collègues de la promotion 2016

«Informatique industrielle».

A tous ceux que je connais de loin ou de près.

HARBOUCHE NOUE EL HOUDA

Remerciements

Avant toute chose, nous remercions « ALLAH » le tout puissant pour la bonne santé, la volonté et la patience qu'il nous a donné tout le long de nos études.

Au terme de ce travail, nous tenons à présenter nos remerciements les plus sincères à Monsieur SAHLI.B qu'il est accepté de nous encadrer et qui nous ont fait profiter de leur connaissance et ses précieux conseils au cours de notre projet de fin d'étude.

Toutefois, il faut souligner que ce travail n'aurait pu voir le jour sans l'incalculable connaissance et savoir faire acquis dans notre honorable faculté « la Faculté des sciences appliquées de Tiaret ».

C'est donc avec une immense fierté que nous adressons nos remerciements les plus distingués à tous nos professeurs.

Qu'ils trouvent, ici, ainsi que toute personne qui a contribué à l'achèvement de ce projet, directement ou indirectement, l'expression de notre sincère gratitude.

Enfin nous exprimons nos remerciements les plus dévoués aux membres du jury qui nous ont honorés en évaluant notre travail.

Sommaire

Liste des Figures.....	I
Liste des Tableaux.....	III

Introduction Générale

Introduction Générale.....	1
----------------------------	---

Chapitre I : les différents types de commande

I.1. Introduction.....	3
I.2. Les sources et les types de lampes.....	3
I.2.1. Les lampes à incandescence	3
I.2.2. Les lampes à décharge	3
I.2.3. Les lampes à LED	4
I.3. Les dispositifs de commande.....	4
I.3.1. Allumage manuel	4
I.3.2. Horloge simple	4
I.3.3. Cellule photosensible avec ou sans réglage de sensibilité.....	5
I.3.4. Calculateur astronomique :	6
I.3.5. Commande par courants porteurs.....	6
I.3.6. Commande par voie hertzienne	7
I.4. Les modes de fonctionnement	7
I.4.1. Fonctionnement permanent.....	7
I.4.2. Fonctionnement semi-permanent (coupure nocturne).....	7
I.4.3.10.Fonctionnement par régulation / réduction de puissance (régulation de tension)	8
I.5. Conclusion	9

Chapitre II: Présentation de pic 16f877A

II.1. Introduction.....	11
II.2. Définition d'un microcontrôleur	11

Sommaire

II.3.	Définition d'un PIC.....	12
II.4.	Les différentes familles de PIC.....	12
II.5.	Identification des PIC.....	13
II.6.	Les avantages du microcontrôleur.....	14
II.7.	Structure minimale d'un PIC.....	14
II.8.	Le choix d'un PIC.....	16
II.10.	Le PIC 16F877A.....	16
II.11.	Caractéristiques principales du pic16F877A.....	16
II.12.	Mémoires internes.....	17
II.12.1.	Mémoire morte FLASH.....	17
II.12.2.	Mémoire RAM.....	17
II.12.3.	Mémoire EEPROM.....	17
II.12.4.	Timer.....	17
II.13.	Les Ports d'entrées / sorties.....	18
II.13.1.	Port A.....	18
II.13.2.	Port B.....	18
II.13.3.	Port C.....	18
II.13.4.	Port D.....	19
II.13.5.	Port E.....	19
II.13.6.	Les registres.....	19
II.13.6.1.	Le registre accumulateur.....	19
II.13.6.2.	Le registre d'index.....	19
II.14.	Architecture interne du PIC16F877A :.....	19
II.14.1.	Caractéristiques de la CPU.....	19
II.14.2.	Caractéristiques des périphériques.....	20
II.15.	Architecture externe du Microcontrôleur PIC 16F877A.....	22
II.16.	Quelques PINS du Microcontrôleur 16F877A [8].....	23

Sommaire

II.16.1.	Horloge	23
II.16.2.	Circuit Reset MCLR	25
II.17.	Module de conversion A/N.....	25
II.18.	ADCON0 [8]	28
II.19.	Registre ADCON1	29
II.19.	Conclusion	29

Chapitre III: Programmation de pic 16f877A et la simulation

III.1.	Introduction.....	31
III.2.	Etapes de développement du programme	31
III.3.	Langage et Compilateur MikroC pour PIC.....	31
III.4.	Compilateur mikroC PRO pour PIC	32
III.5.	Création d'un projet.....	33
III.6.	Compilation	36
III.7.	Les commentaires.....	36
III.8.	Sauvegarder un fichier	37
III.9.	Quelques notions de programmation en C sous MikroC	38
III.10.	Déclaration des variables en C.....	38
III.11.	Les opérateurs en langage C	40
III.12.	Présentation du logiciel ISIS de Proteus	40
III.13.	Simulation par ISIS	45
III.14.	Le capteur de lumière	47
III.14.1.	Définition	47
III.15.	Capteur de Température.....	48
III.15.1.	Capteur LM335	48
III.16.	L'afficheur LCD:.....	49
III.17.	Les LED.....	52
III.18.	Unité de contrôle	52

Sommaire

III.18.1.	Le port A	52
III.18.2.	Le port B	53
III.18.3.	Le port C	53
III.19.	Schéma synoptique de système	54
III.20.	Organigramme de programme.....	56
III.21.	Conclusion	57

Conclusion Générale

Conclusion Générale	59
---------------------------	----

Références Bibliographiques

Références bibliographiques.....	61
----------------------------------	----

Liste des Figures

Chapitre I : les différents types de commande

Figure I.1: L'horloge classique	5
Figure I.2: Le Lumandar	6
Figure I.3: Calculateur astronomique	8

Chapitre II: Présentation de pic 16f877A

Figure II.1: Contenu type d'un microcontrôleur	12
Figure II.2: Exemple de l'indentification pic	13
Figure II.3: Repérage de broches	14
Figure II.4: Architecture interne du pic 16f877A	21
Figure II.5: Brochage du microcontrôleur PIC 16F877A	22
Figure II.6: Oscillateur à Quartz/Céramique	24
Figure II.7: Le module de conversion A/N	26

Chapitre III: Programmation de pic 16f877A et la simulation

Figure III.1: Etape dévolution d'un programme	31
Figure III.2: Interface du logiciel MikroC	32
Figure III.3: Création d'un projet en MikroC	33
Figure III.4: Les configurations de projet	34
Figure III.5: Le programme	35
Figure III.6: La compulsation de programme	36
Figure III.7: Affichage des resultats	37
Figure III.8: Sauvegarder un fichier	37
Figure III.9: logiciel ISIS de Proteus	41
Figure III.10: Les caractéristiques de base pour commencer la simulation dans Isis.	42
Figure III.11: Les dispositifs électroniques	43
Figure III.12: Les bornes de la masse et l'alimentation	43
Figure III.13: Fixation des dispositifs	44
Figure III.14: Effectuer les connexions électriques	44
Figure III.15: Lancement de la simulation	45
Figure III.16: Le logiciel ISIS 7 en mode simulation du projet	47
Figure III.17: Le capteur de lumière	47
Figure III.18: Brochage du LM35	48

Liste des Figures

Figure III.19: Le capteur LM35 utilisé.....	49
Figure III.20: Brochage d'afficheur LCD	50
Figure III.21: Interface de LCD avec le pic 16f877A dans la simulation	51
Figure III.22: Résultat de la LED utilisé	52
Figure III.23: L'oscillateur (20MHZ).....	53
Figure III.24: Schéma synoptique de système	54

Liste des Tableaux

Chapitre II: Présentation de pic 16f877A

Tableau II.1: les différents suffixes avec la gamme de fréquence	25
Tableau II.2: Quelques fréquences courantes du quartz et pour les PICs de type Classique	27
Tableau II.3: Le registre ADCON0.....	28
Tableau II.4: Registre ADCON0.....	28
Tableau II.5: Registre ADCON1	29
Tableau II.6: Registre ADCON1	29

Chapitre III: Programmation de pic 16f877A et la simulation

Tableau III.1: Les fichiers de sortie	37
Tableau III.2: Les caractéristiques des variables.	39

Liste des abréviations

GTC : Gestion technique centralisée

PIC : Programmable Interface Controller.

RAM : Random Access Memory ou mémoire à accès direct

CAN : convertisseur analogique / numérique

DIL : Dual in line

RISC : Reduced Instructions Set Computer

UAL : Unité arithmétique et logique

CP : Compteur de programme

PP : Pointeur de pile

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

L'USART : UNIVERSAL Synchronous Receiver Transmitter

SCI: Serial Communication Interface

RTC : Real time clock ou horloge temps réel

VDD : Voltage Drain Drain

VSS : Voltage Source Source

GND : The Ground ou la mass

MCLR : Master Clear

ADC : Analog to Digital Converter

V_{reff} : tension de référence

LDR : light-dependent resistor ou cellule photoconductrice

LCD : Liquide Crystal Display ou afficheur à cristaux liquide

Introduction Générale

Introduction Générale

L'éclairage public fait partie de l'environnement urbain et il est aujourd'hui inconcevable d'imaginer une ville sans éclairage. Il est un service au cœur de la politique de la ville. Il est en effet garant de la sécurité publique, de l'amélioration du cadre de vie des habitants et de la promotion de l'image de la ville par la mise en valeur du patrimoine qui varie dans la forme,

L'intérêt majeur de ce projet est attribué à la recherche d'une solution simplifiant l'automatisation de la commande d'éclairage public avec un circuit programmable à un prix réduit, basé sur un microcontrôleur PIC16F877A, pour essayer d'éviter tout gaspillage d'électricité.

Pour cela le projet a pour but d'allumer les poteaux électriques extérieurs sans l'intervention de l'homme avec l'utilisation de capteur de lumière et d'un capteur de température commandé par un switch. En outre, il ya des fonctions supplémentaires comme mesure de température et affichage de certaines informations comme l'état de jour (Matin ou soir).

La démarche à suivre tout au long du mémoire est présentée ci-dessous :

- Le Premier chapitre est axé sur les types de système utilisé dans la commande de l'éclairage public et les différents types de Lampes utilisées.
- Le deuxième chapitre est axé sur le microcontrôleur PIC16F877A, que nous avons étudié toutes ses caractéristiques et ses description détaillées.
- Le troisième chapitre présente les logiciels utilisés pour la programmation de notre microcontrôleur PIC16F877A, la simulation et le mode de fonctionnement de notre Projet.

Chapitre I

Les différents types de commande

I.1. Introduction

L'éclairage est une radiation émise par une source lumineuse qui n'est pas perçue par l'œil humain. Il est représenté la grande part de consommation d'énergie et du budget des communes et de leur consommation d'électricité. Il y a beaucoup de façons de réduire la consommation d'énergie, tels que la qualité des lampes qui sont utilisées et les types utilisés de la commande d'éclairage ... etc.

Dans ce chapitre, nous allons représenter certains types d'appareils utilisés dans la commande de l'éclairage public.

I.2. Les sources et les types de lampes

Les sources sont les organes où s'effectue la transformation de l'énergie électrique en rayonnement lumineux de façon plus ou moins performante.

Certaines sources peuvent être alimentées directement depuis le réseau, alors que d'autres nécessitent des dispositifs spécifiques pour fonctionner : amorceur, ballast de stabilisation...

Depuis l'invention de l'éclairage à l'électricité, deux grandes catégories de lampes coexistent :

- Les lampes à incandescence
- Les lampes à décharge

Un nouveau type de lampes pourrait bien révolutionner le monde de l'éclairage :

- Les lampes à LEDS [1]

I.2.1. Les lampes à incandescence

Sont les lampes à filament. Elles sont très utilisées dans le cadre de l'éclairage domestique. L'incandescence produit une excellente qualité de la lumière, cela est son principal avantage.

En revanche, l'incandescence génère des coûts élevés en matière de consommation d'énergie, car seule une partie de l'énergie est transformée en lumière. Les lampes utilisant cette technologie ont une durée de vie très courte, puisqu'elles comportent un filament fragile. Elles ont été complètement remplacées par les lampes à décharges. [1]

I.2.2. Les lampes à décharge

Fonctionnent par la création d'une tension électrique entre deux électrodes nécessitant la présence d'un ballast.

Cette technique offre une efficacité lumineuse plus élevée que l'incandescence (plus de lumière est émise pour une consommation d'énergie moindre). Par contre, la qualité de lumière des lampes à décharge est inférieure à celle des lampes à filament. Cela dit d'importants progrès ont été réalisés en la matière, ce qui a permis d'améliorer l'efficacité lumineuse, mais aussi la qualité de la lumière et leur durée de vie. [1]

De par ces avantages, cette catégorie de lampe est utilisée dans la quasi-totalité des cas pour les applications d'éclairage public.

I.2.3. Les lampes à LED

Utilisent le principe de l'électroluminescence et sont alimentées en continu basse-tension.

Cette technologie est en perpétuel progrès et leur rendement double tous les deux ans, tandis que leur coût de production chute. Les LED n'émettent pas d'ultraviolets, l'émission de chaleur est faible et elle dispose d'une durée de vie est très longue. Par contre, l'efficacité lumineuse des LED varie inversement à leur durée de vie et leur rendement reste malgré tout inférieur aux lampes à décharges.

Le développement des LED et les progrès constants constatés tendent à démontrer qu'elles ont un avenir prometteur en matière d'éclairage.[1]

I.3. Les dispositifs de commande

La commande d'éclairage public est l'organe qui contrôle la mise en service et la mise hors service des appareils d'éclairage qui lui sont raccordés par les réseaux d'alimentation.

L'éclairage public, dans la très grande majorité des cas, doit fonctionner quand la lumière naturelle n'est pas suffisante pour assurer une vision suffisante nécessaire à la sécurité et au confort de tous les usagers des espaces publics (conducteurs, cyclistes, piétons...) Il est donc indispensable que l'éclairage soit en fonction dès la tombée de la nuit lorsque la lumière naturelle devient insuffisante, jusqu'au lever du jour, quand elle est à nouveau suffisante.

La maîtrise de l'énergie au niveau du système de commande peut se faire en agissant sur la maîtrise des temps de fonctionnement et sur la maîtrise de la puissance absorbée par les installations. [2]

Les différents types de commande généralement rencontrés sont les suivants :

I.3.1. Allumage manuel

C'est le plus rudimentaire des modes de commande. L'éclairage est déclenché par un simple interrupteur manœuvré à la demande. Obsolète - à proscrire principalement parce que très ancien et de ce fait réalisé en dehors des actuelles règles de l'art. En particulier, il se rencontre parfois sans organe de protection (ni fusibles, ni disjoncteur). [2]

I.3.2. Horloge simple

Une horloge «classique» est installée et commande l'allumage et l'extinction. Ceci se rencontre parfois lors de diagnostics dans les armoires de commande de l'EP de certains anciens lotissements privés ou résidences où les lotisseurs ont largement minimisé les investissements, mais où l'énergie est prise en charge par la collectivité, ce qui est cela dit, parfaitement illégal si la voirie n'a pas été intégrée dans le domaine public (délit dit « de concussion »).

Inconvénient majeur : les heures de bascule de l'horloge doivent être régulièrement modifiées. Cette contrainte fait que le responsable, pour espacer ses interventions dans l'armoire de commande, a tendance à anticiper largement l'allumage quand les jours raccourcissent et omet.

Parfois de retarder celui-ci quand les jours allongent. L'éclairage fonctionne alors parfois longuement en plein jour, ce qui se traduit par des temps annuels d'allumage supérieurs à 5 000 heures. A proscrire [2]



Figure I.1: L'horloge classique

I.3.3. Cellule photosensible avec ou sans réglage de sensibilité

C'est le système communément appelé « Lumandar ». L'appellation générique est « Commande par cellule photoélectrique » ou « interrupteur crépusculaire ». Le principe réside par la commande de la fermeture du contacteur de commande, pilotée par l'état de sortie d'une cellule photoélectrique. Il existe différentes technologies, mais toutes sont plus ou moins sensibles à dérive au fil du temps (vieillesse) ou en fonction de la température. De ce fait, on assistera à une variation importante des temps annuels de fonctionnement.

Ce dispositif nécessitera donc un réglage périodique faute de quoi les heures d'allumage et d'extinction ne sont pas optimales. [2]



Figure I.2: Le Lumandar

I.3.4. Calculateur astronomique :

Également appelé horloge astronomique. C'est un système datant des années 90. Il est composé d'un boîtier modulaire placé dans le coffret de commande de l'éclairage avec une antenne intérieure est parfois disponible pour une remise à l'heure journalière automatique, afin de pallier la dérive minime du quartz de l'appareil. Il n'y a pas de capteur externe.

Cet appareil calcule l'heure d'allumage et d'extinction en fonction des éléments initialisés à l'installation qui sont principalement la date et l'heure et la longitude et latitude du site (coordonnées géographiques). Un algorithme de calcul permet de connaître chaque jour l'heure exacte de l'aurore et de l'aube et ainsi de commander plus précisément l'allumage et l'extinction. [2]

I.3.5. Commande par courants porteurs

Une des techniques également très employées est celle qui consiste à faire transiter le signal d'allumage / extinction par courants porteurs via le réseau de distribution d'énergie électrique qui alimente les différents postes de distribution publique, sources d'alimentation d'éclairage. Généralement, il s'agit d'un signal à 175Hz, superposé à la tension alternative 50Hz normale du réseau, et acheminé depuis les postes sources sur l'ensemble des grappes HTA concernées aux heures d'allumage et d'extinction calculée par le distributeur. L'investissement est faible, le récepteur est le

plus souvent installé à la charge du distributeur d'énergie. De plus, le temps de fonctionnement est garanti avec fiabilité. [1]

Il existe aussi d'autres systèmes plus ou moins répandus:

I.3.6. Commande par voie hertzienne

Ce principe repose sur l'émission d'ordres d'allumage/extinction depuis un émetteur au niveau de la commande centrale, lui-même opérant généralement grâce à un calculateur astronomique couplé à une cellule photoélectrique. Les ordres émis sont reçus par des récepteurs au niveau de chaque point de livraison d'énergie électrique. [2]

I.4. Les modes de fonctionnement

Le type de commande le plus évolué est l'application à l'éclairage public des technologies de GTC (Gestion technique centralisée). Les dispositifs de GTC se rencontrent à des niveaux plus ou moins complets. Une gestion technique centralisée effectue non seulement l'acheminement dans le sens « aller », des ordres de commande d'allumage du poste central vers chaque départ d'alimentation (voire individuellement jusqu'à chaque point lumineux), et dans le sens « retour » vers le poste central, achemine des informations relatives au fonctionnement des installations, avec au minimum la confirmation que le contacteur du départ commandé a bien été fermé. On distingue la télégestion « à l'armoire » ou « au point lumineux », suivant que l'on peut agir individuellement ou obtenir des informations dont la finesse va jusqu'à chaque départ d'éclairage ou va plus loin jusqu'à chaque point lumineux.

Associés à ces commandes différents modes de fonctionnement sont possibles favorisant plus ou moins la maîtrise de l'énergie:

- Fonctionnement permanent
- Fonctionnement semi-permanent
- Fonctionnement par régulation / réduction de puissance [2]

I.4.1. Fonctionnement permanent

C'est le régime de fonctionnement le plus répandu : allumage le soir, extinction le matin.

Un interrupteur crépusculaire suffit à assurer ce type de fonctionnement. Quand le dispositif d'allumage / extinction est correctement réglé, ce type de fonctionnement engendre un temps annuel d'utilisation de l'ordre de 4100 heures.[2]

I.4.2. Fonctionnement semi-permanent (coupure nocturne)

C'est le plus rudimentaire des moyens de maîtrise de l'énergie, mais c'est aussi le plus efficace Très bien accepté en zone rurale, il l'est cependant moins en zone urbaine.

C'est aussi le moyen de lutte le plus radical contre la pollution lumineuse. Le coût d'une horloge simple étant très faible, le retour sur investissement est très court quel que soit le nombre d'appareils.

L'économie énergétique permise par ce type de dispositif est de plus de 8% par heure d'extinction. Les limites du recours à ce dispositif sont liées à l'inconfort en absence l'éclairage.

L'insécurité (sur la route ou pour les biens et les personnes) en absence d'éclairage est aussi un motif souvent invoqué, mais également source de débat.

La coupure nocturne double le nombre d'allumages des lampes sous pleine puissance. Les lampes s'usent plus vite lors de ces allumages. Il n'est donc pas recommandé d'effectuer une coupure nocturne trop courte. [2]



Figure I.3: Calculateur astronomique

I.4.3. Fonctionnement par régulation / réduction de puissance (régulation de tension)

Il s'agit à proprement parlé d'une réduction de l'éclairement à certaines heures de la nuit.

Cette régulation de tension à l'armoire apporte un nombre important d'avantages à savoir :

- la réduction de la dépense énergétique grâce à la limitation de la puissance aux heures creuses.
- L'annulation des surtensions permettant à la source de travailler à un rendement maximal.
- Le maintien d'un éclairage de nuit contrairement à la coupure nocturne vue précédemment en répondant au souhait de maintien de l'éclairage toute la nuit.

- La préservation de la durée de vie des lampes en éliminant les surtensions et en assurant pour certains modèles des rampes progressives de montée en tension qui permettent de limiter les surintensités appelées sur le réseau lors de l'allumage des installations.[2]

I.5. Conclusion

Dans ce chapitre, nous avons offert un aperçu de l'éclairage public et les différents types de commande.

L'objectif de ce chapitre est de simplifier la compréhension du principe de commande de l'éclairage public pour tirer parti dès les idées à la construction de notre projet.

Chapitre II
Présentation du
PIC16f877A

II.1. Introduction

Les microcontrôleurs sont et continueront à être largement utilisés pour les applications de régulation et de commande de processus. Ce sont de véritables micro-ordinateurs intégrés sur une puce de silicium qui comportent une unité centrale, de la mémoire ou une interface à de la mémoire externe, des ports d'entrée-sortie, une interface pour lignes série (RS-232) ainsi qu'une unité de gestion de temps et d'événements...etc. Les signaux d'entrée-sortie du microcontrôleur peuvent être facilement interfacés à des coupleurs optiques afin d'interfacer des capteurs et des actionneurs industriels.

Pratiquement tous les fabricants de microprocesseurs (Microship, Motorola, Intel, Hitachi, Texas Instrument, Toshiba, ST Microélectronique-ex SGS-Thomson, etc.) proposent une ou plusieurs gammes de microcontrôleurs. Les microcontrôleurs 4 bits servent essentiellement à des tâches simples. De tels microcontrôleurs sont par exemple utilisés au sein d'objets ménagers grand public, tels que des cuisinières, machines à laver ou aspirateurs. Les microcontrôleurs 8 bits sont capables de répondre à des exigences plus élevées et sont utilisés pour la commande de dispositifs informatiques tels que des joysticks, tablettes graphiques et modems. Ils sont également utilisés pour la programmation de petits robots ainsi que pour l'acquisition de données (convertisseurs A/D, etc.). Les microcontrôleurs 16/32 bits sont utilisés pour la commande de machines ou le contrôle de processus, lorsque les contraintes temps réel sont sévères ou lorsque les algorithmes de régulation nécessitent une puissance de calcul importante. Des variantes de microcontrôleurs avec canaux d'accès mémoire direct offrant un grand débit entre mémoire et entrées-sorties sont utilisés dans les applications multimédia et pour le contrôle d'imprimantes laser.

Dans ce chapitre nous présentons une description détaillée sur microcontrôleur (PICs de Microship).

II.2. Définition d'un microcontrôleur

Le microcontrôleur est un dérivé du microprocesseur. Sa structure est celle des systèmes à base de microprocesseurs. Il est donc composé en plus de l'unité centrale de traitement, 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)... [3]

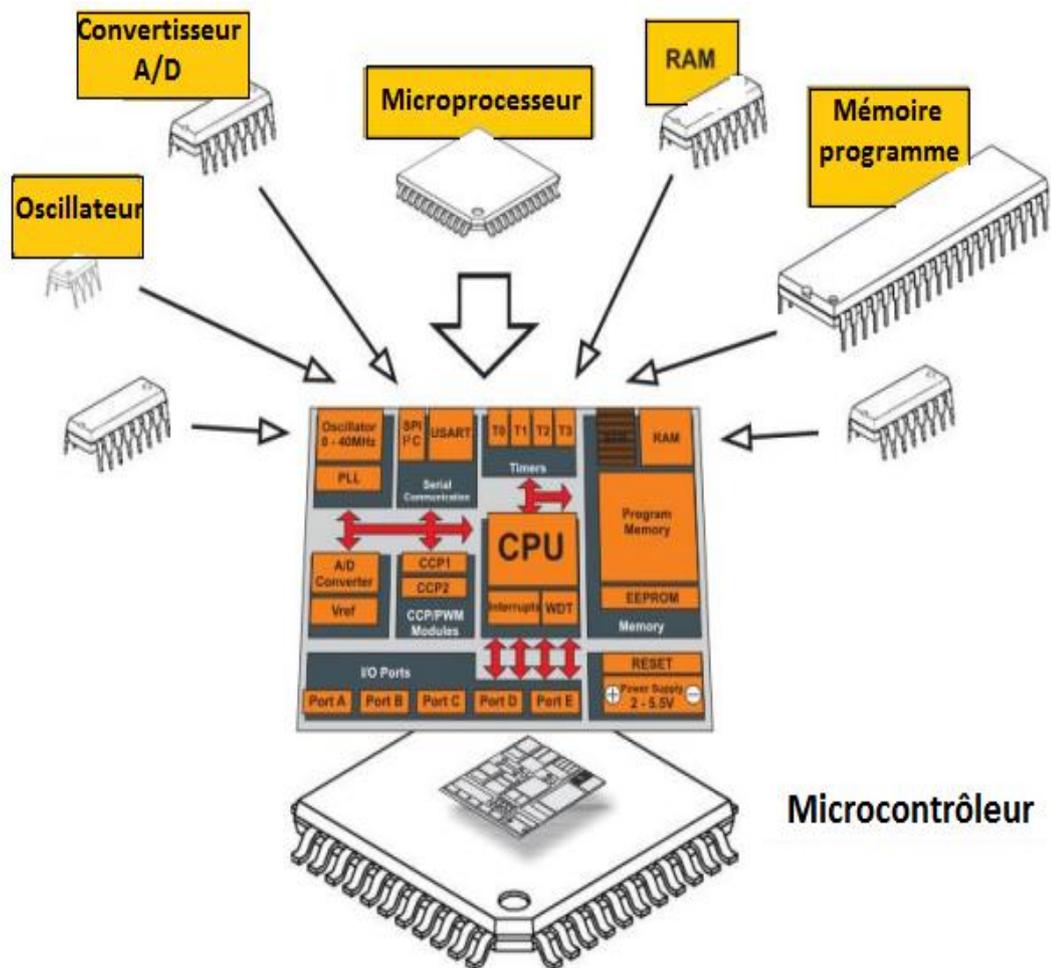


Figure II.4: Contenu type d'un microcontrôleur

II.3. Définition d'un PIC

Un PIC (Programmable Interface Contrôler) est une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de faciliter l'interfaçage avec le monde extérieur sans nécessiter l'ajout de composants externes.

Les PICs sont des composant dits RISC (Reduced Instructions Set Computer), ou encore composant à jeu d'instruction réduit. Le microcontrôleur se trouve dans plusieurs appareils telle que : les téléphones portables, machines à laver, télévisions vidéos ... etc. [3]

II.4. Les différentes familles de PIC

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

Toutes les PICs Mid-Range ont un jeu de 35 instructions, stockent chaque instruction dans un seul mot de programme, et exécutent chaque instruction (sauf les sauts) en un cycle. On atteint donc des très grandes vitesses, et les instructions sont de plus très rapidement assimilées. [4]

II.5. Identification des PIC

Pour identifier une PIC, on utilise simplement son numéro.

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

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

Ensuite, on trouve :

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

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

Une **16F877-20** est une **PIC Mid-Range (16)** d'où la mémoire programme est de type **FLASH (F)** donc réinscriptible de type **877**, et capable d'accepter une fréquence d'horloge de **20MHz**.

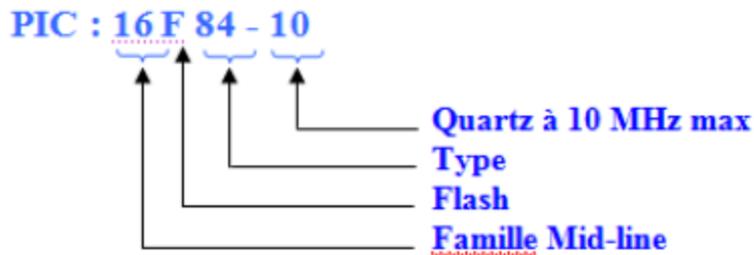


Figure II.5: Exemple de l'indentification pic

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

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

Les microcontrôleurs PIC sont présentés en boîtier DIL (Dual In Line). Un point ou une encoche donne un repérage de la broche 1, ensuite il faut se déplacer vers la droite pour avoir les autres broches. Ont fait le tour du circuit dans le sens trigonométrique. [3]

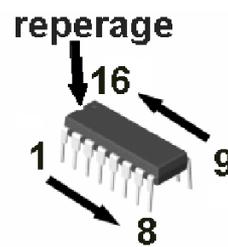


Figure II.6: Repérage de broches

II.6. Les avantages du microcontrôleur

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

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

- Tout d'abord, un microcontrôleur intègre dans un seul et même boîtier ce qui, avant nécessitait une dizaine d'éléments séparés. Il résulte donc une diminution évidente de l'encombrement de matériel et de circuit imprimé.
- Cette intégration a aussi comme conséquence immédiate de simplifier le tracé du circuit imprimé puisqu'il n'est plus nécessaire de véhiculer des bus d'adresses et de données d'un composant à un autre.
- L'augmentation de la fiabilité du système puisque, le nombre des composants diminuant, le nombre des connexions composants/supports ou composants/circuits imprimés diminue.
- Le microcontrôleur contribue à réduire les coûts à plusieurs niveaux.
- Moins cher que les autres composants qu'il remplace.
- Diminuer les coûts de main d'œuvre.
- Réalisation des applications non réalisables avec d'autres composants. [4]

II.7. Structure minimale d'un PIC

La structure minimale d'un PIC est constituée des éléments ci-dessous :

- Une unité arithmétique et logique (UAL) : est chargée d'effectuer toutes les opérations arithmétiques de base (addition, soustraction et logiques etc.).
- Un registre compteur de programme (CP) : est chargée de pointer l'adresse mémoire courante contenant l'instruction à réaliser pour le microcontrôleur.

Le contenu de registre PC évolue selon le pas de programme.

- Registre pointeur de pile (PP) : est essentiellement utilisé lorsque l'on relise un sous programme. Le pointeur de pile est chargé de mémoriser l'adresse courante que contient le compteur de programme avant le saut à l'adresse du sou programme.
- Un registre d'instruction : contient tous les codes binaires correspondants aux instructions à réaliser par le microcontrôleur.
- Une horloge système : permet de cadencer tous les échanges internes et externes au microcontrôleur.
- Un registre d'état : est en relation avec l'UAL et permet de tester le résultat de la dernière opération effectuée par le microcontrôleur. Des bits sont : Positionnés dans le registre d'état et ceux-ci peuvent être testé à l'aide de instruction de branchement pour effectué des sauts conditionnels.
- Des ports d'entrées /sorties : permettent de dialogué avec l'extérieure microcontrôleur, par exemple pour prendre en compte l'état d'un interrupteur (entré logique).
- La mémoire programme : la mémoire programme est constitué de 8K mots de 14 bits. C'est Dans cette zone que nous allons écrire notre programme.
- En effet, il faut 2octets pour codé 14bits .Ceci explique également pour quoi, lorsque nous lisons une PIC vierge, nous allons lire des 0x3FFF. Cela donne en binaire B11111111111111', soit 14 bits. Sachant qu'une instruction est codée sur un mot, donc 8 K mots donnent 8 milliers d'instructions possibles.
- La mémoire EEPROM (Electrical Erasable Programmable Read Only Memory) : est constitue de 256 octets que nous pouvons lire et écrire depuis notre programme. Ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres semi permanents. Leur utilisation implique une procédure spéciale, car ce n'est pas de la RAM, mais bien une ROM de type spécial. Il est donc plus rapide de la lire que d'y écrire.
- La mémoire RAM : c'est celle que nous allons sans cesse utiliser. Toutes les données qui y sont stockées sont perdues lors d'une coupure du courant. La mémoire RAM est subdivisée de plus en deux parties. dans chacune des banques nous allons trouver des (cases mémoires spéciales) appelées REGISTRES SPECIAUX et des cases mémoires (libres). [4]

II.8. Le choix d'un PIC

Le choix d'un PIC est directement lié à m'application envisagé :

Il faut dans un premier temps déterminer le nombre d'entrées/sorties nécessaires pour l'application. Ce nombre d'entrées/sorties nous donne une idée sur la famille du PIC.

Il faut ensuite déterminer si l'application nécessite un convertisseur Analogique/ Numérique ce qui va centrer un peu plus vers le choix du PIC.

La rapidité d'exécution est un élément important, il faut consulter les DATA-BOOK pour vérifier la compatibilité entre la vitesse maximale du PIC choisi et la vitesse max nécessaire au montage.

La taille de la RAM interne et la présence ou non d'une EEPROM pour mémoriser des données est également important pour l'application souhaitée.

La longueur de programme de l'application détermine la taille de la mémoire programme du PIC recherché. [2]

Afin de choisir un PIC adéquat à notre projet, nous avons pensé à l'utilisation du PIC 16F877A

II.10. Le PIC 16F877A

II.11. Caractéristiques principales du pic16F877A

Le PIC 16f877A est caractérisé par :

- Une Fréquence de fonctionnement élevée.
- Une mémoire vive de 368 octets.
- Une mémoire EEPROM pour sauver des paramètres de 256 octets.
- Une mémoire morte de type FLASH de 8K mots (1mot=14bits), elle est réinscriptible à volonté.
- Chien de garde WDT.
- 33 Entrées et sortie.
- 3 Temporisateurs : TIMERS0 (8 bits avec pré diviseur), TIMERS1 (16 bits avec pré diviseur avec possibilité d'utiliser une horloge externe réseau RC ou QUARTZ) et TIMERS2 (8bits avec pré diviseur et post diviseur).
- 2 entrées de capteurs et de comparaison avec PWM (Modulation de largeur d'impulsions).
- Convertisseur analogique numérique 10 bits avec 8 entrées multiplexées maximum.
- Une interface de communication série asynchrone et synchrone (USART /SCI)
- Une interface de communication série synchrone (SSP/SPI et I2C).
- Une tension d'alimentation entre 2 et 5.5 V. [5]

II.12. Mémoires internes

Le PIC 16F877A dispose trois types de mémoire :

II.12.1. Mémoire morte FLASH

C'est la mémoire programme proprement dite, Chaque mémoire unitaire fait 14 bits, La mémoire FLASH est un type de mémoire stable réinscriptible à volonté, c'est ce nouveau type de mémoire qui fait le succès de microcontrôleur PIC.

Dans le cas du 16f877, cette mémoire fait 8k. Lorsque l'on programme en assembleur, on écrit le programme directement dans cette mémoire. [6]

II.12.2. Mémoire RAM

C'est la mémoire d'accès rapide mais labile (c-à-dire qu'elle s'efface lorsqu'elle n'est plus sous tension), cette mémoire contient les registre de configuration de PIC ainsi que les différent registre de données, elle contient également les variables utilisées par le programme.

Cette mémoire RAM disponible sur le PIC 16F877A est de 368 octets, elle est repartie de la mémoire suivante :

- 80 octets en banque 0, adresses 0x20 à 0x6F.
- 80 octets en banque 1, adresses AxA0 à 0Xef.
- 96 octets en banque 2, adresses 0x110 à 16F.
- 96 octets en banque 3, adresses 0x190 à 0x1EF.
- 16 octets communs aux 4 banque, soit 0x70 à 0x7F ; 0xF0 à 0xFF ; 0x170 à 0x17F ; 0x1F0 à 0x1FF. [6]

II.12.3. Mémoire EEPROM

Cette mémoire programme est de 256 octets, elle est électriquement effaçable, réinscriptible et stable, ce type de mémoire est d'accès plus lent, elle est utilisée pour conservés les paramètres après une coupure de courant.

La dresse relative de l'accès EEPROM est comprise entre 0000et 00FF, ce qui nous permet d'utiliser qu'un registre de huit bits pour définir cette adresse [6]

II.12.4. Timer

Le timer est un registre de 8 ou 16 bits, qui est incrémenté avec un rythme fixé avec :

- Les impulsions d'une horloge (fonctionnement timer).
- Changement d'états d'un port (fonctionnement compteur).

Dans le cas de fonctionnement timer , le timer est utilisé comme une base de temps.

Le WDT (watchdog Timer) ou COP (Compteur Operating Properce), est un timer qui vérifie le déroulement de programme, si le programme plante, il génère une reset, pour ce la il faut initialisé le WDT périodiquement.[5]

II.13. Les Ports d'entrées / sorties

La plupart des broches du PIC16F877A sont accessibles en entrée et en sortie tout-ou-rien, c'est-à-dire qu'il peut en entrer ou en sortir un état haut logique (+VCC) ou un état bas logique (0V).

Les broches sont regroupées par ports, chaque port pouvant contenir jusqu'à 8 broches. Il est possible d'utiliser chaque patte indépendamment en entrée, en sortie ou pour d'autres fonctions. [7]

Le PIC 16F877A dispose de 5 ports :

- 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.

II.13.1.Port A

Le port A est formé de six pins donc six entrées /sorties numérotées de RA0 à RA5 qui peuvent être utilisé comme les entrées pour le convertisseur analogique numérique ou utilisé pour le TIMER 0 dans ce dernier cas le pin RA4 sera utilisé comme entrée pour configurer TOCKI est le type drain ouvert

On peut utiliser ce port soit pour la conversion analogique /numérique soit en mode (I/O). [7]

II.13.2.Port B

Rien de particulier à dire sur ce port qui possède 8 pins classique numérotées de RB0 à RB7.On note que le pin RB0 qui en configuration d'entrée est de type *triggre de Schmitt* quand elle est utilisé en mode interruption INI

La lecture simple de RB0 se fait de façon tout à fait classique, en entrée de type TTL. [7]

II.13.3.Port C

Tout d'abord au niveau programmation c'est un PORT tout ce qu'il y a de plus classique comportant 8pins de RC0 à RC7.

On trouve donc un registre TRISC localisé dans la banque 1, qui permet de décider quelles sont les entrées el quelles sone les sorties .le fonctionnement est identique à celui des autres TRIS à savoir que le positionnement d'un bit à 1 place le pin en entrée et que le positionnement de ce bit 0 place la dite pin de sortie.

Au niveau électronique, on remarque que toutes les pins lorsqu'elles sont configurées en entrée sont des entrées de type *trigger de Schmitt* ce qui permet d'éviter incertitudes de niveau sur niveau qui ne sont ni des +5V, donc en général sur les signaux qui varient lentement d'un niveau à l'autre. [7]

II.13.4.Port D

Ce port n'est présent que sur les Pics de type 16f877A, il fonctionne de façon identique aux autres dans son mode de fonctionnement générale, les 8 pins I/O, en mode entrée son de type *trigger de Schmitt* ce port est très utilisé en mode parallèle esclave (slave). [7]

II.13.5.Port E

Il ne comporte que 3 pins, RE0, RE1 et RE2 contrairement aux autres ports

On remarque que les pins REx peuvent être utilisés comme pins d'entrées analogiques. [7]

II.13.6.Les registres

Ils sont des cases mémoires réserves qui gèrent la manière d'exécution des programmes et indiquent les résultats de certaines opérations.

Le nombre et les noms des registres varient d'un microcontrôleur à l'autre, mais certains registres basique sont présents dans tous les microcontrôleur

II.13.6.1. Le registre accumulateur

Il maintient le résultat d'une opération calculé par le CPU.

II.13.6.2. Le registre d'index

Il est utilisé afin de spécifier une adresse dans le cas d'un adressage indirecte. [5]

II.14. Architecture interne du PIC16F87A :

II.14.1.Caractéristiques de la CPU

- CPU à architecture RISC (8 bits)
- Mémoire programme de 8 K mots de 14 bits (Flash),
- Mémoire donnée de 368 Octets,
- EEPROM donnée de 256 Octets,
- 14 sources interruptions
- Générateur d'horloge de type RC ou quartz (jusqu' à 20 MHz)
- 05 ports d'entrée sortie

- Fonctionnement en mode sleep pour réduction de la consommation,
- Programmation par mode ICSP (In Circuit Serial Programming) 12V ou 5V,
- Possibilité aux applications utilisateur d'accéder à la mémoire programme. [3]

II.14.2. Caractéristiques des périphériques

- Timer0: Timer/Compteur 8 bits avec un prédiviseur 8 bits
- Timer1: Timer/Compteur 16 bits avec un prédiviseur de 1, 2, 4, ou 8 ; il peut être incrémenté en mode veille (Sleep), via une horloge externe,
- Timer2: Timer 8 bits avec deux diviseurs (pré et post diviseur)
- Deux modules « Capture, Compare et PWM » :
- Module capture 16 bits avec une résolution max. 12,5 ns,
- Module Compare 16 bits avec une résolution max. 200 ns,
- Module PWM avec une résolution max. 10 bits,
- 0 Convertisseur Analogiques numériques multi-canal (8 voies) avec une conversion sur 10 bits, Synchronous Serial Port (SSP) SSP, Port série synchrone en mode I2C (mode maître/esclave),
- Universel Synchronous Asynchronous Receiver Transmitter (USART) : Port série universel, mode asynchrone (RS232) et mode synchrone. [3]

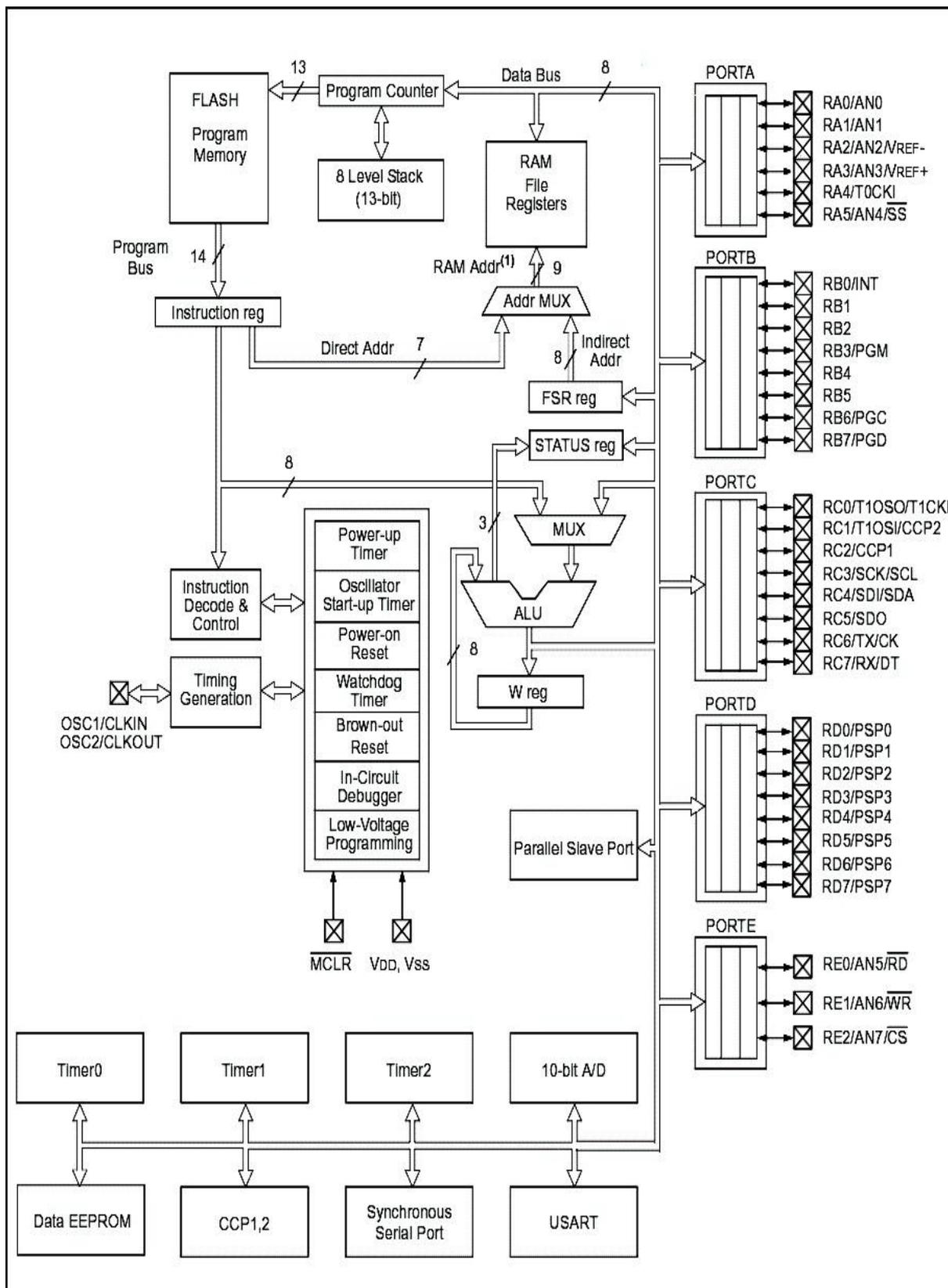


Figure II.7: Architecture interne du pic 16f877A

II.15. Architecture externe du Microcontrôleur PIC 16F877A

Le PIC 16F877A est présenté sous la forme d'un circuit intégré contenu dans un boîtier, il présente 40 broches, 20 de chaque côté. Les broches sont virtuellement numérotées de 1 à 40. La 1ere broche est placée dans le coin situé à gauche de l'encoche de repérage. Il a 4 pins pour l'alimentation (VDD, VSS), 2 pins pour l'oscillateur (OSC1, OSC2), 1 pin pour le RESET (MCLR) et 33 pins d'entrées/sorties, chacune de ses broches a une ou plusieurs fonctions bien précises. [3]

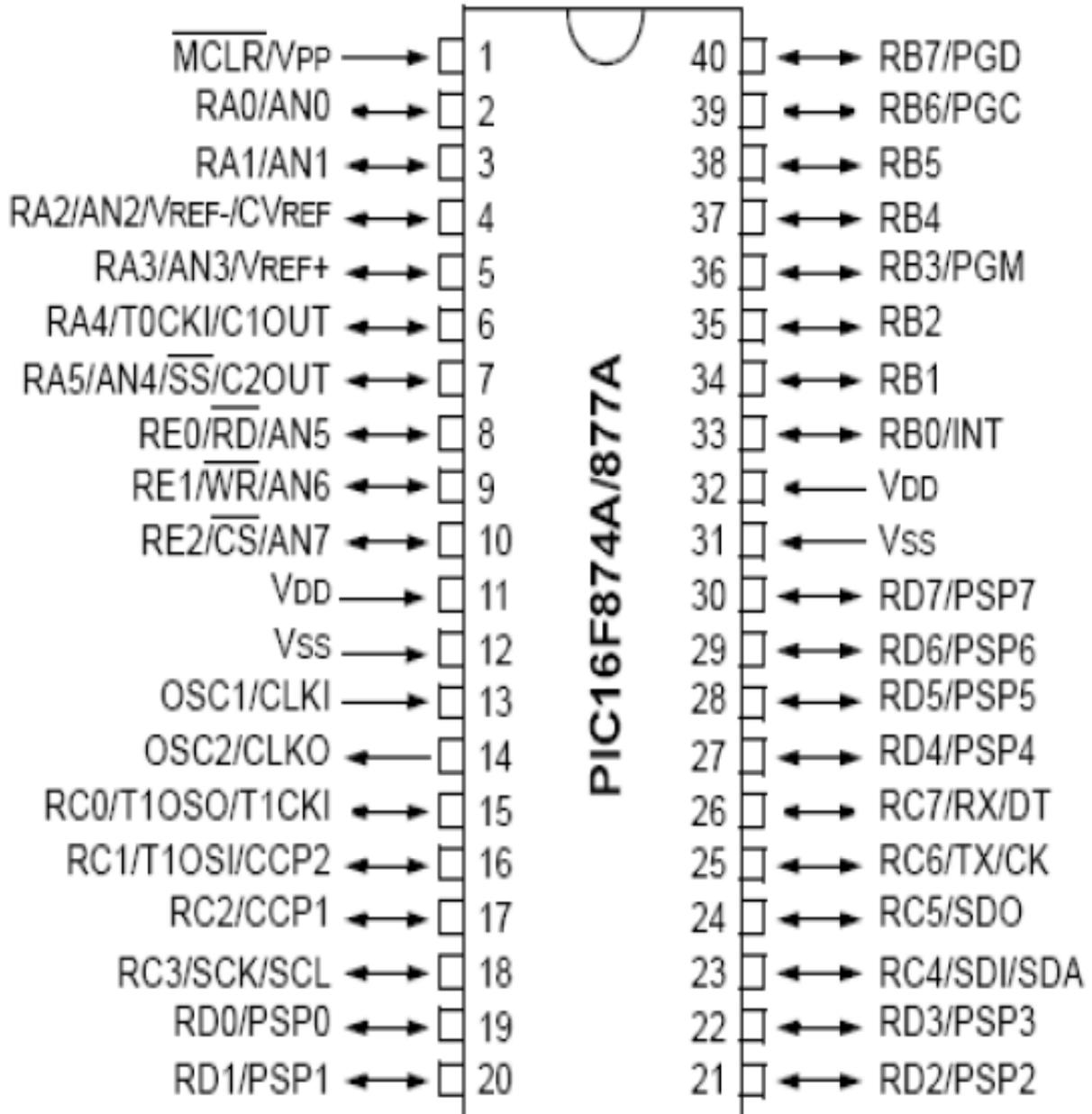


Figure II.8: Brochage du microcontrôleur PIC 16F877A

Le boîtier du PIC16F877A décrit par la figure (II.5) comprend 40 pins : 33 pins d'entrées/ sortie, 4 pins pour l'alimentation, 2 pins pour l'oscillateur et un pin pour le reset (MCLR). La broche MCLR sert à initialiser le microcontrôleur en cas de la mise sous tension, de remise à zéro externe, de chien de garde et en cas de ma baisse de tension d'alimentation. Les broches VDD et VSS servent à alimenter le PIC. 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. Les courants véhiculés dans le pic sont loin d'être négligeables du fait des nombreuses entrées/sorties disponible. Le constructeur a donc décidé de répartir les courants en plaçant 2 pins pour l'alimentation VSS, bien évidemment, pour les mêmes raisons, ces pins sont situées de part et d'autre du PIC, et en positions relativement centrales.

L'oscillateur interne de PIC qui peut être un quartz, un résonateur céramique, un oscillateur externe ou un réseau RC. [5]

II.16. Quelques PINS du Microcontrôleur 16F877A [8]

II.16.1. Horloge

L'horloge est externe. L'horloge. Avec l'oscillateur a Quartz, on peut avoir des fréquences allant jusqu'à 20 MHz selon le type de μ C. Le filtre passe bas (Rs, C1, C2) limite les harmoniques dus à l'écrêtage et Réduit l'amplitude de l'oscillation, il n'est pas obligatoire.

Avec un oscillateur RC, la fréquence de l'oscillation est fixée par Vdd, Rext et Cext.

Elle peut varier légèrement d'un circuit à l'autre.

Dans certains cas, une horloge externe au microcontrôleur peut être utilisée pour synchroniser le PIC sur un processus particulier.

Quelque soit l'oscillateur utilise, l'horloge système dite aussi horloge instruction est obtenue en divisant les fréquences par 4. Dans la suite de ce document on utilisera le terme $F_{osc}/4$ pour designer l'horloge système.

Avec un quartz de 4 MHz, on obtient une horloge instruction de 1 MHz, soit le temps pour exécuter une instruction de $1\mu s$.

Le PIC16F877 peut utiliser 4 types d'horloge comme horloge système :

- XT : Oscillateur à résonateur céramique ou à cristal (quartz) externe, jusqu'à 4 MHz. Il est nécessaire dans ce mode d'ajouter entre les pattes OSC1 et OSC2 du composant un quartz, ainsi que deux condensateurs adaptés entre chacune de ces deux broches et la masse.
- HS : Même fonctionnement que le précédent, mais pour des fréquences supérieures à 4MHz.

Dans modèles d'oscillateurs HS, un oscillateur en quartz ou en céramique est reliée à la OSC1 et OSC2 repères pour établir oscillation. La Figure 2-3 illustre le brochage.

Le choix de l'oscillateur se fait par des registres particuliers, les registres de configuration.

Ces registres ont la particularité de ne pouvoir être écrits que lors de la programmation. Pour les régler on peut soit utiliser la directive par MikroC. [8]

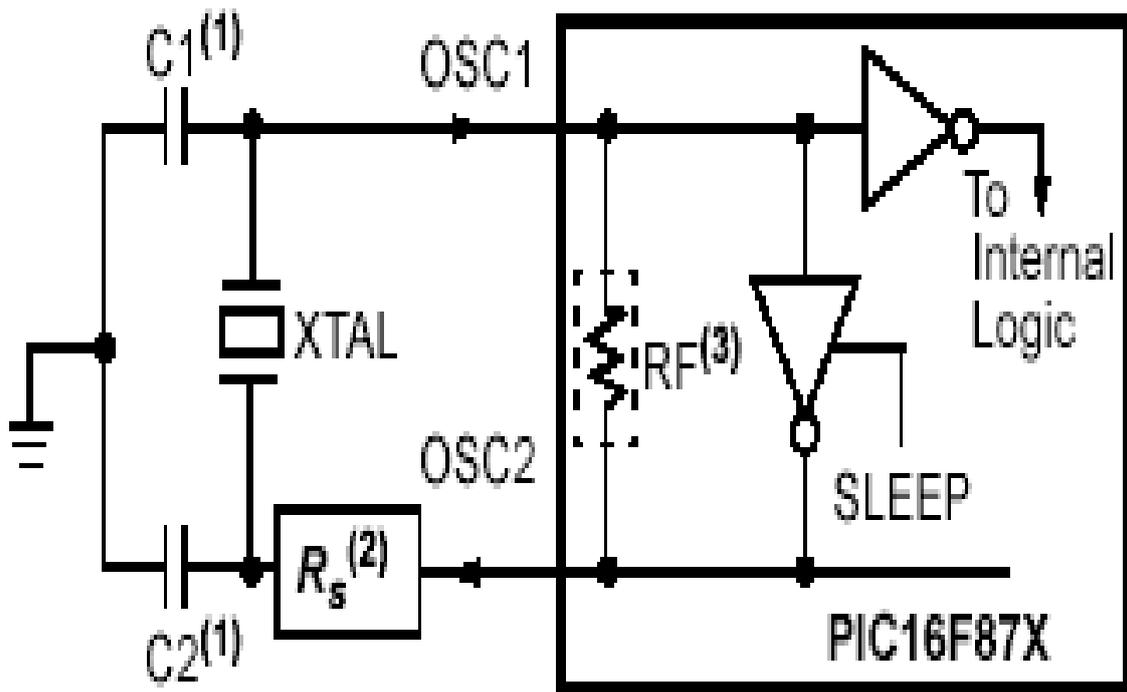


Figure II.9: Oscillateur à Quartz/Céramique

Type d'horloge	fréquence	Suffixe
Faible puissance, oscillateur a quartz	40 MHz	LP
Oscillateur a RC	4 MHz	RC
Oscillateur a quartz standard	4MHz	XT
Oscillateur a quartz rapide	20MHz	HS
Version XT et RC	2MHz	02
Version XT et RC	4MHz	04

Tableau II.1: les différents suffixes avec la gamme de fréquence

II.16.2. Circuit Reset MCLR

Le Reset d'un microcontrôleur peut être déclenché selon plusieurs événements. Ces événements peuvent être d'origine physique, tel que l'appui sur le bouton Reset du montage (broche MCLR) ou encore être dus à une baisse de tension d'alimentation. Un Reset peut également être déclenché suite à un état logiciel tel que le débordement (ou saturation) de la pile (PIC 16).

Pour résumer, un Reset peut être déclenché par :

- un appui sur le bouton Reset du montage (Broche MCLR = 0).
- une instruction logicielle Reset (PIC 16).
- une saturation ou un débordement inférieur de la pile (PIC 16).
- un dépassement de la temporisation du chien de garde (WDT).
- une atteinte du seuil mini de l'alimentation. [8]

II.17. Module de conversion A/N

Le CAN est un périphérique intégré destiné à mesurer un signal analogique (une tension électrique) et le convertir en nombre binaire équivalent qui pourra, être utilisé par un programme. Ce module est constitué d'un convertisseur Analogique Numérique 10 bits dont l'entrée analogique peut être connectée sur l'une des 8 (5 pour 16F876) entrées analogiques externes. 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. L'échantillonneur bloqueur est intègre, il est constitué d'un Interrupteur d'échantillonnage et d'une capacité de blocage de 120 pF. Les tensions de références permettant de fixer la dynamique du convertisseur. Elles peuvent être choisies parmi **Vdd**, **Vss**, **Vrf+** ou **Vrf-**. [8]

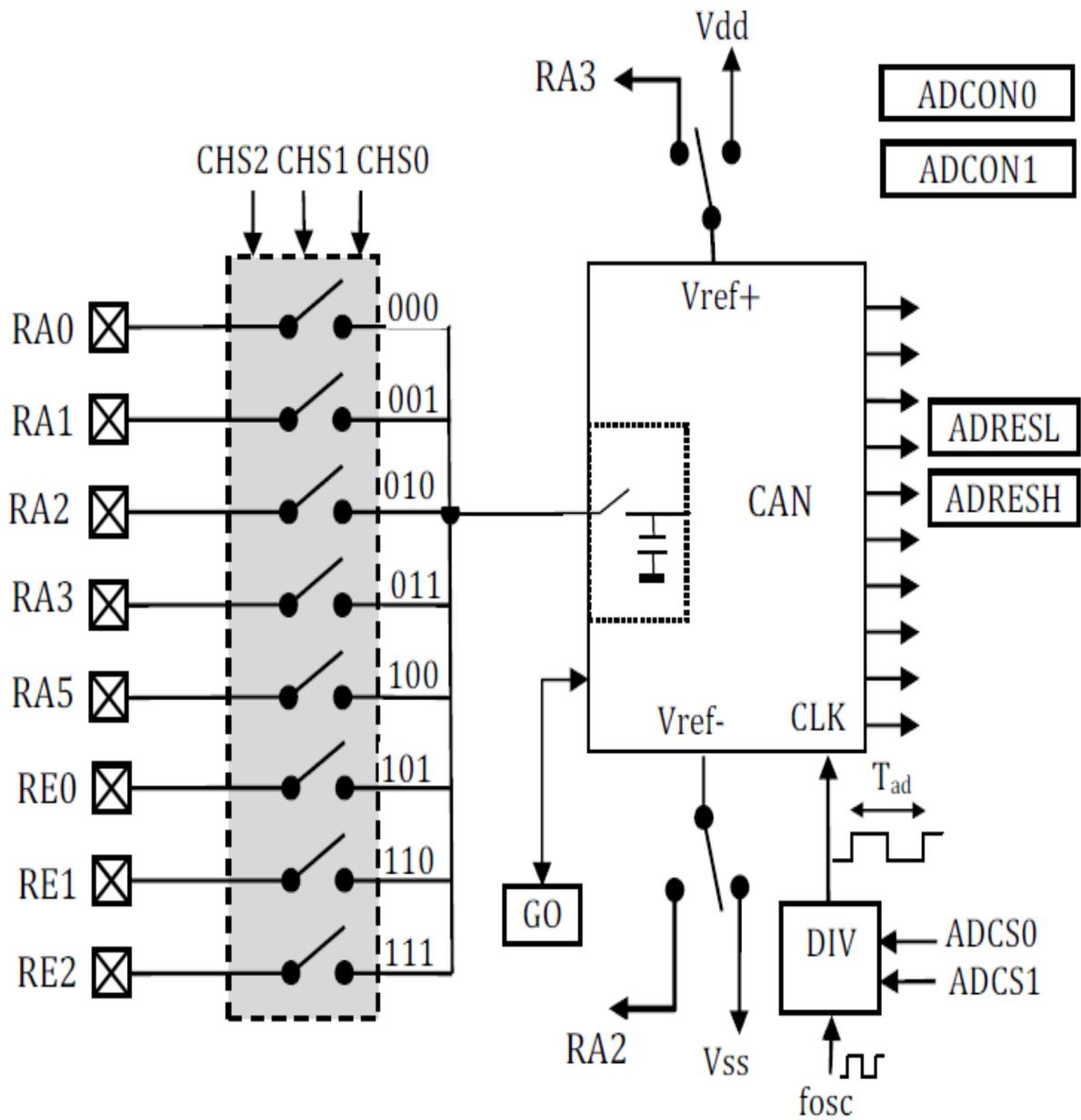


Figure II.10: Le module de conversion A/N

Le microcontrôleur 16F877A travaille avec un convertisseur analogique/numérique qui permet un échantillonnage sur 10 bits. Le signal numérique peut donc prendre 1024 valeurs possibles. On sait que pour pouvoir numériser une grandeur, nous devons connaître la valeur minimale qu'elle peut prendre, ainsi que sa valeur maximale, les PICs considèrent par défaut que la valeur minimale correspond à leur **Vss** d'alimentation, tandis que la valeur maximale correspond à la tension positive d'alimentation **Vdd**.

Le PIC connecte le pin sur laquelle se trouve la tension à mesurer à un condensateur interne, qui va se charger via une résistance interne jusqu'à la tension appliquée. Le pin est déconnecté du condensateur, et ce dernier est connecté sur le convertisseur analogique/numérique interne.

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

diviseur	20Mhz	5Mhz	4 Mhz	2 Mhz	1.25 Mhz	333.3khz
2	100ns	100ns	400ns	500ns	1.6µs	6µs
8	400ns	1.6µs	2µs	4µs	6.4µs	24µs
24	1,6µs	6.4µs	8µs	16µs	25.6µs	96µs
Osc RC	2-6µs	2-6µs	2-6µs	2-6µs	2-6µs	2-6µs

Tableau II.2: Quelques fréquences courantes du quartz et pour les PICs de type Classique

Ce temps est dérivé per division d'horloge principale. La division peut prendre une valeur de 2,8 ou 32. Le temps de conversion Tad ne peut détendre, pour des raisons électroniques, en dessous de 1.6 us pour les versions classiques de 16F87x, et en dessous de 6 us pour les versions LC. Donc en fonction des fréquences utilisées pour le quartz du pic, on choisit le diviseur le plus approprié. Voici le tableau [II.2] qui reprend les valeurs de diviseur à utiliser pour les valeurs 400ns ; 1,6µs ; 2µs ; 4µs ; 6,4µs ; 24µs correspondent au meilleur diviseur en fonction de la fréquence choisie, en ce qui nous concerne puisqu'on utilise une fréquence de 20MHZ, on utilisera la valeur de 2µs dans notre programmation.

Il faut à présent préciser que le PIC nécessite un temps Tad avant le démarrage effectif de la conversion, un temps supplémentaire **Tad** à la fin de la conversion. Résumons donc le temps nécessaire pour effectuer l'ensemble des opérations:

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

Le control du module se fait par les deux registres **ADCON0** et **ADCON1**.

II.18. ADCON0 [8]

ADCS1	A0DCS0	CHS2	CHS1	CHS0	GO/DONE	–	ADON
-------	--------	------	------	------	---------	---	------

Tableau II.3: Le registre ADCON0

B7	ADCS1	A/D conversion Clock Select bit 1
B6	ADCS0	A/D conversion Clock Select bit 0
B5	CHS2	Analog Channel Select bit 2
B4	CHS1	Analog Channel Select bit 1
B3	CHS0	Analog Channel Select bit 0
B2	GO/DONE	A/D Conversion Status bit
B1	Inutilisé	Lu comme 0
B0	ADON	A/D ON bit

Tableau II.4: Registre ADCON0

ADCS1:ADCS0 : Choix de l'horloge de conversion donc du temps de conversion

00 : $F_{osc}/2$

01 : $F_{osc}/8$

10 : $F_{osc}/32$

11 : Oscillateur RC dédiée au CAN (fonctionne seulement pour $F_{OSC} < 1\text{MHz}$).

CHS2:CHS00 : choix de l'entrée analogique

000 = Channel 0, (RA0)

001 = Channel 1, (RA1)

010 = Channel 2, (RA2)

011 = Channel 3, (RA3)

100 = Channel 4, (RA5)

101 = Channel 5, (RE0)

110 = Channel 6, (RE1)

111 = Channel 7, (RE2)

GO/DONE : Une conversion démarre quand on place ce bit à 1. A la fin de la conversion, il est remis automatiquement à zéro. Ce bit peut aussi être positionné automatiquement par le module CCP2.

ADON : Ce bit permet de mettre le module AN en service. [8]

II.19. Registre ADCON1

Ce registre permet de déterminer le rôle de chacune des pins AN0 à AN7. Il permet donc de choisir si un pin sera utilisé comme entrée analogique, comme entrée/sortie standard, ou comme tension de référence. Il permet également de la justification du résultat. Pour pouvoir utiliser un pin en mode analogique, il faudra que ce pin soit configuré également en entrée par TRISA. Le registre ADCON1 dispose, comme tout registre accessible de notre PIC, de 8 bits, dont seulement 5 sont utilisés: [8]

ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
------	---	---	---	-------	-------	-------	-------

Tableau II.5: Registre ADCON1

B7	ADFM	A/D result from et select
B6	Inutilisé	Lu comme<<0>>
B5	Inutilisé	Lu comme<<0>>
B4	Inutilisé	Lu comme<<0>>
B3	PCFG3	Port configuration control bit 0
B2	PCFG2	Port configuration control bit 0
B1	PCFG1	Port configuration control bit 0
B0	PCFG0	Port configuration control bit 0

Tableau II.6: Registre ADCON1

Le bit ADFM permet de déterminer si le résultat de la conversion sera justifié à droite (1) ou à gauche(0). Nous trouvons dans ce registre les 4 bits de configuration des pins liés au convertisseur analogique/numérique. Ces bits nous permettent donc de déterminer le rôle de chaque pin. Comme nous avons 16 combinaisons possibles, nous aurons autant de possibilités de configuration.

PCFG3:PCFG0 : configuration des E/S et des tensions de références.

Les 5 broches de PORTA et les 3 de PORTE peuvent être configurées soit en E/S digitales, soit en entrées analogiques. RA2 et RA3 peuvent aussi être configurées en entrée de référence.

II.19. Conclusion

Dans ce chapitre nous avons présenté brièvement l'architecture et les caractéristiques du microcontrôleur PIC 16F877A.

En conclusion, nous pouvons dire que le microcontrôleur 16F877A peut bien jouer le rôle d'une unité de contrôle pour notre système.

Chapitre III

Programmation de pic 16f877A et la simulation

III.1. Introduction

Le programme est un ensemble de commandes (mnémonique) que devra exécuter le microcontrôleur l'une à l'autre. Le PIC est supportée par plusieurs langages de programmation tel que MikroC for PIC, MPLAB, MikroBasic PRO for PIC, HI-TECH C for PIC ...etc. Dans notre projet nous avons choisi le compilateur de MIKRO C qui est un compilateur en langage C (langage évolué) et qui permet d'intégrer certaines routines en assembleur.

Ce choix est à la fois un choix personnel et un choix technologique parce qu'il est basé sur le langage C.

III.2. Etapes de développement du programme

L'élaboration d'un programme est un travail qui se fait en plusieurs étapes (Figure II.1):

- La première étape : L'algorithme
- La deuxième étape : Ecriture du programme
- La troisième étape : Simulation du programme [2]

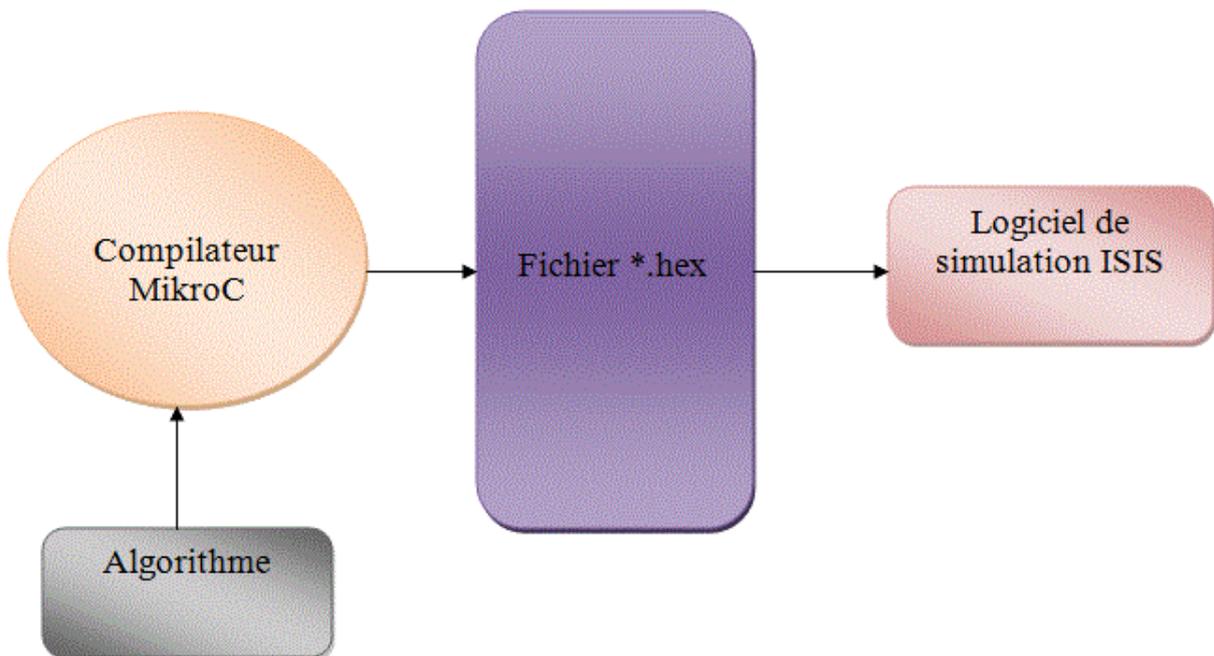


Figure III.11: Etape dévolution d'un programme

III.3. Langage et Compilateur MikroC pour PIC

Le Compilateur **MikroC** pour PIC a trouvé une large application pour le développement de systèmes embarqués sur la base de microcontrôleur. Il assure une combinaison de l'environnement de

programmation avancée IDE (Integrated Développement Environnement), et d'un vaste ensemble de bibliothèques pour le matériel, de la documentation complète et d'un grand nombre des exemples.

Le compilateur **MikroC** pour PIC bénéficie d'une prise en main très intuitive et d'une ergonomie sans faille. Ces très nombreux outils intégrés (mode simulateur, terminal de communication Ethernet, terminal de communication USB, gestionnaire pour afficheurs 7 segments, analyseur statistique, correcteur d'erreur, explorateur de code, mode Débug ICD...) associé à sa capacité à pouvoir gérer la plupart des périphériques rencontrés dans l'industrie (Bus I2C™, 1Wire™, SPI™, RS485, Bus CAN™, USB, gestion de cartes compact Flash et SD™/MMC™, génération de signaux PWM, afficheurs LCD alphanumériques et graphiques, afficheurs LEDs à 7 segments, etc...) en font un outil de développement incontournable pour les systèmes embarqués, sans aucun compromis entre la performance et la facilité de débogage. [9]

III.4. Compilateur mikroC PRO pour PIC

La nouvelle version appelée *mikroC PRO* dispose de très nombreuses améliorations du compilateur *mikroC* : nouvelles variables utilisables, nouvelle interface IDE, amélioration des performances du linker et de l'optimisateur, cycle de compilation plus rapide, code machine généré plus compact (jusqu'à 40 % suivant les cas), nouveaux PIC supportés, environnement de développement encore plus ergonomique, nouveaux exemples d'applications , etc... [9]

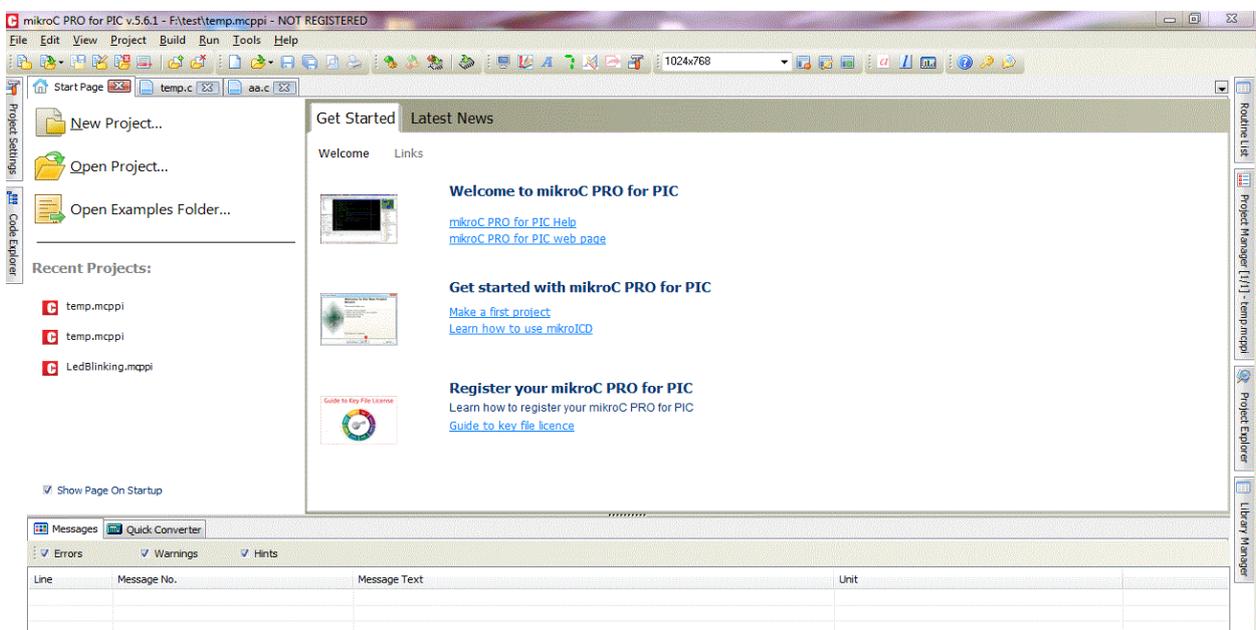


Figure III.12: Interface du logiciel MikroC

III.5. Création d'un projet

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

Les fichiers source peut être compilé que si elles font partie d'un projet.

Le fichier projet contient les informations suivantes :

- 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.)
- D'autres fichiers

La meilleure façon de créer un projet c'est à l'aide de l'Assistant Nouveau projet (menu *Project > New Project*) ou en cliquant sur l'icône *Nouveau projet* à partir de la barre d'outils du projet. [9]

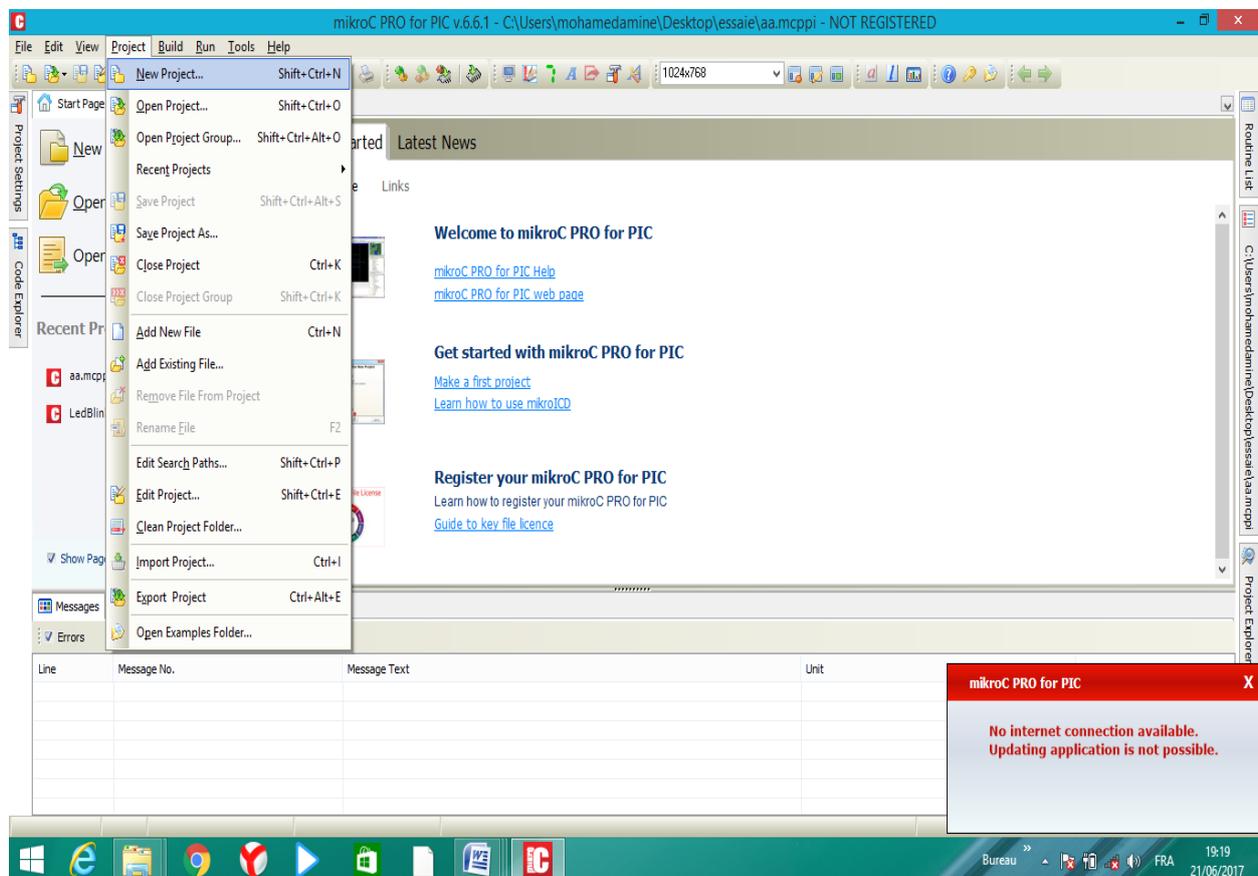


Figure III.13: Création d'un projet en MikroC

Une nouvelle fenêtre apparaîtra. Comme indiqué sur **(Figure III.4)**, il y a plusieurs champs à renseigner comme le nom du projet, l'emplacement du projet, sa description, l'horloge et les options du composant. Le tableau device flags (options composant) est utilisé pour la configuration des paramètres du microcontrôleur.

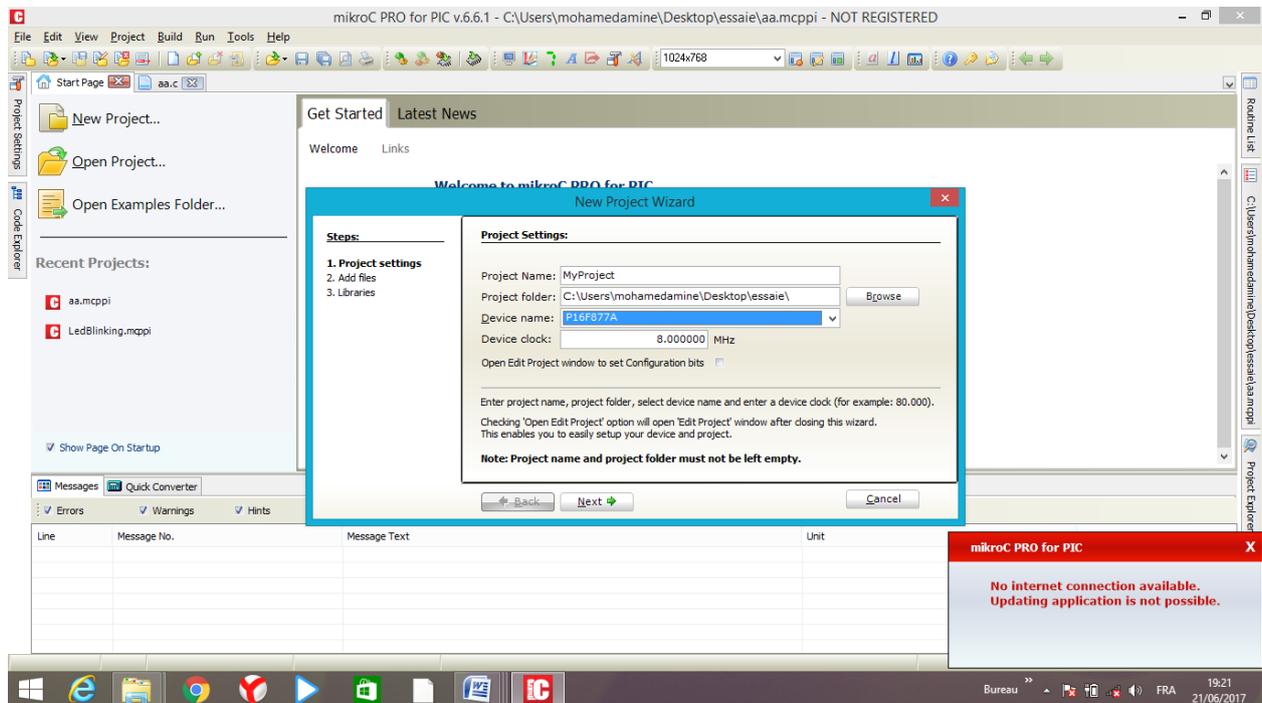
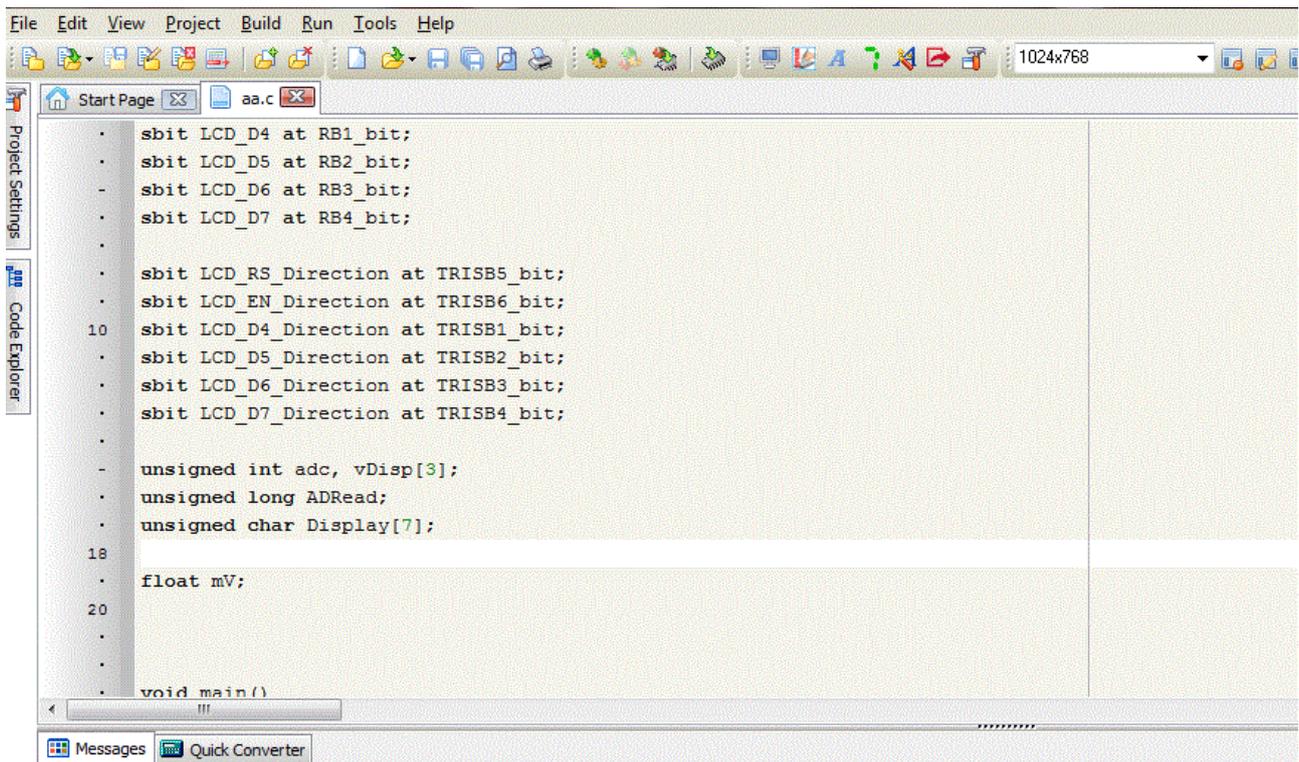


Figure III.14: Les configurations de projet

Après ça, une nouvelle fenêtre (**Figure III.5**) vide s'affiche pour écrire notre programme



```
File Edit View Project Build Run Tools Help
1024x768
Start Page aa.c
Project Settings
Code Explorer
• sbit LCD_D4 at RB1_bit;
• sbit LCD_D5 at RB2_bit;
• sbit LCD_D6 at RB3_bit;
• sbit LCD_D7 at RB4_bit;
•
• sbit LCD_RS_Direction at TRISB5_bit;
• sbit LCD_EN_Direction at TRISB6_bit;
10 sbit LCD_D4_Direction at TRISB1_bit;
• sbit LCD_D5_Direction at TRISB2_bit;
• sbit LCD_D6_Direction at TRISB3_bit;
• sbit LCD_D7_Direction at TRISB4_bit;
•
• unsigned int adc, vDisp[3];
• unsigned long ADRead;
• unsigned char Display[7];
18
• float mV;
20
•
•
• void main()
```

Figure III.15: Le programme

III.6. Compilation

Lorsque vous avez créé le projet et écrit le code source, il est temps de le compiler.

Sélectionnez **Project** → **Build** à partir du menu déroulant ou cliquez sur l'icône **Build** dans la barre d'outils du projet.

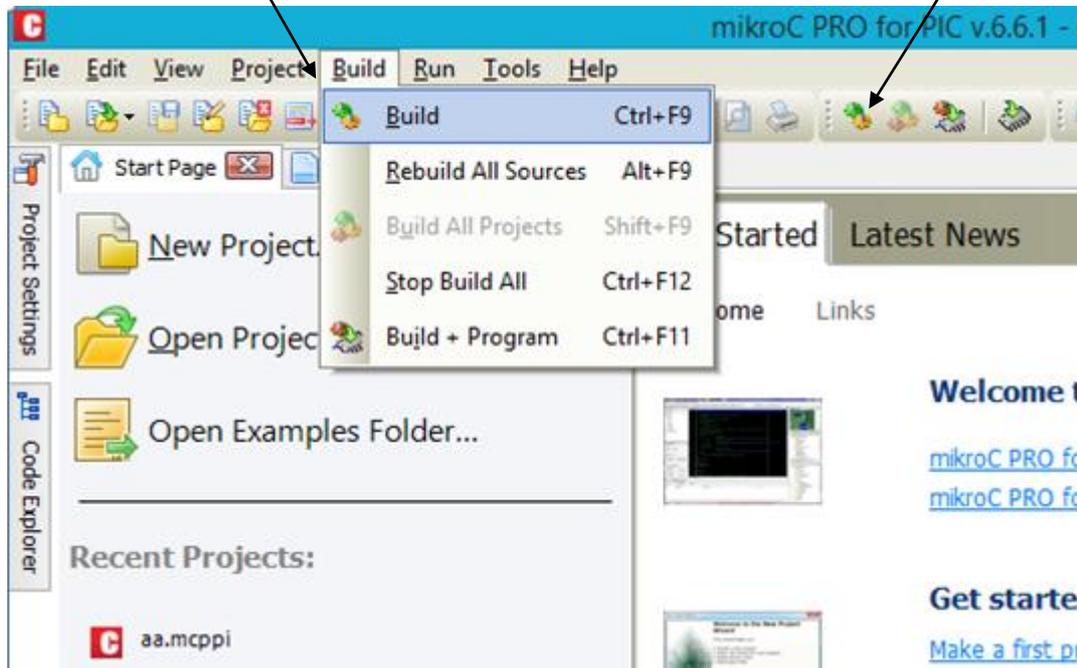


Figure III.16: La compulsation de programme

III.7. Les commentaires

Les commentaires sont utilisés pour préciser le fonctionnement du programme et pour une annotation du programme. Les lignes de commentaires sont ignorées et non compilées par le compilateur.

En mikroC les commentaires de programmes peuvent être de deux types: de *longs commentaires*, s'étendant sur plusieurs lignes, et de *courts commentaires*, occupant une seule ligne.

Un commentaire au début d'un programme peut décrire brièvement le fonctionnement du programme et fournir le nom de l'auteur, le nom de fichier de programme, la date à laquelle le programme a été écrit, et une liste des numéros de version, ainsi que la modification dans chaque version. [9]

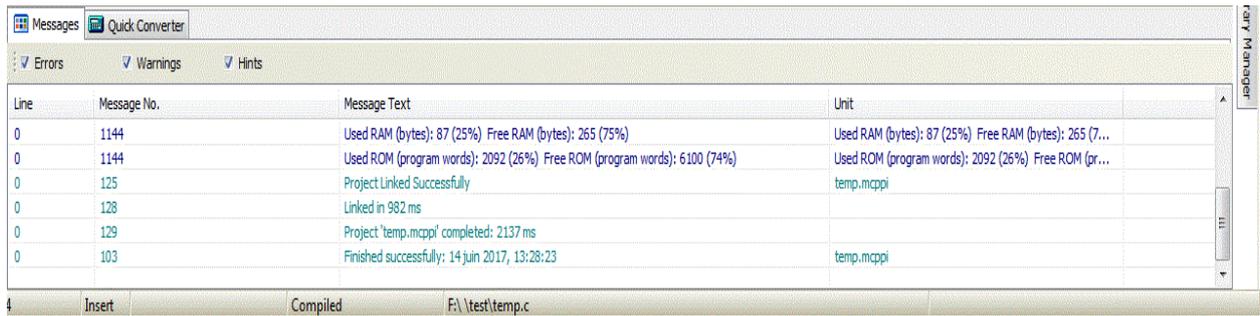


Figure III.17: Affichage des résultat

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
<i>Intel HEX</i>	Code hexadécimal dans le format Intel. Fichier est utilisé pour programmer PIC	<i>.hex</i>
<i>Binary</i>	Fichier compilé pour la bibliothèque <i>mikroC</i> . Les distributions binaires sont des routines qui susceptibles	<i>.mcl</i>

Tableau III.7: Les fichiers de sortie

III.8. Sauvegarder un fichier

Assurez-vous que la fenêtre contenant le fichier que vous voulez sauvegarder est active. Sélectionnez ensuite *Save* depuis le menu *File* ou pressez *Ctrl+S* ou cliquez sur l'icône *Save as* de la barre d'outils *File*. [9]

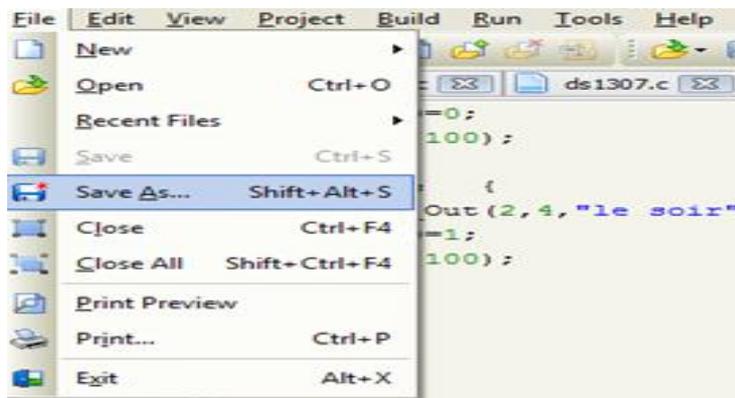


Figure III.18: Sauvegarder un fichier

III.9. Quelques notions de programmation en C sous MikroC

La saisie d'un programme en 'C' répond pratiquement toujours à la même structure. On peut noter que :

- Le symbole “#” est suivi d'une directive.
- Le symbole “//” est suivi d'un commentaire, ou bien “/*” indiquant le début d'un commentaire et “*/” indiquant sa fin.
- Chaque ligne d'instruction se termine par un “;”.
- Le début d'une séquence est précédé du symbole “{”.
- La fin d'une séquence est suivie du symbole “}”.
- La notation des nombres peut se faire en décimal de façon normale ou en hexadécimal avec le préfixe “0x” ou encore en binaire avec le préfixe “0b”. [10]

III.10. Déclaration des variables en C

Les variables de base dans ce compilateur spécifique sont:

- *bit*
- *char*
- *short*
- *int*
- *long*
- *float*
- *double*

Les variables *bit* peuvent stocker un booléen qui peut être vrai ou faux, 0 ou 1.

Les variables *char* sont utilisés pour stocker des caractères codés avec le code ASCII. Il est utile pour stocker des lettres ou des textes.

Une variable *short* contient un entier 8 bits qui peut prendre des valeurs : -127 à 127. [11]

Les variables de type *int* sont des entiers 16 bits, cette variable peut stocker des nombres : -32767 à 32767.

La variable type *long* stocke un entier sur 32 bits, sa plage de valeurs est: -2147483647 à 2147483647.

Les variables type *double* et *float* permettent de stocker des nombres avec virgule.

Les variables ci-dessus peuvent être déclarées, y compris le signe positif et négatif, ou peuvent être déclarées au moyen de la directive *unsigned*.

Dans le tableau ci-dessous, on peut voir les caractéristiques des variables. [11]

Type de variable	Taille en octets	Valeurs
Bit	1	0 ou 1
Char	1	-127 à 127
Short	1	-127 à 127
Int	2	- 32767 à 32767
Long	4	- 2147483647 à 2147483647
Float	4	-1.5x10 ⁴⁵ à 3.4x10 ³⁸
Double	4	-1.5x10 ⁴⁵ à 3.4x10 ³⁸
unsigned char	1	0 à 255
unsigned short	1	0 à 255
unsigned int	2	0 à 65535
unsigned long	4	0 à 4294967295

Tableau III.8: Les caractéristiques des variables.

La déclaration des variables s'effectue en indiquant le type de la variable suivi d'un nom que le développeur attribue arbitrairement à la variable. Au moment de la déclaration d'une variable, il est possible de lui donner une valeur initiale, mais ceci n'est pas strictement nécessaire. Enfin, la déclaration doit se terminer par un point-virgule (;).

Dans les exemples suivants, on peut voir comment les déclarations sont faites:

```

bit VARIABLE1_BIT;           //Déclaration d'une variable type bit.
char CARACTERE;             // Déclaration d'une variable type char.
char CARACTERE2='J';        //Déclaration d'une variable de type char initialisée
                               //avec la valeur ASCII le caractère J.
int ENTIER=1234;             // Déclaration d'une variable entière initialisée avec
                               //la valeur 1234.
float DECIMAL=-12.45 ;      // Déclaration d'une variable avec un point virgule
                               //initialisée avec la valeur -12,45.
double DECIMAL2=56.68;      // Déclaration d'une variable avec un point virgule
                               //initialisée avec la valeur 56,68.
long ENTIER2=-954261;       // Déclaration d'une variable entière de type long
                               // initialisée avec la valeur -954261.

```

Les exemples suivants montrent comment déclarer des variables non signées:

```

unsigned char CARACTERE;    //Déclaration d'une variable type char non signé.
unsigned int ENTIER;         //Déclaration d'une variable type entier non signé.
unsigned long ENTIER2;      // Déclaration d'une variable type entier long non
                               //signé.

```

Les variables peuvent également être déclarées dans un format qui associe plusieurs variables dans un même nombre, ce format est connu comme une chaîne de variables ou un vecteur et peut-être même un tableau de variables, en conclusion ce genre de déclaration peut avoir une ou plusieurs dimensions.[11]

III.11. Les opérateurs en langage C

Le langage C permet de faire des opérations arithmétiques de base entre des variables ou des constantes.

Les opérations arithmétiques disponibles sont:

- *Somme*
- *soustraction*
- *Multiplication*
- *Division*
- *Modulo*

L'exemple suivant montre la somme arithmétique de deux nombres ou plus:

```
int A;
```

```
int B;
```

```
int C;
```

```
C = A+B;           //cet expression stocke la somme de A et B dans la variable C.
```

```
C = A+B+C;        //cet expression stocke. [11]
```

III.12. Présentation du logiciel ISIS de Proteus

Le simulateur ISIS de Proteus est un logiciel puissant, développé par la compagnie électronique Labcenter, qui s'est positionné pour plus de 10 ans comme l'un des outils les plus utiles pour simuler les microcontrôleurs PICmicro.

L'ISIS permet la simulation des familles des PICmicro les plus populaires tels que: 12F, 16F, 18F. En plus des PIC, ISIS peut simuler une variété de dispositifs numériques et analogiques, y compris les afficheurs sept segments, les LCD des caractères et graphiques. ISIS permet de simuler des capteurs de température, l'humidité, la pression et la lumière, entre autres.

Le simulateur peut simuler les actionneurs tels que des moteurs CC, les servomoteurs, les lampes à incandescence, entre autres. Il est possible de simuler des périphériques d'entrée et de sortie comme les claviers, ordinateur et les ports physiques tels que RS232 et USB. Ce simulateur a une grande variété d'instruments de mesure tels que voltmètres, ampèremètres, oscilloscopes et analyseurs de signaux.

En conclusion ces caractéristiques et d'autres font d'ISIS de Proteus, un outil idéal pour la conception et l'étude

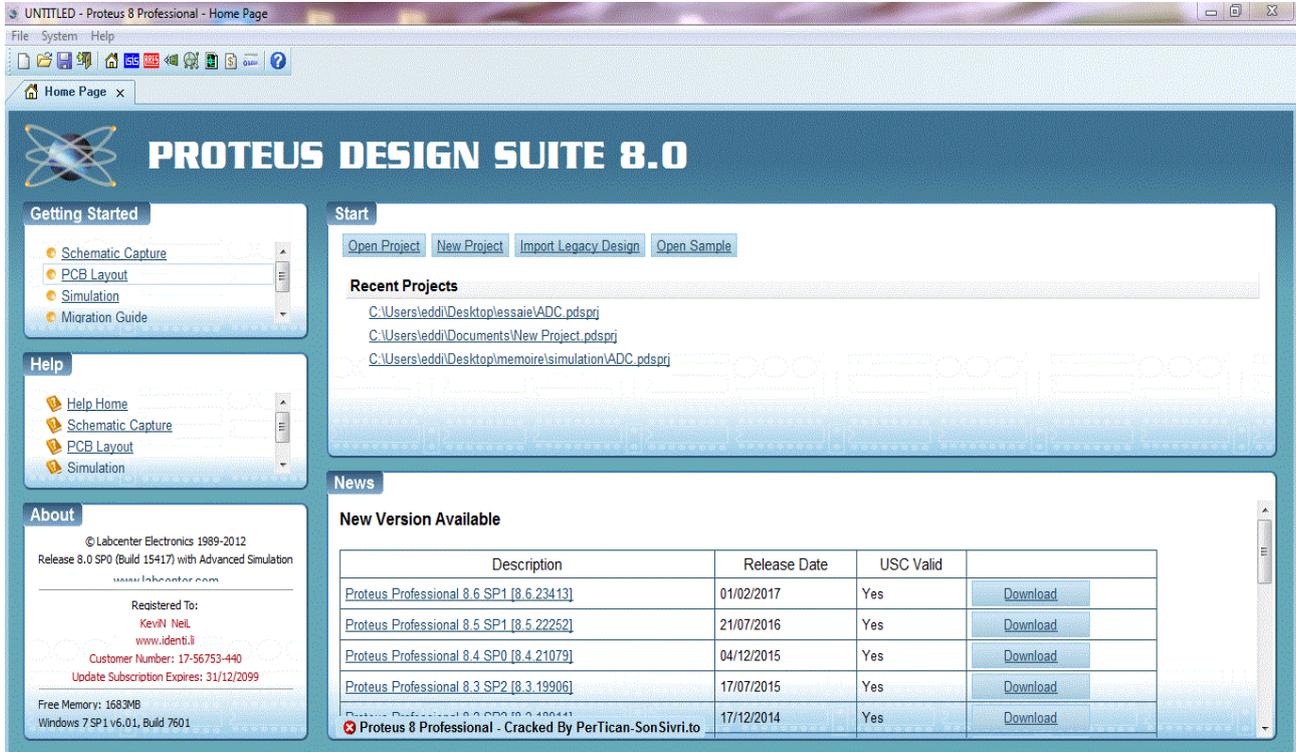


Figure III.19: logiciel ISIS de Proteus

La section suivante montre les caractéristiques de base pour commencer la première simulation dans ISIS. Dans une première étape, on identifie la palette *sélecteur d'objets*, c'est-à-gauche de l'écran, le développeur doit identifier la touche P dans la palette comme le montre l'image ci-dessous:

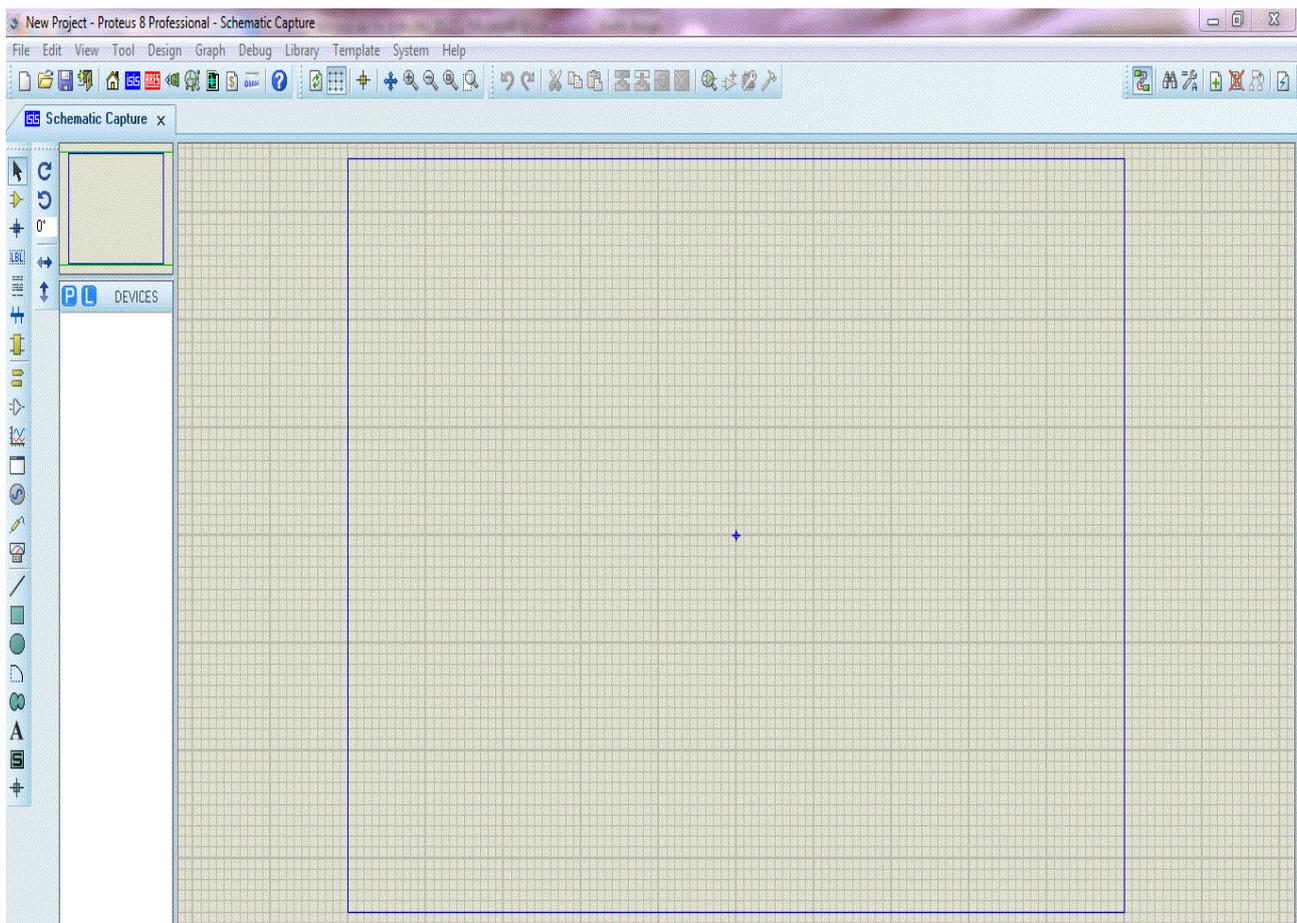


Figure III.20: Les caractéristiques de base pour commencer la simulation dans Isis.

Lorsqu'on appuie sur la touche P, le programme ouvre une nouvelle fenêtre qui permet de trouver les dispositifs électroniques à travers la référence commerciale, ou sous la classification adopté par ISIS.

Pour rechercher un dispositif, on entre la référence dans la zone: Keywords, le programme génère une liste, sur le côté droit de la fenêtre, des dispositifs en fonction de la recherche de l'utilisateur.

Pour sélectionner un dispositif, on double-clique sur sa référence dans la liste donnée. Pour commencer, on repère les dispositifs suivants : SWITCH, LED, RES, Après ce processus, dans la palette on devrait avoir les appareils suivants:



Figure III.21: Les dispositifs électroniques

La prochaine étape consiste à trouver les bornes de la masse et l'alimentation, à cet effet, on clique sur l'icône suivant:  dans la palette d'outils, qui se trouve sur le côté gauche de la fenêtre du programme.

Lorsqu'on clique sur cet outil, une liste de terminaux qui apparaît, on trouve GROUND et POWER, qui correspondent respectivement à la référence électrique ou la masse et à l'alimentation ou Vcc. La borne d'alimentation a une différence de potentiel de 5 volts par défaut. Pour placer ces terminaux dans la zone de travail, on clique sur les éléments dans la palette des terminaux, puis dans la zone de travail. Après cette action, la zone de travail devrait être comme la figure suivante:

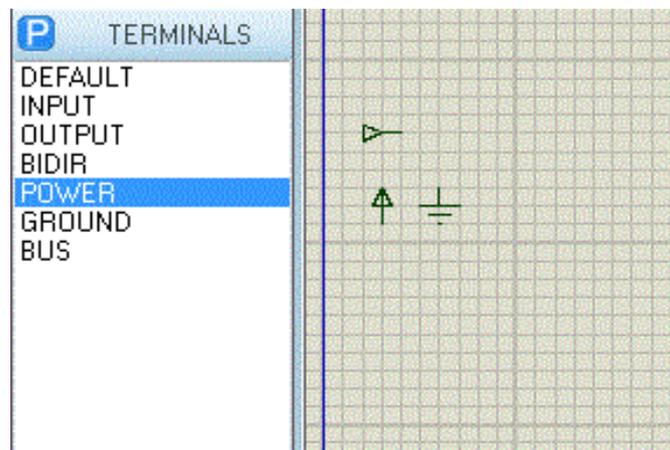


Figure III.22: Les bornes de la masse et l'alimentation

L'étape suivante consiste à fixer les dispositifs dans la zone de travail, on choisit le bouton:  dans la palette d'outils sur la gauche. Pour coller les dispositifs dans l'espace de travail, on procède de la même manière que les terminaux, enfin on aura l'espace de travail suivant:

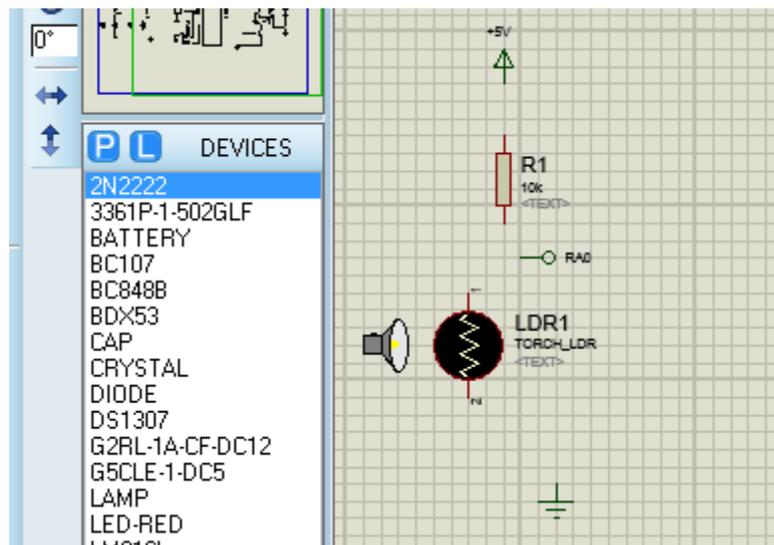


Figure III.23: Fixation des dispositifs

La prochaine étape est d'effectuer les connexions électriques, pour cet exemple, les éléments sont connectés, à cet effet le curseur de la souris prend la forme d'un crayon, pour lier entre deux bornes on clique sur la première puis la prochaine et le programme termine la connexion. À la fin des connexions, l'espace de travail devient:[11]

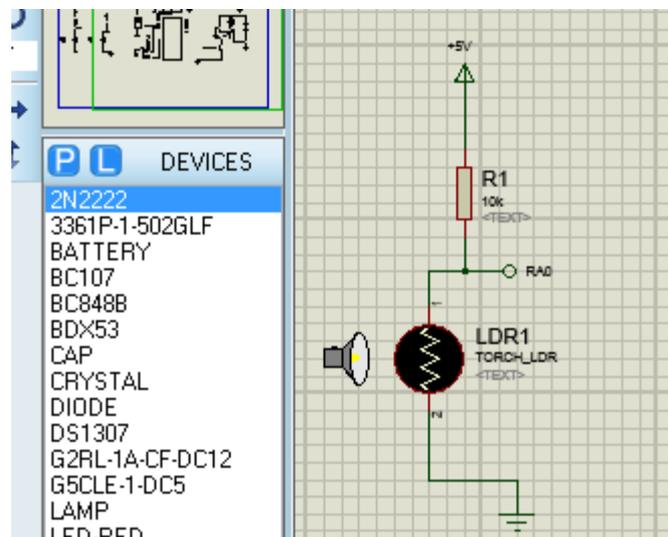


Figure III.24: Effectuer les connexions électriques

Enfin, on peut modifier la valeur de la résistance par un double clic de la souris sur la résistance, avec cette action on affiche une fenêtre pour modifier la valeur de la résistance, qui est par défaut 10k, ce qu'il faut faire est de la changer à 330.

L'étape suivante consiste à lancer la simulation, la palette située dans le coin inférieur gauche de l'écran permet de contrôler les simulations. Cette palette a l'apparence suivante:[11]



Figure III.25: Lancement de la simulation

Enfin, on appuie sur le bouton *play* et la simulation fonctionne, lorsque la simulation est en cours, on peut appuyer sur le bouton poussoir, et voir l'effet sur la simulation. Cet exemple simple montre le processus de création d'un schéma sur ISIS.

La création des circuits en ISIS, implique une autre série d'actions qui seront abordés au cours des exemples. [11]

III.13. Simulation par ISIS

Pour le dessin du schéma électrique de notre projet on utilisée le logiciel ISIS pour le simuler comme indique la figure (**Figure III. 16**)

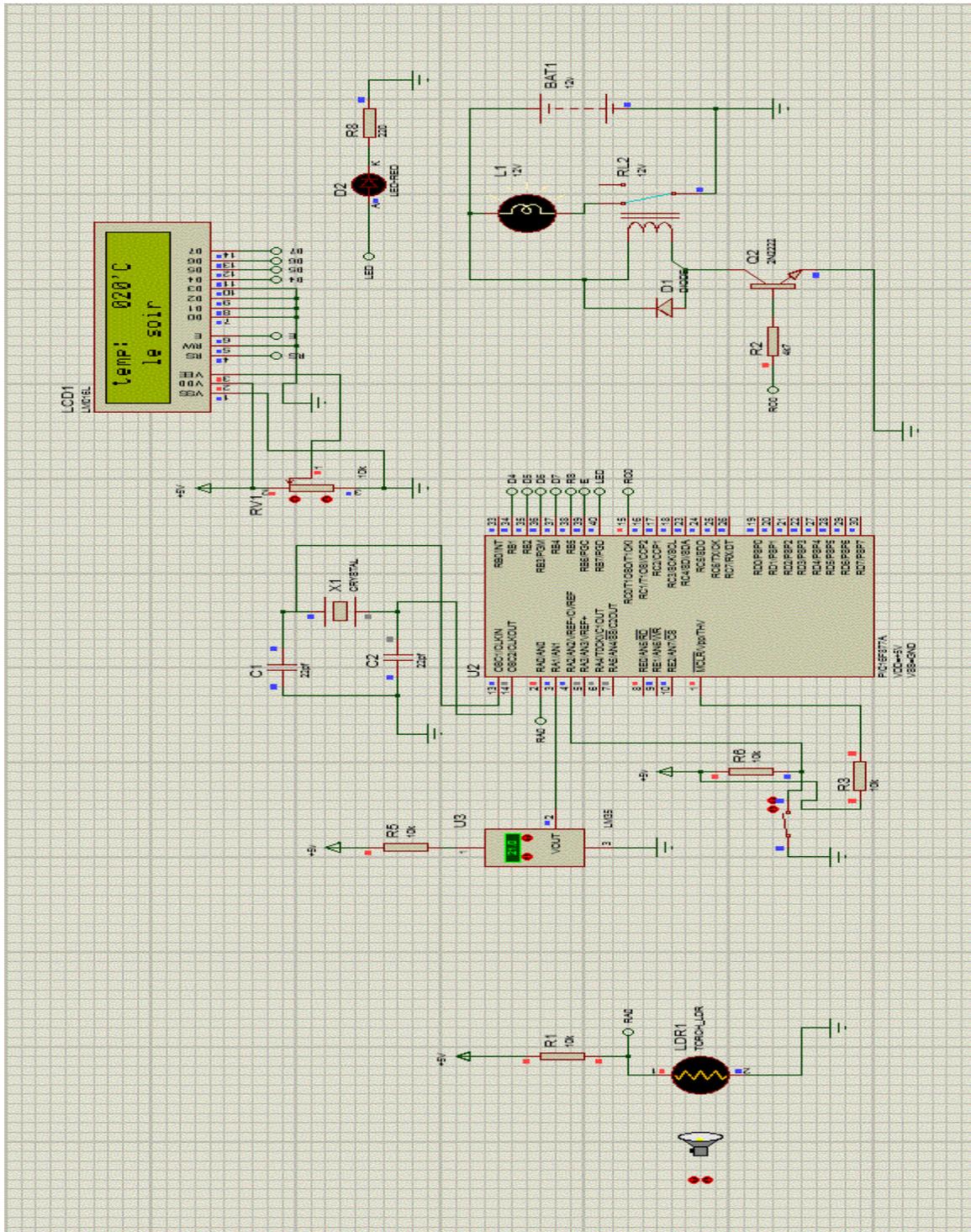


Figure III.26: Le logiciel ISIS 7 en mode simulation du projet

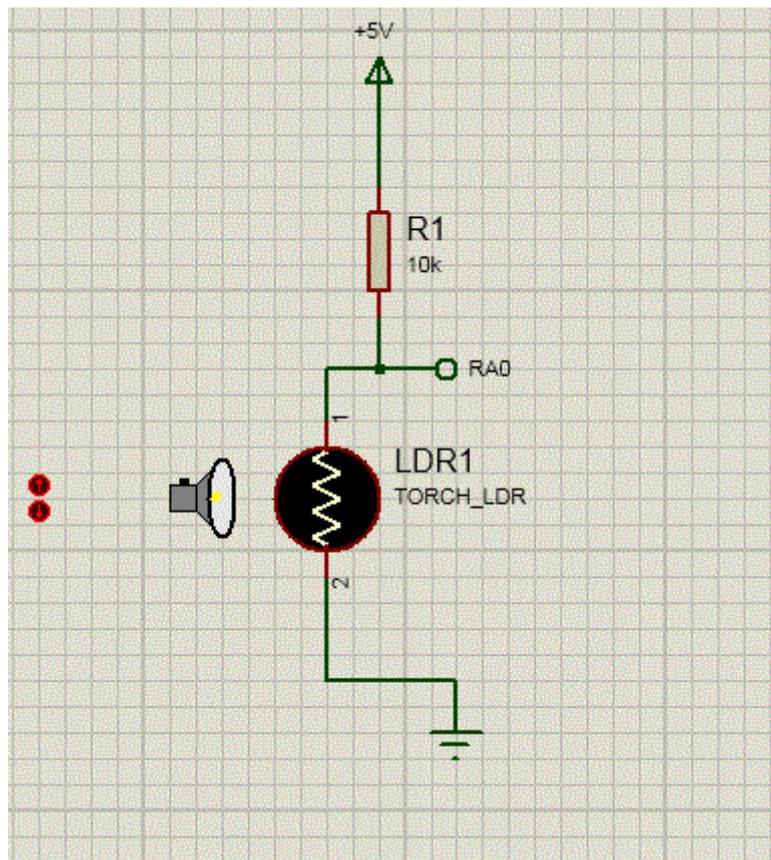
III.14. Le capteur de lumière

III.14.1. Définition

Une photorésistance est un composant électronique dont la résistivité varie en fonction de la quantité de lumière incidente. On peut également la nommer résistance photodépendante (light-dependent resistor (LDR)) ou cellule photoconductrice

Une photorésistance est composée d'un semi-conducteur à haute résistivité. Si la lumière incidente est de fréquence suffisamment élevée, elle transporte une énergie importante. Au-delà d'un certain niveau propre au matériau, les photons absorbés par le semi-conducteur donneront aux électrons liés assez d'énergie pour passer de la bande à la bande de conduction [12]

Dans notre simulation nous avons utilisé un capteur de lumière LDR1

**Figure III.27:** Le capteur de lumière

N° Broche	NOM	Description
1	VSS	Masse
2	VDD	Alimentation positive (V+)
3	VEE	Contraste réglable par potentiomètre *
4	RS	Sélection de registre (0=instruction;1=données)
5	R/W ou RD	Lecture ou écriture (1=lecture;1=écriture)
6	E	Enable (validation, actif au niveau haut)
7	D0	Bit 0 du bus de données
8	D1	Bit 1 du bus de données
9	D2	Bit 2 du bus de données
10	D3	Bit 3 du bus de données
11	D4	Bit 4 du bus de données
12	D5	Bit 5 du bus de données
13	D6	Bit 6 du bus de données
14	D7	Bit 7 du bus de données

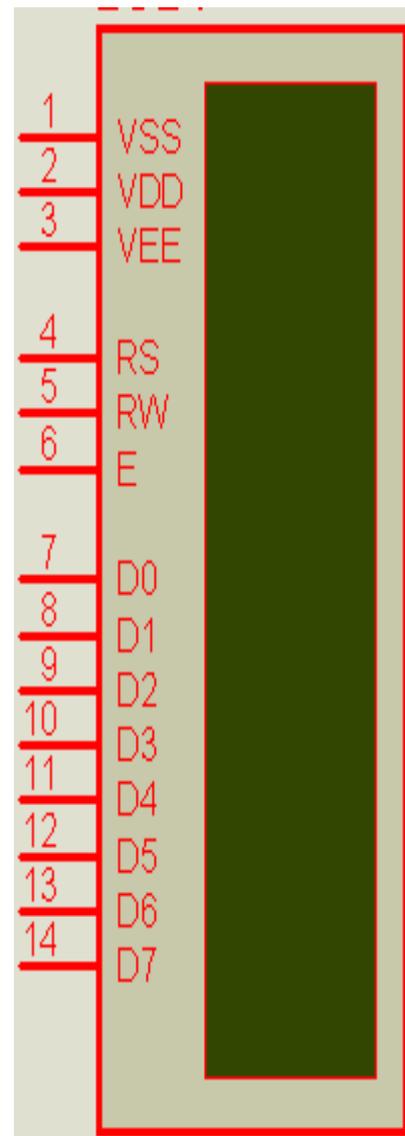


Figure III.30: Brochage d'afficheur LCD

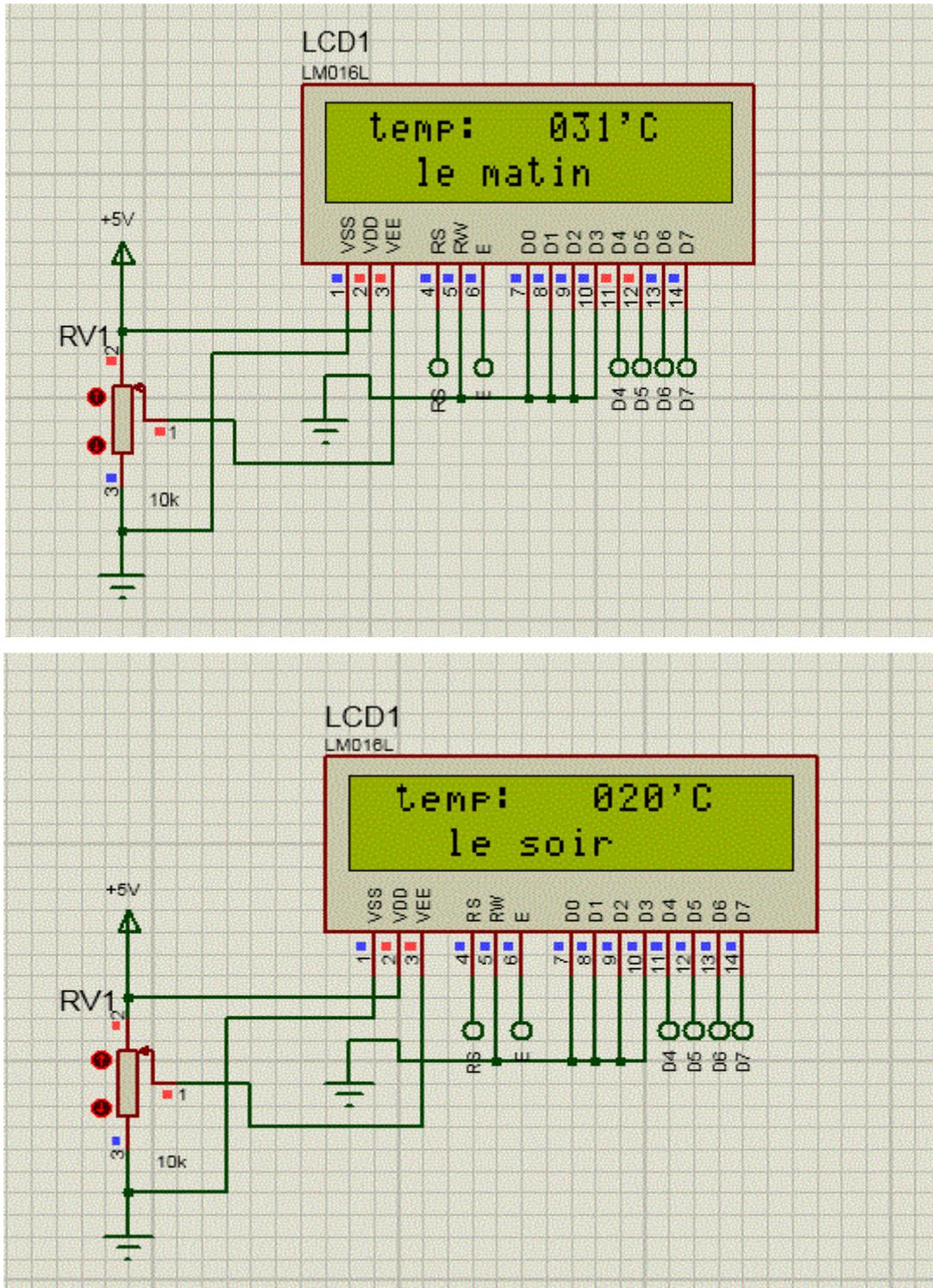


Figure III.31: Interface de LCD avec le pic 16f877A dans la simulation

III.17. Les LED

Une diode LED (*Light-Emitting Diode*) est un composant électronique particulier utilisé comme source de lumière. Pour chacune des LEDs, l'utilisation d'une résistance pour limiter le courant est indispensable. La tension aux bornes d'une LED standard est de l'ordre de 2.5V tandis que le courant peut varier entre 1 mA et 20mA.

Dans notre projet nous avons utilisé une LED détecter si on na un problème dans notre système

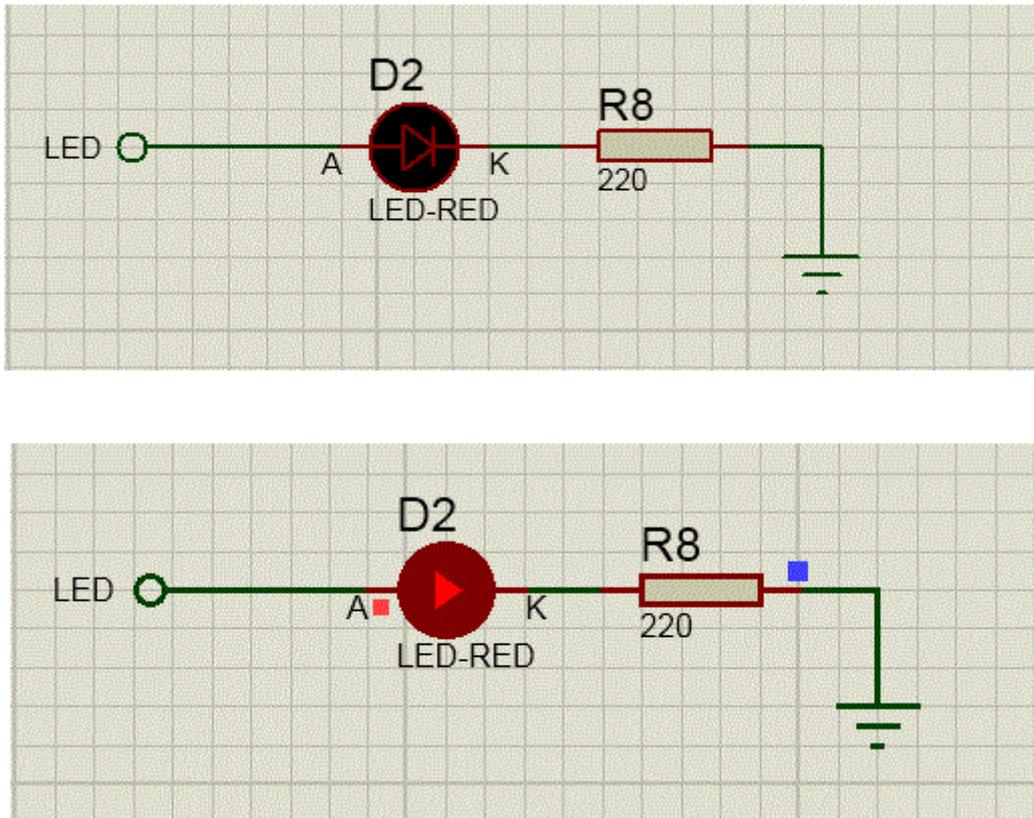


Figure III.32: Résultat de la LED utilisé

III.18. Unité de contrôle

L'unité de contrôle est basée sur le PIC16F877A. Les ports du PIC seront configurés comme suit :

III.18.1. Le port A

Le bit RA0 est configuré comme entrée analogique du capteur de lumière LDR, le bit RA1 est configuré comme entrée analogique du capteur de température lm35 et le bit RA2 configuré comme entrée analogique de switch,

Et les deux bits OSC2, OSC1 comme entrée de l'oscillateur HS 20MHZ.

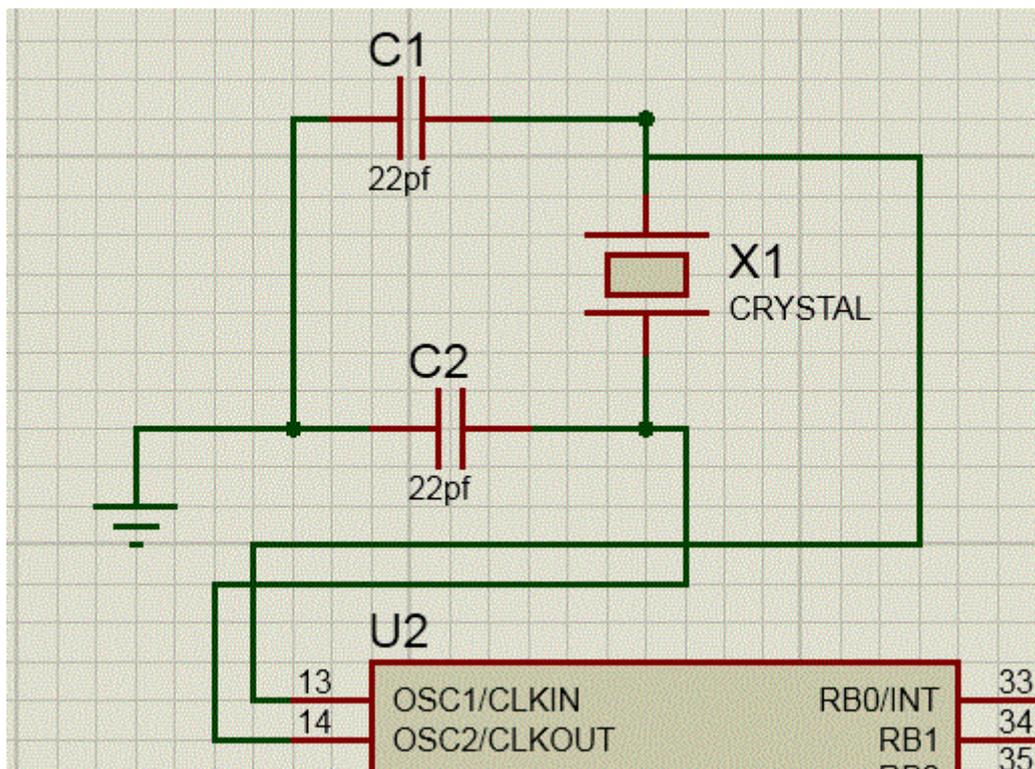


Figure III.33: L'oscillateur (20MHZ)

III.18.2. Le port B

Les six bits restant vers l'afficheur LCD, et le bit RB7 configuré comme sortie de LED de panne

III.18.3. Le port C

Nous utilisons un seul bit de port c, le bit RC0 est configurés comme sortie de relais

III.19. Schéma synoptique de système

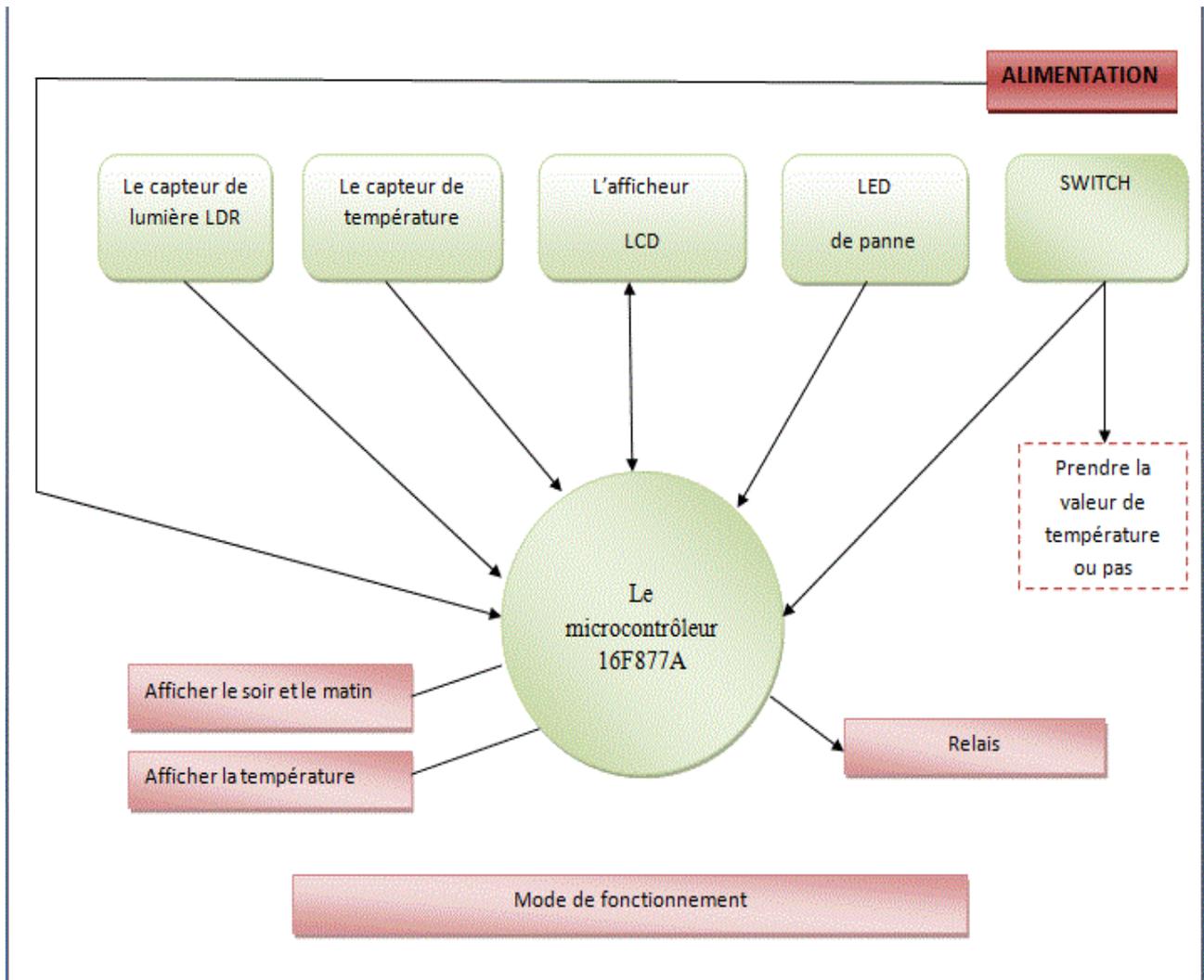


Figure III.34: Schéma synoptique de système

Dans notre système de commande le mode de fonctionnement est basé sur le capteur de lumière, ce dernier envoient un signal électrique vers le PIC, ce signal est limité entre 0 et V_{reff} de telle sorte que $v_{\text{reff}} = 2.5V$, la valeur minimale(0) représente la lumière et la valeur maximal (v_{reff}) représente l'obscurité. La valeur de crépuscule existe entre les deux valeurs précédentes et définie par l'utilisateur.

Dans le cas de jour le PIC envoie un signal à l'interface de puissance à travers le port C (*le bit RC0*), égale à 0 logique pour stopper le flux de courant dans le relais et donc couper l'éclairage.

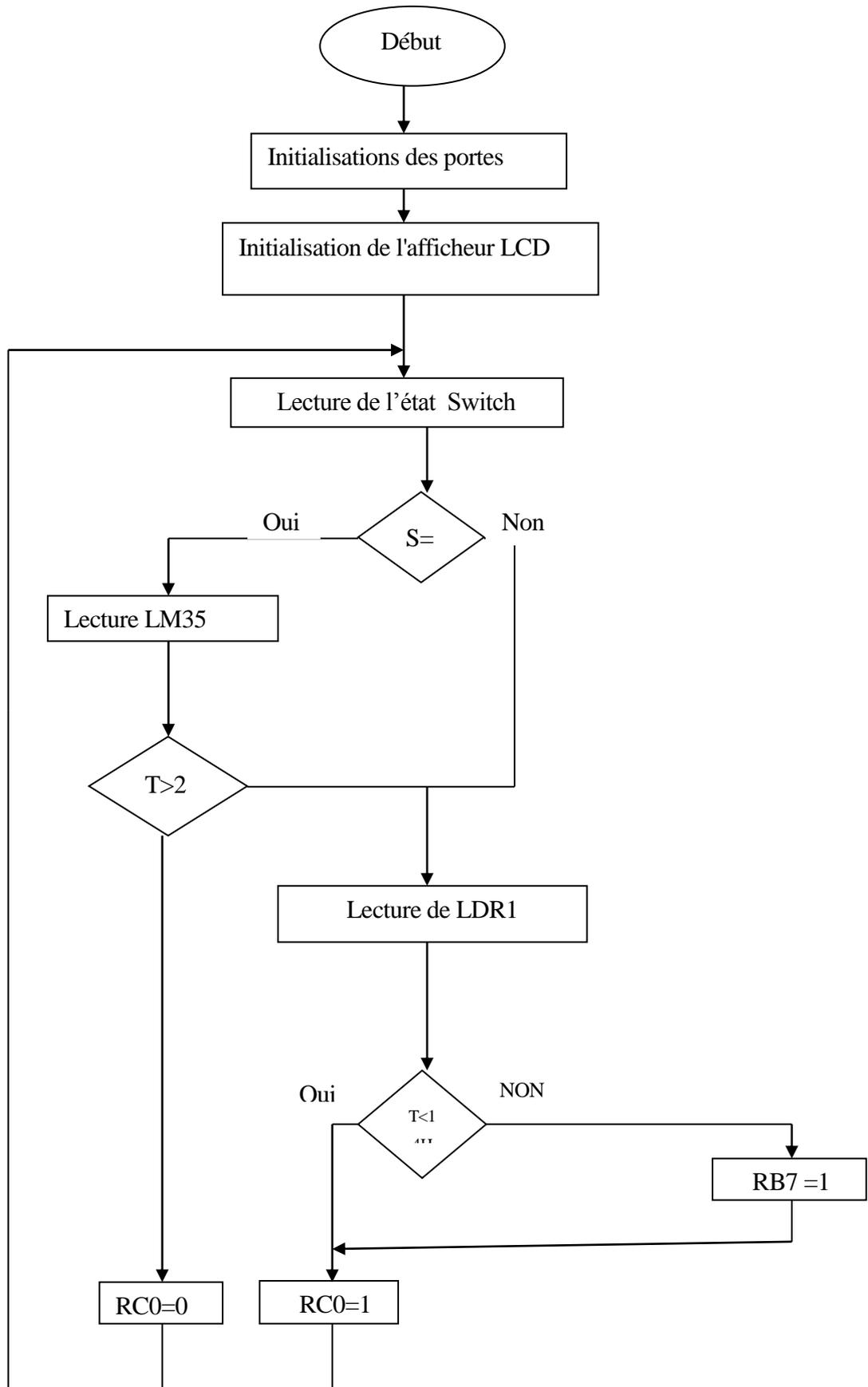
Dans le cas de soir le pic envoie un signal égal à 1 logique pour allumer l'éclairage.

En pratique on a rencontré un problème, lorsque le ciel est nuageux, il y a de l'ombre ou de la poussière qui couvre les capteurs de lumière, ainsi le système ne fonctionne pas correctement. Donc on trouvé une solution qui consiste à rajouter un capteur de température qui indique que c'est encore le jour

du fait que dans la journée la température est toujours élevée par rapport au soir ou la nuit. En plus du capteur, le système est commandé par un Switch pour l'activer ou le désactiver.

Dans notre système le microcontrôleur ne communique pas seulement avec des capteurs optiques, mais aussi avec un capteur de température pour afficher la température en degré sur l'afficheur LCD

III.20. Organigramme de programme



III.21. Conclusion

Dans ce chapitre nous avons étudié en détail les étapes (Programmation, Simulation), après avoir écrit le programme et la vérification de son fonctionnement par le logiciel de simulation.

Conclusion Générale

CONCLUSION GENERALE

Dans notre étude nous avons traité une solution pour la commande du temps d'allumer ou d'éteindre l'éclairage. Nous avons utilisé une solution pour résoudre le problème de capteurs optique, souvent sa principale conséquence est la continuité de fonctionnement de l'éclairage dans la matinée, pour éviter ce problème nous avons proposé initialement de poser une LED de détection de pane.

Aussi nous avons essayé de faire une solution qui ne comptera pas seulement sur le capteur optique mais nous basons aussi sur le capteur de température en cas de nuage, un ombre ou de la poussière qui couvre les capteurs de lumière la indique que c'est encore le jour parce que dans la journée la température est toujours élevée par rapport au soir ou la nuit. En plus se capteur est commandé par un Switch pour l'activer ou le désactiver.

Ces solutions restent ouvertes et évolutives par des nouvelles idées pour une commande optimale dans l'éclairage public et pour la réduction du coût de la consommation d'énergie.

Références

Bibliographiques

Références bibliographiques

- [01] : <http://www.mirari.fr/eKKb> Prédiagnostic énergétique de l'Éclairage public de Barbentane Rapport «de visites et recommandations Réalisé par Olivier PLATON»
- [02] : Mémoire de fin d'étude «Commande de l'éclairage public et mesure de la température à base de PIC18F4550 » Smail Soufiane Université Mohamed Khider Biskra
- [03] : <http://www.technologuepro.com/microcontroleur-2/chapitre-2-microcontroleur-pic-16f877.pdf>
Iset Nabeul Y. RKHISSI KAMMOUN
- [04] : Mémoire de fin d'étude « éclairage public et mesure de température a base de pic 16f877»
Taboui Mouna 2009/2010 université de Tunis
- [05] : Mémoire de fin d'étude «Etude et réalisation d'un régulateur de Température à base du Microcontrôleur PIC 16F877A »
Ahmed chekari Miloud khelil université IBN-KHLDOUN 2009/2010
- [06] : « microcontrôleur based applied digital control » Département of computer engineering near est université Cyprus Dogane Ibrahim.
- [07] : Mémoire de fin d'études «Etude et réalisation d'un gradateur monophasé a base d'un microcontrôleur pic 16f877A » Benouali nabila université IBN-KHLDOUN 2012/2013.
- [08] : Mémoire de fin d'étude «Contrôle de température et d'humidité à base d'un microcontrôleur» Zekri Imane Tahar Mimouna université IBN-KHLDOUN 2015/2016
- [09] : Mémoire de fin d'étude « Programmation en mikroC.
Application pour les microcontrôleurs de la famille PIC»
UNIVERSITE M'HAMED BOGARA DE BOUMERDES
- [10] : (fr.scribd.com/doc/36434413/Programmation-en-C-du-Microcontrolleur-PIC16F877)
"Programmation en C du Microcontrôleur PIC16F877"
- [11] : Support de cours et TP : Conception et simulation des systèmes à pic
- [12] : <https://fr.wikipedia.org/wiki/Photor%C3%A9sistance>, "Photorésistance",
- [13] : Temperature Control System using LM35. Cytron Technologies Sdn Bhd
- [14] : Mémoire de fin d'étude «Mise en œuvre du protocole I2C dans environnement à microcontrôleur Atmel (ATMEGA 32) » Chennoufi Hicham UNIVERSITE Mohamed Khider Biskra juin 2013

Résumé

L'éclairage nocturne des lieux publics, des voies de circulation, des monuments et sites des villes est devenu aujourd'hui une préoccupation majeure des distributeurs d'énergie électrique et des autorités administratives et politiques, représente la majeure partie de la consommation d'énergie des villes et de communes à cause de l'utilisation des systèmes obsolètes. Notre but et de réduire les couts liés à la consommation d'énergie, nous utilisons dans ce projet un système de commande a un prix réduit, dépend un microcontrôleur PIC16F877A qu'il a géré et contrôlé l'éclairage avec l'utilisation des capteurs (de température et de lumière).

ملخص

الإضاءة الليلية للأماكن العمومية، وللطرق المزدهمة، وكذلك الآثار ومواقع المدينة أصبحت اليوم أكبر هاجس لموزعي الطاقة الكهربائية و السلطات الإدارية والسياسية، حيث تمثل الإنارة العمومية الجزء الأكبر من استهلاك الطاقة الكهربائية للمدن والبلديات بسبب استعمال أنظمة قديمة. هدفنا خفض القيمة المالية لاستهلاك الطاقة، استخدمنا في هذا المشروع نظام تحكّم بسعر زهيد حيث يعتمد على المعالج الدقيق PIC16F877A الذي يتحكم ويدير الإنارة باستخدام أجهزة الاستشعار (درجة الحرارة والضوء).

Abstract

The night lightning in public places, traffic lanes, monuments and town's sites has become today a major concern of the electrical energy distributors, also the administrative and politic authorities, that represents the major part of the energy consummation in towns because of the use of the obsolete systems. Our main goal is to reduce the cost linked to the consummation of the energy, in this project we use a system of commands with a reduced price, depending to a microcontroller PIC16F877A that has managed and controlled the lightning with sensors (temperature and light).