

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



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

Réf:.....

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

**MASTER**

*Thème*

Commande de l'éclairage public et  
mesure de la température à base de  
PIC18F4550

Présenté par :  
**Smail Soufiane**  
Soutenu le : Juin 2013

Devant le jury composé de :

Mr A. Benakcha  
Mme O.Terghini  
Mr A. Hezabra

MCA  
MAA  
MAA

Président  
Encadreur  
Examineur

**Année universitaire : 2012 / 2013**

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



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

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

**MASTER**

*Thème*

Commande de l'éclairage public et mesure de la température  
à base de PIC18F4550

**Présenté par :**

*Smail Soufiane*

**Avis favorable de l'encadreur :**

*Terghini Ouarda*

*signature*

**Avis favorable du Président du Jury**

*Benakcha Abdelhamid*

*Signature*

**Cachet et signature**

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



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

## *Thème :*

Commande de l'éclairage public et mesure de la température  
à base de PIC18F4550

**Proposé par :** Terghini Ouarda

**Dirigé par :** Terghini Ouarda

### **Résumé**

L'éclairage public représente chaque année la majeure partie de la consommation d'énergie, à cause de l'utilisation de systèmes de commande obsolètes. L'objectif de ce travail est de trouver une solution pour le contrôle optimal de l'éclairage public et de réduire la consommation d'énergie. Pour cette raison, on a réalisé un système de commande qui présente trois formes du processus de commande de l'éclairage. Notre système a également offre quelques informations supplémentaires telles que la température, la date et l'heure.

La conception et la réalisation de ce système dépend de microcontrôleur PIC18F4550 afin que ce dernier va gérer tous les processus dans le circuit et contrôle l'éclairage.

### **ملخص**

تمثل الانارة العمومية سنويا الجزء الاكبر من استهلاك الطاقة، يكون السبب الشائع لهذا الاستهلاك هو استعمال أنظمة تحكم قديمة. الهدف من هذا العمل هو إيجاد حل آلي للتحكم الأمثل في الانارة العمومية والتقليل من استهلاك الطاقة. ومن أجل ذلك قمنا بإنجاز نظام تحكم يقدم ثلاث اشكال لعملية التحكم. كما يقوم نظامنا بعرض بعض المعلومات الإضافية مثل درجة الحرارة والتاريخ والوقت.

تصميم وتنفيذ هذا النظام يعتمد على المعالج الدقيق PIC18F4550 بحيث هذا الأخير سيقوم بإدارة جميع العمليات في الدارة، والسيطرة على أوقات تشغيل وإيقاف الانارة.

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

*Au nom de dieu clément et miséricordieux*

# *Dédicace*

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

*Je dédie ce modeste travail*

*A mes très chers parents, que dieu les garde et les protège pour leurs soutien moral et  
financier, pour leurs encouragements et les sacrifices qu'ils ont endurés.*

*A mes frères et mes sœurs.*

*A toute ma famille.*

*A mes amis et mes collègues de la promotion 2013*

*« Electronique ».*

*A tous les musulmans dans le monde entier*

# Remerciements

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

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

*Nous remercions. Notre encadreur Madame Ferghini Cuarda pour les efforts qu'elle a déployé, pour nous aider, conseiller, encourager et corriger.*

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

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

*Sans oublier tous nos amis.*

## Résumé

L'éclairage public représente chaque année la majeure partie de la consommation d'énergie, à cause de l'utilisation de systèmes de commande obsolètes. L'objectif de ce travail est de trouver une solution pour le contrôle optimal de l'éclairage public et de réduire la consommation d'énergie. Pour cette raison, on a réalisé un système de commande qui présente trois formes du processus de commande de l'éclairage. Notre système a également offre quelques informations supplémentaires telles que la température, la date et l'heure.

La conception et la réalisation de ce système dépend de microcontrôleur PIC18F4550 afin que ce dernier va gérer tous les processus dans le circuit et contrôle l'éclairage.

## ملخص

تمثل الانارة العمومية سنويا الجزء الاكبر من استهلاك الطاقة، يكون السبب الشائع لهذا الاستهلاك هو استعمال أنظمة تحكم قديمة. الهدف من هذا العمل هو إيجاد حل آلي للتحكم الأمثل في الانارة العمومية والتقليل من استهلاك الطاقة. ومن أجل ذلك قمنا بإنجاز نظام تحكم يقدم ثلاث اشكال لعملية التحكم. كما يقوم نظامنا بعرض بعض المعلومات الإضافية مثل درجة الحرارة والتاريخ والوقت.

تصميم وتنفيذ هذا النظام يعتمد على المعالج الدقيق PIC18F4550 بحيث هذا الأخير سيقوم بإدارة جميع العمليات في الدارة، والسيطرة على أوقات تشغيل وإيقاف الانارة.

## Liste des Tableaux

<b>Tableau. II.1</b> : les caractéristiques générales de PIC18F4550. ....	09
<b>Tableau. III.1</b> : Les bibliothèques du MikroC. ....	26
<b>Tableau. IV.1</b> : Organisation des registres de RTC. ....	55
<b>Tableau. IV.2</b> : le nombre d'heures de fonctionnement de système. ....	60
<b>Tableau. IV.3</b> : Exemple de consommation d'énergie d'un poste. ....	60

## Liste des figures

<b>Fig. I.1</b> Le lumandar. ....	03
<b>Fig. I.2</b> : L'horloge classique. ....	04
<b>Fig. I.3</b> : Calculateur astronomique. ....	04
<b>Fig. II.1</b> Exemple de l'identification PIC18F4550. ....	08
<b>Fig. II.2</b> Brochage du PIC18F4550. ....	10
<b>Fig. II.3</b> Oscillateur à Quartz/Céramique. ....	11
<b>Fig. II.4</b> Architecture interne du PIC18F4550. ....	14
<b>Fig. II.5</b> Synoptique général du CAN pour le PIC18F4550. ....	17
<b>Fig. II.6</b> Registre ADCON0. ....	18
<b>Fig. II.7</b> Registre ADCON1. ....	19
<b>Fig. II.8</b> Registre ADCON2. ....	20
<b>Fig. III.1</b> : Etapes de développement du programme. ....	21
<b>Fig. III.2</b> : Interface du logiciel MikroC. ....	22
<b>Fig. III.3</b> : Création d'un projet. ....	23
<b>Fig. III.4</b> : Les configurations de projet. ....	23
<b>Fig. III.5</b> : Fenêtre de saisie de programme. ....	24
<b>Fig. III.6</b> : Le chronogramme des lignes SDA et SCL. ....	27
<b>Fig. III.7</b> : Le chronogramme de communication I2C. ....	28
<b>Fig. III.8</b> : Interface du logiciel ISIS. ....	31
<b>Fig. III.9</b> : Le logiciel ISIS 7 en mode simulation du projet. ....	32
<b>Fig. III.10</b> : Interface du logiciel PICPgm. ....	33
<b>Fig. III.11</b> : Le schéma du Programmeur de PIC. ....	34
<b>Fig. IV.1</b> : Les coordonnées géographiques (longitude et latitude). ....	43
<b>Fig. IV.2</b> : Carte du monde des fuseaux horaires. ....	43
<b>Fig. IV.3</b> : Les différents blocs de système. ....	46
<b>Fig. IV.4</b> : Schéma de l'alimentation classique. ....	46
<b>Fig. IV.5</b> : Symbole et Capteur LDR. ....	47
<b>Fig. IV.6</b> : schéma de l'interface des capteurs de lumière. ....	48
<b>Fig. IV.7</b> : schéma et brochage du LM35. ....	49
<b>Fig. IV.8</b> : schéma de l'interface de puissance. ....	50
<b>Fig. IV.9</b> : Types d'afficheurs LCD. ....	51
<b>Fig. IV.10</b> : Brochage et schéma d'interface de LCD Nokia 3310. ....	52

<b>Fig. IV.11</b> : Schéma du clavier. ....	54
<b>Fig. IV.12</b> : Brochage du DS1307. ....	55
<b>Fig. IV.13</b> : Schéma du RTC. ....	56
<b>Fig. IV.14</b> : Schéma de l'unité de contrôle. ....	57
<b>Fig. IV.15</b> : Conceptions et fonctionnement de système. ....	58

## Liste des abréviations

GSM: Global System for Mobile Communications

PIC : Programmable Interface Controller.

RISC : Reduced Instructions Set Computer

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

RTC: Real time clock ou horloge temps réel

I2C: Inter Integrated Circuit

VDD: Voltage Drain Drain

VSS: Voltage Source Source

GND: The Ground ou la mass

MCLR: Master Clear

Ko : kilo-octets

ADC: Analog to Digital Converter

CAN : convertisseur analogique / numérique

SPI : Serial Peripheral Interface Bus

SDA : Serial Data Line ou ligne de données bidirectionnelle

SCL : Serial Clock Line ou ligne d'horloge de synchronisation bidirectionnelle

V<sub>reff</sub> : tension de référence

LDR : light-dependent resistor ou cellule photoconductrice

LCD : Liquid Crystal Display ou afficheur à cristaux liquide

**SOMMAIRE**

<b>INTRODUCTION GENERALE</b> .....	1
<b>Chapitre I : Les différents types de commande</b>	
I. 1 Introduction. ....	2
I. 2 Les dispositifs de commande. ....	2
I. 2.1 Allumage manuel. ....	2
I. 2.2 Cellule photosensible avec ou sans réglage de sensibilité. ....	2
I. 2.3 Horloge simple. ....	3
I. 2.4 Calculateur astronomique. ....	4
I. 2.5 Les commandes centralisées. ....	5
I. 2.6 Commande par voie hertzienne. ....	5
I. 3 Les modes de fonctionnement. ....	5
I. 3.1 Fonctionnement permanent. ....	5
I. 3.2 Fonctionnement semi-permanent (coupure nocturne). ....	5
I. 3.3 Fonctionnement par régulation / réduction de puissance (régulation de tension). ....	6
I. 4 Conclusion. ....	6
<b>Chapitre II : Présentation du PIC 18F4550</b>	
II. 1 Introduction. ....	7
II. 2 Définition d'un microcontrôleur. ....	7
II. 3 Définition d'un PIC. ....	7
II. 4 Les différentes familles de PIC. ....	7
II. 5 Identification des PIC. ....	8
II. 6 Le choix d'un PIC. ....	8
II. 7 Caractéristiques principales du pic 18F4550. ....	9
II. 8 Brochage du PIC18F4550. ....	10
II. 9 Les horloges système du PIC18F4550. ....	10
II. 10 Le Reset. ....	11
II. 11 Les Ports d'entrées / sorties. ....	12
II. 11.1 Généralités sur les ports. ....	12
II. 11.2 Les principaux registres liés aux ports. ....	12
II. 12 Architecture interne du PIC18F4550. ....	13
II. 12.1 Organisation de la mémoire. ....	15

II. 12.1.1 Mémoire de programme. ....	15
II. 12.1.2 Mémoire de données. ....	15
II. 12.2 Les timers. ....	16
II. 12.3 Unité arithmétique et logique UAL et registre de travail W. ....	16
II. 12.4 Les convertisseurs analogiques / numériques. ....	16
II. 12.4.1 Registre ADCON0. ....	17
II. 12.4.2 Registre ADCON1. ....	18
II. 12.4.3 Registre ADCON2. ....	19
II.14 Conclusion. ....	20
<b>Chapitre III : Programmation de PIC18f4550</b>	
III. 1 Introduction. ....	21
III. 2 Etapes de développement du programme. ....	21
III. 3 Présentation du MikroC. ....	22
III. 3. 1 Création d'un projet. ....	22
III. 3. 2 Quelques notions de programmation en C sous MikroC. ....	24
III. 3. 3 Les instructions de MikroC. ....	24
III. 3. 4 Les bibliothèques du MikroC. ....	26
III. 3. 5 Bus I2C. ....	27
III. 3. 5. 1 Présentation de Bus I2C. ....	27
III. 3. 5. 2 Principe de fonctionnement du bus I2C. ....	27
III. 3. 5. 3 Bus I2C et MikroC. ....	28
III. 3. 6 Le convertisseur analogique numérique et MikroC. ....	30
III. 4 Présentation du logiciel ISIS. ....	30
III. 4.1 Simulation par ISIS. ....	32
III. 5 Transfert du programme vers PIC. ....	33
III. 5.1 Comment utiliser PICPgm. ....	33
III. 5.2 Programmateur proposé. ....	33
III. 6 Organigramme de programme. ....	35
III. 7 Conclusion. ....	41
<b>Chapitre IV : Conceptions et réalisation de système</b>	
IV. 1 Introduction. ....	42
IV. 2 Etude théorique. ....	42

IV. 2.1 Les équations de l'algorithme utilisé dans les calculs. ....	42
IV. 2.1.1 Les coordonnées géographiques. ....	42
IV. 2.1.2 Les fuseaux horaires. ....	43
IV. 2.1.3 Crépuscule. ....	44
IV. 2.1.4 Les équations. ....	44
IV. 2 Les différents blocs de la carte. ....	45
IV. 2.1 L'Alimentation. ....	46
IV. 2.1.1 Principe du montage. ....	46
IV. 2.2 Le capteur de lumière (LDR). ....	47
IV. 2.2.1 Définition. ....	47
IV. 2.2.2 Principe de fonctionnement. ....	47
IV. 2.2.3 Principe du montage. ....	48
IV. 2.3 Le capteur de Température. ....	49
IV. 2.3.1 Le capteur LM35. ....	49
IV. 2.3.2 Brochage du LM35. ....	49
IV. 2.4 Interface commande puissance. ....	49
IV. 2.5 L'interface utilisateur. ....	50
IV. 2.5.1 L'afficheur LCD. ....	51
IV. 2.5.2 Le clavier (les boutons poussoirs et les LED). ....	53
IV. 2.6 Horloge. ....	54
IV. 2.6.1 Description du composant. ....	54
IV. 2.6.2 Brochage du DS1307. ....	55
IV. 2.6.3 Schéma du module RTC. ....	56
IV. 2.7 Unité de contrôle. ....	56
IV. 3 Fonctionnement de système. ....	58
IV. 4 Consommation d'énergie. ....	60
IV. 5 Conclusion. ....	60
<b>CONCLUSION GENERALE. ....</b>	<b>61</b>
<b>Bibliographie. ....</b>	<b>62</b>
<b>Annexe. ....</b>	<b>64</b>

## INTRODUCTION GENERALE

L'éclairage public est l'un des besoins les plus importants dans notre vie quotidienne, qui varie dans la forme, l'objectif et les utilisations, outre il est représenté une part conséquente de la consommation d'énergie. Avec la libéralisation du marché de l'électricité et l'augmentation importante des coûts de l'énergie facturée aux communes, l'éclairage public est devenu un enjeu majeur. Dans de nombreuses communes, l'énergie consommée par l'éclairage public représente la plus grande proportion de leur facture d'électricité annuelle. Généralement la raison principale est l'utilisation d'anciens systèmes de commande. La commande peut agir sur la durée d'allumage et la quantité de lumière nécessaire. Donc dans le cas d'un mauvais fonctionnement de ces systèmes, il aura une incidence directe sur la consommation d'énergie, ce dernier est constitué une charge financière qui peut grever de manière sensible le budget des communes.

Bien sûr, il y a beaucoup de facteurs qui influent sur la consommation d'énergie, mais les plus influents sont les systèmes de commande de l'éclairage public. Alors la question qui se pose « Quelles sont les solutions possibles pour la commande optimal de l'éclairage public ? ».

Dans ce travail l'intérêt majeur est attribuée à la recherche trouvée d'une solution automatisée pour la commande de l'éclairage public, cette solution sera représenté dans un système embarquée (*un système électronique et informatique autonome spécialisé dans une tâche bien précise*) est basé sur un microcontrôleur (PIC18F4550). En outre, il ya des fonctions supplémentaires comme mesure de température et affichage de certaines informations comme la date et l'heure.

Ce mémoire comporte quatre chapitres :

Le Premier chapitre est axé sur les Considérations Générales où nous avons donné les différents types de système utilisé dans la commande de l'éclairage public.

Le deuxième chapitre est axé sur le microcontrôleur PIC18F4550, que nous avons étudié toutes ses caractéristiques et ses description détaillées.

Le troisième chapitre présente les logiciels utilisés pour la programmation de notre microcontrôleur PIC18F4550 et la simulation.

La conception et la réalisation pratique de notre carte, fait l'objectif du quatrième chapitre Enfin, notre mémoire est clôturé par une conclusion générale.

**Chapitre I**  
Les différents types de commande

## I. 1 Introduction

L'éclairage public représente une part conséquente de la consommation d'énergie. Il ya beaucoup de façons de réduire la consommation d'énergie, tels que la qualité des lampes sont utilisées et les types utilisés de la commande d'éclairage ... Etc. Dans ce chapitre, nous allons identifier certains types d'appareils utilisés dans la commande de l'éclairage public.

## I. 2 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 [1].

Les différents types de commande généralement rencontrés sont les suivants :

### I. 2.1 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) [1].

### I. 2.2 Cellule photosensible avec ou sans réglage de sensibilité

C'est le système communément appelé « lumandar (Fig. I.1) ». 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 [1].



Fig. I.1 : Le lumandar

### I. 2.3 Horloge simple

Une horloge «classique» (Fig. I.2) 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 [1].



Fig. I.2 : L'horloge classique

### I. 2.4 Calculateur astronomique

Également appelé horloge astronomique (Fig. I.3). 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 [1], [2].



Fig. I.3 : Calculateur astronomique

### **I. 2.5 Les commandes centralisées**

Les commandes centralisées reprennent, bien entendu, les fonctions essentielles des organes de commande d'éclairage public. On peut utiliser un interrupteur crépusculaire couplé avec une horloge de précision ou une horloge astronomique radio pilotée. Ces dispositifs permettent, soit par système radio, soit par système de téléphonie de type GSM, de piloter une commande d'éclairage public équipée des modules de réception.

Les commandes centralisées ont pour objectif d'optimiser les durées de fonctionnement et surtout de synchroniser l'ensemble des points d'allumage d'une collectivité.

De plus, il peut permettre de moduler l'éclairage public dans certaines zones [3].

### **I. 2.6 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 [1].

### **I. 3 Les modes de fonctionnement**

Associés à ces commandes différents modes de fonctionnement sont possibles favorisant plus ou moins la maîtrise de l'énergie [1] :

- Fonctionnement permanent
- Fonctionnement semi-permanent
- Fonctionnement par régulation / réduction de puissance

#### **I. 3.1 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.

#### **I. 3.2 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 d'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.

### **I. 3.3 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.

### **I. 4 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 PIC 18F4550

## II. 1 Introduction

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 public. Leur polyvalence, et leur taille les rendent intéressants pour des modules de traitement de données numériques et aussi analogiques. Ils sont certes peu puissants comparés à des processeurs dédiés, mais ils les compensent par leur prix mais surtout leur taille : un microcontrôleur peut être comparé à une carte mère d'ordinateur.

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

## II. 2 Définition d'un microcontrôleur

Un microcontrôleur, est un composant électronique qui rassemble tous les éléments d'un "mini-ordinateur" et qui se présente sous la forme d'un circuit intégré. Un microcontrôleur permet de réaliser des systèmes et montages électroniques programmés. Cela veut dire que l'on pourra, avec le même montage, réaliser des fonctions très différentes qui dépendront du programme qui aura été programmé dans le microprocesseur [4].

## 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 composants dits RISC (Reduced Instructions Set Computer), ou encore composant à jeu d'instruction réduit. Le microcontrôleur se trouve, dans plusieurs appareils tels que : les téléphones portables, machines à laver, télévisions vidéos ... etc. [5].

## II. 4 Les différentes familles de PIC

Les PICs sont subdivisés en 3 grandes familles [5] :

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

## II. 5 Identification des PIC :

Un PIC est généralement identifié par une référence de la forme suivante : **xx(L)XXyy-zz**  
[6]

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

**L** : tolérance plus importante de la plage de tension

**XX** : type de mémoire programme

C : EPROM ou EEPROM

CR : PROM

F : Flash

**yy** : Identificateur.

**zz** : vitesse maximale du quartz de pilotage.

### Exemple :

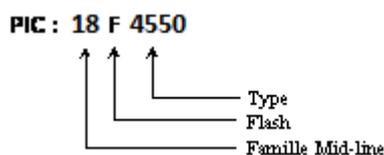


Fig. II.1 exemple de l'identification PIC18F4550

## II. 6 Le choix d'un PIC

Le choix d'un PIC est directement lié à l'application envisagée [5] :

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.

Afin de choisir un PIC adéquat à notre projet, nous avons pensé à l'utilisation du PIC PIC18F4550.

## II. 7 Caractéristiques principales du pic 18F4550

Les caractéristiques principales du 18F4550 sont résumées comme suit :

- 35 lignes d'entrées/sorties, répartis comme suit :
  - Un port de 7 lignes (port A)
  - Un port de 8 lignes (port B)
  - Un port de 8 lignes (port C)
  - Un port de 8 lignes (port D)
  - Un port de 4 lignes (port E)
- Alimentation sous 5 Volts
- Une mémoire de programme de type flash (32Ko (32768 Octets) mots de 16 bits)
- Une mémoire RAM utilisateur de 2048 Octets
- Une mémoire EEPROM de 256 Octets emplacements
- Une interface I2C pour la gestion d'un bus à 2 fils
- Facilité de programmation comme tous les PICs

Le tableau 2.1 présente tous les caractéristiques générales de PIC18F4550 [7]

La caractéristique	Valeur
Fréquence Horloge MHz	48MHz
Mémoire programme FLASH	32Ko (32.
Mémoire données (RAM)	2048 Octets
Mémoire EEPROM	256 Octets
Interruptions	20
Ports parallèles (Nombre des lignes)	A(7), B(8) C(8), D(8), E(4)
Timers	4 (3×16bit + 1×8bit)
CAN 10-bit	13
Instructions	75
Vitesse du CPU (MIPS)	12
Périphériques de communication numérique	1-A/E/USART, 1-MSSP (SPI/I2C)
Capture / Comparateur / Périphériques PWM	1 CCP, 1 ECCP
Comparateurs	2
USB (canaux, vitesse, respect)	1, Full Speed, USB 2.0
Plage de tension de fonctionnement (V)	de 2 à 5,5
Température (C)	-40 À 85

Tableau. II.1 : les caractéristiques générales de PIC18F4550

## II. 8 Brochage du PIC18F4550

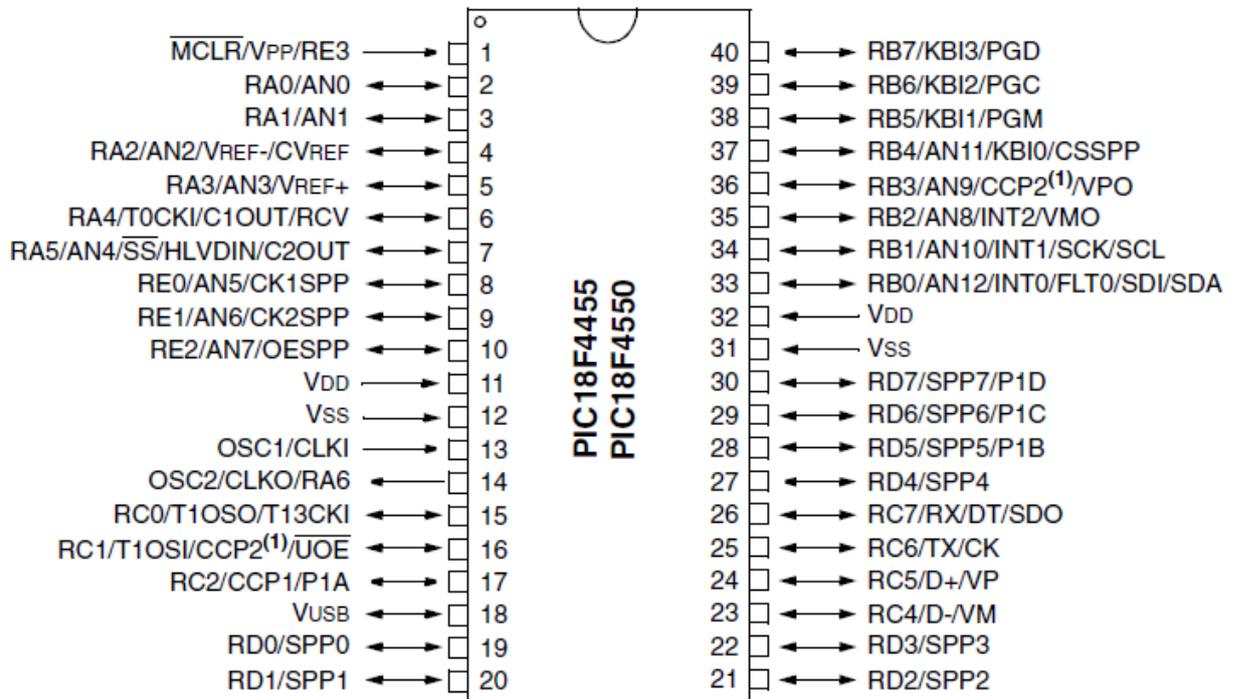


Fig. II.2 : Brochage du PIC18F4550 [7]

On peut distinguer sur ce schéma :

- L'alimentation : VDD (+5V) et VSS (0V)
- Les bornes du quartz (oscillateur a quartz) : OSC1 et OSC2
- L'entrée RESET (MCLR : Master CLear)
- Les différents ports d'Entrées/Sorties : RAX, RBx, RCx, RDx, Rex

## II. 9 Les horloges système du PIC18F4550

Comme la grande majorité des systèmes numériques à l'heure actuelle, les microcontrôleurs sont des composants synchrones, c'est-à-dire que toutes les opérations y sont synchronisées par une horloge commune. Dans le cas d'un microcontrôleur, le nombre d'instructions par seconde exécutées par le processeur est directement proportionnel à la fréquence de l'horloge. Ainsi il est logique de penser que l'idéal est d'utiliser l'horloge la plus élevée possible, car les performances seront ainsi maximisées. Ce raisonnement est juste, mais maladroit, car une autre chose qui est directement proportionnelle à la fréquence d'horloge est la consommation électrique du composant.

Dans le PIC18F4550, en plus d'une horloge système, qui cadencera le cœur du microcontrôleur, on peut aussi avoir à fournir une horloge à certains périphériques tels que le contrôleur USB ou les Timers.

Le PIC18F4550 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, HSPLL, XT et XTPLL, 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.

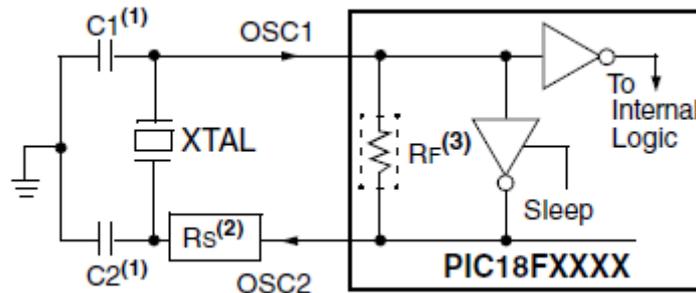


Fig. II.3 : Oscillateur à Quartz/Céramique [7]

- EC : Dans ce mode, il faut directement apporter à la broche OSC1 un signal d'horloge (carré) généré par un oscillateur totalement externe.
- INT : Le PIC18F4550 possède un oscillateur interne de fréquence réglable entre 31kHz et 8MHz.

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 [8]. Pour les régler on peut soit utiliser la directive par MikroC, voir chapitre (III).

## II. 10 Le Reset

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 18). Pour résumer, un Reset peut être déclenché par [8] :

- un appui sur le bouton Reset du montage (Broche MCLR = 0).
- une instruction logicielle Reset (PIC 18).
- une saturation ou un débordement inférieur de la pile (PIC 18).
- un dépassement de la temporisation du chien de garde (WDT).
- une atteinte du seuil mini de l'alimentation.

## II. 11 Les Ports d'entrées / sorties

### II. 11.1 Généralités sur les ports

La plupart des broches du PIC18F4550 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. Pour localiser les broches correspondant aux différents ports, il faut se reporter au brochage du composant dans la (Figure II.2). Sur la plupart des broches, le premier élément de la description est de la forme Rxy, avec x une lettre, correspondant au port auquel appartient la broche, et y le numéro de la broche dans le port. Par exemple la broche 6 du composant correspond à RA4, c'est-à-dire au bit 4 du port A.

Cependant sur les broches 1 et 14 la description sous forme de port n'est pas en premier dans la liste, mais en dernier. C'est parce que ces broches ont des affectations spéciales prioritaires sur les ports. La broche 1 est par défaut la broche de RESET (MCLR), on peut l'utiliser comme RE3 en inhibant le RESET externe grâce à un bit de configuration (comme pour configurer l'horloge). La broche 14 peut être utilisée comme RA6 dans les modes d'horloge qui n'utilisent pas un résonateur externe [8].

### II. 11.2 Les principaux registres liés aux ports

Il y a trois registres que l'on utilise pour commander les ports : les registres TRISx , PORTx et LATx (avec x la lettre correspondant au port) [8].

Les registres TRISx permettent de configurer la direction (entrée ou sortie) de chaque broche du port. Un 1 sur le bit y de ce registre placera Rxy en entrée, un 0 le placera en sortie. Pour le retenir c'est simple : 1 ressemble à la lettre « I » comme « Input », 0 ressemble à la lettre « O » comme « Output ».

Par exemple pour placer RB2 en sortie, on placera le bit 2 de TRISB à 0.

Les registres PORTx permettent de placer la valeur de sortie ou de lire la valeur en entrée du port x. Lorsque la broche y du port x est placée en sortie, placer à 1 le bit y de PORTx placera Rxy à 1, idem avec 0. Lorsque la broche y du port x est placée en entrée, le bit y de PORTx est l'image de la broche Rxy.

Le registre tampon de donnée en sortie appelé **LAT** pour le PIC.

Ces registres ont pour fonction d'éviter certains problèmes qui peuvent apparaître lorsque plusieurs opérations d'écriture/lecture des ports d'entrées/sorties sont effectuées successivement.

## II. 12 Architecture interne du PIC18F4550

La figure (Fig. II.4), présente l'architecture interne du circuit. Il est constitué des éléments suivants [7] :

- Quatre ports d'entrées/sorties
- Une unité arithmétique et logique (ALU)
- Quatre compteurs (Timers) Timer0, Timer1, Timer2, Timer3
- Un compteur de programme (program counter)
- Une mémoire RAM, 2048 octets
- Une mémoire EEPROM de 256 octets de données
- Un registre contenant le code de l'instruction à exécuter (IR)
- Convertisseur analogique numérique 13 canaux 10bit

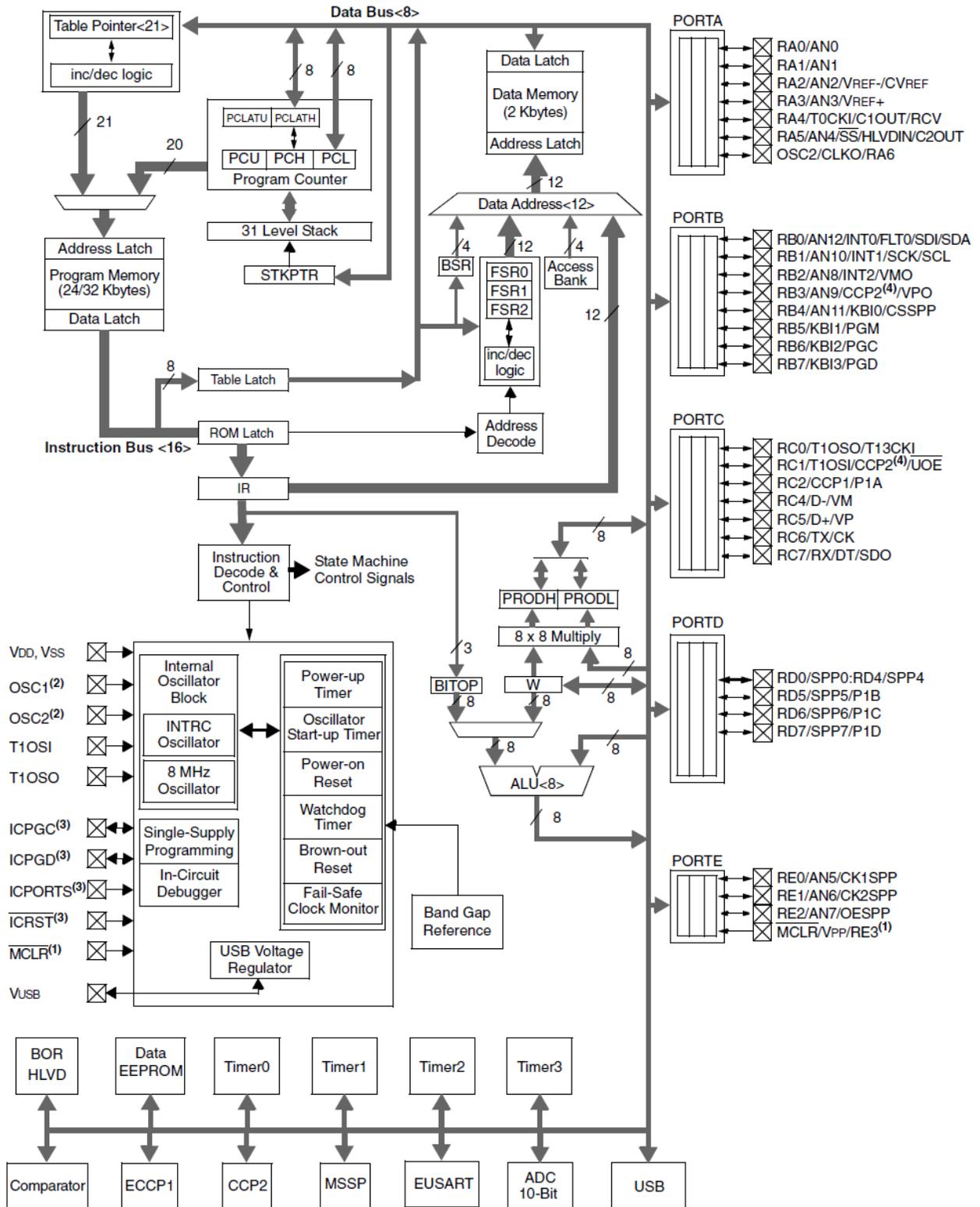


Fig. II.4 Architecture interne du PIC18F4550 [7]

## II. 12.1 Organisation de la mémoire

Le microcontrôleur PIC18F4550 dispose de deux types de mémoires séparées [9] :

- Une mémoire de programme de 32 ko (mémoire FLASH)
- Une mémoire de données de 2048 octets

### II. 12.1.1 Mémoire de programme

La capacité de la mémoire de programme est de 32 ko (kilo-octets). Cependant, chaque instruction est codée sur 16 bits, ce qui signifie que chaque instruction occupera 2 octets.

Ainsi, la mémoire de programme ne pourra stocker que jusqu'à 16384 instructions au maximum [9].

### II. 12.1.2 Mémoire de données

Elle se décompose en deux parties : une RAM et une zone EEPROM

- **La mémoire RAM**

La mémoire RAM (Random Access Memory) appelée mémoire vive, ce type de mémoire perd toutes ses informations stockées lorsqu'elle cesse d'être alimentée. La capacité totale de la mémoire de données est de 2048 octets. Elle est constituée de deux parties :

- Les registres SFR (Special Function Register)

Ce sont les registres de fonctionnement du PIC. L'ensemble de ces registres est souvent appelé fichier des registres.

- Les registres GPR (General Purpose Register)

Sont des positions mémoire que l'utilisateur peut utiliser pour stocker ses variables et ces données. On remarque donc que, indépendamment de leur nature, les positions de la RAM sont toujours appelées registres.

Le plan mémoire est découpé en 16 zones (bank) identiques de 256 octets. La sélection de la zone est réalisée par 4 bits du registre BSR (Bank Select Register).

Les registres internes pour les fonctions spéciales (registres généraux et configuration des périphériques → SFR qui signifie Special Function Registers) se situent aux adresses hautes [5], [9].

- **Mémoire EEPROM (Electrically Erasable Programmable Read Only Memory)**

Le PIC possède une zone EEPROM de 256 octets accessibles en lecture et en écriture par le programme. On peut y sauvegarder des valeurs, qui seront conservées même si l'alimentation est éteinte, et les récupérer lors de la mise sous tension. Leur accès est spécifique et requiert l'utilisation de registres dédiés. La lecture et l'écriture ne peut

s'exécuter que selon des séquences particulières [5].

## II. 12.2 Les timers

Le timer est un élément important dans un PIC. En effet, cette ressource permettra de réaliser de nombreuses fonctions, telle que la temporisation, la génération de signaux PWM (appelée également Modulation de Largeur d'Impulsion, MLI), ou encore les modes Compare ou Capture. Selon le type de PIC rencontré, on trouvera un certain nombre de timers qui seront pour certains associés à des ressources internes du PIC. Un timer est en fait un compteur incrémentant au rythme d'une horloge, et ceci en ayant la possibilité de pré positionner une valeur de départ. Certains timers pourront fonctionner en mode 8 bits, 16 bits ou 32 bits. Dans le PIC18F4550 il existe quatre timers (un timer de 8 bit et le reste de 16 bits) [10].

## II. 12.3 Unité arithmétique et logique UAL et registre de travail W

L'UAL (ou ALU en anglais) sera chargée d'effectuer toutes les opérations logiques et arithmétiques de base, C'est une ALU à 8 bits qui réalise les opérations arithmétique et logique entre l'accumulateur W et n'importe quel autre registre 'F' ou constante K. L'accumulateur W est un registre de travail 8 bits, il n'a pas d'adresse comme les autres SFR. Pour les instructions à deux opérands, c'est toujours lui qui contient un des deux opérands. Pour les instructions à un opérande, celui-ci peut être soit W soit n'importe quel registre F. Le résultat de l'opération peut être placé soit dans le registre de travail W soit dans le registre F [5].

## II. 12.4 Les convertisseurs analogiques / numériques

Pour utiliser un convertisseur analogique / numérique, il existe trois registres internes nommés ADCON0, ADCON1 et ADCON2, qui permettront de configurer et de sélectionner les différentes entrées analogiques (ADC pour *Analog to Digital Converter*) du PIC utilisé. Le paramétrage de ces trois registres conduit à choisir la façon dont se comporteront le ou les convertisseurs utilisés. Les entrées du PIC en relation avec le convertisseur analogique/ digital interne sont les broches AN0 à AN12. Les entrées analogiques (RA0 à RA5) transitent *via* le PORT A du PIC, les entrées AN5 à AN7 font partie du PORT E et les entrées AN8 à AN12 font partie du port B.

Comme vous le voyez sur le schéma général, les bits CHS0 à CHS3 du registre ADCON0 permettent de sélectionner un canal (AN0 à AN12) qui sera en relation avec le convertisseur. Les bits VCFG0 et VCFG1 du registre ADCON1 permettent quant à eux de définir les références de tension (AN2 ou Vss pour Vref- et AN3 ou Vdd pour Vref +). Les bits PCFG0 à PCFG3 du même registre permettent de configurer chaque broche ANO à AN12 soit en entrée analogique soit en entrée / sortie logique. Le troisième registre ADCON2 permettra de sélectionner une

horloge pour le convertisseur analogique (bits ADCS0 à ADCS2), ainsi qu'un temps d'acquisition avant conversion (temps nommé TAD : bits ACQT0 à ACQT2), allant de 0 TAD à 20 TAD (soit 0 top d'horloge à 20 tops d'horloge), puis indiquera grâce au bit ADFM le format de la valeur de la conversion en sortie.

Lorsqu'une conversion est lancée, un bit nommé *GO/DONE* du registre ADCON0 est forcée à «1». Dès que la conversion en cours est terminée, ce bit repasse à « 0 ». On réalise donc une boucle d'attente tant que la conversion n'est pas échue. Notez également que le bit ADON du même registre permet d'alimenter le convertisseur en 5 V : si ce bit est à « 0 », alors le convertisseur ne consomme aucun courant. Le résultat de la conversion sera stocké dans deux registres 8 bits nommés ADRESH et ADRESL. Le résultat étant sur 10 bits, il faudra choisir (ajustement à droite ou ajustement à gauche) les 10 bits parmi les 16 contenus dans ces deux registres, le bit ADFM du registre de configuration ADCON2 aura ce rôle (fig. II.5) [10].

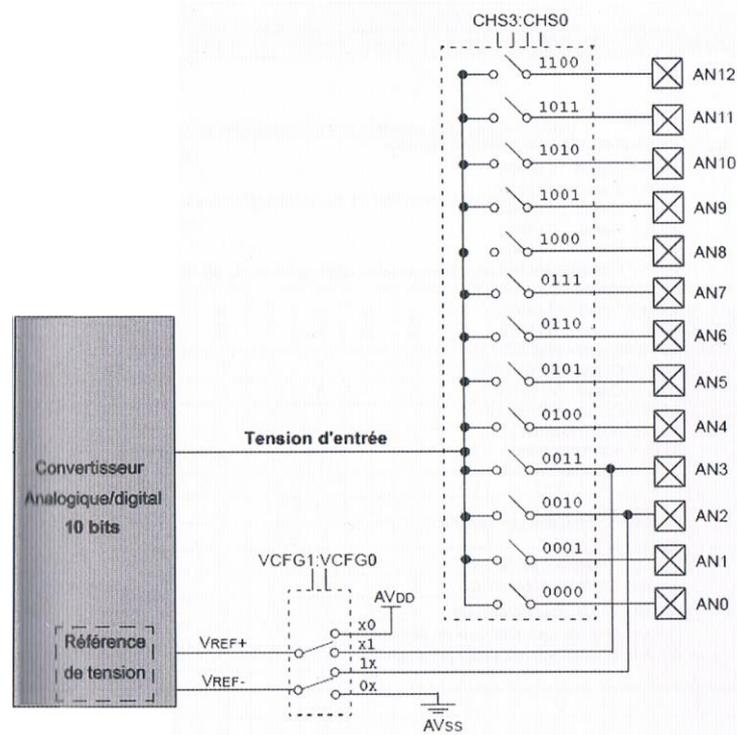


Fig. II.5 Synoptique général du CAN pour le PIC18F4550

### II. 12.4.1 Registre ADCON0

Ce registre permettra [10] :

- de sélectionner un canal parmi 12 (bits CHS0-CHS3) ;
- d'alimenter le module convertisseur (bit ADON) ;
- d'avertir l'utilisateur lorsqu'une conversion en est cours (bit *GO/DONE*) ou de lancer une conversion.

La figure (fig. II.6) résume ces différentes fonctionnalités. Les bits CHS0 à CHS2 permettent de sélectionner le canal à choisir pour la conversion (exemple : 0101 → canal 5, soit l'entrée AN5 du PIC).

Le bit 1 (GO / DONE) permet de lancer une conversion ou bien de savoir si une conversion est en cours d'exécution (0 = fin de conversion).

Le bit 0 ADON permet l'alimentation du convertisseur : si ce bit est à 0, le convertisseur n'est pas alimenté.

ADCON0							
-	-	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
bit 7							bit 0
bit 7-6 <b>inutilisé</b> : lire come '0'							
bit 5-2 <b>CHS3:CHS0</b> : bits de sélection de canal							
0000 = Canal 0 (AN0)							
0001 = Canal 1 (AN1)							
0010 = Canal 2 (AN2)							
0011 = Canal 3 (AN3)							
0100 = Canal 4 (AN4)							
0101 = Canal 5 (AN5)							
0110 = Canal 6 (AN6)							
0111 = Canal 7 (AN7)							
1000 = Canal 8 (AN8)							
1001 = Canal 9 (AN9)							
1010 = Canal 10 (AN10)							
1011 = Canal 11 (AN11)							
1100 = Canal 12 (AN12)							
1101 = non implémenté							
1110 = non implémenté							
1111 = non implémenté							
bit 1 <b>GO/DONE</b> : bit d'état conversion							
Quand ADON = 1:							
- 1 = A/D conversion en cours							
- 0 = A/D conversion terminée							
bit 0 <b>ADON</b> : bit de mise en service							
- 1 = A/D Module convertisseur en service							
- 0 = A/D Module convertisseur hors service							

Fig. II.6 Registre ADCON0

### II. 12.4.2 Registre ADCON1

Ce registre (fig. II.7) permettra [10] :

- de définir les tensions de référence V- et V+, ces tensions pourront être l'entrée AN2 ou AN3, ou bien le +Vdd et le Vss de l'alimentation (bits VCFG1 et VCFG0) ;
- de configurer toutes les broches du port A soit en entrée logique soit en entrée analogique avec des options pour la tension de référence (bits PCFG0-PCFG3).

ADCON1													
-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0						
bit 7						bit 0							
bit 7-6 <b>inutilisé</b> : lire come '0'													
bit 5 <b>VCFG1</b> : bits de configuration de la référence de tension Vref –													
- 1 = Vref – (AN2)													
- 0 = AVss													
bit 4 <b>VCFG0</b> : bit de configuration de la référence de tension Vref +													
- 1 = Vref + (AN3)													
- 0 = AVdd													
Bit 3-0 <b>PCFG3 :PCFG0</b> :bit de configuration des entrées du convertisseur													
bit 1 <b>GO/DONE</b> : bit d'état conversion													
Quand ADON = 1:													
- 1 = A/D conversion en cours													
- 0 = A/D conversion terminée													
bit 0 <b>ADON</b> : bit de mise en service													
- 1 = A/D Module convertisseur en service													
- 0 = A/D Module convertisseur hors service													
<b>PCFG3 : PCFG0</b>	<b>AN12</b>	<b>AN11</b>	<b>AN10</b>	<b>AN9</b>	<b>AN8</b>	<b>AN7</b>	<b>AN6</b>	<b>AN5</b>	<b>AN4</b>	<b>AN3</b>	<b>AN2</b>	<b>AN1</b>	<b>AN0</b>
0000	A	A	A	A	A	A	A	A	A	A	A	A	A
0001	A	A	A	A	A	A	A	A	A	A	A	A	A
0010	A	A	A	A	A	A	A	A	A	A	A	A	A
0011	D	A	A	A	A	A	A	A	A	A	A	A	A
0100	D	D	A	A	A	A	A	A	A	A	A	A	A
0101	D	D	D	A	A	A	A	A	A	A	A	A	A
0110	D	D	D	D	A	A	A	A	A	A	A	A	A
0111	D	D	D	D	D	A	A	A	A	A	A	A	A
1000	D	D	D	D	D	D	A	A	A	A	A	A	A
1001	D	D	D	D	D	D	D	A	A	A	A	A	A
1010	D	D	D	D	D	D	D	D	A	A	A	A	A
1011	D	D	D	D	D	D	D	D	D	A	A	A	A
1100	D	D	D	D	D	D	D	D	D	D	A	A	A
1101	D	D	D	D	D	D	D	D	D	D	D	A	A
1110	D	D	D	D	D	D	D	D	D	D	D	D	A
1111	D	D	D	D	D	D	D	D	D	D	D	D	D

A = entrée Analogique D = entrée / sortie Digitale

Fig. II.7 Registre ADCON1

### II. 12.4.3 Registre ADCON2

Ce registre (fig. II.8) permettra [10] :

- de définir le format de donnée pour la valeur issue de la conversion analogique/numérique (bits *ADFM*) ;
- de configurer le choix de l'horloge utilisée par le convertisseur (bits *ADCS0* à *ADCS2*) ;
- de configurer un temps d'attente avant la conversion et d'obtenir ainsi une à éventuelle stabilisation de la tension à mesurer (bits *ACQT0* à *ACQT2*).

ADCON2							
ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
bit 7							bit 0
<p>bit 7 <b>ADFM</b>: bit de configuration du format de sortie            1 = justification à droite            0 = justification gauche</p> <p>bit 6 <b>Inutilisé</b></p> <p>bit 5-3 <b>ACQT2:ACQT0</b>: bits de sélection des temps d'acquisition            111 = 20 TAD            110 = 16 TAD            101 = 12 TAD            100 = 8 TAD            011 = 6 TAD            010 = 4 TAD            001 = 2 TAD            000 = 0 TAD</p> <p>bit 2-0 <b>ADCS2:ADCS0</b>: bits de sélection de l'horloge pour la conversion            111 = FRC (Horloge provenant de l'oscillateur RC interne)            110 = Fosc/64            101 = Fosc/16            100 = Fosc/4            011 = FRC (Horloge provenant de l'oscillateur RC interne)            010 = Fosc/32            001 = Fosc/8            000 = Fosc/2</p>							

Fig. II.8 Registre ADCON2

## II.14 Conclusion

Ce chapitre est un portail pour notre réalisation et conception matérielle. Ca nous a permis de maîtriser les options de notre PIC et ses caractéristiques afin de les exploiter d'une manière correcte.

**Chapitre III**  
Programmation de PIC18F4550

### III. 1 Introduction

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, flowcode...etc. Dans notre projet nous avons opté pour le compilateur de MIKRO C qui est un compilateur en langage C (langage évolué) et qui permet d'intégrer certaines routines en assembleur. Ce choix est à la fois un choix personnel et un choix technologique parce que il est basé sur le langage C et est facile et en douceur.

### III. 2 Etapes de développement du programme

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

La première étape : L'algorithme

La deuxième étape : Ecriture du programme

La troisième étape : Simulation du programme

La quatrième étape : Transfert du programme vers PIC

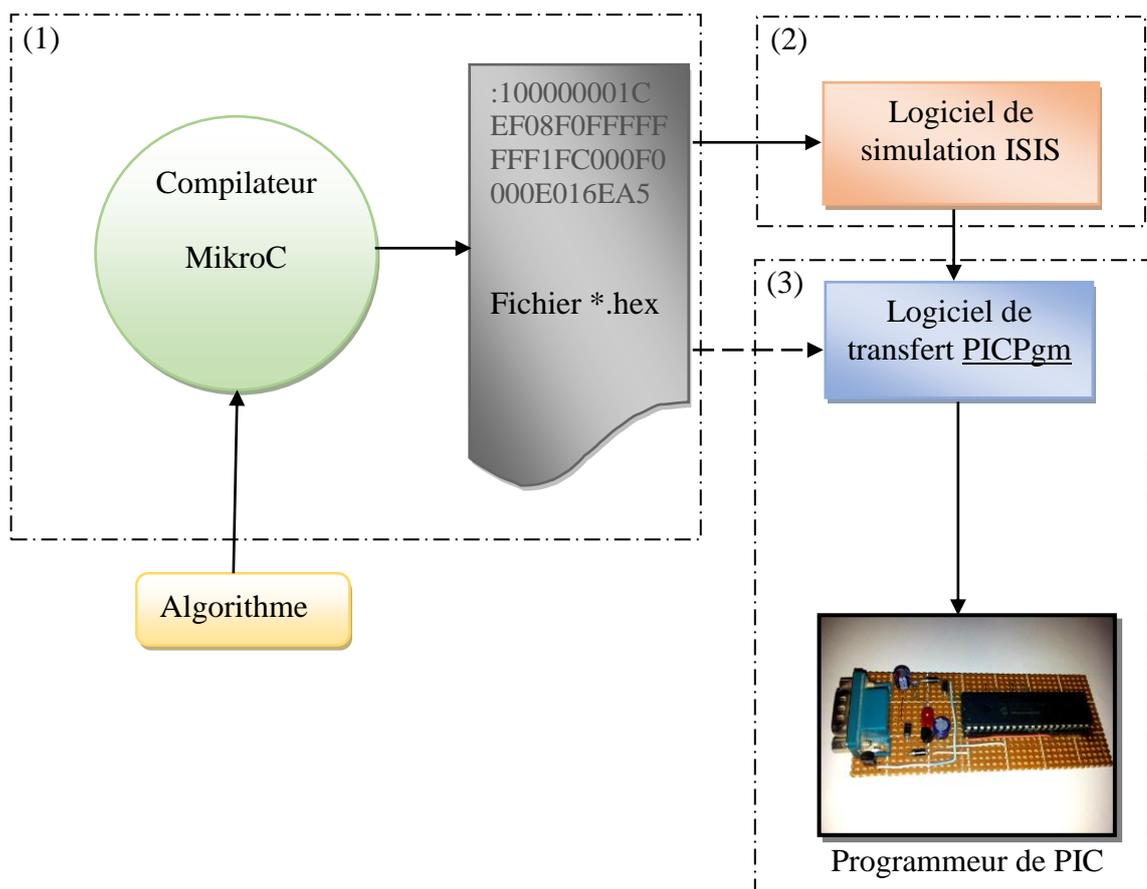


Fig. III.1 : Etapes de développement du programme

### III. 3 Présentation du MikroC :

Le « MikroC » est un compilateur pour PIC Conçu par la société « Mikroelektronika », le compilateur C nouvelle génération "MikroC" pour microcontrôleurs PIC bénéficie d'une prise en main très facile. Il comporte plusieurs outils intégrés (mode simulateur, terminal de communication, gestionnaire 7 segments, analyseur statistique, correcteur d'erreur, explorateur de code...) ; Il a une capacité à pouvoir gérer la plupart des périphériques rencontrés dans l'industrie (Bus I2C, 1Wire, SPI, RS485, Bus CAN, cartes compact Flash, signaux PWM, afficheurs LCD et 7 segments...) de ce fait il est un des outils de développement incontournable et puissant [9].

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 [11].

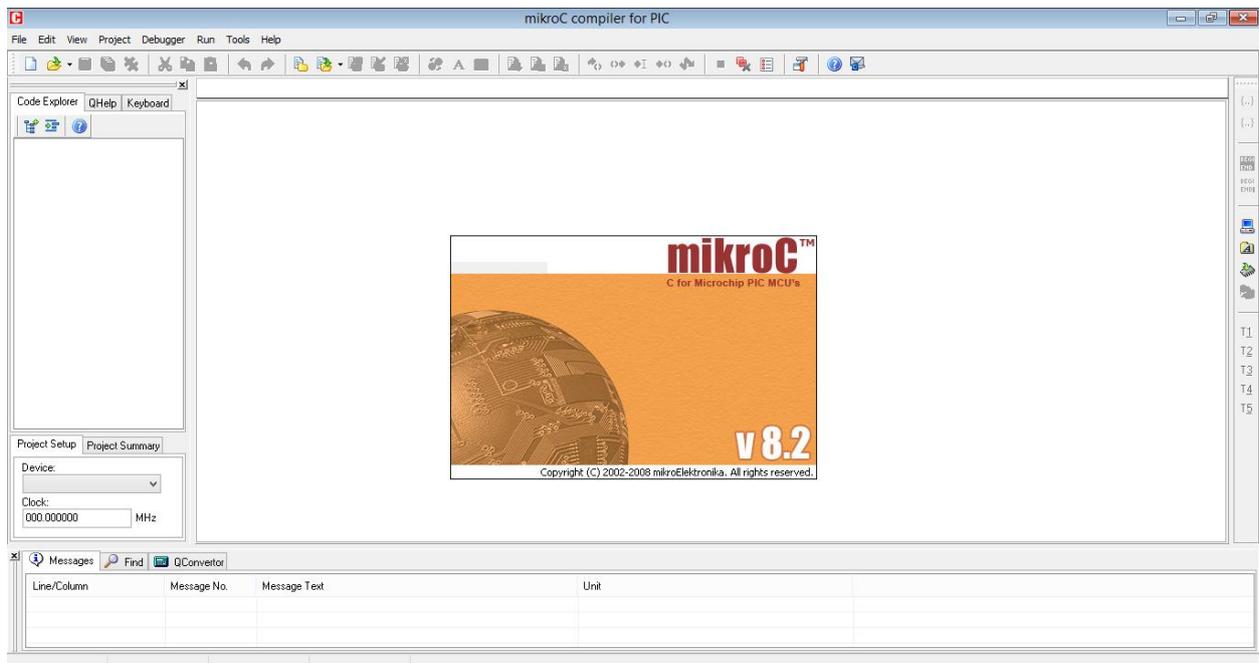


Fig. III.2 : Interface du logiciel MikroC

#### III. 3. 1 Création d'un projet

Le processus de création d'un nouveau projet est vraiment très simple. Sélectionnez New Project (Nouveau Projet) de puis le menu Project (Projet), comme indiqué sur (Fig. III.3).

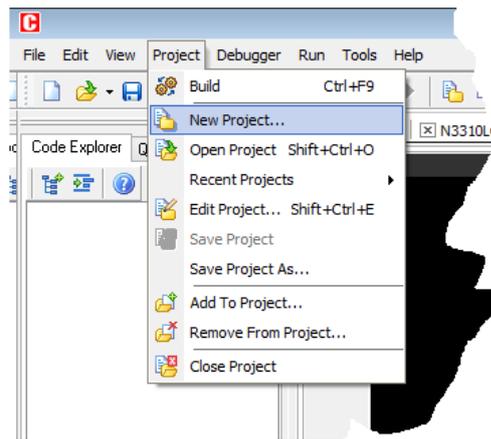
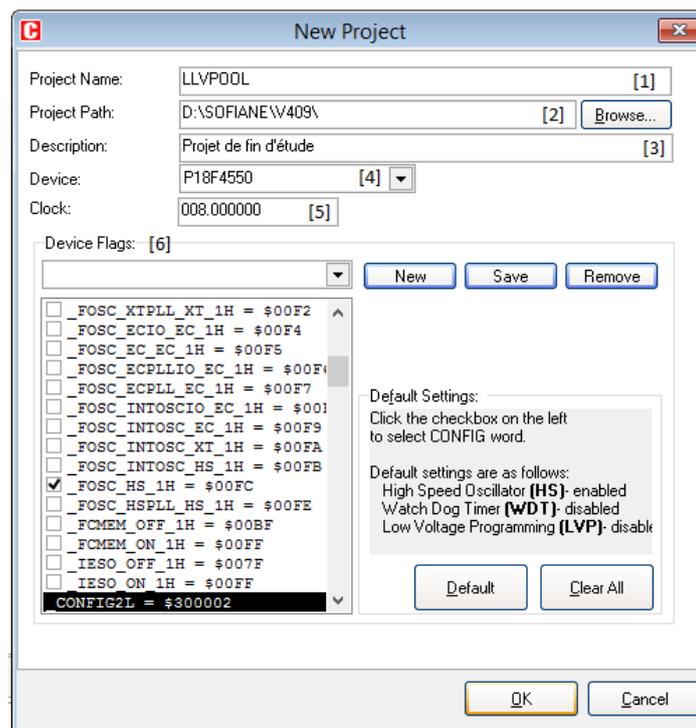


Fig. III.3 : Création d'un projet

Une nouvelle fenêtre apparaîtra. Comme indiqué sur (Fig. 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.



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

Fig. III.4 : Les configurations de projet

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

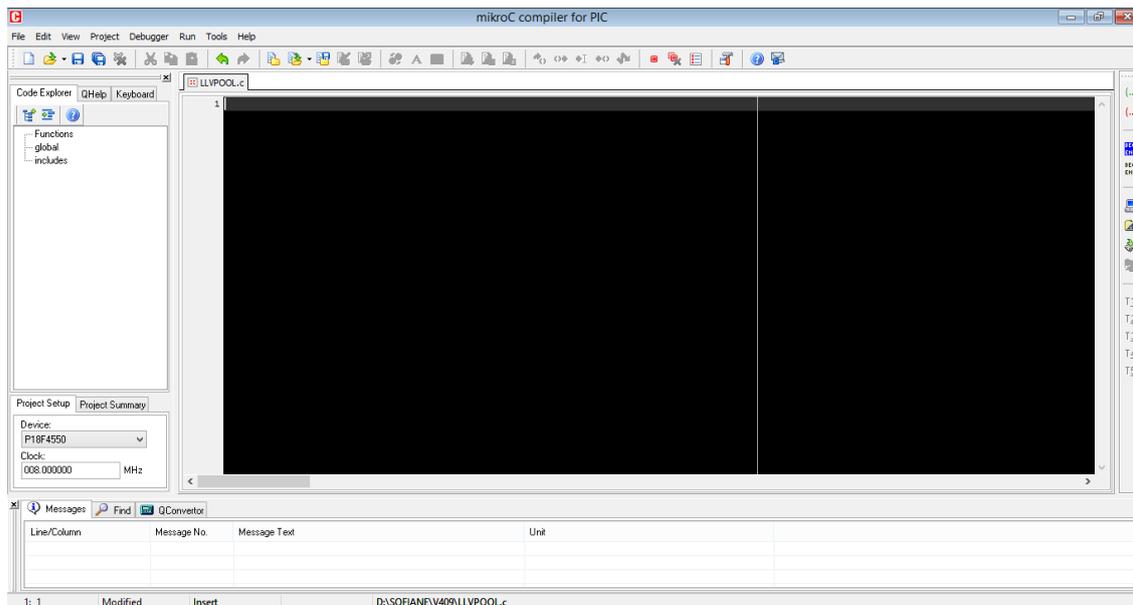


Fig. III.5 : Fenêtre de saisie de programme

### III. 3. 2 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 [12] :

- 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”.

### III. 3. 3 Les instructions de MikroC

Nous allons identifier uniquement les fonctions et les instructions les plus importantes que nous avons utilisées dans notre programme.

#### a. La boucle 'for' :

Elle permet de contrôler le nombre de fois à exécuter un bloc d'instructions. Dans la syntaxe de cette boucle on trouve : la valeur initiale du compteur, ça valeur finale est la valeur avec laquelle il doit être incrémenté (ou décrémenté) chaque fois que le bloc d'instructions est exécuté [10].

```
for (i=0 ;i=10 ;i++) //i est le compteur
{
portd = portd++; // incrémenté PORTD
}
```

Peut-on utiliser la boucle 'for' comme une boucle infinie comme suit :

```
for ( ;; )//boucle infinie
{

portd = portd++; // incrémenté PORTD

}
```

### b. Opérateur de condition 'if / else'

Cet opérateur permet d'exécuter une partie du programme sous certaines conditions. La syntaxe de cet opérateur est la suivante [10] :

```
if (condition)
{
séquences d'instructions à exécuter si la condition est vérifiée
}
else
{
séquences d'instructions à exécuter si la condition n'est pas vérifiée
}
```

### c. Instruction goto :

La directive goto permet de brancher directement à n'importe quel endroit de la fonction courante identifiée par une étiquette. Une étiquette est un identificateur suivi du signe ":".

Exemple :

```
Start1 :
for ( ;; )//boucle infinie
{

portd = portd++; // incrémenté PORTD
DELAY_ms(500); // attendre une demi-seconde
If (portc.f1==0) goto Start2 ;

}
Start2 :
for ( ;; )//boucle infinie
{
portd = portd--; // décrémenté PORTD
DELAY_ms(500); // attendre une demi-seconde
If (portc.f0==0) goto Start1 ;
}
```

### III. 3. 4 Les bibliothèques du MikroC

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

Tableau. III.1 : Les bibliothèques du MikroC

Bibliothèque	Les fonctions	La description
Math	<b>floor</b>	retourne la partie entière entier le plus proche par les valeurs inférieures
	<b>sin, cos, tan</b>	Ce sont les trois fameuses fonctions utilisées en trigonométrie. Le fonctionnement est le même, ces fonctions renvoient un double. Ces fonctions attendent une valeur en <b>radians</b> .
	<b>asin, acos, atan</b>	Ce sont les fonctions arc sinus, arc cosinus et arc tangente, d'autres fonctions de trigonométrie. Elles s'utilisent de la même manière et renvoient un double.
I2C	<b>I2C_Init</b>	Initialise I2C avec horloge souhaitée Exemple : I2C_Init(100000);
	<b>I2C_Start</b>	Détermine si I2C bus est libre et lancer un signal START. Exemple : I2C_Start();
	<b>I2C_Repeated_Start</b>	Répéter un signal START. Exemple : I2C_Repeated_Start();
	<b>I2C_Rd</b>	Lit un octet de l'esclave, et envoie pas signal d'acquittement ACK si le paramètre est 0, sinon il envoie reconnaître. Exemple : take = I2C_Rd(0);
	<b>I2C_Wr</b>	Envoie octet de données (données de paramétrage) par l'intermédiaire du bus I2C. Exemple : I2C_Write(0xA3);
	<b>I2C_Stop</b>	Lancer un signal STOP. Exemple : I2C_Stop();
EEPROM	<b>Eeprom_Read</b>	Lit les données de l'adresse spécifiée. Le paramètre Adresse est de type entier, ce qui signifie qu'il prend en charge MCU avec plus de 256 octets de EEPROM. Exemple : take = Eeprom_Read(0x3F);
	<b>Eeprom_Write</b>	Écrit des données à l'adresse spécifiée. Le paramètre Adresse est de type entier, ce qui signifie qu'il prend en charge MCU avec plus de 256 octets de EEPROM. Exemple : Eeprom_Write(0x32, 19);
ADC	<b>Adc_Read</b>	Lit la valeur analogique du canal spécifié tmp = Adc_Read(1); // lire la valeur analogique du canal 1
Conversions	<b>Bcd2Dec</b>	Convertir 8 bits BCD chiffre vers son équivalent a décimal. Convertir 8 bits décimal valeur vers son équivalent à BCD chiffre.
	<b>Dec2Bcd</b>	

### III. 3. 5 Bus I2C

#### III. 3. 5. 1 Présentation de Bus I2C

Le bus I2C a été inventé pour simplifier la communication entre des composants divers dont le sigle signifie Inter Integrated Circuit ce qui donne IIC et par contraction I2C. Le protocole est initialement proposé par Philips 1980 mais adopté de nos jours par de très nombreux fabricants. C'est un bus de communication de type série [13].

Chez certains constructeurs, ce bus est parfois nommé sous le nom de TWI (Two Wire Interface).

Ce bus n'utilise que 3 fils :

- un signal de données (SDA)
- un signal d'horloge (SCL)
- un signal de référence électrique (masse)

#### III. 3. 5. 2 Principe de fonctionnement du bus I2C

Pour les composants voulant communiquer en I2C il suffit de lier ces 3 fils. Le fonctionnement du bus I2C est assez simple malgré que son protocole puisse sembler complexe. La communication I2C repose sur l'échange entre un maître et des esclaves par adressage. C'est-à-dire que chaque composant dit esclave possède une adresse spécifique sur 7 bits. C'est le composant maître qui gère la communication. Il envoie les signaux de données et le signal d'horloge. Généralement le maître est un composant dit microcontrôleur. Le rôle du signal d'horloge SCL est de rythmer l'envoi des données. Chaque bit du signal de données SDA sera envoyé à chaque changement d'état du SCL (Fig. III.6).

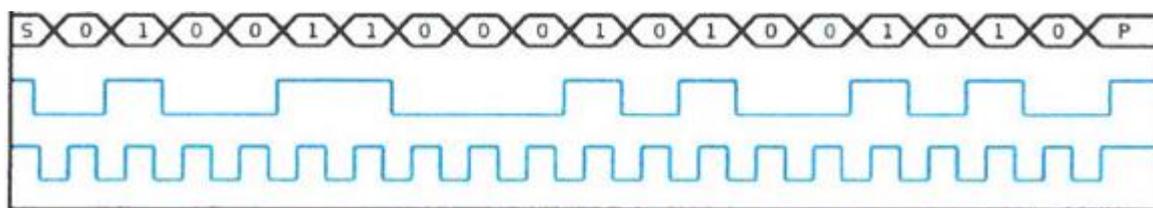


Fig. III.6 : le chronogramme des lignes SDA et SCL

Au début d'une communication, il faut s'assurer que le SDA et SCL sont en état de repos, soit un niveau logique 1 (Fig. III.7).

Ensuite le maître envoie sur le bus I2C, l'adresse de l'esclave avec qui il veut communiquer. On dit qu'il envoie les bits d'adressage. Suite à cela le maître indique le mode d'action (R/W) c'est-à-dire le maître indique s'il veut écrire ou lire les données de l'esclave (0 pour écriture, 1 pour la lecture).

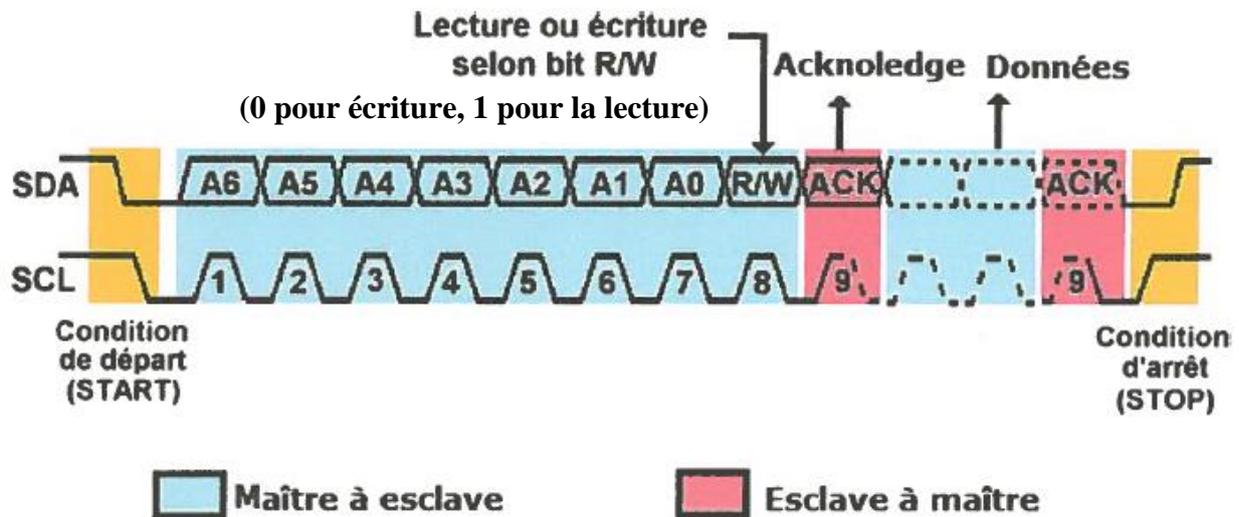


Fig. III.7 : le chronogramme de communication I2C

L'esclave possédant l'adresse indiquée répond par un « Acknowledge » (ACK). C'est un bit dit d'acquiescement pour indiquer qu'il a bien reçu les instructions du maître. Ce bit est symbolisé par un forçage de l'esclave à mettre la ligne SDA à un niveau logique 0.

L'acquiescement reçu par le maître, celui-ci envoie alors les données à l'esclave qui acquiescera à nouveau pour confirmer la réception. Le maître peut alors envoyer autant de données qu'il veut sur un même esclave qui acquiescera à chaque opération.

Pour finir, un maître peut communiquer avec plusieurs esclaves à condition de paramétrer les adresses de tous les esclaves. Il devra alors indiquer, à chaque changement d'esclave, l'adresse du nouveau esclave avec qui il veut communiquer [13].

### III. 3. 5. 3 Bus I2C et MikroC

MikroC fournit une bibliothèque prête du bus I2C (Tableaux 3.1), nous utilisons cette bibliothèque pour communiquer entre le maître (PIC18F4550) et l'esclave (DS1307). L'exemple suivant montre comment écrire des fonctions de lire et d'écrire en l'esclave DS1307 avec les commentaires.

#### a. Fonction d'écriture

```
void write_ds1307(unsigned short address,unsigned short w_data)
{
I2C_Start(); // lancer un signal de départ (le bit de start)
I2C_Wr(0xD0); // envoyer un octet via I2C (adresse de composant + un bit de
mode action (R/W))
```

```
// Adresse 0x68 est l'adresse de l'esclave DS1307 suivis par bit de direction
// (0 pour écriture, 1 pour la lecture), dans ce cas (mode écriture) l'adresse
// 0x68 suivi par 0 -> 0xD0
I2C_Wr(address); // envoyer un octet (adresse du register du DS1307)
I2C_Wr(w_data); // envoyer des données (Les données qui seront écrites)
I2C_Stop(); // envoyer un signal de stop ( bit de stop)
}
```

Un exemple de l'utilisation de cette fonction :

```
write_ds1307(1,Dec2Bcd(59)); // écrire les minutes 59
write_ds1307(2,Dec2Bcd(23)); // écrire l'heur 23 PM
```

## b. Fonction de lecture

```
void read_ds1307(unsigned short address)
{
I2C_Start(); // lancer un signal de départ (le bit de start)
I2C_Wr(0xD0); // envoyer un octet via I2C (adresse de composant + un bit de
mode action (R/W))
// Adresse 0x68 est l'adresse de l'esclave DS1307 suivis par bit de direction
// (0 pour écriture, 1 pour la lecture), dans ce cas (mode écriture) l'adresse
// 0x68 suivi par 0 -> 0xD0
I2C_Wr(address); // envoyer un octet (envoyer l'adresse du registre à lire)
I2C_Repeated_Start();//Ré-envoyer le bit de start
I2C_Wr(0xD1); // envoyer un octet via I2C (adresse de composant + un bit de
mode action (R/W))
// Adresse 0x68 est l'adresse de l'esclave DS1307 suivis par bit de direction
// (0 pour écriture, 1 pour la lecture), dans ce cas (mode lecture) l'adresse
// 0x68 suivi par 1 -> 0xD1
data=I2C_Rd(0); L'esclave envoie le contenu du registre d'adresse dans le
variable data
I2C_Stop(); // envoyer un signal de stop ( bit de stop)
return(data);// Retourne la valeur de la variable data à nom de fonction
}
```

Un exemple de l'utilisation de cette fonction :

```
minute=read_ds1307(1); // lire minute
hour=read_ds1307(2); // lire heur
dminute=Bcd2Dec(minute); dhours=Bcd2Dec(hour); // Conversions de BCD à décimal
```

### III. 3. 6 Le convertisseur analogique numérique et MikroC

Nous avons utilisés le convertisseur analogique numérique (CAN) intégré au PIC, ce CAN se trouve sur le port A et le port E, on doit l'initialiser en entrées analogiques et pour cela on utilise la fonction suivante :

```
val = Adc_Read(0) ; // lire la valeur analogique du canal 0
```

Le CAN de PIC18F4550 est de 10 bits donc les valeurs de val entre 0 et 1023 ( $0 \leq \text{val} \leq 1023$ ) pour que la valeur 0 représente tension 0 volt et la valeur 1023 représente la tension maximum d'entrée (tension de référence V<sub>reff+</sub>).

La tension analogique donnée par :  $V_{\text{analogique}} = \text{pas} \times \text{val}$

Le pas donnée par :  $\text{pas} = V_{\text{reff}} / 1023$

**Exemple :** Pour lire la tension analogique de sortie du capteur de température LM35

(l'explication de LM35 en chapitre 4) avec tension de référence (V<sub>reff+</sub> = 1068 mV) nous allons écrire les instructions suivantes :

```
int val, tempinC ;
ADCON1 = 0b00011001 ; // Configure les entrée analogique et Vref du port A
TRISA = 0xFF ; // PORTA comme entrée
val = ADC_Read(0) ; // lire la valeur analogique du canal 0
val = val*0.001068; // La tension analogique du capteur
tempinC = val/0.01 ; // 0.01 est facteur relatif de sensibilité du capteur
                    //0,01 volt dans tous les degrés Celsius
```

### III. 4 Présentation du logiciel ISIS de Proteus :

Isis est un éditeur de schémas qui intègre un simulateur analogique / logique ou mixte. Toutes les opérations se passent dans cet environnement, aussi bien la configuration des différentes sources que le placement des sondes et le tracé des courbes [14].

Dans cette section nous allons commencer par la présentation de la fenêtre du logiciel ISIS (Fig. III.8).

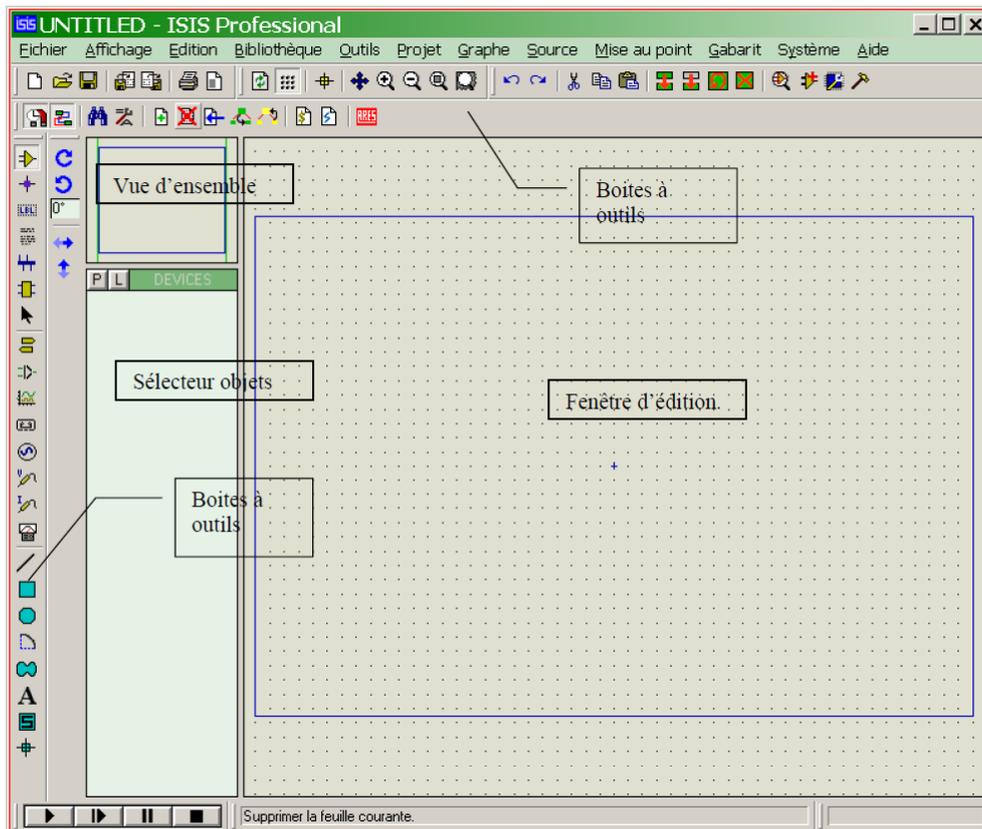


Fig. III.8 : Interface du logiciel ISIS

#### a-Fenêtre d'ensemble (Vue d'ensemble) :

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

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

- Vous pouvez déplacer cette zone de travail en pointant la souris sur la zone désirée de la fenêtre d'ensemble et en effectuant un clic gauche.
- Vous pouvez redéfinir la zone de travail dans la fenêtre d'ensemble en appuyant sur la touche majuscule 'shift' du clavier, associée au déplacement de la souris en maintenant appuyé le bouton gauche. [15]

#### b-Fenêtre d'édition :

La surface la plus grande de l'écran s'appelle "Fenêtre d'édition" et se comporte comme une fenêtre de dessin. C'est là que vous placez et câblez les composants [14].

#### c-La boîte à outils :

Elle est composée d'un ensemble d'icônes dont les fonctions seront détaillées ultérieurement et d'un sélecteur d'objet utilisé pour choisir les boîtiers, le style des pastilles, des traces, des traversées, etc. [15]

### III. 4.1 Simulation par ISIS

Pour le dessin du schéma électrique de notre projet on utilisée le logiciel ISIS pour le simuler comme indique la figure (Fig. III.9).

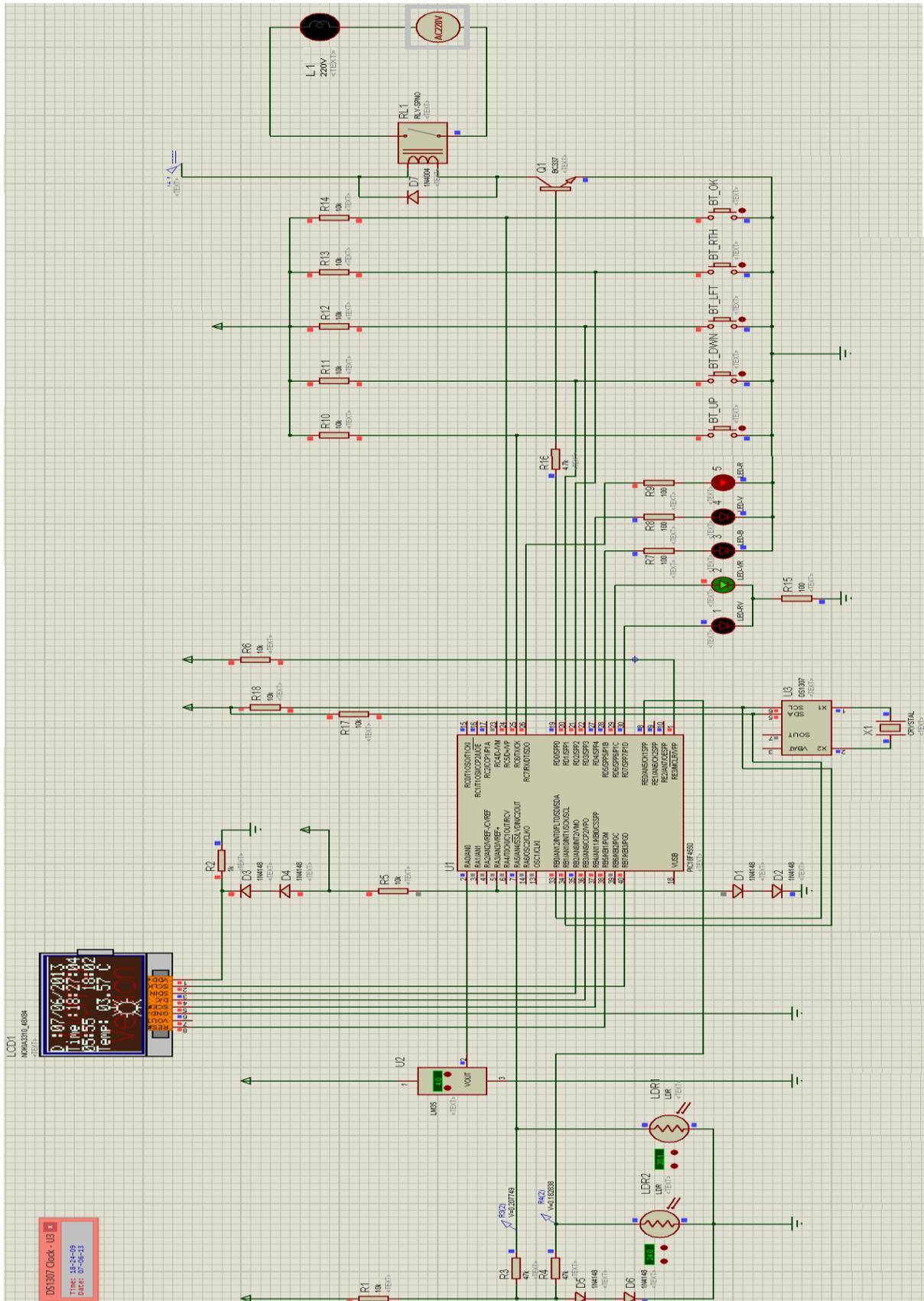


Fig. III.9 : Le logiciel ISIS 7 en mode simulation du projet

### III. 5 Transfert du programme vers PIC

Il existe de nombreux logiciels utilisés dans transfert du programme vers PIC comme IC-PROG, PICPgm, JDM for PIC etc. Dans notre cas nous allons utiliser PICPgm parce que il est plus simple et très facile, la figure suivante (Fig. III.10) représente l'interface du logiciel PICPgm.

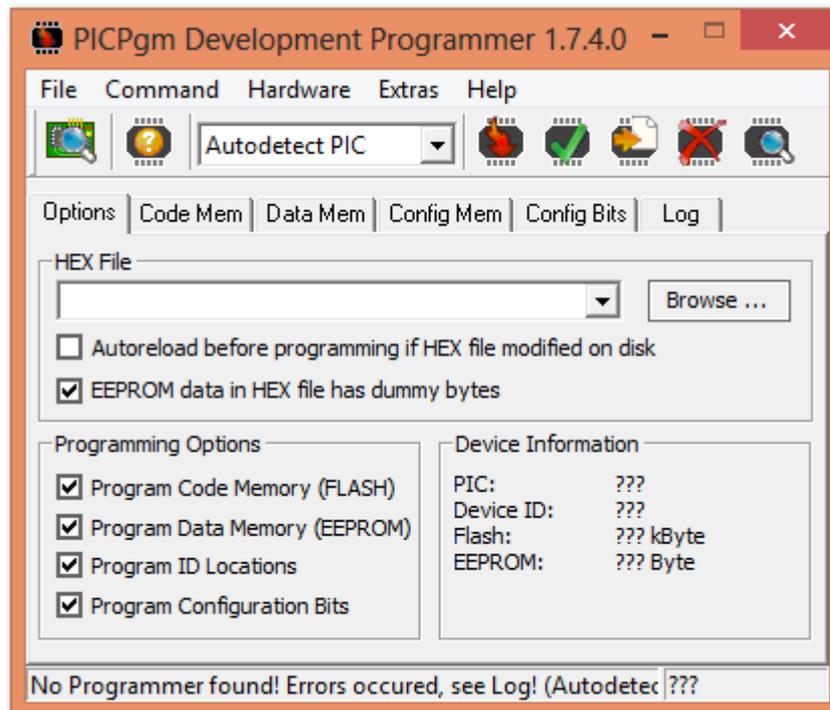


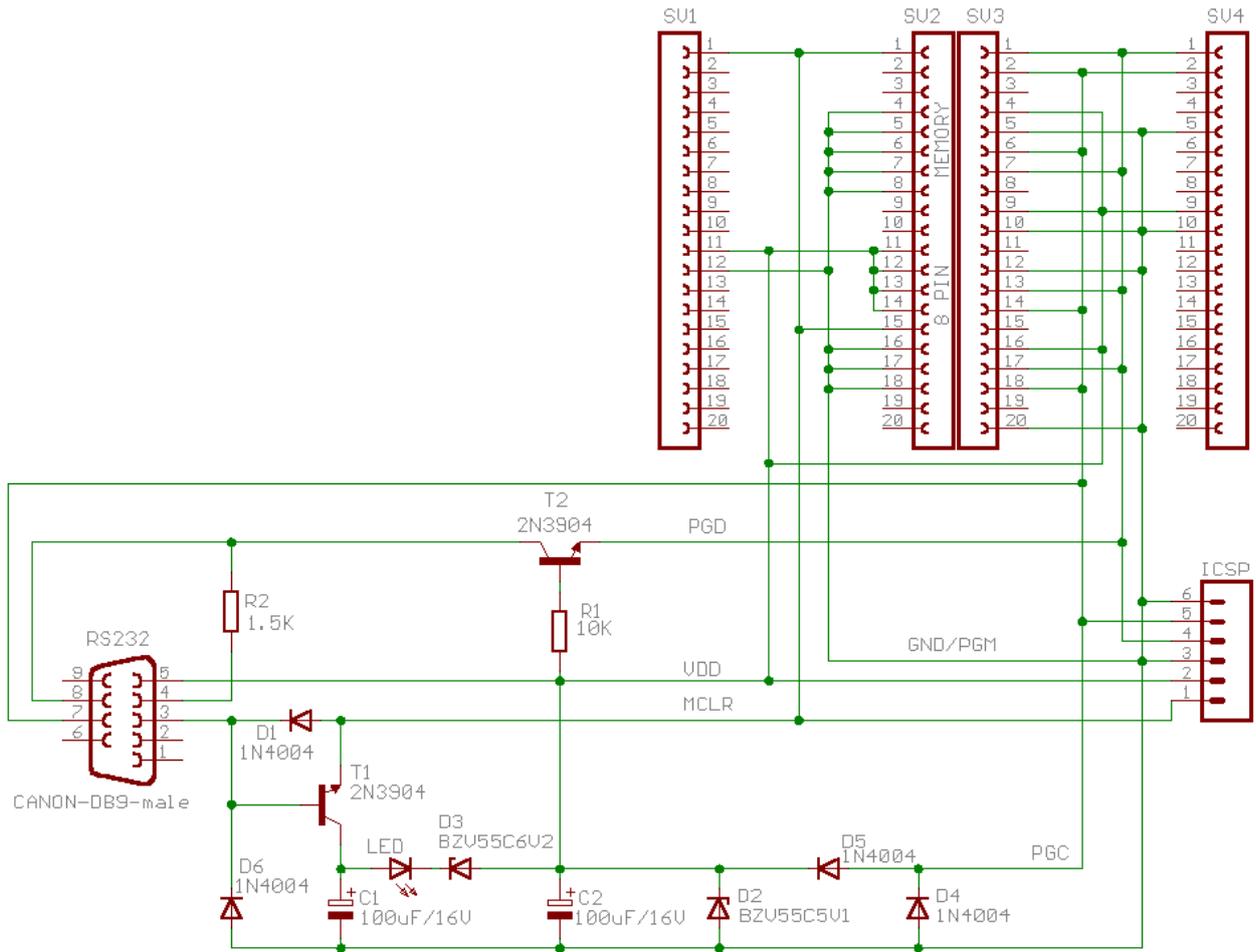
Fig. III.10 : Interface du logiciel PICPgm

#### III. 5.1 Comment utiliser PICPgm ?

L'utilisation du logiciel est plus simple, premièrement nous devons configurer les ports de l'ordinateur et le type de programmeur utilisé et connecter le programmeur par l'ordinateur. Deuxième étape est sélectionné le fichier de programme \*.hex par bouton « Browse » et la troisième étape c'est programmation du PIC par cliquer sur le bouton « Program PIC » .

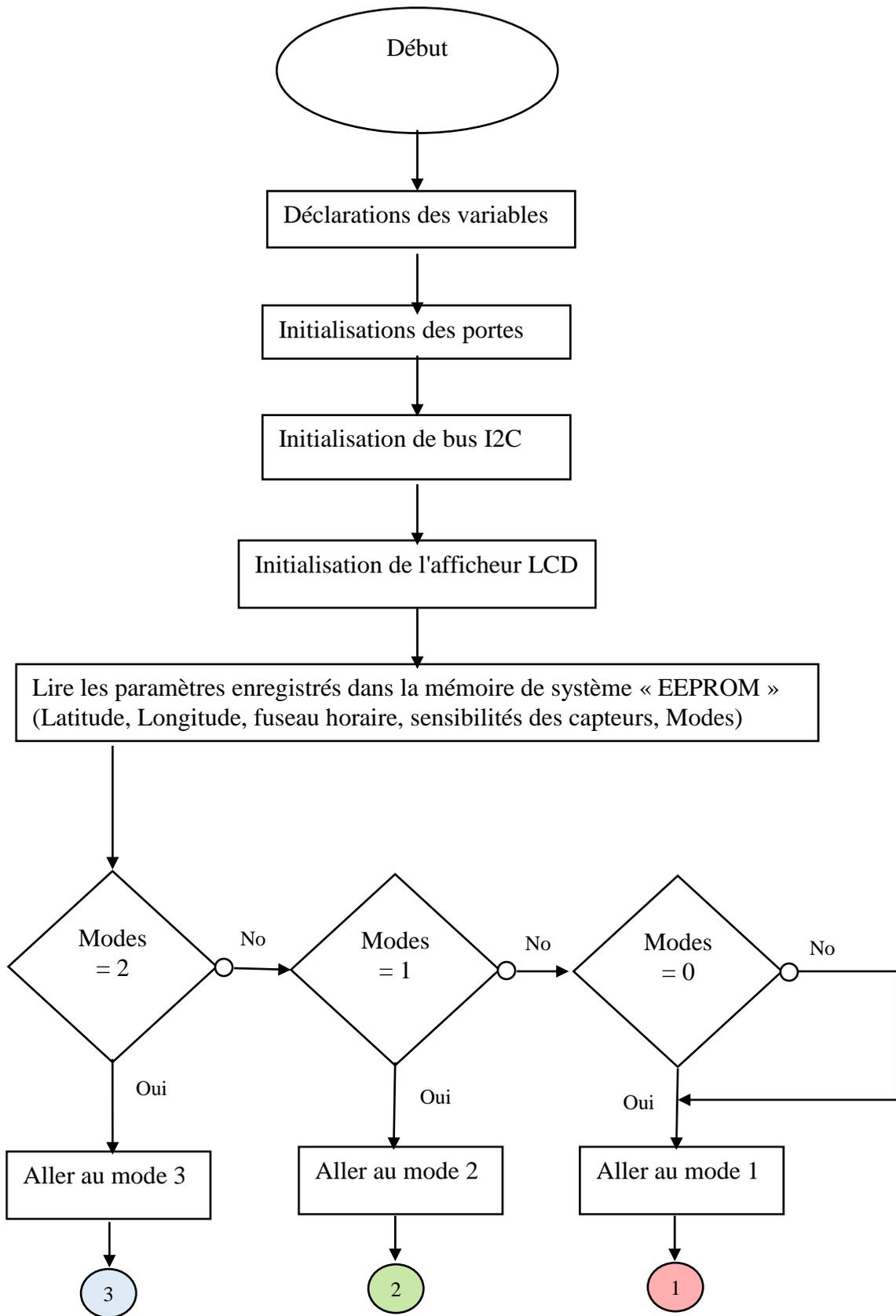
#### III. 5.2 Programmeur proposé :

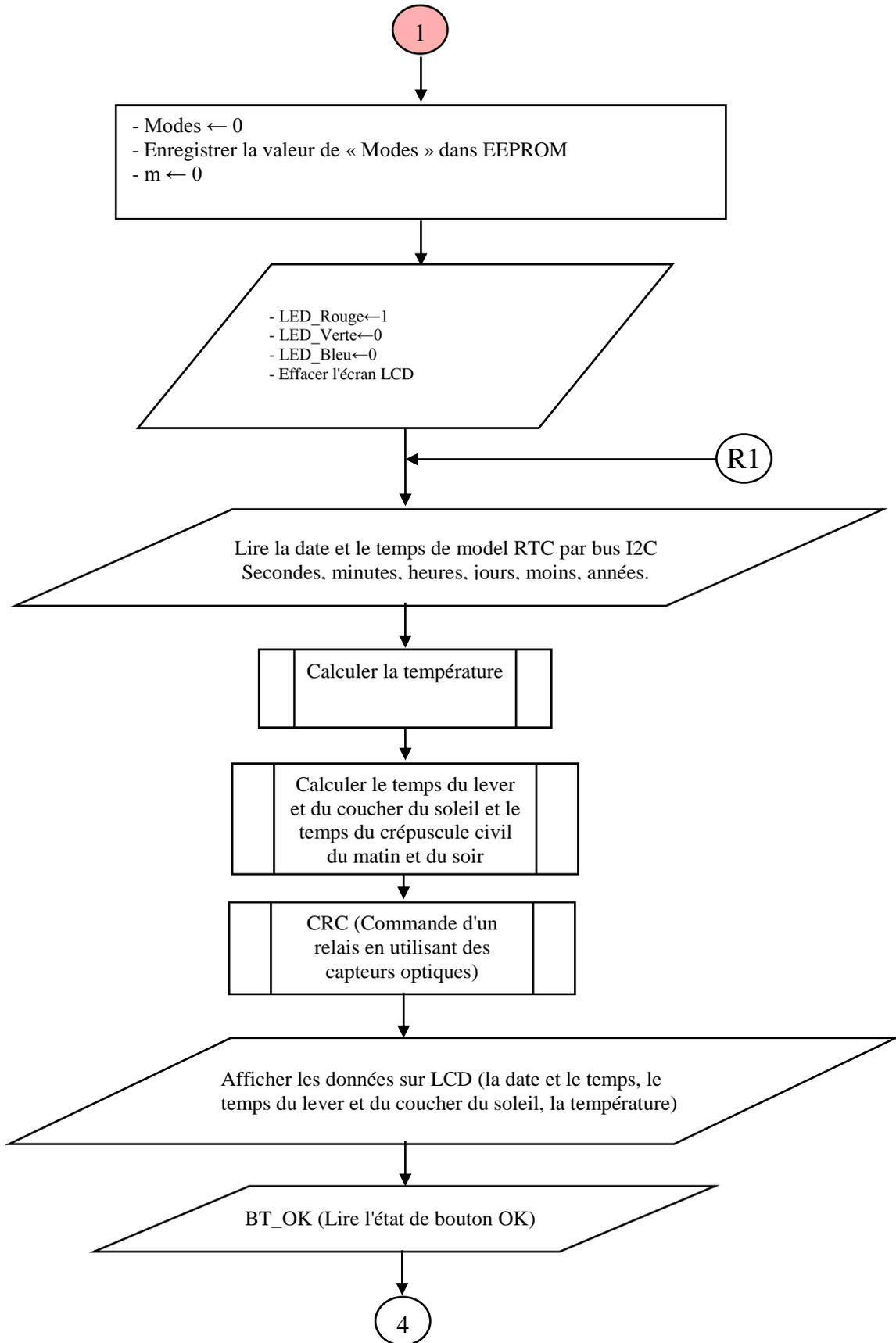
Il existe plusieurs programmeurs de PIC, nous construisons un programmeur JDM trié simple « à bas deux transistors, une résistance, six diodes et deux condensateurs » (Fig. III.11) qui peut programmer le PIC12, PIC16 et les microcontrôleurs PIC18 et certaines familles populaires EEPROM 24C. Le circuit est connecté au PC via un port série et aucune alimentation externe n'est nécessaire.

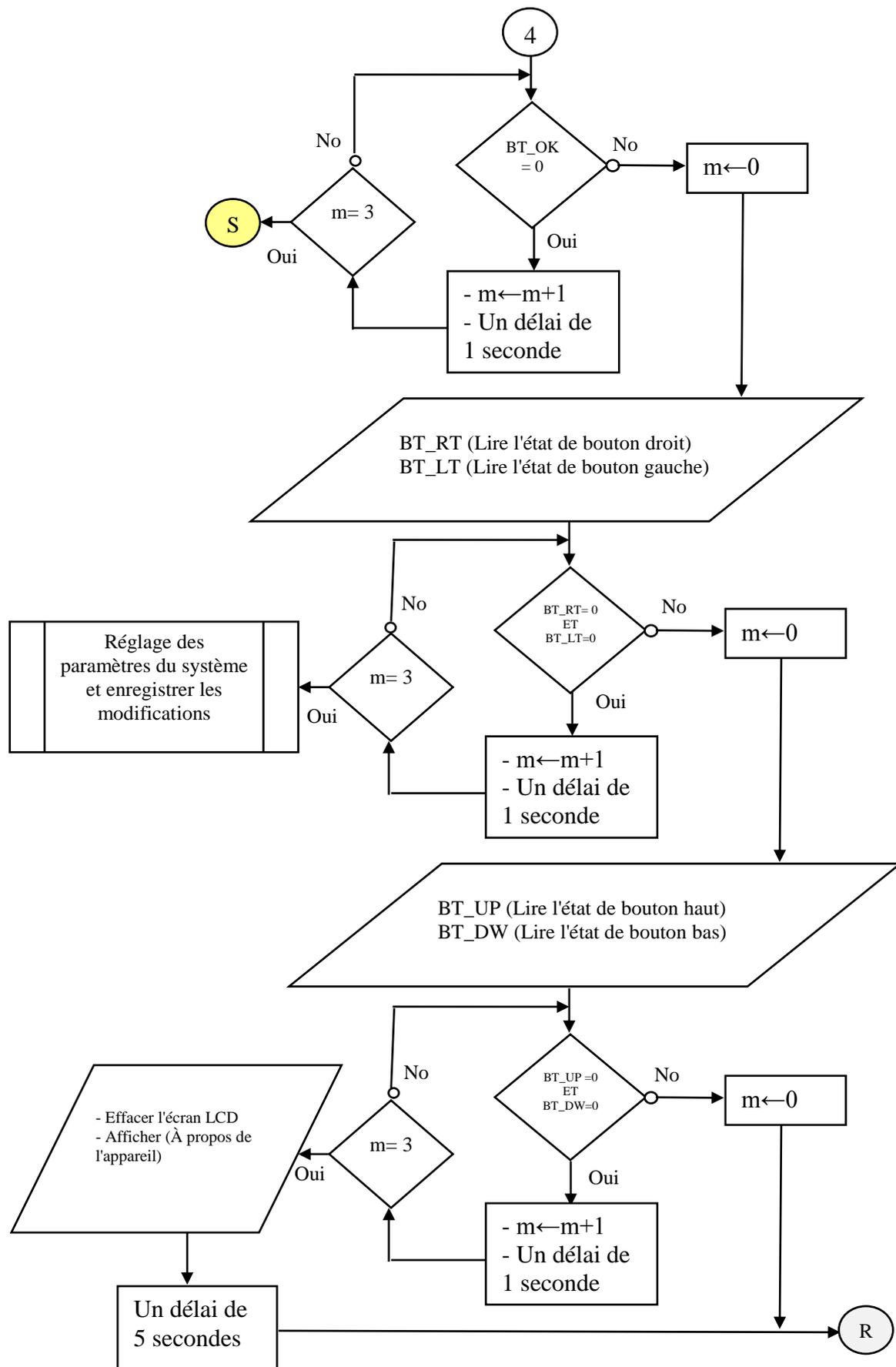


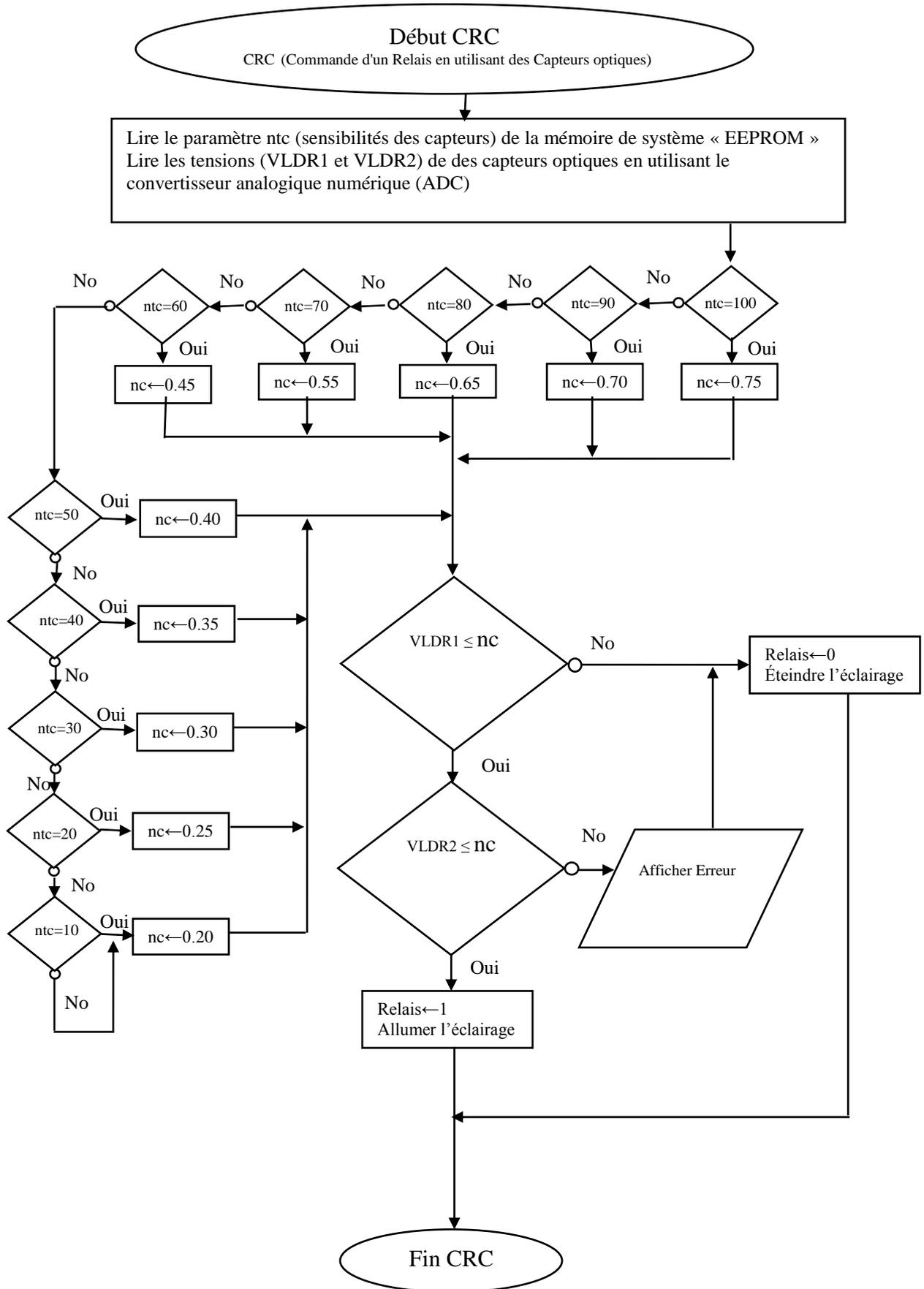
Copyright (C) 2002, OLIMEX Ltd  
<http://www.olimex.com/dev>

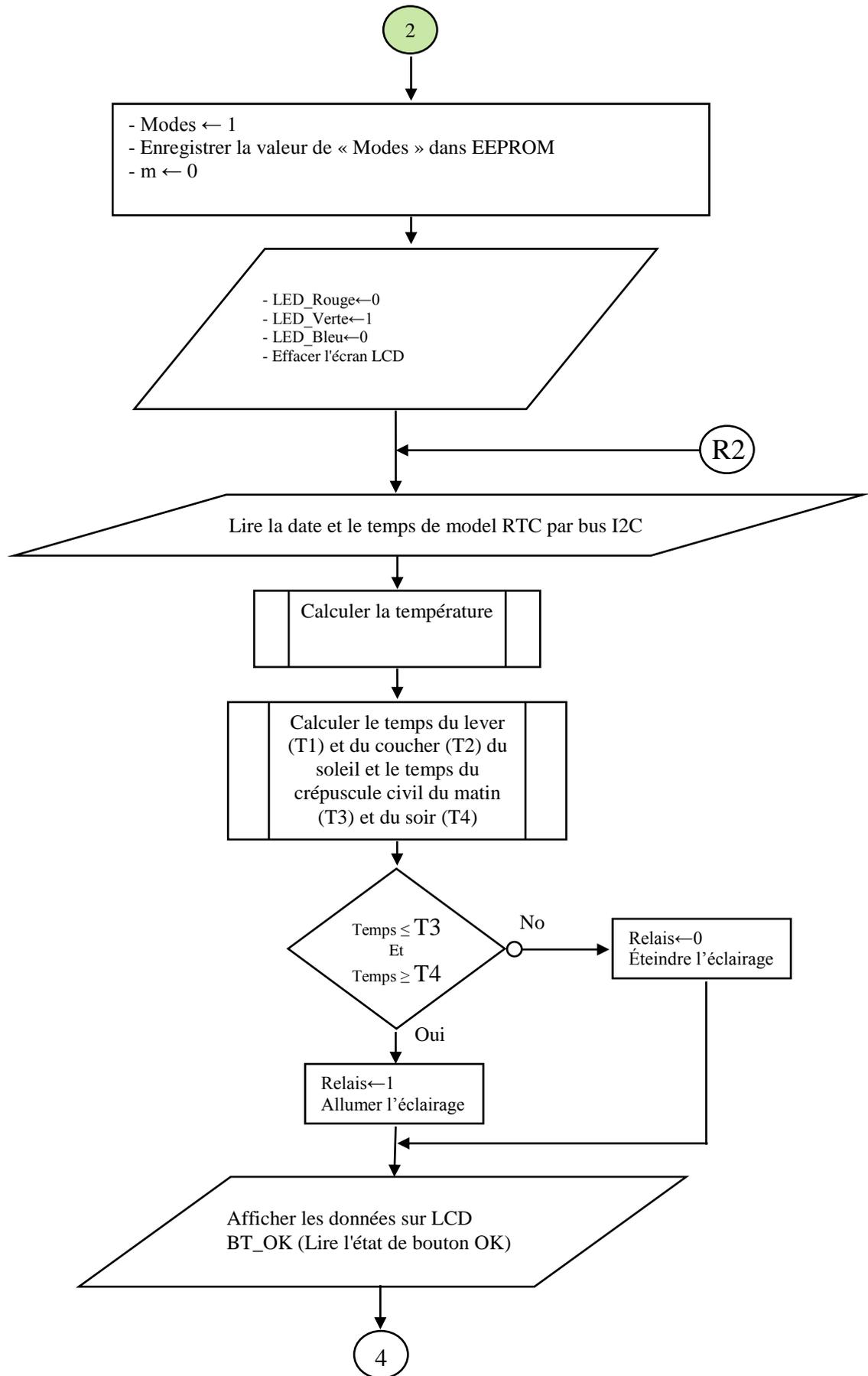
Fig. III.11 : le schéma du Programmeur de PIC [16]

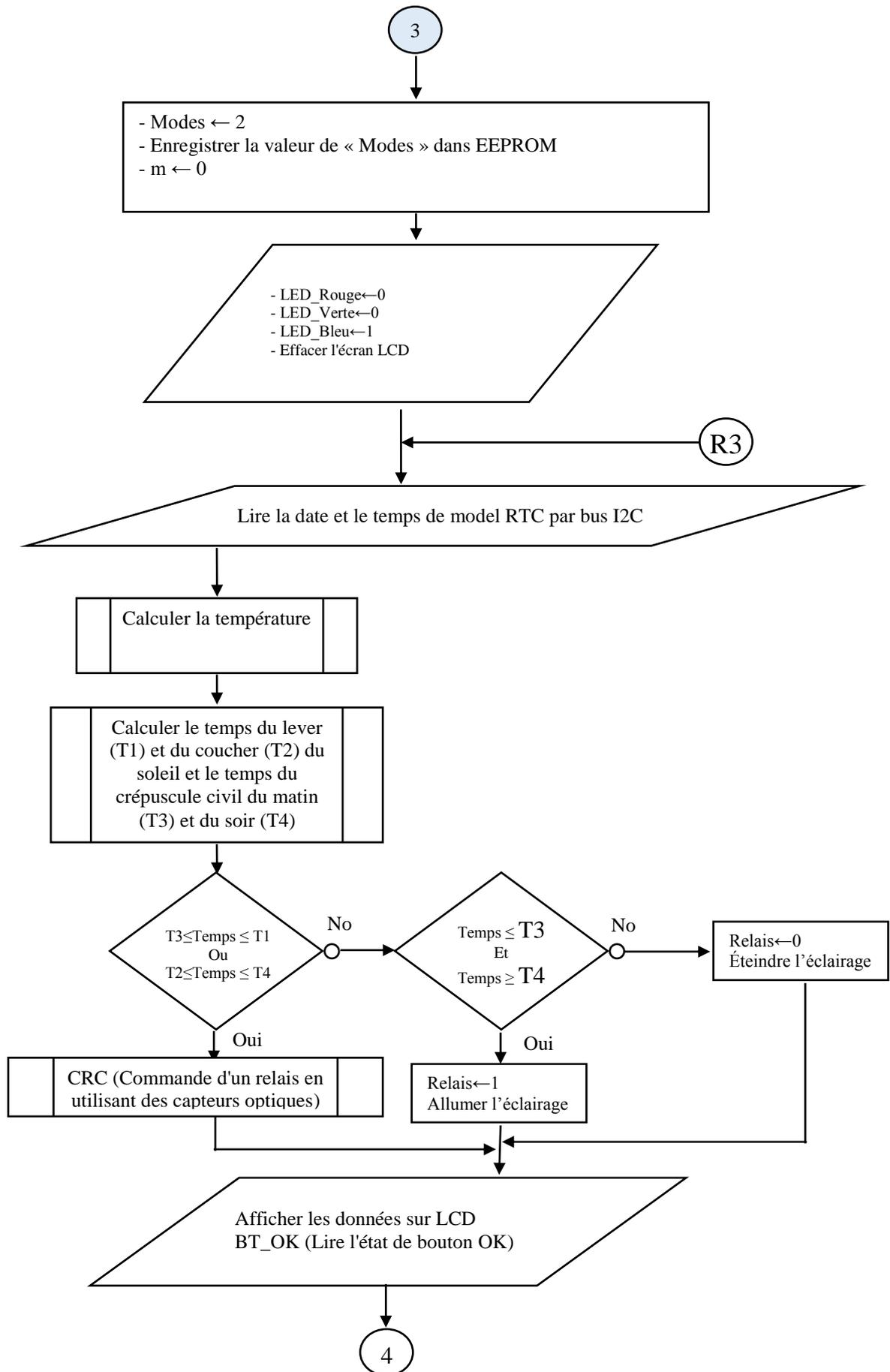
**III. 6 Organigramme de programme :**

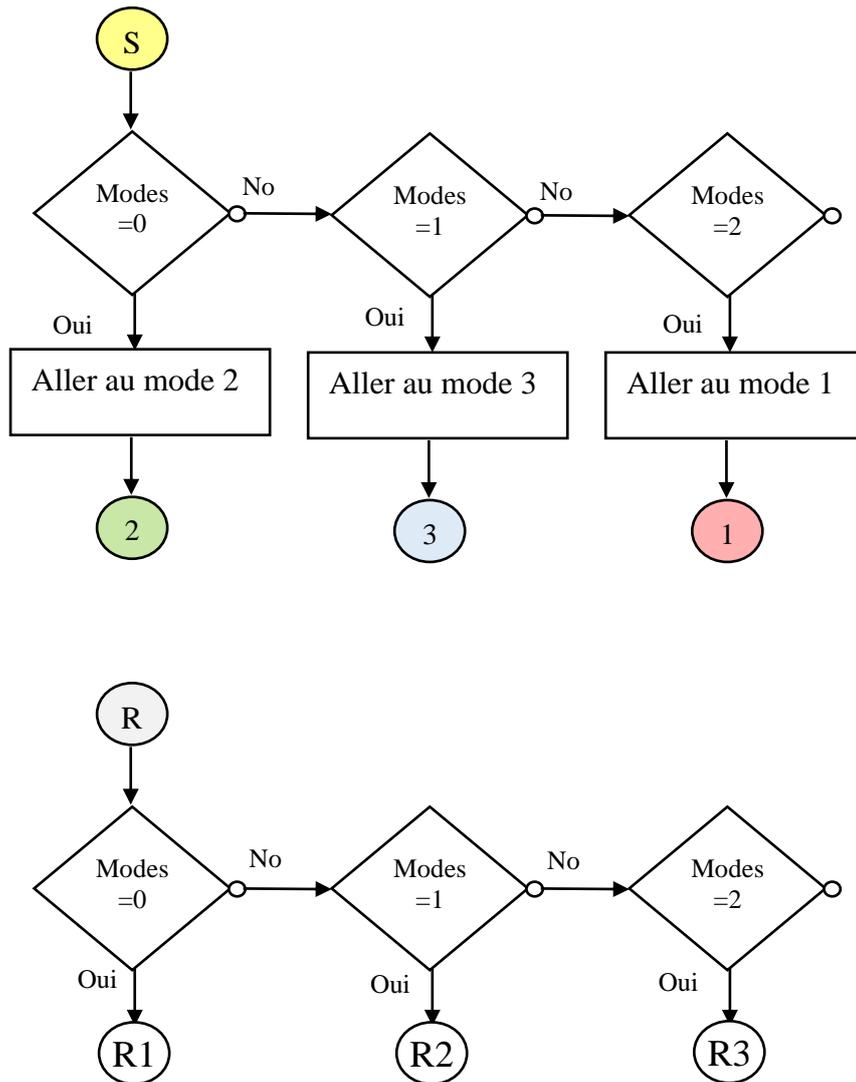












### III. 7 Conclusion :

Afin d'atteindre l'objectif principal du projet, nous avons étudié en détail les trois étapes (programmation, simulation, transfert), après avoir écrit le programme et la vérification de son fonctionnement par le logiciel de simulation. Nous avons envoyé le programme vers le PIC18F4550 pour que nous nous utilisions dans la réalisation pratique de notre système.

**Chapitre IV**  
Conceptions et réalisation de  
système

## IV. 1 Introduction

Dans ce chapitre nous présentons une description détaillée de la solution adoptée pour répondre aux spécifications de notre cahier de charge et en abordant la conception détaillée de chaque partie du système afin d'obtenir une schématisation complète et précise.

## IV. 2 Etude théorique

Notre système de commande est basé sur trois idées, la première idée se base sur des capteurs optiques, la deuxième idée se base sur le calcul de l'heure du lever et du coucher du soleil, la troisième idée se base sur les deux (temps et des capteurs).

L'idée principale de notre appareil est basé sur la lecture des états des capteurs optique et calculer l'heure du lever et coucher du Soleil, ce dernier est le noyau de l'appareil. L'heure du lever et du coucher du Soleil varie en fonction du jour et de l'endroit. Nous ne parlerons pas trop de détails des calculs astronomiques, mais nous aborderons seulement quelques concepts importants comme les coordonnées terrestres et les équations utilisés dans les calculs.

### IV. 2.1 Les équations de l'algorithme utilisé dans les calculs

Plusieurs algorithmes de calcul l'heure du lever et du coucher du soleil sont disponibles sur l'internet. Nous allons utiliser un de ces algorithmes tel qu'il est simple et précis en termes de résultats et facile à mettre en œuvre en termes de programmation.

Avant de commencer à écrire l'algorithme, nous devrions connaître certains termes importants.

#### IV. 2.1.1 Les coordonnées géographiques

Par coordonnées géographiques (ou encore « repères géographiques ») d'un lieu, on entend la latitude, la longitude et le niveau de la mer. Pour se repérer à la surface de la planète. Ce sont les coordonnées angulaires qui permettent le repérage d'un point sur la terre.

La position d'un point M à la surface de la Terre peut être définie par sa latitude  $\varnothing$ , angle de sa normale à l'ellipsoïde terrestre avec le plan de l'équateur, et par sa longitude  $\lambda$ , angle dièdre du méridien du point et d'une méridienne origine [17], [18].

##### a. La longitude

Angle dièdre formé, en un point donné de la surface terrestre, par le méridien du lieu avec le méridien de Greenwich, et compté de 0 à  $\pm 180^\circ$ , 12 h à partir de cette origine, positivement vers l'ouest, négativement vers l'est [18].

**b. La latitude**

Angle formé, en un point donné de la surface terrestre, par la verticale du lieu avec le plan de l'équateur, et compté de 0 à  $\pm 90^\circ$  à partir de l'équateur, positivement vers le nord, négativement vers le sud [18].

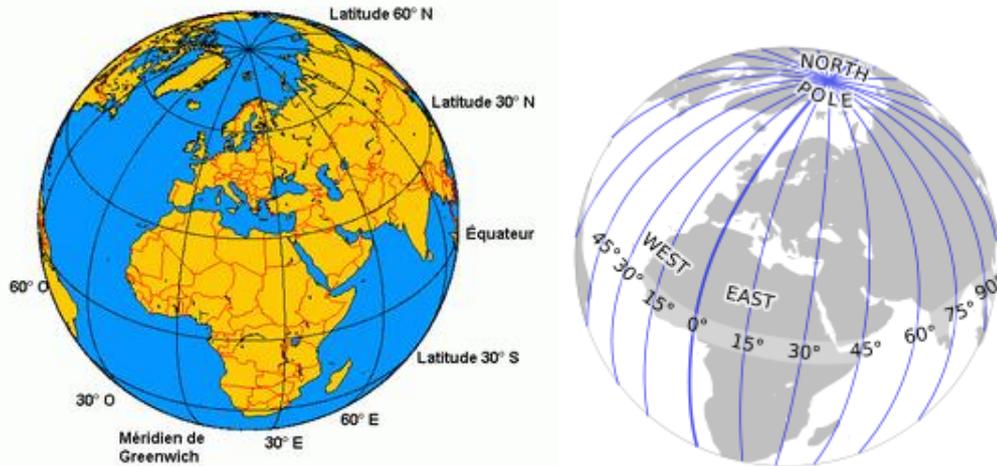


Fig. IV.1 : Les coordonnées géographiques (latitude et longitude)

**IV. 2.1.2 Les fuseaux horaires**

Les fuseaux (Fig. IV.2) horaires sont des divisions géographiques du globe terrestre tous les 15° de longitude, commençant à Greenwich en Angleterre, créées pour aider les gens à déterminer l'heure qu'il est partout dans le monde. [19]

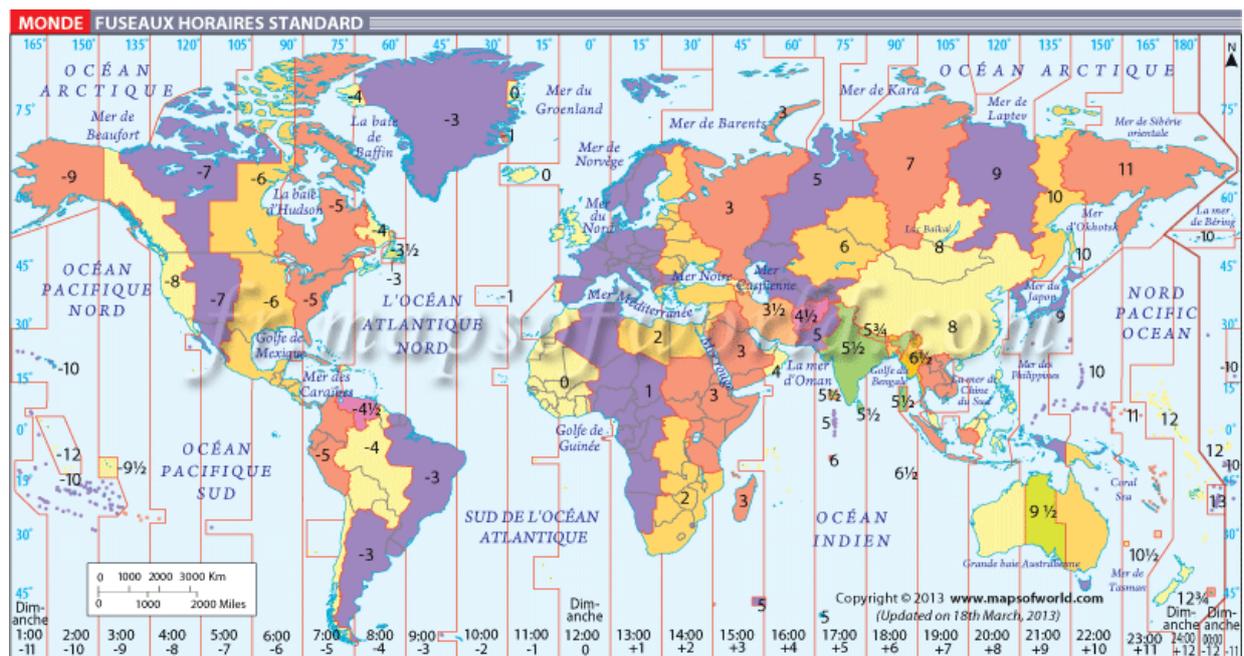


Fig. IV.2 : Carte du monde des fuseaux horaires [20]

### IV. 2.1.3 Crépuscule

Le crépuscule est l'intervalle de temps au soir entre le coucher du Soleil et le moment où le Soleil se situe à un certain nombre de degrés sous l'horizon. Le matin, le crépuscule est l'intervalle de temps entre le moment où le Soleil est encore à un certain nombre de degrés sous l'horizon et le lever du Soleil.

Les phases du crépuscule généralement divisé en trois phases (crépuscule civil, crépuscule nautique, crépuscule astronomique). Dans le cas du crépuscule civil, le centre du disque solaire doit se trouver à  $6^\circ$  sous l'horizon. Pour obtenir ce qu'on appelle le crépuscule nautique, le Soleil doit se situer à  $12^\circ$  sous l'horizon. Pour le crépuscule astronomique, il faut  $18^\circ$ , mais parfois  $15^\circ$  suffisent aussi car à ce moment-là, il fait pratiquement déjà nuit. [21], [22]

#### ❖ Crépuscule civil

Le crépuscule civil est la période où le centre du Soleil est situé à moins de  $6^\circ$  sous la ligne d'horizon. Il subsiste encore suffisamment de lumière pour que la plupart des activités ne nécessitent pas de sources de lumières artificielles [22].

### IV. 2.1.4 Les équations

Les entrées de cet algorithme sont comme suit : le jour (*day*), le mois (*month*) et l'année (*year*), latitude (*latitude*) et longitude (*longitude*), le fuseau horaire (*localOffset*). [23]

Calculer le jour de l'année

```
N1 = floor(275 * month / 9)
N2 = floor((month + 9) / 12)
N3 = (1 + floor((year - 4 * floor(year / 4) + 2) / 3))
N = N1 - (N2 * N3) + day - 30
```

Convertir la longitude vers une valeur de l'heure et calculer un temps approximatif

```
lngHour = longitude / 15
Si l'heure du lever du soleil est nécessaire :
    t = N + ((6 - lngHour) / 24)
Si l'heure du coucher du soleil est nécessaire :
    t = N + ((18 - lngHour) / 24)
```

Calculer l'anomalie moyenne du Soleil

```
M = (0.9856 * t) - 3.289
```

Déterminer la vraie longitude du Soleil

```
L = M + (1.916 * sin(M)) + (0.020 * sin(2 * M)) + 282.634
```

Calculer l'ascension droite du Soleil

```
RA = atan(0.91764 * tan(L))
```

La valeur de l'ascension droite doit être dans le même quadrant que L

```
Iquadrant = (floor(L/90)) * 90
```

```
RAquadrant = (floor(RA/90)) * 90
RA = RA + (Lquadrant - RAquadrant)
```

La valeur de l'ascension droite doit être convertie en heures

```
RA = RA / 15
```

Calculer la déclinaison du Soleil

```
sinDec = 0.39782 * sin(L)
cosDec = cos(asin(sinDec))
```

Calculer l'angle horaire local du Soleil

```
zenith = 90
cosH = (cos(zenith) - (sinDec * sin(latitude))) / (cosDec * cos(latitude))
```

Finalemment calcul de H et de les convertir en heures

```
Si l'heure du lever du soleil est nécessaire :
H = 360 - acos(cosH)
Si l'heure du coucher du soleil est nécessaire :
H = acos(cosH)
H = H / 15
```

Calculer le temps moyen local

```
T = H + RA - (0.06571 * t) - 6.622
```

Calculer le Temps universel coordonné UTC

```
UT = T - lngHour
```

UT potentiellement doit être ajusté dans l'intervalle [0,24) en ajoutant / soustrayant 24

Convertir la valeur UT à fuseau horaire local de latitude / longitude

```
localT = UT + localOffset
```

Convertir la valeur localT à heures et minutes

```
heures = floor(localT)
minutes = floor(localT - heures)*60
```

**Remarque :** la suite des calculs est la même que pour le crépuscule civil, il suffit de remplacer  $zenith = 90$  dans la formule ci-dessus par respectivement  $zenith = 96$ . En fait, la lumière au début du crépuscule civil est peu clair, après quelques minutes il devient une bonne lumière, donc pour obtenir de bons résultats il va utiliser une valeur entre temps de crépuscule civil et l'heure de lever du soleil ( $zenith = 93$ ).

## IV. 2 Les différents blocs de la carte

Dans notre système de commande il y a plusieurs unités comme l'indique la figure ci-dessous (Fig. IV.3), de telle sorte que chaque unité joue un certain rôle, nous allons décrire chaque unité séparément en détail.

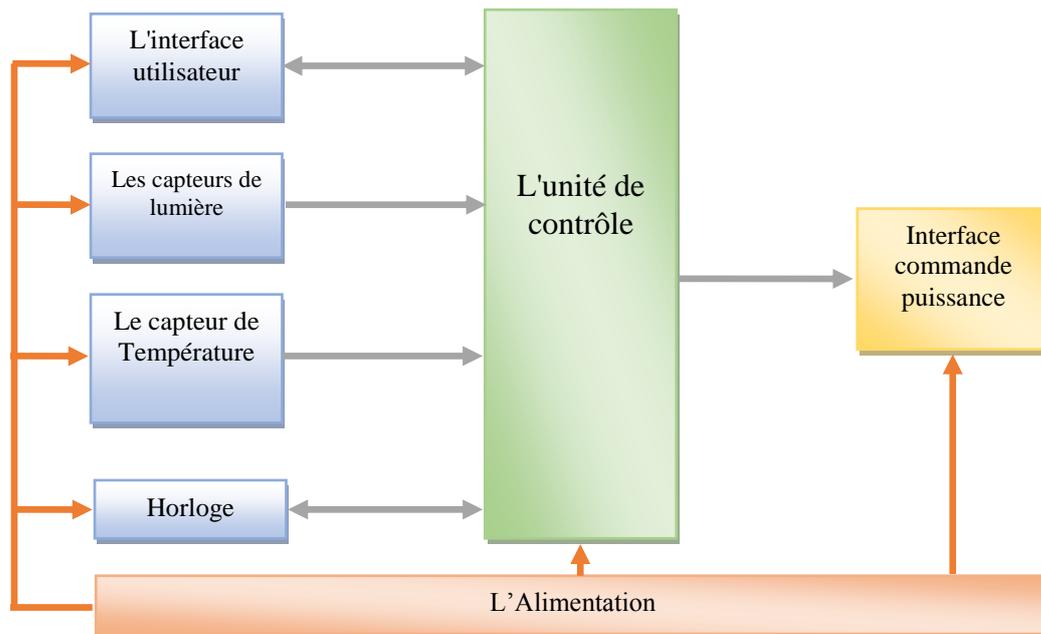


Fig. IV.3 : Les différents blocs de système

#### IV. 2.1 L'Alimentation

Pour alimenter chaque bloc. Nous devons donner une tension stable 5 et 12 volts (pas stable), pour cela nous vous proposerons le montage suivant :

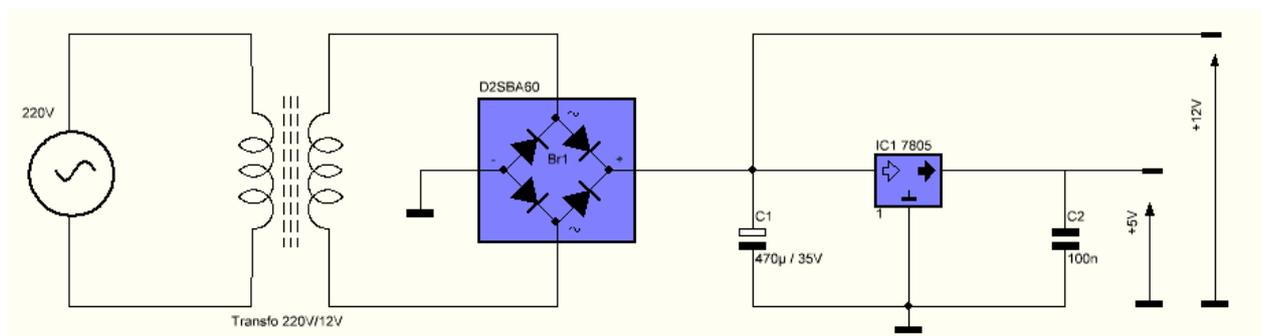


Fig. IV.4 : schéma de l'alimentation classique

##### IV. 2.1.1 Principe du montage

Un transformateur nous fournit une tension alternative de 12v qui est redressée par le pont de diodes et filtrée par le condensateur C1. Ensuite cette tension est régulée à 5v par un régulateur de tension de type LM7805 qui nous fournit une tension positive de 5V avec un courant maximum de 1A, la capacité C2 sert à filtrer les bruits, ses valeur sont données par le constructeur du régulateur.

## IV. 2.2 Le capteur de lumière (LDR)

### IV. 2.2.1 Définition

Une photorésistance (Fig. IV.5) est un composant électronique dont la résistivité varie en fonction de la quantité de lumière incidente. On peut également la nommer résistance photo-dépendante (light-dependent resistor (LDR)) ou cellule photoconductrice [24].

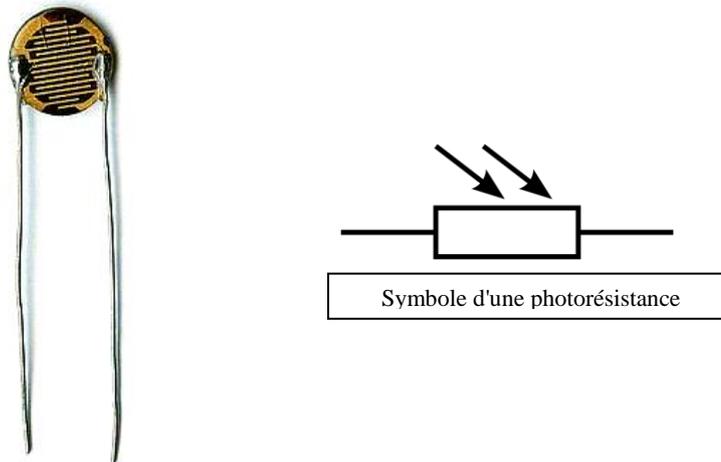


Fig. IV.5 : Symbole et Capteur LDR

### IV. 2.2.2 Principe de fonctionnement

Une photorésistance est composée d'un semi-conducteur à haute résistivité. Si la lumière incidente est de fréquence suffisamment élevée, les photons absorbés par le semi-conducteur donneront aux électrons liés assez d'énergie pour sauter dans la bande de conduction, les électrons libres (avec leurs trous d'électron) ainsi produits abaissant la résistance de l'ensemble [24].

Dans notre carte nous avons utilisée deux capteurs de lumières (LDR 1 et LDR 2), Le premier capteur (LDR 1) qui est considéré comme un capteur principal, et la seconde (LDR 2) fonctionne comme un capteur de réserve et complémentaire du premier capteur.

LDR est un capteur passif donc nous nous devons utiliser une interface entre le capteur et le PIC18F4550 pour présenter la variation de lumière (ou d'ombre) sous la forme d'un signal électrique. La figure suivant (Fig. IV.6) représente un circuit simple composé de trois résistances et deux diodes.

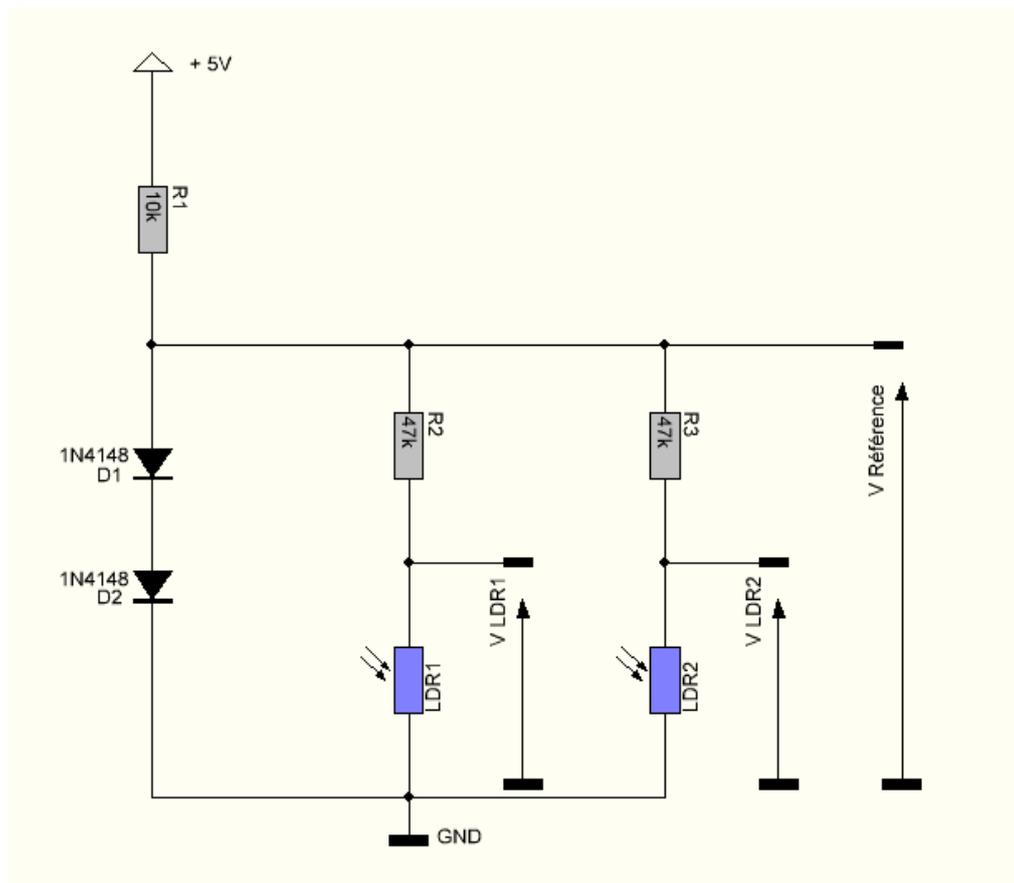


Fig. IV.6 : schéma de l'interface des capteurs de lumières

### IV. 2.2.3 Principe du montage

Le principe de ce circuit est très simple il est basé sur le diviseur de tension (Un circuit électronique constitué de deux résistances en série qui permet de diviser une tension d'entrée). Dans ce cas, nous allons utiliser deux circuits de diviseurs de tension. La tension de sortie du diviseur de tension ( $V_{LDR1}$  et  $V_{LDR2}$ ) donnée par :

$$V_{LDR1} = V_{Référence} \times LDR1 / (LDR1 + R2)$$

$$V_{LDR2} = V_{Référence} \times LDR2 / (LDR2 + R3)$$

Si  $LDR1 = \text{Valeur maximum} \rightarrow V_{LDR1} \approx V_{Référence}$

Si  $LDR1 = \text{Valeur minimum} \rightarrow V_{LDR1} \approx 0$

Si  $LDR2 = \text{Valeur maximum} \rightarrow V_{LDR2} \approx V_{Référence}$

Si  $LDR2 = \text{Valeur minimum} \rightarrow V_{LDR2} \approx 0$

La tension de référence ( $V_{Référence}$ ) est égale à la tension entre les deux diodes et la masse (GND) donnée par :

$$V_{Référence} = VD1 + VD2$$

$VD1, VD2$  : La tension de seuil de la diode (0.62V dans le cas 1N4148)

Le rôle de la résistance  $R1$  est de limiter la circulation de courant dans le circuit et pour diminuer le courant dans les deux diodes ( $D1, D2$  :  $I_{max} = 200\text{mA}$  pour 1N4148).



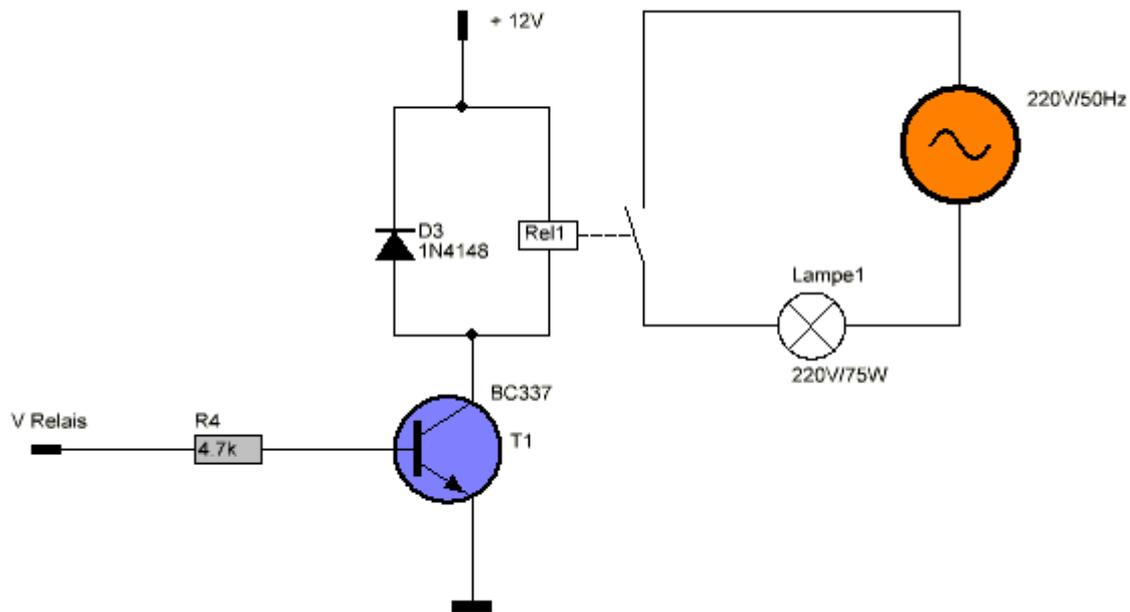


Fig. IV.8 : schéma de l'interface de puissance

Le transistor permet de commander le relais en tout ou rien à partir du signal V Relais. Le relais Rel1 comprend entre ses bornes un bobinage que l'on peut assimiler à une inductance  $L$  en série avec une résistance  $R$ . La diode D3 est une diode de roue libre qui assure la continuité du courant dans l'inductance du relais au blocage du transistor. Sans la diode D3 une surtension destructrice pour le transistor se produirait.

Le transistor BC337 fonctionne en régime de commutation quand son courant base est soit très faible (Transistor bloqué) soit très intense (Transistor saturé). Les deux cas donnée par :

Si  $V \text{ Relais} = 0 \rightarrow$  le transistor est bloqué  $\rightarrow$  l'interrupteur est ouvert

Si  $V \text{ Relais} = 5 \rightarrow$  le transistor est saturé  $\rightarrow$  l'interrupteur est fermé

#### IV. 2.5 L'interface utilisateur

L'interface utilisateur est le médiateur entre l'utilisateur et le microcontrôleur PIC18F4550 dans notre carte, Il est constitué de deux parties (l'afficheur LCD, Le clavier «les boutons poussoirs et les LED»). Ces trois parties capables de saisir les données et de lire les informations.

### IV. 2.5.1 L'afficheur LCD

#### A- Présentation

Les afficheurs à cristaux liquides, appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA).

Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, par leurs dimensions, et aussi par leurs caractéristiques techniques et leur tension de service. Il y a généralement deux types d'afficheurs LCD (Fig. IV.9) [26] :

- Les afficheurs LCD alphanumériques qui affichent des caractères ASCII
- Les afficheurs LCD graphiques qui permettent d'afficher des images

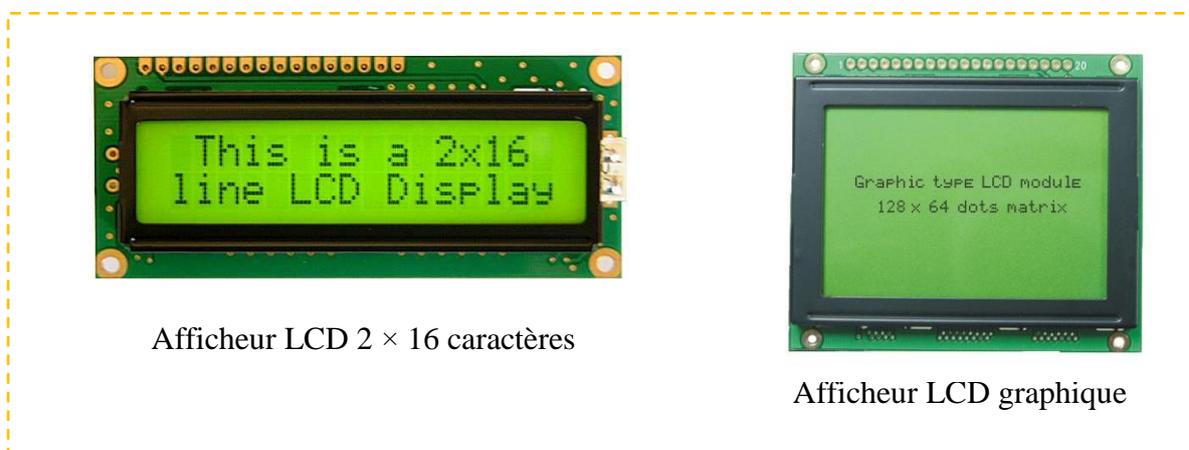


Fig. IV.9 : Types d'afficheurs

Pour afficher les informations (date, heure, heure de lever et de coucher du soleil, la température, états des capteurs...etc.), nous devons au moins cinq lignes de 16 caractères. Nous vous proposons d'utiliser un LCD de téléphone portable NOKIA 3310 qui les surpasse sur tous les points :

- Il est graphique (84x48 pixels)
- Il permet d'afficher jusqu'à 6 lignes de 14 caractères (8x5)
- On peut créer sa propre police de caractères
- Il est facile à piloter (liaison série synchrone à 5 fils)
- Son contraste est époustouflant
- Il ne consomme que 110µA sous 3,3V
- Il dispose d'une vaste bibliothèque prête sur l'internet, compatible avec MikroC
- Des prix vraiment *pas chers*

## B- Brochage

Derrière l'écran à cristaux liquides de téléphone portable NOKIA 3310, on trouve une série de 8 broches (Fig. IV.10) qu'ils exercent les rôles suivantes :

- 1- VDD : alimentation de l'afficheur (tension VDD entre 2,7V et 3,3V)
- 2- SCLK : horloge synchrone
- 3- SDIN : entrée données série
- 4- D/C : sélection donnée/commande
- 5- SCE : sélection du circuit
- 6- GND : masse
- 7- CAP : Cette broche est généralement reliée par un condensateur (1-10 $\mu$ F) externe à la masse
- 8- RES : reset

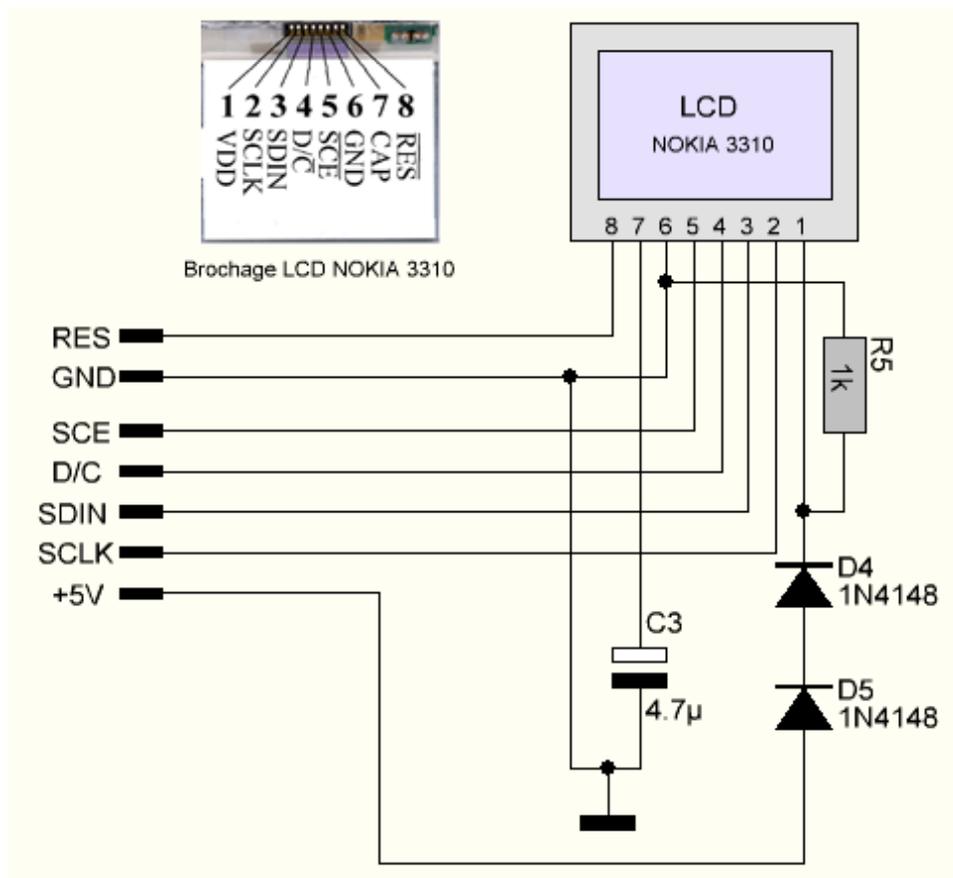


Fig. IV.10 : Brochage et schéma d'interface de LCD Nokia 3310

## IV. 2.5.2 Le clavier (les boutons poussoirs et les LED)

### a. Les boutons poussoirs

Les boutons poussoirs nous permettent de saisir des données et de sélectionner les différentes options utilisées dans notre projet. Nous allons utiliser cinq boutons (haut, bas, gauche, droite, central), ces boutons travaillent comme suit :

- bouton-haut : Utilisé pour changer les valeurs saisies (augmenter).
- bouton-bas : Utilisé pour changer les valeurs saisies (diminuer).
- bouton-gauche : Utilisé pour naviguer entre les valeurs dans la page des paramètres.
- bouton- droit : Utilisé pour naviguer entre les valeurs dans la page des paramètres.
- bouton- droit + bouton-gauche : Utilisés sur l'accès à la page des paramètres ou retour à la page d'accueil.
- bouton- haut + bouton-bas : Utilisés pour l'accès à la page « À propos de l'appareil ».
- bouton- central (Ok) : Utilisé pour choisir le mode de fonctionnement de l'appareil de la page d'accueil ou enregistrer les modifications de la page des paramètres.

L'état logique de toutes les entrées numériques du microcontrôleur peut être modifié par l'intermédiaire des boutons poussoirs. Généralement, il y a deux états logiques peuvent être représentés par ces boutons poussoirs 1 et 0 logique, la figure (IV.11) présente un circuit qui contient cinq boutons poussoirs. L'état logique pour l'état normal dans les bus (B\_LFT, B\_UP, B\_OK, B\_RTH, B\_DWN) est 1, lorsque vous appuyez sur les boutons poussoirs il change à 0.

### b. 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 quatre LEDs, chaque LED présente un mode de fonctionnement avec quatre résistances pour limiter le courant et réduire la tension (Fig. IV.11).

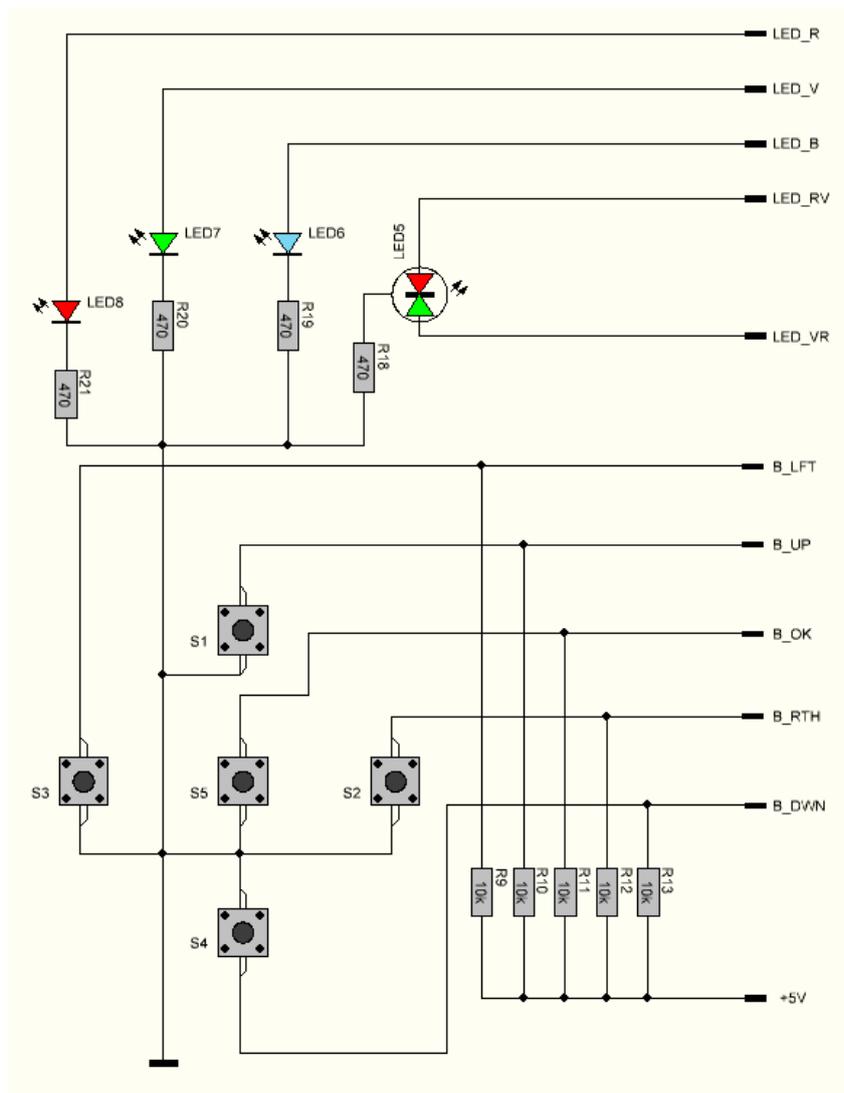


Fig. IV.11 : schéma du clavier

## IV. 2.6 Horloge

Pour ce système de commande on a besoin de connaître en tout moment la date et l'heure actuelle, afin de déterminer les horaires de coupure de l'éclairage ou ré-exécuter. Ceci est fait par une horloge temps réel. L'horloge temps réel utilisé est le DS1307 de Dallas.

### IV. 2.6.1 Description du composant

C'est un RTC, c'est-à-dire une horloge à temps réel qui va fournir les secondes, minutes et heures, mais aussi les jours, mois et heures grâce à un cristal de quartz. Une pile externe lui permet toujours être fonctionnelle même en cas de coupure de l'alimentation. La figure (IV.13) représente le schéma du module RTC DS1307.

Le module DS1307 se contrôle par I2C, il est considéré comme esclave d'adresse 1101000 (0x68 en hexadécimal). Il contient 64 octets de ram, sur c'est 64 octets de ram, 7 contiennent la

date et l'heure (incrémenté automatiquement) que chaque octet représente un registre, plus un octet pour la configuration du DS1307, l'organisation de ces registres dans cet ordre :

Tableau. IV.1 : Organisation des registres de RTC [27]

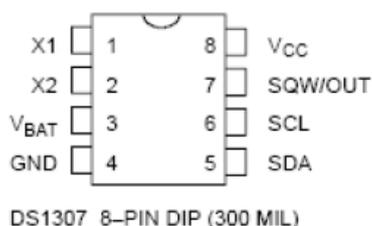
L'adresse	La fonction
00h	les secondes (de 0 à 59)
01h	minutes (de 0 à 59)
02h	heures (de 0 à 23)
03h	jour de la semaine (de 1 à 7)
04h	date du jour (dans le mois, de 1 à 31)
05h	mois (de 1 à 12)
06h	l'année (à partir de 0, il suffit de lui ajouter 2000)
07h	la configuration

Les 56 octets restant sont utilisés comme mémoire RAM externe, on pourra y stocker des données d'usage (des variables par exemples). Les valeurs (jours, mois, années, ...) sont stocké en BCD (décimal codé binaire).

L'avantage d'un tel composant est qu'il ne se décale pas dans le temps. Son système reste parfaitement précis. De plus il tient compte des années bissextiles jusqu'en 2100.

#### IV. 2.6.2 Brochage du DS1307

L'horloge temps réel DS1307 est commercialisé dans un boîtier de 8 broches classiques, comme illustré dans la figure IV.12



Pin	Nom	Rôle
1	X1	cristal
2	X2	cristal
3	Vbat	tension batterie
4	GND	masse
5	SDA	ligne des données (data) I <sup>2</sup> C
6	SCL	ligne d'horloge (clock) I <sup>2</sup> C
7	SQW/out	sortie signal carré
8	Vcc	tension logique 5V

Fig. IV.12 : Brochage du DS1307 [28]

### IV. 2.6.3 Schéma du module RTC

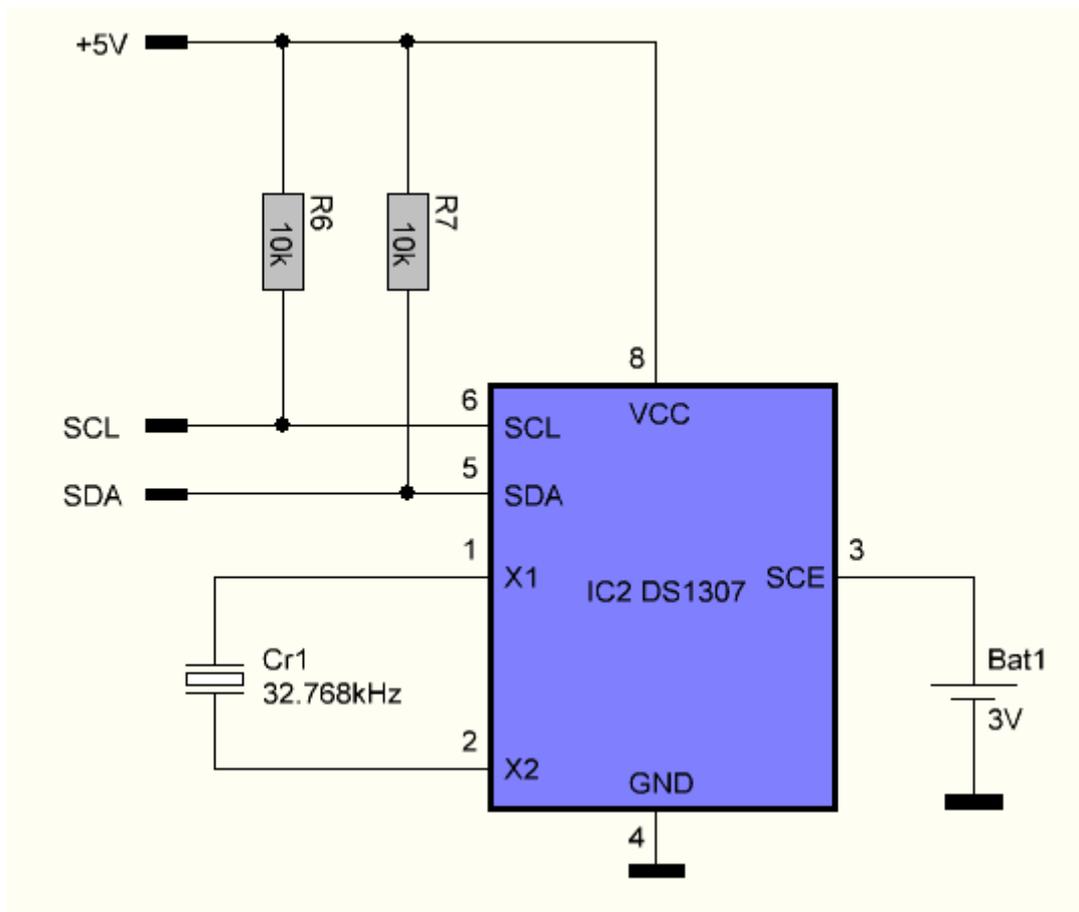


Fig. IV.13 : schéma du module RTC

### IV. 2.7 Unité de contrôle

L'unité de contrôle est basé sur le PIC18F4550 le schéma de cet unité est représenté en figure (fig. IV.14) les ports du PIC seront configurés comme suit :

**Le port A :** le bit RA0 est configuré comme entrée analogique du capture de température, le bit RA5 configuré comme entrée analogique du capture de lumière, le bit RA3 configuré comme entrée de la tension de référence VREF+ du convertisseur analogique/numérique, et les deux bits RA6/OSC2, RA7/OSC1 comme entrée de l'oscillateur HS 20MHZ.

**Le port B :** est configuré comme sortie, les deux bits (RB0/SDA, RB1/SCL) utilisés pour communiquer avec l'horloge temps réel (DS1307) par bus I2C, et les six bits restant vers l'afficheur LCD.

**Le port C :** nous utilisons trois bits comme suite les bits RC5 et RC6 sont configuré comme entrée numérique dès les boutons poussoir, le bit RC7 est configuré comme sortie vers LED rouge.

**Le port D** : le bit RD0 est configuré comme sortie vers l'interface commande puissance.

Les trois bits (RD1, RD2, RD3) sont configuré comme entrée numérique utilisés pour les boutons poussoir, et les quatre bits restant sont configuré comme sorties vers les LED.

**Le port E** : le bit RE0 est configuré comme entrée analogique de la capture de lumière, le bit RE3 est configuré comme entrée du signal reset du bouton poussoir S1.

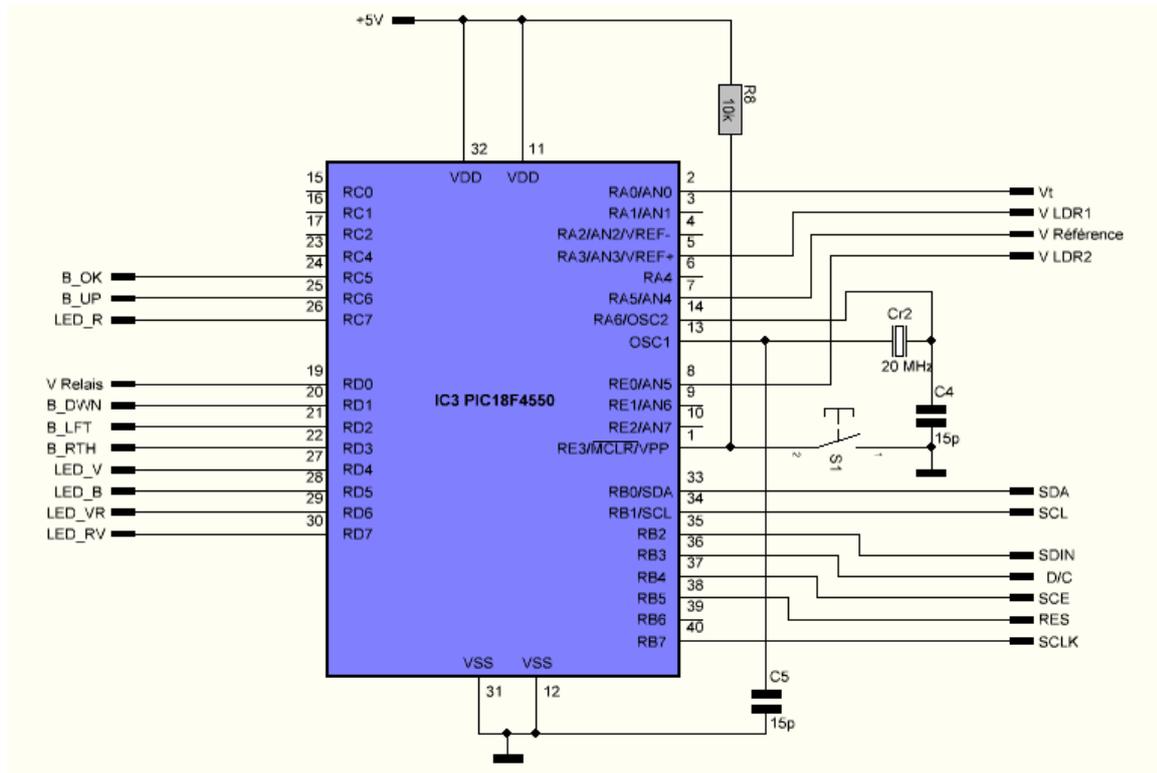


Fig. IV.14 : schéma de l'unité de contrôle

### IV. 3 Fonctionnement de système

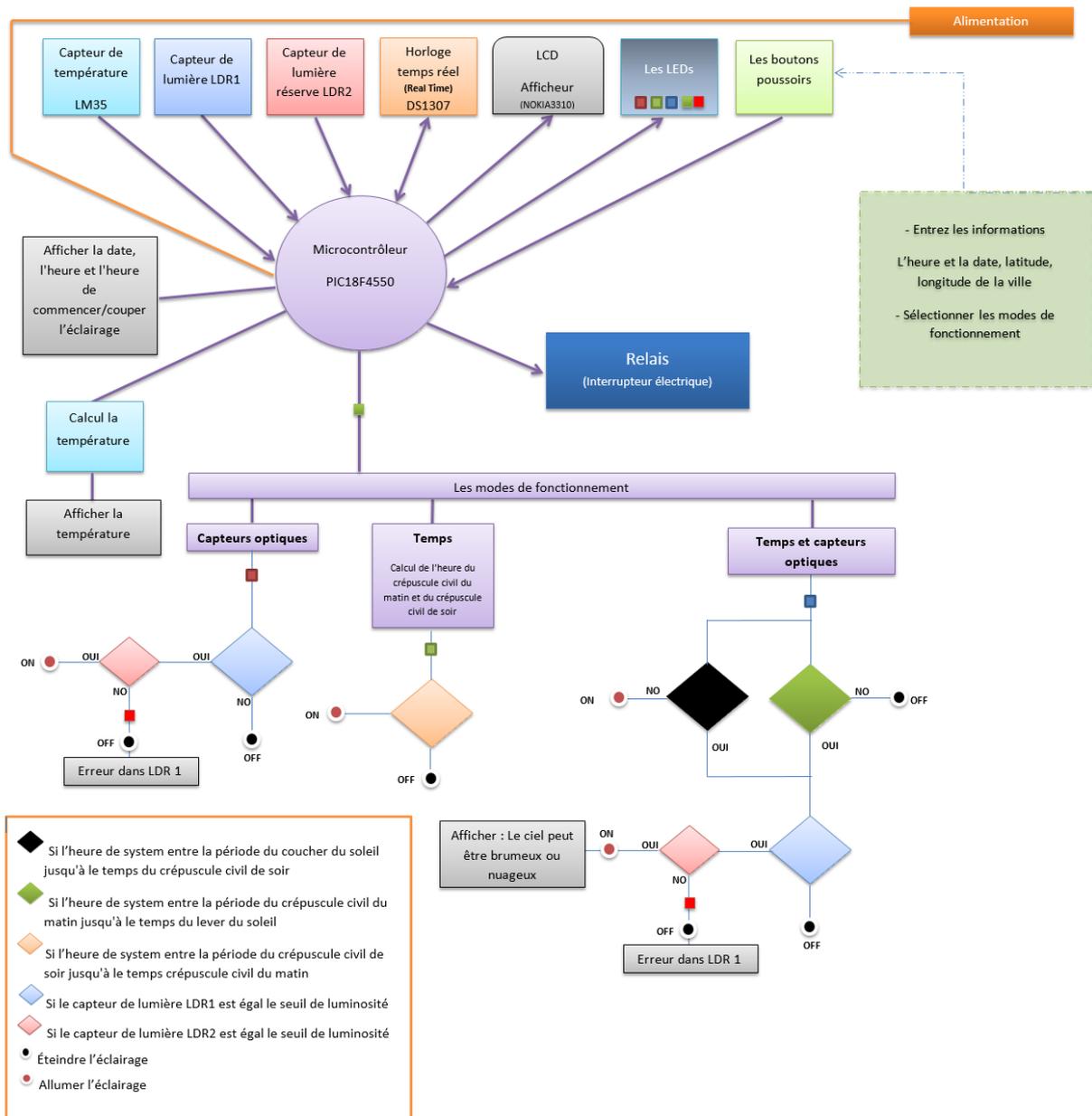


Fig. IV.15 : Conceptions et fonctionnement de système

Dans notre système de commande ils existent trois modes de fonctionnement de base. Le mode souhaité est sélectionné par le bouton de commande « Ok », pour chaque mode clignoter une LED (LED rouge pour le premier mode, le vert pour le deuxième mode et le bleu pour le troisième mode), et une LED rouge-vert clignoter en vert si l'appareil fonctionne correctement et en rouge si il y'a un problème dans le capteur de lumière.

Le premier mode est basé sur les capteurs de lumière, ces derniers envoient un signal électrique vers le PIC, ce signal est limité entre 0 et  $V_{ref}$  de telle sorte que  $v_{ref} = 1.2V$ , la

valeur minimale(0) représente la lumière et la valeur maximal (vreff) représente l'obscurité. La valeur de crépuscule existe entre les deux valeurs précédentes et définie par l'utilisateur.

Dans le cas crépuscule de jour le PIC envoie un signal a l'interface de puissance à travers le port D (*le bit RD0*), égale a 0 logique pour stopper le flux de courant dans le relais et donc couper l'éclairage, dans le cas de crépuscule de soir il envoie un signal égale a 1 logique pour allumer l'éclairage. Le rôle du second capteur est de vérifier le bon fonctionnement du premier capteur.

Le deuxième mode est basé sur le calcul des temps du crépuscule à travers les coordonnées géographiques de l'endroit, et le microcontrôleur à chaque jour calcule les temps du crépuscule (*le crépuscule civil*).

Lorsque l'horloge de l'appareil est égale au temps du crépuscule, le microcontrôleur envoie un signal à l'interface de puissance pour allumer ou éteindre l'éclairage, l'état de l'atmosphère ici ne prend pas en compte parce que dans ce cas les capteurs de lumière sont désactivés.

Le troisième mode fonctionne comme le deuxième mode, selon le temps, mais il prend en compte l'état de l'atmosphère, dans ce cas le microcontrôleur calcule deux périodes de temps (*la première période du coucher du soleil jusqu'à le temps du crépuscule civil de soir et la deuxième période du crépuscule civil du matin jusqu'à le temps du lever du soleil*) et activer les capteurs de lumière dans ces périodes de temps pour confirmer la suffisance de la lumière.

Si la lumière n'est pas suffisante pour l'éclairage dans la première période, le microcontrôleur envoie un signal à l'interface de puissance pour allumer l'éclairage, si non (lumière suffisante) le microcontrôleur envoie un signal pour éteindre l'éclairage, après cette période (la période de nuit), le microcontrôleur envoie toujours un signal pour allumer l'éclairage jusqu'à la deuxième période, dans cette période le microcontrôleur vérifie la suffisance de la lumière par les capteurs, si la lumière est suffisante et clair, il envoie un signal a l'interface de puissance pour couper l'éclairage, si non (n'est pas suffisante) le microcontrôleur envoie un signal pour allumer l'éclairage, après cette période le microcontrôleur envoie un signal pour couper l'éclairage jusqu'à revenir au première période et répétez le même processus.

L'horloge de l'appareil est basée sur RTC (horloge temps réel), pour que le microcontrôleur à chaque seconde il donne le temps et la date de RTC et enregistre également toutes les valeurs des informations (*longitude, latitude, fuseaux horaires, pourcentage des sensibilités des capteurs de lumières*) sur son mémoire interne (*EEPROM*).

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.

#### IV. 4 Consommation d'énergie

Le tableau (Tableau. IV.2) présente une comparaison entre les résultats pratiques obtenus à partir de notre système avec les résultats du site web [29] (*un site qui se spécialise dans le calcul des heures de lever et de coucher du soleil*) concernant les temps du lever et du coucher du soleil et aussi le nombre d'heures de fonctionnement pour deux modes pendant quelques jours et pour l'an 2014 de la ville de Biskra.

Tableau. IV.2 : le nombre d'heures de fonctionnement de système

	Date	Mai 2013		Juin 2013		Août 2013		Octobre 2013		Janvier 2014		2014	
		18	19	02	03	22	23	10	11	01	02		
Résultats du site web	Lever du soleil	05:31	05:30	05:24	05:23	06:03	06:04	06:41	06:41	07:47	07:47		
	Coucher du soleil	19:38	19:39	19:49	19:49	19:18	19:17	18:09	18:08	17:37	17:37		
Résultats de notre système	Lever du soleil	05:35	05:35	05:28	5:28	06:06	06:06	06:40	06:41	07:45	07:45		
	Coucher du soleil	19:35	19:36	19:45	19:46	19:18	19:16	18:12	18:11	17:41	17:42		
Nombre d'heures de fonctionnement avec la deuxième mode		09h, 37min		09h, 19min		10h, 25min		12h, 09min		13h, 41min			~ 4164h
Nombre d'heures de fonctionnement avec la troisième mode	Max	10h		09h, 43min		10h, 47min		12h, 30min		14h, 03min			~ 4310h
	Min	09h, 37min		09h, 19min		10h, 25min		12h, 09min		13h, 41min		~ 4164h	

Le tableau (Tableau. IV.3) présente la consommation d'énergie d'un poste de puissance de 2kwh pour deux modes.

Tableau. IV.3 : Exemple de consommation d'énergie d'un poste

Puissance du poste en kW	Nombre d'heures de fonctionnement avec la deuxième mode	Nombre d'heures de fonctionnement avec la troisième mode	Consommation par an avec la deuxième mode	Consommation par an avec la troisième mode
2 KWh	4164h	Entre 4164h et 4310h	8328 KWh	Entre 8328 KWh et 8620 KWh

#### IV. 5 Conclusion

Dans ce chapitre, nous avons donné quelques concepts théoriques et nous avons expliqué chaque composant principale du notre montage et par la suite le principe de fonctionnement de notre carte.

## CONCLUSION GENERALE

Cette étude nous donne un ensemble de solutions possibles pour une commande optimale dans l'éclairage public et pour la réduction du coût de la consommation d'énergie, où nous avons traité trois solutions pour la commande du temps d'allumer ou d'éteindre l'éclairage.

Nous avons utilisé la première solution pour résoudre le problème des 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 deux capteurs au lieu d'un seul capteur, alors que le processus ne sera pas exécuté uniquement lorsque les lectures lumineuses dans les deux capteurs sont égaux.

Cette solution donne des résultats acceptables en particulier lorsqu'on installe les capteurs éloignés les uns des autres, mais la probabilité de défaillance des deux capteurs reste inévitable.

De là, nous devons passer à la deuxième solution, qui ne comptera pas sur les capteurs optiques, mais nous basons sur le facteur temps, on calcule le temps du crépuscule civil du matin et du soir pour un lieu particulier, où l'allumage de l'éclairage est commandé uniquement par ce temps, les résultats de ce processus est très précis, mais l'état de l'atmosphère n'est pas prise en compte, contrairement à la solution précédente, à partir d'ici, nous avons introduit la troisième solution, qui combine entre la première et la deuxième solution où le temps de marche ou d'arrêt de l'éclairage suivie ou précédé par une période de temps limitée, utilisant les capteurs optiques, les résultats de cette dernière sont très exactes et le processus prend en compte l'état de l'atmosphère en deux périodes de temps limitées, même dans l'existence de la défaillance au niveau des capteurs optiques, ça n'afflue que dans une période de temps limitée

Toutes ces solutions ont été mises en œuvre au sein d'un système unique et laisser le choix entre ces solutions. Ces solutions restent ouvertes et évolutives par des nouvelles idées, comme l'ajout d'une propriété d'organisation opérationnelle de l'éclairage en basant sur le trafic des routes en nuit et ainsi réduire et d'augmenter la quantité de lumière dans ces routes, ou d'ajouter un système d'échange des données à distance entre eux, ou entre les principaux centres de commande.

## Bibliographie

- [1] : ([www.mirari.fr/eKKb](http://www.mirari.fr/eKKb)) : "Prédiagnostic énergétique de l'Éclairage public de Barbentane", 21/03/2013 / 09 :30
- [2] : ([www.siea.fr/](http://www.siea.fr/)) : "La FICHE ENERGIE Eclairage public l'horloge astronomique", 21/03/2013 / 09 :34
- [3] : ([www.la-diege.fr/telechargements/fichiers/eco\\_ep.pdf](http://www.la-diege.fr/telechargements/fichiers/eco_ep.pdf)) : " Eclairage public et économie d'énergie.", 22/03/2013
- [4] : ([www.mon-club-elec.fr](http://www.mon-club-elec.fr)) : "Qu'est-ce qu'un microcontrôleur ?", 10/03/2013 / 15 :21
- [5] : Guessoumi Farid/Saadi Amel, Commande de panneau solaire a l'aide d'une carte a pic, Université Mohamed Khider, Biskra, Algerie, 2009/2010, p23-33
- [6] : ([www.technologuepro.com/microcontrolleur/Chap1.htm](http://www.technologuepro.com/microcontrolleur/Chap1.htm)) : "Les Microcontrôleurs PIC", 21/03/2013
- [7]: ([ww1.microchip.com/downloads/en/devicedoc/39632e.pdf](http://ww1.microchip.com/downloads/en/devicedoc/39632e.pdf)): "PIC18F2455/2550/4455/4550 Data Sheet", 16/03/2013
- [8] :([www.audiovore.zxq.net/electronique.php](http://www.audiovore.zxq.net/electronique.php)), "La programmation des pic 18F", 10/05/2013
- [9] : (<http://ebookbrowse.com/cours-complet-pic18f4520-v1-1-pdf-d253954337>) "Informatique industrielle : le PIC 18F4520 et sa programmation en C" 30 - 32 - 28/03/2013
- [10] : Pascal Mayeux, "Apprendre la programmation des PIC High Performance par l'expérimentation et la simulation", DUNOD Editions : ETSF, Paris, France, Juillet 2010, p68 - 80
- [11] : ([pf-mh.uvt.rnu.tn/491/1/Conception\\_et\\_réalisation\\_\\_d'un\\_enregistreur\\_de\\_données\\_.pdf](http://pf-mh.uvt.rnu.tn/491/1/Conception_et_réalisation__d'un_enregistreur_de_données_.pdf)) Alibi Elmehdi/Jawadi Sami, Rapport de projet de fin d'études Conception et réalisation d'un enregistreur de données, 2010/2011, p43, 1/04/2013/ 9 :15
- [12] : ([fr.scribd.com/doc/36434413/Programmation-en-C-du-Microcontrolleur-PIC16F877](http://fr.scribd.com/doc/36434413/Programmation-en-C-du-Microcontrolleur-PIC16F877)) "Programmation en C du Microcontrolleur PIC16F877", 1/04/2013 / 21 :20
- [13] : ([www.thierry-lequeu.fr/data/RAP-TUILARD-P1.pdf](http://www.thierry-lequeu.fr/data/RAP-TUILARD-P1.pdf)), TUILARD Benjamin, "Rapport de projet tutoré 2ème année Horloge à LED Commandée par protocole I2C avec un RTC", p15-16, 4/04/2013 / 21 :21
- [14] : ([www.gesi.asso.fr/web/ressources-ressources-dl.php?fichier=46](http://www.gesi.asso.fr/web/ressources-ressources-dl.php?fichier=46)), "L'EDITEUR ISIS Version 6.5 SP5", p30-31, 1/04/2013 / 16 :30

- [15] : ([www.siea.fr/](http://www.siea.fr/)), "SAISIE DE SCHEMA ISIS 6", p3, 1/04/2013 / 16 :33
- [16] : ([www.olimex.com/Products/PIC/Programmers/PIC-PG2/](http://www.olimex.com/Products/PIC/Programmers/PIC-PG2/)), "Olimex JDM PIC-PG2"  
2/04/2013 / 21:30
- [17] : ([fr.wikipedia.org/wiki/Coordonnées géographiques](http://fr.wikipedia.org/wiki/Coordonnées_géographiques)), "Coordonnées géographiques",  
10/01/2013 / 19 :38
- [18] : ([www.larousse.fr/encyclopedie/divers/coordonnée/37093](http://www.larousse.fr/encyclopedie/divers/coordonnée/37093)), "Coordonnées géographiques",  
1/05/2013 / 19 :50
- [19] : ([www.docmirror.net/fr/linux/howto/admin/TimePrecision-HOWTO/ar01s02.html](http://www.docmirror.net/fr/linux/howto/admin/TimePrecision-HOWTO/ar01s02.html)),  
"Qu'est-ce qu'un fuseau horaire ?", 1/05/2013 / 20 :00
- [20] : ([fr.mapsofworld.com/time-zone-map/world.html](http://fr.mapsofworld.com/time-zone-map/world.html)), "Carte De Temps Zones Monde",  
1/05/2013 / 15 :21
- [21] : ([www.astro.oma.be/GENERAL/INFO/fri007.html](http://www.astro.oma.be/GENERAL/INFO/fri007.html)), "Le lever et le coucher du Soleil ainsi  
que le crépuscule", 22/04/2013 / 14 :40
- [22] : (<http://fr.wikipedia.org/wiki/Crépuscule>), "Crépuscule", 22/04/2013 / 15 :21
- [23]: ([williams.best.vwh.net/sunrise\\_sunset\\_algorithm.htm](http://williams.best.vwh.net/sunrise_sunset_algorithm.htm)), "Sunrise/Sunset Algorithm",  
10/01/2013 / 15:21
- [24] : ([fr.wikipedia.org/wiki/Photorésistance](http://fr.wikipedia.org/wiki/Photorésistance)), "Photorésistance", 22/04/2013 / 15:30
- [25] :  
([ma.www.free.fr/enseirb/ma\\_www\\_fichiers/rapports\\_projets/2\\_eme\\_annee/projets\\_e2/projet\\_num/e2\\_proj\\_num\\_\\_thermometre\\_\\_rapport.pdf](http://ma.www.free.fr/enseirb/ma_www_fichiers/rapports_projets/2_eme_annee/projets_e2/projet_num/e2_proj_num__thermometre__rapport.pdf) ), SARGOS Mathilde/ ZANCAN Adrien, "Projet  
de système numérique Thermomètre à affichage digital", p7, 10/01/2013 / 15:21
- [26] : ([www.aurel32.net/elec/lcd.php](http://www.aurel32.net/elec/lcd.php)), "Les afficheurs à cristaux liquides", 10/01/2013 / 16:37
- [27] : ([datasheets.maximintegrated.com/en/ds/DS1307.pdf](http://datasheets.maximintegrated.com/en/ds/DS1307.pdf)), "DS1307 datasheets", 2/03/2013 /  
15:21
- [28] : (<http://www.pobot.org/Horloge-et-calendrier-I2C.html>), " Horloge RTC et calendrier I2C",  
2/03/2013 / 15:21
- [29] : ([www.sunrise-and-sunset.com](http://www.sunrise-and-sunset.com)), «Lever et coucher du soleil » (*un site qui se spécialise dans  
le calcul des heures de lever et de coucher du soleil*)

*Annexe*

*Réalisation pratique*

## 1- Caractéristiques du LM7805

### Electrical Characteristics (MC7805/LM7805)

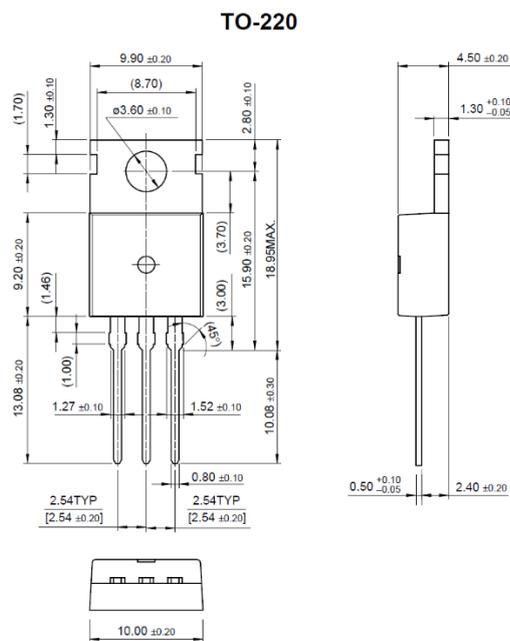
(Refer to test circuit ,0°C < T<sub>J</sub> < 125°C, I<sub>O</sub> = 500mA, V<sub>I</sub> = 10V, C<sub>I</sub>= 0.33μF, C<sub>O</sub>= 0.1μF, unless otherwise specified)

Parameter	Symbol	Conditions	MC7805/LM7805			Unit	
			Min.	Typ.	Max.		
Output Voltage	V <sub>O</sub>	T <sub>J</sub> = +25 °C	4.8	5.0	5.2	V	
		5.0mA ≤ I <sub>O</sub> ≤ 1.0A, P <sub>O</sub> ≤ 15W V <sub>I</sub> = 7V to 20V	4.75	5.0	5.25		
Line Regulation (Note 1)	Regline	T <sub>J</sub> = +25 °C	V <sub>O</sub> = 7V to 25V	-	4.0	100	mV
			V <sub>I</sub> = 8V to 12V	-	1.6	50	
Load Regulation (Note 1)	Regload	T <sub>J</sub> = +25 °C	I <sub>O</sub> = 5.0mA to 1.5A	-	9	100	mV
			I <sub>O</sub> = 250mA to 750mA	-	4	50	
Quiescent Current	I <sub>Q</sub>	T <sub>J</sub> = +25 °C	-	5.0	8.0	mA	
Quiescent Current Change	ΔI <sub>Q</sub>	I <sub>O</sub> = 5mA to 1.0A	-	0.03	0.5	mA	
		V <sub>I</sub> = 7V to 25V	-	0.3	1.3		
Output Voltage Drift	ΔV <sub>O</sub> /ΔT	I <sub>O</sub> = 5mA	-	-0.8	-	mV/°C	
Output Noise Voltage	V <sub>N</sub>	f = 10Hz to 100KHz, T <sub>A</sub> = +25 °C	-	42	-	μV/V <sub>O</sub>	
Ripple Rejection	RR	f = 120Hz V <sub>O</sub> = 8V to 18V	62	73	-	dB	
Dropout Voltage	V <sub>Drop</sub>	I <sub>O</sub> = 1A, T <sub>J</sub> = +25 °C	-	2	-	V	
Output Resistance	r <sub>O</sub>	f = 1KHz	-	15	-	mΩ	
Short Circuit Current	I <sub>SC</sub>	V <sub>I</sub> = 35V, T <sub>A</sub> = +25 °C	-	230	-	mA	
Peak Current	I <sub>PK</sub>	T <sub>J</sub> = +25 °C	-	2.2	-	A	

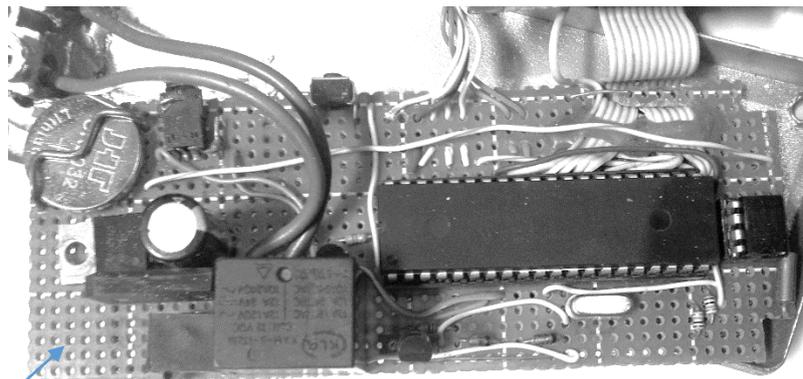
#### Note:

1. Load and line regulation are specified at constant junction temperature. Changes in V<sub>O</sub> due to heating effects must be taken into account separately. Pulse testing with low duty is used.

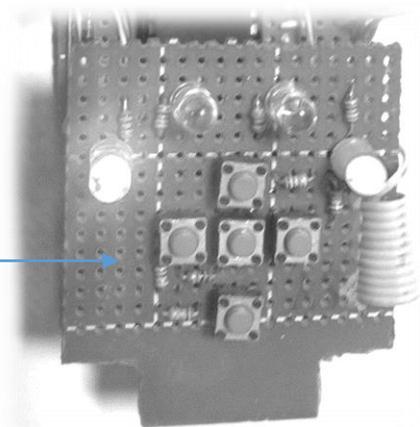
### ➤ Dimensions physiques



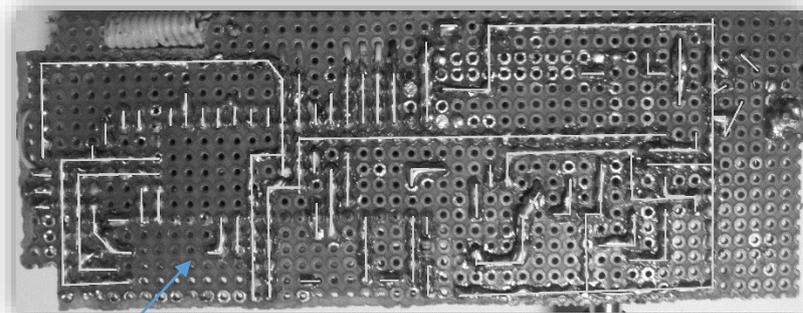
## 2- Carte de commande



La carte de commande sur plaque d'essai perforée. "Côté composant"

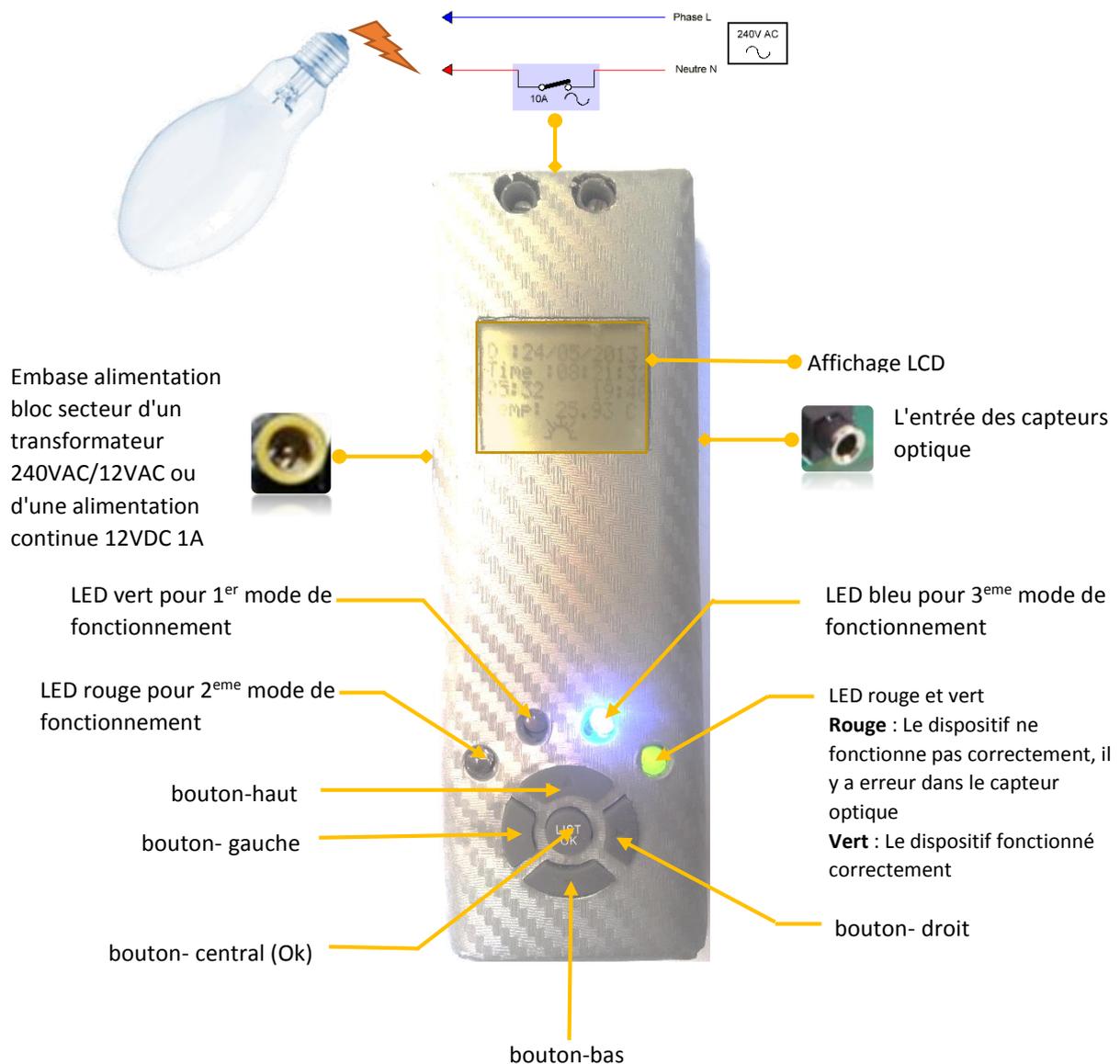


Le clavier (les boutons poussoirs et les LEDs) sur plaque d'essai perforée. "Côté composant"



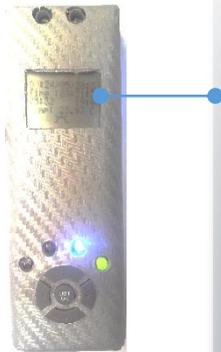
La carte de commande sur plaque d'essai perforée. "Côté cuivre"

### 3- La forme et le design du dispositif final avec la description



- bouton-haut : Utilisé pour changer les valeurs saisies (augmenter).
- bouton-bas : Utilisé pour changer les valeurs saisies (diminuer).
- bouton-gauche : Utilisé pour naviguer entre les valeurs dans la page des paramètres.
- bouton- droit : Utilisé pour naviguer entre les valeurs dans la page des paramètres.
- bouton- droit + bouton-gauche : Utilisés sur l'accès à la page des paramètres ou retour à la page d'accueil.
- bouton- haut + bouton-bas : Utilisés pour l'accès à la page « À propos de l'appareil ».
- bouton- central (Ok) : Utilisé pour choisir le mode de fonctionnement de l'appareil de la page d'accueil ou enregistrer les modifications de la page des paramètres.

➤ **Affichage LCD**



**La page d'accueil**  
*Informations*

La date → D : 24/05/2013  
 L'heure → Time : 08 :22 :32  
 Heure de lever du soleil → 05 :32 19 :40 ← Heure de coucher du soleil  
 Temp : 26 .51 C ← Température en degrés Celsius

☀

- ☀ Jour / Le ciel nuageux
- ☀ Jour
- ☀ Nuit
- EE Il y a erreur dans le capteur optique

**Page des paramètres**  
*Réglage*

Jour/mois/année → 24/05/2013  
 Heures : minutes : seconde → 08 :22 :32  
 Latitude(N : Nord, S : Sud) → 34.85N 005.73E ← Longitude(E : Est, W : Ouest)  
 LOCAL +01 ← +/- Fuseau horaire  
 ☀ 10% ← La sensibilité des capteurs optiques

☀ Jour  
 ☀ Nuit

**Remarque :** longitude et latitude en degrés décimaux  
 Exemple : longitude de la ville de Biskra est 5°43'59"E  
 (5 degrés, 43 minutes et 59 secondes Est)  
 En degrés décimaux  $5 + (43 / 60) + (59 / 3600) = 5.7333$