

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



## **Mémoire de Fin d'Étude**

Présenté à

**L'Université Echahid Hamma Lakhdar d'El Oued**

Faculté de Technologie

Département de Génie Electrique

En vue de l'obtention du diplôme de

## **MASTER ACADEMIQUE**

En Télécommunications

Présenté par

**AHMID Djelloul et HAMMADI Bachir**

# **Thème**

**Carte d'acquisition de données en utilisant le  
microcontrôleur PIC16F876A**

Soutenu le 26/05/2016. Devant le jury composé de :

Mr. TOUHAMI Redha

MAA

Président

Mr. HIMA Abdelkader

MAA

Rapporteur

Mr. AJGOU Riadh

MCB

Examineur

**Année Universitaire 2015/2016**

# *DEDICACES*

*je dédie ce travail à:*

*Ma Mère, Mon Père.*

*Mon épouse.*

*Mes enfants.*

*Mes sœurs.*

*Mes frères.*

*Mes amis.*

*Mes professeurs.*

*AHMID Djelloul*

# *DEDICACES*

*Ce travail est spécialement dédié:*

- ✓ *Aux mémoires de mon père Mosbah, ma mère Ourida et mon frère Djilani "que dieu vous accueille à son paradis".*
- ✓ *A ma femme Farida pour la patience et les sacrifices durant cette période de formation.*
- ✓ *A mes chères enfants Oualid, Imadeddine, Salaheddine, Djoumana et Moudjiberrahmane.*
- ✓ *A mes frères Ali, Smail, Djamel et mes sœurs Nadjat et Hania pour leurs soutien.*
- ✓ *A tous mes chers amis.*
- ✓ *A tous les professeurs du département de Génie Electrique de l'université Echahid Hamma Lakhdar d'Eloued.*

*HAMMADI Bachír*

# *Remercîments*

*Nous tenons à exprimer notre profonde reconnaissance et respect à notre encadreur **HIMA Abdelkader** pour son soutien.*

*Nous exprimons nos profondes et respectueuses gratitudees à nos professeurs: **AJGOU Riadh, BOULILA Mohammed, LAKHDAR Nacereddine, MEDJOURI Abdelkader** pour leurs encouragements.*

*Nos vifs et sincères remerciements sont adressés aux membres de jury.*

*Nous remercions également les professeurs: **SALEMI Saïd, HETIRI Messaoud, TIR Zouhir** et **CHEMSA Ali** et tous les professeurs du département de Génie Electrique de l'université Echahid Hamma Lakhdar d'Eloued.*

## ملخص:

يُندرج مشروعنا ضمن الأنظمة المدمجة و هي أنظمة الكترونية ومعلوماتية ذاتية التسيير. يتضمن هذا المشروع انجاز بطاقة تحصيل البيانات باستخدام أجهزة الاستشعار (درجة الحرارة، ضوء)، حيث يتم استقبال هذه البيانات وتحويلها بواسطة وحدة التحويل التماثلي/الرقمي للتحكم المجهرى الذي يعالج ثم يعرض النتائج على شاشة LCD و كذلك يرسلها الى الحاسوب عبر المنفذ التسلسلي. يعرض الحاسوب هذه القيم على شكل رسومات بيانية. تستعمل مرحلات في هذه البطاقة يمكن التحكم فيها إما عن طريق المتحكم المجهرى أو عن طريق الكمبيوتر. صمم برنامجين لتسيير المتحكم المجهرى و التواصل مع جهاز الكمبيوتر.

**الكلمات المفتاحية:** بطاقة التحصيل، 16F876A، RS232 PIC، أجهزة الاستشعار، برنامج التواصل.

## Abstract:

Our project is within the domain of embedded system which is defined as an electronic and computer system that is autonomous. It consists in realizing a data acquisition board using sensors (temperature, light), the results are read by an analog-digital converter of the microcontroller which processes the values and displays the results on an LCD display and transmit these values to a computer via serial port. PC presents these values in the form of a graph in real time.

Control relays are used in the card and can be controlled either by the PIC or by the control computer. Two programs are implemented for the management of microcontroller and interfacing with the computer.

**Key words:** Acquisition card, 16F876A, RS232, Sensor, interfacing program.

## Résumé:

Un système embarqué est défini comme un système électronique et informatique autonome. Notre Project touche ce Domain. Il consiste à la réalisation d'une carte d'acquisition de données en utilisant des capteurs (température, lumière) dont les résultats sont lus par un convertisseur analogique numérique du microcontrôleur qui traite les valeurs et affiche les résultats sur un afficheur LCD et les envoie vers un micro-ordinateur par le port série. Le PC présente ces valeurs sous forme d'un graphe à temps réel.

Des relais de commande sont utilisés dans la carte et peuvent être contrôlés soit par le PIC soit par l'ordinateur de pilotage. Deux programmes sont réalisés pour la gestion du microcontrôleur et l'interfaçage avec le micro-ordinateur.

**Mots clés :** Carte d'acquisition, PIC 16F876A, RS232, Capteur, Programme d'interfaçage.

## ***Liste des figures***

|                                                                                                |    |
|------------------------------------------------------------------------------------------------|----|
| <b>Figure I.1:</b> Principe d'un capteur.....                                                  | 2  |
| <b>Figure I.2:</b> L'erreur de zéro (offset).....                                              | 4  |
| <b>Figure I.3:</b> L'erreur d'échelle (gain).....                                              | 5  |
| <b>Figure I.4:</b> L'erreur de linéarité.....                                                  | 5  |
| <b>Figure I.5:</b> L'erreur due au phénomène d'hystérésis.....                                 | 5  |
| <b>Figure I.6:</b> L'erreur de mobilité.....                                                   | 6  |
| <b>Figure I.7:</b> LM35.....                                                                   | 6  |
| <b>Figure I.8:</b> Brochage du circuit intégré LM35.....                                       | 7  |
| <b>Figure I.9:</b> Alimentation simple du LM35.....                                            | 8  |
| <b>Figure I.10:</b> LDR.....                                                                   | 8  |
| <b>Figure I.11:</b> Connexion du LDR.....                                                      | 9  |
| <b>Figure I.12:</b> La courbe de sensibilité.....                                              | 9  |
| <b>Figure II.1:</b> Structure interne d'un microcontrôleur.....                                | 11 |
| <b>Figure II.2:</b> PIC 16F876A.....                                                           | 14 |
| <b>Figure II.3:</b> Architecture interne du PIC16F876A.....                                    | 15 |
| <b>Figure II.4:</b> Registre STATUS.....                                                       | 16 |
| <b>Figure II.5:</b> Brochage du 16F876A.....                                                   | 18 |
| <b>Figure II.6:</b> Les Ports du PIC 16F876A.....                                              | 19 |
| <b>Figure II.7:</b> Plan mémoire du PIC16F876A.....                                            | 20 |
| <b>Figure II.8:</b> Le module de conversion A/N.....                                           | 21 |
| <b>Figure II.9:</b> Le port en transmission.....                                               | 24 |
| <b>Figure II.10:</b> Illustration de la transmission de deux octet successifs D0 et D1.....    | 25 |
| <b>Figure II.11:</b> Le port en réception.....                                                 | 26 |
| <b>Figure II.12 :</b> Schéma fonctionnel d'une liaison série asynchrone de la norme RS232..... | 30 |

|                                                                                         |    |
|-----------------------------------------------------------------------------------------|----|
| <b>Figure II.13:</b> la liaison série croisée.....                                      | 30 |
| <b>Figure II.14:</b> Exemple de transmission série.....                                 | 31 |
| <b>Figure II.15:</b> Le MAX 233.....                                                    | 32 |
| <b>Figure II.16:</b> La connexion entre un max233 et un Pic.....                        | 33 |
| <b>Figure II.17:</b> Afficheur LCD.....                                                 | 34 |
| <b>Figure II.18:</b> Brochages du LCD.....                                              | 35 |
| <b>Figure II.19:</b> Diagramme d'écriture et de lecture.....                            | 39 |
| <b>Figure II.20:</b> Le relais.....                                                     | 39 |
| <b>Figure II.21:</b> Connexion du relais.....                                           | 40 |
| <b>Figure II.22 :</b> Transistor bipolaire de type NPN.....                             | 40 |
| <b>Figure II.23:</b> Le quartz.....                                                     | 41 |
| <b>Figure III.1:</b> Interface utilisateur ISIS.....                                    | 42 |
| <b>Figure III.2:</b> Organisation de la boîte à outils ISIS.....                        | 43 |
| <b>Figure III.3:</b> L'environnement <i>IDE</i> du compilateur <i>mikro C PRO</i> ..... | 44 |
| <b>Figure III.4:</b> la fenêtre mikroC terminal.....                                    | 45 |
| <b>Figure III.5:</b> Fenêtre de propriétés du port.....                                 | 46 |
| <b>Figure III.6:</b> Fenetre du programme réalisé avec C#.....                          | 47 |
| <b>Figure III.7:</b> Fenêtre de sélection de port.....                                  | 48 |
| <b>Figure III.8:</b> Fenêtre de sélection d'équipement .....                            | 48 |
| <b>Figure III.9:</b> La fenêtre "Description de la connexion".....                      | 49 |
| <b>Figure III.10:</b> La fenêtre "Connexion".....                                       | 49 |
| <b>Figure III.11:</b> Fenêtre de paramètres du port COM.....                            | 50 |
| <b>Figure III.12:</b> Le programmeur LEAPER-48.....                                     | 50 |
| <b>Figure III.13:</b> Universal IC Writer II.....                                       | 51 |
| <b>Figure IV.1:</b> La carte d'acquisition de données.....                              | 52 |
| <b>Figure IV.2:</b> Schéma synoptique.....                                              | 53 |
| <b>Figure IV.3:</b> schéma électrique de capteurs.....                                  | 53 |

|                                                                      |    |
|----------------------------------------------------------------------|----|
| <b>Figure IV.4:</b> schéma électrique de l'afficheur LCD .....       | 54 |
| <b>Figure IV.5:</b> schéma électrique du port COM.....               | 54 |
| <b>Figure IV.6:</b> schéma électrique de relais.....                 | 55 |
| <b>Figure IV.7:</b> schéma électrique de la carte d'acquisition..... | 56 |
| <b>Figure IV.8:</b> L'afficheur LCD.....                             | 60 |
| <b>Figure IV.9:</b> Le programme d'interfaçage.....                  | 60 |
| <b>Figure IV.10:</b> Le graphe de la température.....                | 61 |
| <b>Figure IV.11:</b> Le graphe de la lumière.....                    | 61 |



## ***Liste des abréviations***

- **PC:** Personal Computer
- **PIC:** Peripheral Interface Controller
- **LCD:** Liquid Crystal Display
- **UART:** Universal Asynchronous Receiver/Transmitter
- **USART:** Universal Synchronous/Asynchronous Receiver/Transmitter
- **ADC:** Analog-to-Digital Converter
- **LDR:** Light Dependent Resistor
- **RS232:** A standard for serial communication transmission of data
- **RXD :** Received Data
- **TXD :** Transmitted Data
- **RAM:** Random Access Memory
- **EEPROM:** Electrically Erasable Programmable Read-Only Memory
- **ALU:** Arithmetic Logic Unit
- **I2C:** Inter-Integrated Circuit
- **MCLR:** Master Clear
- **TTL:** Transistor Transistor Logic
- **CG RAM:** Character Generator RAM
- **DD RAM:** Double Data RAM
- **NPN:** Negative Positive Negative Transistor.
- **USB:** Universal Serial Bus
- **COM:** Communication Port

# ***Sommaire***

Dédicaces

Remerciements

Résumé.....I

Liste de figures.....II

Liste des abréviations.....V

Introduction générale.....1

## ***Chapitre I: Les capteurs***

I.1 Introduction.....2

I.2 Types de capteurs.....2

I.2.1 Capteurs actifs.....2

I.2.2 Capteurs passifs.....3

I.3 Caractéristiques métrologiques d'un capteur.....3

I.3.1 Etendue de mesure.....3

I.3.2 Sensibilité.....3

I.3.3 Précision.....3

I.3.4 Fidélité et justesse.....3

I.3.5 Rapidité.....4

I.3.6 Stabilité.....4

I.3.7 Répétabilité et Reproductibilité.....4

I.3.8 Bruit de fond.....4

I.3.9 Erreur de mesure.....4

I.4 Choix d'un capteur.....6

I.5 Le capteur de température LM35.....6

I.5.1 Brochage du circuit intégré LM35.....7

I.6 La photorésistance LDR.....8

I.6.1 Données techniques.....9

## ***Chapitre II: Les microcontrôleurs et les composants périphériques***

II.1 Introduction.....10

II.2 Les microcontrôleurs.....10

II.2.1 Les avantages des microcontrôleurs.....10

II.2.2 La composition d'un microcontrôleur.....11

II.3 Le PIC.....12

II.3.1 Structure d'un PIC.....12

II.3.2 Structure minimale d'un PIC.....12

II.3.3 Identification d'un PIC.....13

II.3.4 Le choix du microcontrôleur.....14

II.4 Le microcontrôleur PIC 16F876A.....14

II.4.1 Architecture interne du 16f876A.....15

II.4.2 Unité Arithmétique et logique ALU.....16

II.4.3 Registre STATUS.....16

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| II.4.4 Description des autres éléments.....                             | 16 |
| II.4.5 Brochage du PIC 16F876A.....                                     | 18 |
| II.4.6 Plan Mémoires.....                                               | 19 |
| II.4.6.1 Plan Mémoire pour les instructions.....                        | 20 |
| II.4.6.2 Plan mémoire des données et des registres internes.....        | 20 |
| II.4.7 Le module de conversion A/N.....                                 | 21 |
| II.4.7.1 Le registre ADCON0.....                                        | 21 |
| II.4.7.2 Le registre ADCON1.....                                        | 22 |
| II.4.7.3 Déroulement d'une Conversion.....                              | 22 |
| II.4.8 L'USART.....                                                     | 23 |
| II.4.8.1 Mode Asynchrone.....                                           | 23 |
| II.4.8.2 Le port en transmission.....                                   | 24 |
| II.4.8.3 Déroulement de la transmission.....                            | 25 |
| II.4.8.4 Les étapes de transmission (sans interruption, mode 8 bits)... | 26 |
| II.4.8.5 Le port en réception.....                                      | 26 |
| II.4.8.6 Les étapes de réception (sans interruption, mode 8 bits).....  | 27 |
| II.4.9 Jeu d'instructions.....                                          | 28 |
| II.5 La gestion de la liaison RS232 .....                               | 30 |
| II.5.1 Communication série asynchrone à travers le port série RS232...  | 30 |
| II.5.2 Protocole de transmission .....                                  | 30 |
| II.5.3 La conversion des niveaux.....                                   | 32 |
| II.5.4 L'émission.....                                                  | 33 |
| II.5.5 La réception.....                                                | 34 |
| II.6 Les afficheurs à cristaux liquides LCD.....                        | 34 |
| II.6.1 Principe des cristaux liquides.....                              | 34 |
| II.6.2 Brochages.....                                                   | 35 |
| II.6.3 La mémoire.....                                                  | 36 |
| II.6.3.1 La mémoire d'affichage (DD RAM).....                           | 36 |
| II.6.3.2 La mémoire du générateur de caractères (CG RAM).....           | 36 |
| II.6.4 Commande d'un afficheur LCD.....                                 | 37 |
| II.6.4.1 Mode 8 bits.....                                               | 37 |
| II.6.4.2 Mode 4 bits.....                                               | 38 |
| II.7 Le relais.....                                                     | 39 |
| II.8 Le transistor bipolaire NPN en commutation.....                    | 40 |
| II.9 Les quartz.....                                                    | 41 |

## ***Chapitre III: Logiciels utilisés***

|                                                 |    |
|-------------------------------------------------|----|
| III.1 Introduction.....                         | 42 |
| III.2 ISIS Proteus.....                         | 42 |
| III.2.1 L'éditeur de schéma ISIS .....          | 42 |
| III.2.2 Interface utilisateur .....             | 42 |
| III.2.3 Fenêtre d'ensemble .....                | 43 |
| III.2.4 Fenêtre d'édition .....                 | 43 |
| III.2.5 La boîte à outils .....                 | 43 |
| III.2.6 Coordonnées du curseur .....            | 43 |
| III.2.7 Organisation de la boîte à outils ..... | 43 |
| III.3 MIKRO C.....                              | 44 |
| III.3.1 Editeur de code .....                   | 44 |

|                                        |    |
|----------------------------------------|----|
| III.3.2 Assistant de code.....         | 45 |
| III.4 Le Visual C# .....               | 46 |
| III.5 Virtual Serial Port Driver ..... | 47 |
| III.6 Logiciel HyperTerminal .....     | 49 |
| III.7 Le programmeur de PIC.....       | 50 |

## ***Chapitre IV: Conception et réalisation de la carte***

|                                                                                        |    |
|----------------------------------------------------------------------------------------|----|
| IV.1 Introduction.....                                                                 | 52 |
| IV.2 Schéma synoptique de la carte d'acquisition.....                                  | 53 |
| IV.3 Schéma électrique de la carte d'acquisition.....                                  | 56 |
| IV.4 Liste des composants.....                                                         | 56 |
| IV.5 Programmation.....                                                                | 57 |
| IV.5.1 Organigramme du programme pilote de la carte<br>d'acquisition côté PIC .....    | 58 |
| IV.5.2 Organigramme du programme pilote de la carte<br>d'acquisition côté PC(C#) ..... | 59 |
| IV.5.3 Organigramme du sous-programme de commande<br>des relais.....                   | 60 |
| IV.6 Résultats de tests .....                                                          | 61 |
| Conclusion générale.....                                                               | 64 |
| Bibliographie                                                                          |    |
| Annexes:                                                                               |    |
| PIC 16F876A.....                                                                       | i  |
| MAX233A.....                                                                           | v  |
| Afficheur LCD.....                                                                     | vi |

# ***Introduction générale***

## ***Introduction générale***

L'acquisition de données est une étape très importante dans les processus d'industrie. Cette collecte d'information permet de déterminer les niveaux critiques de répondre d'une façon appropriée à contrôler des appareils, des instruments...

L'exploitation des signaux issus des capteurs est l'objet de notre travail, c'est un projet pour la réalisation d'une carte permettant l'acquisition de données sur plusieurs voies et qui puisse fonctionner sur PC.

Cette carte à base d'un microcontrôleur (PIC 16F876A) est destinée pour un usage industriel. Elle acquise les données issues des capteurs et les transmette à l'ordinateur via une communication série RS-232C. Au côté ordinateur, un programme est conçu permettant :

- L'acquisition de la température et l'intensité de la lumière ou autres valeurs physiques.
- Visualisation de ces valeurs sur un afficheur LCD.
- L'affichage de ces valeurs est présenté sous forme d'un graphe qui illustre les changements dans le temps.
- La commande des relais afin d'activer l'éclairage, ventilation, chauffage...

Dans le premier chapitre, on a présenté les principes des capteurs en général et on a concentré sur le capteur de température LM35 et le capteur de lumière LDR.

Le deuxième chapitre est consacré à l'étude des microcontrôleurs (PIC16F876A) et les composants périphériques comme l'afficheur LCD, le relais...

Les logiciels utilisés pour la conception et la réalisation de notre carte sont présentés dans le troisième chapitre.

Enfin, le quatrième chapitre est destiné à montrer les différentes étapes de conception et de réalisation de cette carte ainsi que la présentation des résultats des tests.

**Chapitre I:**  
**Les capteurs**

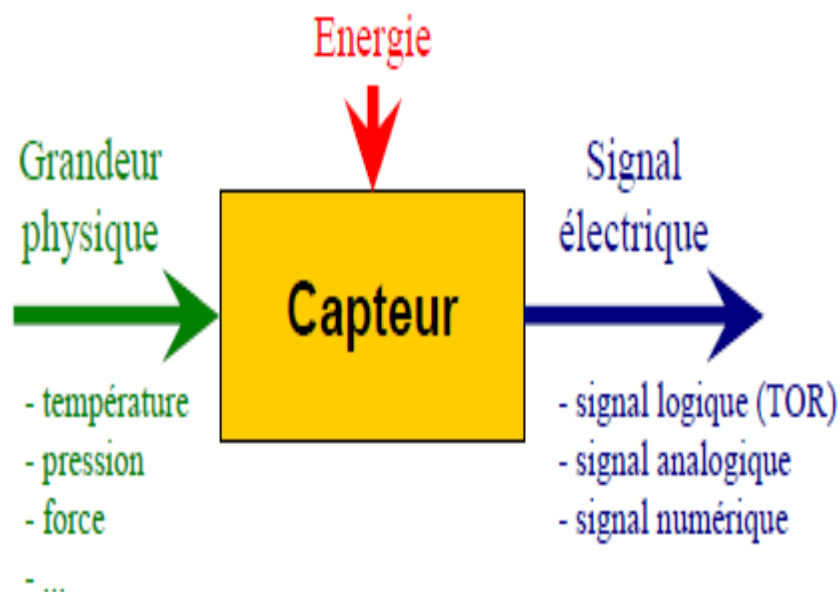
## Chapitre I

### *Les capteurs*

#### **I.1 Introduction :**

Dans des nombreux domaines, on a besoin de contrôler plusieurs paramètres physiques (température, luminosité, force, position, vitesse, ...). Le capteur est l'élément indispensable à la mesure de ces grandeurs physiques.

Un capteur est un organe de prélèvement d'information qui élabore à partir d'une grandeur physique, une autre grandeur physique de nature différente (très souvent électrique). Cette grandeur représentative de la grandeur prélevée est utilisable pour de mesure ou de commande (Voir Figure I.1). [1]



**Figure I.1:** Principe d'un capteur [1]

#### **I.2 Types de capteurs :**

##### **I.2.1 Capteurs actifs :**

Fonctionnant en générateur, un capteur actif est généralement fondé dans son principe sur un effet physique qui assure la conversion en énergie électrique de la forme d'énergie propre à la grandeur physique à prélever, énergie thermique, mécanique ou de rayonnement. [1]



**I.2.2 Capteurs passifs :**

Il s'agit généralement d'impédance dont l'un des paramètres déterminants est sensible à la grandeur mesurée. La variation d'impédance résulte :

- Soit d'une variation de dimension du capteur, c'est le principe de fonctionnement d'un grand nombre de capteur de position, potentiomètre, inductance à noyaux mobile, condensateur à armature mobile.
- Soit d'une déformation résultant de force ou de grandeur s'y ramenant, pression accélération (armature de condensateur soumise à une différence de pression, jauge d'extensomètre liée à une structure déformable). [1]

**I.3 Caractéristiques métrologiques d'un capteur**

Les liens entre un capteur et la grandeur qu'il mesure sont définis par ses caractéristiques d'emploi.

**I.3.1 Etendue de mesure :**

Domaine de mesure pour lequel les indications du capteur ne doivent pas être entachées d'une erreur supérieure à l'erreur maximale tolérée. On appelle les valeurs limites du domaine, « portée minimale » et « portée maximale ». [2]

**I.3.2 Sensibilité :**

C'est le rapport de la variation du signal de sortie à la variation correspondante de la grandeur à mesurer. [2]

**I.3.3 Précision :**

C'est l'aptitude du capteur à donner des indications proches de la valeur vraie de la grandeur mesurée. [2]

**I.3.4 Fidélité et justesse :**

La fidélité est la qualité d'un capteur à fournir des indications identiques pour une même valeur de la grandeur à mesurer.

La justesse est la qualité d'un capteur à fournir des indications précises. [2]

**I.3.5 Rapidité :**

C'est l'aptitude du capteur à suivre dans le temps les variations de la grandeur à mesurer. Il faut donc tenir compte du temps de réponse, de la bande passante et la fréquence de coupure du capteur. [2]

**I.3.6 Stabilité :**

La stabilité qualifie la capacité d'un capteur à conserver ses performances pendant une longue durée (problème de dérive du zéro par exemple). [2]

**I.3.7 Répétabilité et Reproductibilité :**

La répétabilité est l'étroitesse de l'accord entre les résultats de mesures successifs d'une même grandeur effectuée avec la même méthode, par le même observateur, avec les mêmes instruments de mesure et à des intervalles de temps assez courts.

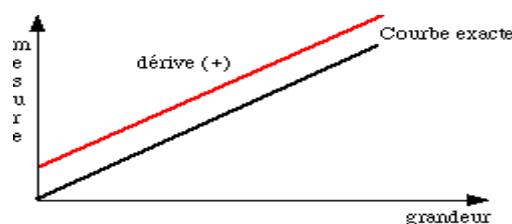
La reproductibilité est l'étroitesse de l'accord entre les résultats de mesures successifs d'une même grandeur dans le cas où les mesures sont effectuées dans les conditions différentes que pour la répétabilité. [2]

**I.3.8 Bruit de fond :**

C'est une variation parasite, souvent aléatoire, du signal de sortie, dont la valeur moyenne est nulle et qui vient se superposer à la valeur à mesurer. [2]

**I.3.9 Erreur de mesure :**

- L'erreur de zéro (offset) (Figure I.2) [1]



**Figure I.2:** L'erreur de zéro (offset)

- L'erreur d'échelle (gain)

C'est une erreur qui dépend de façon linéaire de la grandeur mesurée (Voir Figure I.3). [1]

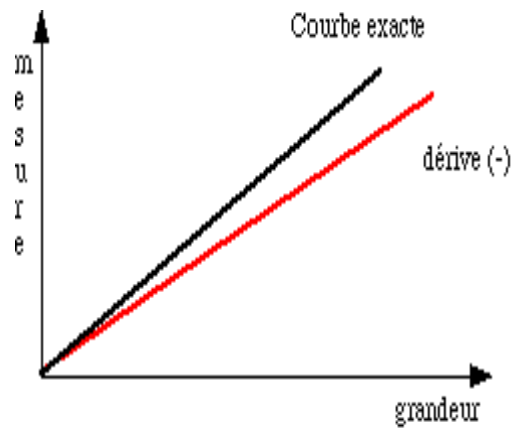


Figure I.3: L'erreur d'échelle (gain)

- L'erreur de linéarité

La caractéristique n'est pas une droite (Voir Figure I.4). [1]

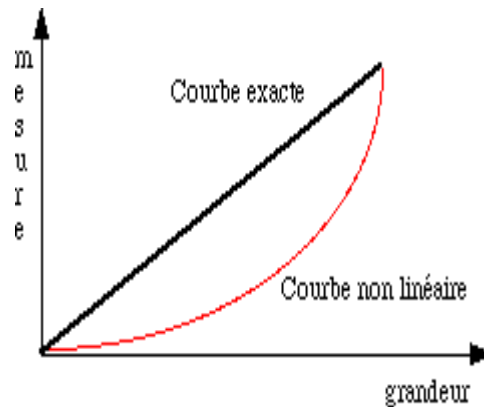


Figure I.4: L'erreur de linéarité

- L'erreur due au phénomène d'hystérésis

Il y a phénomène d'hystérésis lorsque le résultat de la mesure dépend de la précédente mesure (Voir Figure I.5). [1]

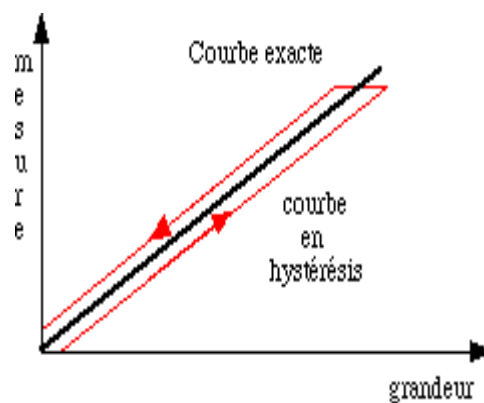
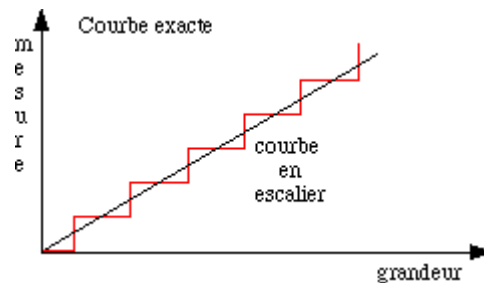


Figure I.5: L'erreur due au phénomène d'hystérésis

- L'erreur de mobilité

La caractéristique est en escalier, cette erreur est souvent due à une numérisation du signal (Voir Figure I.6). [1]



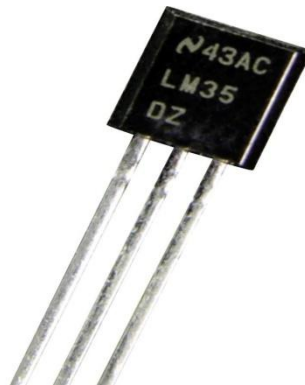
**Figure I.6:** L'erreur de mobilité

#### **I.4 Choix d'un capteur :**

Le choix d'un capteur approprié s'effectue en vérifiant que ses caractéristiques métrologiques sont compatibles avec les conditions imposées par le cahier des charges. Ces conditions sont de deux types :

- Conditions sur la mesurande
- Conditions sur l'environnement de mesure. [2]

#### **I.5 Le capteur de température LM35 :**



**Figure I.7:** LM35

La tension aux bornes d'un semi-conducteur ainsi que le courant qui le traverse dépendent de la température. Ce sont des capteurs dit intégrés et qui ont l'avantage qu'à courant constant  $I$ , la mesure de  $V$  est linéaire en fonction de la température.

Ce type de capteur utilisant des diodes et des transistors ont l'avantage d'être simple à fabriquer et à mettre en œuvre, peu coûteux et très linéaire.

Le capteur de température LM35 (Voir Figure I.7) est un capteur analogique de température qui est précis, peu coûteux, très simple à utiliser et d'une fiabilité à toute épreuve.

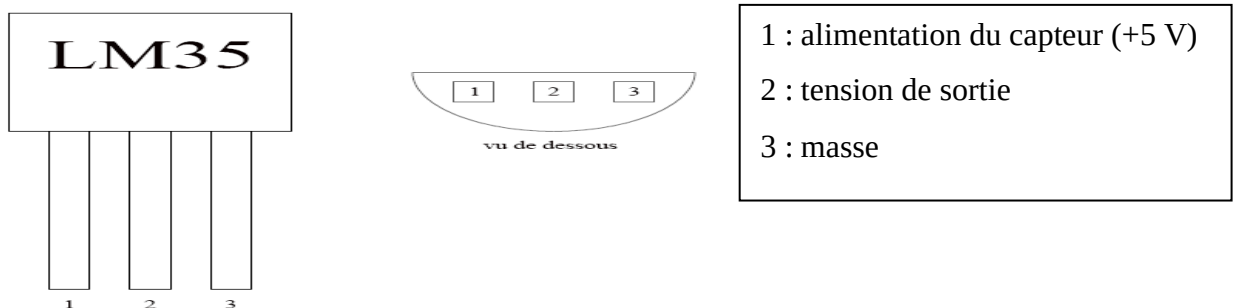
Le capteur de température LM35 est capable de mesurer des températures allant de  $-55^{\circ}\text{C}$  à  $+150^{\circ}\text{C}$  dans sa version la plus précise et avec le montage adéquat.

Les versions grandes publiques sont capables de mesurer des températures comprises entre  $-40^{\circ}\text{C}$  et  $+110^{\circ}\text{C}$ .

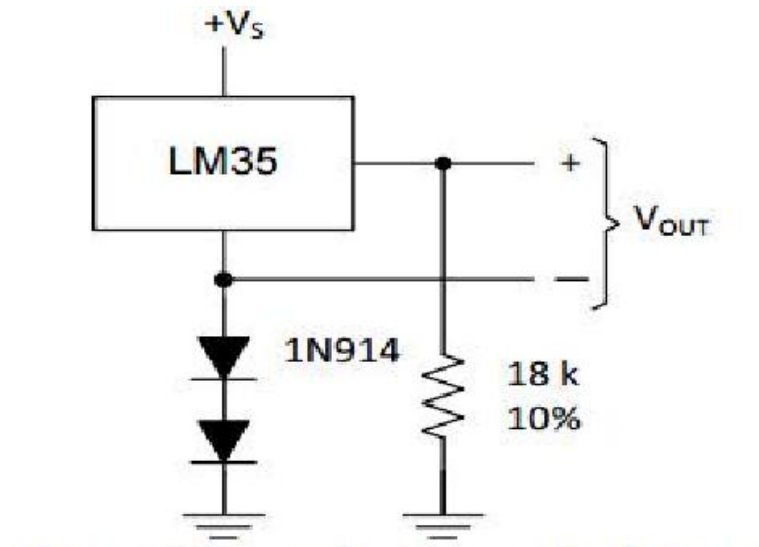
La sortie analogique du capteur est proportionnelle à la température. Il suffit de mesurer la tension en sortie du capteur pour en déduire la température. Chaque degré Celsius correspond à une tension de  $+10\text{mV}$ . L'avantage du capteur LM35 est son pré calibration en sortie d'usine. Tous les capteurs LM35 sont calibrés en degré Celsius lors de la fabrication. La précision garantie par le fabricant est de  $\pm 1^{\circ}\text{C}$  à  $25^{\circ}\text{C}$  et  $\pm 1.5^{\circ}\text{C}$  à  $-55^{\circ}\text{C}$  ou  $+150^{\circ}\text{C}$  pour la version moins précise, ce qui est largement suffisant pour la plupart des applications. La version plus précise du LM35 (nommée "LM35A") a une précision garantie de  $\pm 0.5^{\circ}\text{C}$  à  $25^{\circ}\text{C}$  et  $\pm 1^{\circ}\text{C}$  à  $-55^{\circ}\text{C}$  ou  $+150^{\circ}\text{C}$ .

Une autre des grandes forces du capteur LM35, c'est sa linéarité exemplaire : moins de  $1^{\circ}\text{C}$  d'erreur sur la plage complète de  $-55^{\circ}\text{C}$  à  $+150^{\circ}\text{C}$ . Le capteur LM35 fonctionne avec n'importe quelle tension d'alimentation comprise entre 4 volts et 30 volts. Le capteur LM35 nécessite une alimentation négative pour mesurer des températures en dessous de  $0^{\circ}\text{C}$ . La sortie est proportionnelle à la température, par conséquent  $-10^{\circ}\text{C}$  est équivalent à  $-0,1$  volt. [3]

### I.5.1 Brochage du circuit intégré LM35 :



**Figure I.8:** Brochage du circuit intégré LM35 [3]



**Figure I.9:** Alimentation simple du LM35 [3]

### I.6 La photorésistance LDR :



**Figure I.10:** LDR

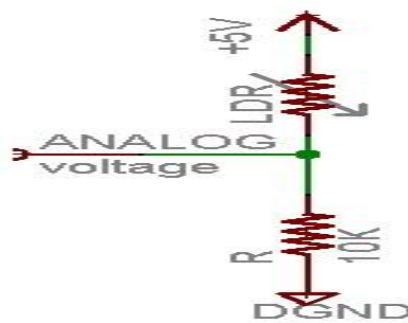
Les photorésistances sont des senseurs qui permettent de détecter la lumière. Elles sont petites, économiques en énergies, faciles à utiliser et ne s'usent pas. Elles sont souvent identifiées sous la dénomination CdS (parce qu'elles sont faites de Cadmium-Sulfite), LDR (pour *Light Dépendant Résistor* ce qui signifie Résistance dépendant de la lumière et Photorésistance (Voir Figure I.10). Fondamentalement, les photorésistances sont des résistances dont la valeur résistive (en ohms  $\Omega$ ) change en fonction de la quantité de lumière qui atteint le capteur. Elles sont abordables, existent sous de nombreux formats/tailles, disponibles sous de nombreuses spécifications (caractéristiques) mais sont très imprécises.

Chaque photorésistance agit un peu différemment d'une autre, même lorsqu'elles proviennent du même processus de fabrication. C'est pour cette raison qu'elles ne doivent pas être utilisées pour déterminer précisément le niveau lumineux en lux ou milli-candela. A la place, on est capable de détecter des variations de lumières élémentaires. [4]

**I.6.1 Données techniques :**

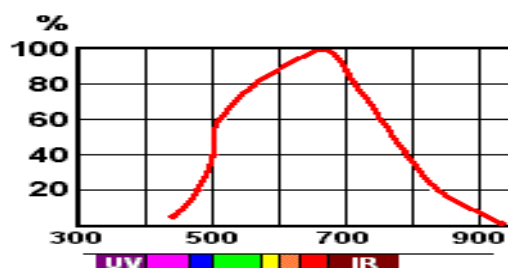
- Etendue de la variation de résistance : de 200K  $\Omega$  (noir) à 10K $\Omega$  (luminosité de 10 lux)
- Sensibilité : Les photorésistances répondent à des lumières de longueur d'onde variant entre 400nm (violet) et 600nm (orange), avec un pic à ~520nm (vert).
- Alimentation : Presque tout type de tension jusqu'à 100V, consomme moins de 1mA en moyenne (cela dépend de la tension d'alimentation utilisée)

La façon la plus facile de mesurer un senseur résistif est de connecter une borne sur l'alimentation et l'autre sur une résistance Pull-Down (Voir Figure I.11). On raccorde ensuite le point de connexion entre "la résistance et la photorésistance" sur une entrée analogique d'un microcontrôleur. [4]



**Figure I.11:** Connexion du LDR [4]

Le capteur LDR est une photorésistance, une cellule photo-électrique sensible à l'intensité lumineuse qu'elle traduit en produisant un petit courant électrique. Sensible à la quantité de lumière reçue. Une variation de lumière (ou d'ombre) provoque une variation de signal. La sensibilité du capteur LDR est proportionnelle à la lumière reçue : Plus le flux lumineux sera intense, plus le nombre d'électrons disponibles pour assurer la conduction. L'efficacité est en fonction de la longueur d'onde de la lumière. La courbe de sensibilité est voisine de celle de l'œil humain (Voir Figure I.12). [5]



**Figure I.12:** La courbe de sensibilité [5]

## **Chapitre II:**

### **Les microcontrôleurs et les** **composants périphériques**



## **Chapitre II**

# ***Les microcontrôleurs et les composants périphériques***

### **II.1 Introduction :**

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

Le microcontrôleur est implanté sur le système et relié aux différents capteurs et actionneurs avec éventuellement des circuits d'interfaçage appropriés. Une liaison avec un ordinateur permet de programmer le circuit et de le tester. Le microcontrôleur est un véritable petit ordinateur dans un module hybride. Outre le processeur, le module dispose d'une horloge, d'une mémoire de travail (RAM), d'une mémoire pour le programme non volatile (EEPROM), de lignes d'entrées sorties binaires et analogiques et d'un port de communication avec un ordinateur (PC). [6]

### **II.2 Les microcontrôleurs :**

Un microcontrôleur est une unité de traitement de l'information de type microprocesseur contenant tous les composants d'un système informatique, à savoir microprocesseur, des mémoires et des périphériques (ports, timers, convertisseurs...). Chaque fabricant a ses familles de microcontrôleur. Une famille se caractérise par un noyau commun (le microprocesseur, le jeu d'instruction...). [7]

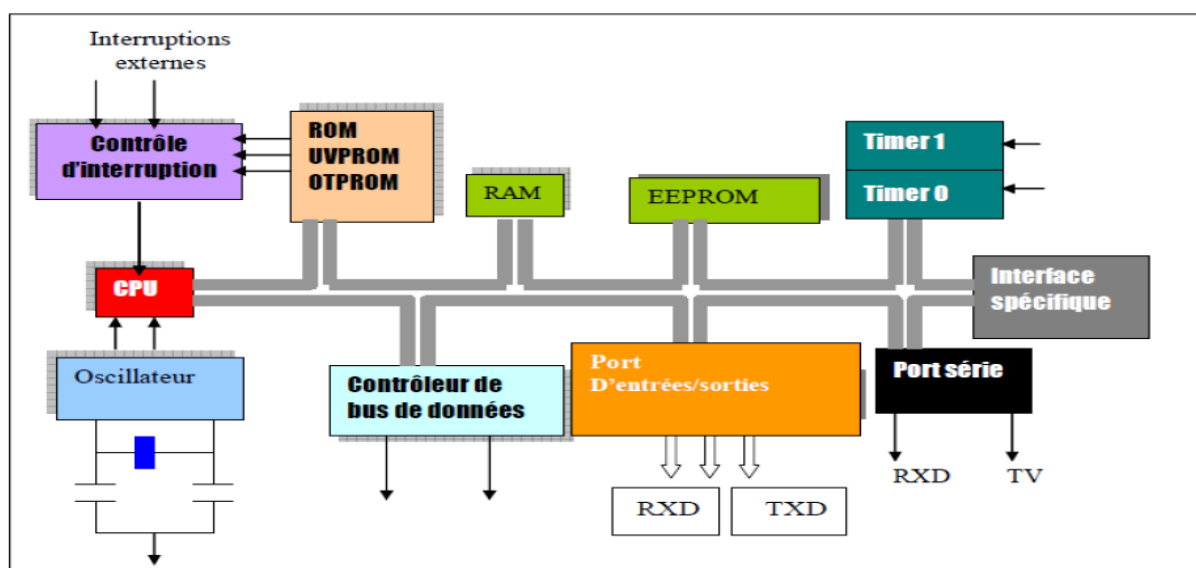
#### **II.2.1 Les avantages des microcontrôleurs : [7]**

L'utilisation des microcontrôleurs pour les circuits programmables a plusieurs points forts. Il suffit pour s'en persuader, d'examiner la spectaculaire évolution de l'offre des fabricants de circuits intégrés en ce domaine depuis quelques années. Nous allons voir que le nombre d'entre eux découle du simple sens.

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

### **II.2.2 La composition d'un microcontrôleur : [7]**

Un circuit microcontrôleur doit contenir dans un seul boîtier tous les éléments de base qu'on verra par la suite. En effet, pour l'analyse des divers systèmes réalisés avant l'avènement des microcontrôleurs, les fabricants des circuits intégrés ont affiné un peu la définition de ce qu'il fallait intégrer pour arriver à un schéma type analogue à la figure suivante (Voir Figure II.1) :



**Figure II.1:** Structure interne d'un microcontrôleur.

## **II.3 Le PIC : [7]**

### **II.3.1 Structure d'un PIC :**

Les PIC (Programmable Interface Contrôler), au même titre que les microprocesseurs, sont composés essentiellement de registres ayant chacun une fonction bien définie. Les PIC possèdent également des périphériques intégrés, tels qu'une mémoire EEPROM, un timer, des ports d'entrées/sorties ou bien encore un convertisseur analogique/numérique. Selon le type de PIC utilisé, on retrouvera en interne un certain nombre de registres et périphériques possédant des caractéristiques différentes. Les différences de caractéristiques selon le PIC utilisé sont :

- La taille de la RAM interne ;
- La mémoire EEPROM intégrée ;
- Le type de mémoire programme : FLASH, EPROM.
- Le timer intégré ;
- Les convertisseurs analogique/numérique intégrés.

### **II.3.2 Structure minimale d'un PIC :**

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

- Une mémoire de programme contient le code binaire correspondant aux instructions que doit exécuter le microcontrôleur. La capacité de cette mémoire est variable selon les PIC.
- Une mémoire RAM sauvegarde temporairement des données. Sa capacité est aussi variable selon les PIC.
- Une Unité Arithmétique et Logique (UAL ou ALU en anglais) est chargée d'effectuer toutes les opérations arithmétiques de base (addition, soustraction, etc.) ainsi que les opérations logiques de base (ET, OU logique, etc.).
- Des ports d'entrées/sorties permettent de dialoguer avec l'extérieur du microcontrôleur, par exemple pour prendre en compte l'état d'un interrupteur (entrée logique), ou encore pour commander un relais (sortie logique).

- Un registre compteur de programme (CP ou PC en anglais), est chargé de pointer l'adresse mémoire courante contenant l'instruction à réaliser par le microcontrôleur. Le contenu du registre PC évolue selon le pas de programme.
- Un registre pointeur de pile (PP ou SP en anglais) est essentiellement utilisé lorsqu'on réalise un sous-programme. Le pointeur de pile est chargé de mémoriser l'adresse courante que contient le compteur de programme avant le saut à l'adresse du sous-programme. Le pointeur restitue l'adresse sauvegardée vers le compteur de programme quand l'exécution du sous-programme est terminée.
- Un registre d'instruction contient tous les codes binaires correspondant aux instructions à réaliser par le microcontrôleur. Le PIC 16F876 comporte 35 instructions.
- Un registre d'état est en relation avec l'UAL et permet de tester le résultat de la dernière opération effectuée par le microcontrôleur. Selon la dernière opération effectuée, des bits sont positionnés dans le registre d'état et ceux-ci peuvent être testés à l'aide d'une instruction de branchement pour effectuer des sauts conditionnels.
- Une horloge système permet de cadencer tous les échanges internes ou externes au microcontrôleur.

La famille des PICs est subdivisée en 3 grandes familles : La famille Base-Line, qui utilise des mots d'instructions de 12 bits, la famille Mid-Range, qui utilise des mots de 14 bits, et la famille High-End, qui utilise des mots de 16 bits.

### **II.3.3 Identification d'un PIC :**

Pour identifier un PIC, nous utilisons simplement son numéro : les 2 premiers chiffres indiquent la catégorie du PIC, 16 indique un PIC Mid-Range, 18 indique PIC High-Range. Vient ensuite parfois une lettre L : Celle-ci indique que le PIC peut fonctionner avec une plage de tension beaucoup plus tolérante. Ensuite, nous trouvons : C indique que la mémoire programme est une EPROM ou plus rarement une EEPROM, CR pour indiquer une mémoire de type ROM ou F pour indiquer une mémoire de type FLASH. À ce niveau, on rappelle que seule une mémoire FLASH ou EEPROM est susceptible d'être effacée. Finalement nous trouvons sur les boîtiers le suffixe « -XX » dans lequel XX représente la fréquence d'horloge maximale que le PIC peut recevoir. Par

exemple –04 pour un 4MHz. Notons dès à présent que les PICs sont des composants STATIQUES, c'est à dire que la fréquence d'horloge peut être abaissée jusqu'à l'arrêt complet sans perte de données et sans dysfonctionnement.

#### **II.3.4 Le choix du microcontrôleur :**

Le choix du microcontrôleur est primordial car c'est de lui que dépendent en grande partie les performances, la facilité d'utilisation et le prix du montage. Le PIC 16F876A possède en plus des instructions très puissantes donc un programme à développer réduit, une programmation simple grâce au mode série. En fait la cause principale du choix du 16F876A est qu'il dispose de l'option du convertisseur A/D pour satisfaire côté acquisition.

#### **II.4 Le microcontrôleur PIC 16F876A :**



**Figure II.2: PIC 16F876A**

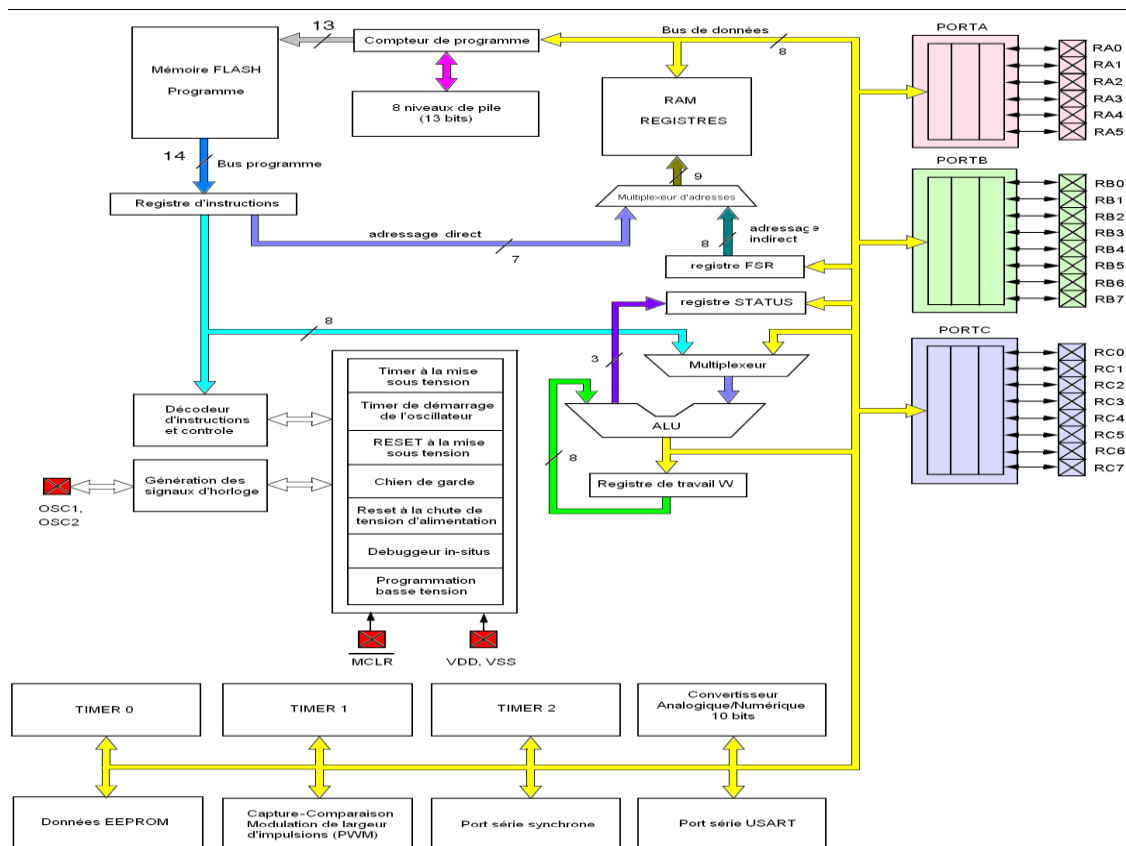
Nous allons maintenant s'intéresser à la structure interne du PIC 16F876A (Voir Figure II.2), avec lequel nous avons travaillé. Les éléments essentiels du PIC 16F876A sont : [8]

- Une mémoire programme de type EEPROM flash de 8K mots de 14 bits,
- Une RAM de donnée de 368 octets,
- Une mémoire EEPROM de 256 octets,
- Trois ports d'entrée sortie, A (6 bits), B (8 bits), C (8 bits),
- Convertisseur Analogiques numériques 10 bits a 5 canaux,
- USART, Port série universel, mode asynchrone (RS232) et mode synchrone
- SSP, Port série synchrone supportant I2C
- Trois TIMERS avec leurs Prescalers, TMR0, TMR1, TMR2

- Deux modules de comparaison et Capture CCP1 et CCP2
- Un chien de garde,
- 13 sources d'interruption,
- Générateur d'horloge, a quartz (jusqu' a 20 MHz) ou a Oscillateur RC
- Protection de code,
- Fonctionnement en mode sleep pour réduction de la consommation,
- Programmation par mode ICSP (In Circuit Serial Programming) 12V ou 5V,
- Possibilité aux applications utilisateur d'accéder à la mémoire programme,
- Tension de fonctionnement de 2 a 5V,
- Jeu de 35 instructions

### II.4.1 Architecture interne du 16f876A : [9]

(Voir Figure II.3)



**Figure II.3:** Architecture interne du PIC16F876A [9]

#### **II.4.2 Unité Arithmétique et logique ALU :**

Comme son nom l'indique, l'unité arithmétique et logique (ALU) effectue au sein du microcontrôleur PIC toutes les opérations arithmétiques et logiques dans un format de données 8 bits (addition, soustraction, opérations logiques, etc...). Ces opérations sont effectuées entre le registre de travail W et un opérande provenant d'un registre de la RAM. Les trois bits de poids faible du registre STATUS sont affectés par le résultat de l'opération effectuée par l'ALU. [9]

#### **II.4.3 Registre STATUS : [9]**



**Figure II.4:** Registre STATUS

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

C (Carry) : ce bit de retenue est mis à l'état logique "1" si l'opération effectuée entraîne un résultat dont le format est supérieur à 8 bits. (Voir Figure II.4)

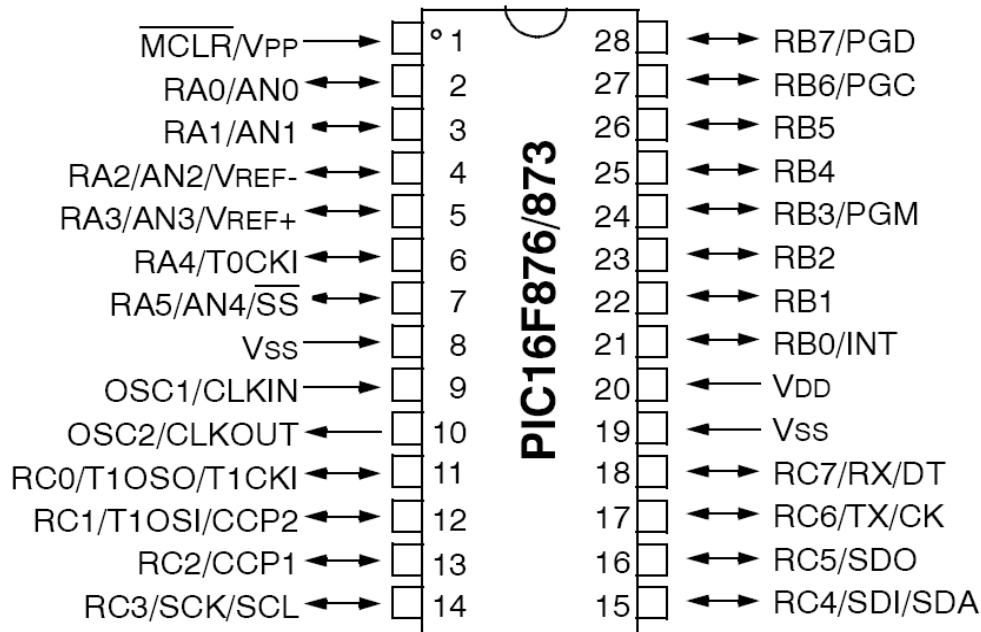
#### **II.4.4 Description des autres éléments : [9]**

- Une mémoire morte de type FLASH de 8 kMots (1Mot = 14 bits).
- Une mémoire vive RAM de 368 octets.
- Une mémoire EEPROM de 256 octets pour sauvegarder des paramètres.
- Un chien de garde WDT (Watch Dog Timer), dispositif permettant une initialisation du PIC lorsque le programme s'exécute en dehors du temps imparti (erreur programme).
- Une surveillance de la tension d'alimentation BOR (Reset si chute de tension VDD).
- 22 lignes d'entrées/sorties (chaque sortie peut supporter un courant maximum de 25mA).

- 3 Temporisateurs (TIMERS) qui permettent de compter des impulsions reçues sur le microcontrôleur PIC ou de compter les cycles d'horloge du microcontrôleur PIC (afin de programmer des temporisations).
- Un module de débogage ICD (In Circuit Debugger).
- 2 modules CCP1-2 (Capture Compare PWM) fonctionnant dans l'un des trois modes suivants :
  - Mode capture (CAPTURE) : ce mode permet en outre d'effectuer des mesures de temps.
  - Mode comparaison (COMPARE) : ce mode permet en outre de générer des événements périodiques.
  - Mode PWM (PULSE WIDTH MODULATION) : ce mode permet de générer des signaux dont le rapport cyclique est variable.
- Un convertisseur analogique/numérique 10 bits avec 5 entrées analogiques multiplexées du port A.
- Une interface de communication USART/SCI (Universal Synchronous & Asynchronous Receiver Transmitter).
- Une interface de communication série synchrone SSP/SPI et I2C.
- Un oscillateur interne qui permet une fréquence maximale d'oscillation  $f$  élevée de 20 MHz. La fréquence de fonctionnement réelle du microcontrôleur PIC est égale à la fréquence de l'oscillateur divisée par 4, soit 5 MHz. Une instruction standard dure donc une période d'horloge, soit 200 ns.
- Une seule tension d'alimentation comprise entre 4V et 5,5V.



**II.4.5 Brochage du PIC 16F876A : [9]**

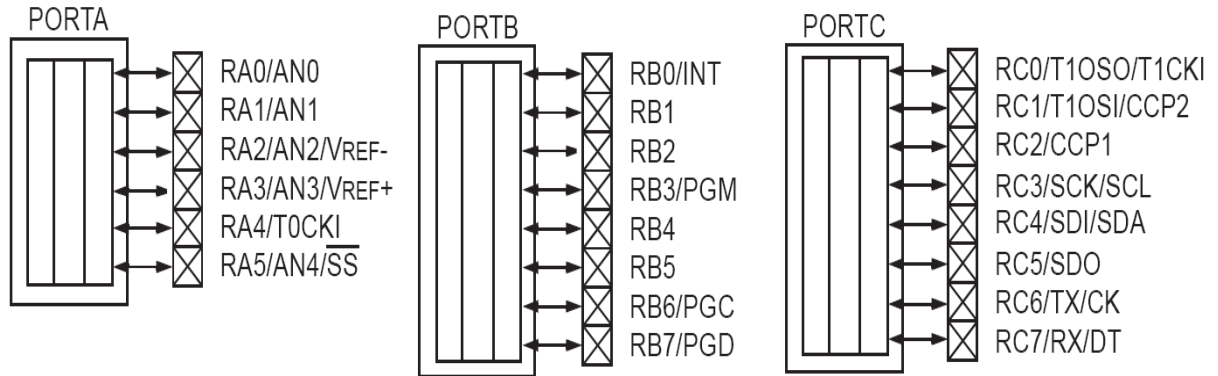


**Figure II.5:** Brochage du 16F876A

(Voir Figure II.5)

- MCLR : cette broche sert à initialiser le microcontrôleur PIC :
  - À la mise sous tension par un front montant (min 72 ms, max 72 ms +1024 x T<sub>osc</sub>). Cette initialisation est appelée POR (POWER ON RESET). Cette broche peut être simplement reliée à l'alimentation VDD si on n'a pas besoin de RESET externe.
  - Par la mise à l'état bas de cette broche, reset externe (EXTERNAL RESET). Si on souhaite implanter un bouton de remise à zéro, on pourra câbler un simple réseau RC sur la broche MCLR. Le microcontrôleur PIC possède d'autres sources de reset mais cette fois-ci internes (WDT et BOR vues précédemment).
- OSC1 et OSC2 : ces broches permettent de faire fonctionner l'oscillateur interne du microcontrôleur PIC de trois façons différentes.
- VDD et VSS : broches d'alimentation du circuit (la tension VDD peut être comprise entre 4V à 5,5V).
- Les ports du microcontrôleur 16F876A sont couplés à différents modules (Timers (ports A et C), convertisseur analogique numérique (port A), plusieurs types de liaisons séries (port C). C'est pour cette raison que le nom de ces broches

comporte plusieurs désignations (ex : RA0/AN0, broches RA0 couplée au convertisseur A/N). (Voir Figure II.6)

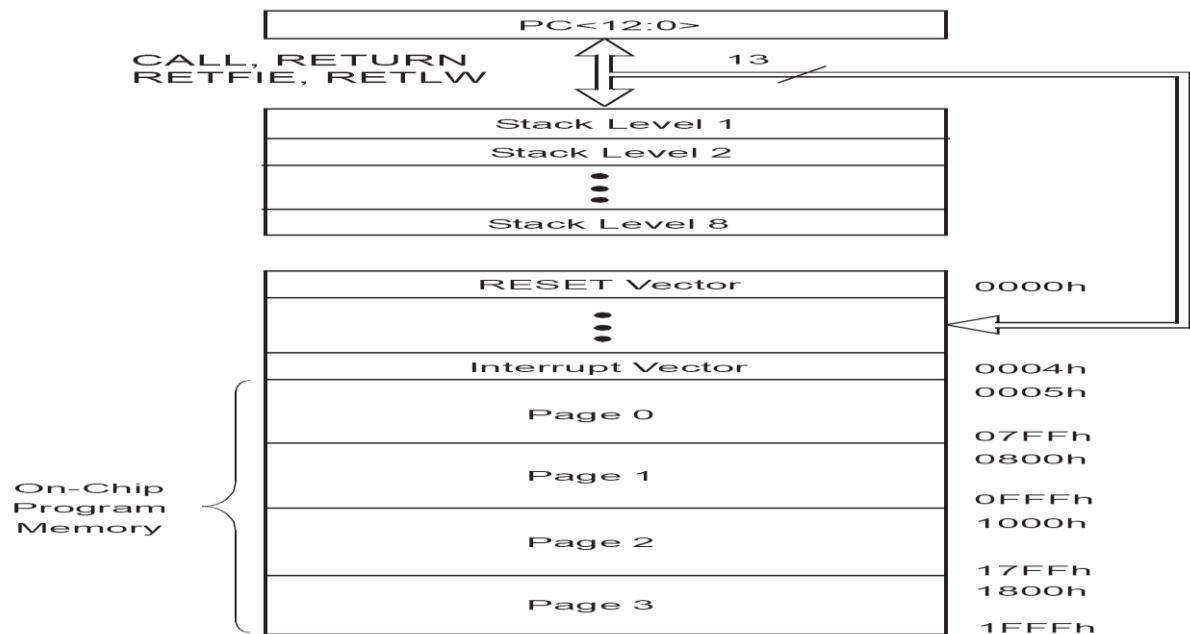


**Figure II.6:** Les Ports du PIC 16F876A [9]

- RA0 à RA5 : 6 broches du port A. Chaque ligne peut être configurée individuellement en entrée ou en sortie.
- RB0 à RB7 : 8 broches du port B. Chaque ligne peut être configurée individuellement en entrée ou en sortie. Les broches RB6/PGC (ProGramming Clock) et RB7/PGD (ProGramming Data) permettent la connexion à un module ICD (In Circuit Debugger, mise au point de programme sur site).
- RC0 à RC7 : 8 broches du port C. Chaque ligne peut être configurée individuellement en entrée ou en sortie.

#### II.4.6 Plan Mémoires : [9]

Comme les microcontrôleurs PIC utilisent un bus pour les instructions et un bus pour les données, il faut considérer deux plans mémoire l'un pour les instructions et l'autre pour les données ainsi que les registres internes (Voir Figure II.7).



**Figure II.7:** Plan mémoire du PIC16F876A

#### **II.4.6.1 Plan Mémoire pour les instructions :**

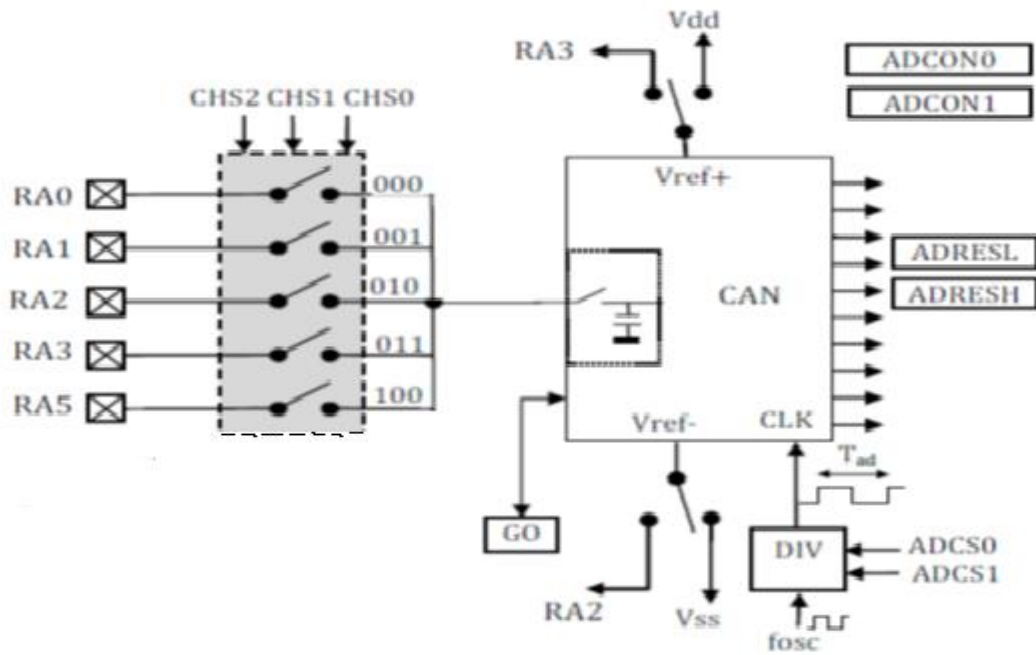
Le plan mémoire est linéaire. Les adresses vont de \$0000 à \$1FFF (8 k mots de 14 bits), par page de 2 K mots. Le vecteur d'initialisation (de reset) est figé à l'adresse \$0000. Les microcontrôleurs PIC n'ont qu'un seul vecteur d'interruption à l'adresse \$0004. Lors d'une demande d'interruption, le sous-programme associé doit déterminer quel périphérique a demandé cette interruption.

Les piles utilisées par les sous programmes ne sont pas implantées en mémoire de données comme dans le cas d'un microcontrôleur classique, mais dans la mémoire programme. Elles sont utilisées lors d'appels de sous programmes, on ne peut pas imbriquer plus de 8 sous programmes.

#### **II.4.6.2 Plan mémoire des données et des registres internes :**

Le plan mémoire des données et des registres internes est découpé en 4 zones ou banks de 128 octets. Pour accéder à une zone, il faut positionner les bits RP0 (bit n°5) et RP1 (bit n°6) du registre STATUS.

#### II.4.7 Le module de conversion A/N :



**Figure II.8:** Le module de conversion A/N

Ce module (Voir Figure II.8) est constitué d'un convertisseur Analogique Numérique 10 bits dont l'entrée analogique peut être connectée sur l'une des 5 entrées analogiques externes. On dit qu'on a un CAN à 5 canaux. Les entrées analogiques doivent être configurées en entrée à l'aide des registres TRISA. L'échantillonneur bloqueur est intégré, il est constitué d'un interrupteur d'échantillonnage et d'une capacité de blocage de 120 pF. Les tensions de références permettant de fixer la dynamique du convertisseur. Elles peuvent être choisies parmi Vdd, Vss, Vr+ ou Vr- [08]

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

##### II.4.7.1 Le registre ADCON0 :

|       |       |      |      |      |         |   |      |
|-------|-------|------|------|------|---------|---|------|
| ADCS1 | ADCS0 | CHS2 | CHS1 | CHS0 | GO/DONE | — | ADON |
|-------|-------|------|------|------|---------|---|------|

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

00 : Fosc/2

01 : Fosc/8

10 : Fosc/32

11 : Oscillateur RC dédié au CAN (fonctionne seulement pour FOSC < 1MHz)

**CHS2 : CHS0** : choix de l'entrée analogique

000 = channel 0, (RA0)

001 = channel 1, (RA1)

010 = channel 2, (RA2)

011 = channel 3, (RA3)

100 = channel 4, (RA5)

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

**ADON** : Ce bit permet de mettre le module AN en service

#### **II.4.7.2 Le registre ADCON1 :**

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

**ADFM** : justification à droite ou à gauche du résultat dans les registres ADRESH et ADRESL

**ADRESH ADRESL**

**1 : justifié à droite 000000XX XXXXXXXX**

**0 : justifié à gauche XXXXXXXX XX000000**

**PCFG3 : PCFG0** : configuration des E/S et des tensions de références. Les 5 broches de PORTA peuvent être configurées soit en E/S digitales, soit en entrées analogiques. RA2 et RA3 peuvent aussi être configurées en entrée de référence.

#### **II.4.7.3 Déroulement d'une Conversion : [8]**

Le PIC dispose d'un échantillonneur bloqueur intégré constitué d'un interrupteur S, d'une capacité de maintien C=120 pF et d'un convertisseur Analogique numérique 10 bits. Pendant la conversion, la tension  $V_e$  à l'entrée du convertisseur A/N doit être maintenue constante. Au départ il faut commencer par faire l'acquisition du signal en fermant l'interrupteur S, ceci se fait à l'aide du registre ADCON0, soit au moment de la validation

du module par le bit ADON soit après un changement de canal si ADON est déjà positionné.

Après la fin de l'acquisition, on peut démarrer une conversion en positionnant le bit GO\_DONE, l'interrupteur S s'ouvre pour assurer le blocage de la tension. La conversion commence, elle est réalisée en 12 TAD, à la fin, le bit GO\_DONE repasse à 0, le drapeau ADIF passe à 1 et le résultat est chargé dans les registres ADRESL et ADRESH. Le module met 2 TAD supplémentaires pour fermer l'interrupteur S ce qui démarre une nouvelle phase d'acquisition pendant laquelle la tension  $V_e$  rejoint la tension analogique d'entrée  $V_a$ . Le temps d'acquisition dépend de la constante de temps RC, R étant la somme des résistances entre le module de conversion et la source de la tension analogique. Après la fin de l'acquisition, on peut démarrer une nouvelle conversion et ainsi de suite.

#### **II.4.8 L'USART : [8]**

L'USART (Universal Synchronous Asynchronous Receiver Transmitter) est l'un des deux modules de communication série dont dispose le PIC 16F876/877. L'USART peut être configuré comme système de communication asynchrone full duplex ou comme système synchrone half duplex. La communication se fait sur les deux broches RC6/TX et RC7/RX qui doivent être configurées toutes les deux en ENTREE par TRISC.

##### **II.4.8.1 Mode Asynchrone :**

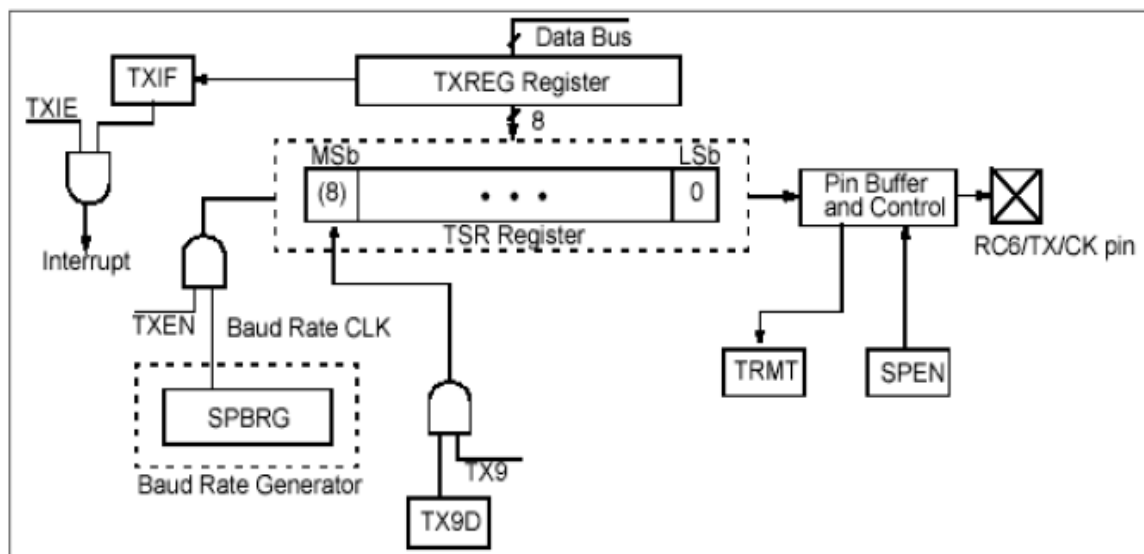
Si on place le bit SYNC du registre TXSTAT à 0, l'USART fonctionne dans le mode asynchrone standard, 10 bits sont transmis ou reçus dans l'ordre ci-dessous :

- 1 bit de START (toujours 0)
- 8 ou 9 bits de donnée (LSB d'abord)
- 1 bit de STOP (toujours 1)
- La transmission se fait sur la broche RC6/TX et la réception sur la broche RC7/RX
- La configuration et le contrôle du port se fait par les registres TXSTA et RCSTA
- La vitesse de communication est fixée par le registre SPBRG et le bit TXSTA.BRGH
- La parité n'est pas gérée d'une façon matérielle, elle peut être gérée par logiciel si son utilisation est nécessaire.

L'accès au port en lecture ou écriture se fait par les registres tampon RCREG et TXREG. La transmission et la réception se font par deux registres à décalage, un pour la transmission (TSR) et un pour la réception (RSR). L'accès aux registres tampon peut se faire alors que les registres à décalage sont en train de transmettre/recevoir une donnée.

Les drapeaux PIR1.RCIF et PIR1.TXIF sont très utiles pour gérer la lecture/écriture dans le port. RCIF est positionné quand le port a terminé une réception et TXIF est positionné quand le buffer de transmission TXREG est "vide".

#### II.4.8.2 Le port en transmission :



**Figure II.9:** Le port en transmission

Le contrôle de la transmission se fait par le registre TXSTA (Voir Figure II.9) :

|      |     |      |      |   |      |      |      |
|------|-----|------|------|---|------|------|------|
| CSRC | TX9 | TXEN | SYNC | — | BRGH | TRMT | TX9D |
|------|-----|------|------|---|------|------|------|

**CSRC** : non utilisé en mode asynchrone

**TX9 et TX9D** : Pour utiliser le mode 9 bits il faut positionner le bit TX9. Le 9ème bit doit être écrit dans TX9D avant d'écrire les 8 bits de données dans TXREG

**TXEN** : permet de valider ou interdire la transmission

**SYNC** : 0 : mode asynchrone, 1 : mode synchrone

**BRGH** : sélectionne le mode haut débit du générateur de baud rate

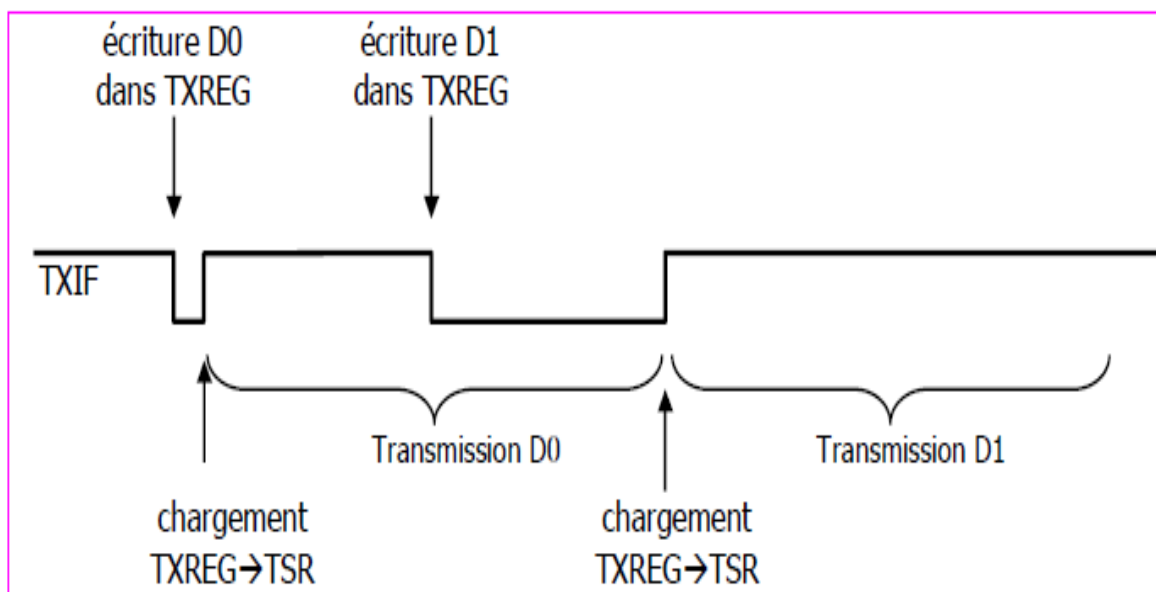
**TRMT** : Indicateur de l'activité du registre à décalage de transmission TSR

1 : TSR libre, 0 : TSR en activité

### II.4.8.3 Déroulement de la transmission :

Quand on écrit un octet D dans le registre TXREG, le drapeau PIR1.TXIF passe à 0, ensuite, deux situations sont possibles :

- Le registre de transmission TSR n'est pas occupé, alors la donnée D est transférée immédiatement dans TSR qui commence sa transmission bit par bit. Le drapeau TXIF repasse à 1 pour nous dire que nous pouvons de nouveau écrire dans TXREG.
- Le registre de transmission TSR est occupé à transmettre un octet qui lui été donne auparavant. La donnée D attend dans TXREG, et le drapeau TXIF reste à 0 jusqu'à ce que TSR termine de transmettre l'octet précédent. La donnée D est alors transférée dans TSR qui commence sa transmission bit par bit. Le drapeau TXIF repasse à 1 pour nous dire que nous pouvons de nouveau écrire dans TXREG.



**Figure II.10:** Illustration de la transmission de deux octet successifs D0 et D1

A la mise sous tension, tous les drapeaux sont à zéro y compris TXIF. Dès qu'on valide la transmission en positionnant le bit TXEN, le drapeau TXIF passe automatiquement à 1, sauf si on écrit d'abord un octet dans TXREG et on valide ensuite le bit TXEN, dans ce cas, dès la validation de TXEN, le contenu de TXREG est transféré dans TSR qui commence sa transmission et le drapeau TXIF passe à 1.

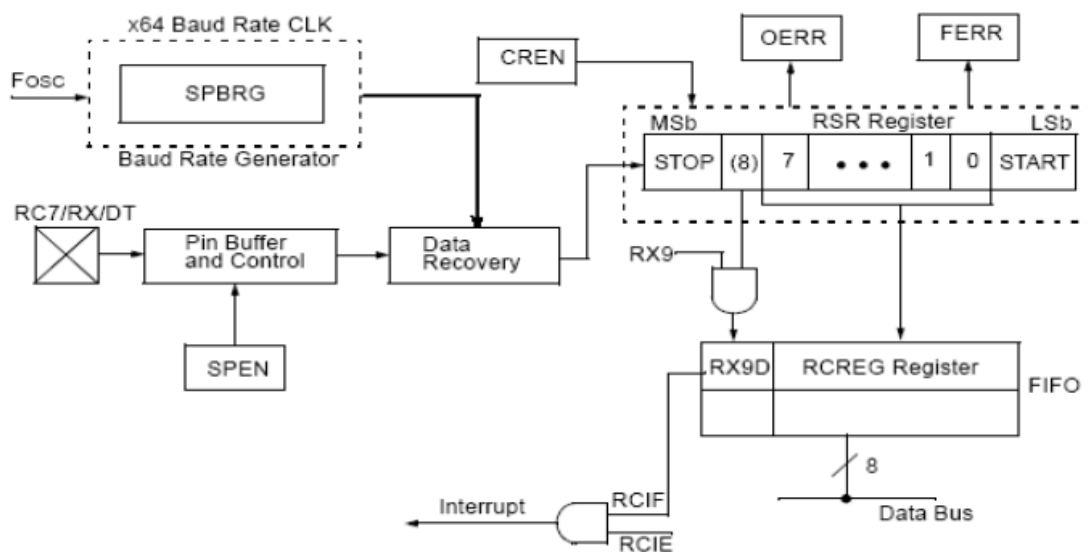
Comme on vient de le voir, le drapeau TXIF est géré automatiquement, on ne peut pas le modifier directement par programme (Voir Figure II.10).



**II.4.8.4 Les étapes de transmission (sans interruption, mode 8 bits) :**

- 1) S'assurer que l'interruption TXI n'est pas validée
- 2) Configurer la broche TX/RC6 en entrée
- 3) Configurer le registre TXSTA (mode 8 bits, valider transmission, asynchrone, BRGH)
- 4) Initialiser le registre SPBRG pour définir la vitesse de transmission
- 5) Valider le port avec le bit RCSTA.SPEN
- 6) Vérifier que le drapeau PIR1.TXIF=1 c.à.d. TXREG est vide
- 7) Placer la donnée à transmettre dans le registre TXREG
- 8) Recommencer au point 6) tant qu'on a des données à transmettre

**II.4.8.5 Le port en réception :**



**Figure II.11:** Le port en réception

La réception est validée par le bit CREN (Voir Figure II.11).

La réception d'un octet démarre à la réception du START bit qui commence toujours par une transition 1 → 0

A la réception du stop bit le contenu du registre à décalage de réception RSR est recopié dans le registre tampon de réception RCREG. Le drapeau RCIF (PIR1.5) est positionné, l'interruption associée est déclenchée si elle est validée. Le drapeau RCIF est remis à zéro automatiquement au moment de la lecture dans RCREG.

Le registre RCREG est un registre double (FIFO a 2 positions). On peut donc avoir 2 octets en attente dans ce registre et être en train de recevoir un 3eme dans le registre a décalage RSR. A la fin de la réception du 3eme octet, si RCREG est toujours plein, alors le dernier octet reçu est perdu et le bit OERR (Overrun ERRor bit) est positionne ce qui provoque l'arrêt des transferts du registre a décalage RSR vers le buffer RCREG. Pour reprendre la réception il faut réinitialiser le module de réception en mettant à 0 puis a 1 le bit CREN.

Le drapeau PIR1.RCIF ne passe à 0 que quand la pile RCREG est vide. Si on reçoit un 0 a la position du STOP bit qui doit être toujours à 1, alors le bit FERR (Framing ERRor) est positionne. Ce bit ainsi que le 9eme bit (si utilise) sont bufférisés, Ils doivent être lus dans le registre RCSTA avant la lecture du registre RCREG.

Le contrôle de la réception se fait par le registre RCSTA

|      |     |      |      |       |      |      |      |
|------|-----|------|------|-------|------|------|------|
| SPEN | RX9 | SREN | CREN | ADDEN | FERR | OERR | RX9D |
|------|-----|------|------|-------|------|------|------|

**SPEN** : Validation du port série (1 = valide, 0 = Inhibe)

**RX9** : Validation du mode 9 bits (1 = mode 9 bits, 0 = mode 8 bits)

**SREN** : validation de réception d'u seul octet (non utilise en mode asynchrone)

**CREN** : Validation du mode réception continue (1 = valide, 0 = inhibe)

**ADDEN** : validation du mode détection d'adresse en mode 9 bits, utilise en mode multiprocesseurs (1= valide, 0 = Inhibe)

**FERR** : Erreur de synchronisation, lecture seule.

**OERR** : Erreur débordement du buffer de réception, lecture seule.

**RX9D** : En mode 9 bits, le 9eme bit est reçu ici

### **II.4.8.6 Les étapes de réception (sans interruption, mode 8 bits) :**

- 1) S'assurer que l'interruption RCI n'est pas validée
- 2) Configurer la broche RX/RC7 en entrée
- 3) Initialiser le registre SPBRG pour définir la vitesse de communication
- 4) Configurer le registre TXSTA (asynchrone, BRGH)

- 5) Configurer le registre RCSTA (validation port, mode 8 bits, valider réception continue)
- 6) Attendre que drapeau RCIF passe à 1 ce qui annonce la fin de réception d'un octet
- 7) Lire l'octet reçu dans le registre RCREG
- 8) Recommencer au point 6) tant qu'on a des données à recevoir

#### **II.4.9 Jeu d'instructions :**

Les PICs sont conçus selon une architecture RISC. Programmer avec un nombre d'instructions réduit permet de limiter la taille de leur codage et donc de la place mémoire et du temps d'exécution. Voici la liste des instructions :

- **addlw k:** Additionne une constante k à W. Résultat dans W.
- **addwf f,d:** Additionne une variable à l'adresse f avec W. Résultat dans f si d=1 ou dans W si d=0.
- **andlw k:** ET logique entre une constante k et W. Résultat dans W.
- **andwf f,d:** ET logique entre une variable à l'adresse f et W. Résultat dans f si d=1 ou dans W si d=0.
- **bcf f,b:** Met le bit b de f à 0.
- **bsf f,b:** Met le bit b de f à 1.
- **btfsf f,b:** Teste le bit b de f. Si ce bit vaut 0 alors l'instruction suivante est ignorée (instruction NOP exécuté à la place).
- **btfsf f,b:** Teste le bit b de f. Si ce bit vaut 1 alors l'instruction suivante est ignorée (instruction NOP exécuté à la place).
- **call k:** Appel à un sous/programme (PC mémorisé dans la pile système).
- **clrf f:** Met 0 dans f.
- **clrw:** Met 0 dans W.
- **clrwdt:** Fait un reset du chien de garde (registre WDT counter mis à 0).
- **comf f,d:** Complément à 1 de f. Résultat dans f si d=1 ou dans W si d=0.
- **decf f,d:** Décrémente de -1 la variable à f. Résultat dans f si d=1 ou dans W si d=0.

- **decfsz f,d:** Décrémente de  $-1$  la variable à f. Résultat dans f si d=1 ou dans W si d=0. Si le résultat de l'opération vaut 0, alors l'instruction suivante est ignorée (instruction NOP exécuté à la place).
- **goto k:** Branchement inconditionnel à l'adresse k.
- **incf f,d:** Incrémente de  $+1$  la variable à f. Résultat dans f si d=1 ou dans W si d=0.
- **incfsz f,d:** Incrémente de  $+1$  la variable à f. Résultat dans f si d=1 ou dans W si d=0. Si le résultat de l'opération vaut 0, alors l'instruction suivante est ignorée (instruction NOP exécuté à la place).
- **iorlw k:** OU logique entre une constante k et W. Résultat dans W.
- **iorwf f,d:** OU logique entre une variable à l'adresse f et W. Résultat dans f si d=1 ou dans W si d=0.
- **movf f,d:** Met la variable contenu à l'adresse f dans W si d=0 ou dans f si d=1.
- **movlw k:** Met la variable k dans W
- **movwf f:** Met la variable contenu dans W à l'adresse f.
- **nop:** Aucune opération effectué (1 cycle machine exécuté).
- **retfie:** Fin du programme interruption (dépile la pile système).
- **retlw k:** Fin d'un sous/programme (associée à call). Met la variable k dans W.
- **return:** Fin d'un sous/programme (associée à call).
- **rlf f,d:** Rotation logique à gauche de la variable stockée à l'adresse f. Résultat dans f si d=1 ou dans W si d=0.
- **rrf f,d:** Rotation logique à droite de la variable stockée à l'adresse f. Résultat dans f si d=1 ou dans W si d=0.
- **sleep:** Met le microprocesseur en mode sommeil.
- **sublw k:** Soustraction :  $k-W$ , résultat dans W.
- **subwf f,d:** Soustraction : une variable à l'adresse f-W, résultat dans f si d=1 ou dans W si d=0.
- **swapf f,d:** Permute les deux quartets de f. Résultat dans f si d=1 ou dans W si d=0.

- **xorlw k:** OU Exclusif entre une constante k et W. Résultat dans W.
- **xorwf f,d:** OU Exclusif entre une variable à l'adresse f et W. Résultat dans f si d=1 ou dans W si d=0.

## **II.5 La gestion de la liaison RS232 : [11]**

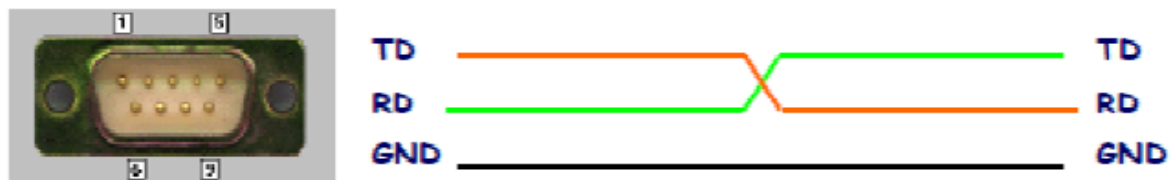
### **II.5.1 Communication série asynchrone à travers le port série RS232 :**

Les liaisons séries permettent la communication entre deux systèmes numériques en limitant le nombre de fils de transmission (Voir Figure II.12). La liaison série aux normes de RS232 est utilisée dans tous les domaines de l'informatique. Elle est de type asynchrone. Son schéma fonctionnel est le suivant :



**Figure II.12 :** Schéma fonctionnel d'une liaison série asynchrone de la norme RS232

La transmission série nécessite au moins 2 fils de communication, l'un pour la transmission (Tx) et l'autre pour la réception (RX) et un fil de masse (Voir Figure II.13).



**Figure II.13:** la liaison série croisée

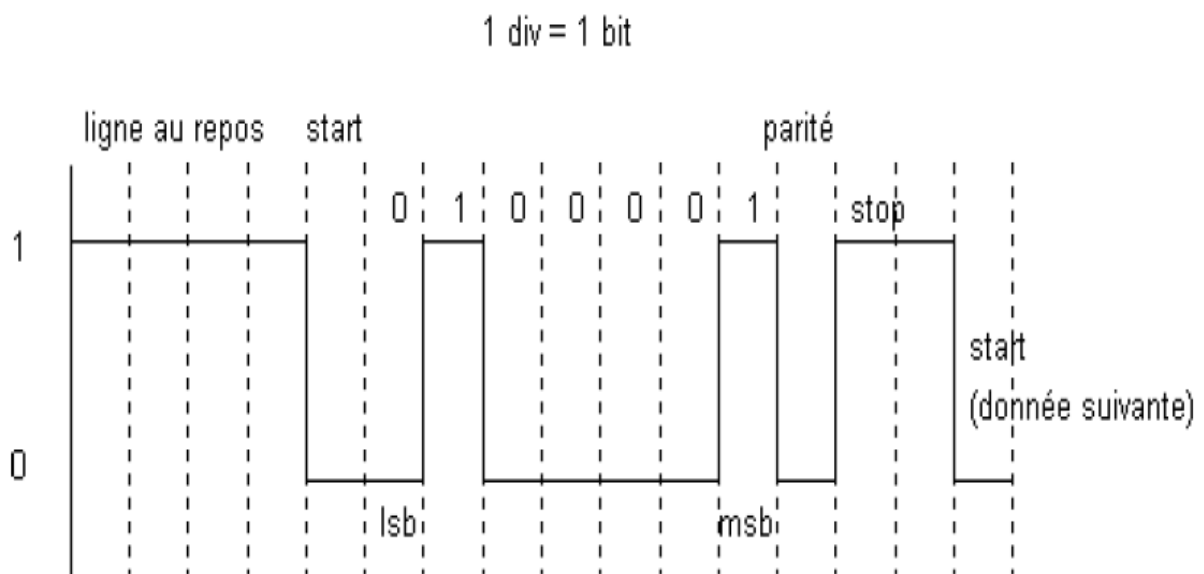
### **II.5.2 Protocole de transmission : [12]**

Afin que les éléments communicants puissent se comprendre, il est nécessaire d'établir un protocole de transmission. Ce protocole devra être le même pour les deux éléments afin que la transmission fonctionne correctement. Paramètres rentrant en jeu :

- Longueur des mots : 7 bits (ex : caractère ascii) ou 8 bits
- La vitesse de transmission : les différentes vitesses de transmission son réglables à partir de 110 bauds (bits par seconde) de la façon suivante : 110 bds, 150 bds, 300 bds, 600 bds, 1200 bds, 2400 bds, 4800 bds, 9600 bds.

- Parité : le mot transmis peut être suivi ou non d'un bit de parité qui sert à détecter les erreurs éventuelles de transmission. Il existe deux types de parité.
  - Parité paire : le bit ajouté à la donnée est positionné de telle façon que le nombre des états 1 soit paire sur l'ensemble donne + bit de parité ex : soit la donnée 11001011 contenant 5 état 1, le bit de parité paire est positionné à 1, ramenant ainsi le nombre de 1 à 6.
  - Parité impaire : le bit ajouté à la donnée est positionné de telle façon que le nombre des états 1 soit impaire sur l'ensemble donne + bit de parité ex : soit la donnée 11001011 contenant 5 état 1, le bit de parité paire est positionné à 0, laissant ainsi un nombre de 1 impaire.
- Bit de start : la ligne au repos est à l'état logique 1 pour indiquer qu'un mot va être transmis la ligne passe à l'état bas avant de commencer le transfert. Ce bit permet de synchroniser l'horloge du récepteur.
- Bit de stop : après la transmission, la ligne est positionnée au repos pendant 1, 2 ou 1,5 période d'horloge selon le nombre de bits de stop. Le bit de start apparait en premier dans la trame puis les données (poids faible en premier), la parité éventuelle et le (les) bit(s) de stop. Exemple :

Soit à transmettre en parité paire, avec 2 bits de stop, le caractère B dont le codage ascii est 0100001 (Voir Figure II.14) :

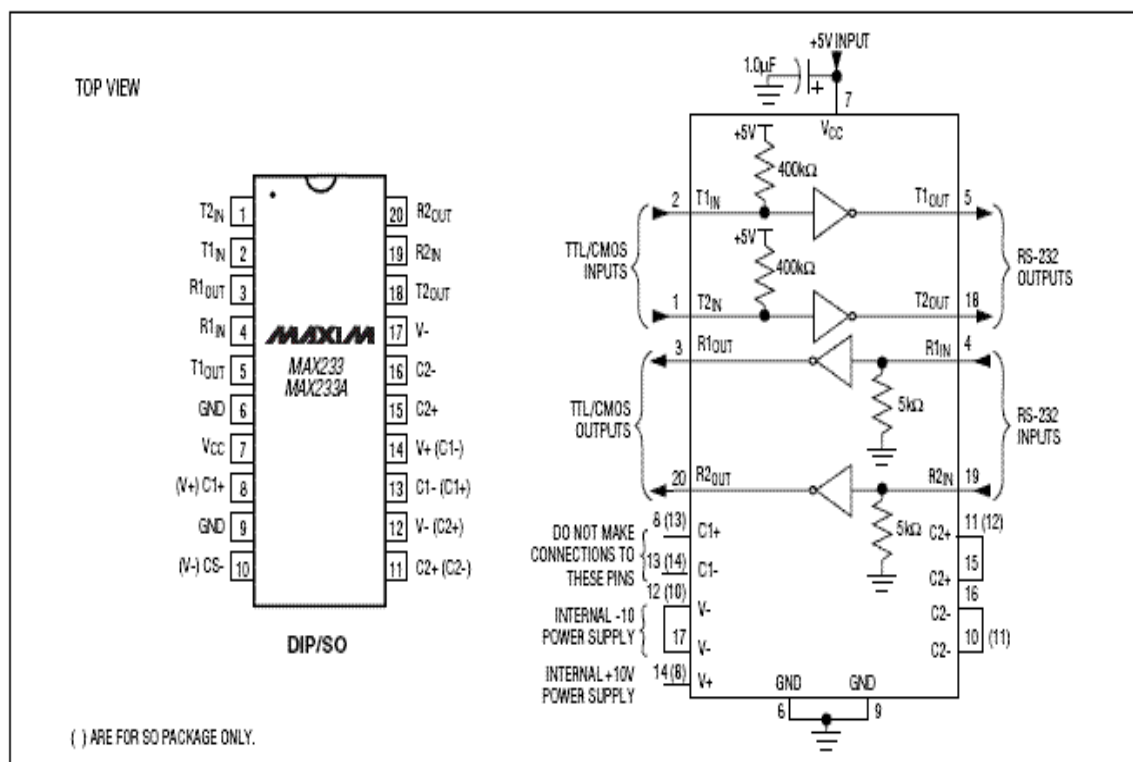


**Figure II.14:** Exemple de transmission série

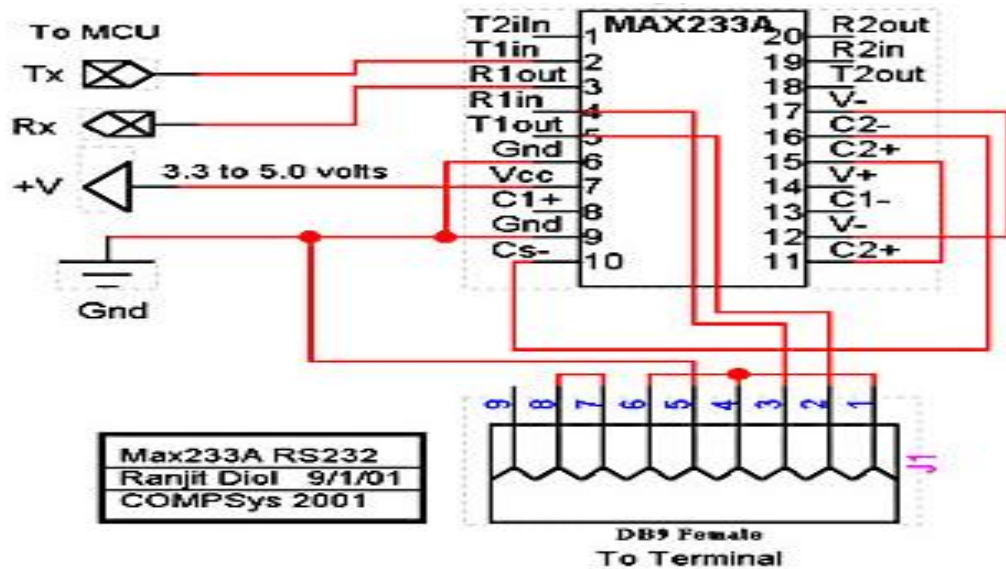
### II.5.3 La conversion des niveaux : [12]

Les cartes électroniques à base de microcontrôleurs fonctionnent très souvent avec des niveaux TTL soit 0-5Volt, 0V pour le niveau 0 et 5Volt pour le niveau 1. Brancher donc directement une ligne RS232 sur un microcontrôleur n'aurait donc aucun sens et pourrait aussi endommager le système en imposant des tensions de 25volt.

Pour rendre compatible une ligne RS232 avec une carte de ce type il existe un composant très simple d'utilisation que nous allons étudier : le max232. En regardant son schéma interne ci-dessous, nous constatons directement qu'il est premièrement doté d'un convertisseur de tension, au travers des capacités C1 et C3 il génère une tension de 10Volt depuis les 5Volt (doubleur de tension), et au moyen des capacités C2 et C4 il génère une tension de -10Volt à partir de la tension de 10Volt. Il est bien sur évident que le circuit est muni de tout un système, avec un oscillateur, des diodes etc..., afin d'intégrer ce convertisseur DC-DC. Il existe une version de ce circuit, le max233 (Voir Figure II.15), où les capacités sont intégrées directement dedans. La valeur des capacités va dépendre de la version du circuit intégré utilisé :



**Figure II.15: Le MAX 233**



**Figure II.16:** La connexion entre un max233 et un Pic

Nous avons vu précédemment comment connecter un max233 à un port série(PC), nous allons voir maintenant comment connecter un microcontrôleur à ce max233 (Voir Figure II.16) afin de communiquer avec le port série, et ainsi avoir un échange entre un ordinateur et une carte électronique. Ce genre de communication est beaucoup utilisée dans les systèmes embarqués, les modems, routeurs, machines afin soit de changer les paramètres du système soit de pouvoir en continu échanger des données. Même si le port USB peut faire songer à la mort du port série, celui-ci par sa simplicité de mise en œuvre et ses preuves de fidélité faites depuis longtemps, reste toujours fort présent même dans les nouveaux développements.

Beaucoup de microcontrôleurs sont actuellement dotés de la gestion automatique de la communication, on les reconnaît par la fonction UART présente. UART signifie : Universal Asynchronous Receiver Transmitter.

#### **II.5.4 L'émission :**

Premièrement le niveau de la ligne au repos en TTL est à l'état 1. Nous déterminons une sortie sur les broches du pic TX. Lorsque la communication n'a pas lieu TX=1 au repos. Si nous désirons envoyer un caractère de 8bits, la première étape est l'émission d'un bit « start » qui est un niveau de logique 0 (TX=0). Les bits sont envoyés dans l'ordre du plus faible au plus fort. Une fois le caractère envoyé nous allons remettre la ligne à l'état de repos et nous assurer que rien ne va être transmis sur la ligne avant au moins une période ce qui marquera un bit de stop.



### **II.5.5 La réception :**

La réception se passe relativement de la même façon, sauf que nous n'aurons plus une sortie du pic, mais une entrée RX, sur laquelle nous lirons la valeur. Dans un premier temps nous allons attendre le début de transmission, c'est-à-dire l'envoi du bit «start», soit le passage de la ligne au niveau 0 ce qui permet la synchronisation avec l'horloge de l'émetteur.

### **II.6 Les afficheurs à cristaux liquides LCD : [13]**



**Figure II.17:** Afficheur LCD

Les afficheurs à cristaux liquides (Voir Figure II.17), autrement 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 et diffèrent les uns des autres, non seulement par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétro éclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module.

#### **II.6.1 Principe des cristaux liquides :**

L'afficheur est constitué de deux lames de verre, distantes de 20 µm environ, sur lesquelles sont dessinées les mantisses formant les caractères. L'espace entre elles est rempli de cristal liquide normalement réfléchissant (pour les modèles réflectifs). L'application entre les deux faces d'une tension alternative basse fréquence de quelques volts (3 à 5 V) le rend absorbant. Les caractères apparaissent sombres sur fond clair. N'émettant pas de lumière, un afficheur à cristaux liquides réflectif ne peut être utilisé qu'avec un bon éclairage ambiant. Sa lisibilité augmente avec l'éclairage. Les modèles

transmissifs fonctionnent différemment : normalement opaque au repos, le cristal liquide devient transparent lorsqu'il est excité ; pour rendre un tel afficheur lisible, il est nécessaire de l'éclairer par l'arrière, comme c'est le cas pour les modèles rétro éclairés.

### II.6.2 Brochages :

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



**Figure II.18:** Brochages du LCD

(Voir Figure II.18)

| Broche | Nom | Fonction                                                                                                                                                                                                                   |
|--------|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1      | GND | Masse                                                                                                                                                                                                                      |
| 2      | Vcc | Alimentation positive +5V                                                                                                                                                                                                  |
| 3      | VEE | 0-5V Cette tension permet, en la faisant varier entre 0 et +5V, le réglage du contraste de l'afficheur.                                                                                                                    |
| 4      | RS  | Sélection du registre (Register Select)<br>Grâce à cette broche, l'afficheur est capable de faire la différence entre une commande et une donnée. Un niveau bas indique une commande et un niveau haut indique une donnée. |
| 5      | R/W | Lecture ou écriture (Read/Write)<br>L : Écriture<br>H : Lecture                                                                                                                                                            |

|    |      |                                                                                                                                      |
|----|------|--------------------------------------------------------------------------------------------------------------------------------------|
| 6  | EN   | Entrée de validation (Enable) active sur front descendant. Le niveau haut doit être maintenue pendant au moins 450 ns à l'état haut. |
| 7  | D0   | Bus de données bidirectionnel 3 états (haute impédance lorsque E=0)                                                                  |
| 8  | D1   |                                                                                                                                      |
| 9  | D2   |                                                                                                                                      |
| 10 | D3   |                                                                                                                                      |
| 11 | D4   |                                                                                                                                      |
| 12 | D5   |                                                                                                                                      |
| 13 | D6   |                                                                                                                                      |
| 14 | D7   |                                                                                                                                      |
| 15 | Led+ | Anode rétro éclairage (+5V)                                                                                                          |
| 16 | Led- | Cathode rétro éclairage (masse)                                                                                                      |

NB : Les broches 15 et 16 ne sont présentes que sur les afficheurs LCD avec retro éclairage.

### II.6.3 La mémoire :

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

#### II.6.3.1 La mémoire d'affichage (DD RAM):

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

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

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

#### **II.6.3.2 La mémoire du générateur de caractères (CG RAM) :**

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

La CG RAM utilise des mots de 8 bits de large, mais seul les 5 bits de poids faible apparaissent sur le LCD. Ainsi D4 représente le point le plus à gauche et D0 le point le plus à droite. Par exemple, charger un octet de la CG RAM à 1Fh fait apparaître tous les points de cette rangée ; charger un octet à 00h éteint tous ces points. Les 8 lignes d'un caractère doivent être chargées dans la CG RAM. Elle est peut-être utilisée pour créer des caractères en vidéo inversée, des caractères avec des accents, etc. La limitation d'un total de 8 caractères peut être contournée en utilisant une bibliothèque de 8 symboles résidant dans le système hôte. Un maximum de 8 caractères peut être affiché à la fois.

La CG RAM peut être rechargée périodiquement en fonction des besoins. Si un caractère de cette mémoire est actuellement sur l'afficheur est changé, alors le changement est immédiatement apparent sur l'afficheur.

#### **II.6.4 Commande d'un afficheur LCD :**

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

##### **II.6.4.1 Mode 8 bits :**

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

une lecture. Les données seront valides sur les lignes D0 à D7 lors de l'état haut de la ligne EN.

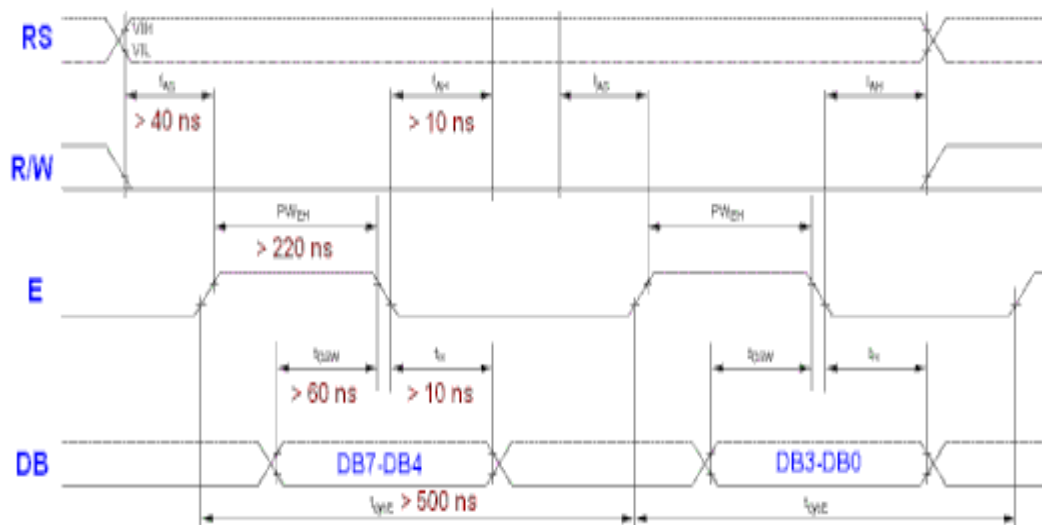
#### **II.6.4.2 Mode 4 bits :**

Il peut, dans certains cas, être nécessaire de diminuer le nombre de fils utilisés pour commander l'afficheur, comme, par exemple lorsqu'on dispose de très peu de broches d'entrées sorties disponibles sur un microcontrôleur. Dans ce cas, on peut utiliser le mode quatre bits de l'afficheur LCD. Dans ce mode, seuls les 4 bits de poids fort (D4 à D7) de l'afficheur sont utilisées pour transmettre les données et les lire (Voir Figure II.19). Les 4 bits de poids faible (D0 à D3) sont alors connectés à la masse. On a donc besoin, hors alimentation de sept fils pour commander l'afficheur. Les données sont alors écrites ou lues en envoyant séquentiellement les quatre bits de poids fort suivi des quatre bits de poids faible. Une impulsion positive d'au moins 450 ns doit être envoyée sur la ligne E pour valider chaque demi-octet ou nibble.

Dans les deux modes, on peut, après chaque action sur l'afficheur, vérifier que celui-ci est en mesure de traiter l'information suivante. Pour cela, il faut demander une lecture en mode commande, et tester le flag Busy BF. Lorsque BF=0, l'afficheur est prêt à recevoir une nouvelle commande ou donnée.

Il se peut qu'on dispose encore de moins de broches disponibles dans l'application envisagée. Dans ce cas, on peut alors relier la ligne R/W à la masse de façon à forcer l'afficheur en écriture. On a alors besoin, hors alimentation de seulement six fils en mode 4 bits, et dix fils en mode 8 bits, pour commander l'afficheur, mais on ne peut alors plus relire l'afficheur. Ceci n'est pas gênant dans la mesure où on sait ce qu'on a écrit sur l'afficheur, mais on ne peut alors plus relire le flag Busy. Il faut alors utiliser des temporisations après chaque écriture sur l'afficheur. On perd alors un peu en temps d'affichage, mais on gagne une broche d'entrée sortie.

**Interface 4 bits : mode écriture**



**Figure II.19:** Diagramme d'écriture et de lecture

**II.7 Le relais:**



**Figure II.20:** Le relais

Un relais (Voir Figure II.20) est un pré actionneur constitué de :

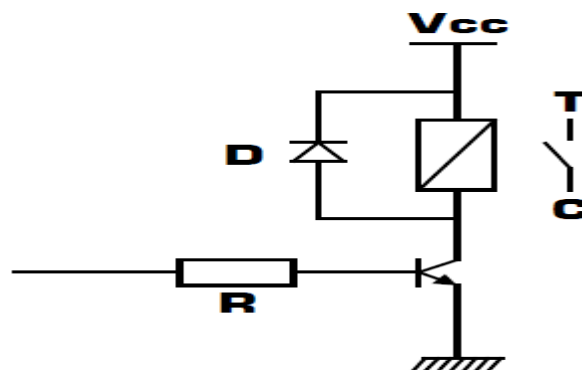
- Un électroaimant (bobine + circuit ferromagnétique)
- Une palette mobile supportant un contact mobile
- Un contact fixe
- Un ressort de rappel du contact mobile

En alimentant la bobine, le contact mobile est déplacé fermant ainsi le contact électrique. En l'absence de courant dans la bobine le ressort de rappel maintient le contact ouvert.

Le relais est une solution à la commande en puissance. Il assure en outre une isolation galvanique en mettant en œuvre un mouvement Mécanique. Le choix d'un relais dépend de la charge à piloter et donc de la tension max et du courant max devant alimenter la charge. C'est le pouvoir de coupure du relais.

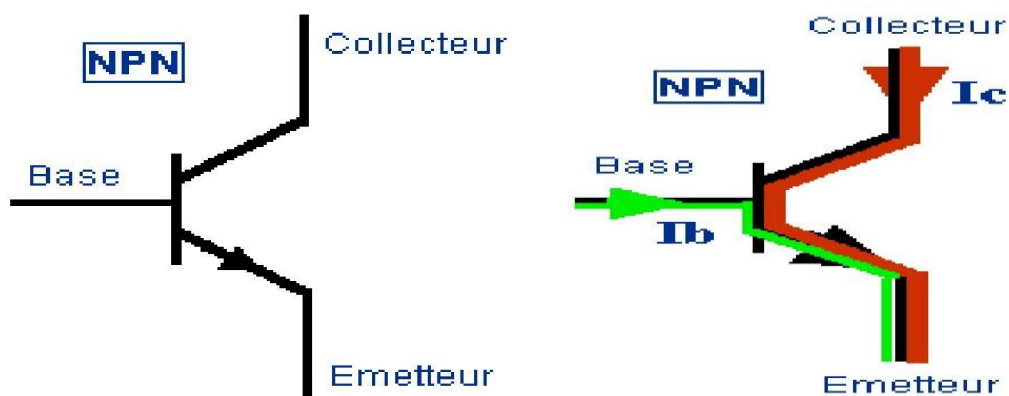
La charge peut également être inductive et poser des problèmes de surtension et donc d'étincelles destructrices sur les contacts du relais. L'absence de liaison électrique entre la haute tension de la charge et la commande de la bobine assure une sécurité pour l'électronique de commande et pour l'utilisateur.

Lorsque le courant bobine est brutalement coupé il se produit aux bornes de la bobine une très brève surtension inverse. Ce phénomène électrique peut être très dangereux pour le transistor qui alimente le relais, on utilise une diode en parallèle avec la bobine du relais pour la protection du transistor (Voir Figure II.21).



**Figure II.21:** Connexion du relais

## II.8 Le transistor bipolaire NPN en commutation : [15]



**Figure I.22 :** Transistor bipolaire de type NPN

Le transistor bipolaire peut-être assimilé à un interrupteur commande (Voir Figure II.22). Il possède 3 broches : une base, un collecteur et un émetteur. Il a deux modes de fonctionnement : le mode linéaire (amplification) et le mode bloque/sature.

Un fonctionnement en interrupteur commande consiste à activer la base, pour qu'elle permette au courant présent dans le collecteur, de circuler jusqu'à l'émetteur.

- Lorsque  $V_{be} = 0$  le transistor est bloqué  $\Rightarrow I_c = I_e = 0$  et  $V_{ce}$  est positif,  $V_{ce} < V_{ce\_max}$  sinon le transistor grille.
- Lorsque  $V_{be} = 0,7\text{ V}$  le transistor est saturé (tension de seuil de la diode base émetteur), le transistor est passant. Pour être saturé :  $I_b > I_c/\beta$

### **II.9 Les quartz :**

Les quartz (Voir Figure II.23) sont obtenus en taillant un mono cristal de  $\text{SiO}_2$  artificiellement produit. La fabrication du mono cristal nécessite des opérations délicates (croissance du cristal à partir d'un germe dans des conditions de température et de pression précises). Il en est de même pour la taille (angle par rapport aux axes cristallographiques, dimensions). Le bout de  $\text{SiO}_2$  est recouvert d'électrodes sur 2 de ses faces et encapsulé dans un boîtier, en général métallique pour les quartz et en résine pour les résonateurs céramiques. Pour les quartz, un ressort permet de maintenir la plaquette au milieu du boîtier. Du fait de son conditionnement, un quartz est assez sensible aux chocs (fortes accélérations ou décélérations). Il est aussi sensible aux vibrations, un montage soigné et une bonne suspension sont donc indispensables.

Le quartz est constitué d'une lame de matériau piézoélectrique prise entre deux électrodes. Sous l'action de la tension alternative  $u$ , la lame vibre à la même fréquence, comme tout système mécanique, la lame possède des fréquences de vibrations propres les propriétés électriques du dipôle sont liées aux vibrations de la lame. Au voisinage de la fréquence de résonance propre de la lame, l'impédance du dipôle subit des variations importantes, en module et en phase, la fréquence de résonance dépend un peu de la température, et le choix de la taille du quartz permet de minimiser cette influence.



**Figure II.23:** Le quartz



**Chapitre III:**  
**Logiciels utilisés**

## **Chapitre III**

### **Logiciels utilisés**

#### **III.1 Introduction :**

La conception et la simulation du fonctionnement de la carte d'acquisition de données est faite par les logiciels suivantes :

- ISIS Proteus : Conception et simulation du schéma électrique
- MikroC: Programmation du PIC en langage C
- Visual C# : Création d'une interface communicante avec la carte
- Virtual Serial Port Driver : Émulateur de port série virtuel
- ICWriter : Transfert du programme source .Hex du PC vers le PIC 16F876A

#### **III.2 ISIS Proteus : [16]**

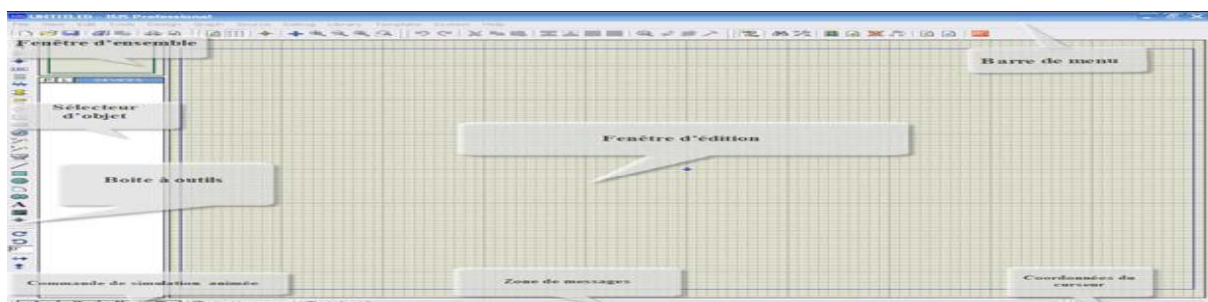
Le simulateur garanti un environnement de réalisation et de test du circuit électronique et aussi nous permet de s'assurer que le programme conçu est opérationnel avant de le sauvegarder sur l'EEPROM interne du microcontrôleur. ISIS produit une liste de matériel et des rapports de contrôle des règles électriques.

##### **III.2.1 L'éditeur de schéma ISIS :**

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

##### **III.2.2 Interface utilisateur :**

(Voir Figure III.1)



**Figure III.1: Interface utilisateur ISIS**

### III.2.3 Fenêtre 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.

### III.2.4 Fenêtre d'édition :

C'est dans cette fenêtre qu'on peut éditer notre circuit. Elle ne représente que la partie du circuit qui est définie par le cadre vert dans la fenêtre d'ensemble.

### III.2.5 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.....

### III.2.6 Coordonnées du curseur :

Les coordonnées déterminent la position du curseur par rapport à l'origine qui par défaut se trouve au centre de la fenêtre d'édition. Les coordonnées sont affichées en 1/1000 de pouce.

### III.2.7 Organisation de la boîte à outils :

(Voir Figure III.2)



Figure III.2: Organisation de la boîte à outils ISIS

## III.3 MIKRO C : [17]

Le langage *mikroC* pour PIC a trouvé une large application pour le développement de systèmes embarqués sur la base de microcontrôleur (Voir Figure III.3). Il assure une combinaison de l'environnement de programmation avancée *IDE* (Integrated Development Environment), et d'un vaste ensemble de bibliothèques pour le matériel, de la documentation complète et d'un grand nombre des exemples.

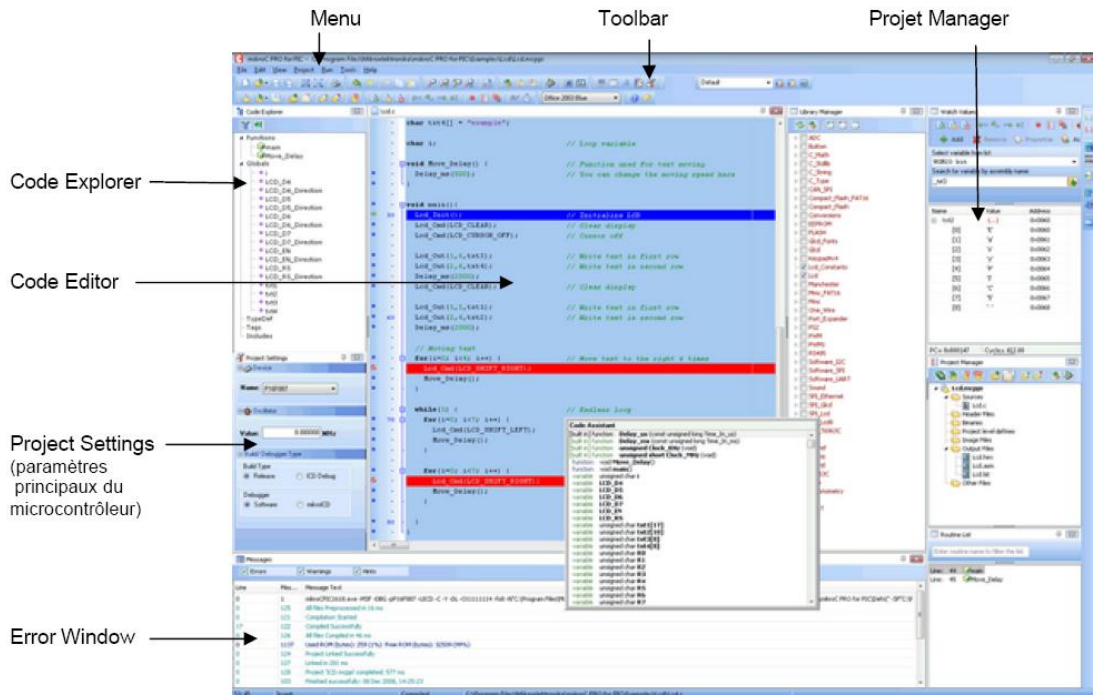


Figure III.3: L'environnement *IDE* du compilateur *mikro C PRO*

## III.3.1 Editeur de code :

L'éditeur de code est le même que toute éditeur de texte standard pour l'environnement de Windows, y compris Copier, Coller, Annuler etc.... Il possède en plus des ressources comme suit :

- Coloration syntaxique réglable
- Assistant de code
- Assistant de paramètre
- Mise en forme automatique

Dans la boîte de dialogue *Options* on peut configurer la coloration syntaxique, l'aide pour le code et paramètres, la correction automatique etc. Pour accéder à ce dialogue

on clique sur *Tools > Options* du menu déroulant ou sur l'icône



### III.3.2 Assistant de code :

Si vous imprimez les premières lettres du mot et puis appuyez sur *Ctrl + Espace*, tous les identificateurs autorisés correspondant aux caractères imprimés seront offerts dans une fenêtre. Maintenant, nous pouvons continuer à réduire le choix de taper ou d'utiliser la souris pour sélectionner l'option appropriée dans la proposée et appuyez sur *Entrée*.

Le MikroC PRO assure un outil de programmation rapide, simple et doté d'une interface de test pour la communication série entre l'ordinateur et le PIC, c'est la fenêtre mikroC terminal (Voir Figure III.4).

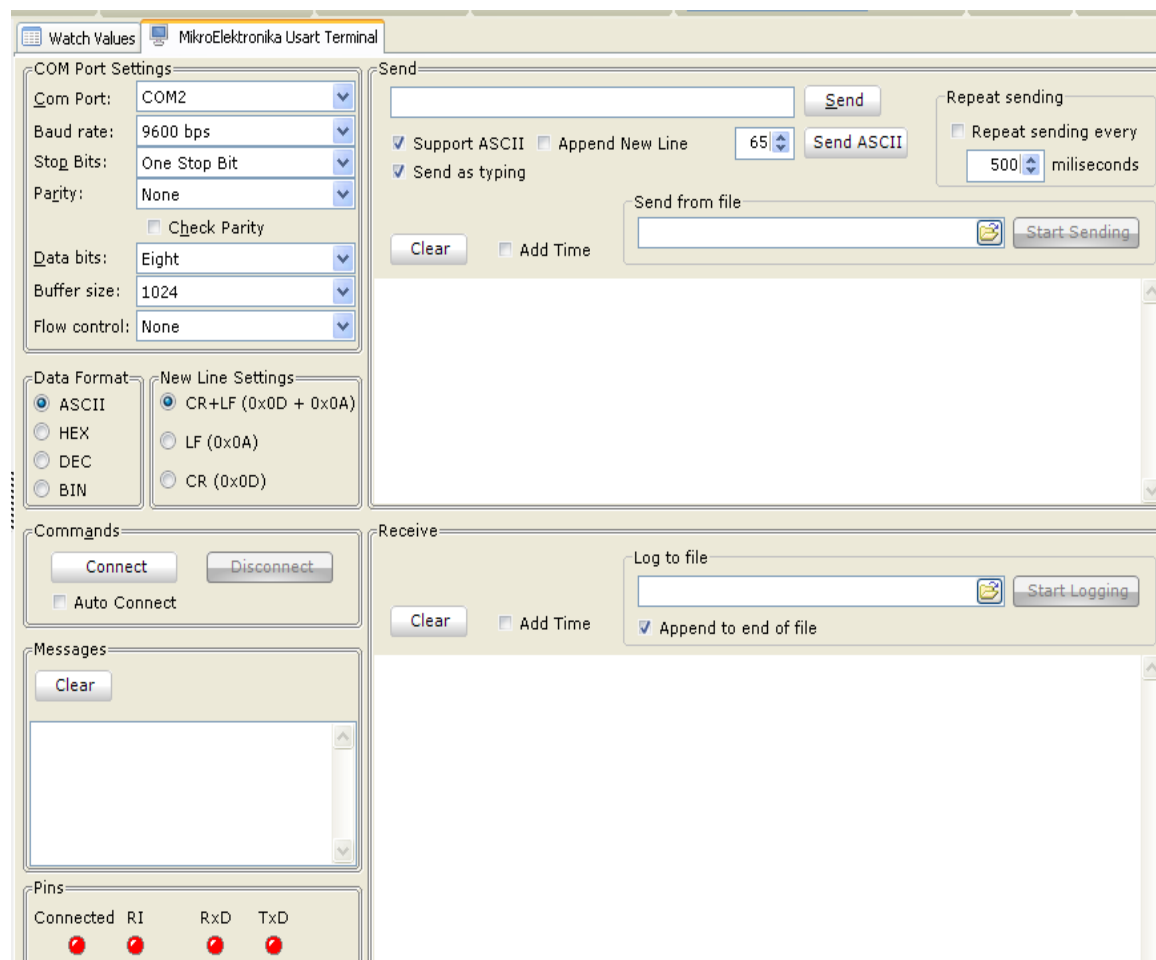
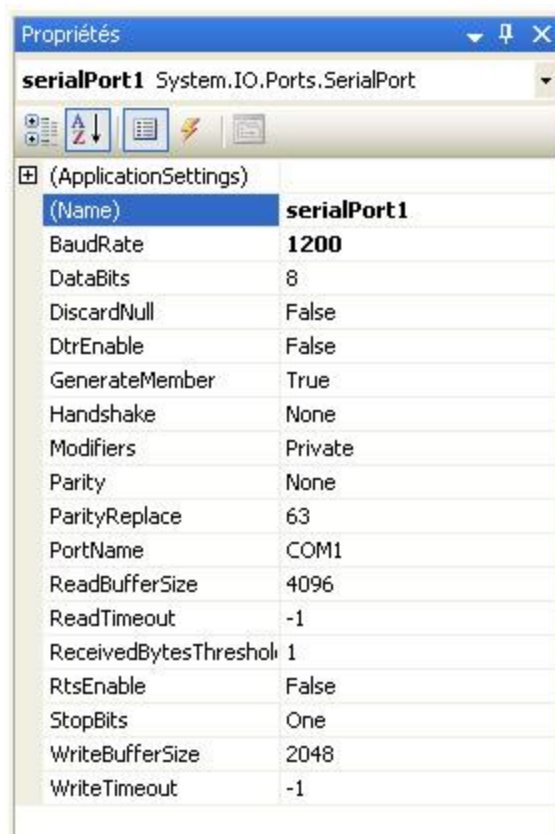


Figure III.4: la fenêtre mikroC terminal

**III.4 Le Visual C# : [18]**

Le Visual C# est un langage de programmation créé par Microsoft qui permet d'exploiter l'entièreté du framework DotNet. Dans notre application, le C# apporte l'avantage de présenter une interface de développement RAD (rapide application design), ce qui nous permet de créer une interface communicante avec un projet électronique. De plus C# est muni de base d'un objet pour communiquer avec le port COM (Voir Figure III.5).



**Figure III.5:** Fenêtre de propriétés du port

Nous pouvons d'une part configurer les différents paramètres et d'autre part créer un événement qui va traiter l'information quand celle-ci arrivera dans le buffer d'entrée de l'ordinateur.

On a réalisé un programme en Visual C# qui permet l'acquisition des résultats (température, luminosité) à travers le port série du PC, puis l'affichage sur ordinateur sous la forme d'un graphe qui présente les changements des valeurs reçues de la carte d'acquisition. Ce programme peut aussi faire un télé-contrôle de la carte d'acquisition (Voir Figure III.6).

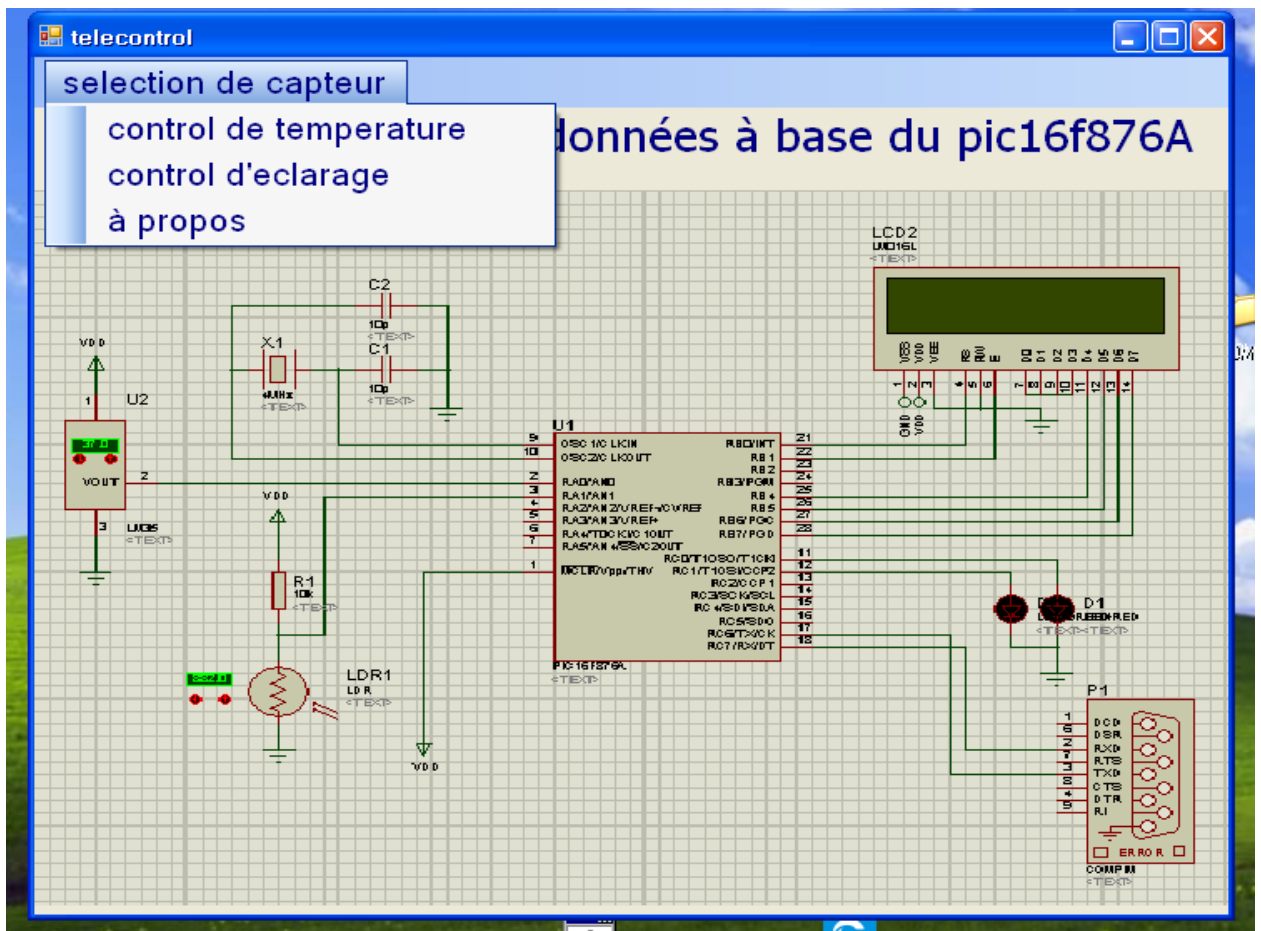


Figure III.6: Fenêtre du programme réalisé avec C#

### III.5 Virtual Serial Port Driver : [19]

C'est un émulateur de port série virtuel qui crée des ports série virtuels et les connecter par deux via un câble virtuel null-modem. Les applications des deux côtés d'une paire pourront échanger les données de façon que tout ce qui est écrit sur le premier port apparaisse sur la seconde et vice versa.

Tous les ports série virtuels fonctionnent et se comportent exactement comme ceux réels, en émulant tous leurs paramètres. On peut créer autant de paires de ports virtuels, donc on aura toujours assez de ports série et pas besoin de matériel supplémentaire surchargeant.

Virtual Serial Port Driver nous offre la possibilité de créer un nombre illimité des ports série virtuels (Voir Figure III.7). Les ports créés sont situés dans le groupe "ports" du Gestionnaire des périphériques, ce qui signifie qu'ils seront visibles et facile à accéder par tout le matériel que vous installez. Nous pourrions indiquer quelle application a un accès au port défini.

Les ports série virtuels créés avec Virtual Serial Ports Driver émulent et supportent toutes les lignes de signaux matériels standards (DTR/DSR, RTS/CTS, RING, ERROR, DCD, etc.) (Voir Figure III.8). Comme dans la connexion null-modem réelle, DTR du port local est connecté à DSR et DCD du port distant, et RTS du port local est connecté à CTS du port distant. En plus, on peut laisser le schéma de brochage par défaut ou choisir une boucle de retour ou bien personnaliser selon notre configuration matérielle.

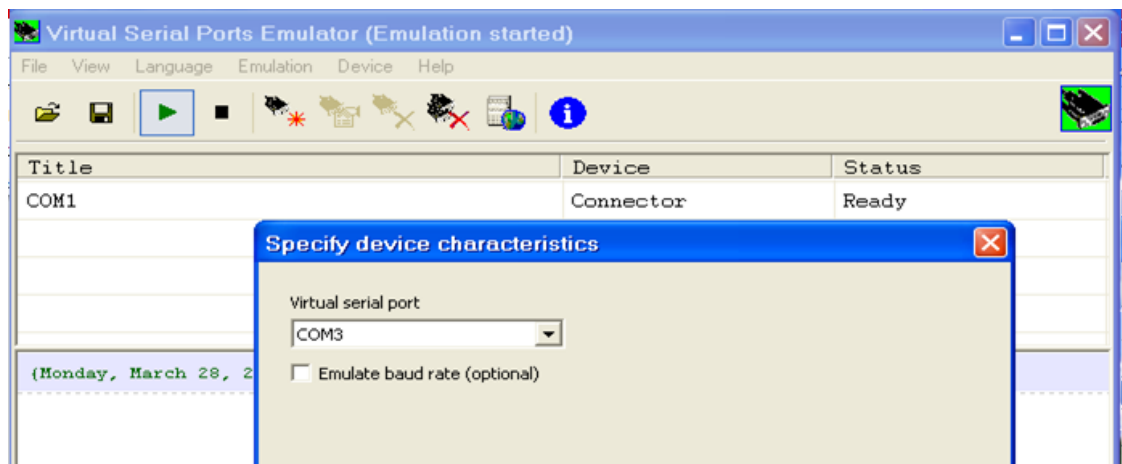


Figure III.7: Fenêtre de sélection de port

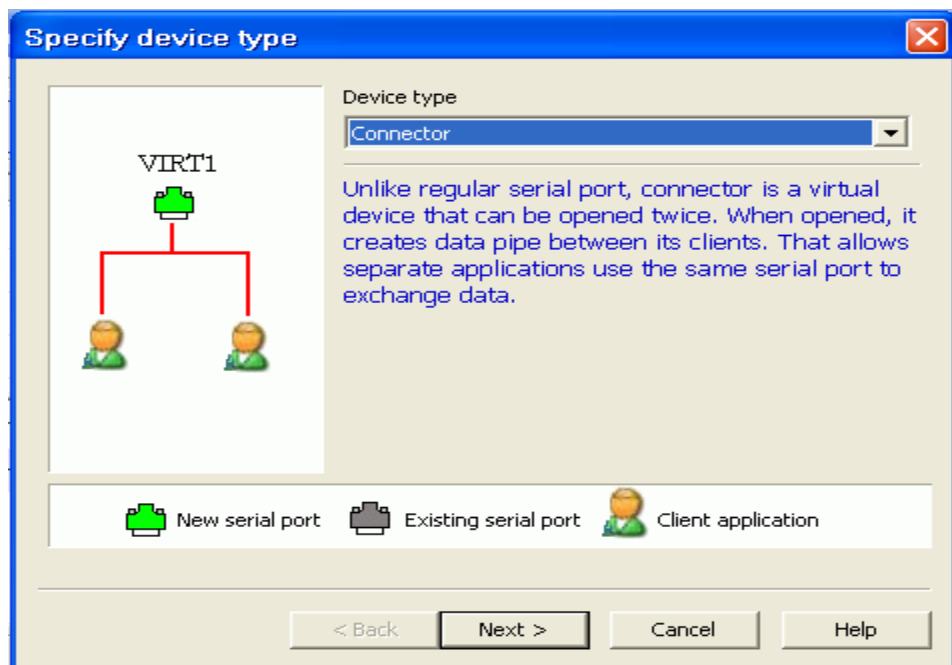


Figure III.8: Fenêtre de sélection d'équipement



### III.6 Logiciel HyperTerminal :

HyperTerminal est un programme qui permet de se connecter à d'autres ordinateurs à l'aide d'un modem ou d'un câble Null Modem. C'est une interface graphique légère capable d'établir une connexion entre ordinateurs par le réseau téléphonique (et un modem) ou câble sur le port série (COM).

La fenêtre "Description de la connexion" s'ouvre, on entre un nom pour la nouvelle connexion (par exemple : télécom univhamalakh) (Voir Figure III.9).



Figure III.9: La fenêtre "Description de la connexion"

On Choisi la valeur par défaut (COM1) dans la fenêtre "Connexion" (Voir Figure III.10)



Figure III.10: La fenêtre "Connexion"

On définit les paramètres du port COM1 (Voir Figure III.11)



**Figure III.11:** Fenêtre de paramètres du port COM

### III.7 Le programmeur de PIC : [20]



**Figure III.12:** Le programmeur LEAPER-48

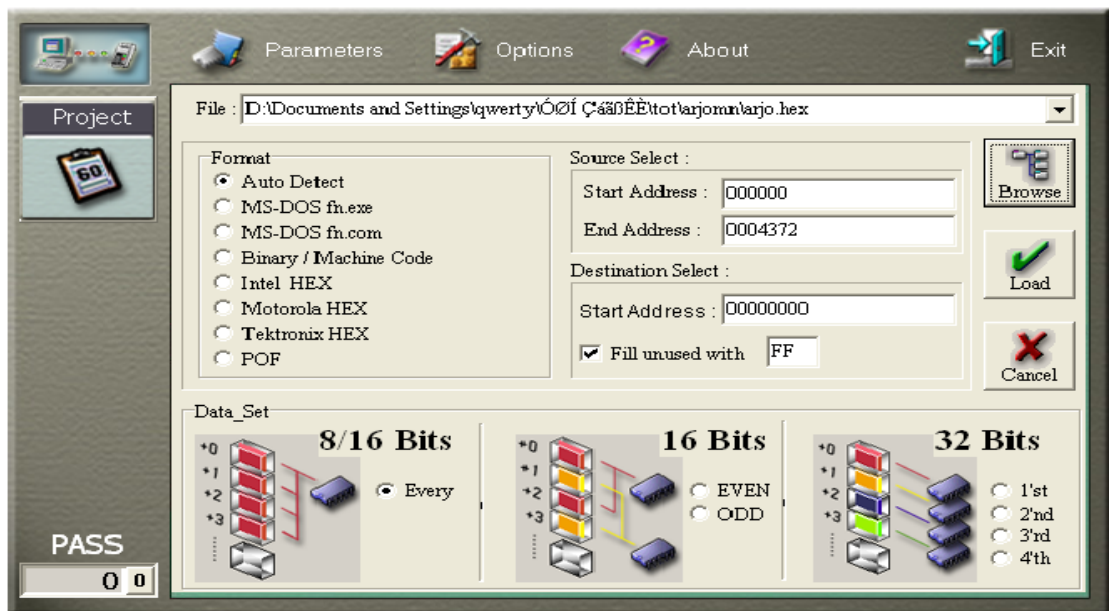
Le LEAPER-48 (Voir Figure III.12) est un programmeur de champ professionnel en format de poche. Le LEAPER-48 supporte EPROM, EEPROM, FLASH EPROM, EEPROM série, microcontrôleurs sans nécessiter de circuits ni d'adaptateurs supplémentaires.

Le LEAPER-48 est un programmeur universel qui se connecte sur PC par l'intermédiaire du port USB. Il est équipé d'un support 48DIP et sa librairie contient plus de 3000

références de composants. Il supporte les microcontrôleurs les plus courants (Microchip PIC, Atmel AVR, Intel/Philips 87Cxx), les Eeproms 27xxx, 27Cxxx), les EEproms (24Cxx, 93Cxx, 28Cxx) ainsi que les flashes Eeproms (28Fxx, 29xxx, 39xxx, 49xxx) jusqu'à une capacité de 64Mb. Pour les flashes Eeproms il est également possible de choisir les secteurs sur lesquels on désire travailler ou non. Le LEAPER-48 supporte les composants low-voltage jusqu'à 2,5V, il est équipé d'un système de détection automatique des composants insérés. Le logiciel fonctionne sous

Le transfert du programme à la mémoire flash du microcontrôleur, PIC 16F876A, s'effectue à l'aide d'un logiciel, Universal IC Writer II (Voir Figure III.13), fournit avec le programmeur LEAPER-48. Les étapes du transfert du fichier code (. HEX) sont les suivant :

- 1- La connexion du programmeur au PC via une fiche USB.
- 2- L'ouverture du programme Universal IC Writer II, et l'insertion du PIC 16f876A.
- 3- Définir le model du microcontrôleur à programmer.
- 4- L'ouverture du fichier source, et la configuration du type d'horloge utilisé dans notre système.
- 5- Le transfert du programme au PIC peut s'effectuer.



**Figure III.13:** Universal IC Writer II

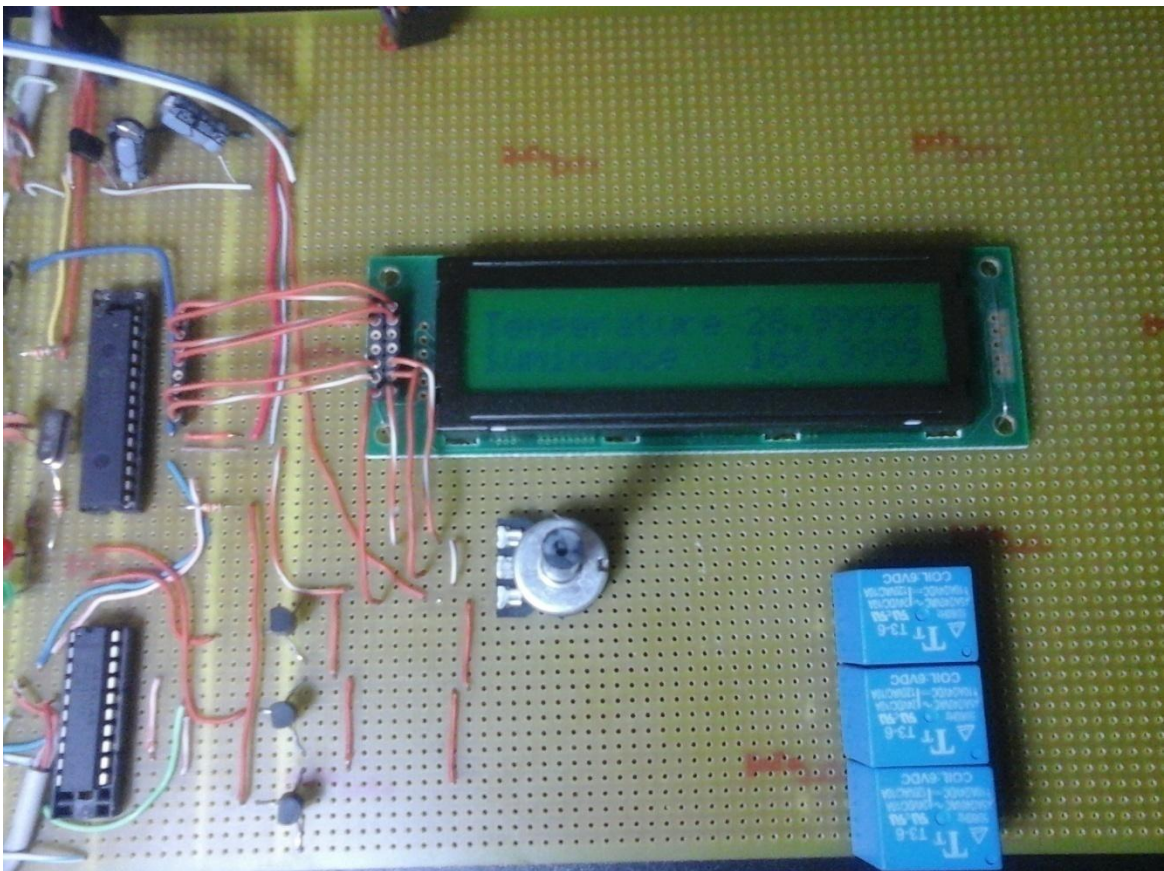
**Chapitre IV:**  
**Conception et réalisation de**  
**la carte**

## **Chapitre IV**

### ***Conception et réalisation de la carte***

#### **IV.1 Introduction :**

Le présent chapitre présente les différentes étapes de la conception de la carte d'acquisition de données à base d'un PIC 16f876A (Voir Figure IV.1).



**Figure IV.1:** La carte d'acquisition de données

## IV.2 Schéma synoptique de la carte d'acquisition :

(Voir Figure IV.2)

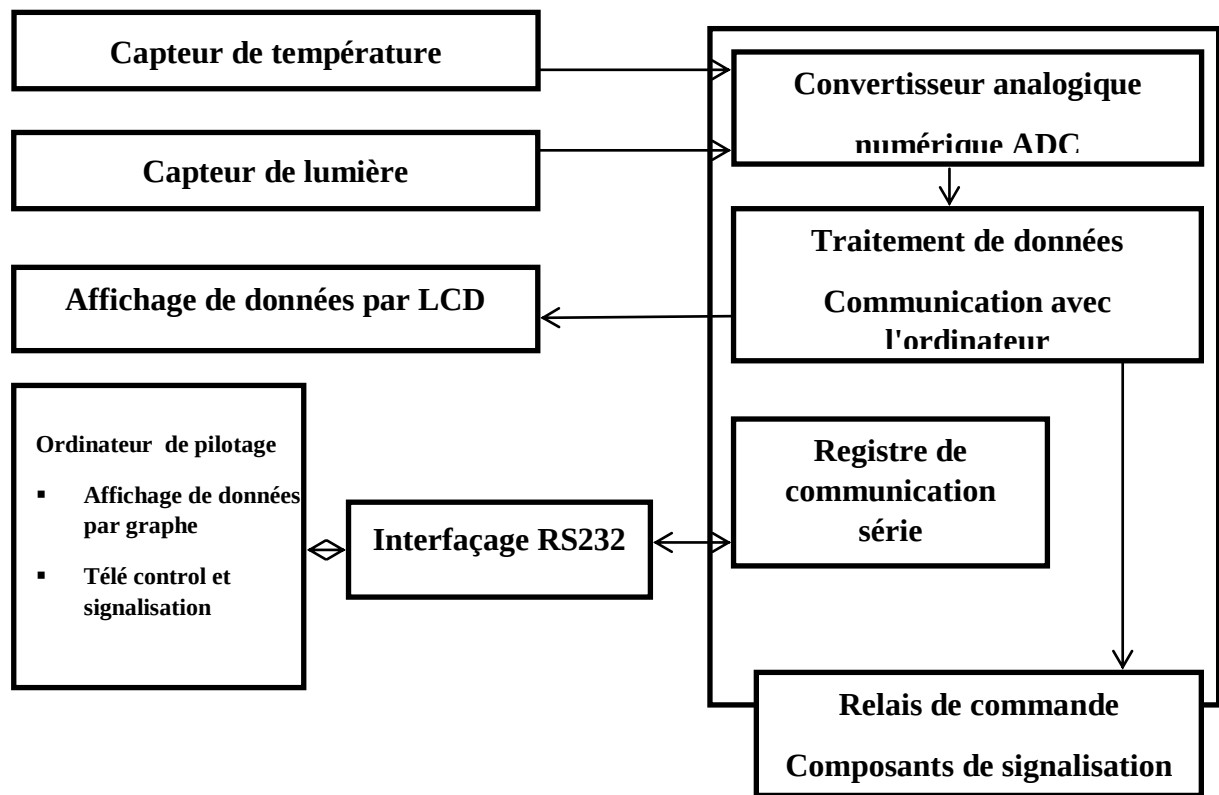


Figure IV.2: Schéma synoptique

Les capteurs de température et de lumière génèrent des signaux analogiques proportionnels respectivement à la température et la lumière. Ces données analogiques sont converties en données numériques par le convertisseur ADC du PIC 16F876A (Voir Figure IV.3).

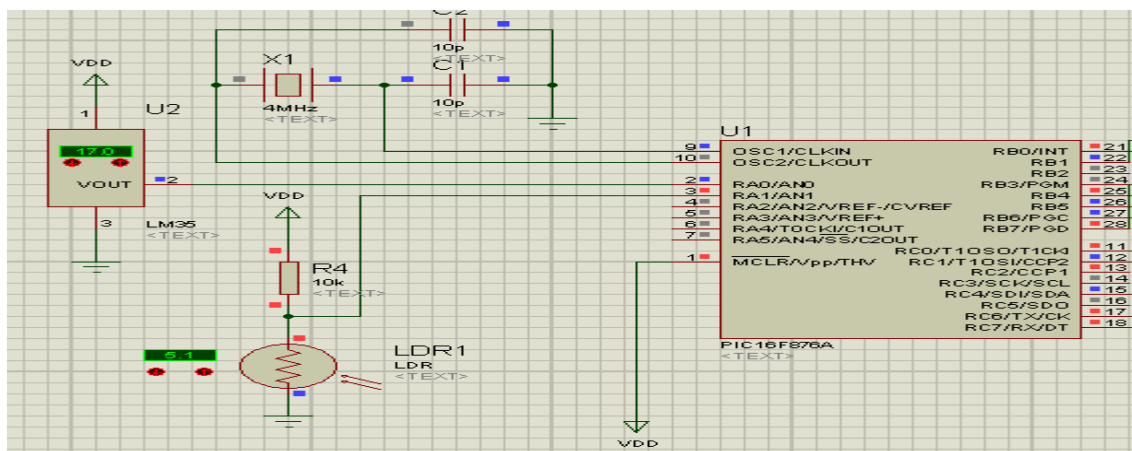


Figure IV.3: schéma électrique de capteurs

Après traitement de ces données, elles sont d'une part affichées par l'afficheur LCD (Voir Figure IV.4) et d'autre part à l'ordinateur de pilotage via un interface RS232 (Voir Figure IV.5).

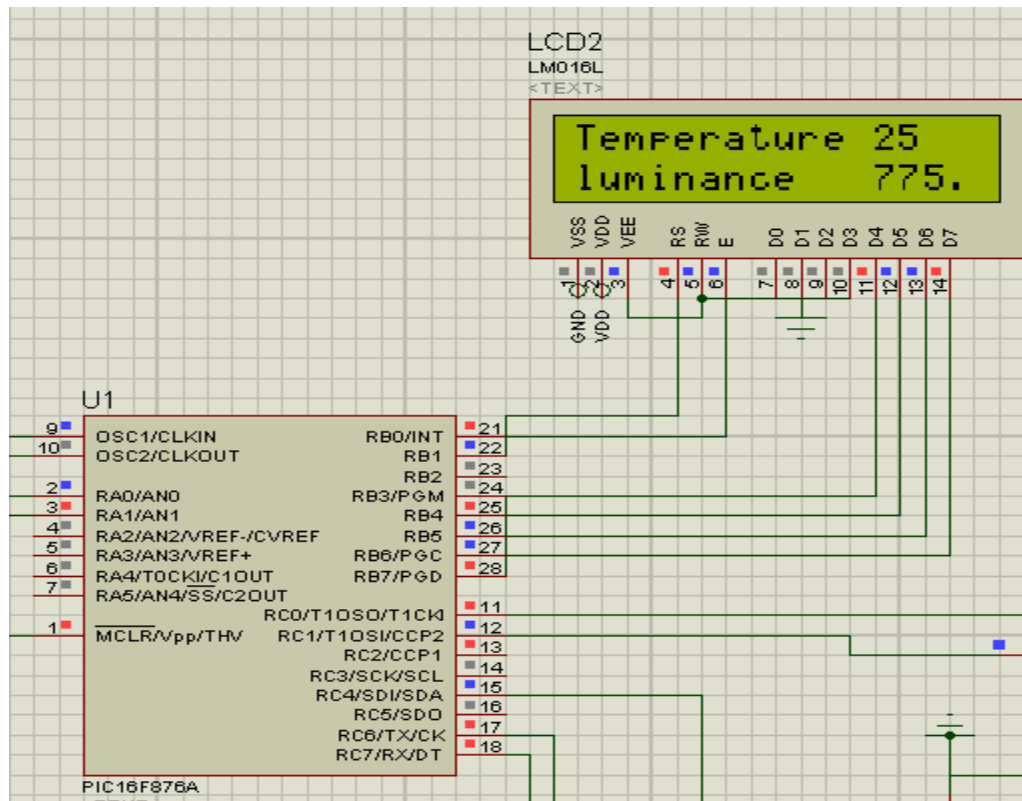


Figure IV.4: schéma électrique de l'afficheur LCD

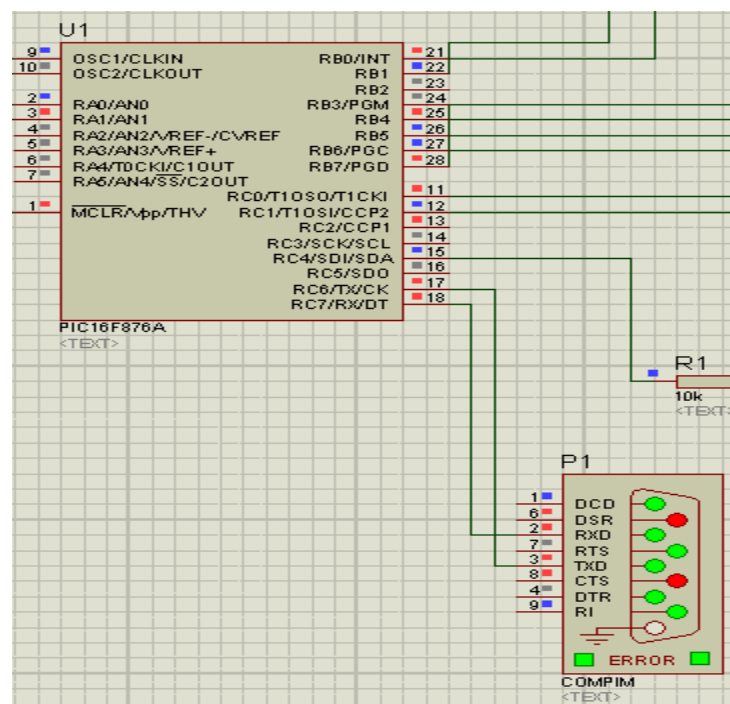
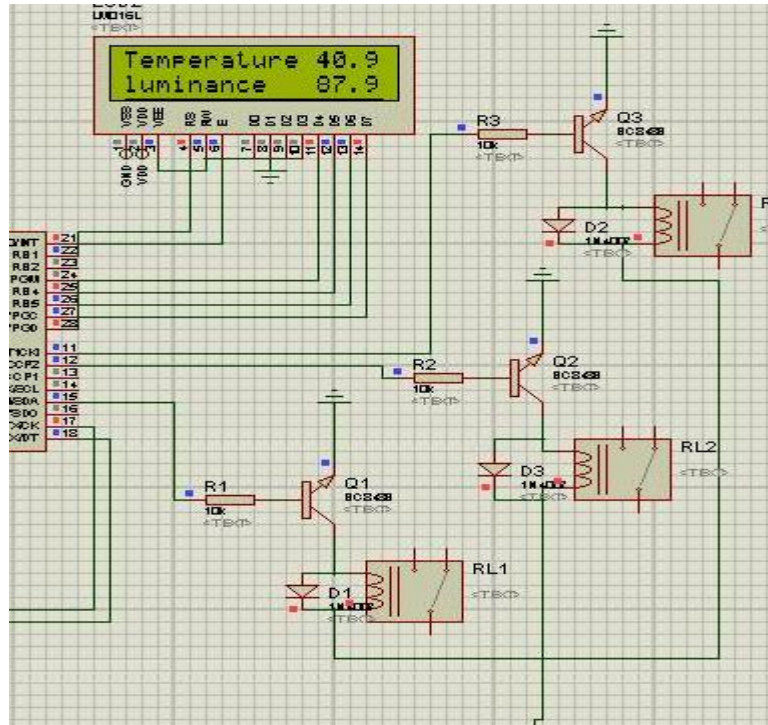


Figure IV.5: schéma électrique du port COM



Les relais de commande sont activés soit par le PIC soit par l'ordinateur de pilotage (Voir Figure IV.6).



**Figure IV.6:** schéma électrique de relais



## IV.3 Schéma électrique de la carte d'acquisition :

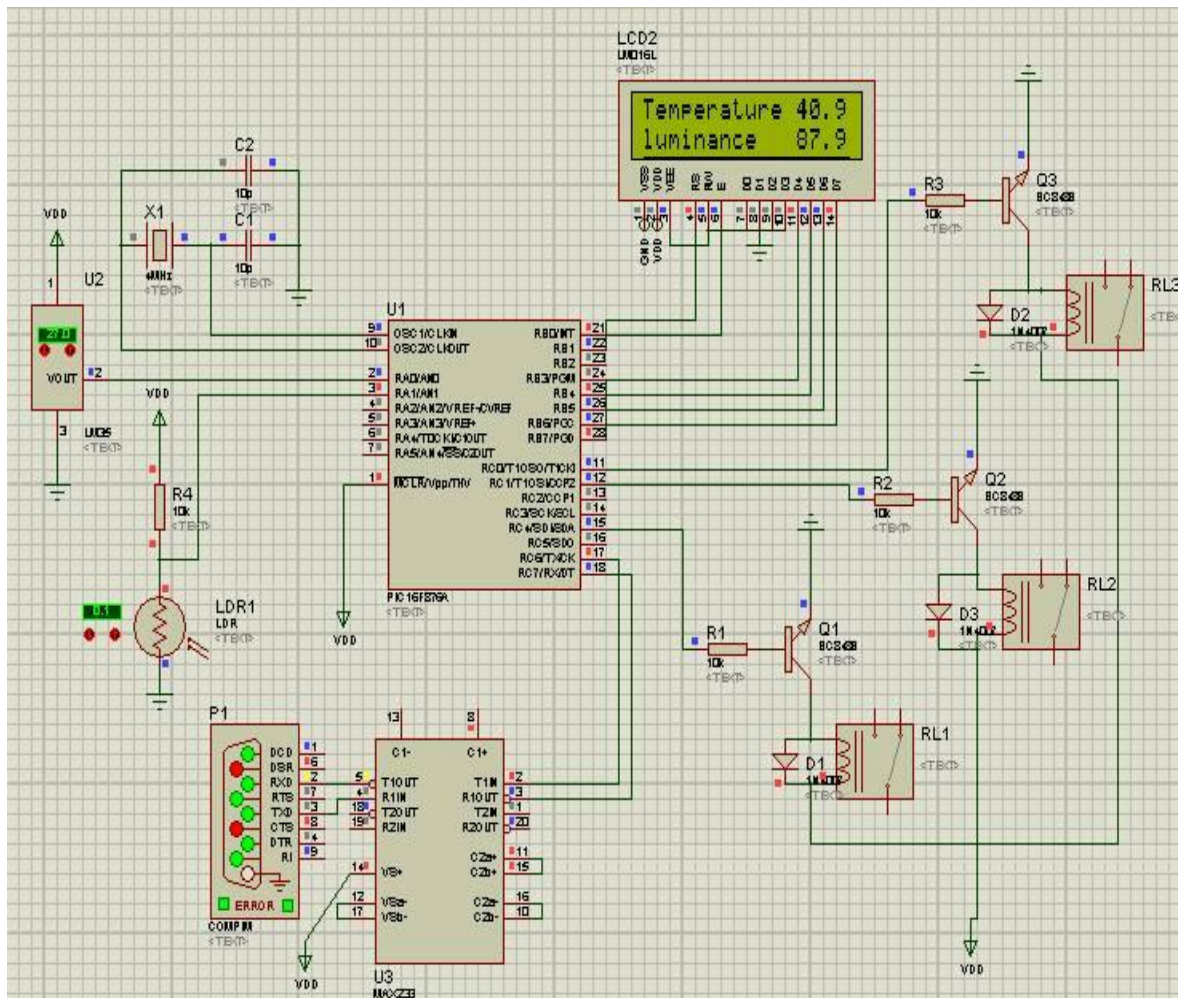


Figure IV.7: schéma électrique de la carte d'acquisition

## IV.4 Liste des composants :

(Voir Figure IV.7)

- PIC16F876A
- Afficheur LCD LM016
- Capteur de température LM35
- Capteur de lumière LDR
- MAX233A
- Quartz 4Mhz
- Transistors Q1=Q2=Q3=BC848B
- D1=D2=D3=1N4007

- Condensateurs  $C1=C2=10\text{pF}$
- Resistances  $R1=R2=R3=R4=10\text{K}\Omega$
- 03 Relais

#### **IV.5 Programmation :**

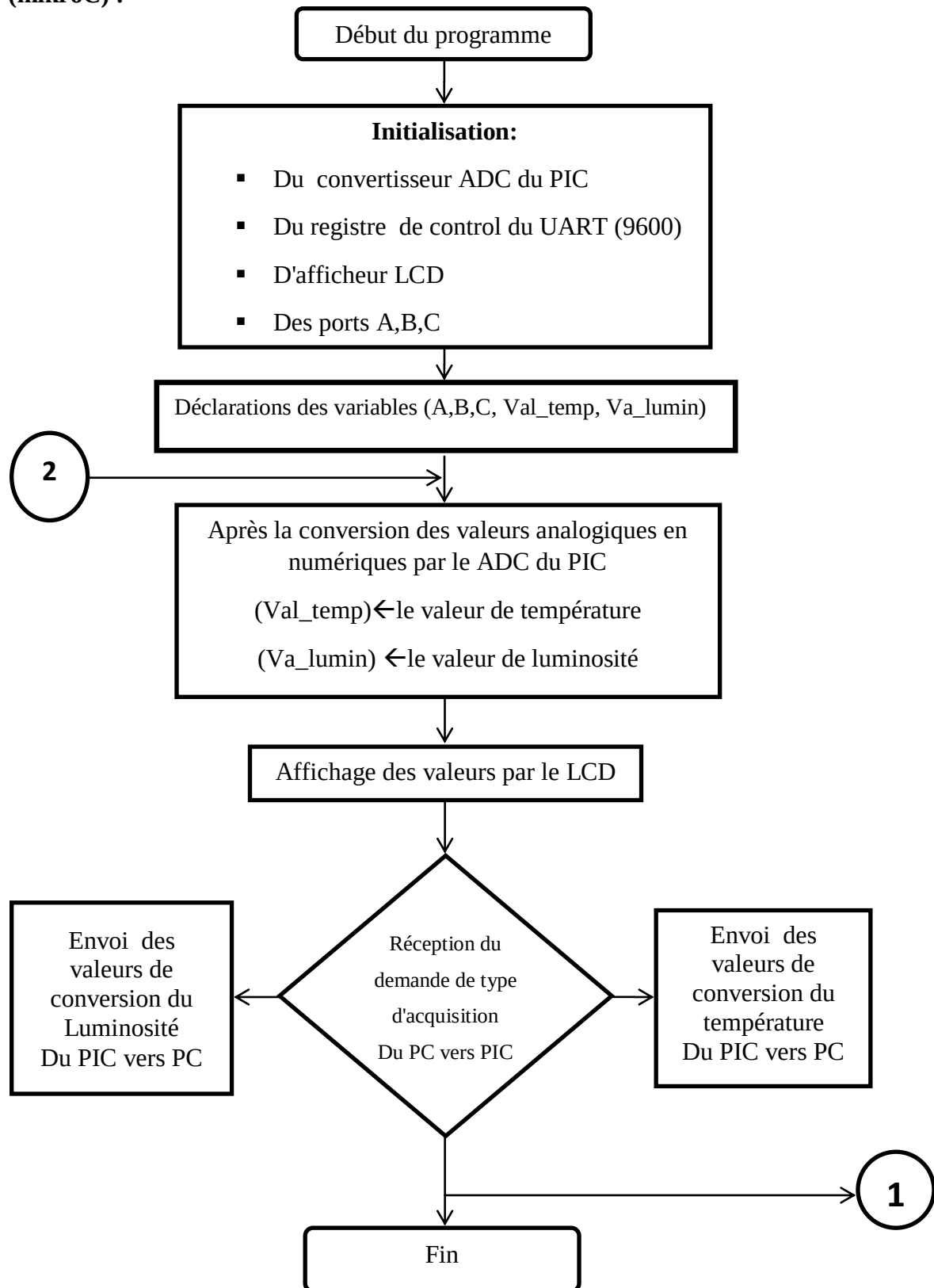
Dans cette partie, on donne trois organigrammes :

Organigramme du programme pilote de la carte d'acquisition côté PIC (mikroC)

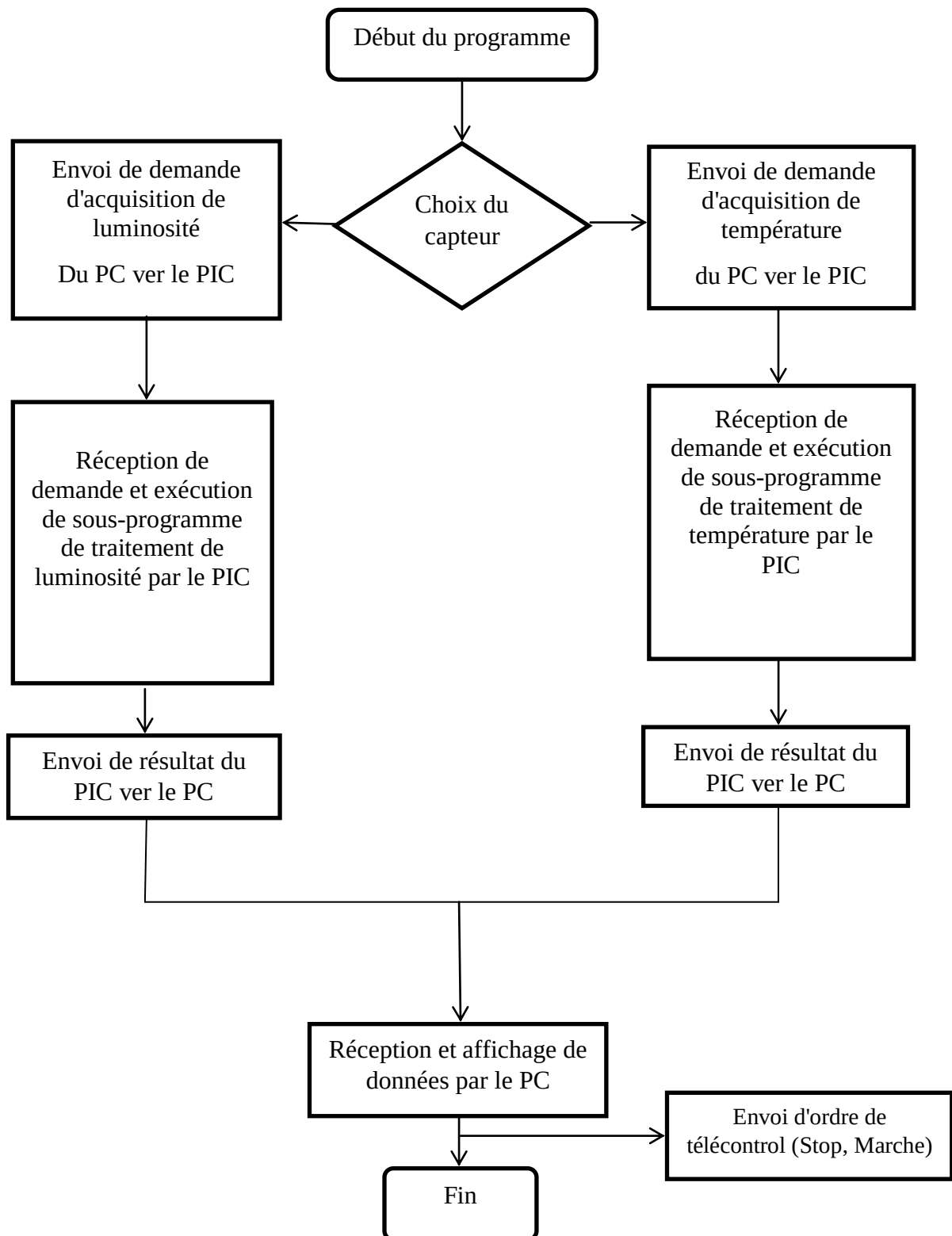
Organigramme du programme pilot de la carte d'acquisition côté PC(C#)

Organigramme du sous-programme de commande des relais

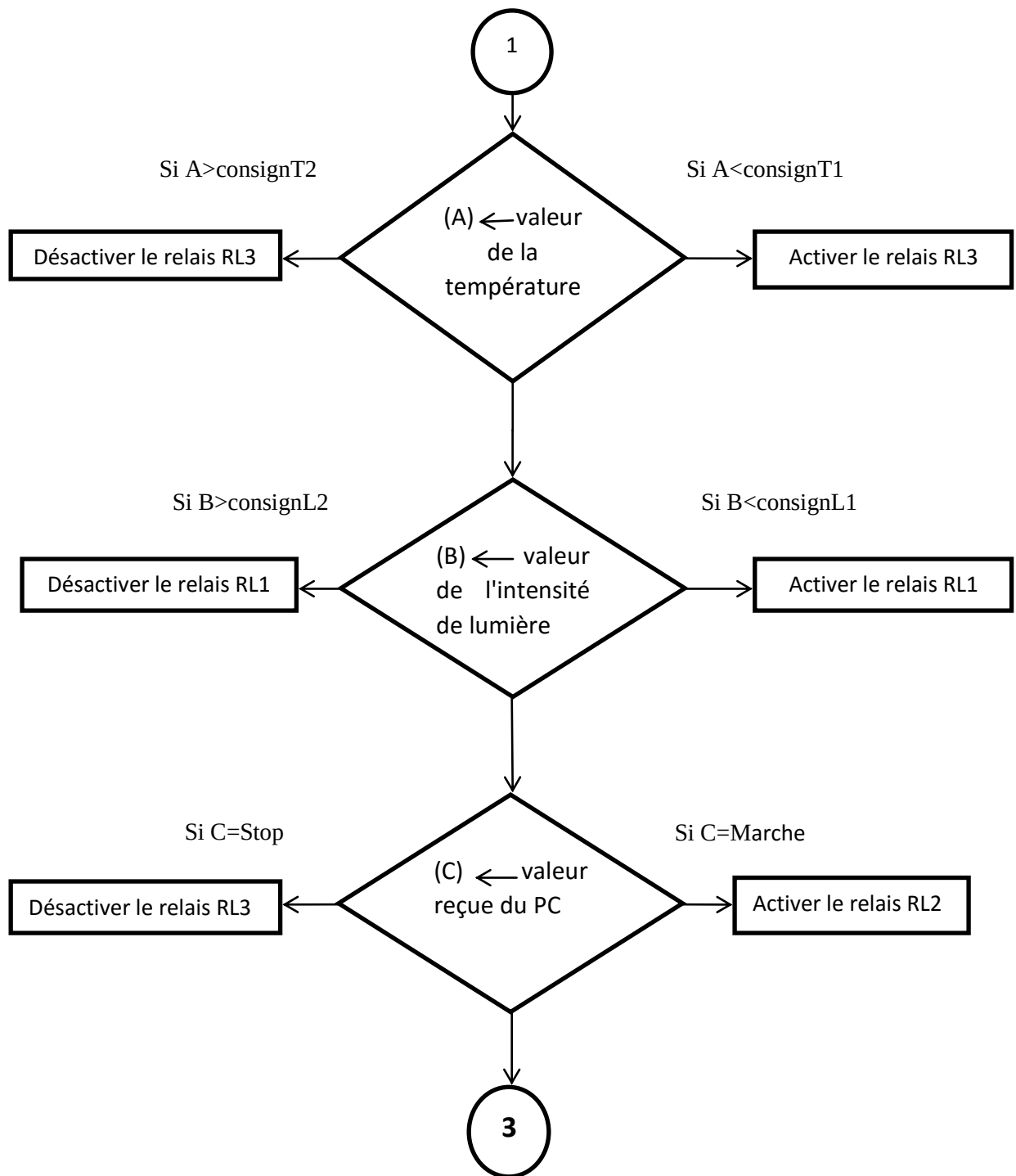
#### IV.5.1 Organigramme du programme pilote de la carte d'acquisition côté PIC (mikroC) :



## IV.5.2 Organigramme du programme pilot de la carte d'acquisition cotée PC(C#):



## IV.5.3 Organigramme du sous-programme de commande des relais:



## IV.6 Résultats de tests :

Après la mise en marche de la carte d'acquisition, la valeur de la température ainsi que celle de la luminance sont illustrées sur l'afficheur LCD (Voir Figure IV.8) :

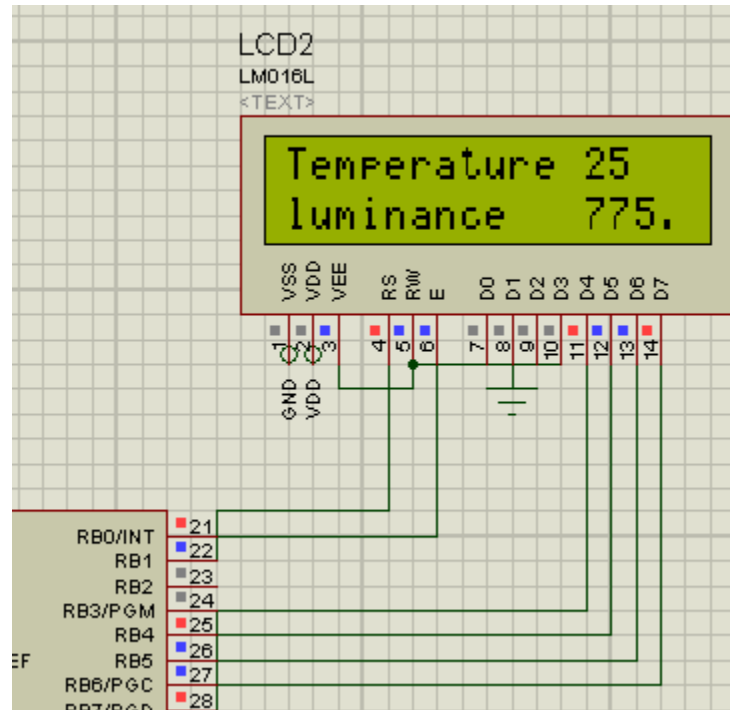


Figure IV.8: L'afficheur LCD

On lance le programme d'interfaçage pilote au niveau de l'ordinateur (Voir Figure IV.9).

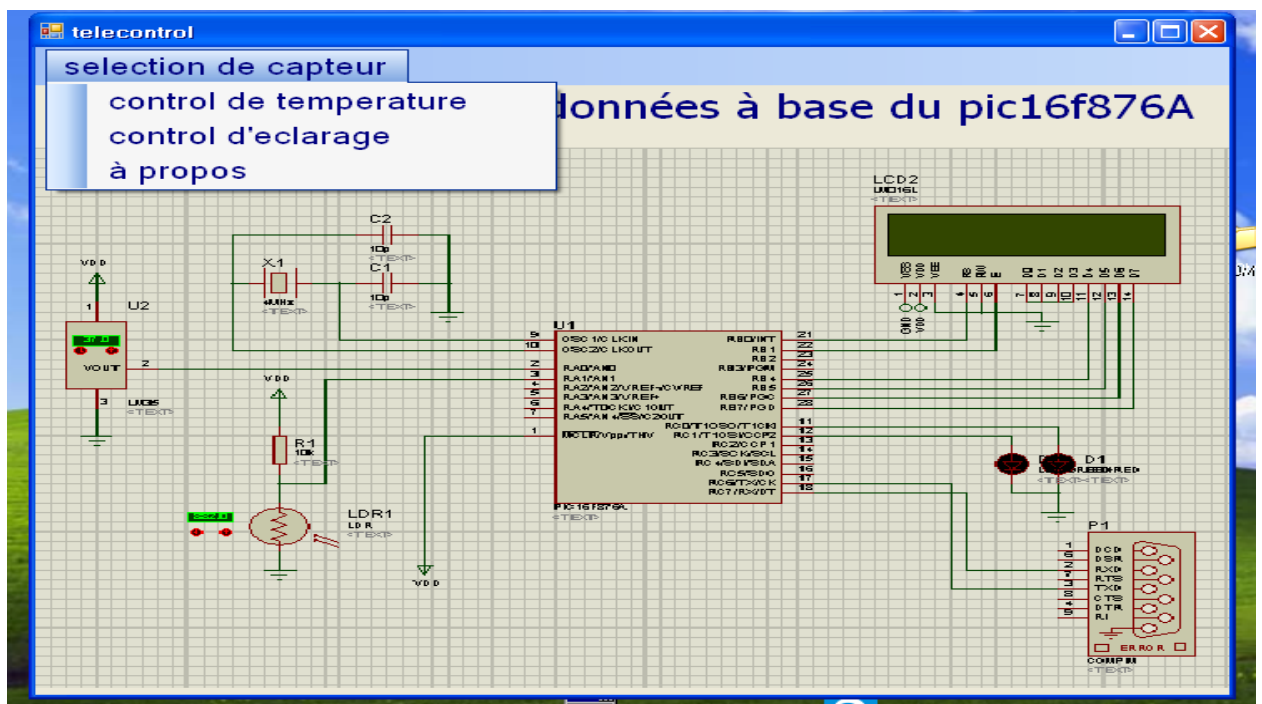
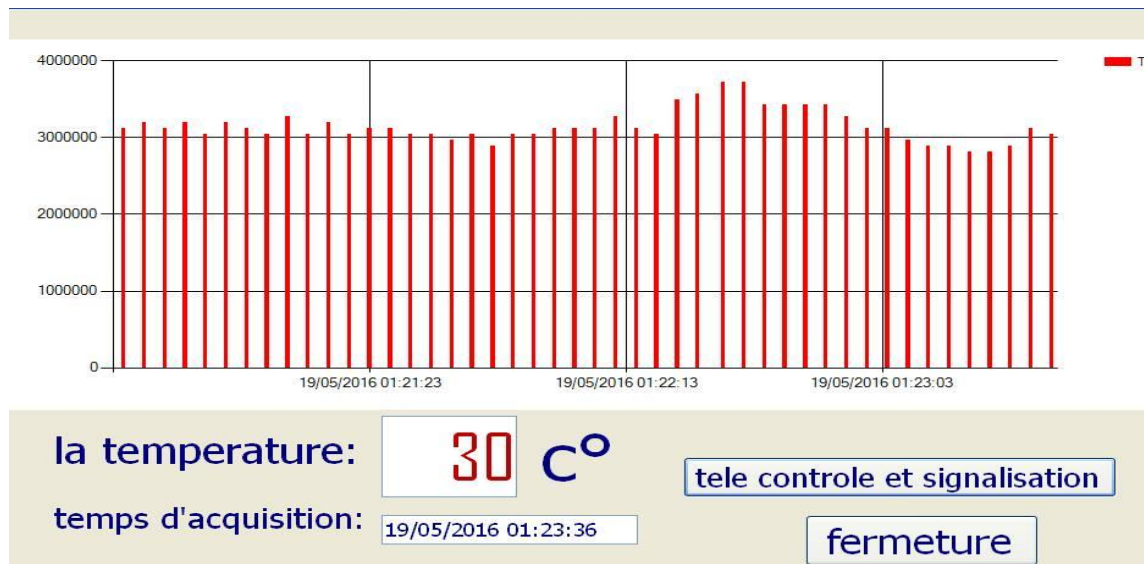


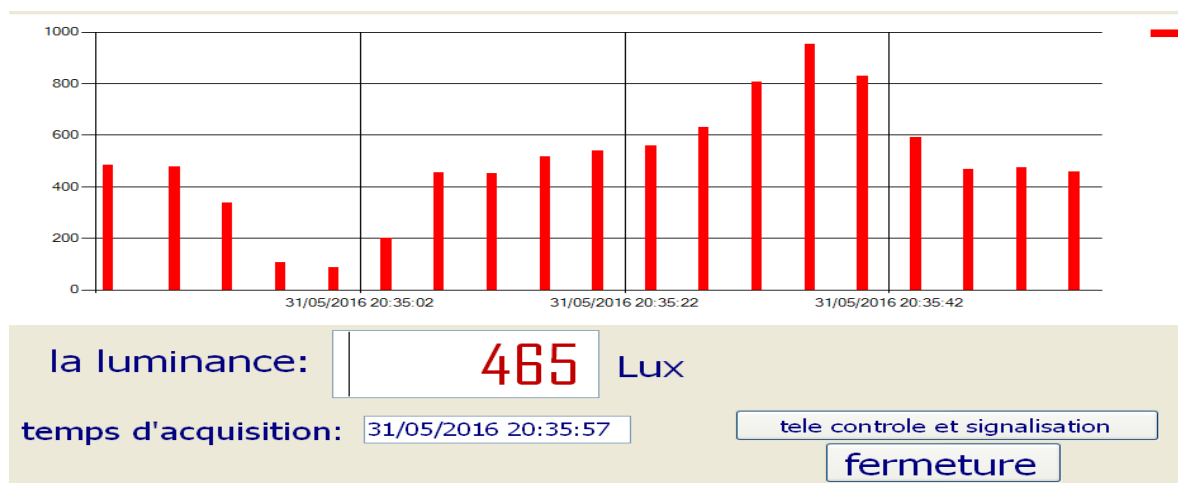
Figure IV.9: Le programme d'interfaçage

Si on choisit du menu "sélection de capteur" l'option "contrôle de température", le graphe de la température s'affiche montrant les différentes valeurs de température reçues (Voir Figure IV.10).



**Figure IV.10:** Le graphe de la température

Si on choisit maintenant du menu "sélection de capteur", l'option "contrôle d'éclairage", le graphe de la luminosité s'affiche montrant les différentes valeurs de lumière reçues (Voir Figure IV.11).



**Figure IV.11:** Le graphe de la lumière

Le bouton de télé-control et signalisation permet d'activer un LED ou un relais suivant la nécessité.

**Le contrôle de la température :**

- Si la température est inférieure a une valeur consigne, le relais RL3 est activé.
- Si la température est supérieure a une valeur consigne, le relais RL3 est désactivé.

**Le contrôle d l'intensité de lumière :**

- Si l'intensité de lumière est inférieure a une valeur consigne, le relais RL1 est activé.
- Si l'intensité de lumière est supérieure a une valeur consigne, le relais RL1 est désactivé.

**Le Télé contrôle :**

- Si on clique sur le bouton marche, le relais RL2 est mis en marche.
- Si on clique sur le bouton arrêt, le relais RL2 est arrêté.

(Voir Figure IV.6)



## **Conclusion générale**

## ***Conclusion générale***

La réalisation de la carte d'acquisition de données est effectuée en deux parties :

**Partie électronique :** c'est la réalisation pratique de la carte électronique pour la mesure de la température et l'intensité de lumière, la communication avec le PC et la commande de relais.

**Partie informatique :** c'est la programmation de la carte d'acquisition pour lecture et affichage des résultats et aussi la transmission des données et l'exécution des commandes reçues par le PC. La réalisation d'une interface graphique en C# permet la réception des données, l'affichage des résultats et aussi la commande de la carte d'acquisition.

Notre projet a été une expérience très bénéfique. Il nous a permis de mettre le premier pas dans le monde pratique des systèmes embarqués. L'élaboration de ce travail, nous a aidé à approfondir nos connaissances théoriques et à acquérir une bonne expérience au niveau de la réalisation pratique.

En fin nous espérons que notre travail va être amélioré au futur par d'autres étudiants. Comme perspectives nous proposons :

- La communication via le port USB ou la technique Wifi.
- Le contrôle de la carte à distance via Internet.
- L'enregistrement des résultats sous forme d'une base de données.

# **Bibliographie**

## ***Bibliographie***

- [1] <http://www.les-electroniciens.com/sites/default/files/cours/capteurs.pdf> 2016
- [2] [http://geea.org/IMG/doc/les\\_capteurs.doc](http://geea.org/IMG/doc/les_capteurs.doc) 2016
- [3] <https://www.carnetdumaker.net/articles/mesurer-une-temperature-avec-un-capteur-lm35-et-une-carte-arduino-genuino/> 2016
- [4] <http://wiki.mchobby.be/index.php?title=Photo-r%C3%A9sistance> 2016
- [5] <https://tpe2013stenay.wordpress.com/179-2/4-les-capteurs-de-lumiere-ou-photoresistances/> 2016
- [6] <http://sti.discip.ac-caen.fr/sites/sti.discip.ac-caen.fr/IMG/pdf/picbasic1s.pdf> 2016
- [7] [http://pf-mh.uvt.rnu.tn/491/1/Conception\\_et\\_r%C3%A9alisation\\_d\\_un\\_enregistreur\\_de\\_donn%C3%A9es.pdf](http://pf-mh.uvt.rnu.tn/491/1/Conception_et_r%C3%A9alisation_d_un_enregistreur_de_donn%C3%A9es.pdf) 2016
- [8] <http://www.oumnad.123.fr/Microcontrôleurs/PIC16F877.pdf> 2016
- [9] <http://blewando.dlinkddns.com/cours/sin/006/fichier.pdf> 2016
- [10] <http://www.technologuepro.com/TP-miniprojet-electronique/miniprojet-2-MICROCONTROLEURS-PIC-MICROSHIP.pdf> 2016
- [11] [http://foad.ensicaen.fr/pluginfile.php/11646/mod\\_resource/content/3/uart\\_2015x.pdf](http://foad.ensicaen.fr/pluginfile.php/11646/mod_resource/content/3/uart_2015x.pdf) 2016
- [12] <http://www.roboticus.org/electronique/31-rs232/4-description-de-lrs232-et-mise-en-place-dun-max232.html> 2016
- [13] <https://www.aurel32.net/elec/lcd.php> 2016
- [14] [http://www.siloged.fr/cours/CR\\_RELAIS.PDF?PHPSESSID=69385f765ebd4338991c8625cbfcfd2](http://www.siloged.fr/cours/CR_RELAIS.PDF?PHPSESSID=69385f765ebd4338991c8625cbfcfd2) 2016
- [15] [https://wiki.mdl29.net/lib/exe/fetch.php?media=elec:cours\\_transistor.pdf](https://wiki.mdl29.net/lib/exe/fetch.php?media=elec:cours_transistor.pdf) 2016
- [16] Mr. SAADI Ramzy, Mr. GUESBAYA Tahar; "Réalisation de carte à microcontrôleur pour le contrôle de bras manipulateur via un pc"; Mémoire de fin d'études en vue d'obtention diplôme MASTER; Faculté des sciences et de la technologie - Département de génie électrique Université Mohamed Khider Biskra 2010
- [17] V. TOURTCHINE; "Programmation en mikroC. Application pour les microcontrôleurs de la famille PIC"; faculté des sciences - département physique université M'hamed Bogara de Boumerdes; 2012
- [18] <http://www.roboticus.org/electronique/rs232/7-exemple-dapplication-un-voltmetre-sur-port-serie.html> 2016
- [19] <http://www.eltima.com/fr/products/vspdxp/> 2016
- [20] <http://www.transplanet.fr/boutique/pdf/LEAPER-48.pdf> 2016

# ***Annexes***



# PIC16F87XA

## 28/40/44-Pin Enhanced Flash Microcontrollers

### Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

### High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input  
DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory,  
Up to 368 x 8 bytes of Data Memory (RAM),  
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin  
PIC16CXXX and PIC16FXXX microcontrollers

### Peripheral Features:

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

### Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital  
Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference  
(VREF) module
  - Programmable input multiplexing from device  
inputs and internal voltage reference
  - Comparator outputs are externally accessible

### Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash  
program memory typical
- 1,000,000 erase/write cycle Data EEPROM  
memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™)  
via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC  
oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

### CMOS Technology:

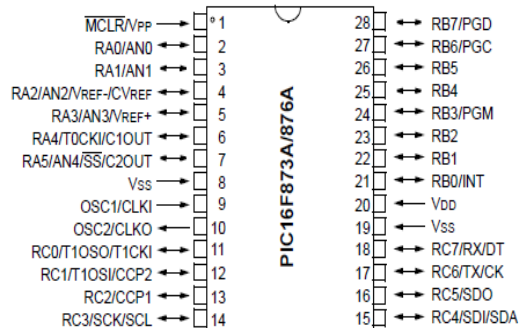
- Low-power, high-speed Flash/EEPROM  
technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

| Device     | Program Memory |                               | Data<br>SRAM<br>(Bytes) | EEPROM<br>(Bytes) | I/O | 10-bit<br>A/D (ch) | CCP<br>(PWM) | MSSP |               | USART | Timers<br>8/16-bit | Comparators |
|------------|----------------|-------------------------------|-------------------------|-------------------|-----|--------------------|--------------|------|---------------|-------|--------------------|-------------|
|            | Bytes          | # Single Word<br>Instructions |                         |                   |     |                    |              | SPI  | Master<br>I²C |       |                    |             |
| PIC16F873A | 7.2K           | 4096                          | 192                     | 128               | 22  | 5                  | 2            | Yes  | Yes           | Yes   | 2/1                | 2           |
| PIC16F874A | 7.2K           | 4096                          | 192                     | 128               | 33  | 8                  | 2            | Yes  | Yes           | Yes   | 2/1                | 2           |
| PIC16F876A | 14.3K          | 8192                          | 368                     | 256               | 22  | 5                  | 2            | Yes  | Yes           | Yes   | 2/1                | 2           |
| PIC16F877A | 14.3K          | 8192                          | 368                     | 256               | 33  | 8                  | 2            | Yes  | Yes           | Yes   | 2/1                | 2           |

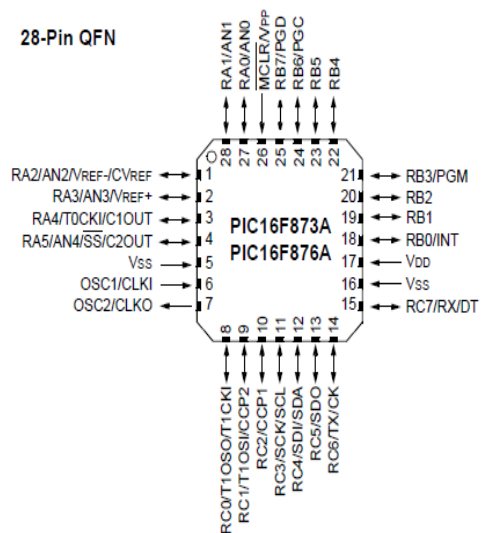
# PIC16F87XA

## Pin Diagrams

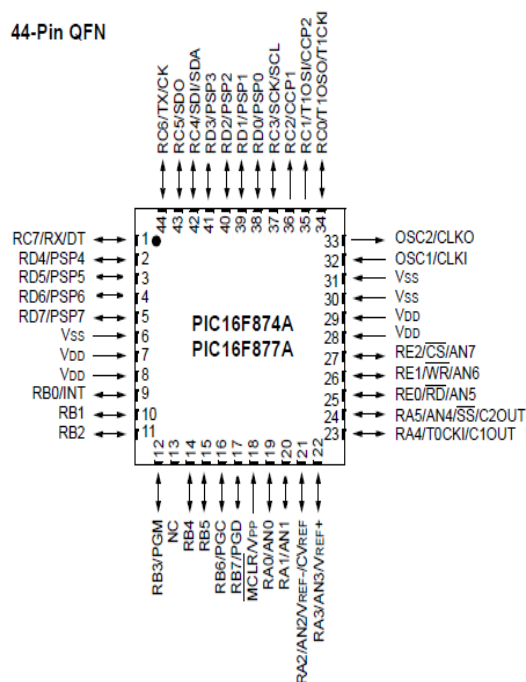
### 28-Pin PDIP, SOIC, SSOP



### 28-Pin QFN



### 44-Pin QFN



## PIC16F87XA

### 1.0 DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages. All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A
- The 28-pin devices have three I/O ports, while the 40/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

The available features are summarized in Table 1-1. Block diagrams of the PIC16F873A/876A and PIC16F874A/877A devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2 and Table 1-3.

Additional information may be found in the PICmicro® Mid-Range Reference Manual (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip web site. The Reference Manual should be considered a complementary document to this data sheet and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

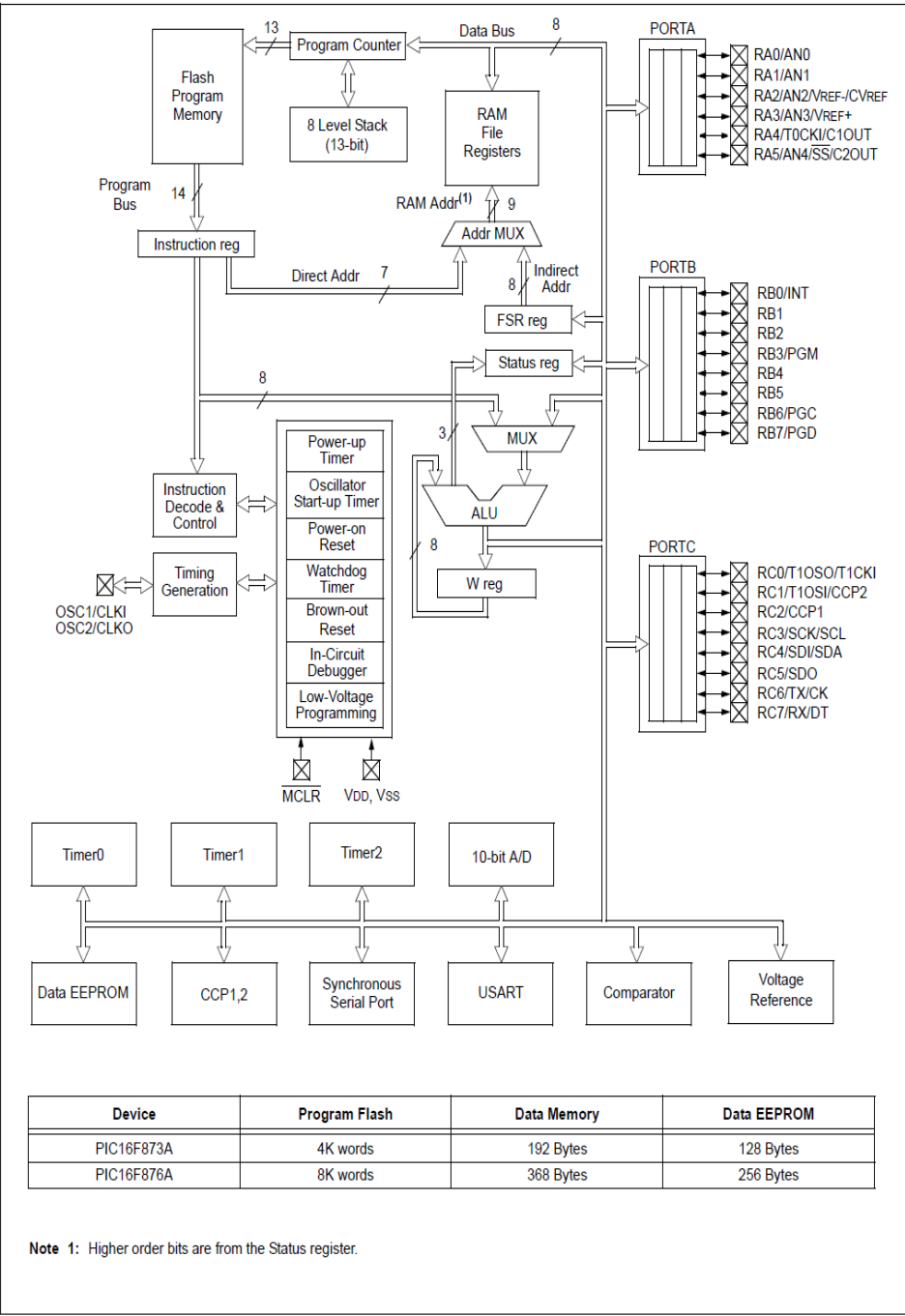
TABLE 1-1: PIC16F87XA DEVICE FEATURES

| Key Features                        | PIC16F873A                                              | PIC16F874A                                              | PIC16F876A                                              | PIC16F877A                                              |
|-------------------------------------|---------------------------------------------------------|---------------------------------------------------------|---------------------------------------------------------|---------------------------------------------------------|
| Operating Frequency                 | DC – 20 MHz                                             | DC – 20 MHz                                             | DC – 20 MHz                                             | DC – 20 MHz                                             |
| Resets (and Delays)                 | POR, BOR (PWRT, OST)                                    | POR, BOR (PWRT, OST)                                    | POR, BOR (PWRT, OST)                                    | POR, BOR (PWRT, OST)                                    |
| Flash Program Memory (14-bit words) | 4K                                                      | 4K                                                      | 8K                                                      | 8K                                                      |
| Data Memory (bytes)                 | 192                                                     | 192                                                     | 368                                                     | 368                                                     |
| EEPROM Data Memory (bytes)          | 128                                                     | 128                                                     | 256                                                     | 256                                                     |
| Interrupts                          | 14                                                      | 15                                                      | 14                                                      | 15                                                      |
| I/O Ports                           | Ports A, B, C                                           | Ports A, B, C, D, E                                     | Ports A, B, C                                           | Ports A, B, C, D, E                                     |
| Timers                              | 3                                                       | 3                                                       | 3                                                       | 3                                                       |
| Capture/Compare/PWM modules         | 2                                                       | 2                                                       | 2                                                       | 2                                                       |
| Serial Communications               | MSSP, USART                                             | MSSP, USART                                             | MSSP, USART                                             | MSSP, USART                                             |
| Parallel Communications             | —                                                       | PSP                                                     | —                                                       | PSP                                                     |
| 10-bit Analog-to-Digital Module     | 5 input channels                                        | 8 input channels                                        | 5 input channels                                        | 8 input channels                                        |
| Analog Comparators                  | 2                                                       | 2                                                       | 2                                                       | 2                                                       |
| Instruction Set                     | 35 Instructions                                         | 35 Instructions                                         | 35 Instructions                                         | 35 Instructions                                         |
| Packages                            | 28-pin PDIP<br>28-pin SOIC<br>28-pin SSOP<br>28-pin QFN | 40-pin PDIP<br>44-pin PLCC<br>44-pin TQFP<br>44-pin QFN | 28-pin PDIP<br>28-pin SOIC<br>28-pin SSOP<br>28-pin QFN | 40-pin PDIP<br>44-pin PLCC<br>44-pin TQFP<br>44-pin QFN |



# PIC16F87XA

FIGURE 1-1: PIC16F873A/876A BLOCK DIAGRAM



## **+5V-Powered, Multichannel RS-232 Drivers/Receivers**

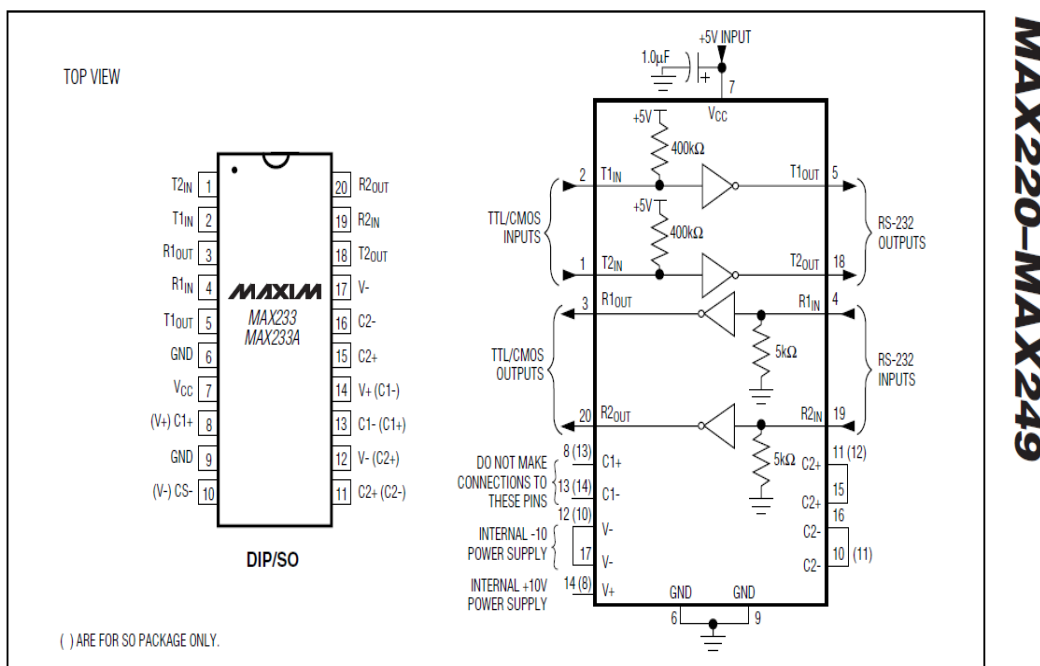


Figure 11. MAX233/MAX233A Pin Configuration and Typical Operating Circuit

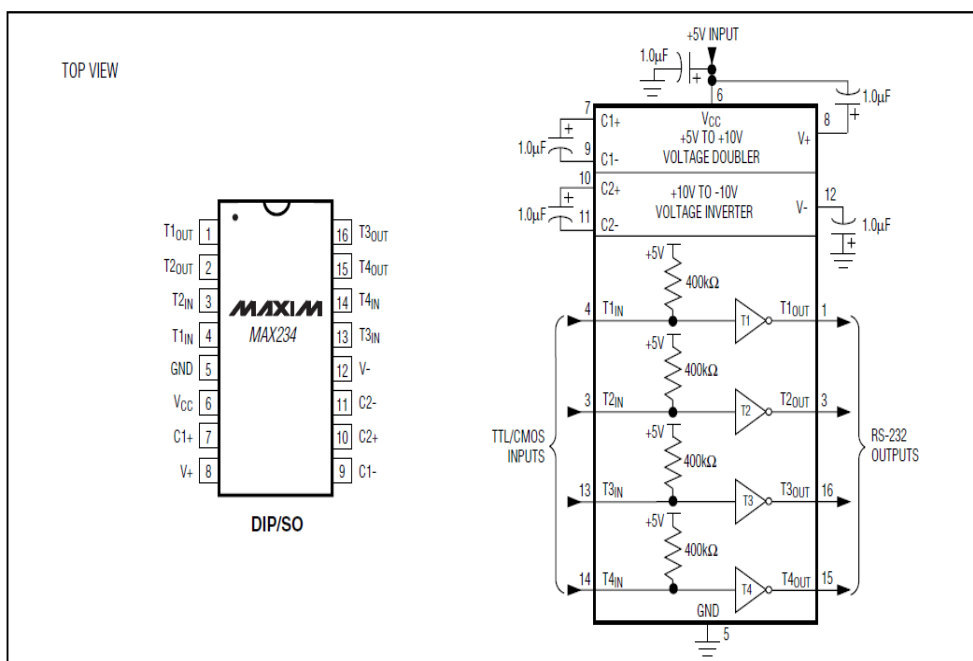
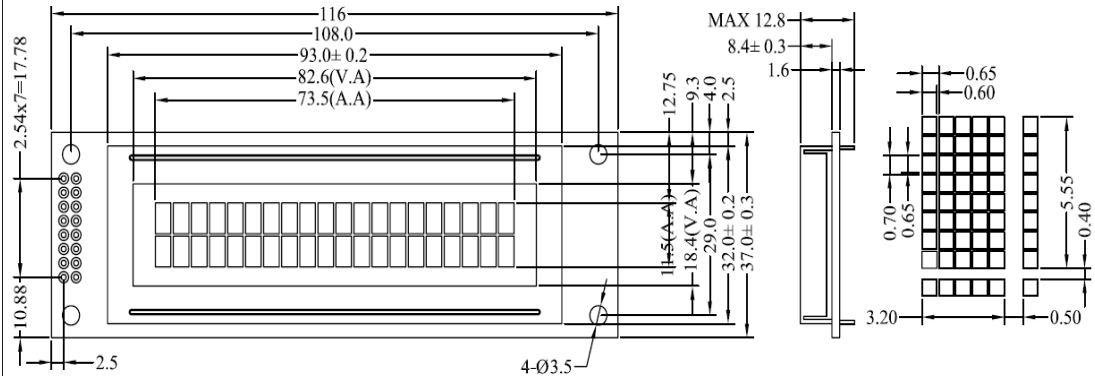


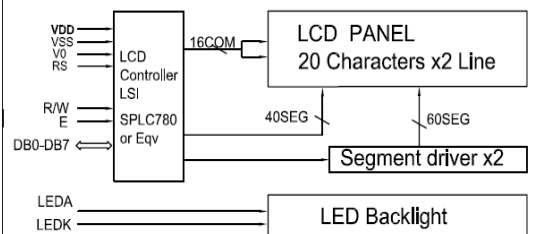
Figure 12. MAX234 Pin Configuration and Typical Operating Circuit

**202-1** (20 chracters x2 Lines)**1/16DUTY, 1/5BIAS****1.DIMENSION OUTLINE****2.MECHANICAL SPECIFICATIONS**

| ITEM                   | SPECIFICATIONS | ITEM | REMARK                        |
|------------------------|----------------|------|-------------------------------|
| Module Size(L×W×H)     | 116×37.0×12.8  | mm   | Reference Dimensional Outline |
| View Area(W×H)         | 82.6×18.4      | mm   |                               |
| Effective V/Area       | 73.5×11.5      | mm   |                               |
| Number of Characters   | 20CH×2Lines    | -    |                               |
| Dot Pitch(W×H)         | 0.65×0.70      | mm   |                               |
| Dot Size(W×H)          | 0.60×0.65      | mm   |                               |
| Weight(Reflective/Led) | -              | g    |                               |

**3.ABSOLUTE MAXIMUM RATINGS**

| ITEM                  | SYMBOL           | CONDITION | STANDARD |                       |
|-----------------------|------------------|-----------|----------|-----------------------|
|                       |                  |           | MIN      | MAX                   |
| Logic Voltage         | V <sub>DD</sub>  | Ta=25℃    | -0.3V    | 7V                    |
| LCD Voltage           | V <sub>LCD</sub> |           | -0.3V    | 13V                   |
| Input Voltage         | V <sub>I</sub>   |           | -0.3V    | V <sub>DD</sub> +0.3V |
| Operation Temperature | T <sub>OP</sub>  | —         | -20℃     | 70℃                   |
| Storage Temperature   | V <sub>OP</sub>  | —         | -30℃     | 80℃                   |

**4.BLOCK DIAGRAMMECHANICAL****5.LED BACKLIGHT SPECIFICATIONS**

| ITEM                 | SYMBOL         | TYPE | MAX | UNIT |
|----------------------|----------------|------|-----|------|
| Ta=25℃               |                |      |     |      |
| Forward Voltage      | V <sub>f</sub> | 4.1  | 4.3 | V    |
| Forward Current      | I <sub>f</sub> | 120  | —   | mA   |
| Emission Vave Length | λ <sub>P</sub> | 568  | —   | nm   |

**6.INTERFACE PIN CONNECTIONS**

| ITEM | SYMBOL  | LEVEL | FUNCTIONS                      |
|------|---------|-------|--------------------------------|
| 1    | VSS     | 0V    | Power Ground                   |
| 2    | VDD     | +5V   | Power Supply For Logic         |
| 3    | V0      | —     | Contrast adjust                |
| 4    | RS      | H/L   | H:data L:command               |
| 5    | R/W     | H/L   | H:read L:write                 |
| 6    | E       | H,H→L | Enable signal                  |
| 7-14 | DB0-DB7 | H/L   | Data Bus                       |
| 15   | LEDA    | +5V   | Power supply For LED Backlight |
| 16   | LEDK    | 0V    |                                |

**7.ELECTRICAL CHARACTERISTICS**

| ITEM                      | SYMBOL             | MIN  | TYPE | MAX             | UNIT |
|---------------------------|--------------------|------|------|-----------------|------|
| Ta=25℃                    |                    |      |      |                 |      |
| Logic Power               | V <sub>DD</sub>    | 4.5  | 5    | 5.5             | V    |
| Input High Voltage        | V <sub>IH</sub>    | 2.2  | —    | V <sub>DD</sub> | V    |
| Input Low Voltage         | V <sub>IL</sub>    | -0.3 | —    | 0.6             | V    |
| Output High Voltage       | V <sub>OH</sub>    | 2.4  | —    | V <sub>DD</sub> | V    |
| Output Low Voltage        | V <sub>OL</sub>    | 0    | —    | 0.4             | V    |
| Logic Current             | I <sub>DD</sub>    | —    | 1.5  | 3.0             | mA   |
| Operation Voltage For LCD | V <sub>DD-V0</sub> | —    | 5    | —               | V    |