



*LICENCE SCIENCES ET TECHNIQUES*  
*Génie Electrique*

**RAPPORT DE FIN D'ETUDES**

*Intitulé :*

*Conception et réalisation du  
thermomètre électronique*

*Réalisé Par :*

*- LEFAF Adil*

*Encadré par :*

*P<sup>r</sup> Hassan EL MOUSSAOUI* (FST FES)

*Soutenu le 17 Juin 2015 devant le jury*

*Pr EL MOUSSAOUI* (FST FES)

*Pr AHAITOUF* (FST FES)

*Pr RAZI* (FST FES)

*Année universitaire 2014/2015*

# *DÉDICACES*

## *Nous tenons à dédier ce travail :*

A notre famille pour leur soutien durant notre formation

A tous nos amis qui avec eux nous avons partagé les moments de souffrance et de joie.

A tous les organismes universitaires qui veillent sur les intérêts des étudiants au corps administratif et éducatif et aussi aux organisations estudiantines.

Nous espérons qu'ils trouveront dans ce travail toute notre reconnaissance et tout notre amour.

# *REMERCIEMENT*

Je tiens à exprimer ma profonde gratitude et mes remerciements les plus sincères à mon professeur encadrant : Pr Hassan EI MOUSSAOUI qui m'a confié ce travail.

J'ai eu le privilège de travailler parmi votre équipe et d'apprécier vos qualités et vos valeurs. Vos sérieux, vos compétences et vos sens du devoir nous ont énormément marqués.

Veillez trouver ici l'expression de mon respectueuse considération et mon profonde admiration pour toutes vos qualités scientifiques et humaines. Ce travail est pour moi l'occasion de vous témoigner mon profonde gratitude.

Je tiens à exprimer ma sincère gratitude aux membres du jury qui ont bien voulu accepter de m'honorer ainsi de leur présence.

# SOMMAIRE

<b>Dédicaces.....</b>	<b>1</b>
<b>Remerciement .....</b>	<b>2</b>
<b>Introduction.....</b>	<b>5</b>
<b>Chapitre I : <i>Les outils du projet</i> .....</b>	<b>6</b>
1- Capteur de température analogique .....	6
2- Capteur de température numérique .....	7
3-les pics.....	9
4-la carte easypic v 7.....	12
5-L'afficheur LCD.....	14
6-L'afficheur 7 segments .....	15
7-programmation avec miKro C .....	15
7.1-Langage C & microcontrôleure .....	15
7.2-Le programmeur Pic Flash .....	17
<b>Chapitre II : <i>Thermomètre à base d'un capteur analogique</i> .....</b>	<b>19</b>
1- Introduction .....	19
2- Processus de mesure de température .....	20
3- Conclusion .....	25
<b>Chapitre III : <i>Thermomètre à base d'un capteur analogique</i> .....</b>	<b>26</b>
1- Introduction .....	26
2- Processus de mesure de température .....	26
3- Conclusion .....	28

<b>Conclusion générale .....</b>	<b>29</b>
<b>Annexe .....</b>	<b>30</b>
<b>Bibliographie.....</b>	<b>37</b>

# **INTRODUCTION**

**Au fil du temps, l'électronique connaît une grande évolution et par conséquent de nouveaux circuits améliorés ont fait leur apparition.**

**L'utilisation des microcontrôleurs ne connaît de limite que l'ingéniosité des concepteurs, on les trouve dans nos cafetières, les magnétoscopes, les radios .....Une étude menée en l'an 2004 montre qu'en moyenne, un foyer américain héberge environ 240 microcontrôleurs.**

**Les microcontrôleurs sont de taille tellement réduite qu'ils peuvent être sans difficulté implantés sur l'application même qu'ils sont censés piloter.**

**Leur prix et leurs performances simplifient énormément la conception de système électronique et informatique.**

**Notre cahier de charge consiste à étudier le principe de fonctionnement et de la conception des thermomètres électroniques à mémoire, afin de les réaliser.**

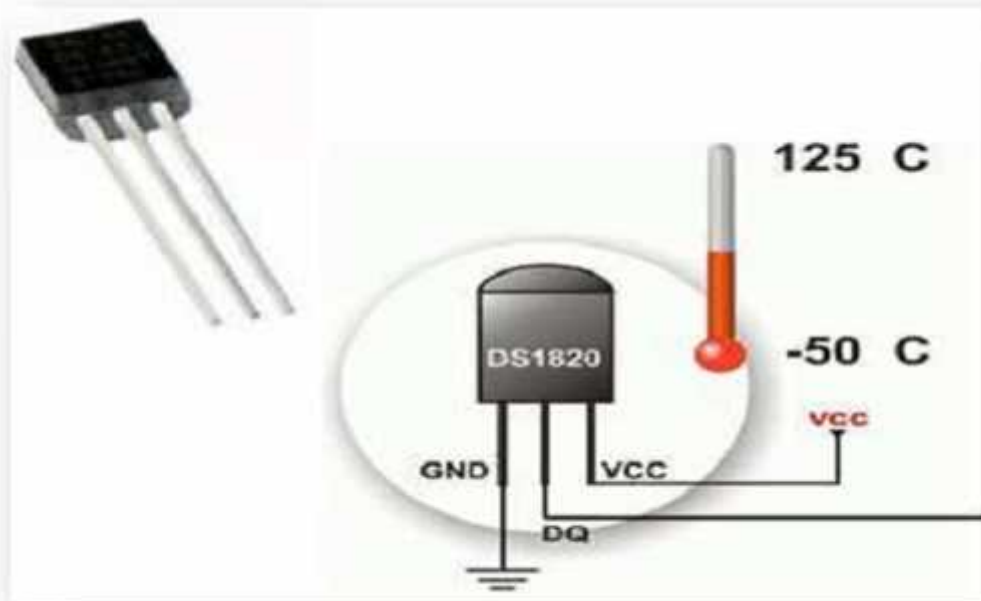
**Nous allons mesurer la température à l'aide d'un capteur et d'afficher le résultat sur l'écran LCD et sur un afficheur 7 segments tous ce processus ce fait grâce à un microcontrôleur que l'on programme.**

**Tout d'abord nous avons choisi un capteur analogique qui nécessite un convertisseur analogique numérique c'est le M35DZ.**



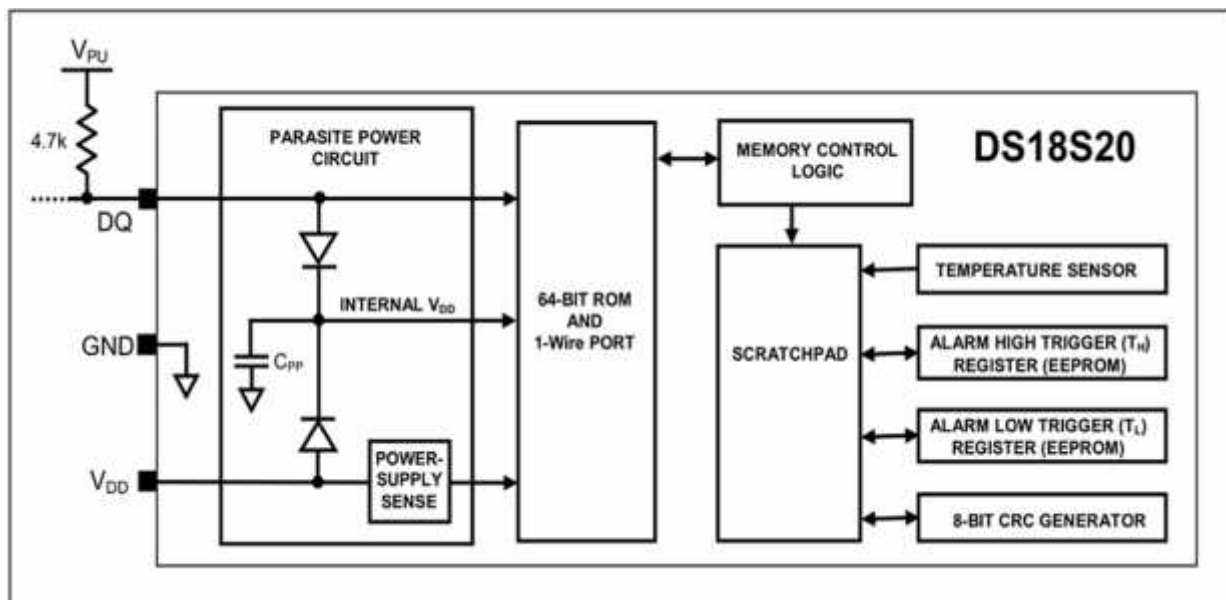






**Figure 3 : Brochage d'un capteur DS1820**

L'architecture interne de ce capteur est la suivante :



**Figure 4 : Architecture interne d'un DS1820**

### 3-les pics :

Le PIC est un élément indispensable dans ce projet, il permet d'afficher la température sur le LCD ou bien sur un afficheur 7 segments, comme il joue le rôle d'un convertisseur analogique-numérique pour le capteur LM35DZ.

Les microcontrôleurs PIC (ou PIC micro dans la terminologie du fabricant) forment une famille de microcontrôleurs de la société Microchip. Ces microcontrôleurs sont dérivés du PIC1650 développé à l'origine par la division microélectronique de Général Instrument. Cependant, à l'époque du développement du PIC1650 par General Instrument, PIC était un acronyme de « Programmable Intelligent Computer » ou « Programmable Integrated Circuit ».

-Famille de PICS : La famille des PICs est subdivisée à l'heure actuelle 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 dont font partie les 16F877A et 18F45K22), et la famille High-End, qui utilise des mots de 16 bits.

- 2 chiffres : famille du PIC (10, 12, 16, 17, 18, 24, 30, 32, 33) —le PIC14 existe, c'est le PIC14000.
- 1 lettre : type de mémoire de programme (C ou F). Le F indique en général qu'il s'agit d'une mémoire flash et donc effaçable électroniquement. La lettre C indique en général que la mémoire ne peut être effacée que par exposition aux ultra-violets (exception pour le PIC16C84 qui utilise une mémoire EEPROM donc effaçable électriquement). Un L peut être ajouté devant pour indiquer qu'il s'agit d'un modèle basse tension (exemple : 2 V à 5,5 V si LF — 4,2 V à 5,5 V si F).
- un nombre de 2 à 4 chiffres : modèle du PIC au sein de la famille. Toutefois il y a maintenant des exceptions : PIC18F45K22 par exemple.
- un groupe de lettres pour indiquer le boîtier et la gamme de température.

-Le PIC utilisé dans le projet PIC 18F45K22:

Le microcontrôleur livré avec la carte est le 18F45K22. Il s'agit d'un microcontrôleur 8 bits dont les principales caractéristiques sont les suivantes :

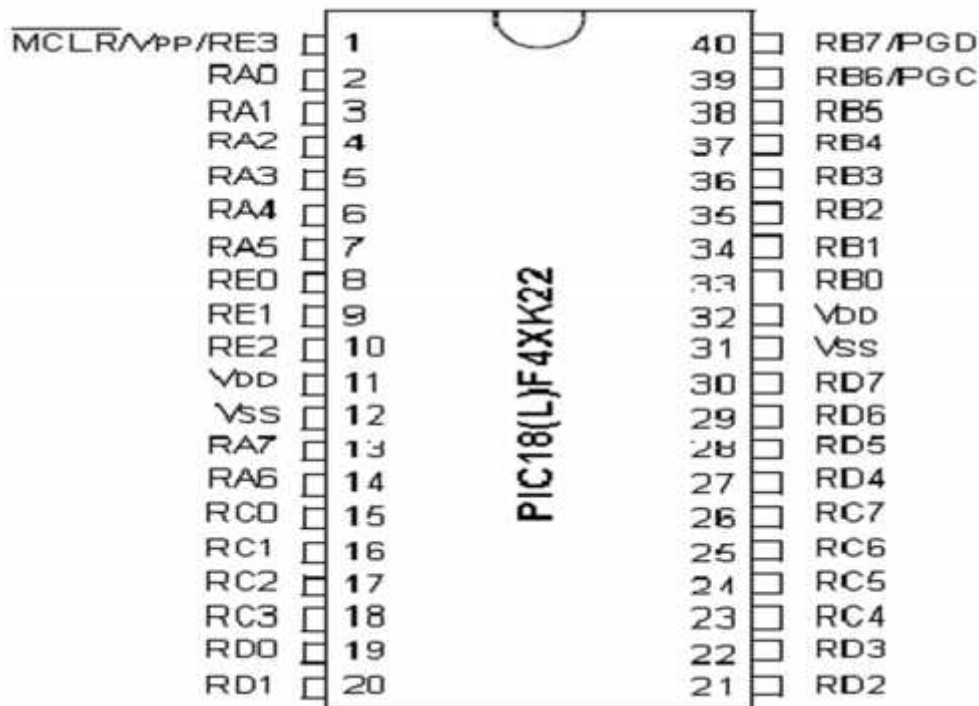
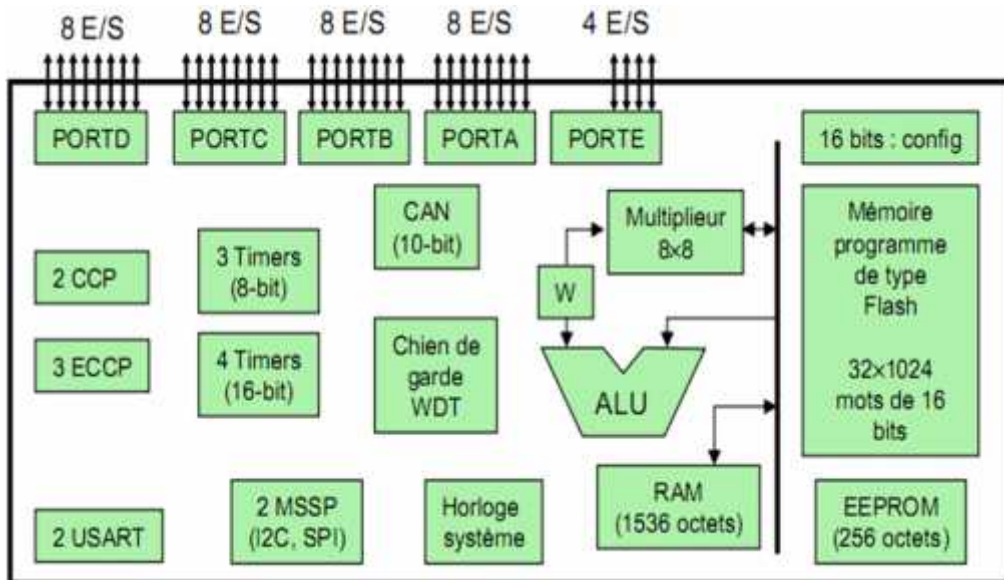


Figure 5 : de pic 18F45K22

- 32Ko de mémoire flash de programme.
- Oscillateur quartz/interne/externe jusqu'à 64MHz.
- 16MIPS.
- Alimentation : 2,3v à 5,5v.
- Debugger ICD.
- Module ADC, comparateur analogique, DAC, support écran et touches tactiles.
- Communication I<sup>2</sup>C, SPI, RS232, RS485, LIN.
- Timers.
- 5 ports I/O parallèles numériques.



[Figure 6 : Bloc diagramme du PIC 18F45K22](#)

- TIMERS : Microchip appelle TIMER des compteurs
- WDT : Watch Dog Timer (Chien de garde)
- (E)CCP : (Enhance)Capture/Compare/PWM
- Capture : permet la mesure de temps
- Compare : permet la production de signaux rectangulaires
- MSSP : Master Synchronous Serial Port (Port Série Synchrone Maître)
- SPI : communications séries synchrones sans protocole logiciel
- I2C : standard Philips, communications séries synchrones avec protocole logiciel
- A/E/USART : (Addressable/Enhanced) Universal Synchronous Asynchronous Receiver Transmitter (RS232 – RS485)
- CAN : convertisseur analogique numérique sur 10 bits à 28 entrées multiplexées

#### 4-la carte easypic v 7 :

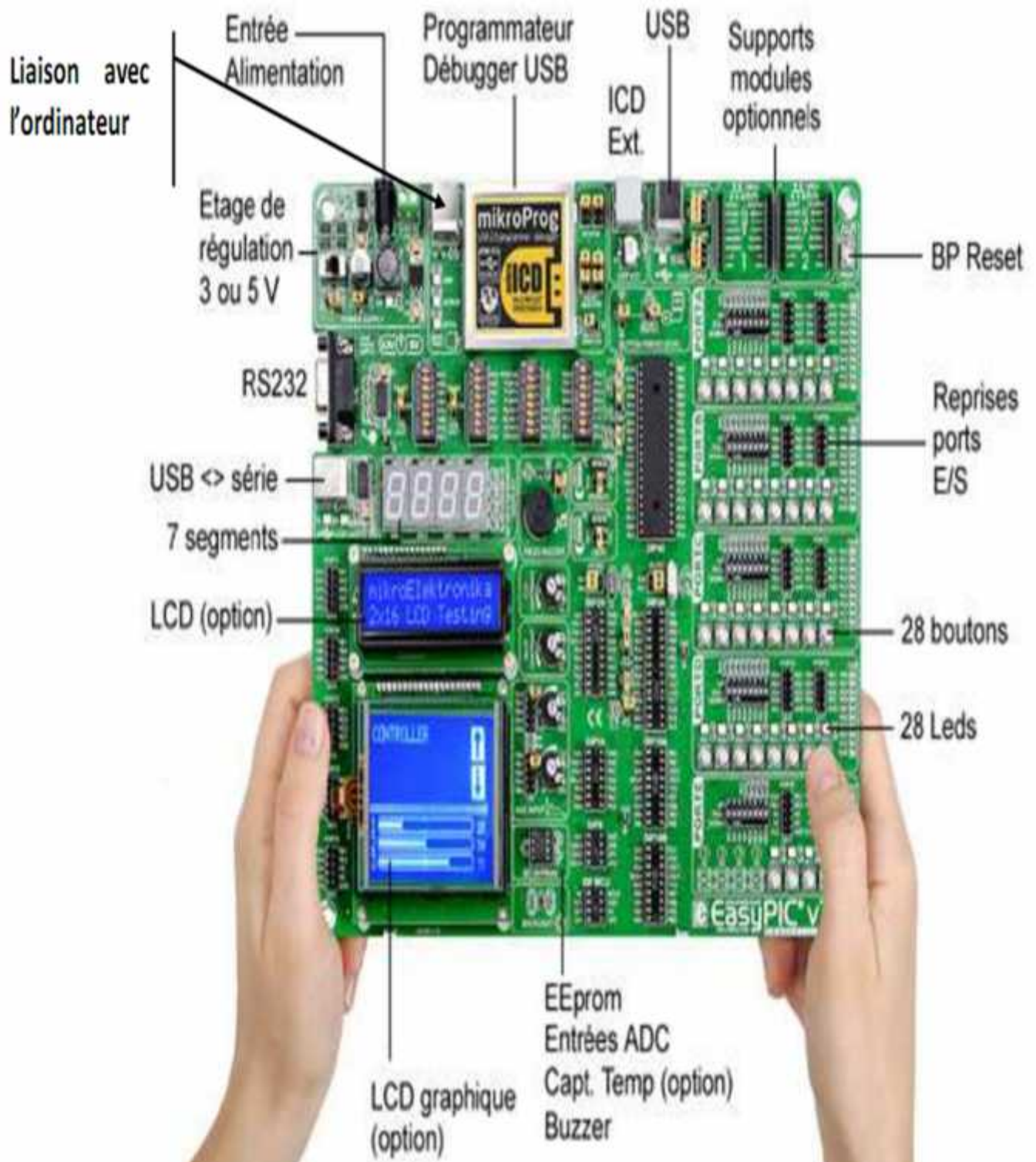
La carte de développement EasyPIC v7 de la société Mikroelektronika permet de développer des projets utilisant les microcontrôleurs Microchip de la famille PIC 12/16/18.

A l'achat elle est équipée du PIC 18F45K22 dont nous verrons les caractéristiques par la suite

#### EasyPIC v7 intègre un grand nombre de fonctionnalités :

- Leds et boutons poussoirs connectables aux différents ports.
- Switches et cavaliers de configurations afin de s'adapter aux microcontrôleurs et à leurs caractéristiques.
- Des connecteurs E/S au format HE10 reliés aux différents ports des microcontrôleurs.
- Programmeur sur la carte permettant la programmation in situ (sans avoir à enlever le microcontrôleur).
- Module ICD permettant le débogage du programme en simulation et en temps réel sur la carte.
- Des connecteurs afin de relier des capteurs (DS18S20 ...) directement sur la carte.
- Un afficheur LCD et un afficheur GLCD tactile.
- 4 afficheurs 7 segments.
- Module de communication RS232, USB.
  
- Connecteur d'extension mikroBUS sockets permettant l'ajout de « click Boards » Bluetooth, MP3, CAN SPI etc...
- Potentiomètres reliés aux entrées des convertisseurs analogique/numérique.
- Une mémoire E<sup>2</sup>PROM I<sup>2</sup>C.

- Un buzzer.



**FIGURE 7 :la carte easy pic v7**

### **5-L'afficheur LCD 2 X 16 :**

L'afficheur LCD est en mode 4 bits : les octets de données et de commandes sont envoyés en 2 temps quartet par quartet sur les bits RB0 à RB3 du port B. Les lignes de contrôle RS, E sont reliés sur les bits RB4 et RB5. Pour utiliser cet afficheur avec la bibliothèque LCD fournie, il faut placer les lignes suivante en début de programme :

#### ***// Attribution de noms aux broches sur lesquelles l'afficheur est connecté :***

```
sbitLCD_RS atRB4_bit;  
sbit LCD_EN atRB5_bit;  
sbitLCD_D7 atRB3_bit;  
sbitLCD_D6 atRB2_bit;  
sbitLCD_D5 atRB1_bit;  
sbitLCD_D4 atRB0_bit;
```

#### ***// configuration des broches d'entrée/sortie :***

```
sbit LCD_RS_Direction at TRISB4_bit;  
sbit LCD_EN_Direction at TRISB5_bit;  
sbit LCD_D7_Direction at TRISB3_bit;  
sbit LCD_D6_Direction at TRISB2_bit;  
sbit LCD_D5_Direction at TRISB1_bit;  
sbit LCD_D4_Direction at TRISB0_bit;
```

## 6-L'afficheur 7 segments – 4 digits :

Chaque afficheur 7-segments est constitué de 7+1 DELs qui permettent d'afficher les chiffres de 0 à 9, quelques lettres de l'alphabet et le séparateur décimal (DP). RA0 à RA3 : sélection de l'afficheur (portA) RD0 à RD6 : activation des 7 segments (portD) RD 7 : activation du point DP (portD).

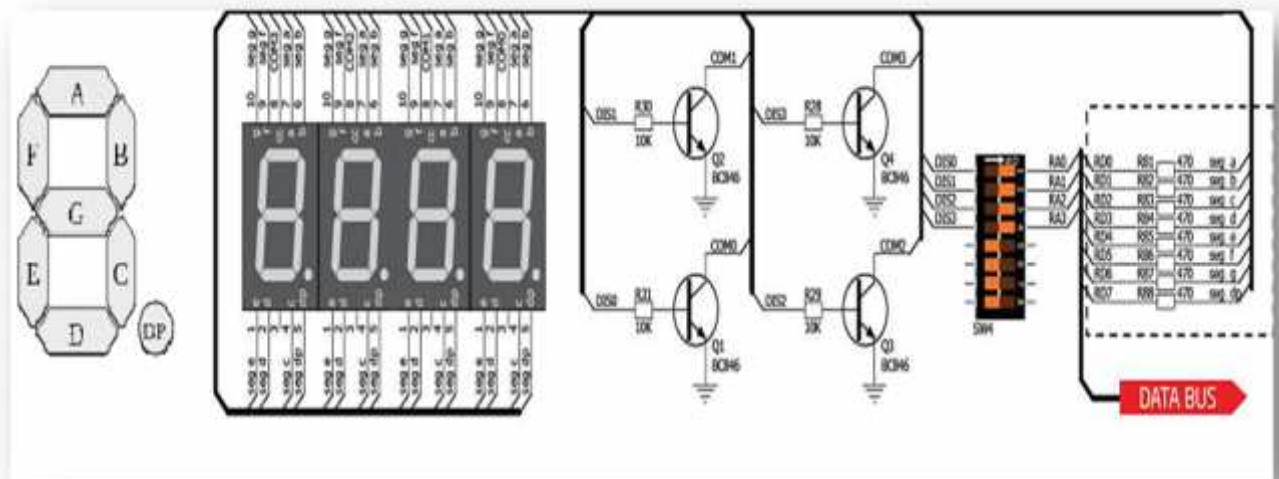


FIGURE 8 :afficheur 7 segments

## 7-programmation avec miKro C :

### 7.1-Langage C & microcontrôleure :

-De plus en plus, les programmes pour micro-contrôleur sont écrits en langage C. Ce langage permet de développer rapidement des programmes, des bibliothèques qui pourront être ensuite utilisées sur différentes machines.

Pourquoi un langage tel que le C ?

- Universel : il n'est pas dédié à une application !
- Moderne : structuré, déclaratif, récursif, avec structures de contrôle et de déclaration.
- Proche de la machine : manipulations de bits, pointeurs, possibilité d'incorporer de l'assembleur, etc.



- Portable : le même code peut être utilisé sur plusieurs machines (il faut toutefois faire attention à ne pas créer des fonctions spécifiques à une machine).

- Extensible : il est possible de créer des bibliothèques ou d'en incorporer

Vous avez accès à la majorité des fonctionnalités du C de la norme ANSI C. Vous pouvez donc écrire quelque chose comme `vitesse = 0x27` pour faire l'affectation d'une variable codée sur 8 bits, ou encore `if(vitesse == limite)` pour effectuer un test

d'égalité entre deux variables entières codées sur un nombre de bits appropriés.

Plus précisément, vous avez accès :

- aux déclaration de variables, constantes, tableaux, pointeurs, structures, etc.
- aux opérateurs arithmétiques et logiques,
- aux opérateurs d'affectation, incrémentation et décrémentation,
- aux opérateurs de contrôle de flux (test et boucles),
- à la conversion de type,
- à l'intégration de routines en assembleur.

Vous avez accès à la majorité des fonctionnalités du C de la norme ANSI C. Vous pouvez donc écrire quelque chose comme `vitesse = 0x27` pour faire l'affectation d'une variable codée sur 8 bits, ou encore `if(vitesse == limite)` pour effectuer un test

d'égalité entre deux variables entières codées sur un nombre de bits appropriés.

Par contre, les instructions gérant les entrées et sorties (écran, clavier, disque, etc.)

n'ont pas toujours de sens dans ce type de contexte.

## 7.2-Leprogrammeur Pic Flash :

-pour comander la maquette easy pic v7 il faut installer d'abord le microC PRO for PIC version 6.4.0 ,et après ouvrir le logiciel microC,après appuyer sur project après new project pour ouvrir une nouvelle fenêtre dans laquelle on va écrire le programme.Après il faut appuyer sur next après on arrive a cette image :

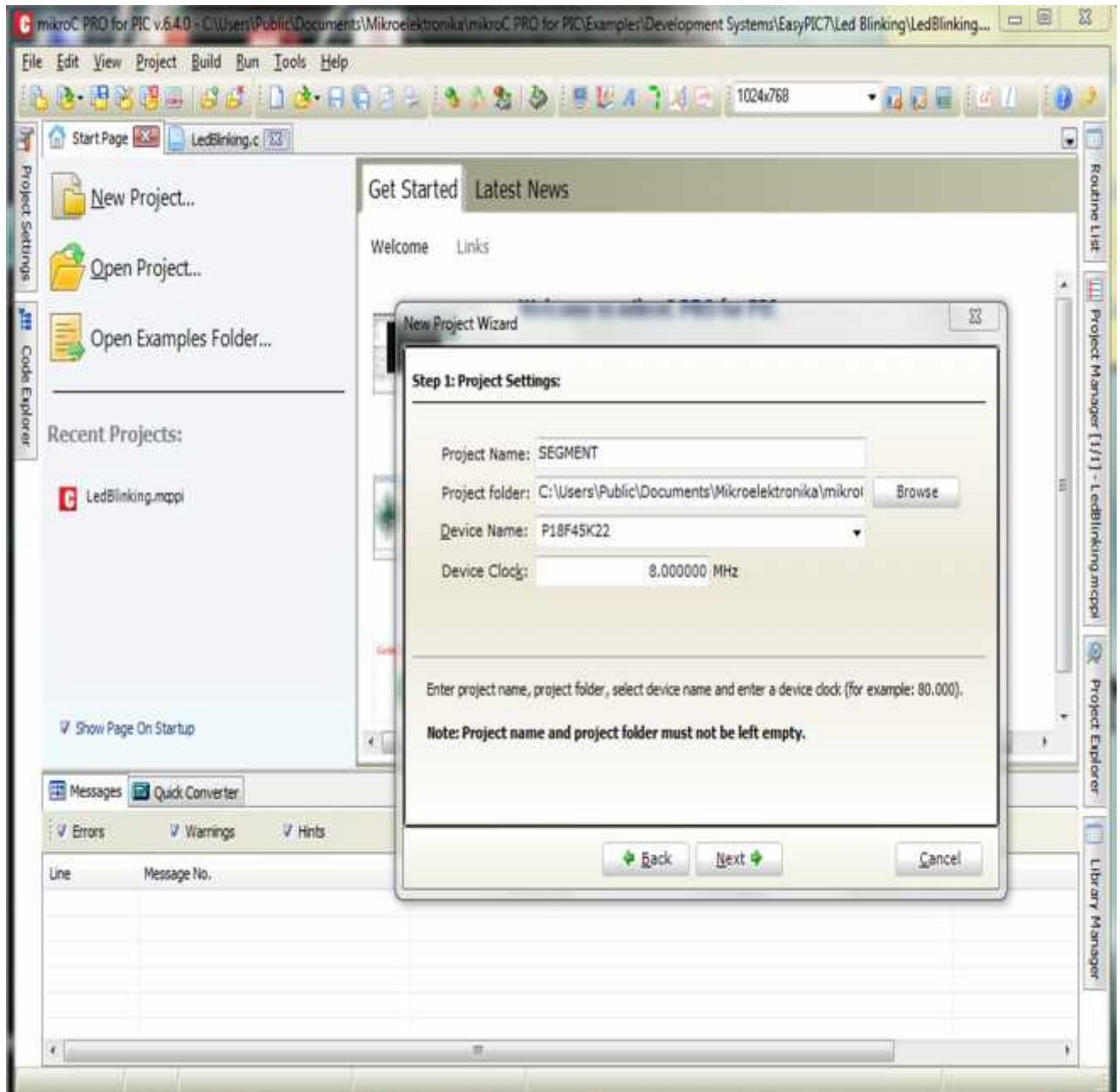
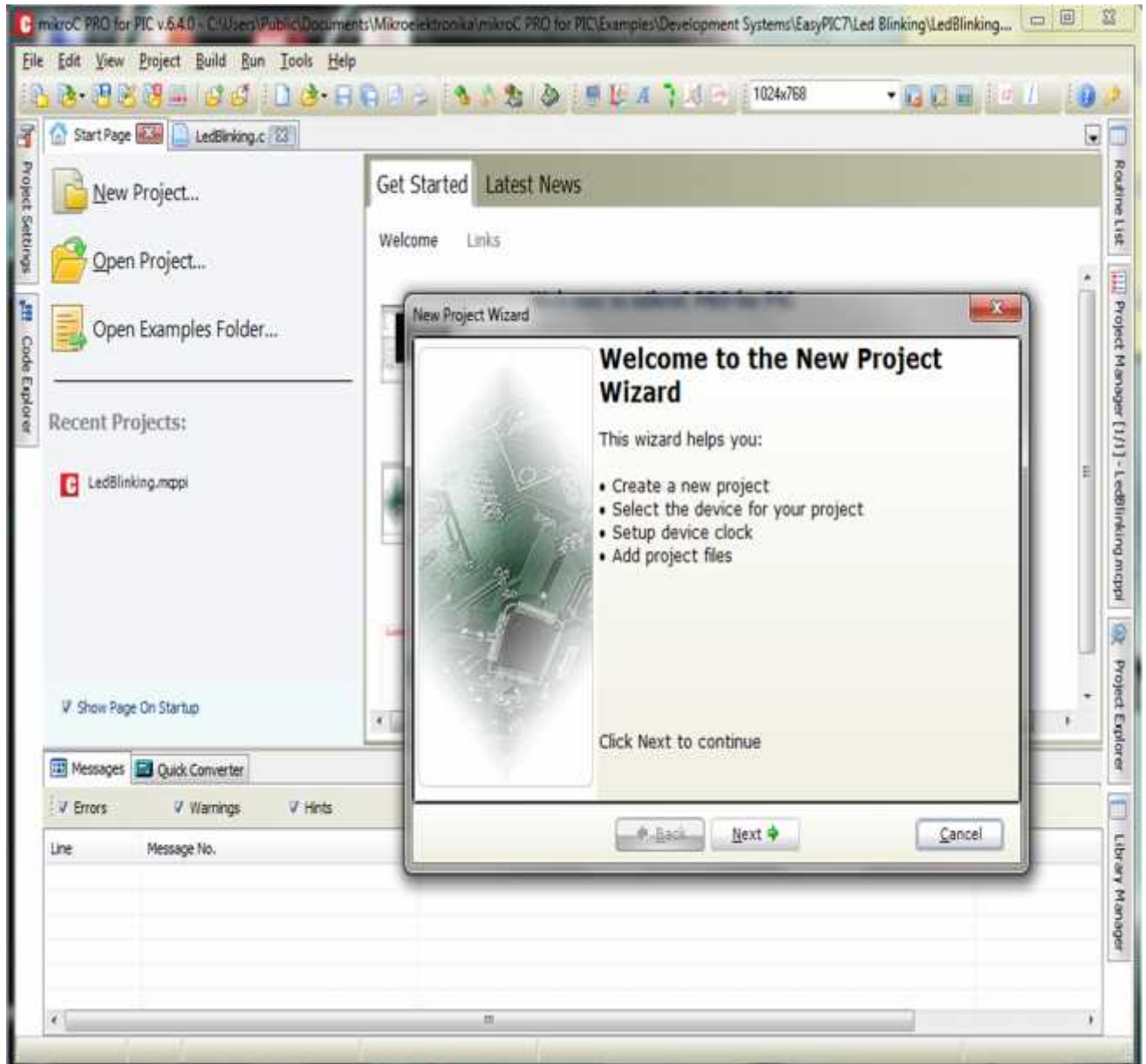


Figure 9 :programme pic flash

- project name :le nom de programme ;
- project folder :la place ou ana enregistrer le fichier de programme ;
- device name :le type de pic qu'on a
- Ensuit il faut appuiyer sure nexte on arrive a la page suivante :



**Figure 10 :programme pic flash**

-Puis appuiyer sur nexte et en fin on arrive a ouvrir la page de commande ou on peut ecrire le programme tout simplement.



Ce thermomètre devra faire la mesure de la température à l'aide de LM35DZ, d'effectuer la conversion analogique numérique de la donnée à l'aide d'un PIC 18F 8520, et finalement d'afficher la température en Celsius sur un écran à cristaux liquides (LCD) ou bien un afficheur 7 segments.

## 2-Processus de mesure de température :

### *a- Acquisition de la température par le capteur :*

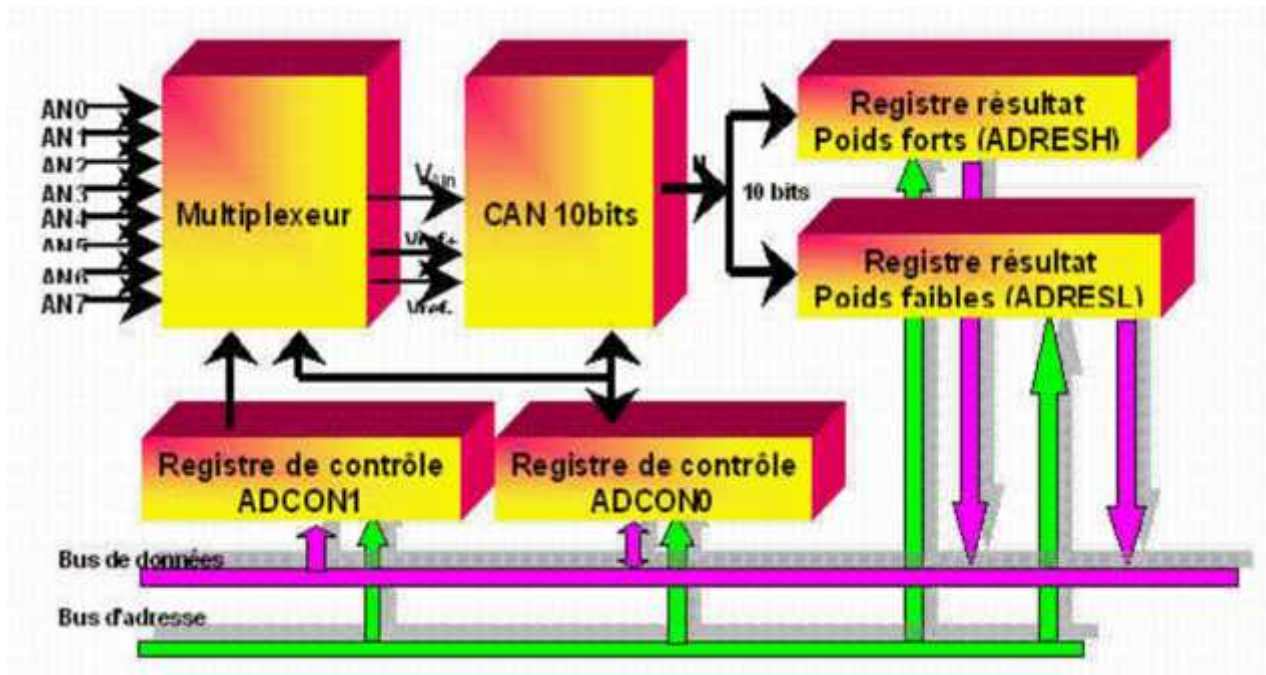
Le LM35 peut être facilement appliqué de la même manière que d'autres capteurs de température à circuit intégré. Ce capteur peut être collé à une surface et sa température sera dans environ  $0,01^{\circ}\text{C}$  de la température de surface. Cela suppose que la température de l'air ambiant est presque même que la température de surface, si la température de l'air est beaucoup plus élevée ou plus basse que la température de surface, la température actuelle de LM35 serait à un intermédiaire de température entre la température de surface et la température de l'air.

À la fin de la mesure, on obtient une tension de sortie en relation linéaire avec la température exprimée en degrés Celsius. L'étape suivante est de convertir cette grandeur analogique en une grandeur numérique. Pour cette raison, on utilise le PIC qui intègre un convertisseur analogique numérique.

### *b- La conversion de la valeur analogique :*

#### *La fonction CAN du PIC 18F :*

La fonction CAN du PIC 18F se décompose de la façon suivante :



[Figure 13: Schéma fonctionnel de la fonction conversion analogique numérique du PIC 18F.](#)

- ❖ Une fonction multiplexage qui permet de diriger les huit lignes analogiques vers une sortie.
- ❖ La fonction CAN qui est un convertisseur 10 bits à approximation successive. L'entrée de cette fonction s'effectue sur un échantillonneur bloqueur qui permet de maintenir constante, la tension à convertir durant le temps de conversion.
- ❖ Une fonction mémorisation qui permet de mémoriser le résultat de la conversion dans deux registres de 8bits chacun (ADRESH : résultat poids forts ; ADRESL : résultat poids faibles).
- ❖ Une fonction de contrôle qui permet de sélectionner les entrées analogiques et de contrôler la conversion (le lancement et la fin de la conversion) qui est réalisée par deux registres ADCON0 et ADCON1.
- ❖ Les bus d'adresses et de données permettent la communication avec l'unité arithmétique logique et les autres registres mémoires du microcontrôleur.

### a-Rôle des registres résultats ADRESH et ADRESL :

A la fin de la conversion le résultat sur 10 bits est stocké dans deux registres de 8 bits ADRESH et ADRESL. Selon la valeur du bit ADFM (bit 5 du registre ADCON1) la justification du résultat se fait à droite ou à gauche. Les bits n'appartenant pas au résultat de la conversion sont positionnés à 0 (voir figure ci-dessous).

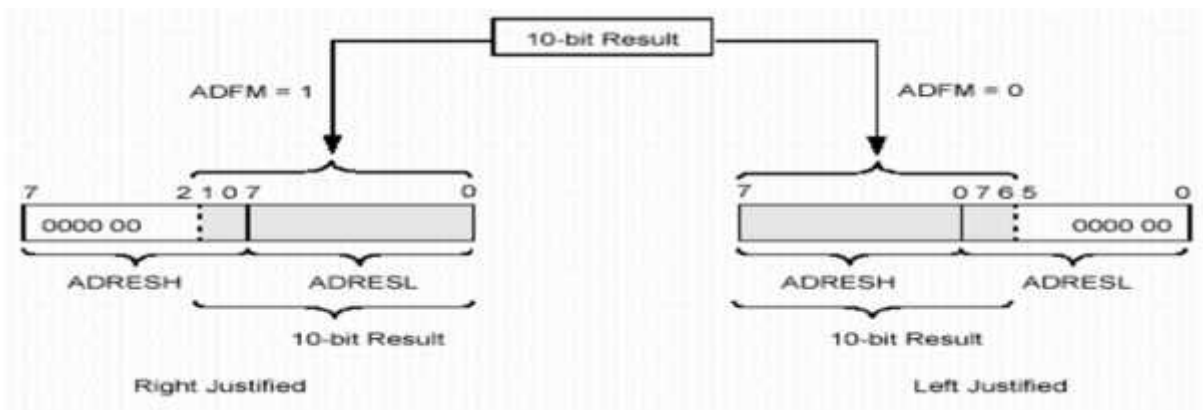


Figure 14: Schéma fonctionnel de ADRESH et ADRESL

### b- Rôle du registre ADCON0 :

Le registre ADCON0 contrôle les opérations de la fonction conversion analogique numérique. Il permet de :

- sélectionner la fréquence d'horloge du convertisseur ;
- sélectionner le canal ou l'entrée analogique sur laquelle doit être effectuée la conversion ;
- lancer la conversion ;

### c- Rôle du registre ADCON1 :

Le registre de contrôle ADCON1 permet de :

- choisir la justification droite ou gauche du résultat ;
- sélectionner les entrées analogiques utilisées ;
- sélectionner les entrées analogiques AN2 et AN3 pour régler l'échelle de conversion par des tensions extérieures (tensions de références).

## d-Description des étapes pour réaliser une conversion analogique numérique :

### Etape 1 : Configuration de la fonction CAN :

- ✚ choisir les broches analogiques, les tensions de références, les entrées numériques (registre ADCON1) ;
- ✚ sélectionner l'entrée analogique sur laquelle la conversion doit être effectuée (registre ADCON0) ;
- ✚ sélectionner la fréquence de l'horloge de conversion (registre ADCON0) ;
- ✚ activer la fonction de conversion (registre ADCON0).

### Etape 2 : Configuration des interruptions (si nécessaire) :

- ✚ mettre à 0 le bit ADIF (A/D converter interrupt flag, bit 6 du registre PIR1) ;
- ✚ mettre à 1 le bit ADIE (A/D converter interrupt enable, bit 6 du registre PIE1) ;
- ✚ mettre à 1 le bit GIE (global interrupt enable, bit 7 du registre INTCON).

### Etape 3 : Attendre le temps de préparation de la conversion (Acquisition time).

Ce temps correspond au temps de charge du condensateur de l'échantillonneur Bloqueur (CHOLD). Il est de l'ordre de 20 $\mu$ s.

### Etape 4 : Lancer la conversion :

- ✚ mettre à 1 le bit GO/DONE du registre ADCON0.

### Etape 5 : Attendre l'achèvement de la conversion analogique numérique :

- ✚ soit en scrutant le passage à 0 du bit GO/DONE ou le passage à 1 du bit ADIF ;
- ✚ soit en attendant une interruption de la fonction CAN.

### Etape 6 : Mémorisation du résultat :

- ✚ lire le résultat dans les registre ADRESH et ADRESL ;
- ✚ mettre à 0 le bit ADIF si cela est nécessaire.



### **Etape 7 : Lancement d'une nouvelle conversion :**

✚ retourner à l'étape 1 ou 2 si cela est nécessaire. Un temps d'attente de 2 TAD est nécessaire entre deux conversions. TAD temps de conversion analogique numérique pour un bit. Une conversion analogique numérique complète nécessite 12 TAD pour les 10 bits de conversion.

Pour créer le programme de la conversion de tension en une grandeur numérique, on utilise la fonction « ADC\_Read ».

Le code développé pour la gestion du capteur de température est donné en ANNEXE.

Pour bénéficier pleinement de la plage 0 °C à +100 °C avec le CAN inclus dans le PIC, ou plutôt pour conserver une assez bonne précision, il convient de multiplier la tension mesurée (fournie par le LM35) dans un rapport de 4,88. L'idée de départ est qu'une tension de 1,0 V corresponde à +20 °C, et qu'une tension de +5 V corresponde à une température de +100 °C.

Comme la résolution du CAN (convertisseur analogique / numérique) du PIC est de 10 bits, on dispose de 1024 pas de quantification. Si on fait l'économie de 23 pas sur les 1024, on dispose d'une correspondance directe entre valeur du CAN et valeur de la température mesurée (rapport de 10 entre les deux).

$$**V_{max \text{ pour } 100 \text{ } ^\circ\text{C}} = 5 \text{ V} / 1024 * 1000 = 4,88 \text{ V}**$$

Après création d'un programme en Mikro C, on a réalisé le schéma du thermomètre sous ISIS. La première étape est de connecter la sortie du capteur à la broche du PIC à la quelle il va lire la valeur à fin de la convertir. La deuxième étape est de charger le PIC par le programme.



## Chapitre III : Thermomètre à base d'un capteur analogique :

### 1- Introduction :

Dans cette partie, on a utilisé le capteur numérique DS1820 pour réaliser le thermomètre. Les étapes séquentielles de la mesure de la température sont les suivantes :



Figure 16 : processus de mesure de température

### 2- Processus de mesure de température :

#### Acquisition de la température par le capteur :

Le DS 1820 délivre directement un mot binaire, image de la température, en sortie. Pour initialiser la température à mesurer et le convertisseur analogique numérique, le maître(PIC) doit émettre une commande de conversion. Suite à la conversion, les données thermique résultantes sont stockés dans les 2 bites du registre de température en mémoire .

« Scratchpad ». Le «Scratchpad » est une zone mémoire contenant la valeur de la température après conversion. À la fin du processus, le DS1820 revient à son état initial.

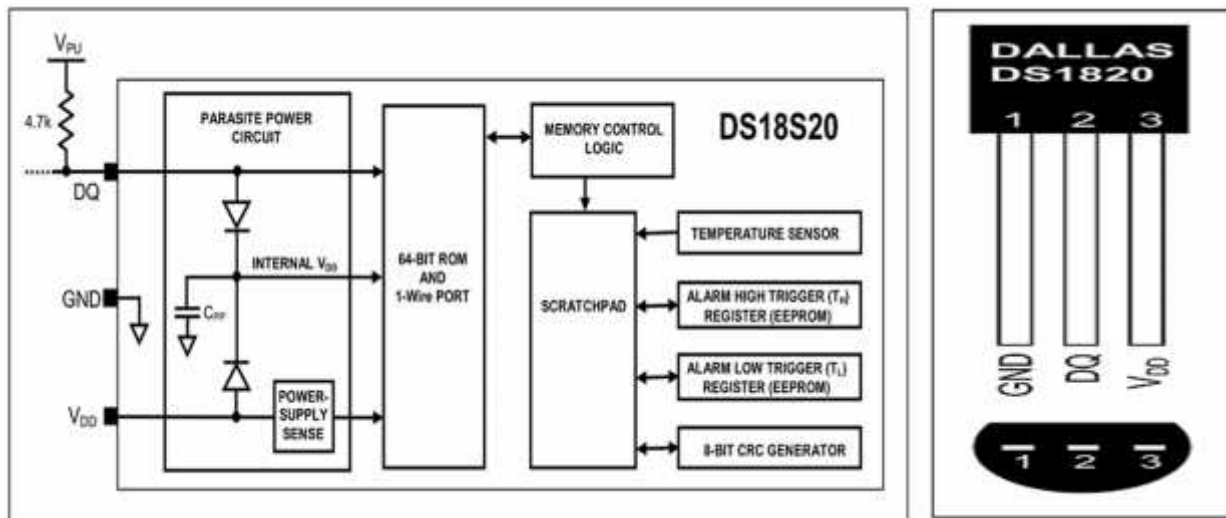


Figure 17 : Architecture interne d'un DS1820 :

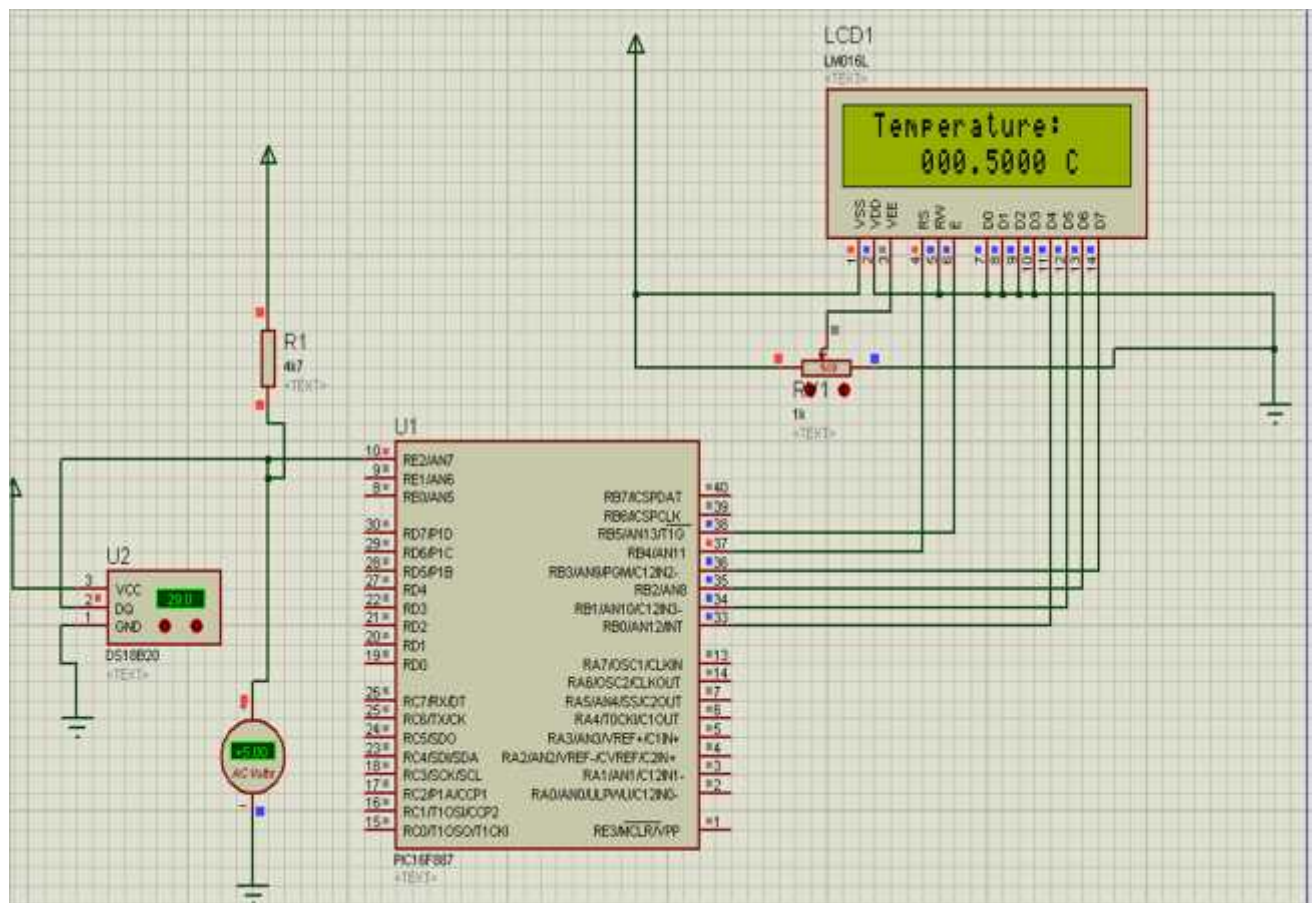


Figure 18 : Simulation sous ISIS d'un capteur DS1820

### 3-conclusion :

Le thermomètre de DS1820 Digital fournit 9 à 12 lectures (configurables) de la température de bit qui indiquent la température du dispositif. Le premier avantage de ce type du capteur c'est que l'information est envoyée de DS1820 au PIC 18F 8520 au dessus d'une interface One -Wire, de sorte que seulement un fil (et terre) doivent être relié d'un microcontrôleur à un DS1820. La puissance pour la lecture, l'écriture, et exécuter des conversions de la température peut être dérivée de la ligne de données elle-même sans des besoins de source d'énergie externe. Un autre avantage pour ce type du capteur, c'est qu'il délivre directement un mot binaire, image de la température, en son sortie. Ils intègrent le capteur, le circuit de mise en forme et le convertisseur analogique numérique.

## **CONCLUSION GÉNÉRALE**

La réalisation de ce projet simple nous a énormément appris, soit au niveau de l'électronique, ou bien au niveau de la programmation des microcontrôleurs (programmation embarquée). Nous avons aussi acquis de nouvelles connaissances au niveau de la gestion du temps et des équipes.

Nous étions fascinés par le domaine d'électronique et d'informatique, et nous trouvons la combinaison de ces deux domaines vivants dans les microcontrôleurs qui sont utilisés dans tous les systèmes embarqués, ils sont très répandus en robotique et en automatisme.

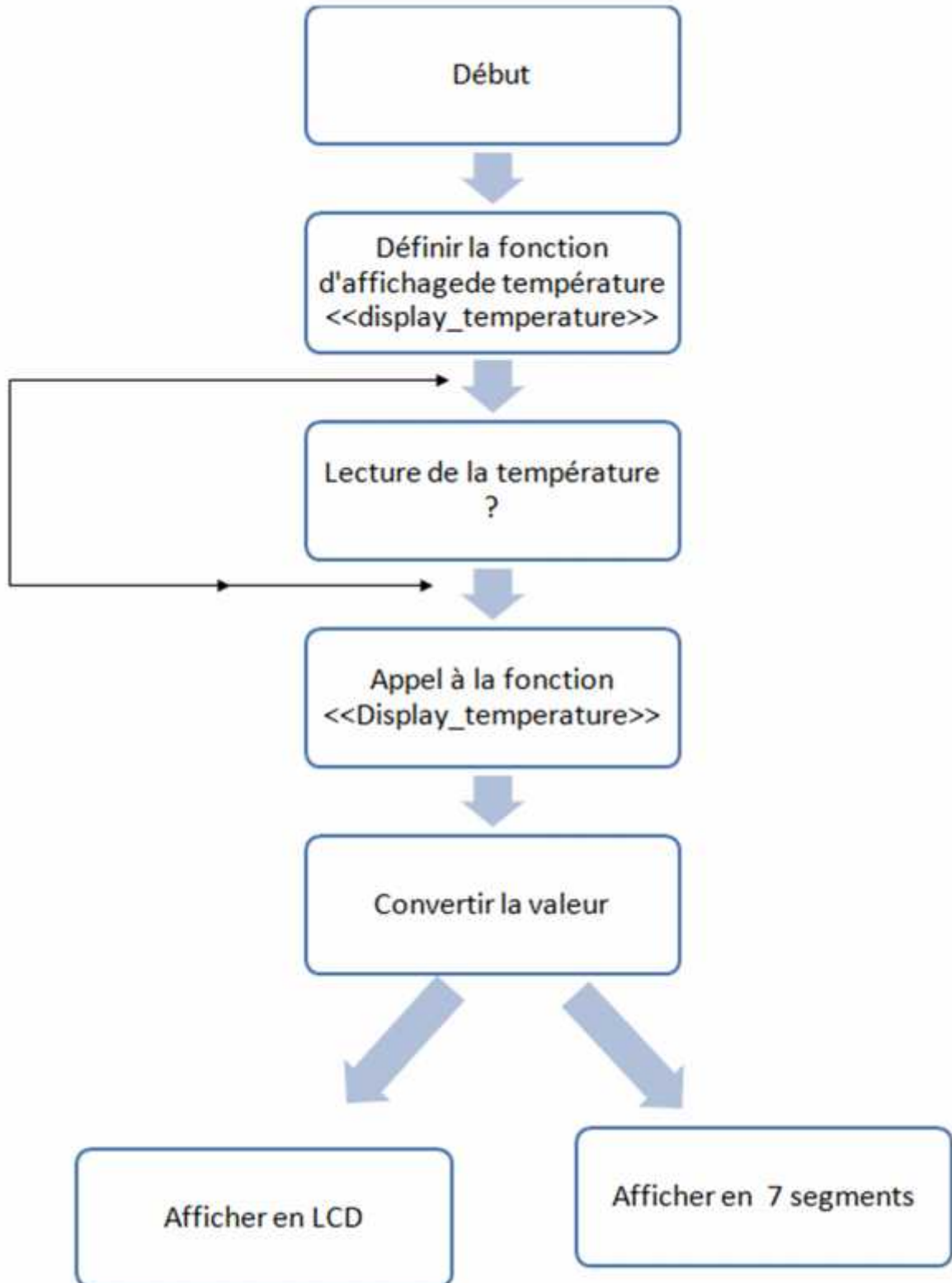
Ce projet nous a permis d'améliorer nos connaissances, ainsi d'acquérir les comportements répondant dans le secteur d'emploi. La partie que nous avons développée correspond à nos objectifs.

Ce travail reste, comme toute œuvre humaine, incomplet et perfectible, nous espérons qu'il sera une initiative visant à développer notre connaissance dans les prochains jours.

Nous souhaitons que notre travail vous ait intéressé.

# ANNEXE

## 1- Organigramme :



## 2- Programme de capteur LM35DZ :

### Affichage sur LCD :

**// Module de connexion du LCD**

**sbit LCD\_RS at Rb0\_bit;**

**sbit LCD\_EN at Rb1\_bit;**

**sbit LCD\_D4 at Rb2\_bit;**

**sbit LCD\_D5 at Rb3\_bit;**

**sbit LCD\_D6 at Rb4\_bit;**

**sbit LCD\_D7 at Rb5\_bit;**

**sbit LCD\_RS\_Direction at TRISb0\_bit;**

**sbit LCD\_EN\_Direction at TRISb1\_bit;**

**sbit LCD\_D4\_Direction at TRISb2\_bit;**

**sbit LCD\_D5\_Direction at TRISb3\_bit;**

**sbit LCD\_D6\_Direction at TRISb4\_bit;**

**sbit LCD\_D7\_Direction at TRISb5\_bit;**

**char bb1,bb2;**

**unsigned int a,aa,bb;**

**void main() {**

**trisc=0x00; portc=0x00;**

**trisd=0x00; portd=0x00;**

**ADC\_init(); //activer le convertisseur analogique**

**Lcd\_Init(); // Initialize LCD**

**Lcd\_Cmd(\_LCD\_CLEAR);**

**Lcd\_Cmd(\_LCD\_CURSOR\_OFF);**

**lcd\_out(1,1,"tempe:");**

**lcd\_out(1,11,"C");**

**for(;;)**

**{**



```

a=ADC_read(a); //afficher la valeur numérique sur a
portc=a; //8bits le plus faible
portd=a>>8; //afficher les 2bits le plus fort
  if(a>=559){

      lcd_out(1,6,"+");
      aa=(a-559)/2;
      if(aa>99)
          aa=99;
          bb1=aa/10;
          lcd_chr (1,7,bb1+48);
          bb2=aa-(bb1*10);
          lcd_chr (1,8,bb2+48);
      }

      else{
          lcd_out(1,6,"-");
          aa=(a-457)/2;
          if(aa>99)
              aa=99;
              bb1=aa/10;
              lcd_chr (1,7,bb1+48);
              bb2=aa-(bb1*10);
              lcd_chr (1,8,bb2+48);
          }
      }
  }
}

```

## 2- Programme de capteur DS1820 :

**// LCD module connections**

**sbit LCD\_RS at RB4\_bit;**

**sbit LCD\_EN at RB5\_bit;**

**sbit LCD\_D4 at RB0\_bit;**

**sbit LCD\_D5 at RB1\_bit;**

**sbit LCD\_D6 at RB2\_bit;**

**sbit LCD\_D7 at RB3\_bit;**

**sbit LCD\_RS\_Direction at TRISB4\_bit;**

**sbit LCD\_EN\_Direction at TRISB5\_bit;**

**sbit LCD\_D4\_Direction at TRISB0\_bit;**

**sbit LCD\_D5\_Direction at TRISB1\_bit;**

**sbit LCD\_D6\_Direction at TRISB2\_bit;**

**sbit LCD\_D7\_Direction at TRISB3\_bit;**

**// End LCD module connections**

**// Set TEMP\_RESOLUTION to the corresponding resolution of used**

**DS18x20 sensor:**

**// 18S20: 9 (default setting; can be 9,10,11,or 12)**

**// 18B20: 12**

**const unsigned short TEMP\_RESOLUTION = 9;**

**char \*text = "000.0000";**

**unsigned temp;**

**void Display\_Temperature(unsigned int temp2write) {**

```

const unsigned short RES_SHIFT = TEMP_RESOLUTION - 8;
char temp_whole;
unsigned int temp_fraction;

// Check if temperature is negative
if (temp2write & 0x8000) {
    text[0] = '-';
    temp2write = ~temp2write + 1;
}

// Extract temp_whole
temp_whole = temp2write >> RES_SHIFT ;

// Convert temp_whole to characters
if (temp_whole/100)
    text[0] = temp_whole/100 + 48;
else
    text[0] = '0';

text[1] = (temp_whole/10)%10 + 48;        // Extract tens digit
text[2] = temp_whole%10 + 48;           // Extract ones digit

// Extract temp_fraction and convert it to unsigned int
temp_fraction = temp2write << (4-RES_SHIFT);
temp_fraction &= 0x000F;
temp_fraction *= 625;

```

```

// Convert temp_fraction to characters
text[4] = temp_fraction/1000 + 48;    // Extract thousands digit
text[5] = (temp_fraction/100)%10 + 48; // Extract hundreds digit
text[6] = (temp_fraction/10)%10 + 48; // Extract tens digit
text[7] = temp_fraction%10 + 48;     // Extract ones digit

// Print temperature on LCD
Lcd_Out(2, 5, text);
}

void main() {
    ANSEL = 0;                        // Configure AN pins as digital I/O
    ANSELH = 0;
    C1ON_bit = 0;                     // Disable comparators
    C2ON_bit = 0;

    Lcd_Init();                       // Initialize LCD
    Lcd_Cmd(_LCD_CLEAR);              // Clear LCD
    Lcd_Cmd(_LCD_CURSOR_OFF);        // Turn cursor off
    Lcd_Out(1, 1, " Temperature: ");
    // Print degree character, 'C' for Centigrades
    Lcd_Chr(2,13,223);                // Different LCD displays have
different char code for degree
// If you see greek alpha letter try typing 178
instead of 223

    Lcd_Chr(2,14,'C');

```

```

//--- Main loop
do {
    //--- Perform temperature reading
    Ow_Reset(&PORTE, 2);           // Onewire reset signal
    Ow_Write(&PORTE, 2, 0xCC);    // Issue command SKIP_ROM
    Ow_Write(&PORTE, 2, 0x44);    // Issue command CONVERT_T
    Delay_us(120);

    Ow_Reset(&PORTE, 2);
    Ow_Write(&PORTE, 2, 0xCC);    // Issue command SKIP_ROM
    Ow_Write(&PORTE, 2, 0xBE);    // Issue command
    READ_SCRATCHPAD

    temp = Ow_Read(&PORTE, 2);
    temp = (Ow_Read(&PORTE, 2) << 8) + temp;

    //--- Format and display result on Lcd
    Display_Temperature(temp);

    Delay_ms(500);
} while (1);
}

```

# **BIBLIOGRAPHIE**

## **Ouvrages :**

**Datasheet PIC18F45k22, Microchip**

**Datasheet LM35DZ, Microchip**

**Datasheet EasyPic V7**

## **Sites internet :**

**"<http://fr.wikipedia.org/wiki/Microcontr%C3%B4leur>"**

**"[www.mikroe.com/eng/products/view/757/easypic-v7-development-system/](http://www.mikroe.com/eng/products/view/757/easypic-v7-development-system/)"**

**"[www.elektor.fr/120096](http://www.elektor.fr/120096)"**

**"[www.mikroelektronica.com](http://www.mikroelektronica.com)"**





















































