



Diplôme de Licence

*Electronique Télécommunication et Informatique
(ETI)*

Rapport de PROJET de FIN D'ETUDES

Intitulé :

*Conception et réalisation du thermomètre
électronique à mémoire*



Réalisé Par :

HASSANI ALAOUI Fatima Zahra

Encadré par :

P^r Hassan EL MOUSSAOUI

P^r Tijani LAMHAMDI

Soutenu le mardi 21 Juin 2011 devant le jury

Pr A. MECHAQRANE

Pr RAZI

Pr H. EL MOUSSAOUI

Pr T. LAMHAMDI



Dédicace

A mes chers parents

Vous représentez pour moi le symbole de bonté par excellence, la source de tendresse et l'exemple de dévouement qui n'a pas cessé de m'encourager. Rien en monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être.

Aucune dédicace ne saurait être assez éloquente pour exprimer ce que vous méritez pour tous les sacrifices que vous m'avez cessé de me donner depuis ma naissance, durant mon enfance et même à l'âge adulte.

A mes chers frères : Aziz, Hicham et Mohammed

Les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour vous. Vous êtes toujours présent dans tous les moments difficiles par votre soutien moral.

A mes professeurs

Un profond respect et un remerciement particulier pour vos efforts fournis.

A mes chères collègues du laboratoire de recherche

En témoignage de l'amitié qui nous unit et des souvenirs de tous les moments que nous avons passé ensemble, je vous dédie ce travail et je vous souhaite une vie pleine de santé et de bonheur.

A mes chers collègues et amis

Je vous dédis ce travail avec tous mes vœux de bonheur, de santé et de réussite.



Remerciement

Je tiens à exprimer ma profonde gratitude et mes remerciements les plus sincères à mes professeurs encadrant : Pr Hassan El MOUSSAOUI et Pr Tijani LAMHAMDI qui m'ont 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.

Mes remerciements vont droit également au Mr ABDI pour son dévouement et à tous ceux qui de près ou de loin ont contribué à ma formation, qu'ils trouvent en ces mots l'expression de ma profonde reconnaissance.

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

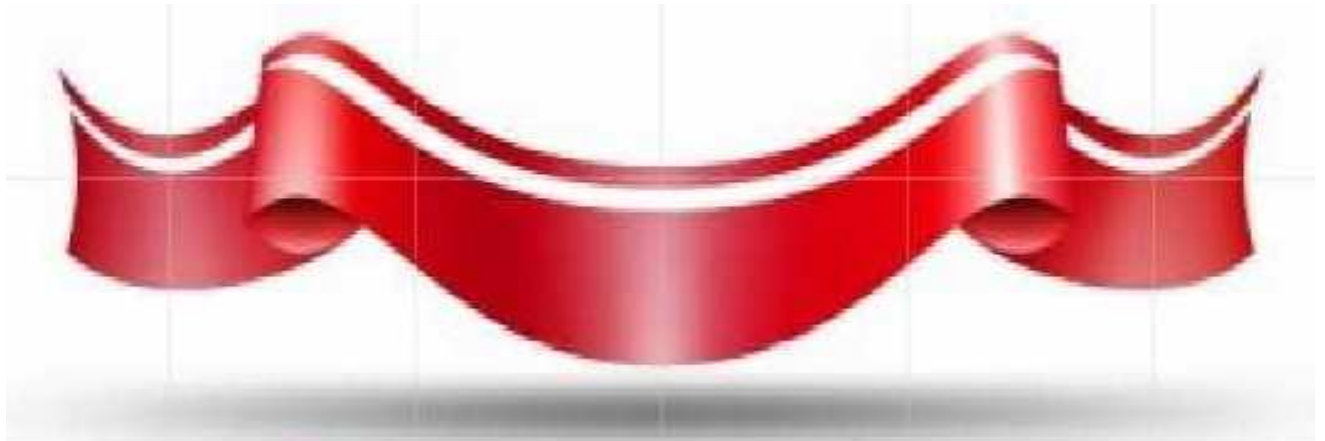
Sommaire



Sommaire

Introduction	5
Présentation du lieu du stage.....	7
Chapitre 1 : Les outils du projet.....	11

Introduction.....	12
Les capteurs.....	12
Les PIC.....	17
La carte BIGPIC6.....	20
L'afficheur LCD.....	21
Le logiciel ISIS.....	23
Chapitre 2 : Thermomètre à base d'un capteur analogique.....	27
Introduction.....	28
Processus de mesure de température.....	28
Conception et réalisation du thermomètre.....	33
Conclusion.....	36
Chapitre 3 : Thermomètre à base d'un capteur numérique.....	37
Introduction.....	38
Processus de mesure de température.....	38
Conception et réalisation du thermomètre.....	43
Conclusion.....	45
Conclusion.....	46
Annexe.....	48
Programme du thermomètre à base d'un capteur numérique.....	49
Programme du thermomètre à base d'un capteur analogique.....	51
Prise en main de la carte BIGPIC6.....	53
Bibliographie.....	60



Introduction

Dans la vie quotidienne, la notion de température est l'un des thèmes les plus abordés. La température est là, elle est partout ! On n'y prête plus attention, tellement c'est devenu un phénomène courant et « transparent ».

Toutefois, avec l'évolution de la science et des technologies, les hommes ont été à plusieurs reprises confrontés à des problèmes liés à la température : dilatation des métaux, variation des résistances, problème des semi-conducteurs etc.... Dans ce sens est né l'idée de mesurer la température.

Le premier thermomètre a été inventé en 1654 par le duc de Toscane à Florence. Il portait 50 graduations. En hiver, il descendait jusqu'à 7 degrés et montait, en été, à 40 degrés. Dans la glace qui fond, il montrait 13 degrés.

Plusieurs phénomènes font face à des variations de températures. Ainsi, de nos jours, l'on distingue plusieurs types de capteurs de température, selon le phénomène en présence. En outre les thermomètres à dilatation pour la dilatation de la matière, le thermomètre à mercure, plus utilisé pour prélever la température humaine, et surtout les thermomètres électroniques que l'on rencontre dans de nombreux domaines aujourd'hui, vu l'évolution de tout ce qui est électronique.

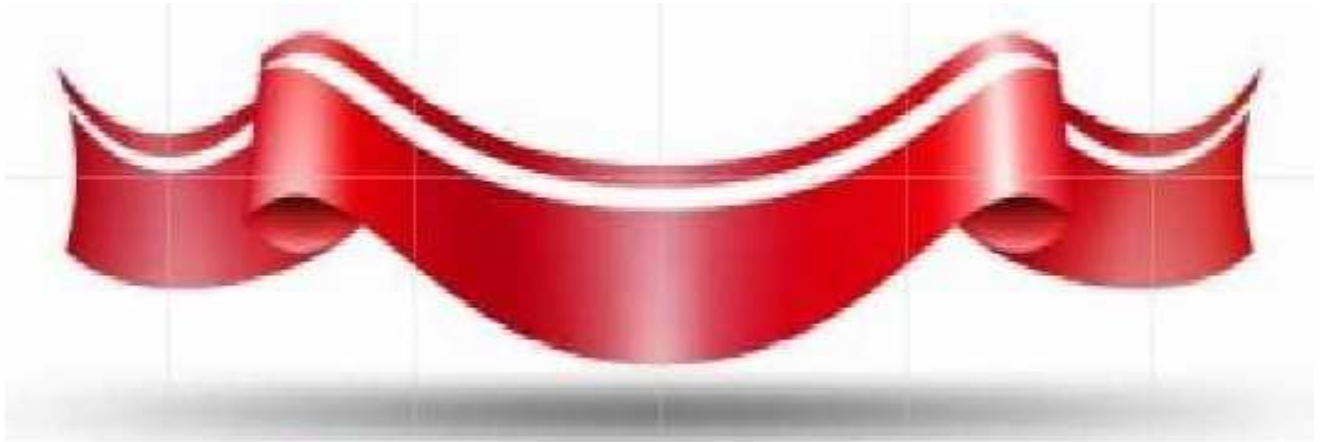
Notre cahier de charge consiste à étudier le principe de fonctionnement et de la conception des thermomètres électroniques à mémoire, afin de les réaliser. Nous allons mesurer la température à l'aide d'un capteur et d'afficher le résultat sur l'écran LCD, tous ce processus ce fait grâce à un microcontrôleur que l'on programme.

Cette réalisation est faite par deux méthodes, afin de choisir la conception la plus efficace. La différence entre les deux réalisations réside dans le choix du capteur. Pour la première conception, nous avons choisi un capteur analogique qui nécessite un convertisseur analogique numérique. Alors que dans la deuxième conception, nous avons choisi un capteur numérique qui donne les mesures directement.

Ce projet n'utilise pas les sources de température négative, donc nous allons mesurer que les températures supérieures à zéro. Nous avons choisi une plage entre 0°C et 100°C.

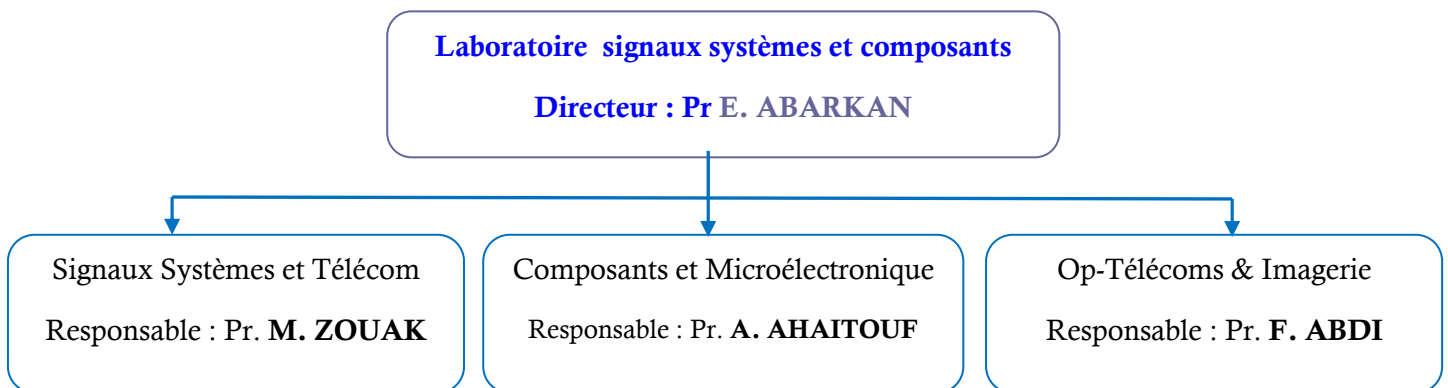
Ce document comprend trois chapitres. Pour le premier chapitre, nous allons décrire les outils utilisés dans ce projet. Pour le deuxième et le troisième chapitre, nous allons faire l'étude de processus de mesure de température, la simulation et la réalisation des thermomètres par les deux conceptions.

Présentation du laboratoire de recherche



Présentation de lieu du stage

Notre Projet de Fin d'Etudes a été effectué au Laboratoire Signaux Systèmes et Composants (L.S.S.C) situé au département génie électrique de la Faculté des Sciences et Techniques de Fès. Ce laboratoire est créé depuis 1996 sous la direction du Professeur **El Hossain ABARKAN**. L'organigramme du laboratoire est donné ci-dessous :



1. Axes de recherche et de développement :

Le laboratoire est composé de trois équipes de recherche qui développent les thèmes suivants:

* Equipe 1 : Signaux systèmes et télécommunication

- ❖ Localisation de ressources en radiocommunication mobile
- ❖ Formation à distance : E-learning
- ❖ Intelligence artificielle : Réseaux de neurones & algorithmes génétiques
- ❖ Robotique

* Equipe 2 : Composants et microélectronique

- ❖ Technologie des composants électroniques, optiques et des capteurs
- ❖ Composants à base de semi-conducteur, de faibles dimensionnalités - nano structures
- ❖ Modélisation, caractérisations et tests de fiabilité
- ❖ CAO des circuits intégrés et description par VHDL
- ❖ Conception d'IP pour la compression et le traitement d'image statiques et dynamiques (JPEG, JPEG 2000,...)
- ❖ CEM des circuits intégrés

* Equipe 3 : OP télécoms et Imagerie

- ❖ Etude des composants avancés pour les télécommunications et le traitement de l'information
 - La modulation optique : configurations, modélisation et conception
 - Caractérisation des composants pour les télécommunications optiques hauts débits (modulateurs EO massifs et intégrés MZ : étude des dérives, fibre optique cristalline : propriétés optiques non linéaires et compensation de la dispersion, capteurs à fibres
 - Elaboration et caractérisation de nouveaux matériaux céramiques pour l'optique et les radios fréquences
- ❖ Imagerie
 - Reconnaissance des formes et analyse des scènes : formes 3D tomographie.
 - Réduction de la redondance et norme de compression
 - Architecture des processeurs vidéo

Les trois équipes travaillent en étroite collaboration et développent des thématiques concordantes et complémentaires. On retrouve les équipes de microélectroniques et les équipes d'imagerie ensemble sur des projets de recherches communs et sur des thèses encadrées en commun. On retrouve aussi l'axe signaux et télécommunication en relation avec les autres équipes sur des travaux de recherches communes. Le but étant de former des jeunes chercheurs dans des domaines d'excellence et de pointe leur permettant ainsi d'acquérir des compétences variées et complémentaires capables d'aboutir à un système complet en partant des composants de base. Scientifiquement l'équipe du laboratoire a mis au point des formations phares qui ont connue et qui connaissent un franc succès il s'agit des maîtrises IEEA, Télécommunication du DESS Microélectronique et actuellement la Filière ingénieur (SET) Systèmes Electroniques et Télécommunications, le MASTER Microélectronique, de Télécommunication et Informatique Industrielle et la Licence (ETI) Electronique Télécommunication et informatique. L'objectif majeur fixé par l'équipe du laboratoire consiste à développer des thématiques de recherche en adéquation avec les filières et les compétences d'enseignement.

2. Projets de recherche de la structure :

Dans la suite nous présentons quelques projets financés sur lequel les équipes du laboratoire travaillent actuellement:

- ✿ **Projet 1:** Système de Transport Intelligent
- ✿ **Projet 2 :** Etude, Modélisation et Conception d'une architecture VLSI du codeur JPEJ2000- Vidéo et implémentation sur un circuit programmable de type FPGA. Ces deux premiers projets sont financés par le Fond National de Soutien à la Recherche Scientifique et au Développement Technologique et l'Organisme de Recherche
- ✿ **Projet 3 :** Fibres Cristallines de Niobate de Lithium pour l'Emission Infrarouge ce projet entre dans le cadre d'une convention CNRS-CNRST.
- ✿ **Projet 4 :** Conception de nouvelles structures de Convertisseurs Analogique Numérique Sigma Delta à Courants commutés Hautes performances pour application audio (ACTION INTEGREE MAROC-TUNISIENNE)
- ✿ **Projet 5 :** Localisation des sources mobiles (Action intégrée avec l'Université de Nantes et Rennes)

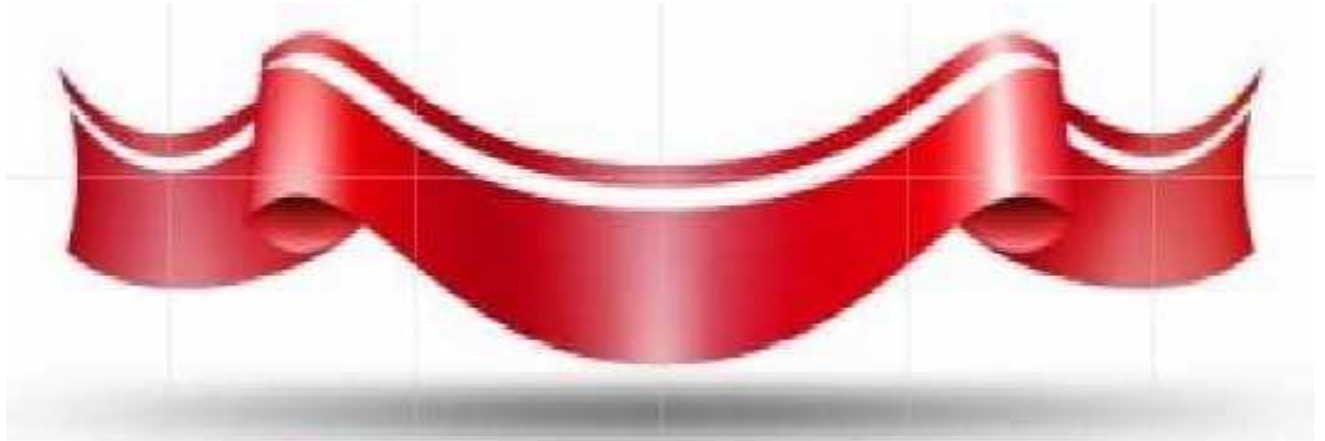
✿ **Projet6** : Electroluminescence des Dispositifs Electroniques Microélectroniques et Optoélectroniques : Mesures, Simulation et Modélisation (Projet d'Université Sidi Mohammed Ben Abdellah).

3. Ressources humaines

L'équipe du laboratoire LSSC est composée de 14 Professeur de l'enseignement de l'enseignement supérieur, de 4 Professeur habilité, d'un Professeur assistant et 20 doctorants.

Chapitre 1:

Les outils du projet



I. Introduction :

Avec les progrès de la technologie, plus particulièrement de l'électronique, il n'est plus rare de nos jours de voir les appareils usuels remplacés par des appareils électroniques. Nous pouvons penser entre autres aux thermomètres électroniques.

Pour réaliser notre thermomètre électronique, nous avons utilisé de nouveaux logiciels et composants.

II. Les capteurs de température :

1. Définition :

Cet organe est chargé de prélever une grandeur physique et de la transformer en grandeur exploitable.

La grandeur physique (ou mesurande) n'est généralement pas exploitable directement, elle constitue le signal d'entrée du capteur. La grandeur exploitable est souvent de nature électrique. Elle constitue le signal de mesure (ou sortie ou réponse) du capteur. Elle est une image de la grandeur à mesurer et doit être indépendante des autres grandeurs perturbatrices.

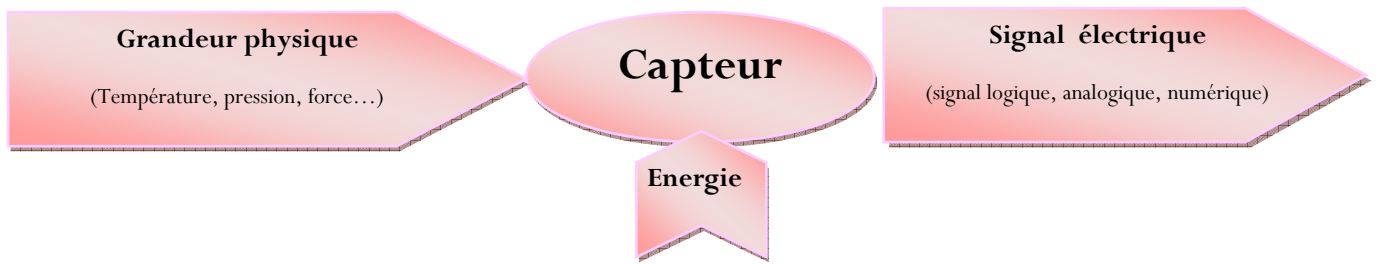


Figure 1.1 : Fonctionnement des capteurs

2. *Les types de capteurs :*

Les capteurs à sortie électrique peuvent être classés selon trois grandes catégories :

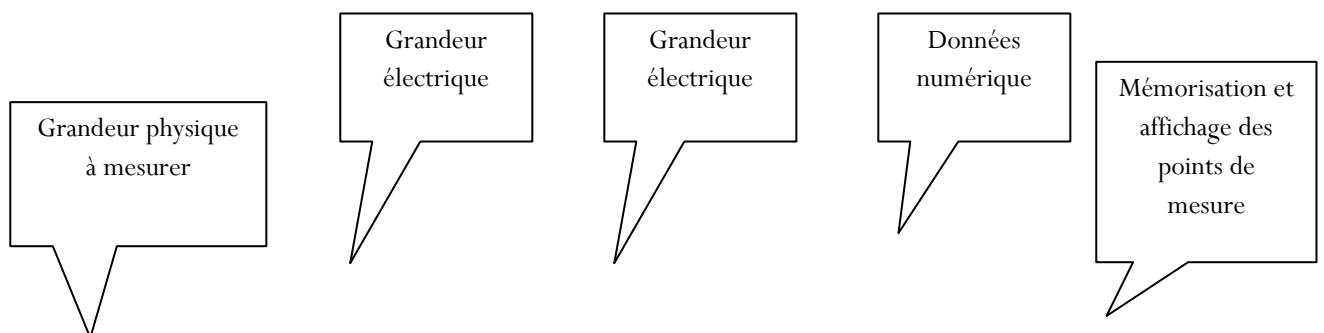
- * **Capteurs analogiques :** le signal, élaboré par le capteur, est de nature analogique.
- * **Capteurs numériques :** Le signal élaboré par le capteur, est directement codé sous une forme numérique au sein du même capteur.
- * **Capteurs logiques :** le signal ne comporte que deux états ou valeurs possibles, on parle de capteur tout ou rien.

3. *Mode de fonctionnement des capteurs :*

Les capteurs fonctionnent selon deux principes de base suivant l'origine du signal électrique de sortie. On distingue :

- * **les capteurs actifs** fonctionnant en générateur, une partie de l'énergie physique prélevée sur le mesurande est transformée directement en une énergie électrique qui constitue le signal de sortie.
- * **les capteurs passifs**, il s'agit généralement d'impédance dont l'un des paramètres déterminants est sensible à la grandeur mesuré.

4. *Chaine de mesure :*



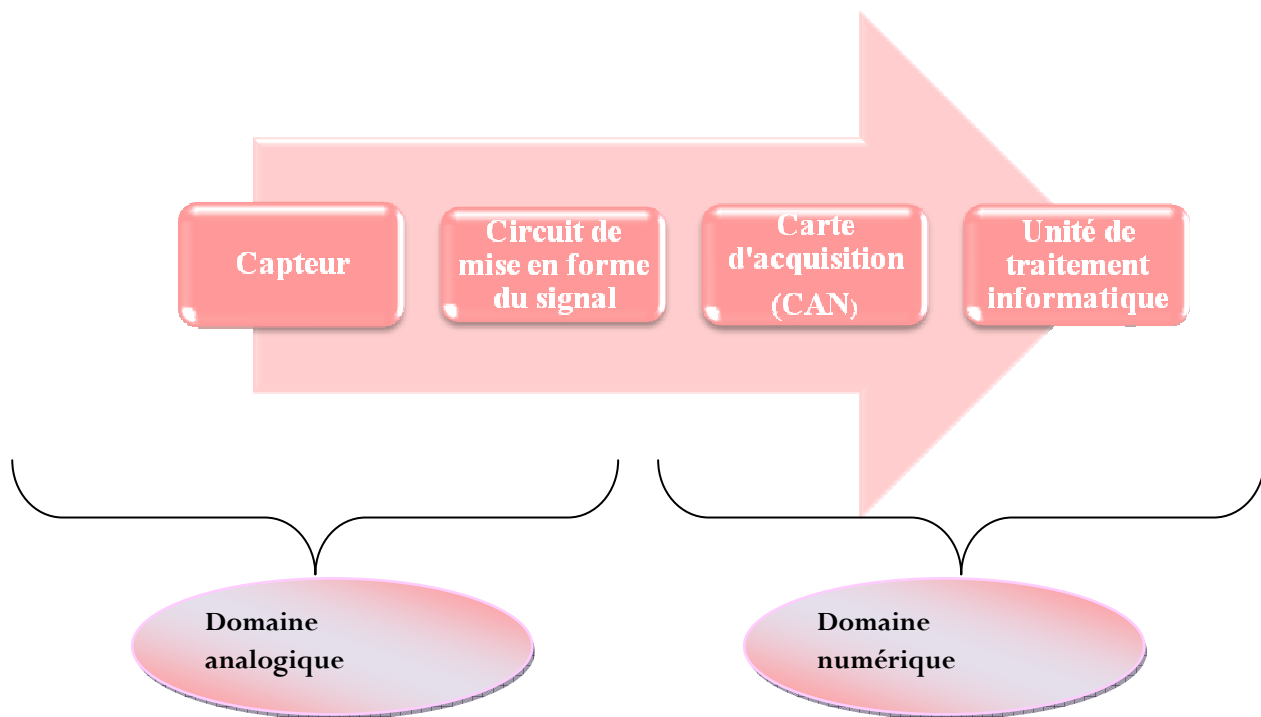


Figure 1.2 : chaîne de mesure des capteurs

La structure de base d'une chaîne de mesure comprend au minimum quatre étages :

- * **Un capteur** sensible aux variations d'une grandeur physique et qui, à partir de ces variations, délivre un signal électrique.
- * **Un conditionneur de signal** dont le rôle principal est l'amplification du signal délivré par le capteur pour lui donner un niveau compatible avec l'unité de numération ; cet étage peut parfois intégrer un filtre qui réduit les perturbations présentes sur le signal.
- * **Une unité de numérisation** qui va échantillonner le signal à intervalles réguliers et affecter un nombre (image de la tension) à chaque point d'échantillonnage.
- * **L'unité de traitement informatique** peut exploiter les mesures qui sont maintenant une suite de nombres (enregistrement, affichage de courbes, traitement Mathématiques, transmissions des données).

De nos jours, compte tenu des possibilités offertes par l'électronique et l'informatique, les capteurs délivrent un signal électrique et la quasi-totalité des chaînes de mesure sont des chaînes électronique et informatique.

Remarque : Certains capteurs, par exemple le thermomètre DALLAS DS 1820, délivrent directement un mot binaire, image de la température, en leur sortie. Ils intègrent le capteur+le circuit de mise en forme+CAN.

5. *Les éléments de métrologie :*

- * **L'incertitude (dX) :** Le résultat de la mesure (x) d'une grandeur(X) n'est pas complètement défini par un seul nombre. Il faut au moins la caractériser par un couple(x, dx) et une unité de mesure. dx est l'incertitude sur x. Les incertitudes proviennent des différentes erreurs liées à la mesure. Ainsi on a : $x-dx < X < x+dx$.

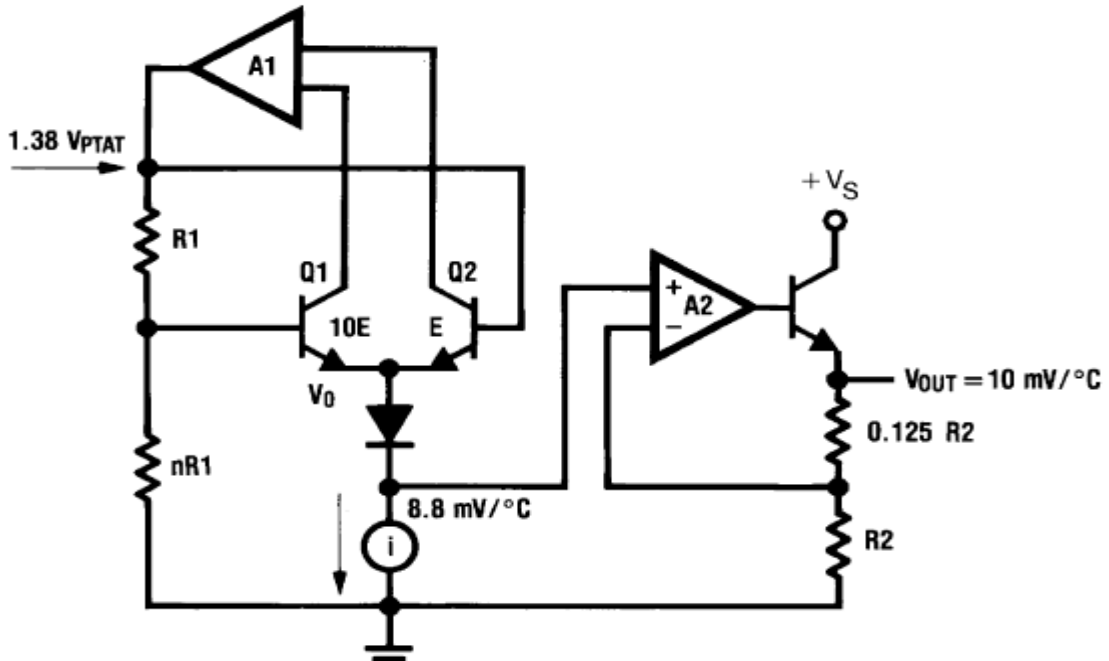


Figure 1.4 : Architecture interne d'un LM35DZ

b. Capteur de température numérique :

Dans la deuxième partie du projet nous avons utilisé le capteur de température à sortie numérique, fabriqué par DALLAS de type DS1820.

Le DS1820 est un capteur de température pouvant travailler sur une plage allant de -55°C à $+125^{\circ}\text{C}$ avec une précision atteignant $0,5^{\circ}\text{C}$, mais on peut améliorer la résolution par un algorithme. Un des avantages de ce type de capteur est qu'il est connecté au microcontrôleur que par 1 seul fil (1-Wire). Pas besoin donc d'un convertisseur CAN pour récupérer l'information de température. L'interface 1-Wire permet une grande simplification du circuit en général mais aussi, entre autres, de connecter plusieurs capteurs sur le même pin de notre microcontrôleur.

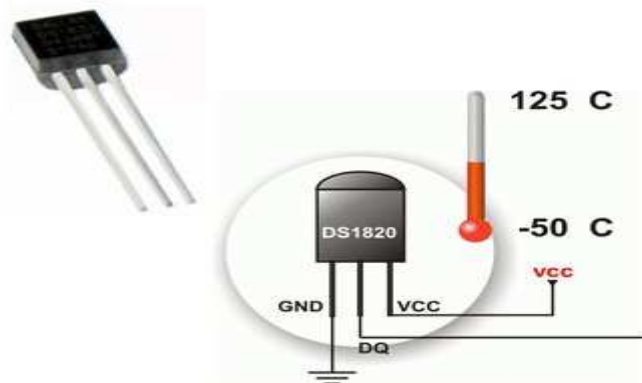


Figure 1.5 : Brochage d'un capteur DS1820

L'architecture interne de ce capteur est la suivante :

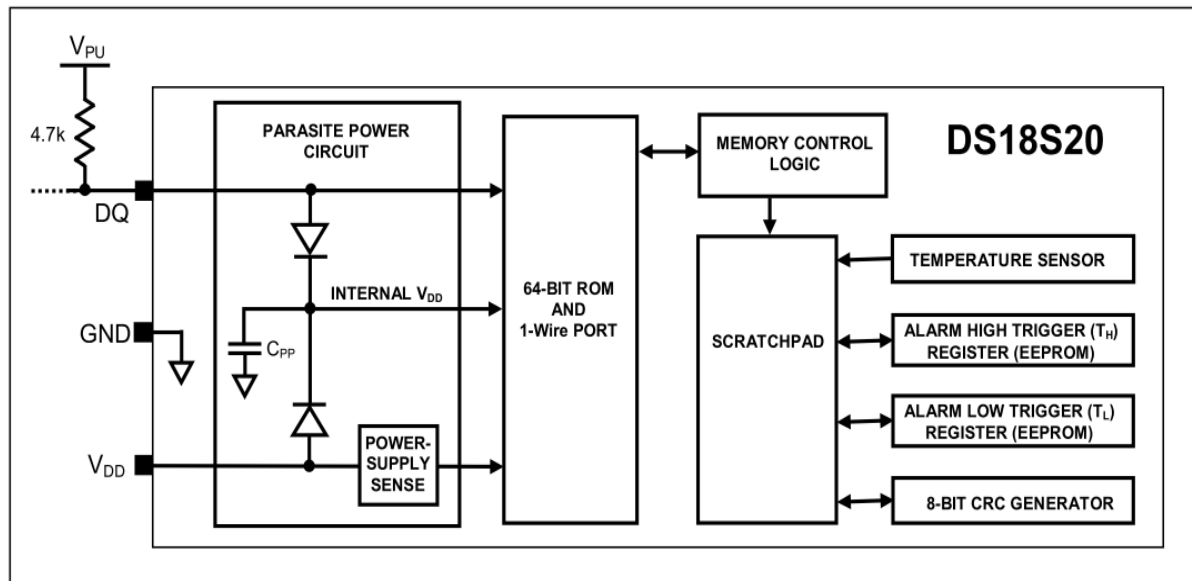


Figure 1.6 : Architecture interne d'un DS18S20

Le temps de conversion maximale est de 750 ms et les données sont stockées dans un registre dénommé Scratchpad, ou il faut aller les récupérer lors de la programmation.

III. Les PIC :

1. Introduction :

Le PIC est un élément indispensable dans ce projet, il permet d'afficher la température sur le LCD, 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 General Instrument. Cependant, à l'époque du développement du PIC1650 par General Instrument, PIC était un acronyme de « Programmable Intelligent Computer » ou « Programmable Integrated Circuit ».

2. Architecture interne :

Un microcontrôleur est une unité de traitement de l'information de type microprocesseur à laquelle on a ajouté des périphériques internes permettant de réaliser des montages sans nécessiter l'ajout de composants annexes. Un microcontrôleur peut donc fonctionner de façon autonome après programmation.

Le schéma ci-dessous présente les principaux blocs fonctionnels présents à l'intérieur d'un PIC ainsi qu'une courte description. Pour un schéma plus précis, se référer à la datasheet d'un PIC.

Mémoire
RAM

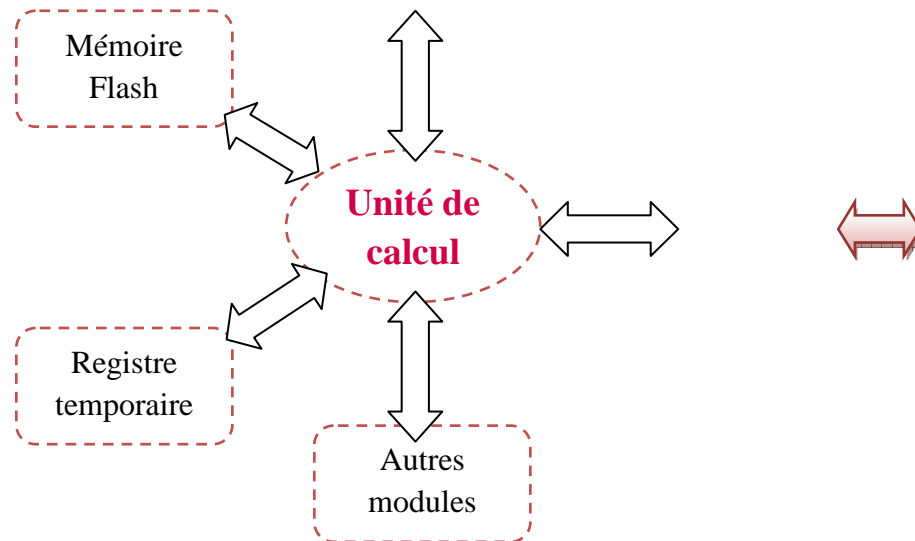


Figure 1.7 : Architecture simplifié d'un PIC

Description des blocs fonctionnels:

- * **Mémoire flash** : C'est une mémoire réinscriptible qui conserve ses données lorsque le PIC n'est pas alimenté. Elle est utilisée pour stocker le programme. A chaque ligne du programme est attribuée une adresse qui permettra à l'unité de calcul de se repérer.
- * **Mémoire RAM** : C'est une mémoire volatile (qui s'efface quand le PIC n'est plus alimenté). Les variables utilisées au cours du programme sont stockées à cet endroit.
- * **Unité de Calcul** : C'est le cœur du microcontrôleur. Ici se déroulent toutes les opérations à une vitesse définie par la fréquence d'horloge (fréquence d'oscillation divisée par 4).
- * **Registre temporaire W** : C'est ici qu'est stockée la valeur nécessaire pour une opération de l'unité de calcul.
- * **Ports E/S (Entrées/Sorties)** : Ce sont les unités qui font le lien entre ce qui se passe à l'intérieur du PIC et l'extérieur.
- * **Modules annexes** : Toutes les fonctions annexes (timers, comparateurs, convertisseurs analogiques/numériques ...).

3. **Familles de PIC :**

➤ Les modèles de PIC courants sont repérés par une référence de la forme :

- * **2 chiffres** : famille du PIC (10, 12, 16, 17 et 18).
- * **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. Un L peut être ajouté devant pour indiquer qu'il s'agit d'une modèle basse tension.
- * **un nombre de 2 à 4 chiffres** : modèle du PIC au sein de la famille.
- * **un groupe de lettres** pour indiquer le boîtier et la gamme de température.

Par exemple, le PIC18LF4682-I/P est un microcontrôleur de la famille PIC18, basse tension (L), à mémoire flash (F), modèle 4682, gamme de température industrielle (I) et boîtier DIL40.

4. *Le PIC utilisé dans le projet PIC 18F8520:*

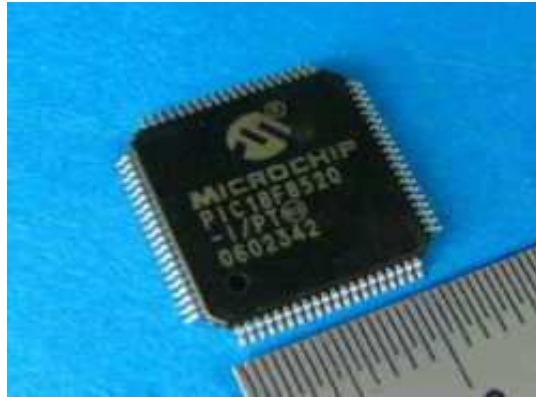


Figure 1.8 : PIC18F 8520

a. *Description :*

- * MICROCONTROLEUR 8 BITS FLASH CMS, série: PIC18,
- * Nombre d'E/S:68,
- * Taille mémoire de programme:16 kWords,
- * Taille mémoire, RAM:2048Byte,
- * Vitesse de processeur:40MHz,
- * Nombre de timers:6
- * Périphériques: ADC, PWM, Timer,
- * Type de boîtier CI numérique: TQFP,
- * Gamme de tension d'alimentation:4.2V à 5.5V,
- * Température de fonctionnement:-40°C à +85°C,
- * Nombre de broches:80,
- * Fréquence, horloge:40MHz,
- * Nombre d'entrées CAN:16,
- * Nombre de bits CAN:10,
- * Numéro de la fonction logique:18F8520.

b. *Architecture :*

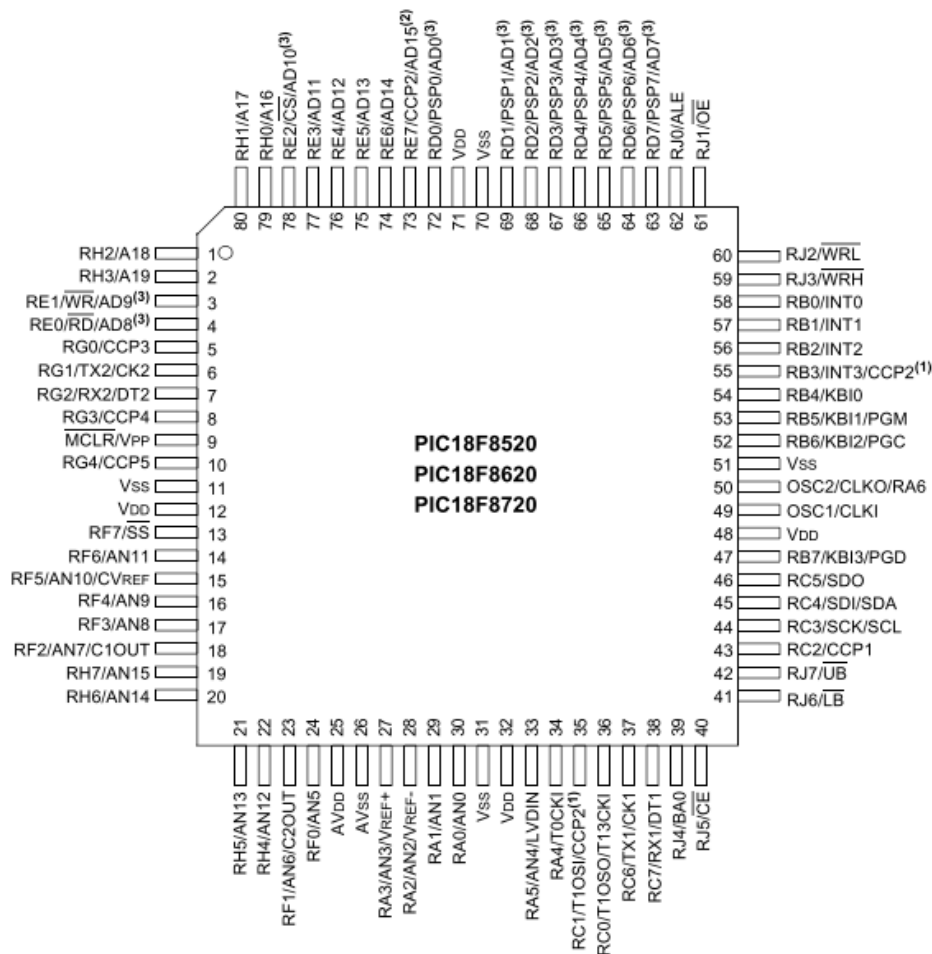


Figure 1.9 : Architecture d'un PIC 18F85200

4. *Programmation des PIC :*

Les PIC sont des processeurs dits RISC, c'est-à-dire processeur à jeu d'instruction réduit. Plus on réduit le nombre d'instructions, plus facile et plus rapide en est le décodage, et plus vite le composant fonctionne. Cependant, il faut plus d'instructions pour réaliser une opération complexe.

Le cycle instruction d'un PIC 8 bits dure 4 coups d'horloge. La plupart des instructions durent un cycle, sauf les sauts qui durent deux cycles. On atteint donc des vitesses élevées.

Avec un quartz de 4 MHz (ou l'horloge interne), on obtient donc 1000000 de cycles/seconde, or, comme le PIC exécute pratiquement 1 instruction par cycle, hormis les sauts, cela donne une puissance de l'ordre de 1 MIPS (1 million d'instructions par seconde).

IV. La carte « BIGPIC6 »:

1. *Présentation du système :*

Le système de développement BigPic6 est un outil de grande qualité destiné à la programmation et à l'expérimentation sur microcontrôleur PIC de Microchip. La carte inclue un programmeur avec mikroICD servant l'interface entre le microcontrôleur et le pc utilisateur. On n'a donc qu'à écrire un code dans un des compilateurs qui sont compatible avec cette carte, générer un fichier .hex puis programmer le microcontrôleur grâce au programmeur PICflash. De nombreux modules embarqués, comme l'afficheur LCD graphique 128x64, l'afficheur LCD alphanumérique 2x16, extenseur de port etc.... seront d'une grande aide pour la simulation.



Figure 1.10 : La carte BigPic6

V.

a.

Introduction sur les LCD:

Les afficheurs LCD (Liquid Crystal Display) sont devenues incontournables dans toutes applications qui demandent la visualisation de paramètres, il s'agit donc d'une interface Homme/Machine. Ils sont très utilisés dans les montages à microcontrôleur, et permettent une grande convivialité. Ils peuvent aussi être utilisés lors de la phase de développement d'un programme, car on peut facilement y afficher les valeurs de différentes variables. Au paravent onéreux et difficile à mettre en œuvre, ils sont maintenant bon marché.



Figure 1.11 : Afficheur LCD

b.

Principe des LCD :

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 réfléchifs). 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 réfléchif 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.

c. *Brochage :*

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

<u>Broche</u>	<u>Nom</u>	<u>Niveau</u>	<u>Fonction</u>
<u>1</u>	<u>Vss</u>	-	<u>Masse</u>
<u>2</u>	<u>Vdd</u>	-	<u>Alimentation positive +5V</u>
<u>3</u>	<u>Vo</u>	<u>0-5V</u>	<u>Cette tension permet, en la faisant varier entre 0 et +5V, le réglage du contraste de l'afficheur.</u>
<u>4</u>	<u>RS</u>	<u>TTL</u>	<u>Sélection du registre (Register Select)</u> <u>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.</u>
<u>5</u>	<u>R/W</u>	<u>TTL</u>	<u>Lecture ou écriture (Read/Write)</u> <u>L : Écriture</u> <u>H : Lecture</u>
<u>6</u>	<u>E</u>	<u>TTL</u>	<u>Entrée de validation (Enable) active sur front descendant. Le niveau haut doit être maintenue pendant au moins 450 ns à l'état haut.</u>
<u>7</u>	<u>D0</u>	<u>TTL</u>	<u>Bus de données bidirectionnel 3 états (haute impédance lorsque E=0)</u>
<u>8</u>	<u>D1</u>	<u>TTL</u>	
<u>9</u>	<u>D2</u>	<u>TTL</u>	
<u>10</u>	<u>D3</u>	<u>TTL</u>	
<u>11</u>	<u>D4</u>	<u>TTL</u>	
<u>12</u>	<u>D5</u>	<u>TTL</u>	

<u>13</u>	<u>D6</u>	<u>TTL</u>	
<u>14</u>	<u>D7</u>	<u>TTL</u>	
<u>15</u>	<u>A</u>	-	<u>Anode rétro éclairage (+5V)</u>
<u>16</u>	<u>K</u>	-	<u>Cathode rétro éclairage (masse)</u>

Tableau 1.1 : Brochage de LCD

d. 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.

*** 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 :

<u>Ligne</u>	<u>Visible</u>	<u>Invisible</u>
<u>Haut</u>	<u>00H.....0FH</u>	<u>10H.....27H</u>
<u>Bas</u>	<u>40H.....4FH</u>	<u>50H.....67H</u>

Tableau 1.2 : adresses correspond à la DD RAM

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.

*** 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.

La CG RAM 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 la CG RAM qui est actuellement sur l'afficheur est changé, alors le changement est immédiatement apparent sur l'afficheur.

VI. Logiciel ISIS :

1. Introduction:

Proteus est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels inclus dans Proteus permettent la CAO (conception assistée par ordinateur) dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle: ISIS, ARES.

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle. Outre la popularité de l'outil, Proteus possède d'autres avantages :

- ✓ Pack contenant des logiciels facile et rapide à comprendre et utiliser.
- ✓ Le support technique est performant.
- ✓ L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet.

2. ISIS: Intelligent Schematic Input System

Le logiciel ISIS de Proteus est principalement connu pour éditer des schémas électriques puis procéder à une animation en temps réel du montage. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.



Figure 1.12 : Logiciel ISIS

3. Prise en main du logiciel ISIS :

a. La saisie de schéma:

L'écran se compose de plusieurs parties:

- * la zone de travail, dans laquelle sera disposé le schéma,
- * la zone de contrôle qui permet de se déplacer d'un endroit à un autre de la zone de travail,
- * le clavier,
- * la liste des objets sélectionnés pour réaliser le schéma.

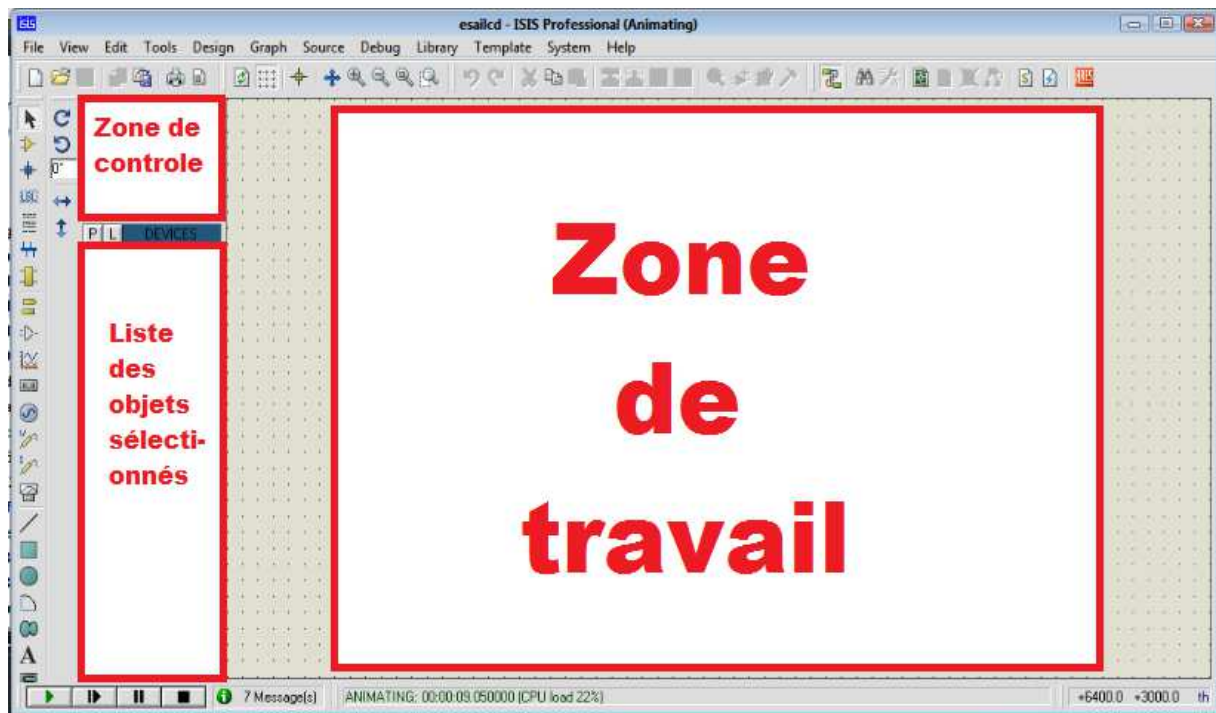
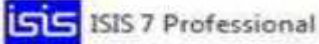



Figure 1.23 : Présentation de page d'ISIS

* Démarrer le logiciel :

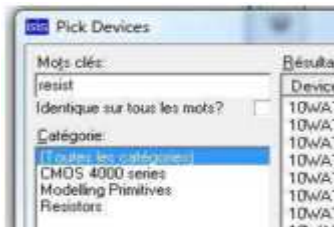
On lance depuis le raccourci du bureau ou du menu Démarrer de l'ordinateur le logiciel Proteus – ISIS : 

* Création de la liste des composants :

✓ On sélectionne l'icône composant : 

✓ Dans la fenêtre du sélectionneur d'objets, on clique sur P : 

-Une nouvelle fenêtre (**Pick Devices**) de choix de composants s'ouvre : On trouve les composants désirés à partir **d'un mot clé**, ou dans **la catégorie** proposée, puis dans la **sous-catégorie** (par défaut choisissez dans la sous-catégorie **Generic**).



-On clique une fois pour faire apparaître le composant, s'il convient on double clique pour qu'il s'inscrive sur la liste de travail **DEVICES**.

✓ On procède ainsi, pour placer dès le début du travail tous les composants que nous aurons besoin par la suite sur la liste du travail, puis on ferme la fenêtre en cliquant sur OK.

* Placement des composants sur le schéma :

✓ On sélectionne le composant à placer dans la liste faite précédemment et le placer dans la zone de travail de la fenêtre d'édition.

✓ Eventuellement le tourner ou lui faire une symétrie à l'aide des outils d'orientation du menu de la boîte à outils :



✓ Pour les composants placés, il faut définir leurs paramètres au cas du besoin (Par exemple : définir la valeur d'une résistance): on clique gauche sur le composant, le symbole devient rouge, on re clique gauche, une fenêtre pour éditer les propriétés s'ouvre.


* Placement des différentes connexions externes :

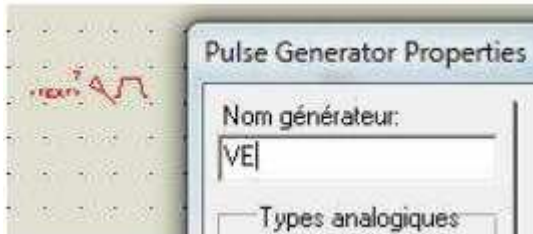
Pour **GROUND** ; **OUTPUT** ; **INPUT**...

- ✓ Après avoir cliqué sur cette fonction  , on sélectionne la connexion et après l'avoir mise dans le « panier », la placer sur le schéma.

* **Placement des alimentations et des générateurs :**

Il faut maintenant placer les différentes alimentations ou générateurs utilisés.


- ✓ On clique sur  et on sélectionne un générateur (DC, SINE, PULSE...).




- ✓ Ensuite on clique sur le schéma pour le placer (ici Pulse = rectangle).
- ✓ Les alimentations ou générateurs placés, il faut leur donner un nom et régler les différents paramètres qui sont fonctions du type de générateurs choisis.
- ✓ Lorsqu'on clique gauche sur le générateur, le symbole devient rouge, lorsqu'on recliqgue gauche, une fenêtre pour éditer les propriétés s'ouvre. Grâce à cette fenêtre on donne un nom et une amplitude des tensions, et on peut aussi régler la forme de la fréquence du signal.

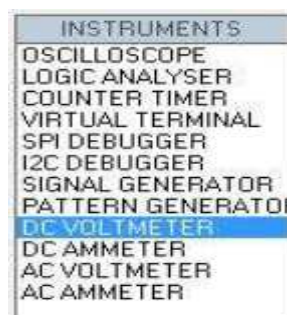
* **Réalisations des connexions :**

Il nous reste à relier les connexions entre les différents symboles.

- ✓ Sélectionner dans le menu :  Aller sur l'extrémité d'un composant, cliqué, le début de la liaison est créé, puis aller cliquer sur le composant à relier pour finir la liaison, le logiciel positionne seul le chemin.

b. Placement des instruments de mesure :

Il faut choisir le ou les appareils de mesure que l'on souhaite utiliser en cliquant sur l'icône : .

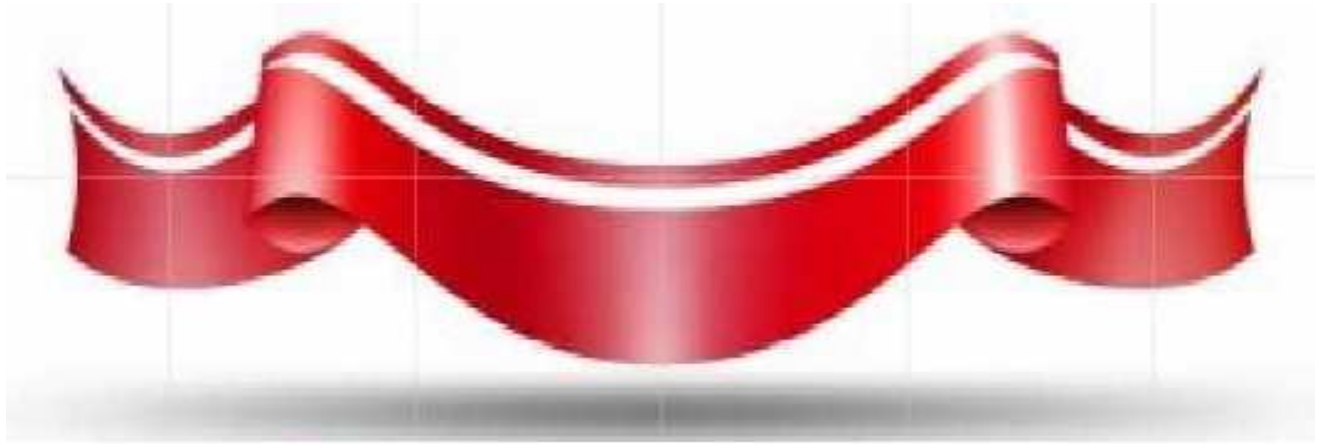


c. Animation et simulation du fonctionnement :

Pour lancer l'animation. Cliquer sur la flèche, comme sur un appareil multimédia.



Chapitre 2:
Thermomètre à base
d'un capteur analogique



I. Introduction :

Dans cette partie, on a utilisé le capteur analogique LM35DZ pour réaliser le thermomètre. Les étapes séquentielles de la mesure de la température sont les suivantes :

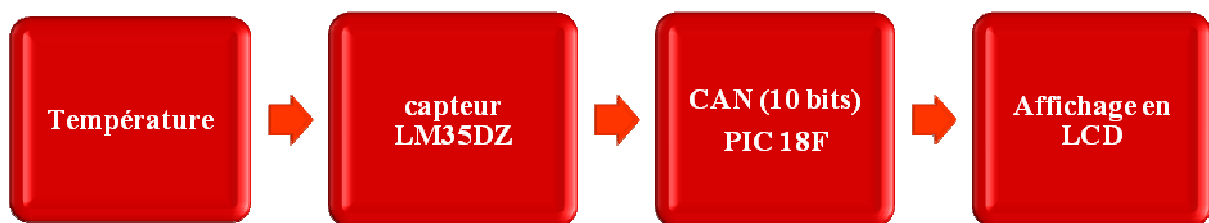


Figure 2.1 : chaîne de mesure de température

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).

2.

La conversion de la valeur analogique :

La tension issue de ce capteur est numérisée régulièrement par le convertisseur analogique numérique du PIC 18F en mode 10 bits.

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

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

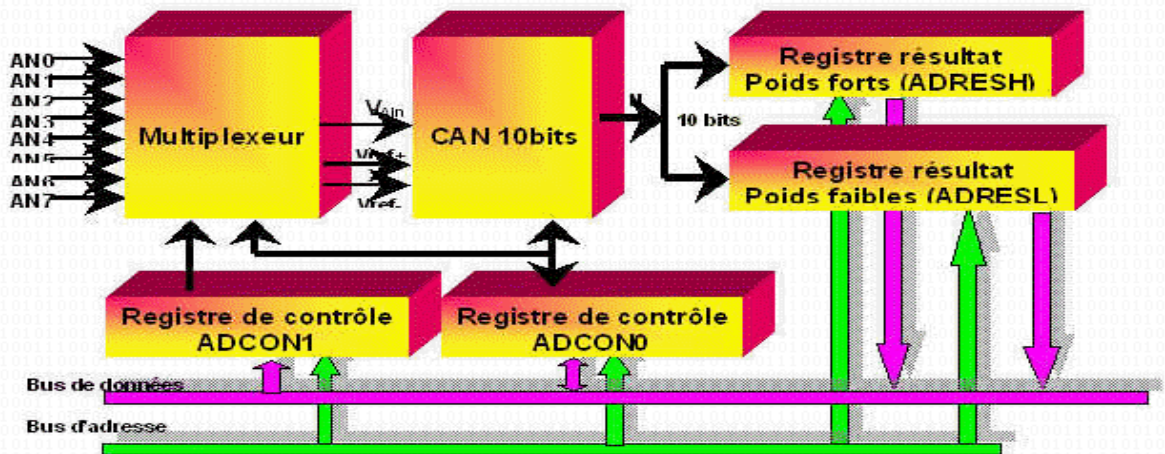


Figure 2.3 : Schéma fonctionnel de la fonction conversion analogique numérique du PIC 18F

- ✓ 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 8 bits 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.

**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).

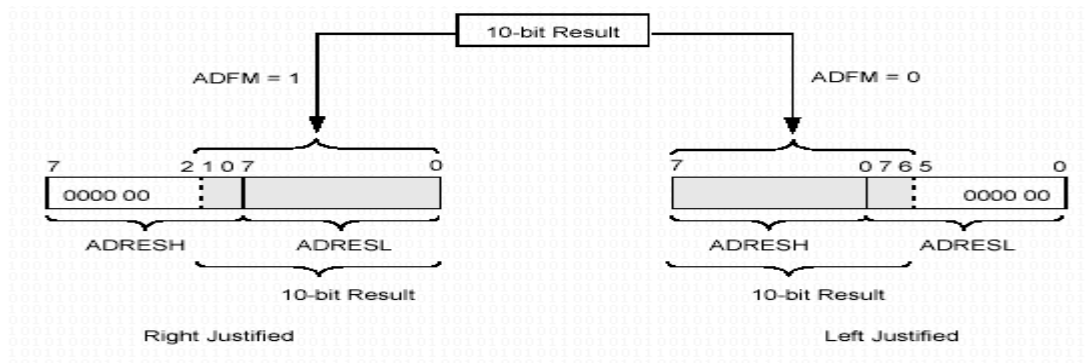


Figure 2.4 : Schéma fonctionnel de ADRESH et ADRESL



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 ;
- ✓ savoir lorsque celle-ci est achevée.



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).



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 (C_{HOLD}). 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 $2 T_{AD}$ est nécessaire entre deux conversions. T_{AD} temps de conversion analogique numérique pour un bit. Une conversion analogique numérique complète nécessite $12 T_{AD}$ pour les 10 bits de conversion.

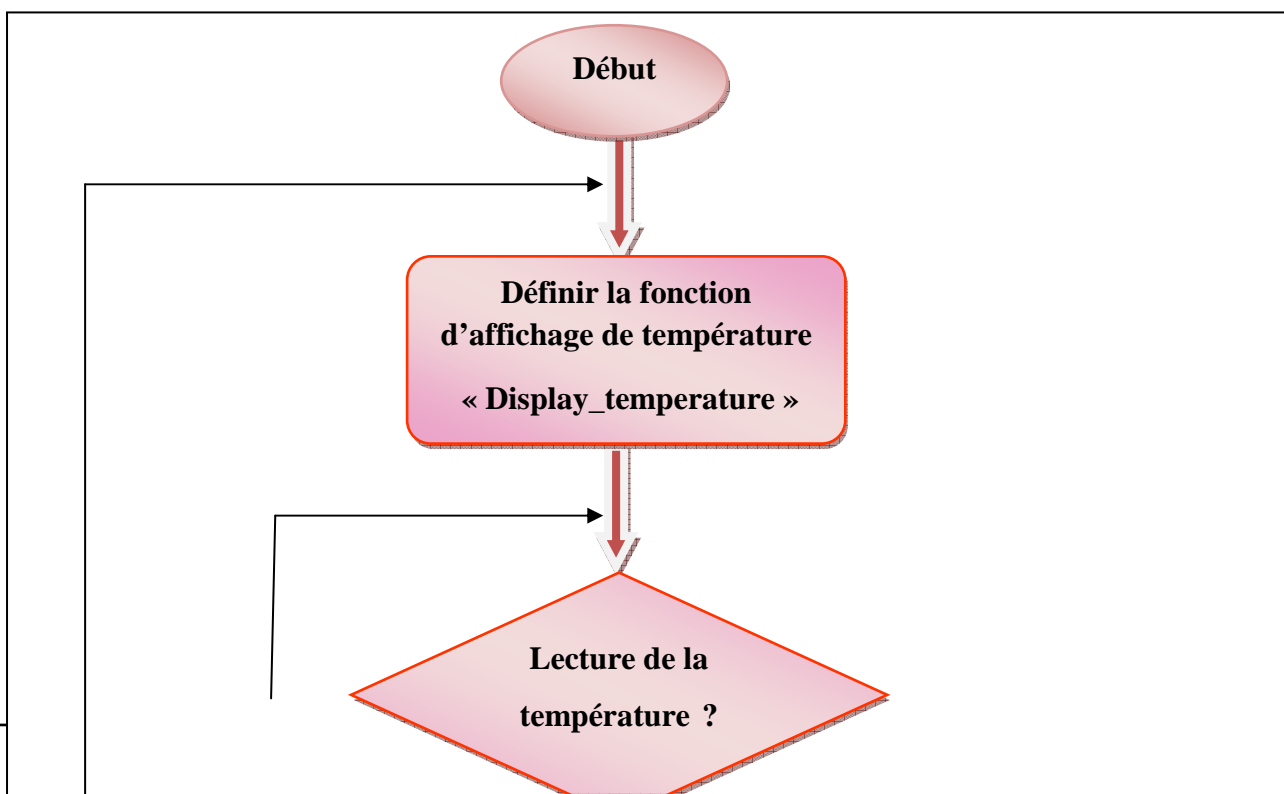
3. *Programme de mesure de température:*

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.



L'organigramme :



Non

Oui

Appel à la fonction
« Display_température »

Convertir la valeur

Afficher en LCD

III.

réalisation du

thermomètre :

Figure 2.5 : Organigramme de calcul de température

1. Branchement :

Un convertisseur A/N situé dans le microcontrôleur convertit une tension analogique en un nombre codé sur 10 bits. Une tension d'entrée variant entre 0V et 5V peut être fournie par les entrées de test A/N. Le jumper J11 permet de sélectionner la broche sur laquelle la conversion aura lieu à savoir R0, R1, R2 ou R3. La résistance R16 a une fonction de protection dans la mesure où elle sert à limiter le courant à travers le potentiomètre et les broches du microcontrôleur. La valeur de l'entrée analogique peut être modifiée linéairement en utilisant le potentiomètre P3.



Figure2.6 : RA0 utilisé comme entrée du CAN

La conversion A/N s'effectue via la broche RA0

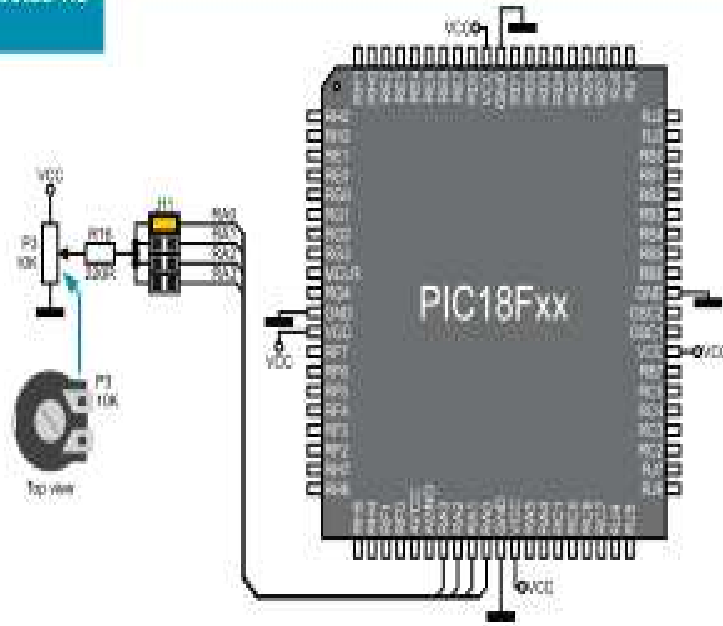


Figure 2.7 : Connexion entre PIC et CAN

2. *Simulation sous ISIS :*

Le logiciel permet de simuler le schéma du thermomètre afin de détecter certaines erreurs dès l'étape de conception.

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).

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

Après création d'un programme en MikroC, 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 laquelle il va lire la valeur à fin de la convertir. La deuxième étape est de charger le PIC par le programme.

Avec LM35: une tension de 0 V (valeur CAN = 0) correspond à 0 °C et une tension de 4,88 V (valeur CAN = 1000) correspond à 100 °C. Le montage qui suit effectue l'adaptation requise, le gain de l'étage amplificateur est de 4,88 (4,86 en théorie).

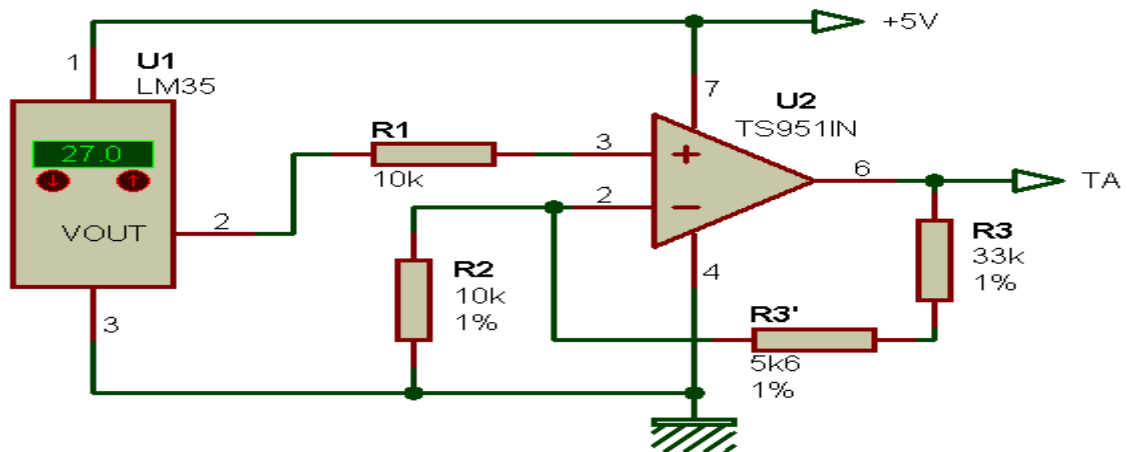


Figure 3.8 : schéma d'adaptation pour le LM35DZ

L'amplificateur opérationnel (AOP) U2 qui se charge de ce travail d'amplification est un modèle TS9511IN de type "rail-to-rail". Cela est impératif si on veut pouvoir bénéficier d'un affichage correct sur les premiers et derniers degrés de la plage de valeur. Si on avait employé un AOP standard (style LM741), la plage de mesure aurait été limitée tant dans les valeurs basses que dans les valeurs élevées (ordre de grandeur de la plage qui aurait été réellement utilisable : de +10 °C à +90 °C). Notez que même avec un AOP rail-to-rail tel que celui utilisé, on ne peut pas vraiment descendre à 0 °C. Le fait que l'on n'ait besoin que de 4,88 V pour représenter la température de 100 °C ne nous arrange bien. La mesure est fiable à partir de 2 °C. Il y a bien sûr moyen de faire mieux, mais le circuit présenté ici à l'avantage d'être vraiment simple pour une plage de valeur qui est tout de même plus que correcte. Les résistances R2, R3 et R3' doivent impérativement être des modèles de précision 1% ou mieux, de façon à garantir un taux d'amplification aussi proche que possible de celui recherché.

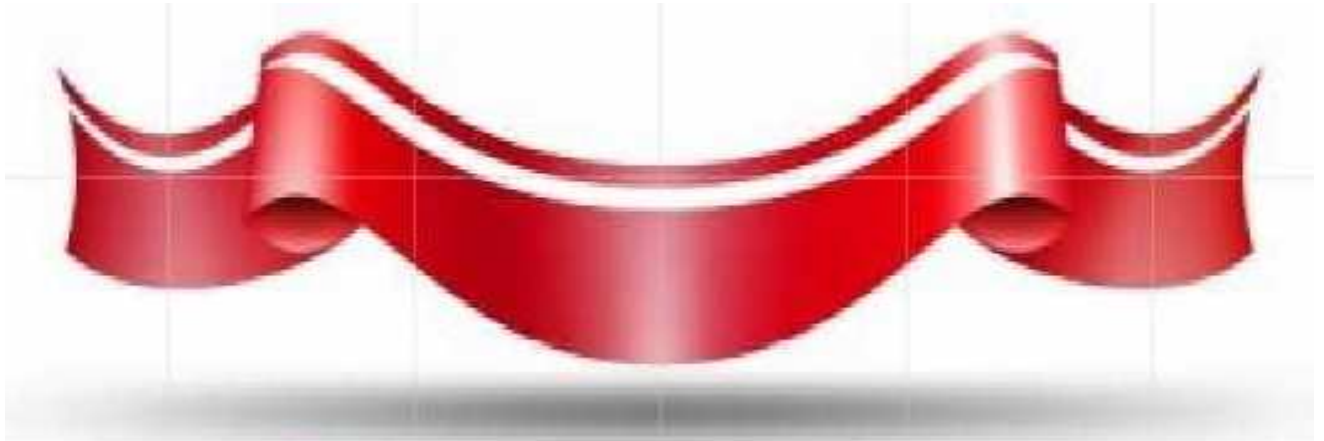
IV. Conclusion :

Le LM-35 appareil fournit un moyen idéal pour mesurer avec précision la température.

Il est également très facile de lire le LM-35 en utilisant un convertisseur analogique numérique.

Parmi les points forts de ce capteur, mentionnons sa consommation très faible (de l'ordre de 60 μ A), d'où une puissance dissipée également très faible, et sa linéarité qui demeure excellente sur toute sa plage de sensibilité. Cette plage va de 0°C à 100°C pour les LM35DZ.

Chapitre 3:
Thermomètre à base
d'un capteur numérique



I. 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 :

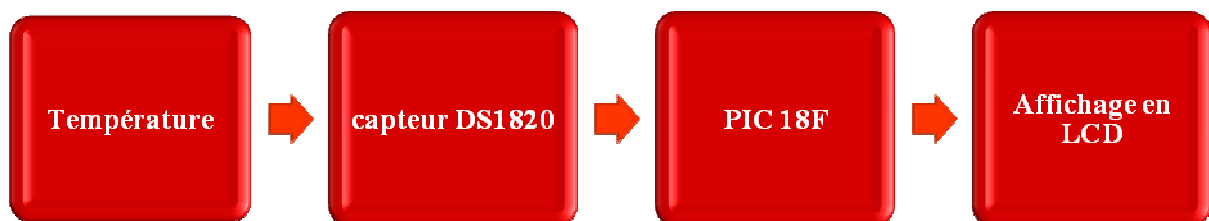


Figure 3.1 : processus de mesure de température

Tout le processus s'effectue dans la carte BigPic6.

II. Etude de processus de mesure de température :

1. 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 bytes 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.

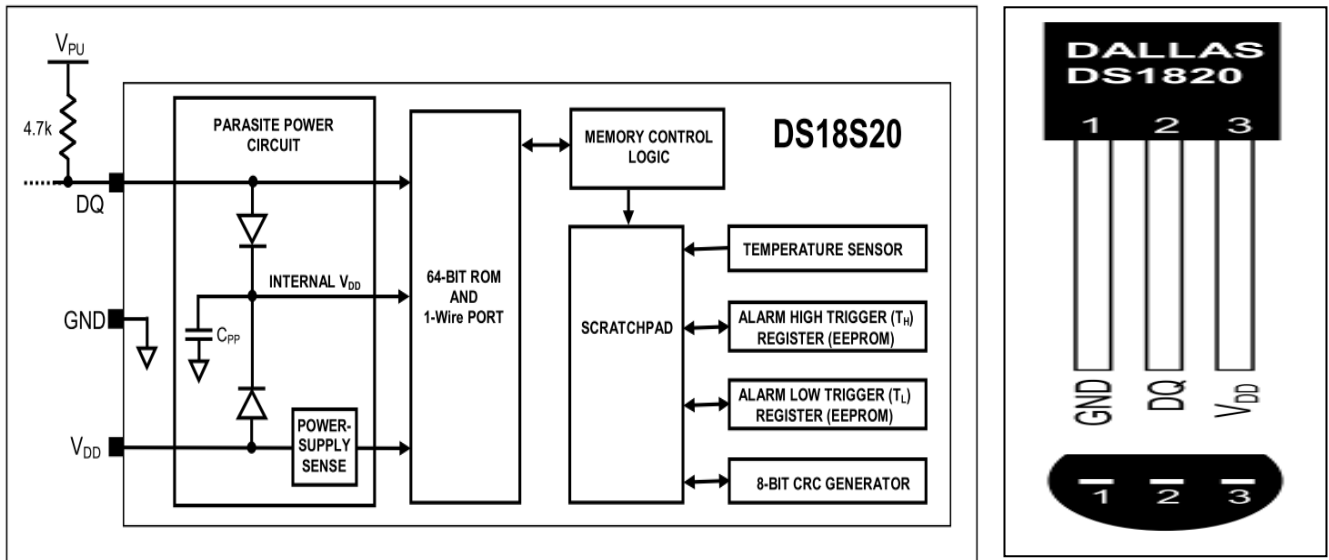


Figure 3.2 : Architecture interne d'un DS1820

2. *Protocole de communication avec le PIC :*

Le capteur communique avec le PIC à l'aide d'une entrée/sortie digitale ; le montage d'une résistance de 4,7 kΩ est nécessaire sur le bus de données.

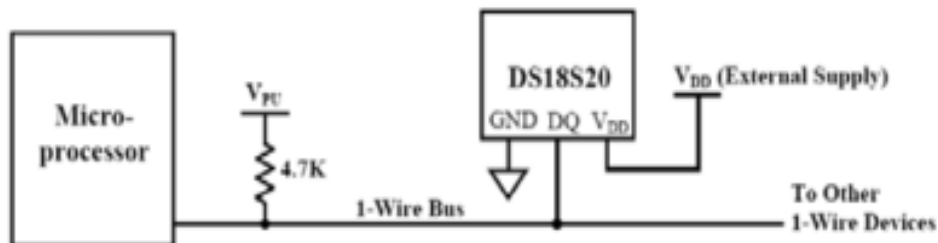


Figure 3.3 : Protocole de communication entre PIC et DS1820



Protocole de communication ONE-Wire :

Le bus 1-Wire (ou OneWire) est un bus conçu par Dallas Semiconductor. Le niveau de tension utilisé sur ce bus est +5V (niveau TTL).

Ce bus supporte une topologie série, parallèle ou en étoile. Il fonctionne suivant le principe maître / esclave.

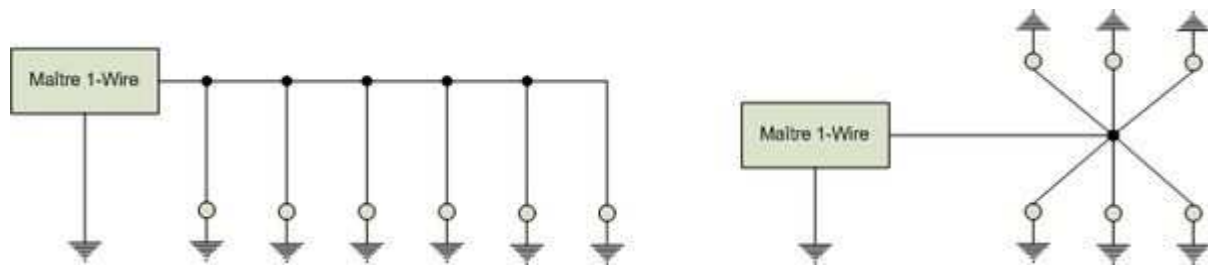


Figure 3.4 : Différent câblage de One-Wire

L'avantage de ce bus est qu'il peut être utilisé en mode « parasite » (alimentation à partir du fil de données). Cela permet d'utiliser seulement 2 fils (et non un seul comme le nom le laisse supposer), un fil de données et un fil de masse.

Généralement utilisé pour des mesures de températures, il existe une gamme complète de composants compatibles.

Chaque circuit possède une adresse physique unique, gravée dans la puce à la fabrication. C'est une des raisons expliquant la réticence de Dallas à permettre la création d'esclave 1-Wire autres que ceux qu'il produit : avoir la maîtrise de l'identification des esclaves du bus.



Initialisation :

Avant toute opération sur le DS1820 celui doit être initialisé suivant une séquence. Lorsque l'impulsion d'initialisation est suivie par l'impulsion de présence générée par le capteur cela signifie que le DS18B20 est prêt pour toute commande.

Pour réaliser une initialisation le microcontrôleur place la ligne au niveau bas pour un minimum de $480\ \mu\text{s}$ avant de se mettre en état d'écoute. La ligne est remise à l'état haut par la résistance de pull-up. Après avoir détecté l'état haut le capteur attend $15\ \mu\text{s}$ à $60\ \mu\text{s}$ avant de mettre à son tour la ligne à l'état bas pour $60\ \mu\text{s}$ à $240\ \mu\text{s}$. Une représentation graphique de la séquence est donnée ci-dessous.

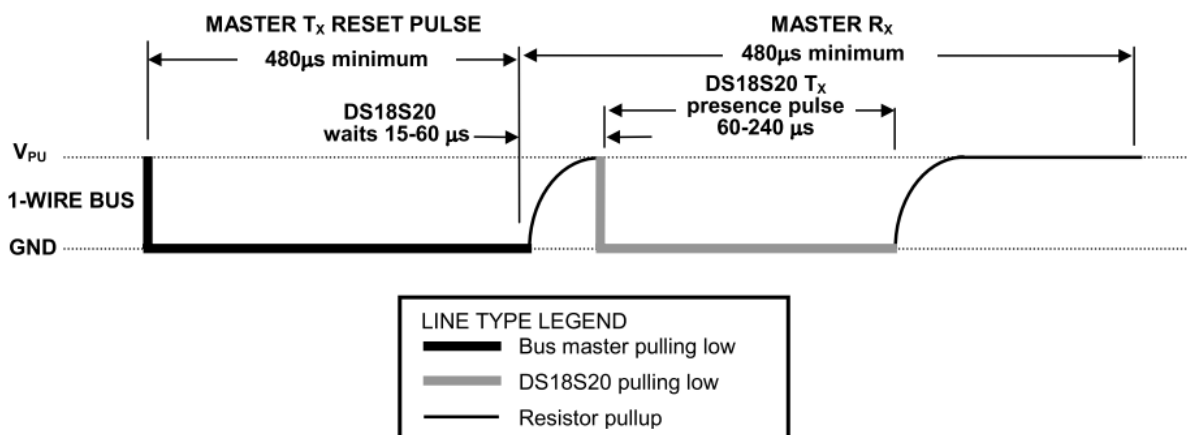


Figure 3.5 : Initialisation de DS1820



Lecture d'un bit :

Dans le même ordre d'idée la lecture d'un bit « 0 » ou « 1 » suit la séquence suivante :

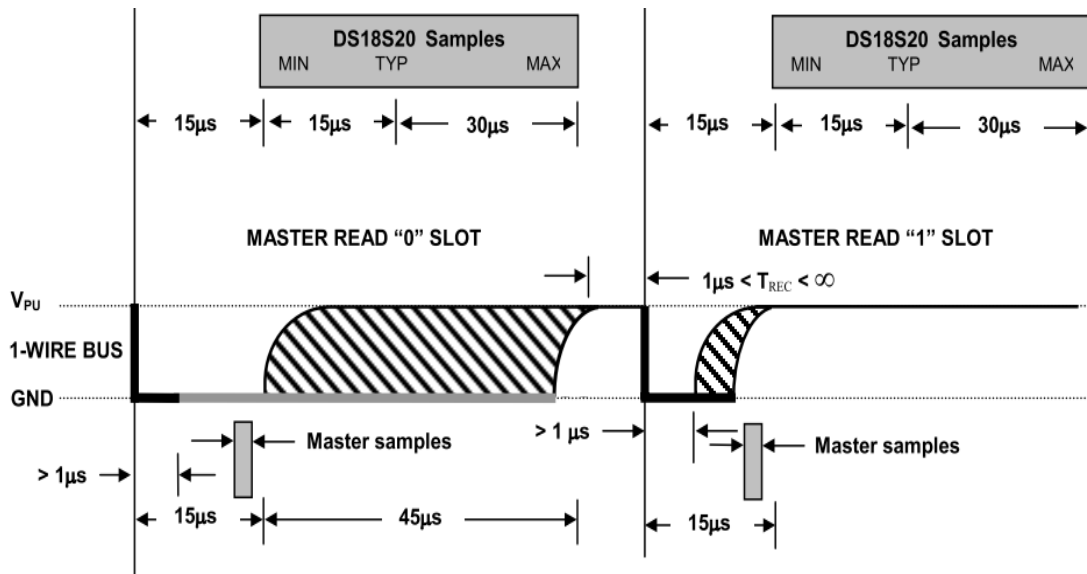


Figure 3.6 : Lecture d'un bit

Pour la lecture d'un bit l'hôte met la ligne à '0' pendant 1 us, et après 15 us la ligne est mise par le capteur à l'état haut pour signifier '1' et à l'état bas pour un '0'.



Ecriture d'un bit

La séquence pour écriture d'un bit « 0 » ou « 1 » est la suivante :

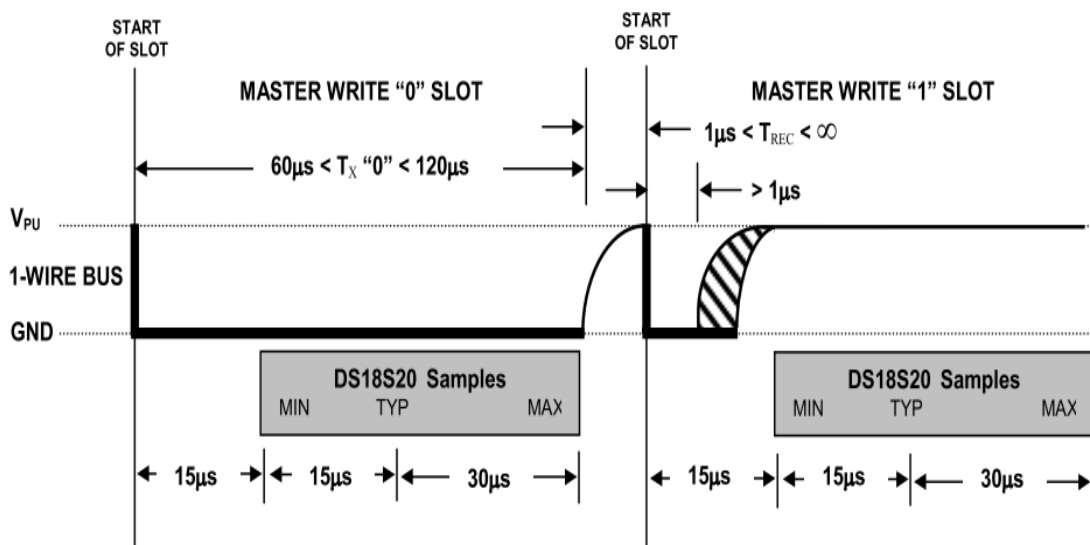


Figure 3.7 : Ecriture d'un bit

L'écriture d'un bit commence lorsque l'hôte met la ligne à '0'. Pour écrire un bit '1' la ligne doit être revenir à l'état haut avant 15 us tandis que pour écrire '0' la ligne doit être maintenue à '0' jusqu'à 60 us. chaque cycle d'écriture doit prendre au minimum 60us.

3. *Programme de mesure du température:*



Les commandes :

Le DS1820 possède un jeu commande permettant de réaliser toute les opérations nécessaire, parmi celles-ci voici celles que nous utiliserons :

- ✓ Lecture du scartchpad (BEh),
- ✓ Ecriture dans le scratchpad (4Eh),
- ✓ Lancer conversion de la température (44h),
- ✓ Passer vérification de la ROM (CCh) lorsqu'il n y a qu'un seul capteur sur la ligne la désignation n'est pas nécessaire.

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



L'organigramme :

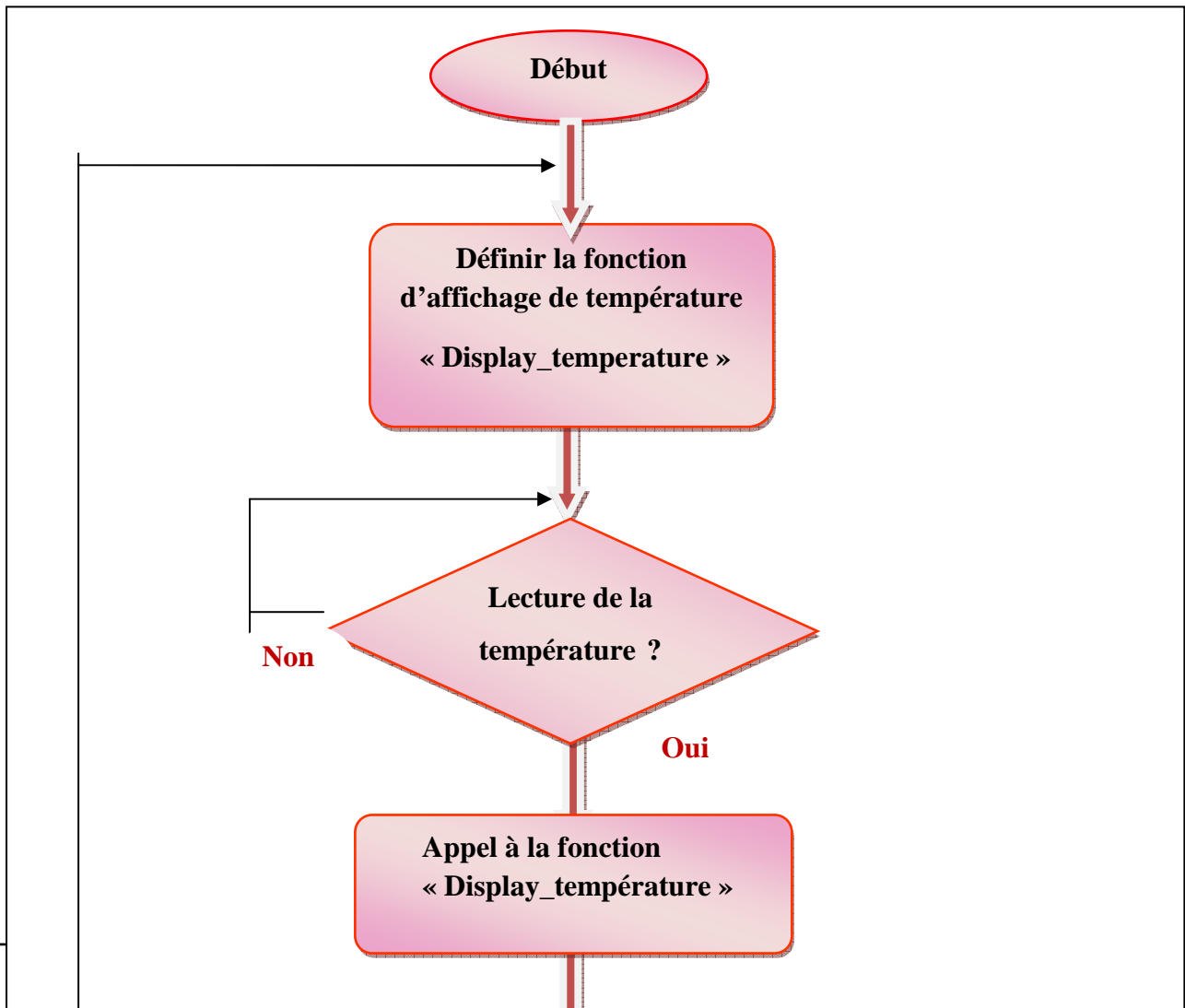




Figure 3.8 : Organigramme de calcul de température

III.

CONCEPTION ET REALISATION DU

thermomètre :

1. Branchement :

La communication One-Wire permet la transmission de données par une seule ligne de communication, ce qui signifie qu'une seule broche du microcontrôleur est utilisée. Ce capteur requiert une tension d'alimentation de 3V à 5.5V.

Le système de développement BigPic6 est équipé d'une interface de connexion destinée au DS1820. Le capteur peut utiliser les broches RE2 ou RE5 pour communiquer avec le microcontrôleur. Le choix de la broche s'effectue grâce aux interrupteurs 7 et 8 du DIP Switch SW12.

Pour le branchement de DS1820 avec le microcontrôleur on a le schéma suivant :

Capteur de température relié au microcontrôleur via RE5

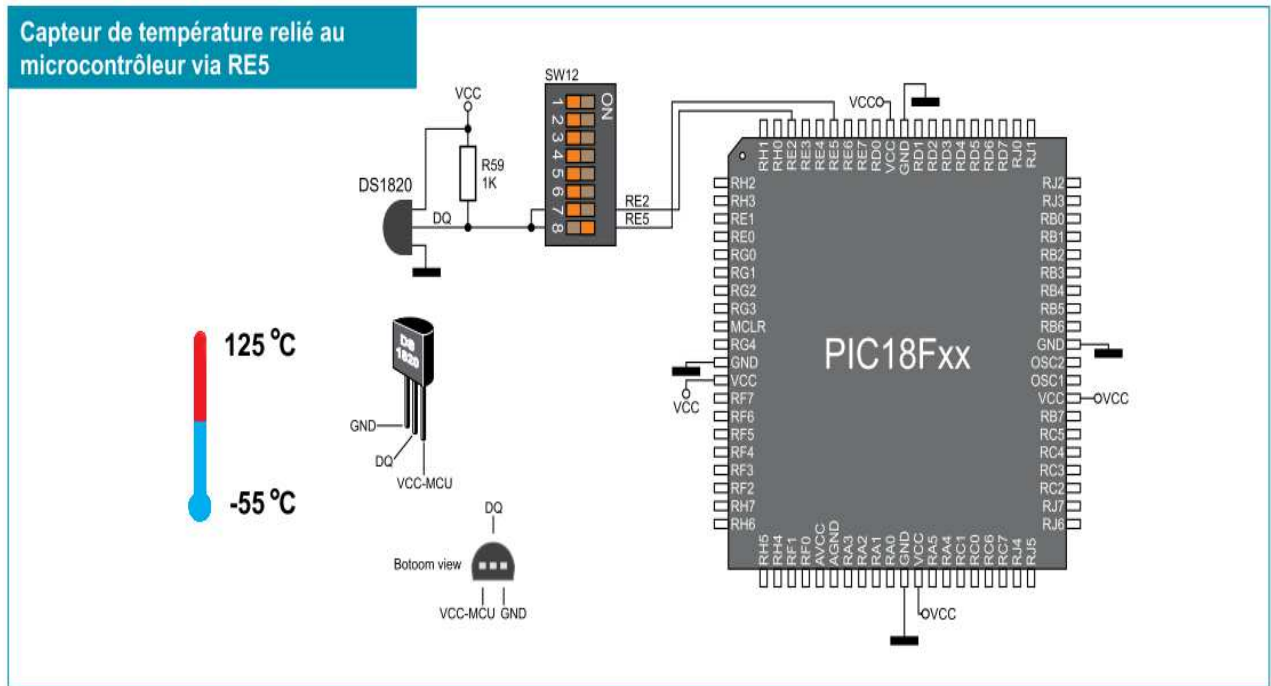


Figure 3.9 : Connexion entre DS1820 et PIC

2. Simulation sous ISIS :

Le logiciel permet de simuler le schéma du thermomètre afin de détecter certaines erreurs dès l'étape de conception.

Après création d'un programme en MikroC, on a réalisé le schéma du thermomètre sous ISIS. La première étape est de connecter la sortie du capteur au PIC. La deuxième étape est de charger le PIC par le programme.

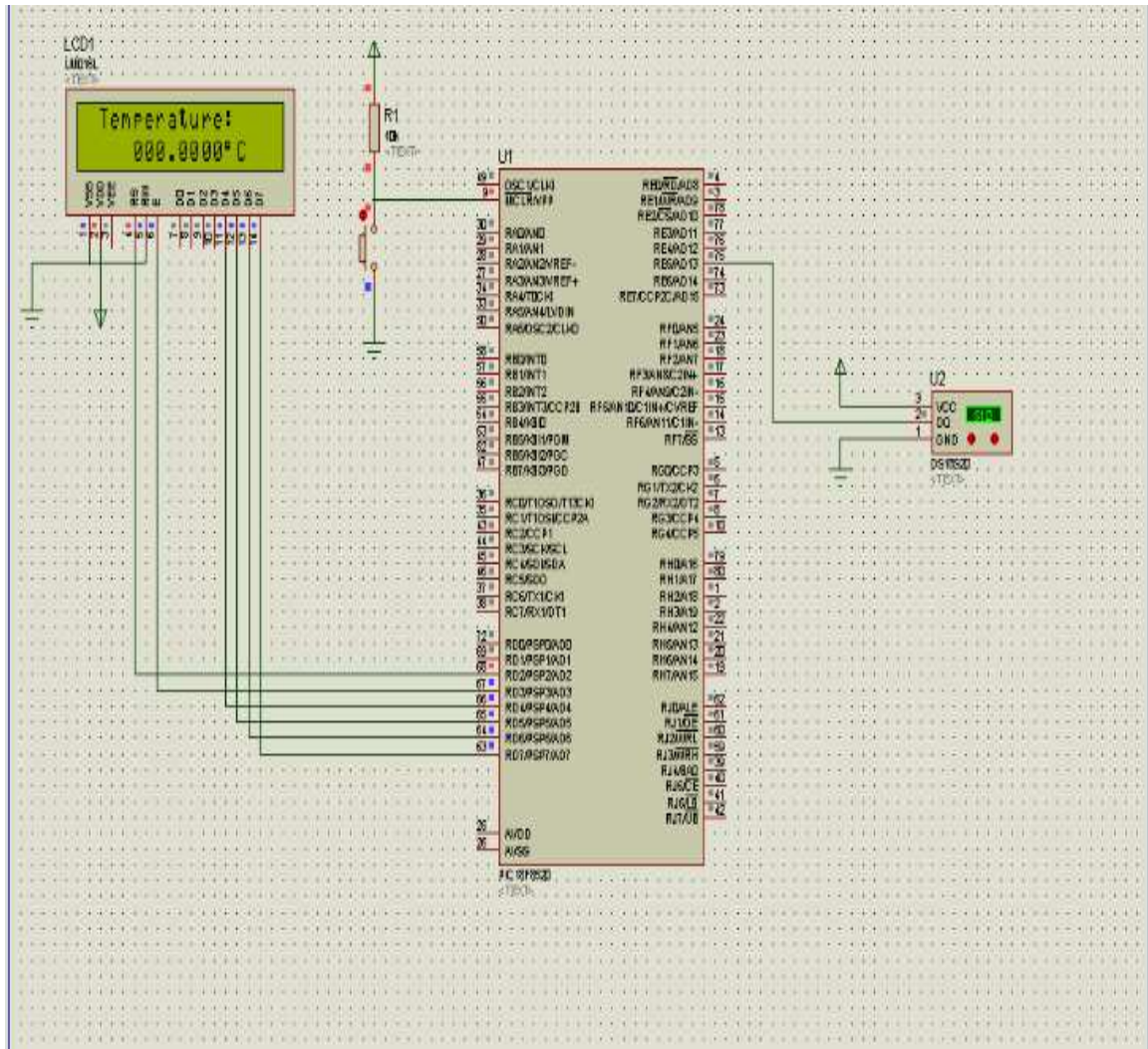


Figure 3.10 : Simulation sous ISIS d'un capteur DS1820

La simulation de thermomètre est faite avec succès, ce qui me donne l'accès à la réalisation.

3. Réalisation :

Après création d'un programme en MikroC, On utilise PICflash pour charger un fichier hexadécimal (Fichier .Hex) dans le microcontrôleur.

Enfin, on a pu réaliser le thermomètre.

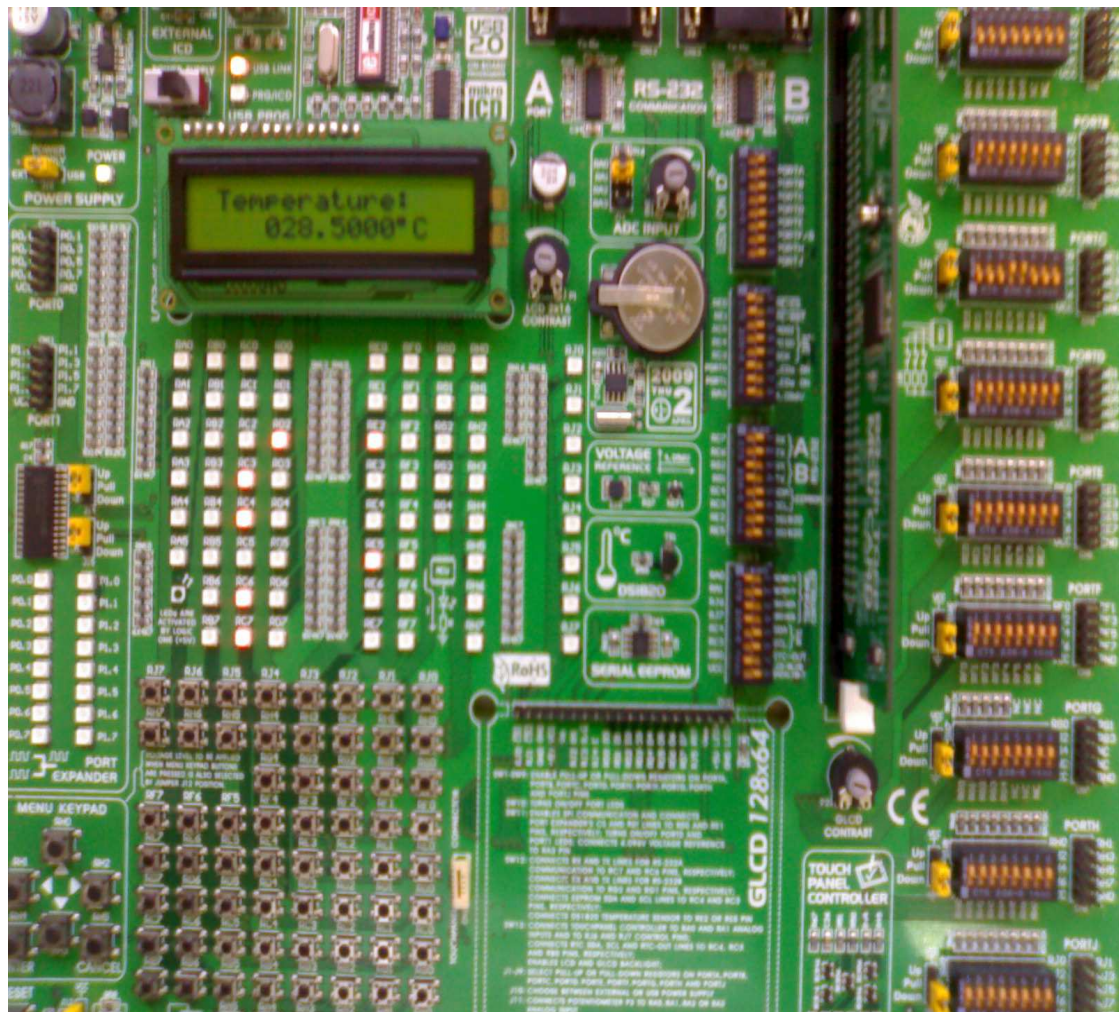


Figure 3.11 : Réalisation du thermomètre

IV. 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



Conclusion

D'après l'étude faite sur les deux types des thermomètres, en partant de l'étude de processus de fonctionnement, jusqu'à la réalisation, on peut conclure que le thermomètre à base du capteur numérique DS1820 est plus efficace que le thermomètre à base d'un capteur analogique LM35DZ, car il émet un signal purement numérique. De plus, le circuit de thermomètre à base du capteur DS1820, est plus simplifié car le DS1820 intègre le capteur, le circuit de mise en forme du signal et le CAN.

Au cours de cette étude, nous avons abordé plusieurs thèmes. C'est dans ce cadre que nous avons mis en pratique nos connaissances acquises ces dernières années, et d'autre connaissance qu'on a appris durant ce stage, mais également nous avons appris à nous adapter pour optimiser et non plus appliquer pour réaliser.

Le développement de l'architecture a consisté essentiellement au choix des éléments du système et leur étude afin de les interfacer en facilitant le développement logiciel.

Le développement logiciel que nous avons réalisé en langage C embarqué pour le microcontrôleur a été la tâche la plus difficile du travail car à coté des difficultés commune à toute tâche de programmation en langage embarqué, il y avait les nombreuses particularités du compilateur MikroC à prendre en compte.

Dorénavant on sait de quelle manière accéder à la Conception électronique assisté par ordinateur et la programmation des microcontrôleurs. Tous les programmes utilisés fonctionnent, cela augmente les chances que ce projet soit repris pour une utilisation future qui pourrait servir de base dans le développement des plusieurs systèmes qui nécessitent la mesure de température, un domaine aussi vaste que l'étendue de l'imagination.

Annexe



I. Programme de capteur DS1820:

```
////////////////////Initialisation du LCD////////////////////
```

```
// Module de connexions du LCD
```

```
sbit LCD_RS at LATD2_bit;
sbit LCD_EN at LATD3_bit;
sbit LCD_D4 at LATD4_bit;
sbit LCD_D5 at LATD5_bit;
sbit LCD_D6 at LATD6_bit;
sbit LCD_D7 at LATD7_bit;
```

```
sbit LCD_RS_Direction at TRISD2_bit;
sbit LCD_EN_Direction at TRISD3_bit;
sbit LCD_D4_Direction at TRISD4_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D7_Direction at TRISD7_bit;
```

```
// Fin du module de connexion du LCD
```

```
// Définir la résolution de DS1820
```

```
const unsigned short TEMP_RESOLUTION = 9;
```

```
char *text = "000.0000";
```

```
unsigned temp;
```

```
////////////////////La fonction de calcul de température////////////////////
```

```
void Display_Temperature(unsigned int temp2write) {
  const unsigned short RES_SHIFT = TEMP_RESOLUTION - 8;
  char temp_whole;
  unsigned int temp_fraction;
```

```
// Extraire temp_whole
```

```
temp_whole = temp2write >> RES_SHIFT ;
```

```
// Convertir temp_whole en caractère
```

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

```
text[1] = (temp_whole/10)%10 + 48;
```

```
text[2] = temp_whole%10 + 48;
```

```

// Extraire temp_fraction et la convertir en un entier
temp_fraction = temp2write << (4-RES_SHIFT);
temp_fraction &= 0x000F;
temp_fraction *= 625;

// Convertir temp_fraction en caractère
text[4] = temp_fraction/1000 + 48;
text[5] = (temp_fraction/100)%10 + 48;
text[6] = (temp_fraction/10)%10 + 48;
text[7] = temp_fraction%10 + 48;

// Afficher la température en LCD
Lcd_Out(2, 5, text);
}

void main() {

    ADCON1 |= 0x0F;           // Configurer AN pins comme une entrée numérique
    CMCON |= 7;              // désactiver le comparateur

    Lcd_Init();              // Initialiser LCD
    Lcd_Cmd(_LCD_CLEAR);    // effacer LCD
    Lcd_Cmd(_LCD_CURSOR_OFF); // Tourner cursor off
    Lcd_Out(1, 1, " Temperature: ");

    // Afficher degree en 'C'
    Lcd_Chr(2,13,223);

    Lcd_Chr(2,14,'C');

    do {

        Ow_Reset(&PORTE, 5);           // Onewire reset signal
        Ow_Write(&PORTE, 5, 0xCC);     // Issue command SKIP_ROM
        Ow_Write(&PORTE, 5, 0x44);     // Issue command CONVERT_T
        Delay_us(120);

        Ow_Reset(&PORTE, 5);
        Ow_Write(&PORTE, 5, 0xCC);     // Issue command SKIP_ROM
        Ow_Write(&PORTE, 5, 0xBE);     // Issue command
        READ_SCRATCHPAD
        Delay_ms(400);
    } while(1);
}

```

```

temp = Ow_Read(&PORTE, 5);
temp = (Ow_Read(&PORTE, 5) << 8) + temp;

//l'appel de la fonction de calcul de température
Display_Temperature(temp);

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

```

II. Programme de capteur

LM35DZ:

```

////////////////////Initialisation du LCD////////////////////

// Module de connexion du LCD
sbit LCD_RS at RC4_bit;
sbit LCD_EN at RC5_bit;
sbit LCD_D4 at RC0_bit;
sbit LCD_D5 at RC1_bit;
sbit LCD_D6 at RC2_bit;
sbit LCD_D7 at RC3_bit;

sbit LCD_RS_Direction at TRISC4_bit;
sbit LCD_EN_Direction at TRISC5_bit;
sbit LCD_D4_Direction at TRISC0_bit;
sbit LCD_D5_Direction at TRISC1_bit;
sbit LCD_D6_Direction at TRISC2_bit;
sbit LCD_D7_Direction at TRISC3_bit;

// Définir le message
char message1[] = "Temperature";

char *tempC = "000.0";

// Variables pour stocker la valeur de la température
unsigned int tempinC;
unsigned long temp_value;
void Display_Temperature()

```

```

{

// Convertir Temp en caractère
if (tempinC/10000)

// 48 est le code de caractère en décimal pour l'afficher en LCD
tempC[0] = tempinC/10000 + 48;
else tempC[0] = ' ';
tempC[1] = (tempinC/1000)%10 + 48;
tempC[2] = (tempinC/100)%10 + 48;

// Convertir temp_fraction en caractère
tempC[4] = (tempinC/10)%10 + 48;

// Afficher la température en LCD
Lcd_Out(2, 1, tempC);

}
void main()
{

    TRISA2_bit = 1;

    ADCON0 = 0b01001000;
    CMCON |= 0x07;
    TRISC = 0b00000000;
    TRISA = 0b00001110;
    Lcd_Init(); // Initialize LCD
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Cmd(_LCD_CURSOR_OFF);
    Lcd_Out(1,1,message1);
    Lcd_Chr(2,6,223);

    Lcd_Chr(2,7,'C');

    do
    {
        temp_value = ADC_Read(2);
        temp_value = temp_value*1168;
        tempinC = temp_value/1000;
        tempinC = tempinC*10;
        Display_Temperature();
        Delay_ms(1000);
    }
}

```

```

} while(1);
}

```

III. Prise en main de la carte

BIGPIC6 :

1. Architecture interne:

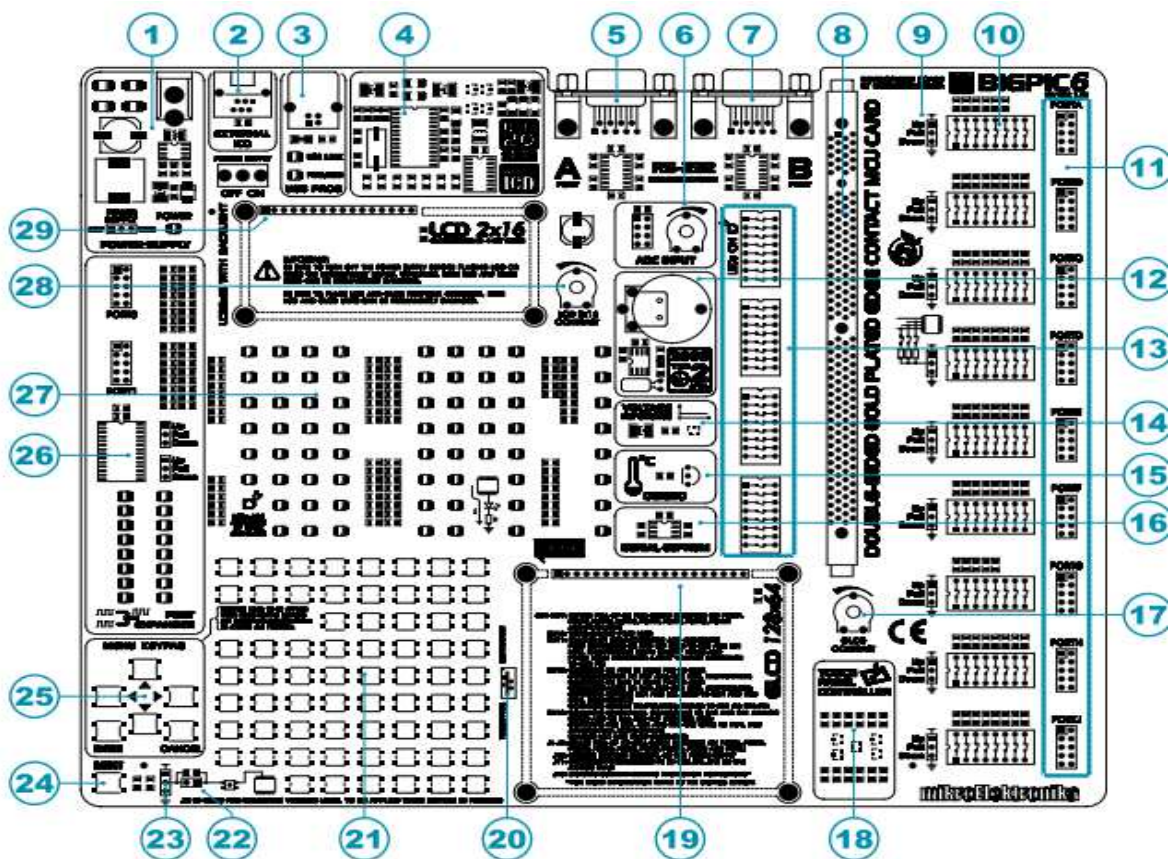


Figure 4.1 : Architecture interne de BigPic6

1. Alimentation régulé en tension.
2. Connecteur pour dégorgeur Microchip (ICD2 ou ICD3).
3. Connecteur USB du programmeur embarqué.
4. Programmeur USB 2.0 avec mikroICD.
5. Connecteur A pour communication RS-232.
6. Entrés de tests du convertisseur A/N.
7. Connecteur B pour communication RS-232
8. Connecteur DIMM-168P pour carte MCU.
9. Sélection des résistances de tirage.

10. DIP switches pour activer les résistances de tirage.
11. Connecteur des ports E/S.
12. Horloge temps réelles(RTC).
13. DIP switches d'activation/désactivation des modules intègres.
14. Voltage de référence 4.096V.
15. Capteur de température DS1820.
16. EEPROM série.
17. Ajusteur de contraste de l'afficheur LCD graphique.
18. Contrôleur du panneau tactile.
19. Connecteur de l'afficheur LCD graphique.
20. Connecteur de panneau tactile.
21. Boutons poussoirs de simulation des entrées numériques.
22. Résistance de protection ON/OFF.
23. Sélecteur d'état logique des broches.
24. Bouton Reset.
25. Clavier numérique MENU.
26. Extenseur de port.
27. 67 LEDs d'indication de l'état logique des broches.
28. Ajuster de contraste de l'afficheur LCD alphanumérique.
29. Connecteur de l'afficheur LCD alphanumérique.

2. Connexion au PC :

● Etape 1 :

On commence tous d'abord par l'installation des pilotes USB et du programmeur PICflash fournis dans les manuels associés. Il n'est pas possible de programmer les microcontrôleurs PIC sans avoir préalablement installé ces pilotes.

● Etape 2 :

On utilise le câble USB fournit pour connecter le système de développement EasyPIC6 au PC. Une extrémité au câble USB devra être connectée à la carte de développement comme décrit figure 1-12, tandis que l'autre extrémité devra être connectée au PC. Avant d'établir la connexion, assurez vous que le jumper J10 est bien placé sur la position USB comme indiqué la figure 1-13.

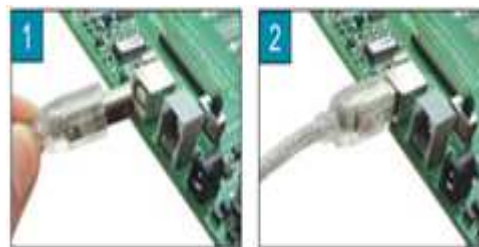


Figure 4.2 : Connexion du câble USB

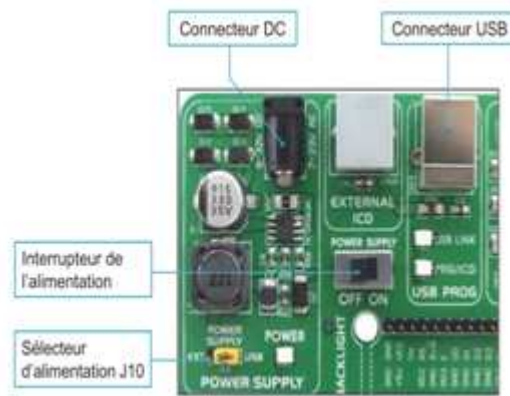


Figure 4.3 : Alimentation électrique

● **Etape 3 :**

On démarre le système de développement en poussant l'interrupteur d'alimentation sur la position ON. Les LEDs 'POWER' et 'USB LINK' vont s'allumer, signifiant que le système de développement est prêt à être utilisé. On utilise le programmeur PICflash pour charger le code dans le microcontrôleur et on se sert de la carte pour tester et développer le projet.

3. *Microcontrôleurs supportés :*

Le système de développement BigPIC6 est équipé d'un connecteur DIMM-168P utilisé pour brancher une carte MCU. Le MCU intégrant un microcontrôleur 80 broches, comme décrit figure 1-14. Cette carte possède également un oscillateur ainsi qu'un panneau de connexions de 80 broches relié au microcontrôleur. Le nom de la broche du microcontrôleur est associé à chaque point de connexion. Ces points de connexions facilitent l'utilisation de la carte lorsqu'elle est ensuite utilisée dans un dispositif cible.



Figure 4.4 : Carte MCU avec microcontrôleur-80 broches

4. *La programmation du microcontrôleur :*

a. *Le compilateur MikroC :*

Le compilateur mikroC pour PIC sauvegarde les applications au sein de projets qui s'apparentent à un fichier 'projet' unique (avec l'extension .ppc) ainsi qu'à un ou plusieurs fichiers sources (avec l'extension .c). L'environnement du compilateur mikroC pour PIC ne permet la gestion que d'un seul projet à la fois. Les fichiers sources ne peuvent être compilés que s'ils font partis d'un projet.

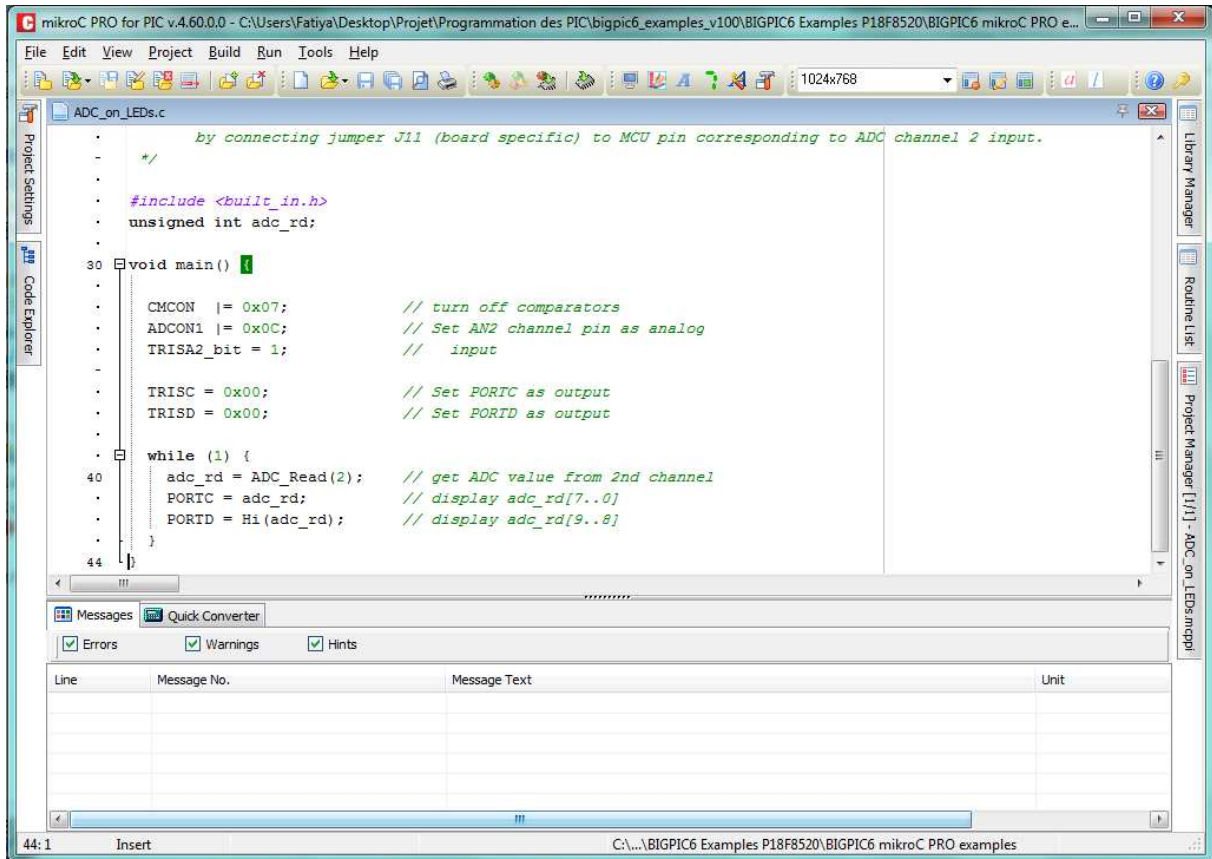


Figure 4.5 : Logiciel MikroC

b. Programmeur embarqué PICflash USB 2.0 :

Le programmeur PICflash avec assistance mikroICD est un outil haute performance destiné à la programmation des familles de microcontrôleurs PIC 16F et PIC 18F de MIKROCHIP. Sa conception particulière et sa facilité d'utilisation en font un outil très populaire aussi bien chez les débutants que chez les professionnels. Le programmeur PICflash communique avec le microcontrôleur via un câble USB servant également à alimenter le programmeur. De plus, c'est un dispositif à faible consommation, ce qui le rend idéal pour fonctionner avec des PC portables.

Le programmeur PICflash possède une option pour sélectionner le microcontrôleur à programmer. Il est un outil indispensable à la programmation des microcontrôleurs. La carte a un programmeur PICflash embarqué doté de la technologie mikroICD. On utilise PICflash pour charger un fichier hexadécimal (Fichier Hex) dans le microcontrôleur. La

figure 1-18 présente les liens entre le compilateur, le programmeur PICflash et le microcontrôleur.

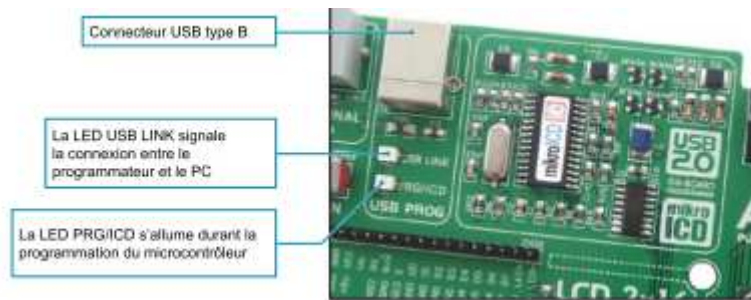


Figure 4.6 : Programmateur PICflash

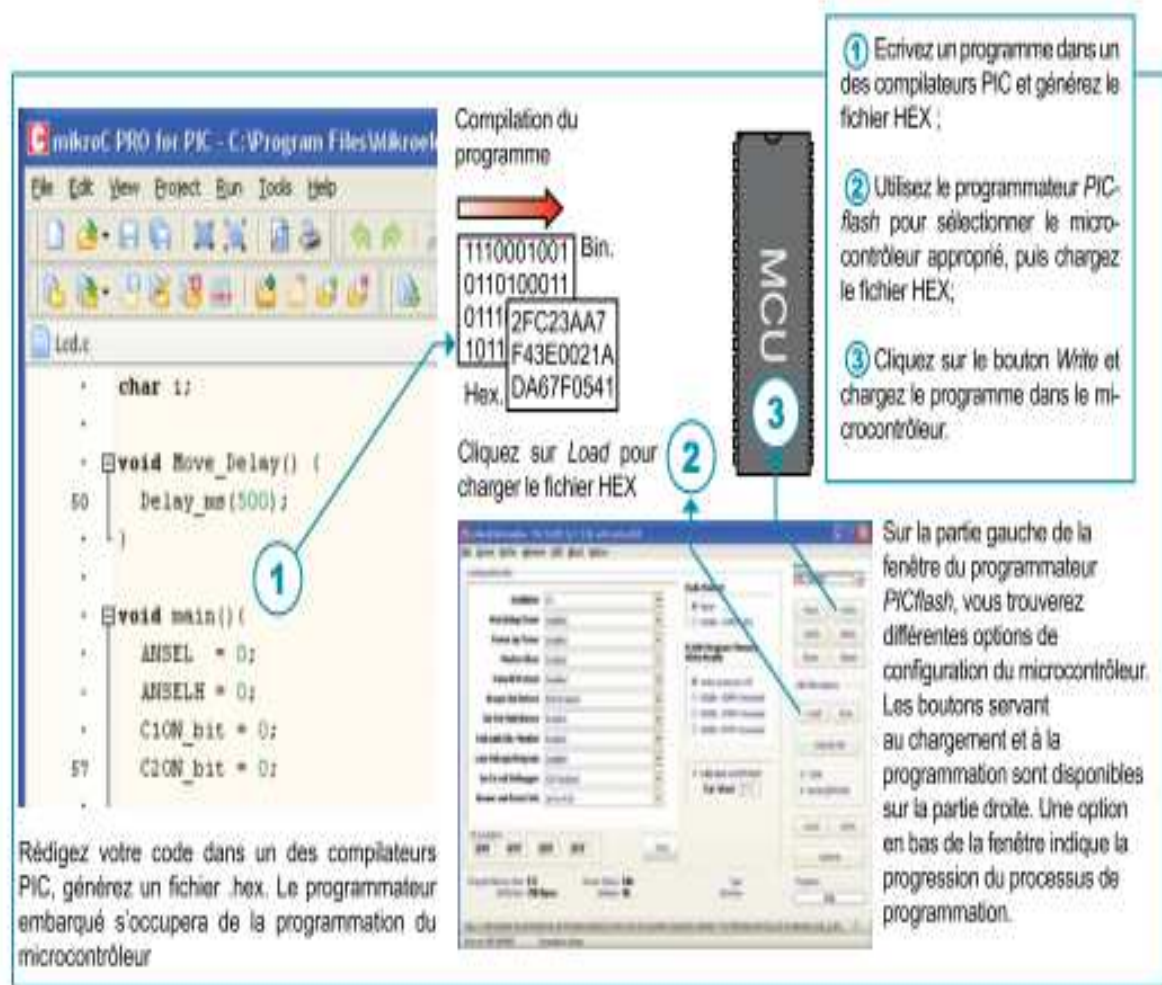


Figure 4.7 : Principe de programmeur

Toutes les actions en rapport avec la programmation des microcontrôleurs sont effectuées automatiquement. Cependant, de nombreuses options de configuration de la programmation sont disponibles dans le programme PICflash.

c. MikroICD :

Le mikroICD fait partie intégrante du programmeur embarqué. Son rôle est de tester et de déboguer les programmes en temps réel. Le processus de test et débogage s'effectue en affichant l'état des registres et variables du microcontrôleur au cours de l'exécution du programme. Le MikroICD est intégré dans tous les compilateurs PIC mikroElektronika (mikroC, mikroPASCAL...)

Le débogueur mikroICD communique avec le microcontrôleur par les broches de programmation. Par conséquent, ces dernières ne peuvent pas être utilisées comme broches E/S lors du débogage.

Le débogueur mikroICD offre également la possibilité d'exécuter le programme pas à pas. Stopper l'exécution du programme dans le but d'examiner l'état des registres en utilisant des points d'arrêts.

5. Afficheur LCD 2x16 :

La carte de développement BigPIC6 est équipée d'un connecteur pour afficheur LCD alphanumérique 2x16. Ce connecteur est relié au microcontrôleur via PORTD. Le potentiomètre P1 est utilisé pour ajustement du contraste. L'interrupteur LCS-GLCD BLACKLIGHT du DIP Switch SW13 sert à allumer ou éteindre le rétro éclairage.

La communication entre le microcontrôleur et l'afficheur LCD s'effectue en mode 4 bits. Les symboles alphanumériques sont représentés sur deux lignes, chacune pouvant contenir jusqu'à 16 caractères de tailles 7x5 pixels.



Figure 4.8 : Connecteur de l'afficheur LCD 2x16

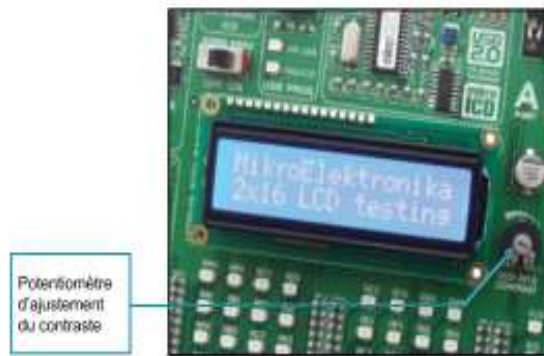


Figure 4.9 : Afficheur LCD 2x16

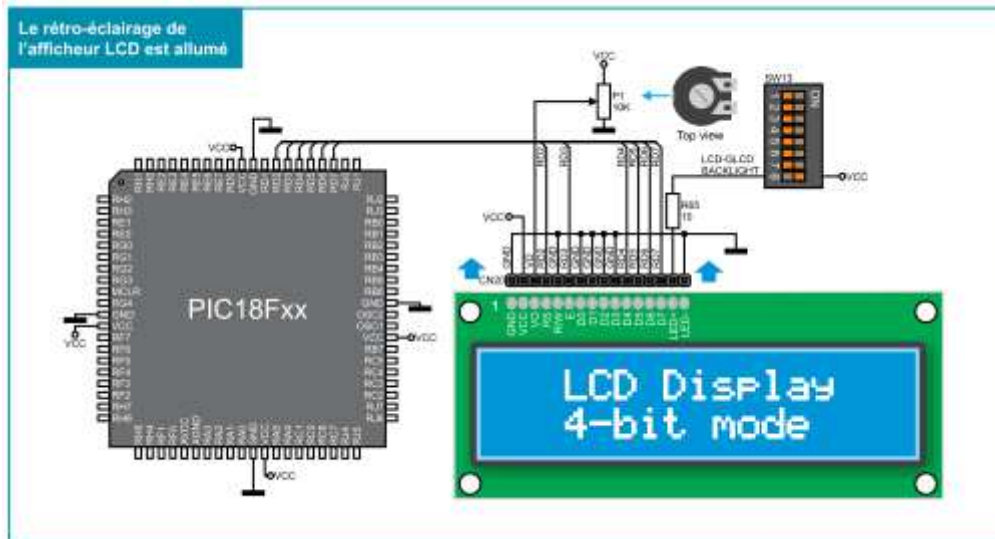


Figure 4.10 : connexion entre l'afficheur LCD et PIC

Bibliographie & Webographie

Ouvrages

Datasheet PIC18F8520, Microchip

Datasheet LM35DZ, Microchip

Datasheet Dallas DS1820

Mode d'emploi PICflash, Microchip

Mode d'emploi BIGPIC6, Microchip

Sites internet

http://clubelek.insa-lyon.fr/joomla/fr/base_de_connaissances/electronique/initiation_au_microcontroleur.php

http://fr.wikipedia.org/wiki/Microcontr%C3%B4leur_PIC

<http://fr.farnell.com/microchip/pic18f8520-i-pt/microcontroleur-8-bits-flash-cms/dp/9762353>