

N° Ordre...../Faculté/UMBB/2017

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE  
UNIVERSITE M'HAMED BOUGARA-BOUMERDES



Faculté des Hydrocarbures et de la Chimie



Département automatisation des Procèdes  
Industriels et Electrification

Mémoire de fin d'étude  
En vue de l'obtention du diplôme

**MASTER**

Présenté par  
**SIDI MAMMAR Meriem**  
**OUZERROUT Thanina**

Filière : Hydrocarbure  
Spécialité : Automatisation des procédés industriels : Commande Automatique

**Thème**

**Etude et réalisation d'une régulation de type hybride  
avec ordinateur et microcontrôleur.**

**Application au réglage de température.**

**Membres de jury :**

Dr H. HABBI H.

Dr M. HAMADACHE

Dr A. CHAIB.

Dr H. FEKHAR.

Président

Examineur

Examineur

Promoteur

Année Universitaire : 2016/2017

# *Remerciements*

*Nous tenons à remercier notre encadreur Monsieur H.FEKHAR, pour le suivi de notre travail et pour l'encouragement constant qu'il n'a cessé de prodiguer tout au long de cette étude.*

*Nous remercions tous les enseignants du département automatisation et électrification des procédés industriels, qui ont assuré notre formation durant tout le cycle d'étude.*

*Nous apportons également nos sincères remerciements à l'ensemble des membres de notre club scientifique « AEPI NEW VISION » avec lesquels nous avons passé de formidables années.*

*Nous tenons à présenter nos vifs remerciements à nos familles et nos proches pour leur soutien et leurs encouragements.*

*Enfin un grand merci pour tous ceux qui ont collaboré de près ou de loin à la réalisation de ce modeste travail.*

*OUZERROUT Thanina  
SIDI MAMMAR Meriem*

---

*Dédicace*

*Je dédie ce travail à mes chers parents,*

*A mon petit frère Chabane,*

*A ma sœur adorée Kenza,*

*A mon binôme MERIEM.*

---

*Dédicace*

*Je dédie ce travail à mes parents et à mes grands parents*

*A mes frères Billal et Lounis,*

*A ma sœur Malak ,*

*A mon binôme NINA et à tous mes amis.*

# Sommaire

Liste des figures.....	I
Liste des tableaux.....	III
Introduction générale.....	IV

## Chapitre I

### Représentation et description du système

<b>INTRODUCTION.....</b>	<b>2</b>
I.1 Présentation et description du four électrique avec son circuit de puissance.....	2
I.1.1 Présentation du four de laboratoire.....	3
I.1.2 Constituants du four.....	3
I.1.3 Le circuit de puissance du four électrique.....	4
I.2 Présentation générale du capteur de température .....	7
I.2.1 Définition des capteurs de température.....	7
I.2.2 Classification des capteurs de température.....	7
I.2.3 Capteur de température à semi-conducteur AD590.....	8
I.3 Le circuit de conditionnement du capteur.....	10
I.4 Le filtre passe-bas.....	11
I.4.2 Le filtre passe-bas « Sallen & Key ».....	11
<b>CONCLUSION.....</b>	<b>14</b>

## Chapitre II

### Identification de la dynamique du four électrique

<b>INTRODUCTION.....</b>	<b>16</b>
II.1 Acquisition des données E/S.....	17
II.2 Détermination des caractéristique du four électrique.....	19
II.2.1 Test d'environnement (déterministe/stochastique).....	19
II.2.2 Mono variable /multi variable .....	19
II.2.3 Test de stationnarité (invariant / variable dans le temps).....	20
II.2.4 Test de linéarité (linéaire / non linéaire).....	20
II.3 Estimation des paramètres.....	20
II.3.1 La méthode de STREJC.....	20

II.4	Validation du modèle identifié.....	22
II.4.1	Validation du modèle du four .....	23
II.4.2	Validation du modèle du filtre.....	21
<b>CONCLUSION.....</b>		<b>24</b>

## **Chapitre III**

### **Etude et réalisation de la carte de communication**

<b>INTRODUCTION.....</b>		<b>26</b>
III.1	Structure générale de la carte d'interface.....	26
III.1.1	Unité d'adaptation.....	27
III.1.1.1	Circuit MAX238.....	28
III.1.2	Unité d'entrée.....	29
III.1.2.1	Circuit d'adaptation de la tension issue du capteur.....	29
III.1.2.2	Le Microcontrôleur 18F458.....	30
III.1.2.3	L'afficheur LCD .....	37
III.1.2.4	Circuit de conversion parallèle série.....	39
III.1.3	Unité de sortie.....	40
III.1.3.1	Circuit de conversion série/parallèle .....	40
III.1.3.2	Convertisseur numérique analogique DAC 0800 .....	42
III.1.3.3	Le circuit LM741.....	43
III.2	Le schéma général de la carte d'interface.....	43
III.3	Présentation du port de communication du PC .....	45
<b>CONCLUSION.....</b>		<b>47</b>

## **Chapitre IV**

### **Implémentation du programme de commande**

<b>INTRODUCTION.....</b>		<b>49</b>
IV.1	La commande en boucle fermée.....	49
IV.1.1	Intérêt de la régulation.....	50
IV.1.2	Principe de la régulation.....	50
IV.1.3	Le choix de la forme de la fonction de transfert du régulateur.....	52
IV.1.3.1	Régulateur à action proportionnelle (P).....	52
IV.1.3.2	Régulateur à action proportionnelle et intégrale (PI).....	53

IV.2	Présentation du software.....	54
IV.2.1	Le software d'acquisition et l'affichage des données.....	54
IV.2.1.1	Présentation du MikroC.....	54
IV.2.1.2	L'affichage sur LCD.....	55
IV.2.1.3	La conversion analogique _ digitale (ADC).....	56
IV.2.1.4	Le logiciel de simulation ISIS PROTEUS.....	57
IV.2.1.5	Le logiciel PICKit 2.....	58
IV.2.2	Programmation et réalisation des interfaces graphiques.....	59
IV.2.2.1	Présentation du compilateur Visual C #2010.....	59
IV.2.2.2	Les interfaces graphiques.....	60
IV.2.2.3	L'organigramme simplifié de software sous visual C#.....	61
IV.3	Evaluation des performances du système.....	63
IV.3.1	Evaluation de la commande.....	63
IV.3.1.1	Commande par un régulateur P non linéaire.....	64
IV.3.1.2	Commande par un régulateur PI non linéaire.....	65
IV.3.2	Réglages des paramètres du régulateur.....	66
IV.3.2.1	Méthodes empiriques de Ziegler& Nichols.....	67
<b>CONCLUSION.....</b>		<b>70</b>
<b>CONCLUSION GENERALE.....</b>		<b>71</b>

## Liste des figures

Figure I.1	le four disponible au niveau du laboratoire.....	2
Figure I.2	Représentation schématique d'un four à résistance.....	2
Figure I.3	Le différents constituants du four du laboratoire.....	4
Figure I.4	Schéma de circuit de puissance du four électrique .....	4
Figure I.5	Schéma simplifié de circuit de puissance du four.....	5
Figure I.6	Différents régimes de fonctionnement du Triac.....	5
Figure I.7	Représentation schématique d'un triac.....	6
Figure I.8	Une photo réelle du capteur.....	8
Figure I.9	La structure interne du capteur de température à semi-conducteur AD590.....	9
Figure I.10	La Caractéristique du courant en fonction de l'alimentation du capteur AD590.....	9
Figure I.11	L'emplacement des trois capteurs.....	10
Figure I.12	Schéma électronique du circuit de conditionnement du capteur AD590.....	10
Figure I.13	Schéma d'un filtre passe-bas « Sallen & Key » .....	12
Figure I.14	Le schéma du filtre passe-bas « Sallen & Key » utilisé dans notre montage .....	12
Figure I.15	Montage réel de filtre passe-bas sur une plaque d'essai.....	12
Figure I.16	Réponse indicielle réelle du filtre relevée par l'oscilloscope.....	14
Figure II.1	La procédure d'identification.....	17
Figure II.2	Schéma synoptique ayant servi à l'acquisition de la réponse temporelle du four.....	17
Figure II.3	Réponse du four relevée par l'oscilloscope après filtrage.....	18
Figure II.4	Réponse réelle du four tracée sous MATLAB.....	19
Figure II.5	la courbe identifiée simulée sous MATLAB.....	22
Figure II.6	Courbes de réponse de système réelle et celle du modèle identifié.....	23
Figure III.1	La répartition principale du système.....	26
Figure III.2	Diagramme bloc général de la carte d'interface.....	27
Figure III.3	Structure interne du MAX238.....	28
Figure III.4	Branchement du MAX238 avec RS232 et 7404DM.....	28
Figure III.5	Schéma de branchement circuit d'adaptation tension du Capteur.....	29
Figure III.6	Les différentes broches du PIC 18F458.....	31
Figure III.7	Brochage de l'oscillateur à quartz.....	32
Figure III.8	Architecture interne du PIC18F458.....	34
Figure III.9	L'organisation de la mémoire RAM.....	35
Figure III.10	Afficheur LCD 4*16 caractères.....	37



Figure III.11	Schéma de brochage de l'afficheur LCD.....	38
Figure III.12	Schéma de branchement du CMOS4021-ADC.....	39
Figure III.13	Schéma bloc de fonctionnement du CMOS4094.....	40
Figure III.14	Schéma de brochage du CMOS4094.....	41
Figure III.15	Architecture interne DAC0800 (Doc. Motorola).....	43
Figure III.16	Schéma de brochage du DAC0800 avec LM741.....	43
Figure III.17	Le schéma général de la carte d'interface.....	44
Figure III.18	la carte d'interface réalisée sur une plaque d'essai.....	45
Figure III.19	Connecteur DB9 Mâle et femelle.....	46
Figure IV.1	Les chaines d'un système de régulation automatique .....	49
Figure IV.2	Schéma bloc d'un système de régulation .....	50
Figure IV.3	Schéma bloc de notre système de régulation numérique.....	51
Figure IV.4	Photo réelle de système de régulation en boucle fermée.....	51
Figure IV.5	Structure d'un régulateur proportionnel non linéaire.....	52
Figure IV.6	Structure d'un régulateur proportionnel et intégrale non linéaire.....	53
Figure IV.7	L'interface de travail du logiciel MikroC.....	55
Figure IV.8	L'organigramme de programme d'acquisition et d'affichage.....	57
Figure IV.9	L'interface de travail de PROTEUS ISIS.....	58
Figure IV.10	L'interface de travail du logiciel PICKit2.....	59
Figure IV.11	L'interface graphique (E/S) contrôle de température avec un régulateur P non linéaire.....	60
Figure IV.12	L'interface graphique (E/S) contrôle de température avec un régulateur PI non linéaire.....	61
Figure IV.13	L'organigramme simplifié de software sous visual C#.....	62
Figure IV.14	Courbe de réponse du four et le signal de contrôleur P non linéaire.....	64
Figure IV.15	Courbe de réponse du four et le signal de contrôleur PI non linéaire.....	66
Figure IV.16	Schéma explicatif de l'exploitation de la réponse indicielle.....	67
Figure IV.17	Courbe de réponse du four et le signal de contrôleur PI non linéaire à paramètres déterminés par la synthèse.....	69

## Liste des tableaux

Tableau II.1	Données de la méthode de STRETJC.....	21
Tableau II .2	Tableau de l'erreur relative calculé pour le modèle du four électrique.....	23
Tableau III.1	les caractéristiques du PIC 18F458.....	33
Tableau III.2	Bits de configuration du registre ADCON0.....	36
Tableau III.3	Sélection des canaux.....	36
Tableau III.4	Configuration de la fréquence de conversion.....	37
Tableau III.5	Bits de configuration du registre ADCON1.....	37
Tableau III.6	Répartition des différents signaux du connecteur DB9.....	46
Tableau III.7	Répartition des différents signaux utilisés pour la carte d'interface.....	47
Tableau VI.1	Le résumé des instructions les plus importantes du Mikro C.....	56
Tableau IV.2	Calcul des paramètres par la méthode de Ziegler & Nichols.....	68
Tableau IV.3	Résultats de la synthèse.....	68

## Introduction générale

Dans le cadre de fonctionnement d'une unité de production industrielle, les performances sont jugées selon des critères de qualité du produit, de sécurité, de sûreté de fonctionnement, et de respect de l'environnement (qualité des rejets, . . .). L'automatique apporte des solutions pratiques pour remplir au mieux l'ensemble de ces critères (satisfaction du cahier des charges). La mise en œuvre de ces solutions consiste à conduire – ou piloter – le procédé.

La conduite des procédés recouvre des activités telles que la planification, l'ordonnancement, la surveillance, la supervision et la commande. Le principe général de la commande des procédés est connu sous le terme de commande en boucle fermée (feedback control).

La régulation regroupe l'ensemble des moyens matériels et techniques mis en œuvre pour maintenir la grandeur réglée à une valeur égale à celle de la consigne malgré les perturbations indésirables.

Le four électrique à résistance est le plus connu des appareils électrothermiques. Son utilisation industrielle remonte en effet au début des années vingt et sa technique n'a cessé de se perfectionner depuis. Leur principe est extrêmement simple, ce qui facilite l'implémentation d'une stratégie efficace pour le contrôle de sa température. Néanmoins, l'évolution à long terme des systèmes due à leur usure et leur vieillissement fait que les différents régulateurs doivent être périodiquement réglés. Ce qui nécessite de rassembler les connaissances du comportement dynamique du four, par une analyse physique des phénomènes mis en jeu, et une analyse des données expérimentales utilisant les différentes techniques de traitement de l'information.

Ce mémoire traite une commande numérique de type hybride en utilisant le micro-ordinateur et le microcontrôleur pour le contrôle de la température du four électrique disponible au niveau de laboratoire.

Le premier chapitre sera consacré en premier lieu à la description et la présentation du four électrique avec son circuit de puissance, en deuxième lieu au capteur AD590 et son circuit de conditionnement et enfin la présentation du filtre passe-bas utilisé pour l'élimination de hautes fréquences.

Nous abordons dans le deuxième chapitre l'identification du four électrique en exploitant des données expérimentales ainsi que la validation du modèle identifié.

Dans le troisième chapitre, nous intéressons essentiellement à développer une carte électronique d'interface pour la communication entre le procédé et le micro-ordinateur. Ainsi que la présentation de chaque composant individuellement.

Dans le dernier chapitre nous effectuons une régulation numérique afin d'améliorer les performances de notre procédé en utilisant deux types de régulateurs, proportionnel et proportionnel intégral, introduits dans un programme software de contrôle. Ainsi l'élaboration des interfaces graphiques, les graphes résultants et les interprétations des résultats.

# *Chapitre I*

*Présentation et description du  
système*

## INTRODUCTION :

Les fours électriques à résistances sont d'un emploi universel dans l'industrie. Ils sont les plus connus des appareils électrothermiques, leur utilisation industrielle remonte en effet au début des années vingt et leur technique n'a cessé de se perfectionner depuis.

Ces fours électriques à résistance sont utilisés dans l'industrie sidérurgique, raffinage et de l'industrie chimique, de l'industrie cimenterie, de l'industrie verrière et des industries céramiques et des produits réfractaires.

Dans ce chapitre, on s'intéresse à la description du four électrique à résistance disponible au niveau de laboratoire, et la présentation de son circuit de puissance ainsi que le capteur utilisé.

### I.1 Présentation et description du four avec son circuit de puissance

#### I.1.1 Présentation du four de laboratoire [1]

C'est un four électrique industriel dont la chaleur est produite par une résistance chauffante commandée en tension par un amplificateur de puissance. La figure (I.1) représente le four disponible au niveau du laboratoire.



Figure I.1 Le four disponible au niveau du laboratoire

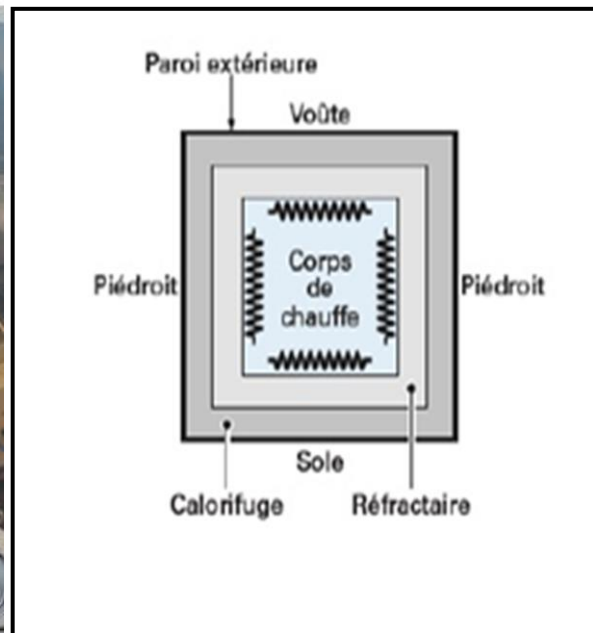


Figure I.2 Représentation schématique d'un four à résistance

Ce four électrique fonctionne avec une alimentation alternative de 220V~, il a deux modes de fonctionnement : le mode **manuel** et le mode **automatique**.

En mode manuel, la consigne de la température est introduite par le clavier alors que dans le mode automatique, le signal de commande est une tension continue comprise entre 0 et 25V.

### **I.1.2 Constituants du four [1][2]**

Le four électrique du laboratoire est essentiellement constitué des éléments suivants :

#### ➤ **Une résistance chauffante**

La résistance chauffante constitue l'élément essentiel du four. Elle est caractérisée par les paramètres suivants : la température limite supérieure d'emploi des résistances, la température de fusion des éléments résistifs, la température de la charge et la température de l'enceinte.

La résistance chauffante du four de laboratoire peut recevoir un signal de commande jusqu' à 25 volts DC.

- **Une enceinte**

Réalisée à partir des matériaux à caractère isolant ou réfractaire. Elle est chauffée à l'aide d'une résistance électrique, très bien calorifugée pour réduire le plus possible les déperditions thermiques. La charge à chauffer est placée dans cette enceinte.

- **Un bac**

C'est un récipient fixe destiné à contenir le liquide à chauffer, ce dernier est caractérisé par sa nature, sa forme et sa température initiale.

Le bac du four de laboratoire a une capacité de 3 litres de liquide à chauffer.

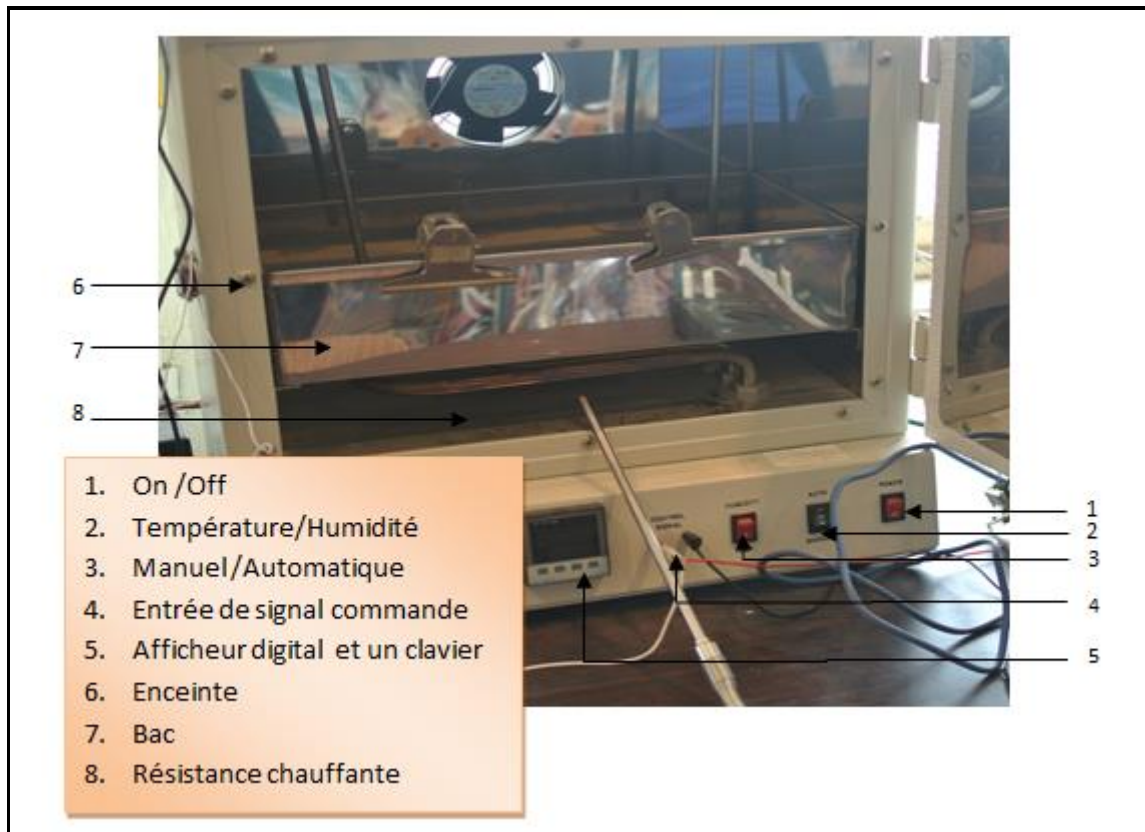


Figure I.3 Les différents constituants du four de laboratoire.

### I.2.2 Le circuit de puissance du four électrique [1]

La structure du circuit de puissance du four électrique est représentée par la figure (I.4), elle est constituée essentiellement d'un Triac qui est l'élément de base, nous donnons ci-dessous une brève description de son fonctionnement. La tension aux bornes de la résistance chauffante est modulée par le signal d'entrée de la gâchette.

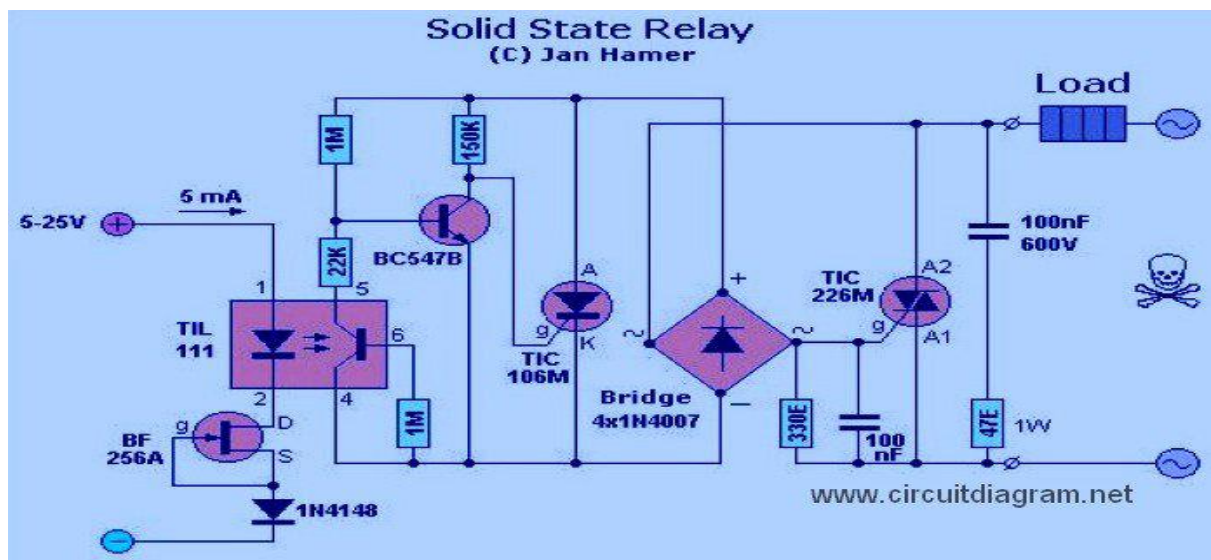


Figure I.4 Schéma de circuit de puissance du four électrique.



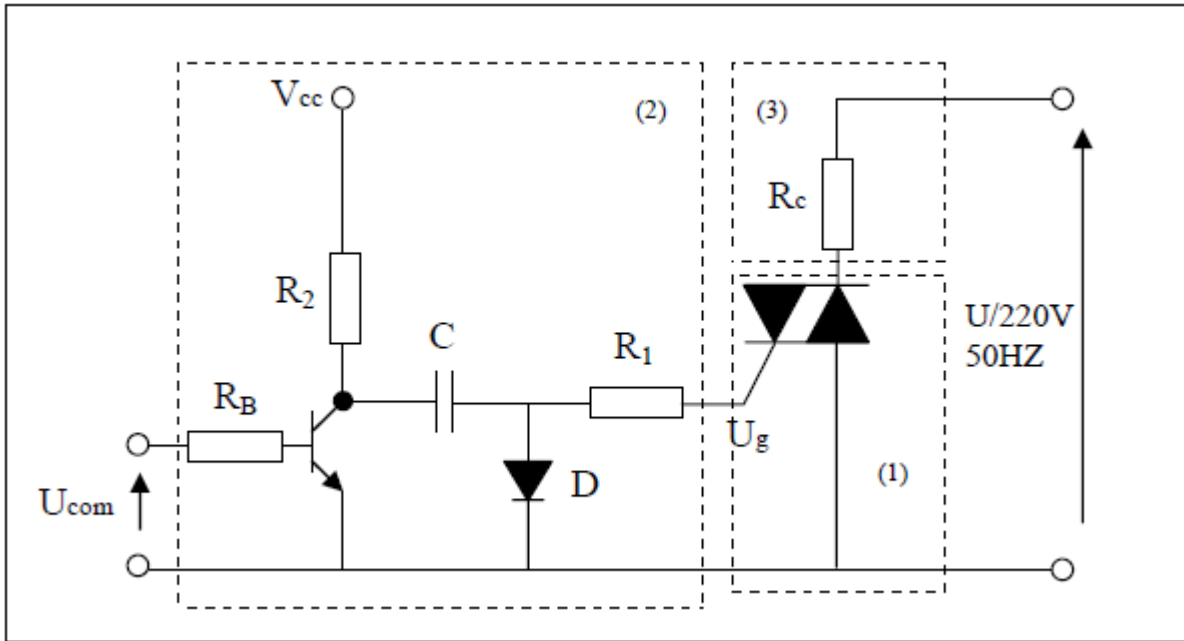


Figure I.5 Schéma simplifié de circuit de puissance du four.

Les différents régimes de conduction des Triacs se divisent en quatre quadrants, qui sont montrés par la figure (I.6). Pour notre cas nous utilisons les quadrants (1) et (3) afin de moduler les alternances positives et négatives de la tension du secteur 220 volts.

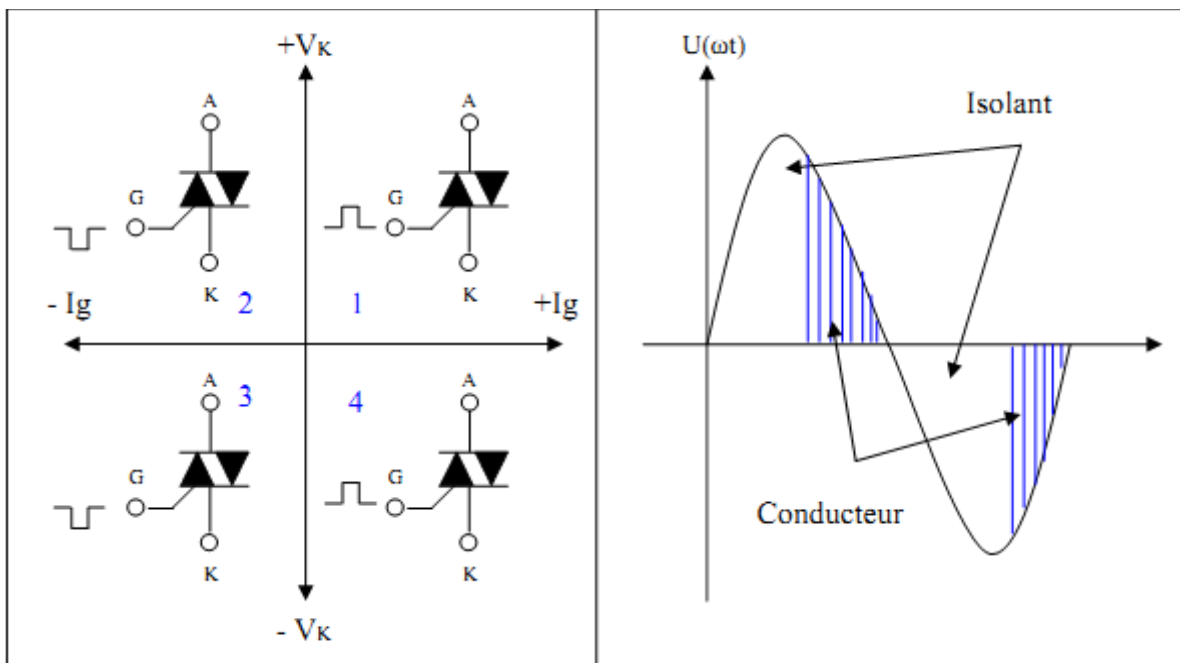
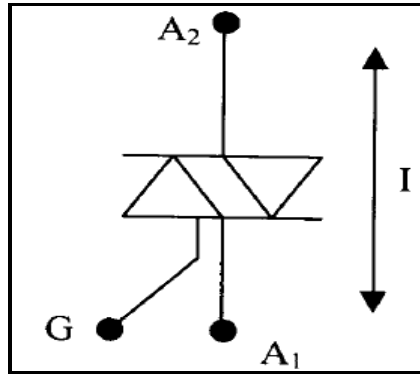


Figure I.6 Différents régimes de fonctionnement du Triac.

- **Description du Triac [3]**

Le composant sur lequel repose le principe de notre montage est le triac. Ce composant est un interrupteur commandé.



**Figure I.7 Représentation schématique d'un triac.**

En théorie, un triac possède deux modes de fonctionnement :

- Il n'est pas amorcé: il est équivalent à un interrupteur ouvert ou à une résistance infinie.
- Il est amorcé: il est équivalent à un interrupteur fermé ou à un fil.

Cependant, il arrive que, dans la pratique, cela se révèle différent.

- **La structure du triac**

Le triac est composé de trois bornes :

- Une électrode  $A_2$ ;
- Une électrode  $A_1$ ;
- Une électrode appelée gâchette G.

- **Principe de fonctionnement**

Le triac est conçu pour laisser passer le courant lorsqu'il est amorcé aussi bien de  $A_2$  vers  $A_1$  que de  $A_1$  vers  $A_2$ .

Le triac se rapproche du thyristor, mais ce dernier ne laisse passer le courant que dans un sens de  $A_2$  vers  $A_1$  (on parle d'anode  $A_2$  pour et de cathode pour  $A_1$ ).

En effet, le triac peut être assimilé à l'association de 2 thyristors en antiparallèle.

Le déclenchement donc la fermeture du circuit peut se faire par un courant de gâchette **positif** ou **négatif** quelle que soit la polarité de  $A_2$ .

### ➤ Amorçage du triac

Pour amorcer un triac, on injecte entre la borne  $A_1$  et la gâchette un courant  $I_G$  supérieur à  $I_{GT}$ , nous rendons le triac passant, le courant principal du circuit traversant le triac  $I_t$  peut alors croître.

Lorsque ce dernier dépasse le courant d'accrochage  $I_1$ , le composant est amorcé et  $I_G$  n'a plus d'influence, donc la gâchette peut être déconnectée.

### ➤ Désamorçage du triac

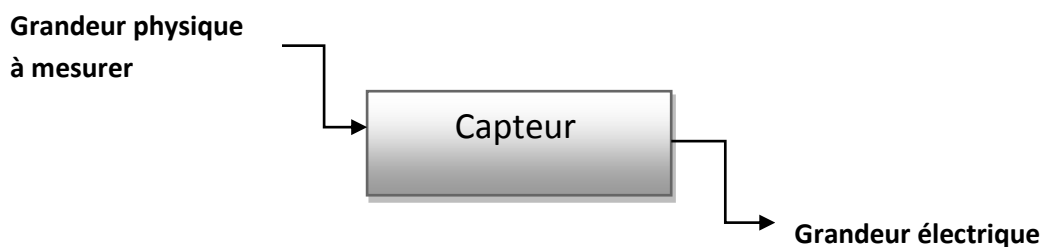
La seule manière de désamorcer le triac consiste à ramener le courant  $I_T$  en dessous du courant de maintien  $I_H$ .

## I.2 Présentation générale du capteur de température

Pour les fours industriels, la température représente la grandeur la plus importante, cette dernière influe sur les produits à fabriquer et doit être régulée et surveillée avec précision pour éviter toute variation de la qualité dans le produit final.

### I.2.1 Définition des capteurs de température

Les capteurs de température permettent de transformer la grandeur physique (la température) d'un processus ou d'une installation en un signal électrique au départ presque toujours analogique.



### I.2.2 Classification des capteurs de température

- **Capteurs de température de contact** : échange de chaleur entre l'objet et le capteur jusqu'à l'équilibre.
- **Pyromètres optiques (sans contact)** : basés sur la relation entre la température d'un corps et son rayonnement optique (infra-rouge ou visible).

Dans notre étude nous avons utilisé un capteur de température de contact à semi-conducteur qui est l'AD590.

### I.2.3 Capteur de température à semi-conducteur AD590 [4]

Les capteurs AD590 sont fabriqués par la société ANALOG DEVICES, ces derniers sont des éléments sensibles qui convertissent la mesure de la température en un courant de sortie proportionnel. La technologie de ces capteurs convient particulièrement aux applications de mesure et de régulation entre  $-55^{\circ}\text{C}$  et  $150^{\circ}\text{C}$ , là où la fiabilité, la linéarité et la précision des circuits électroniques est nécessaire.



Figure 1.8 Une photo réelle du capteur.

#### ➤ Principe de fonctionnement

La tension aux bornes du semi-conducteur (formant une diode ou un transistor) et le courant qui le traverse, dépend de la température.



- À courant constant  $I$ , la mesure de  $V$  est linéaire en fonction de la température

$$V = aT + b$$

$a$ : dépend de l'élément sensible

#### • Caractéristiques de l'AD590

Les caractéristiques essentielles de l'AD590 sont :

- Tension maximale directe (E+ à E-) :  $+44\text{V}$  ;
- Un courant de 1 micro-ampère par degré Kelvin ( $1\mu\text{A}/^{\circ}\text{K}$ ) ;
- Tension maximale inverse :  $-20\text{V}$  ;
- Alimentation : Plage de tension :  $+4\text{Vcc}$  à  $30\text{Vcc}$  ;
- Sortie: Courant nominal à  $25^{\circ}\text{C}$  ( $298,2^{\circ}\text{K}$ ) :  $298,2\mu\text{A}$ .

- **Avantages et inconvénients**

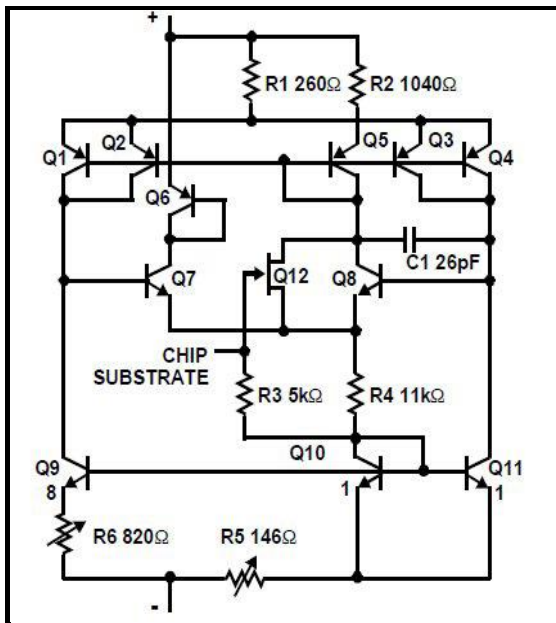
**Avantages**

- Simplicité ;
- Non linéarité faible.

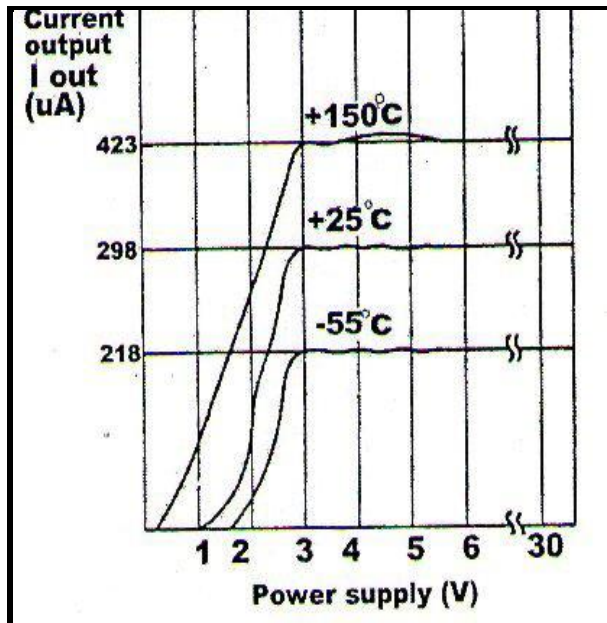
**Inconvénients**

- Etendue de mesure limitée (de  $-50^{\circ}\text{C}$  à  $150^{\circ}\text{C}$ ) ;
- Circuit de conditionnement relativement simple.

La structure interne de ce capteur est présentée en figure (I.9).



**Figure I.9** La structure interne du capteur de température à semi-conducteur AD590.



**Figure I.10** La Caractéristique du courant en fonction de l'alimentation du capteur AD590.

Pour plus de précision nous avons utilisé dans notre étude trois capteurs de température de même nature de type AD590, le premier est placé à l'extrémité droite, le deuxième au milieu et le dernier à l'extrémité gauche comme est illustré à la figure (I.11), la valeur envoyée au PC sera la moyenne de ces trois mesures.



Figure I.11 L'emplacement des trois capteurs

### I.3 Le circuit de conditionnement du capteur

Le circuit de conditionnement est utilisé pour transmettre en temps réel le signal de mesure obtenu via le capteur.

Le capteur AD590 génère un courant  $I$  proportionnel à la température en degré kelvin ( $1\mu A / 1^\circ \text{ Kelvin}$ ).

Le schéma du circuit de conditionnement est présenté en **figure (I.12)**.

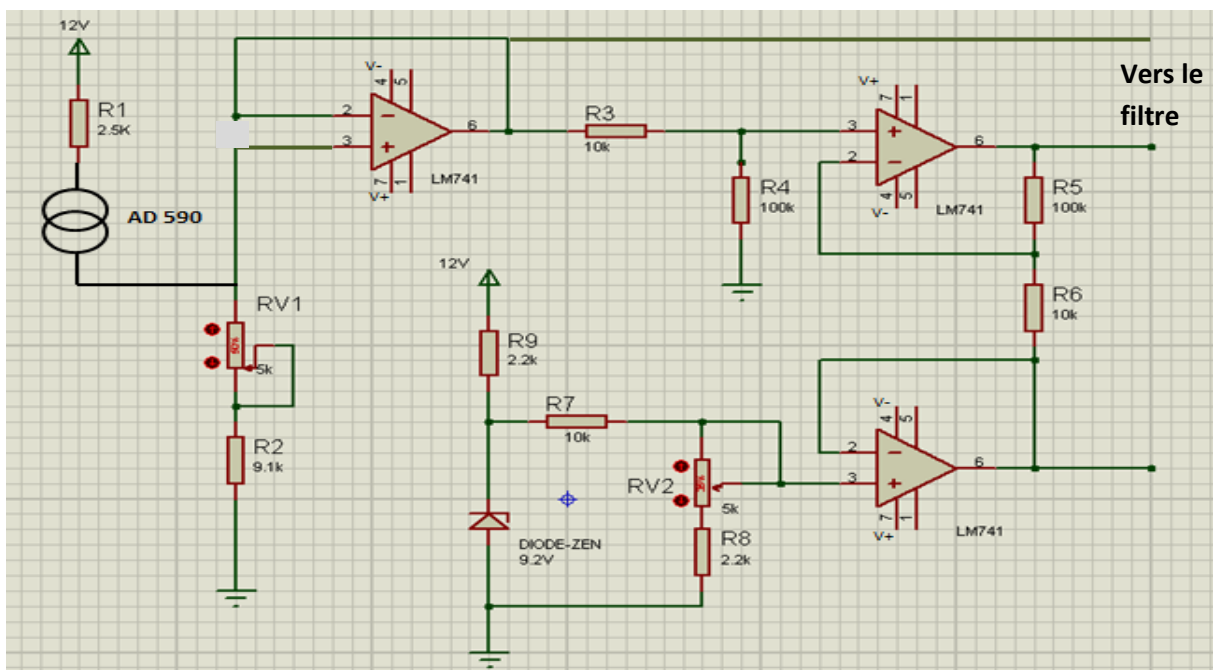


Figure I.12 Schéma électronique du circuit de conditionnement du capteur AD590

Ce circuit de conditionnement est constitué essentiellement d'un convertisseur courant tension qui convertit le courant en voltage exploitable et le normalise en ajustant une résistance variable afin d'obtenir un pas de  $0.1V/^\circ C$ .

Ce circuit est associé à un filtre passe bas du 2<sup>ème</sup> ordre « Sallen & Key » pour éliminer les signaux de haute fréquence et aussi à un diviseur de tension qui sert à diviser la tension sortante du capteur sur deux, c -à- d si on prend le maximum de température  $100^\circ C$ , qui correspond à  $10V$ , alors la tension qui peut être générée par les circuits de conditionnement après le passage par le diviseur de tension est de  $[0 - 5V]$ , qui est toujours inclus dans la marge de sécurité de notre PIC.

#### **I.4 Le filtre passe-bas**

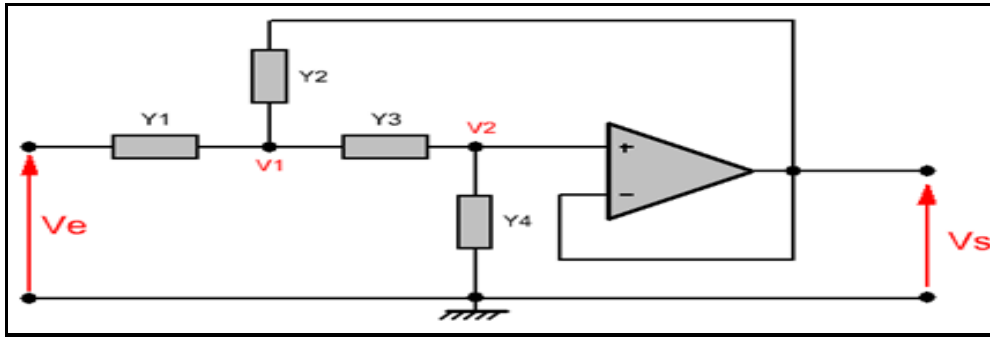
Un filtre passe-bas laisse passer les basses fréquences et qui atténue les hautes fréquences, c'est-à-dire les fréquences supérieures à la fréquence de coupure. Il pourrait également être appelé coupe-haut. Le filtre passe-bas est l'inverse du filtre passe-haut. Ces deux filtres combinés forment un filtre passe-bande. [6]

Le concept de filtre passe-bas est une transformation mathématique appliquée à des données (un signal). L'implémentation d'un filtre passe-bas peut se faire numériquement ou avec des composants électroniques. Cette transformation a pour fonction d'atténuer les fréquences supérieures à une fréquence de coupure et cela, dans le but de conserver uniquement les basses fréquences. La fréquence de coupure du filtre est la fréquence séparant les deux modes de fonctionnement idéaux du filtre : passant ou bloquant. [7]

#### **II .4.1 Le filtre passe-bas « Sallen & Key » [8]**

Un grand nombre de filtres actifs ont la structure suivante dite de « Sallen & Key » du second ordre. Ces filtres sont construits à partir de réseaux RC, comportant seulement des résistances et des condensateurs. L'absence de bobine d'auto-induction permet de les faire fonctionner à basse fréquence.

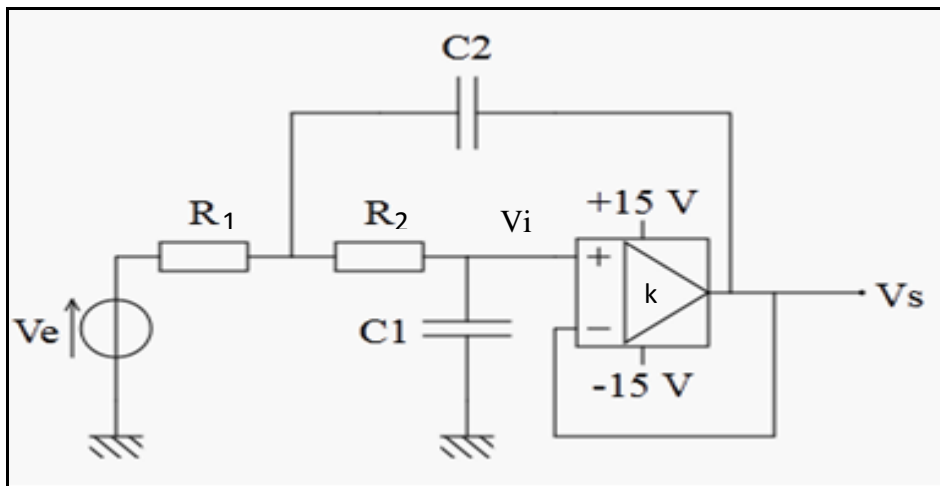
La figure (I.13) suivante montre le schéma d'un filtre passe-bas de « Sallen & Key ».



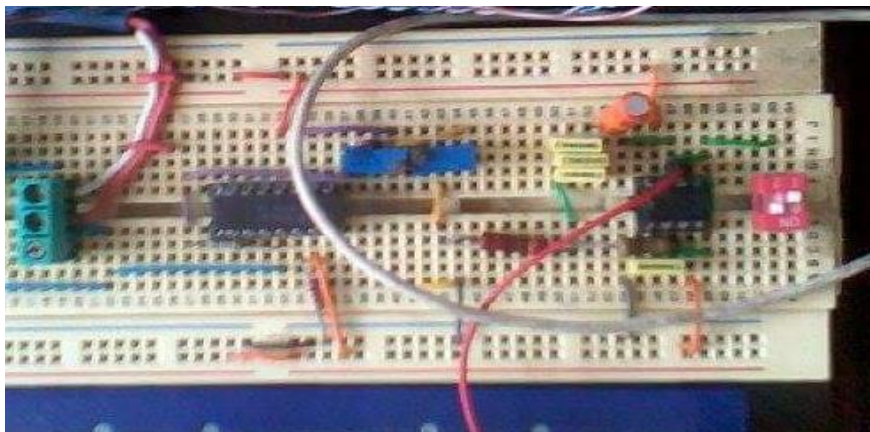
**Figure I.13** Schéma d'un filtre passe-bas de « Sallen & Key ».

L'élément actif est un amplificateur de tension de gain K. Idéalement, l'amplificateur doit avoir une impédance d'entrée assez grande pour pouvoir être considérée comme infinie, une impédance de sortie nulle. Il réalise la fonction suivante :

$$V_s(t) = K \times V_i(t) \quad (\text{I.1})$$



**Figure II.12** Le schéma du filtre passe-bas « Sallen & Key » utilisé dans notre montage.



**Figure I.13** Montage réel de filtre passe-bas sur une plaque d'essai



Pour réaliser un filtre passe-bas du second ordre on utilise deux résistances R identiques et deux condensateurs :  
Soit

$$Y_1 = \frac{1}{R_1}$$

$$Y_2 = j c_1 \omega$$

$$Y_3 = \frac{1}{R_2}$$

$$Y_4 = j c_2 \omega$$

La fonction de transfert normalisée de la forme passe-bas du second ordre pour ce montage est la suivante :

$$\frac{V_s}{V_e} = \frac{Y_1 Y_3}{(Y_1 + Y_2)(Y_3 + Y_4) + Y_3(Y_4 - Y_2)} \quad \mathbf{I.2}$$

La fonction de transfert H de passe-bas du second ordre est sous la forme :

$$H(P) = \frac{K}{1 + 2jm \frac{\omega}{\omega_c} + (j \frac{\omega}{\omega_c})^2} \quad \mathbf{I.3}$$

✓ Les calculs nous donnent, en remplaçant dans l'équation générale chaque admittance par son expression

$$H(j\omega) = \frac{\frac{1}{R_1 R_2}}{\left(\frac{1}{R_1} + j c_1 \omega\right) \left(\frac{1}{R_2} + j c_2 \omega\right) + \frac{1}{R_2} (j c_2 \omega - j c_1 \omega)} \quad \mathbf{I.4}$$

$$H(j\omega) = \frac{\frac{1}{R_1 R_2}}{\frac{1}{R_1 R_2} + \frac{j c_2 \omega}{R_1} + \frac{j c_1 \omega}{R_2} + c_1 c_2 (j\omega)^2 + j\omega \frac{(C_2 - C_1)}{R_2}} \quad \mathbf{I.5}$$

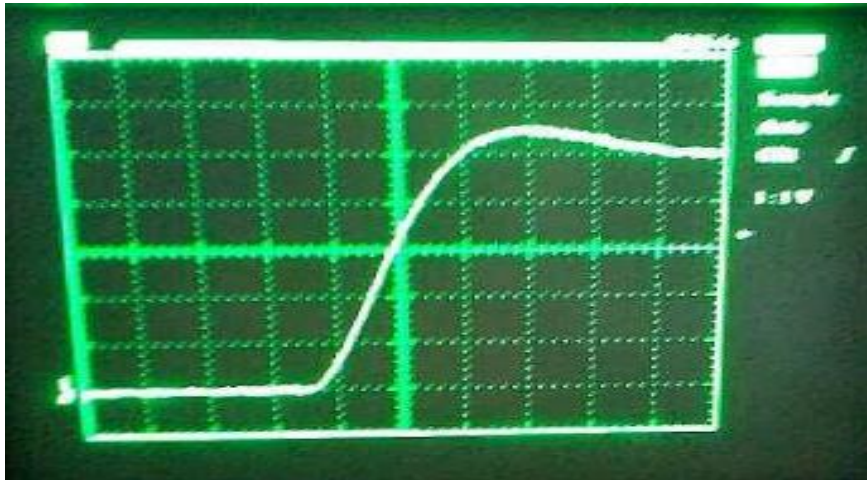
$$H(j\omega) = \frac{1}{1 + C_2(R_1 + R_2)j\omega + R_1 R_2 C_1 C_2 (j\omega)^2} \quad \mathbf{I.6}$$

✓ Ensuite, phase d'identification par rapport à la forme normalisée:

$$\frac{1}{\omega c} = \sqrt{R_1 R_2 C_1 C_2} \quad \longleftrightarrow \quad \omega c = \frac{1}{\sqrt{R_1 R_2 C_1 C_2}} \quad \mathbf{I.7}$$

$$\frac{2m}{\omega c} = C_2(R_1 + R_2) \quad \longleftrightarrow \quad m = \frac{C_2(R_1 + R_2)}{2\sqrt{R_1 R_2 C_1 C_2}} \quad \mathbf{I.8}$$

Avec  $K = 1$



**Figure II.8 Réponse indicielle réelle du filtre relevée par l'oscilloscope.**

## CONCLUSION

Dans ce chapitre nous avons décrit le procédé à commander qui est le four électrique disponible au niveau de laboratoire ainsi que son circuit de puissance, ensuite nous avons présenté le capteur de température utilisé AD590 et ses caractéristiques principales, enfin nous avons présenté le filtre basse bande de deuxième ordre utilisé .

# *Chapitre II*

## *Identification de la dynamique du four électrique*

## INTRODUCTION

La fonction de transfert réelle d'un procédé industriel est pratiquement impossible à déterminer. Il est alors nécessaire d'utiliser un modèle qui soit le plus représentatif possible de ce procédé.

Identifier un procédé, c'est rechercher à partir d'enregistrements, les paramètres qui caractérisent son modèle. Parmi les nombreuses méthodes d'identification existantes, nous utilisons des méthodes simples applicables sans matériel spécial et sans connaissances théoriques particulières.

La connaissance des paramètres caractéristiques d'un procédé peut-être utile en particulier dans les domaines suivants:

- Réglage des actions dans les boucles de régulation ;
- Choix des modes de régulation ;
- Modélisation des procédés pour des correcteurs numériques, afin de réaliser des régulations par modèle interne de référence.

Dans ce chapitre, on s'intéresse à l'identification de la dynamique du four électrique avec les trois éléments qui le compose, à savoir son circuit de puissance à triac, sa plaque chauffante et son capteur de température, utilisant la méthode d'identification qui permet de trouver un modèle de comportement traduisant le plus fidèlement le procédé autour d'un point de fonctionnement.

L'identification des systèmes est une procédure itérative dont l'objectif est l'obtention d'un modèle qui « valide ». La figure (II.1) ci-dessous représente les étapes importantes dans la procédure de l'identification.

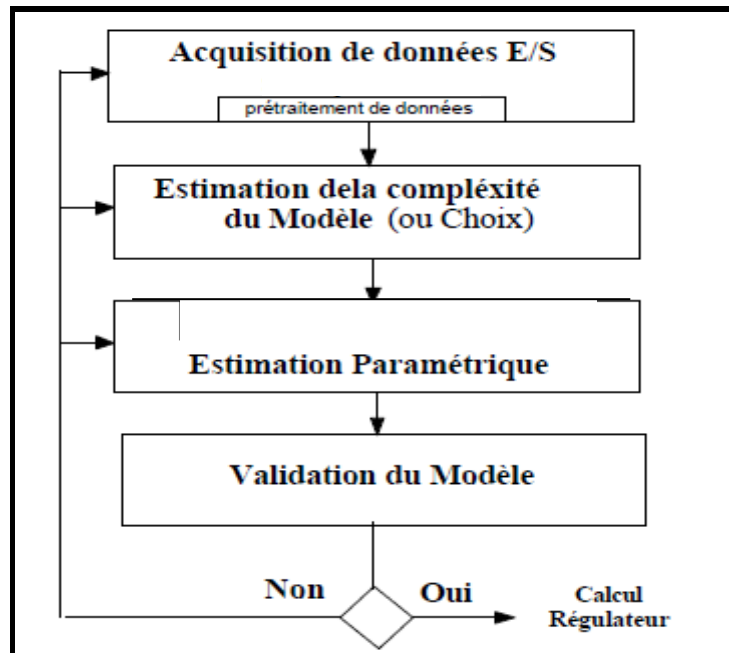


Figure II.1 La procédure d'identification

### II.1.1 Acquisition des données E/S

Cette partie expérimentale consiste à relever la courbe de réponse décrivant la dynamique du système disponible. Le schéma synoptique de la structure ayant servi à l'acquisition de la réponse temporelle du four en boucle ouverte est présenté en figure (II.2).

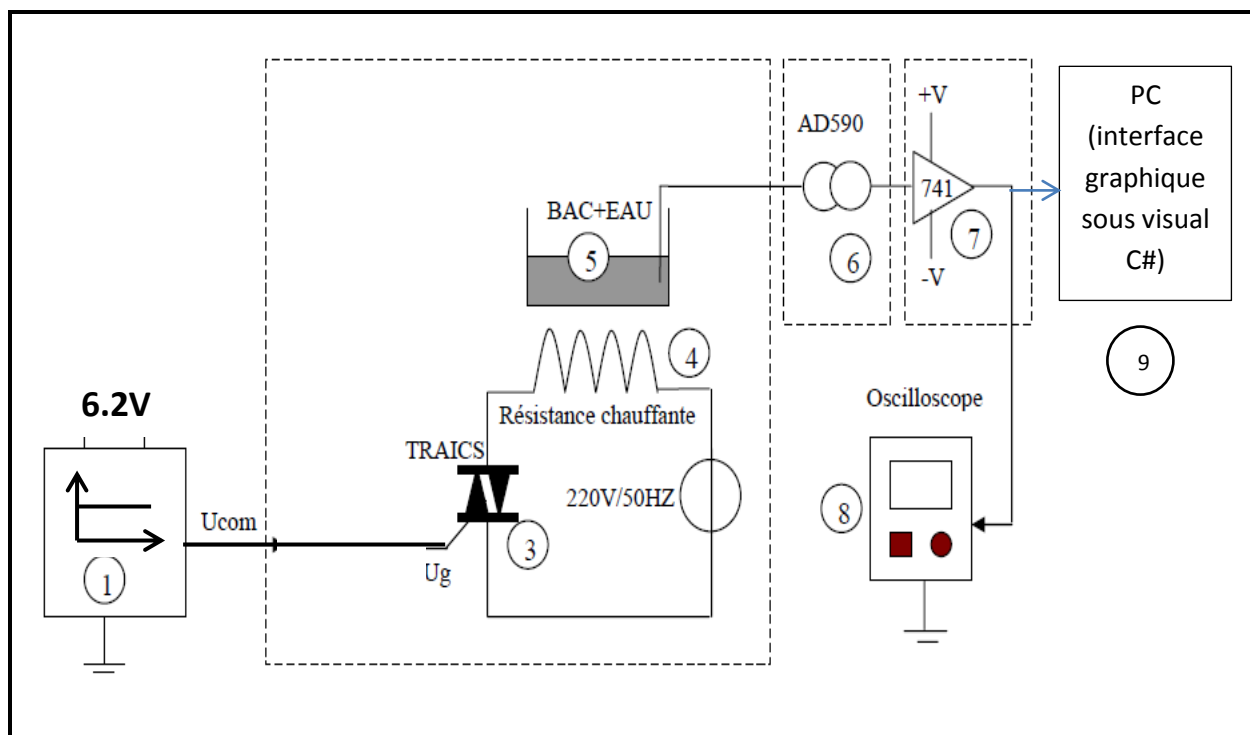


Figure II.2 Schéma synoptique ayant servi à l'acquisition de la réponse temporelle du four

Cet ensemble est constitué des éléments suivants :

- Module générateur de signal d'entrée bloc (1), tel que le signal d'entrée choisi correspond à un échelon de 6.2V
- Logique de contrôle de Triac à fin de générer l'impulsion de gâchette bloc (2).
- Les blocs (4) et (5) correspondent à la résistance chauffante avec un bac métallique pouvant contenir le liquide à chauffer.
- Les blocs (6), (7) et (8) représentent respectivement le capteur AD590 avec son circuit de conditionnement ainsi que l'oscilloscope à mémoire ayant suivi à l'enregistrement de la courbe de réponse.
- Le bloc (9) représente l'interface graphique utilisé pour le stockage des données.

La réponse temporelle relevée par l'oscilloscope est présentée à la figure (II.3).

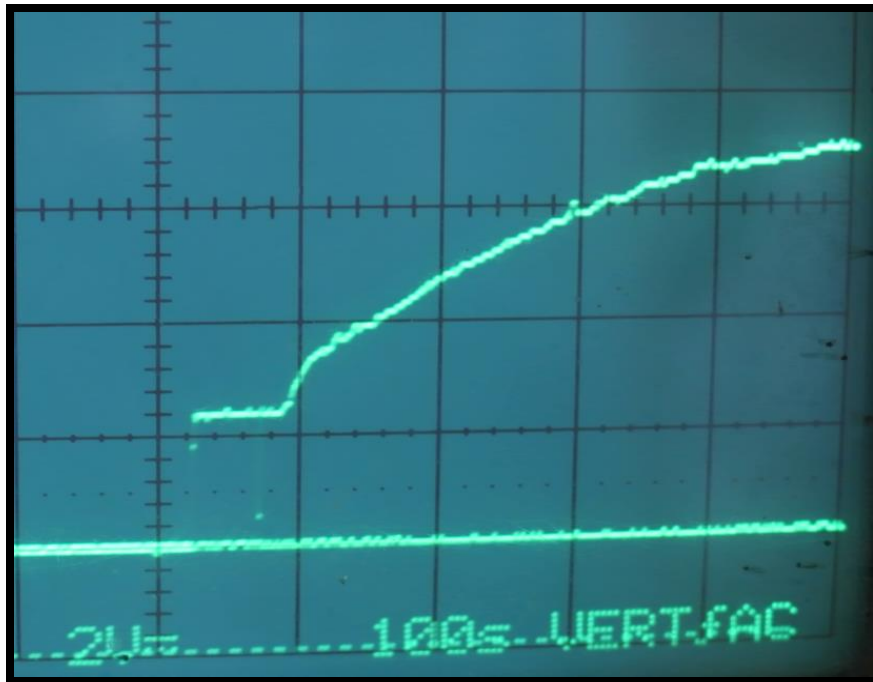


Figure II.3 Réponse du four relevé par l'oscilloscope après filtrage

Pour tracer la courbe de réponse sous Matlab, nous avons exploité les données fournies par l'interface graphique réalisée sous visual C#. Cette variation de température est représentée à la figure (II.4).

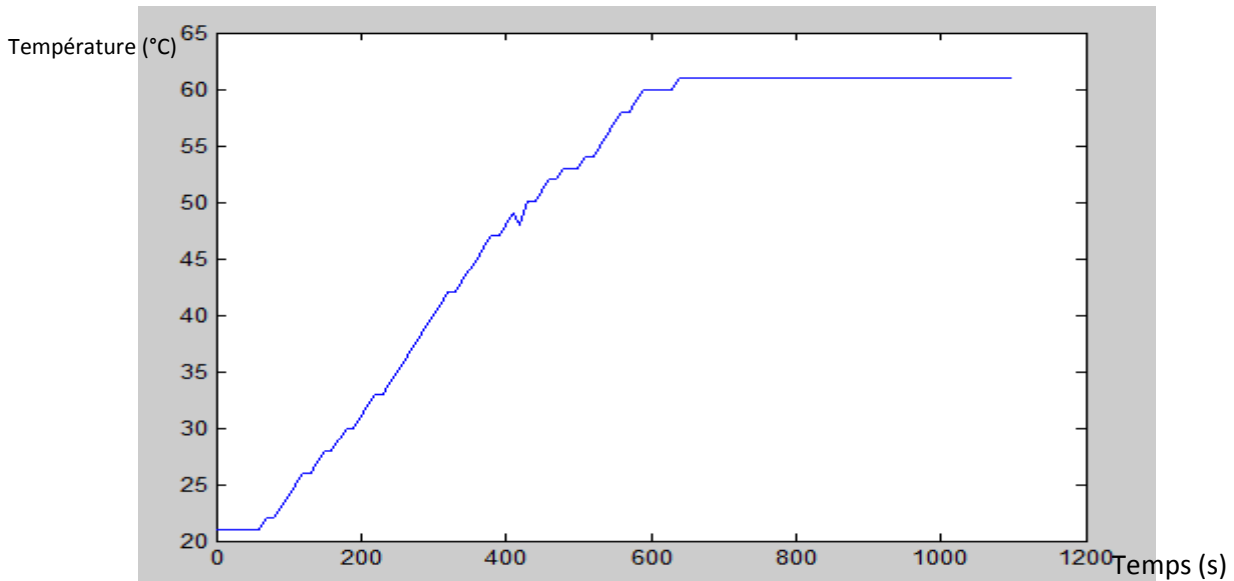


Figure II.4 Réponse réelle du four tracée sous MATLAB.

## II.2 Détermination des caractéristiques du four de laboratoire

La détermination de ces caractéristiques est faite en effectuant plusieurs tests sur le système en boucle ouverte.

### II.2.1 Test d'environnement (déterministe/stochastique)

Un modèle déterministe possède des entrées et des paramètres non bruités, de telle façon que son comportement soit parfaitement prévisible en avance. Alors que dans un modèle stochastique, les relations entre les variables sont données en termes de valeurs statistiques. Ce modèle contient un ou plusieurs paramètres ou entrées bruités permettant à son comportement de varier aléatoirement au cours du temps.

Pour notre processus, Nous appliquons un échelon de 5v. On répète l'expérience plusieurs fois tout en appliquant le même échelon, et on enregistre l'évolution de la sortie en fonction du temps pour chaque essai.

Nous avons constaté que pour les différents essais, les valeurs de température sont très proches. Il s'agit donc **d'un système déterministe**.

### II.2.2 Mono variable /multi variable

Un modèle mono variable est constitué d'une expression comportant une seule entrée et une seule sortie. Par opposition, un modèle multi variable est défini par des équations contenant plusieurs entrées et plusieurs sorties.

Pour notre processus, Nous avons une seule entrée sur laquelle on agit (le voltage) et une seule sortie (température). Il s'agit donc **d'un système mono variable**.

### II.2.3 Test de stationnarité (invariant / variable dans le temps)

Tout processus ayant des grandeurs ne changeant pas dans le temps et ne dépendent pas de son origine, est décrit par un modèle mathématique invariant dans le temps appelé aussi : modèle stationnaire. Sinon, il sera décrit par un modèle non stationnaire.

Pour notre processus, Nous appliquons à des différents instants, un échelon de différents voltages. On constate que la réponse du four ne dépend pas du choix de l'origine du temps. Il s'agit donc **d'un système stationnaire**.

### II.2.4 Test de linéarité (linéaire / non linéaire)

Dans le cas où la relation entre l'entrée et la sortie est linéaire, le modèle est appelé modèle linéaire. Par contre, si cette relation contient un ou plusieurs termes non linéaires, le modèle est non linéaire.

Pour notre processus, on applique, tout d'abord, un échelon de température de 10 %, autour d'un point de fonctionnement nous appliquons par la suite un échelon de température de 100 % de l'entrée, qui correspond à 50°C, Nous constatons que les sorties (températures en fonction du temps) ne sont pas proportionnelles. Il s'agit donc **d'un système non linéaire**.

On conclut qu'il s'agit d'un système mono variable, stationnaire, déterministe, non linéaire.

## II.3 Estimation des paramètres

### II.3.1 La méthode de STREJC

Cette méthode d'identification est basée sur les propriétés géométriques de la réponse indicielle, elle permet d'identifier des systèmes sous la forme :

$$G(s) = \frac{Ke^{-\tau}}{(1+T)^n} \quad \text{II.1}$$

#### a. Détermination de K :

Le coefficient de proportionnalité K est déterminé des conditions d'expériences comme le rapport de l'amplitude du signal de sortie à celui d'entrée

$$K = \frac{y(t)}{e(t)} = \frac{6.1}{6.2} = 0.98 \quad \text{II.2}$$

#### b. Détermination de n:

- On trace le mieux possible la tangente au point d'inflexion de la réponse indicielle.
- La tangente découpe un segment  $T_a$  sur l'axe des temps au bout d'un certain temps  $T_u$  (comportant un temps de retard inconnu).



- On calcule le rapport  $T_u / T_a$  et on choisira du tableau, la valeur de  $T_u/T_a$  qui correspond à une valeur de  $n$  entier, immédiatement inférieure.

### Application numérique :

$$T_u=100s.$$

$$T_a=392s.$$

Nous trouvons : Le rapport :  $T_u/T_a=100/392=0.255$

II .3

**Tableau II.1 Données de la méthode de STRETCJ**

n	$T_u/T$	$T_a/T$	$T_u/T_a$
1	0	1	0
2	0.28	2.72	0.1
3	0.8	3.7	0.22
4	1.42	4.46	0.32
5	2.10	5.12	0.41
6	2.81	5.70	0.49

### c. Détermination de la constante de temps T :

Connaissant la valeur de  $n$  et  $T_a$  on détermine la constante du temps  $T$  à l'aide de l'une des deux dernières colonnes du tableau.

Dans notre cas, pour  $n=3$ ,  $T_a / T = 3,695$  alors

$$T = \frac{T_a}{3.695} = \frac{392}{3.695} = 106,09\text{sec.} \quad \text{II .4}$$

### d. Détermination du temps de retard fictif $\tau$ :

Afin de compenser l'erreur due à la détermination du point d'inflexion, on introduit un retard fictif

$$\tau = T_{uR} - T_{uT} \quad \text{II .5}$$

$T_{uT}$ : déterminée à partir du tableau de **Stréjc** .

$T_{uR}$ : Valeur réelle de la grandeur  $T_u$  fixée sur la réponse indicielle.

Dans notre cas  $T_{uR} = 100\text{sec}$

$$\text{Alors} \quad \frac{T_{uR}}{T_a} = 0,218 \quad \text{II .6}$$

D'où :  $Tu_T = 0,218 * Ta = 0,218 * 392 = 85,456 \text{ sec.}$  **II.7**

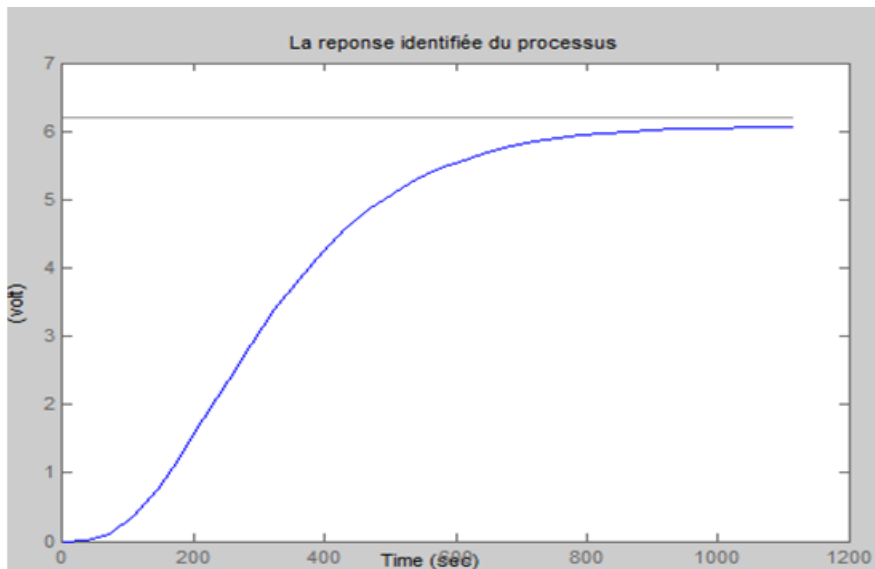
$$\tau = 100 - 85,456 = 14,544 \text{ sec}$$

Nous trouvons :  $\tau = 14,544 \text{ sec}$

Donc, nous trouvons la fonction de transfert suivante :

$$G(s) = \frac{0,98}{(1 + 106,09s)^3} * e^{-14,544s}$$

Après simulation sous MATLAB, la courbe obtenue est illustrée en figure (II.4).



**Figure II.5 La courbe identifiée simulée sous MATLAB.**

#### II.4 Validation du modèle identifié

Le test de validation permet pour un procédé donné de choisir le meilleur modèle, respectivement la meilleure structure et le meilleur algorithme pour l'estimation des paramètres.

La validation du modèle a les buts suivants :

- Evaluer la qualité du modèle qui doit représenter un compromis entre son ajustement aux données et sa complexité.
- La validation de nos fonctions de transfert, nous utilisons la méthode de calcul de l'erreur relative.

- Afin de comparer les deux courbes, réelle et identifiée, nous les avons représenté dans le meme graphe, en utilisant la structure hold on de MATLAB

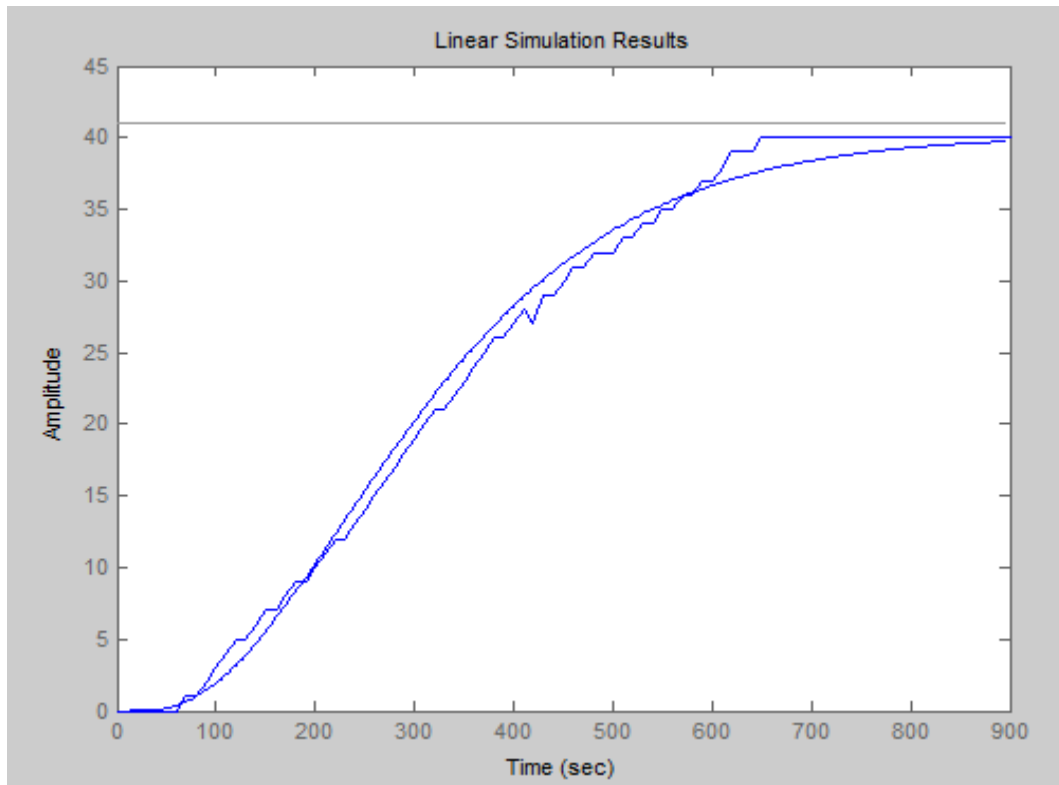


Figure II.6 Courbes de réponse de système réelle et celle du modèle identifié

#### II.4.1 Validation du modèle du four

Pour le calcul d'erreur relative, nous remplissons le tableau (II.2).

Tableau II .2 Tableau de l'erreur relative calculé pour le modèle du four électrique.

T(s)	Tm(t)	Ts(t)	Abs(Tm-Ts)
100	2	3	1
200	10	10	0
300	20.2	19	1.2
400	28.2	27	1.2
500	33.6	32	1.6
600	36.7	39	2.3
700	40	40	0

- A partir du ta bleau (2), on calcul l'erreur de calcul

$$\text{Erreur} = \frac{(\text{erreure max})}{(41)} * 100 \quad \text{II.19}$$

- Appliction numérique

$$\text{Erreur} = \frac{2.3}{41} = 5.6\% \quad \text{II.20}$$

**Conclusion :**

Dans ce chapitre nous avons identifié le four électrique du laboratoire ainsi que le filtre utilisé, en utilisant la méthode graphique de **STREJC**.

Le résultat obtenu donne une erreur de **5.6 %** pour le four ce qui nous a permis la validation de modèle trouvé.

# *Chapitre III*

*Etude et réalisation de la  
carte d'interface  
électronique*

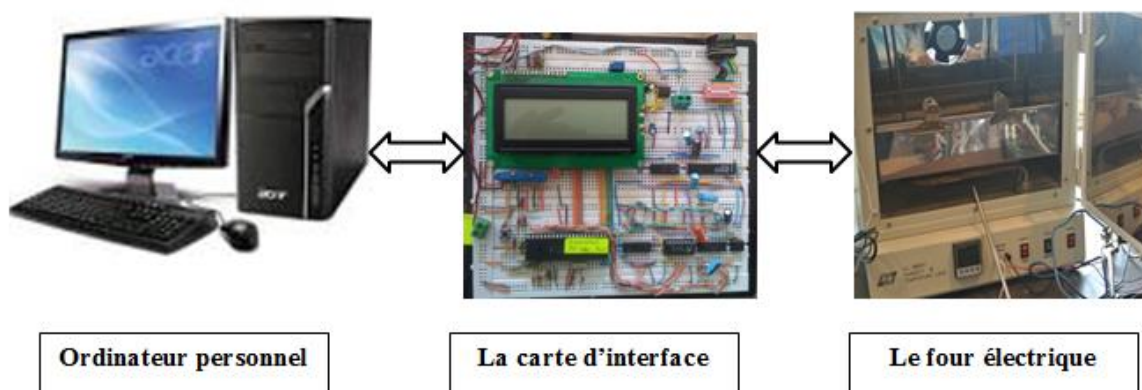
## INTRODUCTION

Dans les chapitres précédents nous avons élaboré une description de notre processus représenté par le four électrique du laboratoire ainsi que le capteur avec son circuit de conditionnement.

Pour commander le four électrique à résistance nous avons choisi une commande hybride qui est une combinaison entre la commande par microcontrôleur et la commande par PC. Cette dernière nécessite l'utilisation d'une carte d'interface entre le micro-ordinateur et le four électrique.

Dans ce chapitre, nous étudions la partie Hardware du projet en présentant la structure générale de la carte électronique de communication ainsi que les composants qui la constituent et leurs schémas de brochages. D'une manière générale, on peut diviser notre système en trois parties principales (Voir la figure III.1).

- 1) L'ordinateur personnel (PC)
- 2) La carte électronique d'interface
- 3) Four électrique.



**Figure III.1 La répartition principale du système**

### III.1 Structure générale de la carte d'interface

Le capteur de température génère un signal analogique, ce signal doit être converti en un signal numérique avant qu'il soit envoyé vers le PC à travers le port série. Lorsque le signal numérique est présent dans le PC le Software de contrôle joue son rôle.

Afin de générer le signal de commande, de nature analogique et variable, il est nécessaire d'utiliser un convertisseur numérique/analogique de hautes performances, associé à un registre à décalage.

La carte d'interface peut être divisée en trois unités (voir figure III.2) :

- Unité d'entrée ;
- Unité d'adaptation ;
- Unité de sortie.

Par la suite, nous examinons les composants de chaque unité indépendamment et comment sont liés les uns aux autres.

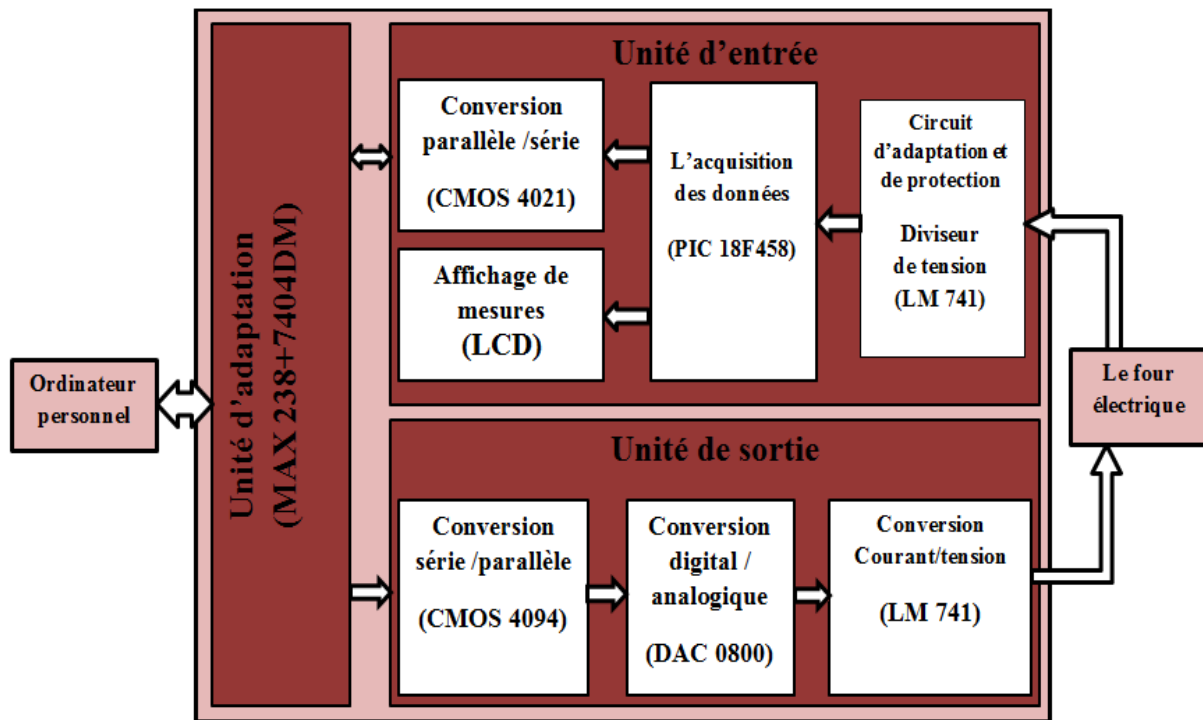


Figure III.2 Diagramme bloc général de la carte d'interface

### III.1.1 Unité d'adaptation

Le port série du PC de type RS-232 génère les données entre -12v et +12v. La famille TTL (Transistor-Transistor Logic) fonctionne entre 0V et 5V, pour cela il est nécessaire d'utiliser un circuit adaptatif pour convertir la tension d'une plage à une autre.

La solution adaptée pour la conversion des signaux RS232 au standard TTL, consiste à utiliser un circuit intégré sous la dénomination MAX238.

III.1.1.1 Circuit MAX238 [9]

Le MAX238 est un circuit intégré, fabriqué par la société MAXIM dont la fiabilité n'est plus à démontrer. Il permet d'obtenir une conversion dans les deux sens RS232/TTL et TTL/RS232. Sa structure interne est représentée à la figure (III.3).

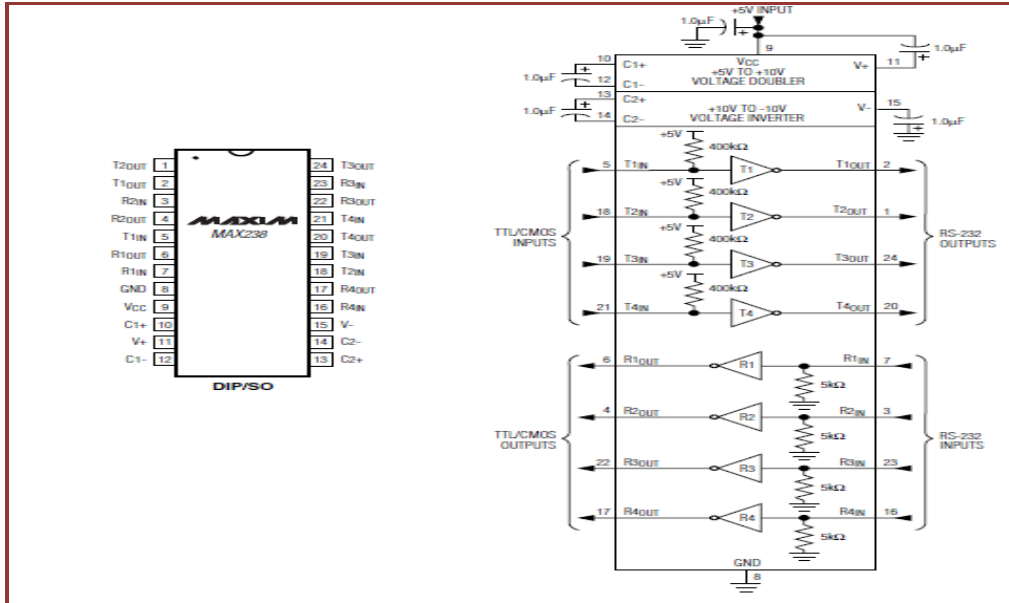


Figure III.3 Structure interne du MAX238

A partir du schéma interne du MAX238 (figure III.3), nous constatons que les signaux sortants du MAX238 sont inversés en état logique. Ça veut dire, à l'état logique binaire 1, l'état logique à la sortie du MAX238 est 0 et vice-versa. Pour assurer une fonction complète de la conversion, nous ajoutons à la sortie du circuit MAX238 un autre circuit logique qui est le 7404 DM. Le schéma de branchement du MAX238 avec le port série et le 7404 est représenté en figure (III.4).

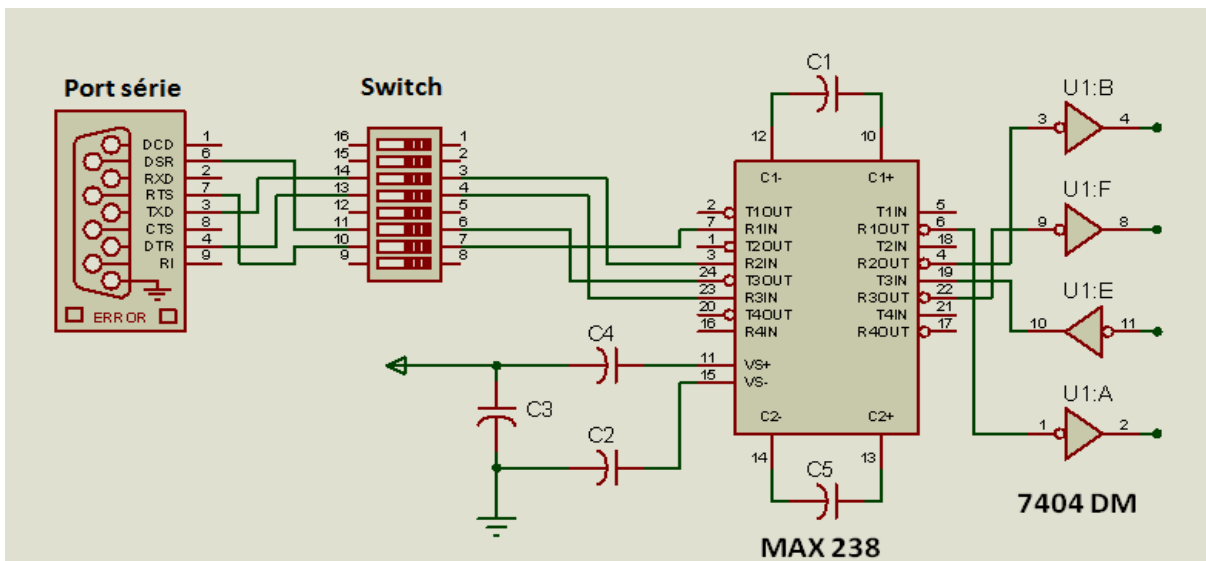


Figure III.4 Branchement du MAX238 avec RS232 et 7404DM



### III.1.2 Unité d'entrée

L'unité d'entrée est une circuiterie électronique qui contient des organes d'acquisition et de traitement qui permettent de rendre les signaux issus des capteurs de température exploitables (conversion analogique/numérique, conversion parallèle /série) et aussi d'afficher les valeurs réelles de température.

Cette unité est constituée des organes suivants :

1. Circuit d'adaptation de la tension issue du capteur
2. Microcontrôleur (18F458)
3. Afficheur alphanumérique(LCD)
4. Convertisseur parallèle/série (CMOS 4021)

#### III.1.2.1 Circuit d'adaptation de la tension issue du capteur

Afin d'adapter les signaux du convertisseur avec les signaux issus du capteur de température, nous devons élaborer un circuit adaptateur. Le microcontrôleur permet de recevoir une tension  $V_{in}$  de 5Volts au maximum, tandis que le capteur AD590 avec son circuit de conditionnement délivre un voltage compris entre 0Volts et 10Volts correspondant respectivement à 0 °C et la température maximale du four.

Pour remédier à ce problème, nous avons proposé un montage en diviseur de tension actif à base d'un amplificateur opérationnel de type LM741. Le schéma de branchement de ce circuit est représenté en figure (III.5).

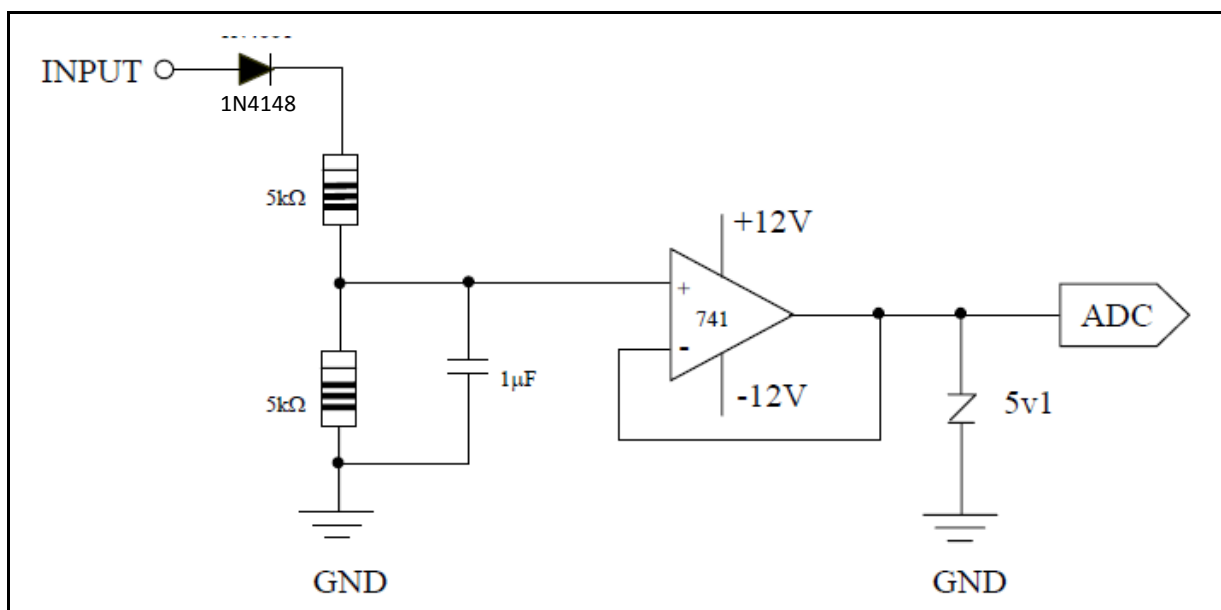


Figure III.5 Schéma de branchement circuit d'adaptation tension du Capteur

Pour protéger le PIC contre les surtensions, nous avons ajouté un composant stabilisateur à 5.1Volts. Au niveau de l'entrée nous complétons la protection contre les inversions de tension à l'aide d'une diode à semi conducteur à commutation rapide de type 1N4148.

### III.1.2.2 Le Microcontrôleur 18F458 [10, 11, 12,13]

#### III.1.2.2.1 Présentation du microcontrôleur

L'élément d'acquisition principal de la carte d'interface est le **PIC18F458** (Peripheral Interface Controller) qui est une unité de traitement de l'information de type processeur embarqué à laquelle est ajouté des périphériques internes (les périphériques étant intégrés RAM, EEPROM, Ports...etc.) permettant de réaliser des montages sans nécessiter l'ajout des composants externes. Ils sont appréciés pour leur rapidité d'exécution et leur simplicité de mise en œuvre et leur prix de revient faible.

L'avantage principal du processeur embarqué est que celui-ci travaille avec un programme logé dans une mémoire flash, donc facilement modifiable.

#### III .1.2.2.2 Les principaux constituants du PIC

Le microcontrôleur est constitué essentiellement de :

- Une unité de calcul (**A.L.U**)
- Une mémoire de type **RAM** de 1536 Octet
- Une mémoire programme de type flash
- Une mémoire EEPROM
- Les différents registres
- Des ports entrées/sorties de 33 lignes
- Une mémoire **SP** (pile)

#### III.1.2.2.3 Le choix du PIC

Le choix du PIC est directement lié à l'application envisagée et avant il faut déterminer :

- Le nombre d'entrées /sorties nécessaires pour l'application.
- La nécessité des convertisseurs **A/D**.
- La rapidité d'exécution (il faut vérifier la vitesse maximale du **PIC** choisi et la vitesse maximale nécessaire pour l'application).
- La taille de la **RAM** interne.

Il est indispensable de disposer soit d'un DATA-BOOK soit d'un tableau comparatif pour choisir le **PIC** le plus adapté à l'application envisagée. Dans notre cas, notre choix est porté sur le **18F458**.

Ce dernier possède des instructions très puissantes telle que:

- Une programmation simple donc un programme réduit.
- La disposition de plus de mémoire programme (ce qui est nécessaire pour développer les programmes les plus longs).
- La disposition de nombre important d'entrée/sortie qui assure la communication avec les périphériques branchés autour du **PIC**.
- La présence d'un convertisseur analogique/ numérique (indispensable pour la mesure d'un signal généré par les capteurs), avec 8 canaux multiplexés repartis en port A et port E.

#### III.1.2.2.4 Brochage du PIC18F458

Ce microcontrôleur se présente sous la forme d'un boîtier DIL à 40 broches comme schématisé dans la figure(III.6).

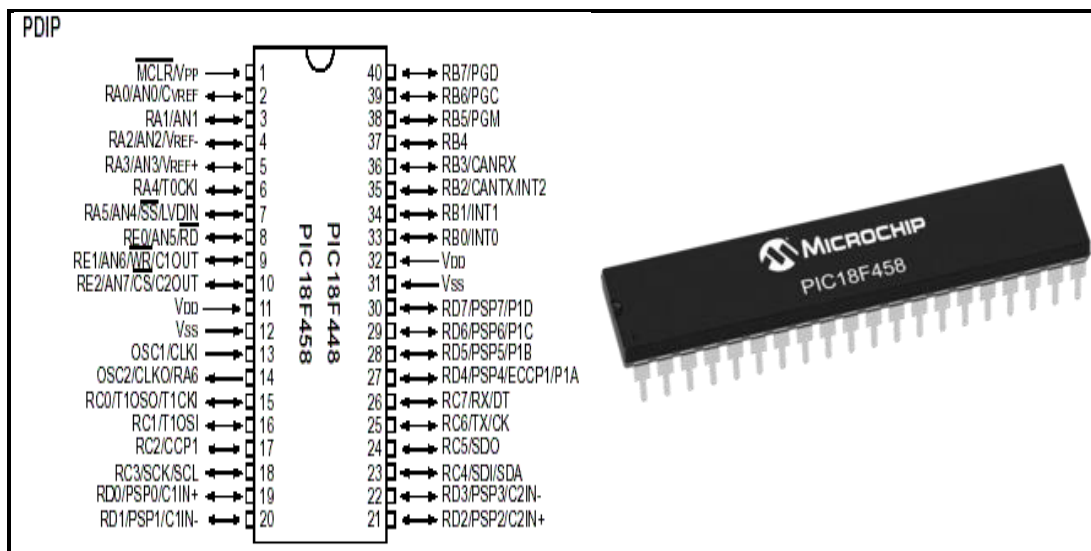


Figure III.6 Les différentes broches du PIC 18F458

Le PIC **18F458** comprend 40 pins: 33pins d'entrées/sorties, 4 pins pour l'alimentation, 2pins pour l'oscillateur et un pin pour le Reset (MCLR).

➤ Le **PIC18F458** contient les 5 ports suivants :

- Port A : 6 pins I/O numérotés de RA0 à RA5
- Port B : 8 pins I/O numérotés de RB0 à RB7

- Port C : 8 pins I/O numérotés de RC0 à RC7
  - Port D : 8 pins I/O numérotés de RD0 à RD7
  - Port E : 3 pins I/O numérotés de RE0 à RE2
- Les broches  $V_{DD}$  et  $V_{SS}$  servent pour l'alimentation du PIC. Il y a deux connections  $V_{DD}$  et deux connections  $V_{SS}$ , la présence de ces 2 pins s'explique pour une raison de dissipation thermique. Les courants véhiculés dans le PIC sont loin d'être négligeables du fait des nombreuses entrées/sorties disponibles. Pour cette raison la répartition des courants est faite en plaçant 2 pins pour l'alimentation  $V_{SS}$ , bien évidemment, pour les mêmes raisons, ces pins sont situés de part et d'autre du PIC, et en positions relativement centrales.
- La broche MCLR sert à initialiser le PIC en cas de la mise sous tension, de remise à zéro externe, en cas de la baisse de tension d'alimentation cette broche doit être reliée à la  $V_{DD}$  via un switch et une résistance de  $1K\Omega$ .
- Les broches OSC1 et OSC2 ou CLKIN et CLKOUT permettent de faire fonctionner l'oscillateur interne du PIC qui peut être un quartz (voir figure III.7), un résonateur céramique, un oscillateur externe ou un réseau RC. La fréquence de l'horloge interne du PIC est obtenue en divisant par 4 la fréquence de l'horloge externe. Pour un quartz à 16MHz, la fréquence interne est donc de 4MHz donc la durée d'un cycle est de  $1/4 \cdot 10^6$  s.

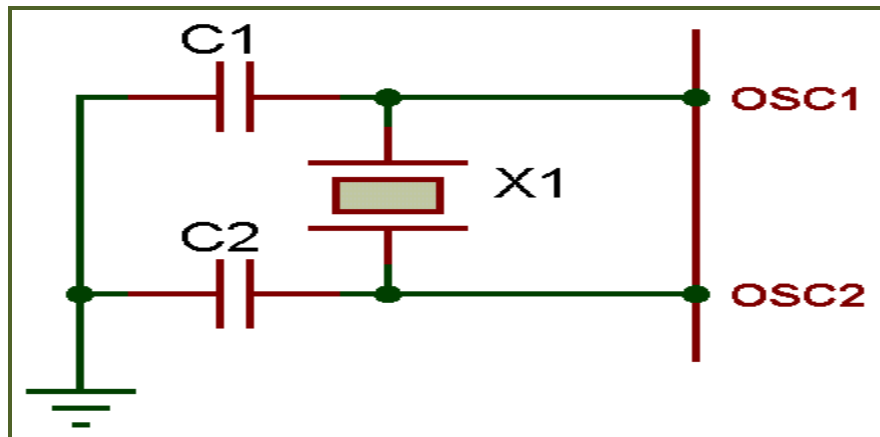


Figure III.7 Brochage de l'oscillateur à quartz

Lors de la programmation, la broche MCLR doit être portée à un niveau compris entre 12V et 14V et le PIC18F458 commence à exécuter le programme en appliquant un signal d'horloge sur la broche RB6 (broche39) et les informations binaires transitent en série sur la broche RB7 (broche40). Chacune des informations qui transitent sur la broche RB7 est validée à la retombée du signal d'horloge sur la broche RB6.

### III.1.2.2.5 Les principales caractéristiques du PIC 18F458

Le PIC 18F458 est caractérisé par : (voir Tableau III.2)

**Tableau III.1 Les caractéristiques du PIC 18F458**

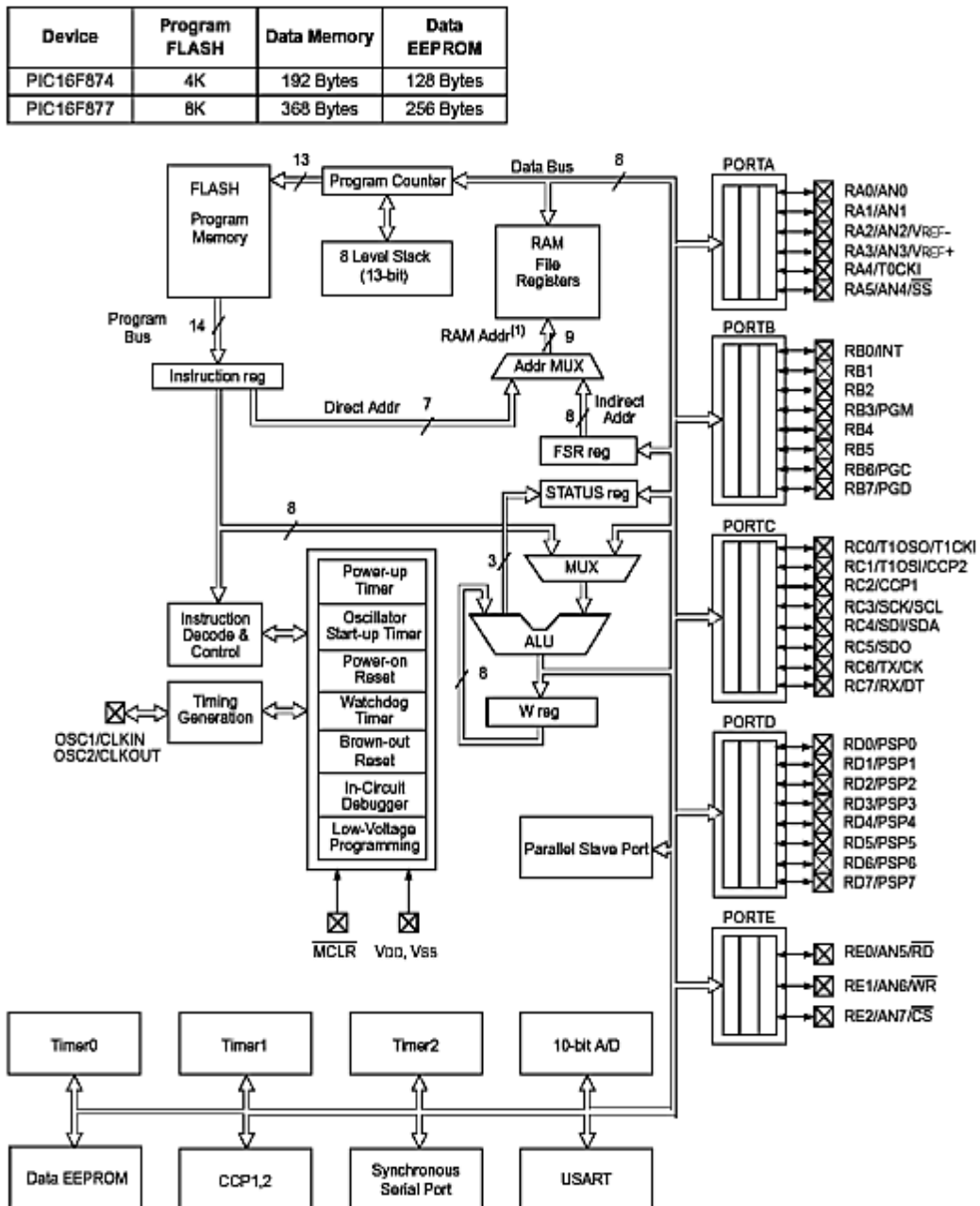
PIC	FLASH	RAM	EEPROM	I/O	A/D	PORT Parallèle	Port série
18F458	32k bits	1.536	256	33	8	PSP	1-A/E/USART, 1- MSSP(SPI/I2C)

- Une fréquence de fonctionnement élevée, jusqu'à 40MHz.
- Une mémoire vive de 1.536 octets que nous allons détailler plus tard.
- Une mémoire EEPROM pour sauvegarder des paramètres de 256 octets.
- Une mémoire morte de type FLASH de 32 Kbits, elle est réinscriptible à volonté.
- Chien de garde WDT.
- 33 entrées et sorties, chaque sortie peut sortir un courant maximum de 25mA.
- 4 Temporisateurs TIMER :
  - TIMER0** :8bits/16bits temporisation /compteur avec 8bits avec pré diviseur,
  - TIMER1** : (16 bits avec pré diviseur),
  - TIMER2** :8bits temporisateur /compteur avec registre de période 8 bits (base de temps pour PWM),
  - TIMER3** :16bits temporisateur /compteur
- 2 entrées de captures et de comparaison avec PWM (Modulation de largeur d'impulsions)
- Convertisseurs analogique/numérique 10 bits avec 8 entrées multiplexées maximum
- Une interface de communication série asynchrone et synchrone.
- Une tension d'alimentation entre 0et 5v.
- Cycle écriture et effacement en mémoire flash de 100000.

### III.1.2.2.6 Architecture interne du PIC 18F458

La figure (III.8) montre les différents composants du PIC18F458 dont on distingue les mémoires RAM, ROM et EEPROM, l'unité arithmétique et logique (U.A.L), les différents ports, les différents bus, l'unité de conversion A/D...etc.

Pour pouvoir manipuler et programmer les Pics en assembleur en particulier, il est indispensable de connaître les différents registres spécialisés SFR (TRISA, PORTA, ADCON1...) et ceux banalisés GPR (ou l'on stocke les différentes variables) de la mémoire RAM et leur adressage même en langages les plus évolués tel que le MikroC, la connaissance des registres spécialisés est essentielle. La figure(III.9) montre l'organisation de la mémoire RAM. Constituée de 16 Banks (page mémoire).



Note 1: Higher order bits are from the STATUS register.

Figure III.8 Architecture interne du PIC18F458

❖ **L'organisation de la mémoire RAM**

La figure III.9 ci-dessus illustre l'organisation de la mémoire RAM, en particulier la position des Registres Spéciaux SFR (SPECIAL FUNCTION REGISTERS) et GPR (GENERAL PERPOSE REGISTERS).

Addr.	Name	Addr.	Name	Addr.	Name	Addr.	Name
00h	INDF	80h	INDF	100h	INDF	180h	INDF
01h	TMR0	81h	OPTION_REG	101h	TMR0	181h	OPTION_REG
02h	PCL	82h	PCL	102h	PCL	182h	PCL
03h	STATUS	83h	STATUS	103h	STATUS	183h	STATUS
04h	FSR	84h	FSR	104h	FSR	184h	FSR
05h	PORTA	85h	TRISA	105h	WDTCON	185h	SRCON
06h	PORTB	86h	TRISB	106h	PORTB	186h	TRISB
07h	PORTC	87h	TRISC	107h	CM1CON0	187h	BAUDCTL
08h	PORTD	88h	TRISD	108h	CM2CON0	188h	ANSEL
09h	PORTE	89h	TRISE	109h	CM2CON1	189h	ANSELH
0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH
0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh	INTCON
0Ch	PIR1	8Ch	PIE1	10Ch	EEDAT	18Ch	ECON1
0Dh	PIR2	8Dh	PIE2	10Dh	EEADR	18Dh	ECON2
0Eh	TMR1L	8Eh	PCON	10Eh	EEDATH	18Eh	Not Used
0Fh	TMR1H	8Fh	OSCCON	10Fh	EEADRH	18Fh	Not Used
10h	T1CON	90h	OSCTUNE	110h		190h	
11h	TMR2	91h	SSPCON2				
12h	T2CON	92h	PR2				
13h	SSPBUF	93h	SSPADD				
14h	SSPCON	94h	SSPSTAT				
15h	CCPR1L	95h	WPUB				
16h	CCPR1H	96h	IOCB				
17h	CCP1CON	97h	VRCON				
18h	RCSTA	98h	TXSTA				
19h	TXREG	99h	SPBRG				
1Ah	RCREG	9Ah	SPBRGH				
1Bh	CCPR2L	9Bh	PWM1CON				
1Ch	CCPR2H	9Ch	ECCPAS				
1Dh	CCP2CON	9Dh	PSTRCON				
1Eh	ADRESH	9Eh	ADRESL				
1Fh	ADCON0	9Fh	ADCON1				
20h		A0h					
	General Purpose Registers 96 bytes		General Purpose Registers 80 bytes		General Purpose Registers 96 bytes		General Purpose Registers 96 bytes
7Fh		FFh		17Fh		1EFh	

Figure III.9 L'organisation de la mémoire RAM

❖ **Etude de convertisseur Analogique/Numérique**

Le convertisseur analogique/numérique est un périphérique intégré, destiné à recevoir une tension et la convertir en une valeur numérique qui pourra alors être utilisée par un système de traitement numérique.

Le PIC18F458 dispose de 8 convertisseurs analogique/numérique répartis sur le portA et le portE (voir figure III.6) qui permettent un échantillonnage sur 10bits. Ces broches peuvent être utilisées comme simple entrées-sorties digitales ou bien comme entrées des unités analogiques AN0, AN1,..., AN7. La conversion se fait avec la configuration des registres suivant. Dont la figure (III.10) Ci-dessous illustre l'organigramme de l'algorithme de conversion.

- ADCON0
- ADCON1
- ADRESL
- ADRESH

➤ **Registre ADCON0**

Ce registre en BANK0 permet de configurer les paramètres de fonctionnement des convertisseurs comme l'indique le tableau III.2.

**Tableau III.2 Bits de configuration du registre ADCON0**

B7	B6	B5	B4	B3	B2	B1	B0
ADS1	ADCS0	CHS2	CHS1	CHS0	GO/DOWN	-	ADON

Les fonctions des bits sont définies de la manière suivante :

ADON=1 : Conversion valide

ADON=0 : Conversion non valide

GO/DOWN=1: Start conversion

GO/DOWN=0: Stop conversion

CH0, CH1, CH2 permet de sélectionner les canaux comme l'indique le tableau (III.3)

**Tableau III.3 Sélection des canaux**

CH2	CH1	CH0	Channel select
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

ADCS1, ADCS2 sont les bits de sélection de la fréquence de conversion en fonction de FOSC selon le tableau (III.4).



**Tableau III.4 Configuration de la fréquence de conversion**

ADCS1	ADSC2	Fréquence
0	0	Fosc/2
0	1	Fosc/8
1	0	Fosc/32
1	1	RC

**Fosc** correspond à la fréquence du quartz /4

➤ **Registre ADCON1**

Les registres situés en BANK1 permet de configurer les broches de port A et de port E en entrées digitales ou analogiques et de sélectionner les registres où doivent être sauvegardés des résultats sur 10bits voir tableau(III.5).

**Tableau III.5 Bits de configuration du registre ADCON1**

B7	B6	B5	B4	B3	B2	B1	B0
ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0

Le mode de sélection est illustré par le tableau disponible en datasheet au niveau de site web de microchip.

### III.1.2.3 L'afficheur LCD

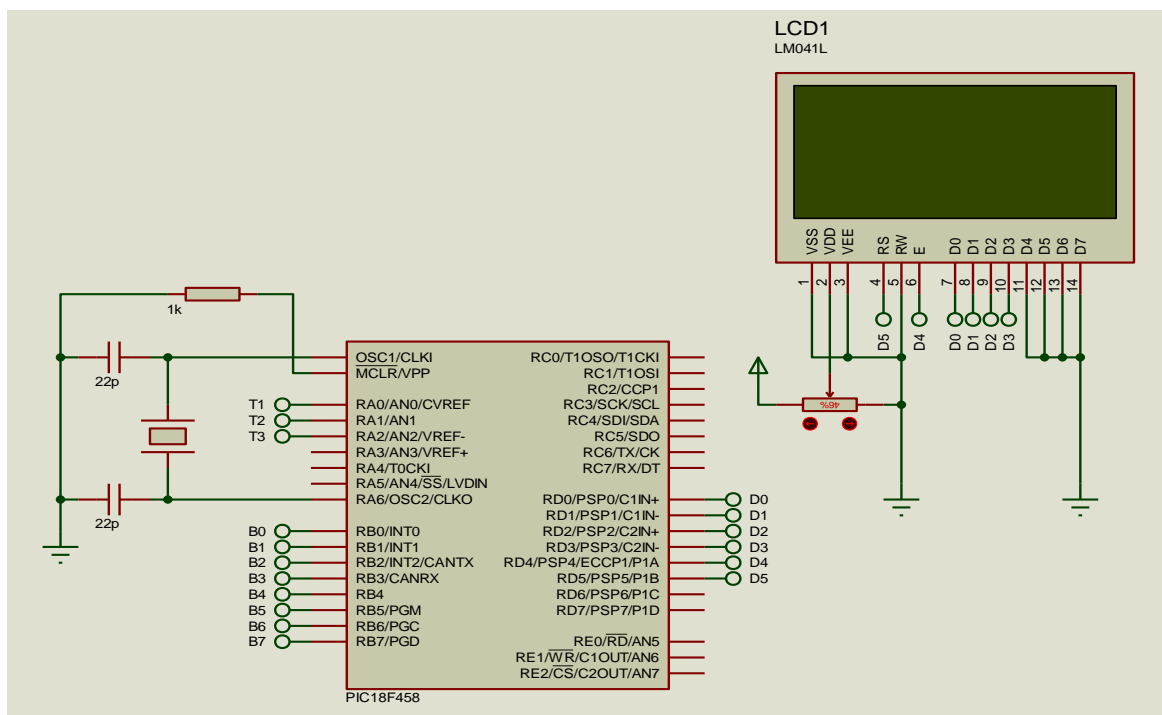
Pour pouvoir afficher les différentes mesures de température issues des trois capteurs ainsi que leur valeur moyenne, nous avons choisi un afficheur LCD (Liquide Cristal Digital) alphanumérique de 4 \*16 caractères.

**Figure III.10 Afficheur LCD 4\*16 caractères**

Les afficheurs à cristaux liquides sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement.

Ils sont relativement bons marchés et s'utilisent avec beaucoup de facilité. Plusieurs afficheurs sont disponibles et ne diffèrent les uns des autres, non seulement par leurs dimensions (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leurs tensions de service [14].

La figure (III.11) représente le schéma de brochage de notre afficheur LCD.



**Figure III.11 Schéma de brochage de l'afficheur LCD**

Nous distinguons les broches suivantes :

- 3 broches d'alimentation et de réglage de contraste ( $V_{SS}$ ,  $V_{DD}$ ,  $V_{EE}$ ).
- 3 lignes de contrôle : RS (Registre Select), R/W (Read,Write) et E (enable) .
- 8 lignes de données ( $D_0$  à  $D_7$ ).

L'afficheur LCD utilisé est le "GDM1604B" compatible qui dispose également d'une mémoire vive DDRAM (Display Data RAM) ou mémoire vive d'affichage de données qui contient les caractères, sa capacité est de 80 caractères dont il suffit d'envoyer le code ASCII du caractère pour obtenir son affichage sur écran.

#### ❖ Fonctionnement de l'afficheur [14]

L'afficheur dispose de deux registres permettant de gérer celui-ci :

### a- Le registre d'instruction IR (Instruction Register)

C'est le registre de contrôle, suivant la valeur que l'on met dedans, l'afficheur exécute des opérations de configurations, exemple : effacement de l'écran. Il permet aussi de positionner le curseur parmi les 80 adresses de la DDRAM.

### b- Le registre des données DR (Data Register)

Suivant la valeur que l'on met dedans l'afficheur peut :

- Afficher un caractère (Code ASCII ou spécifique).
- Créer une ligne d'une matrice d'un nouveau caractère.

#### III.1.2.4 Circuit de conversion parallèle / série [6]

Le HCF4021BE est un circuit intégré de type CMOS « Complementary Metal Oxide Semi-conductor ». Ce dispositif est un registre à décalage de 8 entrées parallèles et une sortie série, dont la cadence de conversion est contrôlée par le signal d'horloge DTR (pin10). La sortie des données est disponible au niveau du pin 3. Dans ce dispositif, l'entrée est commandée par le signal d'entrée pin9. Les données sont décalées de manière sérielle pendant la transition positive du signal. Le schéma de connexion du registre à décalage CMOS4021 au DAC est illustré par la figure (III.12)

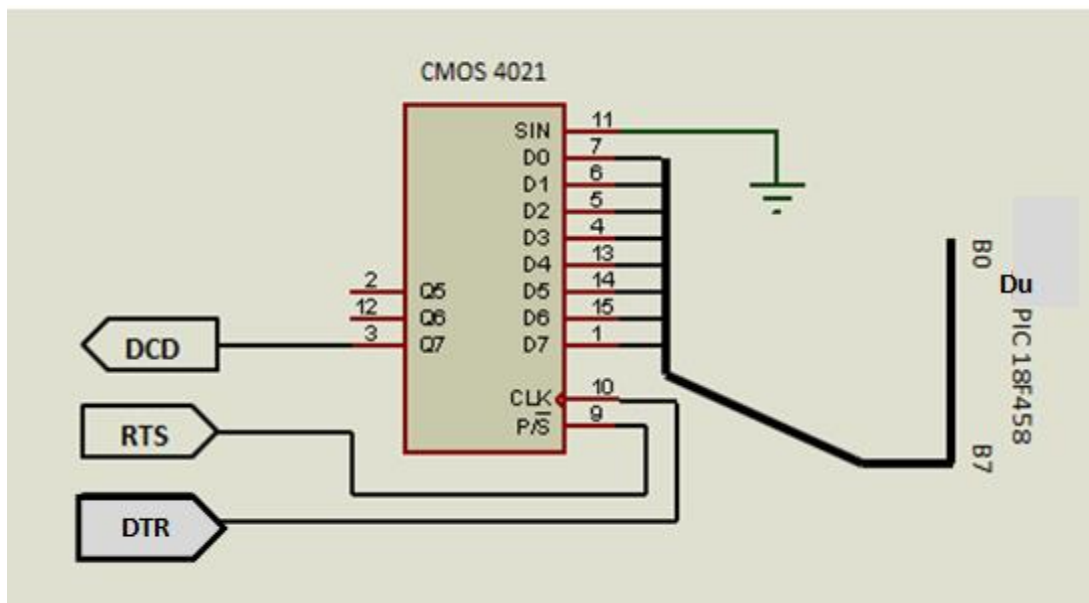


Figure III.12 Schéma de branchement du CMOS4021-ADC

La configuration des pins est comme suite :

- Les entrées parallèles du circuit CMOS4021 sont connectées au port B du PIC.
- Le signal d'horloge (pin10) est engendré par la sortie DTR du port série.

- Le pin3 est connecté avec la sortie DCD (entrée des données séries) du port série.
- Le signal STROBE qui permet le stockage (pin9) est connecté avec la sortie RTS du port série.

### III.1.3 Unité de sortie

L'unité de sortie est une circuiterie électronique qui contient des organes de traitement qui permettent de faire la conversion série/ parallèle, conversion numérique/analogique et enfin une conversion courant/tension dans le but de rendre le signal de commande utile.

#### III.1.3.1 Circuit de conversion série/parallèle

Pour ce type de processus, le signal de commande délivré par le micro-ordinateur est de nature série sur 8 bits, tandis que le convertisseur numérique analogique DAC0800 ne fonctionne qu'avec des signaux parallèles. Donc, L'utilisation du port série pour la commande nécessite en premier lieu l'utilisation des convertisseurs série / parallèle tel que le HCF4094BE.

Le (HCF4094BE) est un registre à décalage a huit étapes sériées (voir figure (III.13)), ayant des unités de stockage liées à chaque étape afin d'enregistrer les données issues du port série, qui se déplacent par la suite vers le bloc de sortie parallèle. [15]

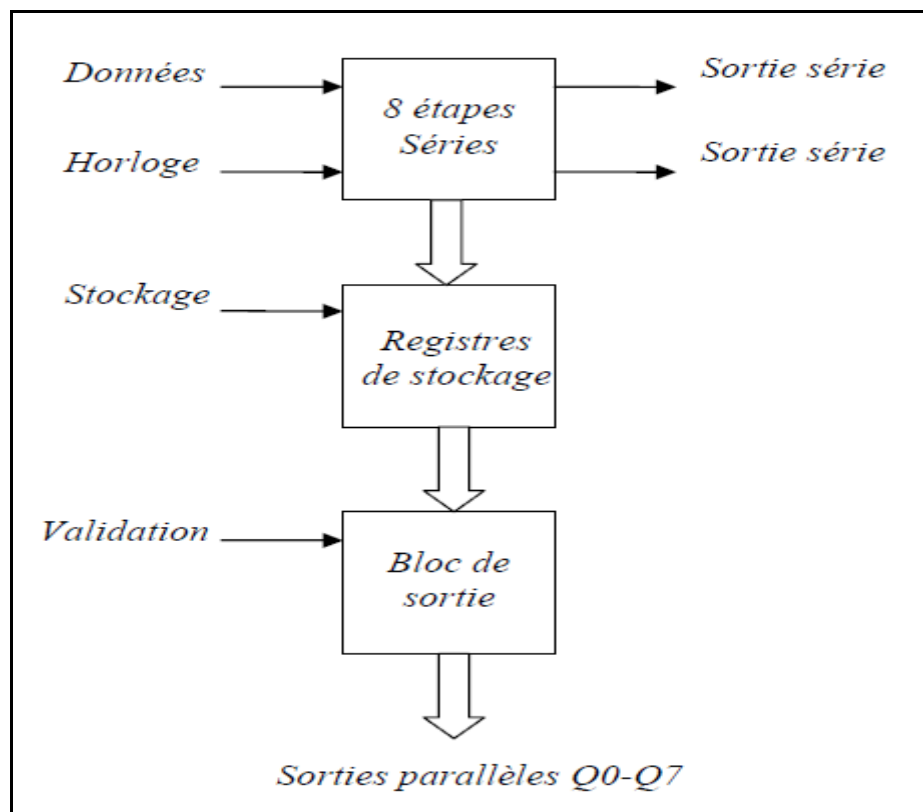


Figure III.13 Schéma bloc de fonctionnement du CMOS4094

Les données issues du port série via la broche TXD sont décalées lorsque le signal d'horloge (pin 3) est de niveau haut, ceci engendre le déplacement des bits vers les éléments de stockage du registre. La validation des sorties est obtenue à l'aide des broches 1 (RTS) et 15 (ENABLE) du registre CMOS4094.

Le HCF4094BE est un circuit intégré monolithique (formé d'un seul bloc) fabriqué suivant la technologie CMOS [16]. Son schéma de brochage est montré à la figure(III.14)

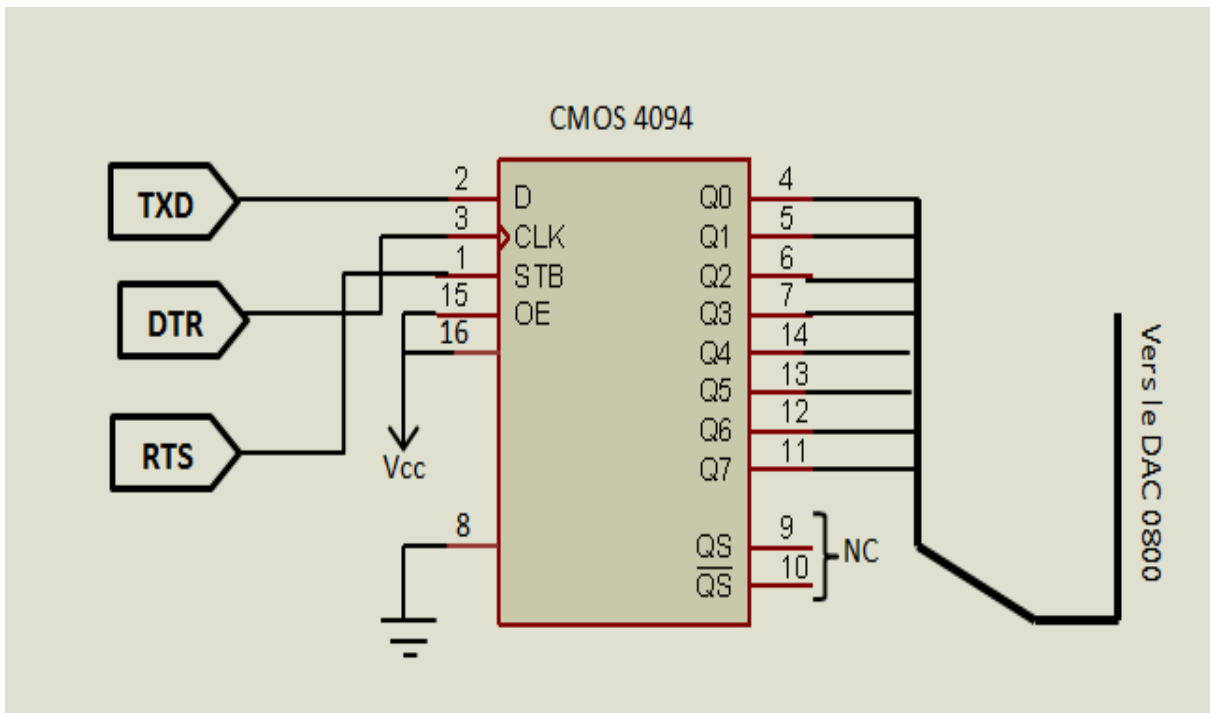


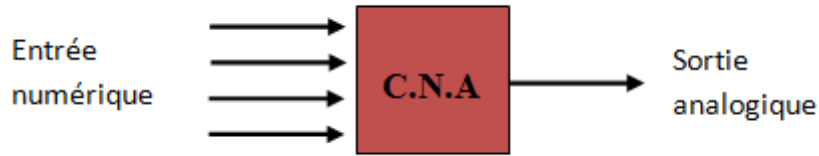
Figure III.14 Schéma de brochage du CMOS4094

Les fonctions des principales broches sont définies de la manière suivante :

- Les sorties parallèles Q0 à Q7 sont reliées directement avec le convertisseur numérique analogique DAC ou la broche 11 correspond au LSB, tandis que la broche 4 représente le MSB.
- Le signal d'horloge (pin3) est contrôlé par la sortie DTR du port série.
- Le pin2 est connecté avec la sortie TXD (sortie des données séries) du port série.
- Le signal qui permet le transfert (pin1) est connecté avec la sortie RTS du port série afin de valider la sortie (STROBE).
- Le signal chip select (pin15) est relié avec le signal Vcc afin d'activer le circuit de manière continue.

### III.1.3.2 Convertisseur numérique analogique DAC 0800

Un convertisseur numérique - analogique permet de traduire une information numérique (binaire) en une information analogique, c'est à dire en une grandeur physique (courant, tension...).



#### III.1.3.2.1 Présentation du DAC0800 [17]

Le DAC0800 est un circuit intégré qui permet de convertir une donnée numérique de 8 bits à une donnée analogique il nécessite des tensions de référence de +10Volts et -10Volts, son schéma de brochage est illustré à la figure (III.16).

#### III.1.3.2.2 Caractéristiques principales

- Temps de conversion : 100 ns
- Erreur en pleine échelle :  $\pm 1$  LSB
- Tension en sortie jusqu'à 20 V
- Sorties complémentaires en courant
- Interfaçage direct en TTL et CMOS
- Alimentation de  $\pm 4,5V$  à  $\pm 18V$
- Basse consommation : 33 mW à  $\pm 5V$
- Conversions unipolaires ou bipolaires
- Coût modéré.

#### III.1.3.2.3 La structure interne du DAC 0800

La structure interne indiquée à la Figure (III.15) montre le bloc de décodage qui reçoit les 8 bits à convertir (broches 5 à 12) et contrôle les commutateurs. Ici les courants entrants dans le convertisseur et leurs poids binaires sont obtenus par des miroirs de courant. Les broches 4 et 2 fournissent les courants complémentaires, image de la conversion. La grandeur de référence, en tension ou en courant, est appliquée au niveau des broches 14 et 15. La notice complète fournit le mode de câblage et de nombreuses applications.

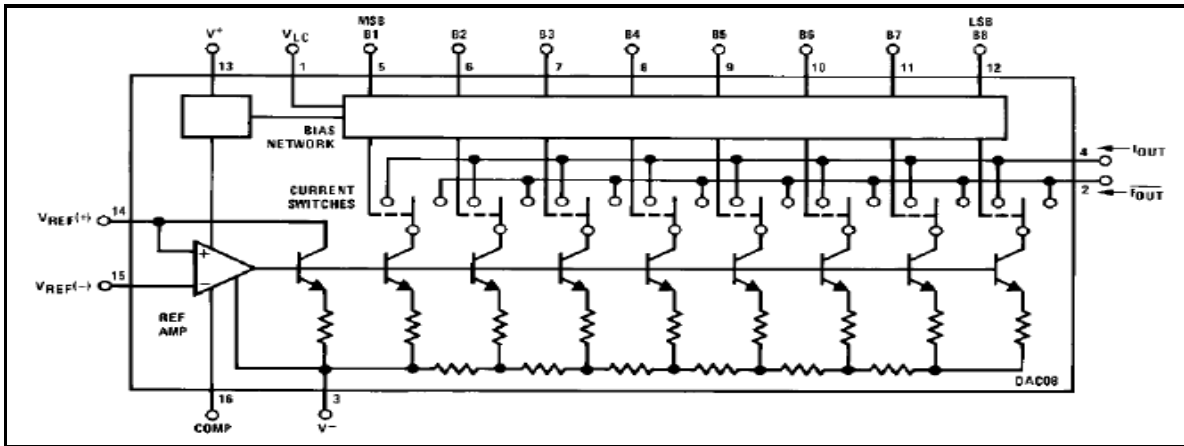


Figure III.15 Architecture interne du DAC0800 (Doc. Motorola)

### III.1.3.3 Le circuit LM741 [16]

Comme la sortie du convertisseur numérique/analogique est une sortie différentielle en courant alors il est nécessaire d'utiliser un amplificateur opérationnel LM741 monté en convertisseur courant voltage.

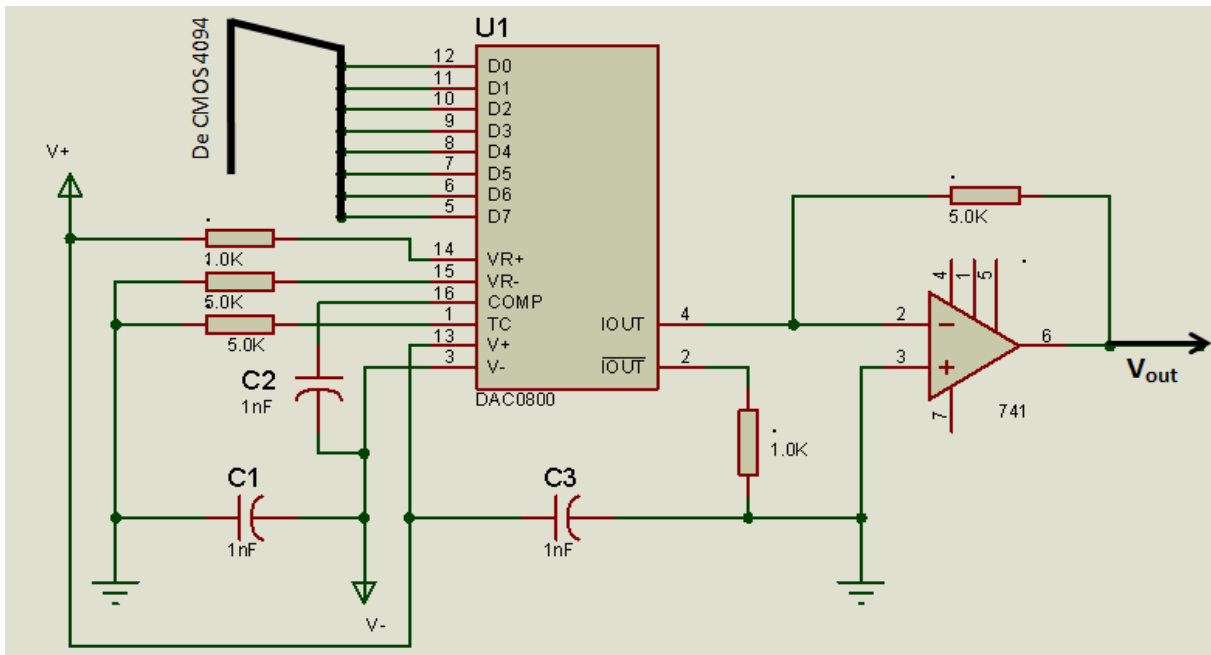


Figure III.16 Schéma de brochage du DAC0800 avec LM741

### III.2 Le schéma général de la carte d'interface

La figure (III.17) représente la configuration générale de la carte d'interface en utilisant ISIS Proteus et la figure (III.18) représente notre carte électronique d'interface réalisée sur une plaque d'essai au niveau de laboratoire.

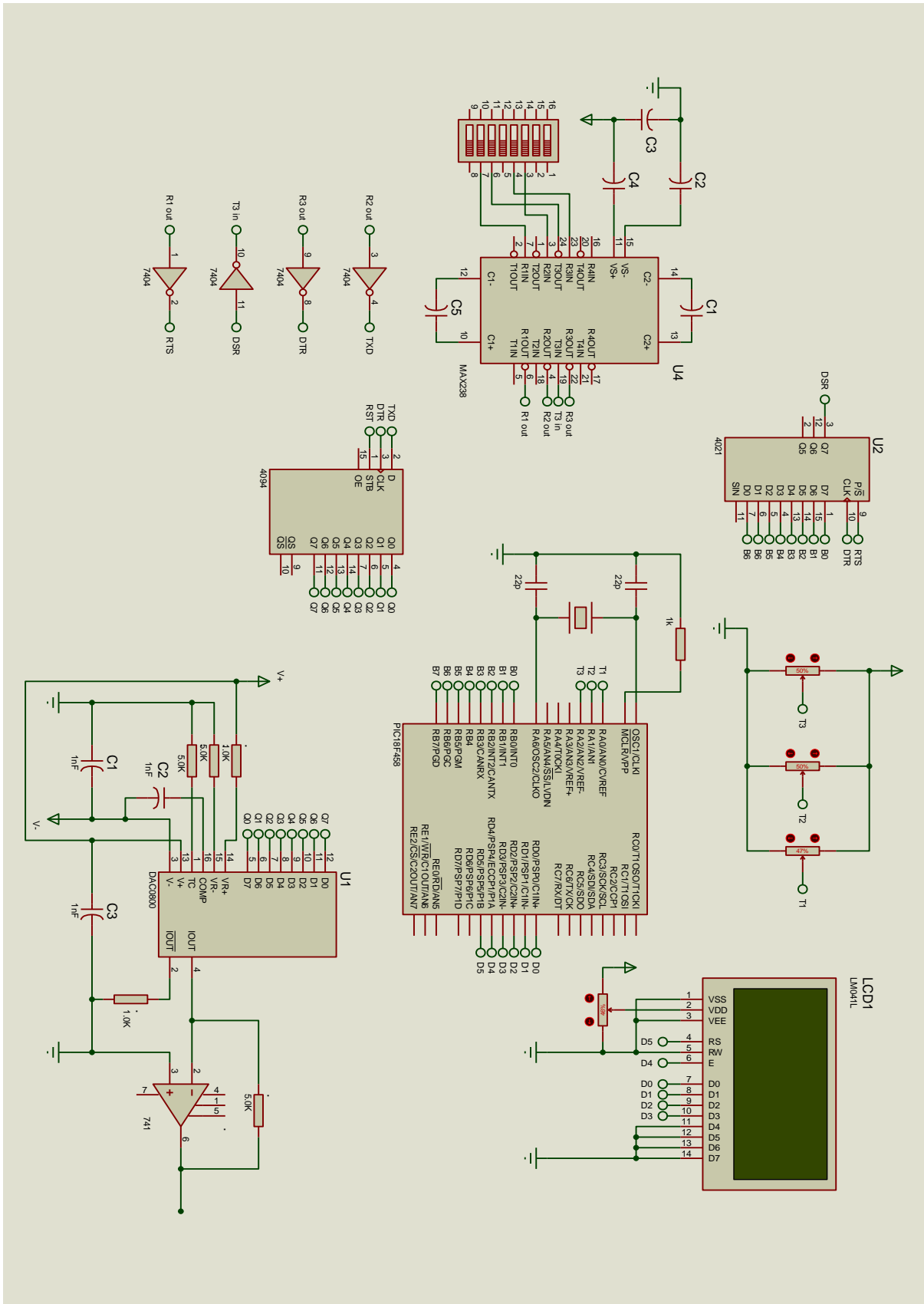
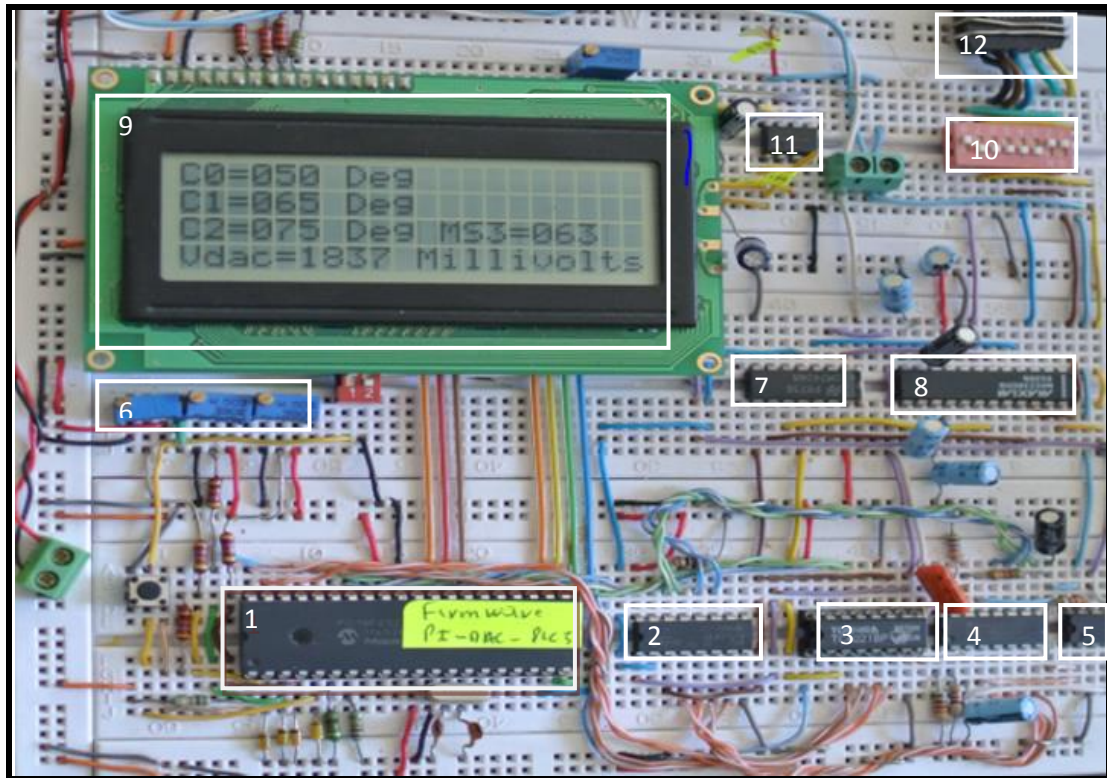


Figure III.17 Le schéma général de la carte d'interface





Circuit intégré 01 : PIC18f458

Circuit intégré 02 : CMOS4094

Circuit intégré 03 : CMOS4021

Circuit intégré 04 : DAC0804

Circuit intégré 05 : LM741 (Conversion  
courant/tension)

Bloc 06 : Potentiomètres de test

Circuit intégré 07 : 7404DM

Circuit intégré 08 : MAX238

Bloc 09 : Afficheur Alphanumérique

Bloc 10 : Switch

Circuit intégré 11 : LM741 (Filtre)

Bloc12 : Connecteur DB9

**Figure III.18** La carte d'interface réalisée sur une plaque d'essai

### III.3 Présentation du port de communication du PC [6]

Les ports séries (également appelés RS-232, nom de la norme à laquelle ils font référence) représentent les premiers modèles d'interfaces des PC. Le terme série désigne un envoi de données via un fil unique (les bits sont envoyés les uns à la suite des autres) selon le protocole nommé RS232. La communication série se fait de façon asynchrone, cela signifie qu'aucun signal de synchronisation (appelé horloge) n'est nécessaire, donc les données peuvent être envoyées à intervalle de temps arbitraire. En contrepartie, le périphérique doit être capable de distinguer les caractères (un caractère à une longueur de 8 bits) parmi la suite de bits qui lui est envoyée.

C'est la raison pour laquelle dans ce type de transmission, chaque caractère est précédé d'un bit de début (appelé bit START) et d'un bit de fin (bit STOP). Ces bits de

contrôle, nécessaires pour une transmission série, gaspillent 20% de la bande passante (pour 10 bits envoyés, 8 servent à coder le caractère, 2 servent à assurer la réception).

Nous avons choisi ce type de port, compte tenu de son avantage par rapport au port parallèle du PC car la portée du signal issue du port « COM » est d'environ 15 mètres donc beaucoup plus par rapport au port parallèle.

En effet le port série permet une communication entre l'ordinateur et le procédé pour des distances peuvent atteindre 20 mètres environ. Cependant la vitesse de transmission maximum est d'environ 115000 bits/seconde. Les ports séries sont généralement intégrés dans la carte mère, c'est pourquoi des connecteurs présents à l'arrière du boîtier, et reliés à la carte mère par une nappe de fils, permettent de connecter un élément extérieur. Les connecteurs séries possèdent généralement 9 broches et se présentent sous la forme suivante (figure (III.19))

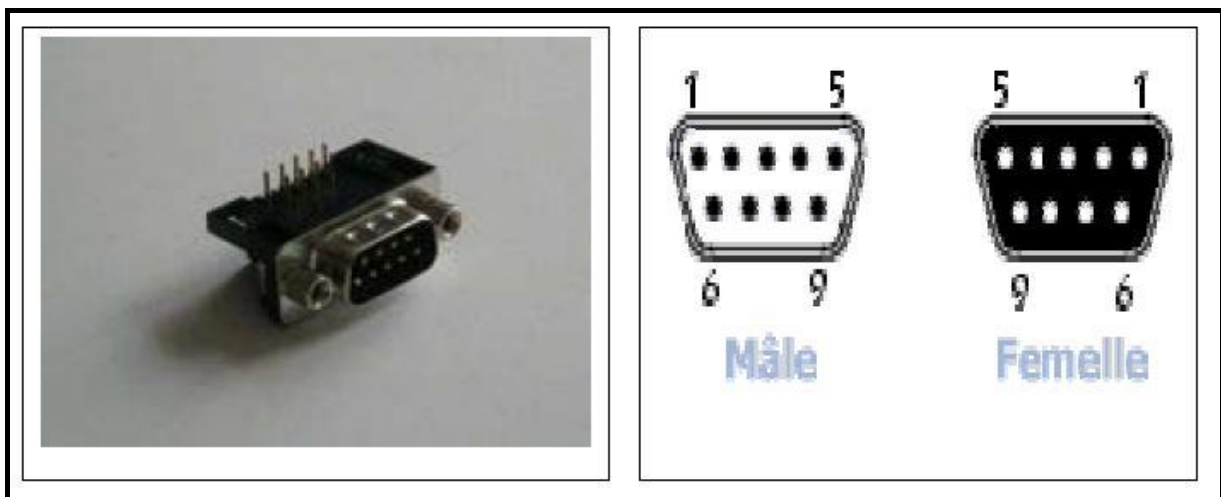


Figure III.19 Connecteur DB9 mâle et femelle

La répartition des broches est donnée selon le tableau (III.7)

Tableau III.7 Répartition des différents signaux du connecteur DB9

Broches	Signal	désignation	Description	signal
1	Data Carrier Detect	<b>DCD</b>	Détection de porteuse	<b>Entrée</b>
2	Receive Data	<b>RD</b>	Réception de données	<b>Entrée</b>
3	Transmit Data	<b>TXD</b>	Emission de données	<b>Sortie</b>
4	Data Terminal ready	<b>DTR</b>	Terminal de données prêt	<b>Sortie</b>
5	Signal Ground	<b>SG</b>	Masse de signal	<b>GND</b>
6	Data Set Ready	<b>DSR</b>	Données prêtes	<b>Entrée</b>
7	Request To Send	<b>RTS</b>	Requête d'émission	<b>Sortie</b>
8	Clear To Send	<b>CTS</b>	Prêt pour l'émission	<b>Entrée</b>
9	Ring Indicator	<b>RI</b>	Indicateur d'appel	<b>Entrée</b>

La répartition des signaux par rapport à la carte électronique d'interface s'effectue selon le tableau (III.8) ci-dessous

**Tableau III.8 Répartition des différents signaux utilisés pour la carte d'interface**

<b>Désignation broche</b>	<b>Fonction au niveau de la carte</b>
DCD	<b>Lecture température via ADC</b>
TXD	<b>Consigne température via DAC</b>
DTR	<b>Signal d'horloge pour conversion série - parallèle</b>
RTS	<b>Validation registres CMOS 4094 et 4021</b>
GND	<b>Masse digitale</b>

### **CONCLUSION :**

Dans ce chapitre nous avons étudié et réalisé la carte électronique utilisée pour le contrôle du four électrique du laboratoire. Cette carte électronique d'interface est constituée d'un microcontrôleur qui permet d'assurer l'acquisition, la conversion analogique/numérique et aussi commander l'afficheur LCD, un convertisseur numérique/analogique associés à des registres à décalage ainsi que les circuits d'adaptation.

Cependant, au préalable nous avons testé et vérifié individuellement l'ensemble de circuits intégrés constituant cette carte d'interface. Cet essai nous a permis de confirmer le bon fonctionnement et la fiabilité de la carte électronique avant son couplage au port de micro-ordinateur.

# *Chapitre IV*

*Implémentation du programme  
de commande du four  
électrique*

## INTRODUCTION

Après avoir étudié et réalisé la carte d'interface électronique, assurant la communication entre le procédé à commander et l'ordinateur, nous intéresserons dans ce chapitre à la réalisation de la boucle de régulation afin d'améliorer les performances de fonctionnement de notre système à commander ainsi l'élaboration du programme software de contrôle en utilisant deux langages de programmation Visual C# 2010 et le Mikro C, les interfaces graphiques, les graphes résultants et les interprétations des résultats.

Nous étudions deux types de commande avec feedback ou nous proposons un réglage non linéaire :

- Commande avec contrôleur P avec un élément non linéaire.
- Commande avec contrôleur PI avec un élément non linéaire.

### IV.1 La commande en boucle fermée [18]

La régulation est un ensemble de théories mathématiques et une technique de raisonnement qui concerne la prise de décision et la commande des systèmes.

L'idée serait que le signal d'erreur soit nul, à tout instant. Cette condition n'est pas réalisée en général, du fait de l'une au moins des circonstances suivantes :

- La présence de perturbations qui affectent le processus ; le maintien de la grandeur de sortie à la valeur de consigne constitue un problème de régulation.
- Les variations de la consigne : il s'agit alors d'un problème de suivi.

Le système de régulation automatique possède deux chaînes illustrées en figure (IV.1)

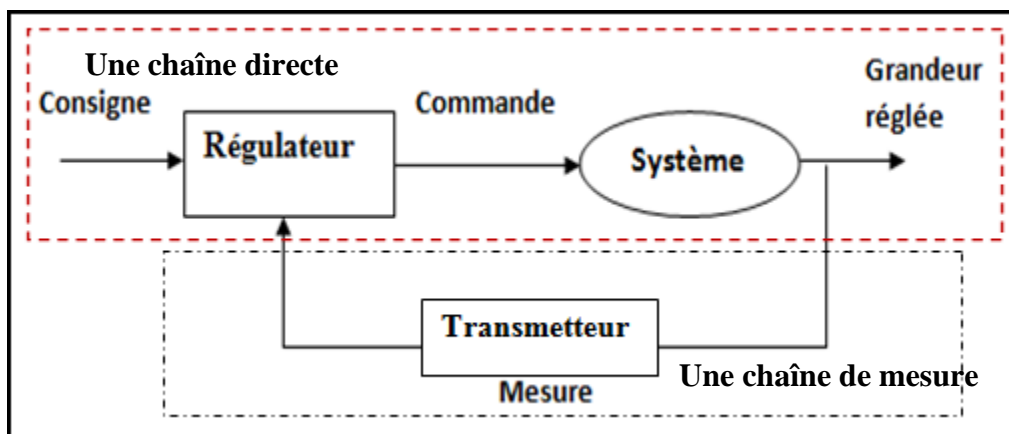


Figure IV.1 Les chaînes d'un système de régulation automatique

### IV.1.1 Intérêt de la régulation [18]

Les systèmes automatiques permettent avant tout de réaliser des opérations qui ne peuvent pas être confiées à l'homme, pour différentes raisons. Parmi celles-ci :

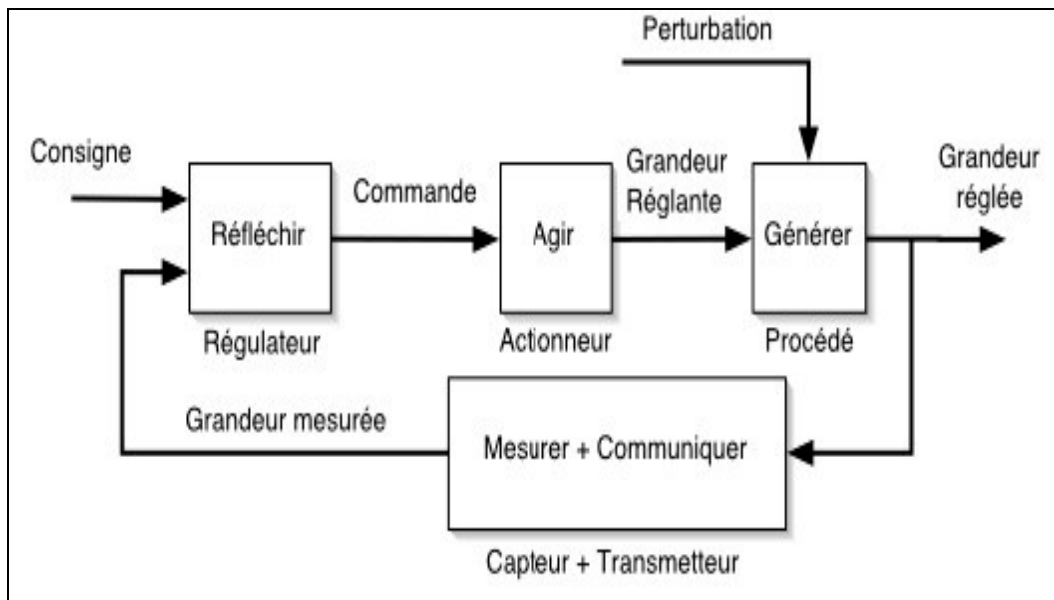
- La précision
- La répétitivité
- La recherche d'une diminution Coûts par l'augmentation de rendements
- La recherche de performances élevées (rapidité des réponses, régularité des produits, coût, qualité,..)

### IV.1.2 Principe de la régulation [18]

Pour réguler un système physique, il faut :

- Mesurer la grandeur réglée avec un capteur.
- Réfléchir sur l'attitude à suivre. Le régulateur compare la grandeur réglée avec la consigne et élabore le signal de commande.
- Agir sur la grandeur réglante par l'intermédiaire d'un organe de réglage.

Le schéma bloc d'un système de commande (Système de régulation) est donné par la figure (IV.2).



**Figure IV.2 Schéma bloc d'un système de régulation**

Pour la régulation de température de notre procédé on applique une commande numérique, le schéma fonctionnel est illustré en figure (IV.3).

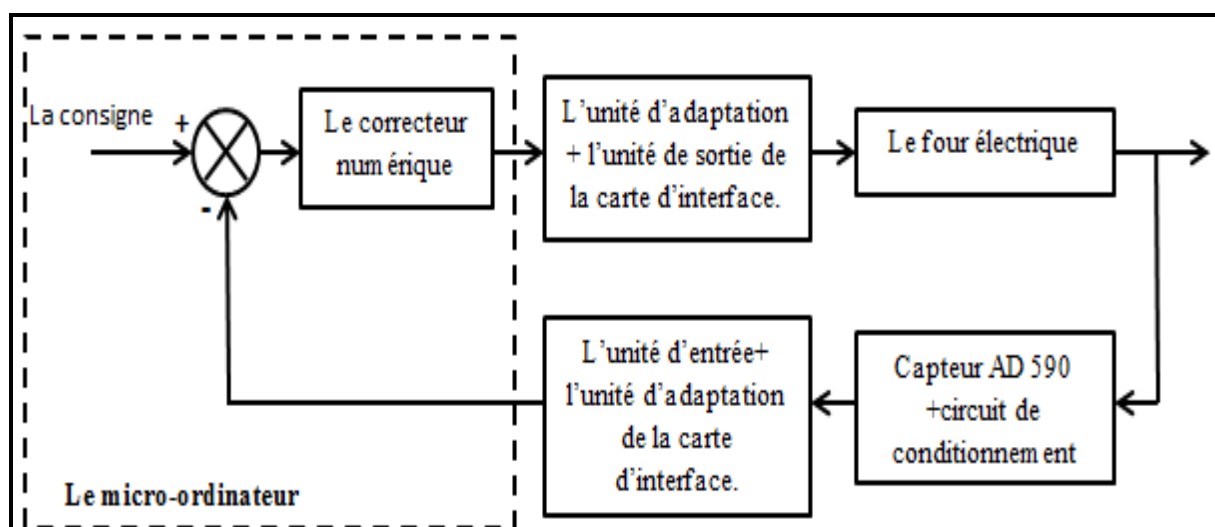


Figure IV.3 Schéma bloc de notre système de régulation numérique

La figure (IV.4) ci-dessous représente une photo réelle de système de régulation en boucle fermée

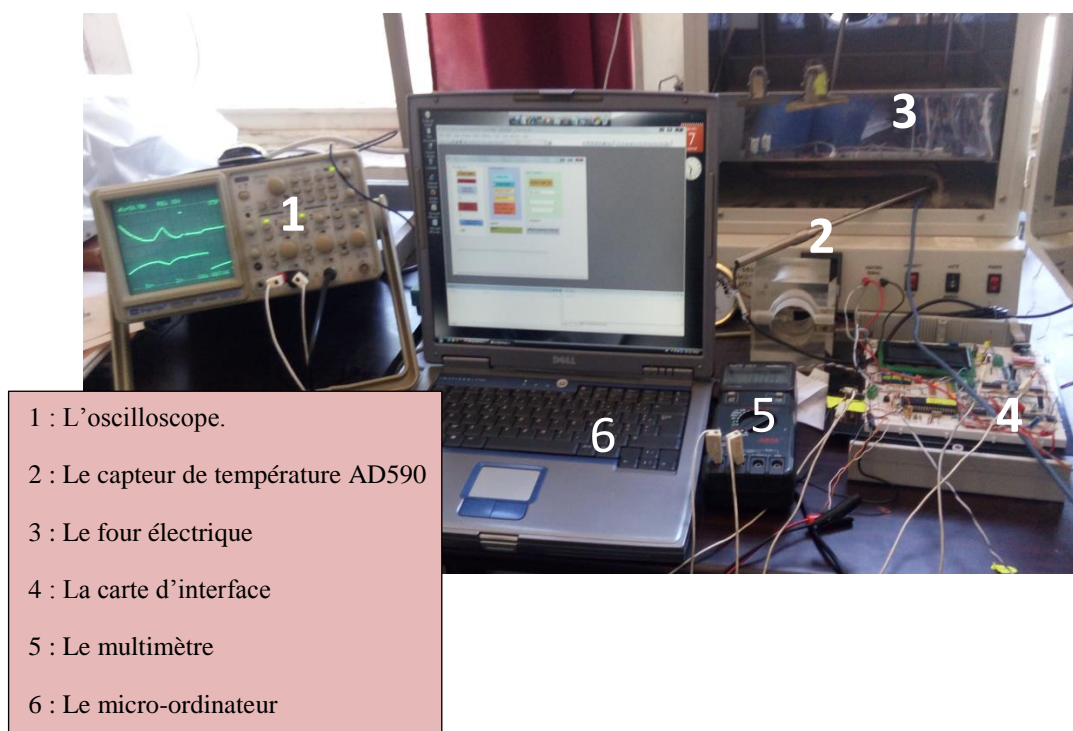


Figure IV.4 Photo réelle de système de régulation en boucle fermée

### IV.1.3 Le choix de la forme de la fonction de transfert du régulateur [18]

Ce choix s'effectue en tenant compte du critère de performances chiffrant la précision dynamique et statique que l'on veut obtenir.

Pour notre cas nous envisageons, les régulateurs numériques P et PI, pour lesquels nous associons une caractéristique non linéaire afin d'éviter la saturation de l'unité de conversion numérique /analogique (DAC) avec son amplificateur, donc ces régulateurs fonctionnent d'une façon non linéaire.

#### IV.1.3.1 Régulateur à action proportionnelle (P)

Ce correcteur élémentaire est le correcteur de base, il agit principalement sur le gain du système de régulation et permet donc d'améliorer notablement la rapidité. Mais le système répond avec un écart statique, la figure (IV.4) illustre la structure d'un régulateur non linéaire proportionnel.

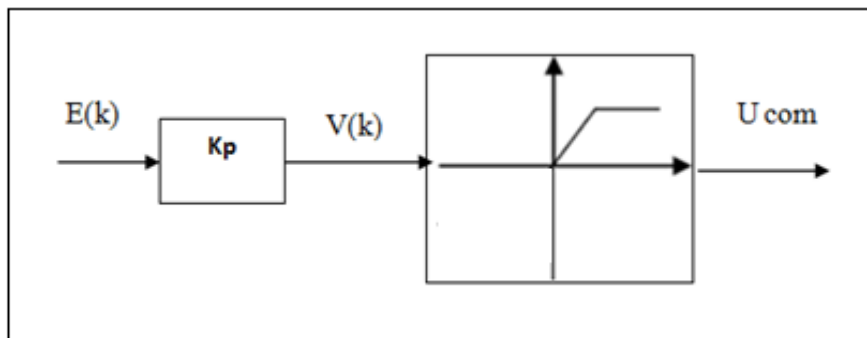


Figure IV.5 Structure d'un régulateur proportionnel non linéaire

La forme de la fonction de transfert du régulateur P continu est donnée par l'équation (IV.1) suivante :

$$C(s) = Kp \quad (\text{IV.1})$$

$s$  : correspond à l'opérateur de Laplace des fonctions de transfert par les systèmes continus.

La loi de commande du régulateur P numérique peut être obtenue par discrétisation de celle d'un régulateur P analogique. Nous obtenons donc la fonction de transfert discrète suivante :

$$C(z) = Kp \quad (\text{IV.2})$$

Où

$z$  : est l'opérateur des fonctions de transfert des systèmes échantillonnés et discrets.

L'algorithme de ce correcteur est donné par l'expression récurrente suivante :

$$U(k) = Kp \times e(k) \quad (\text{IV.3})$$



Puisque il y a l'existence d'un bloc de non linéarité, le signal de commande est comme suit :

$$\begin{aligned}
 U_{com} &= K_p \times e(k) && \text{Si } U_{min} < K_p \times e(k) < U_{max} \\
 U_{com} &= U_{max} = 10V && \text{Si } K_p \times e(k) > U_{max} \\
 U_{com} &= U_{min} = 0V && \text{Si } K_p \times e(k) < U_{min}
 \end{aligned}$$

➤ **L'effet d'un régulateur P dans une chaîne de régulation**

**Faible gain  $K_p$**  : Si cette action est faible, on a une réponse lente et molle assortie d'une grande erreur statique.

**Fort gain  $K_p$**  : Si cette action est importante, on obtient une réponse rapide mais avec risque de dépassement et d'oscillation. Par contre, l'erreur statique est plus faible.

#### IV.1.3.2 Régulateur à action proportionnelle et intégrale (PI)

Dans le cas de la commande d'un système par action proportionnelle seule, on constate un écart statique en régime établi. Pour éliminer celui-ci, on utilise en complément une action intégrale qui tient compte de l'écart entre la mesure et la consigne. La valeur de l'action intégrale s'exprime en unité de temps (généralement en seconde).

La Figure (IV.5) illustre la structure d'un régulateur PI non linéaire.

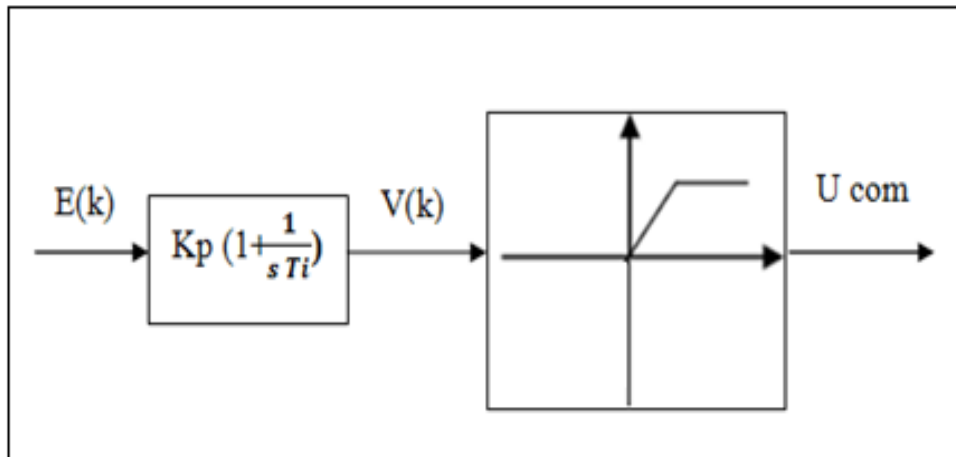


Figure IV.6 Structure d'un régulateur proportionnel et intégrale non linéaire

La forme standard de la fonction de transfert du régulateur PI continu est donnée par la relation suivante :

$$C(s) = K_p \left( 1 + \frac{1}{sT_i} \right) \quad (\text{IV.4})$$

$s$  : correspond à l'opérateur de Laplace des fonctions de transfert par les systèmes continus

La loi de commande du régulateur PI numérique peut être obtenue par discrétisation de celle d'un régulateur PI analogique. Nous obtenons donc la fonction de transfert discrète suivante :

$$C(z) = K_p + K_i \frac{z}{z-1} \quad (\text{IV.5})$$

Où :

$z$  : est l'opérateur des fonctions de transfert des systèmes échantillonnés et discrets.

L'algorithme de ce correcteur est donné par l'expression récurrente suivante :

$$U(k) = K_p e(k) + K_i \sum_{i=0}^k e(i) \quad (\text{IV.6})$$

L'implémentation de la composante intégrale du contrôleur est effectuée de manière **réursive**.

$$\text{Soit : } U(k+1) = U(k) + k_i \times e(k+1) \quad (\text{IV.7})$$

$T_i$  : est la constante du temps d'intégration.

## IV.2 Présentation du software

### IV.2.1 Le software d'acquisition et d'affichage des données

#### IV.2.1.1 Présentation du MikroC [19]

MikroC est un outil puissant, développé, et riche en options, utilisée pour la programmation des microcontrôleurs. L'objectif est de fournir au utilisateur les plus simples moyens et solutions afin de développer des applications pour des systèmes embarqués sans compromettre les performances et le contrôle.

Le MikroC offre un (IDE) très avancé, large ensemble de bibliothèques de matériel, la documentation complète et beaucoup de fonction prête à l'emploi.

Avec des outils utiles mis en œuvre, de nombreux exemples de code pratique, ensemble des routines intégrées, et une aide complète, le MikroC est un outil rapide et fiable. Le MikroC permet de développer et déployer des applications complexes.

- Ecrire un code source C en utilisant le très avancé éditeur de code.
- Utiliser les bibliothèques du MikroC fait accélérer considérablement le développement (acquisition de données, la mémoire, affichage, les conversions, les communications.....)
- Surveiller la structure du programme variables et fonctions dans l'explorateur de code.
- Inspecter le déroulement du programme et de déboguer la logique d'exécution avec l'intégration Debugger,

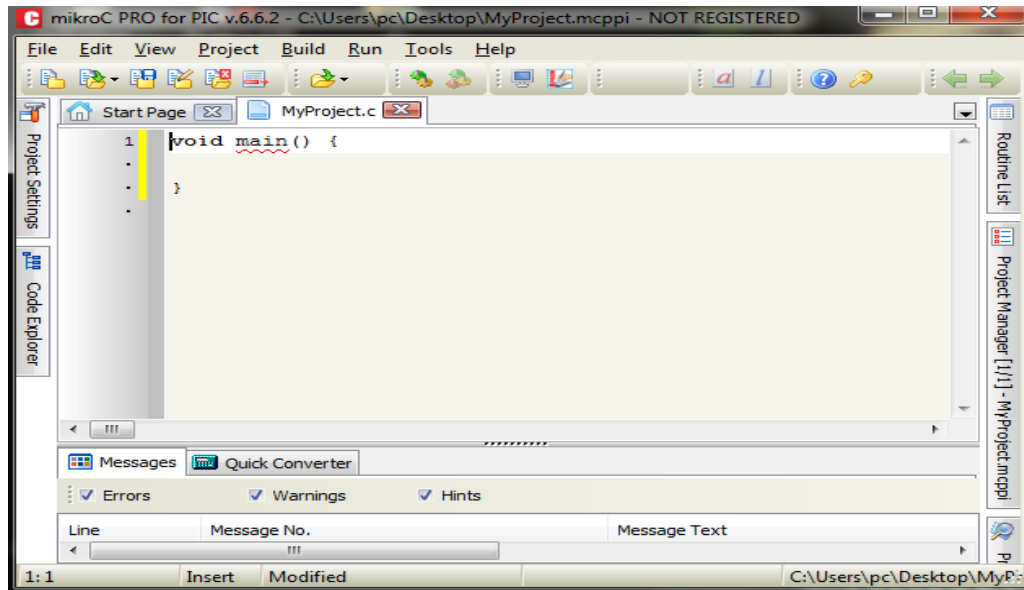


Figure IV.7 L'interface de travail du logiciel MikroC

#### IV.2.1.2 L'affichage sur LCD

##### ➤ La configuration de LCD

L'afficheur LCD est un élément important dans notre carte, il joue le rôle d'une HMI (Human Machine Interface), donc c'est notre unité d'affichage et de visualisation, il existe deux types de configuration pour le piloter, une avec huit lignes et autre avec quatre seulement.

Pour notre LCD nous avons utilisé celle de quatre lignes qui est la configuration la plus simple avec l'instruction suivante

\* **void** Lcd\_Init(**unsigned short** \*port) ;

D0	→	RD0
D1	→	RD1
D2	→	RD2
D3	→	RD3
E	→	RD4
RS	→	RD5
RW	→	GND

Nous utilisons le mode 4 bits pour les lignes de données soit au totale 7 lignes comme l'indique les schémas que nous avons élaboré en chapitre II.

##### ➤ Les instructions d'affichage sur LCD [22]

L'affichage du caractère ou de texte est obtenu avec la commande

\*void Lcd\_Out(unsigned short row)

Les instructions les plus importantes du C sont résumées au niveau du tableau IV.1 ci-dessous.

**Tableau IV.1 Le résumé des instructions les plus importantes du C**

LCD commandes	L'objectif
LCD_FIRST_ROW	Déplacer le curseur à la 1 ère ligne
LCD_SECOND_ROW	Déplacer le curseur à la 2 ème ligne
LCD_THIRD_ROW	Déplacer le curseur à la 3 ème ligne
LCD_FOURTH_ROW	Déplacer le curseur à la 4 ème ligne
LCD_CLEAR	Supprimer tous ce qui est affiché sur LCD
LCD_RETURN_HOME	Retourner le curseur dans sa position initiale, Display Data RAM ne sera pas modifié.
LCD_CURSOR_OFF	Désactiver le curseur
LCD_UNDERLINE_ON	Souligner au dessous du curseur
LCD_BLINK_CURSOR_ON	Activer le clignement du curseur
LCD_MOVE_CURSOR_LEFT	Déplacer le curseur vers la gauche
LCD_MOVE_CURSOR_RIGHT	Déplacer le curseur vers la droite
LCD_TURN_ON	Alimenter le LCD
LCD_TURN_OFF	Désalimenter le LCD
LCD_SHIFT_LEFT	Déplacer l'affichage sans le changer vers la gauche
LCD_SHIFT_RIGHT	Déplacer l'affichage sans le changer vers la droite

### IV.2.1.3 La conversion analogique \_ digitale (ADC)

Comme il est indiqué dans le chapitre précédent, le PIC18F458 contient une unité de conversion analogique digitale, le pilotage de la conversion et la configuration des ports comme entrées analogiques se fait par les deux registres ADCON0 et ADCON1, la conversion se fait sur 10 bits, le résultat sera stocké dans l'un des registres suivants : ADRESH, ou ADRESL qui se situent dans la RAM du PIC. La figure (IV.8) montre l'organigramme de programme d'acquisition et d'affichage sous MikroC.

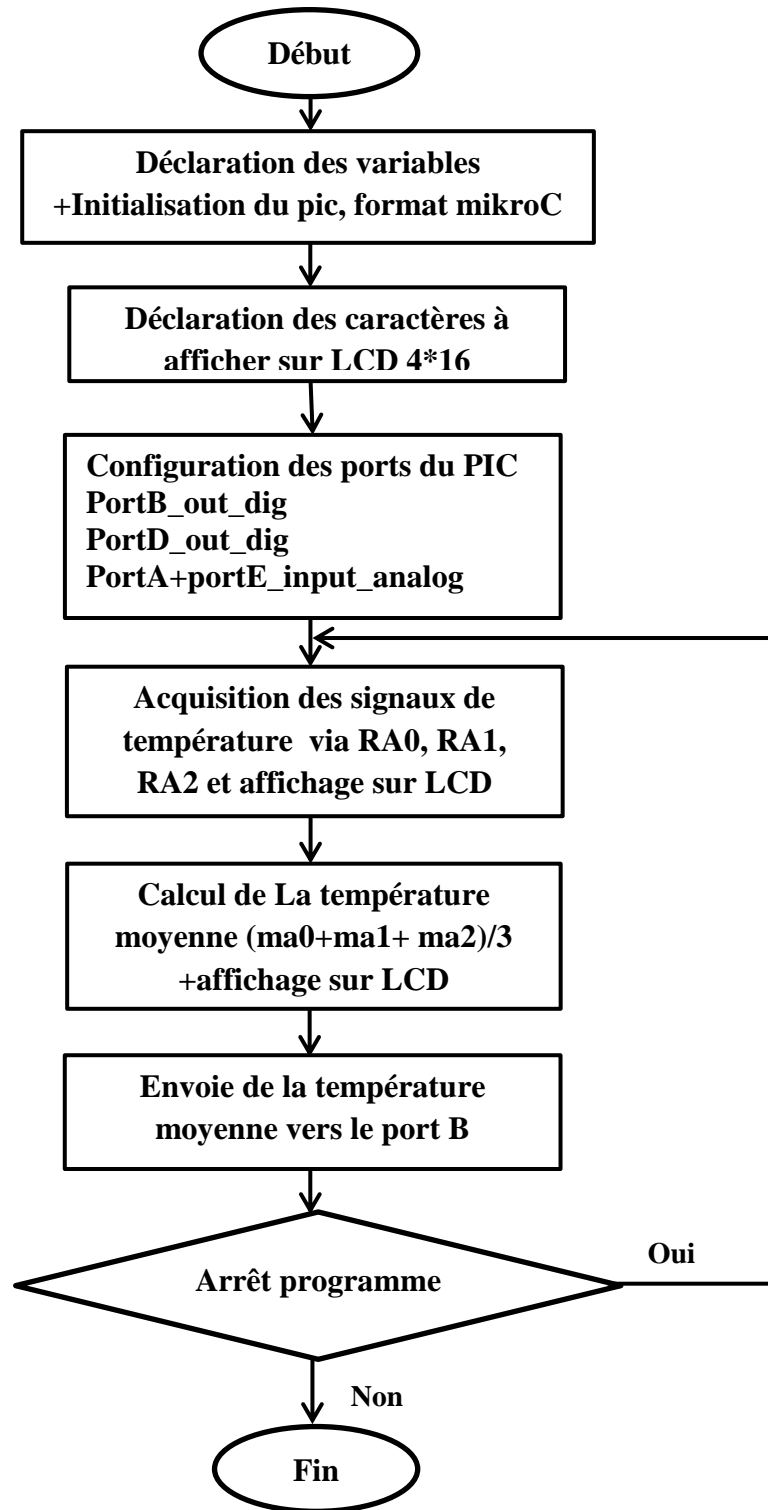


Figure IV.8 L'organigramme de programme d'acquisition et d'affichage

#### IV.2.1.4 Le logiciel de simulation ISIS PROTEUS [20]

Après avoir fixé l'idée du projet et l'élaboration du programme, on a commencé à fixer le design de notre circuit de commande. Les outils de conception de circuit offrent des automatismes dans le placement des composants et le routage des circuits, le PROTEUS ISIS

est un logiciel de simulation et de conception des cartes électroniques, la force de son architecture est de permettre la simulation par graphes et la simulation interactive des processus, il est possible de simuler un circuit basé sur des microprocesseurs et de le tester de manière interactive, de plus ISIS regroupe des fonctionnalités utilisées à la création d'un PCB afin qu'un même projet puisse être exporté vers un autre logiciel de création des circuits imprimés. Son environnement en figure (IV.9).

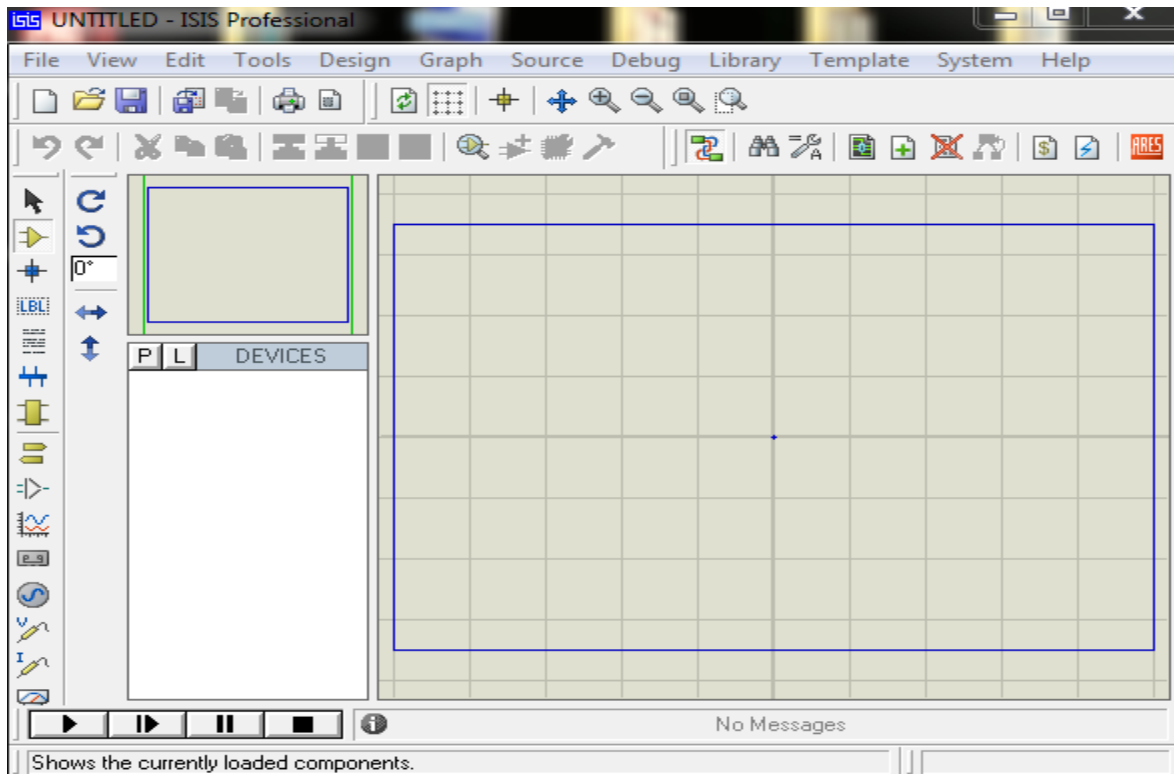


Figure IV.9 L'interface de travail de PROTEUS ISIS

#### IV.2.1.5 Le logiciel PICKkit 2 [21]

Le transfert du programme de MikroC vers le microcontrôleur de MICROSHIP est fait à partir d'un outil appelé programmeur qui est présenté en figure (IV.10) par l'intermédiaire du logiciel PICKkit 2.

##### ➤ Les étapes de branchement

1. Placement du PIC sur le programmeur ;
2. Branchement de programmeur au port USB du PC ;
3. lancement du PICKkit 2 ;
4. Changement du programme.hex :Import hex, et le code apparaît dans program memory ;
5. Cliquer sur « Write » pour lancer la programmation.

L'interface de travail du logiciel PICKit 2 nous permet de voir le PIC utilisé, la zone de mémoire programme et la zone mémoire EEPROM (voir la figure (IV.10)).

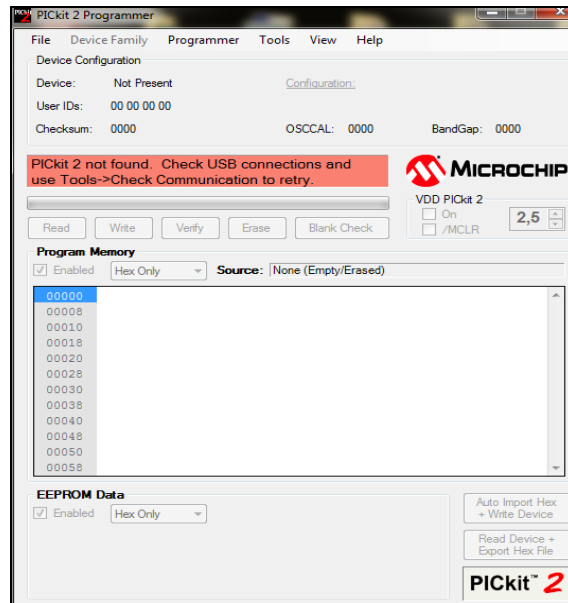


Figure IV.10 L'interface de travail du logiciel PICKit2

## IV.2.2 Programmation et réalisation des interfaces graphiques

### IV.2.2.1 Présentation du compilateur Visual C #2010 [22]

Le Visual C#2010 version 12.0, est un produit de Microsoft Visual Studio Express, qui est un ensemble d'environnements de développement intégrés gratuits développé par Microsoft.

Le Visual C#2010 est un langage de programmation conçu pour la création d'une large gamme d'applications qui s'exécutent sur le .NET Framework. C# est simple, puissant, de type sécurisé et orienté objet. Avec ses nombreuses innovations, Visual C# permet le développement rapide d'applications tout en conservant la simplicité et l'élégance des langages de style C.

Le Visual C# avec un éditeur de code complet, un compilateur, contenu des modèles de projet, des concepteurs, des Assistants Code, un débogueur puissant et simple d'utilisation, ainsi que d'autres outils. La bibliothèque de classes +NET Framework donne accès à de nombreux services de système d'exploitation et d'autres classes utiles, bien conçues qui permettent d'accélérer considérablement le cycle de développement. C'est un langage de programmation orienté objet qui ait la puissance de calcul avec la facilité d'utilisation de l'interface graphique visuelle.

### IV.2.2.2 Les interfaces graphiques

Ces interfaces graphique nous permettent de

- Implémentation des paramètres du régulateur, la période d'échantillonnage et la consigne de température avec la validation de chaque valeur.
- Ouverture du port de communication COM à l'aide de l'objet (Radio Button).
- Exécution du programme (le Timer en mode (ON)) par le bouton «Start timer »
- La possibilité d'arrêter la commande (le Timer en mode (OFF)) à l'aide du bouton « Stop timer ».
- Affichage de la température mesurée en numérique et en analogique ainsi que l'écart en valeur analogique à l'aide des objets (Label).

#### ➤ L'interface graphique (E/S) pour un régulateur P

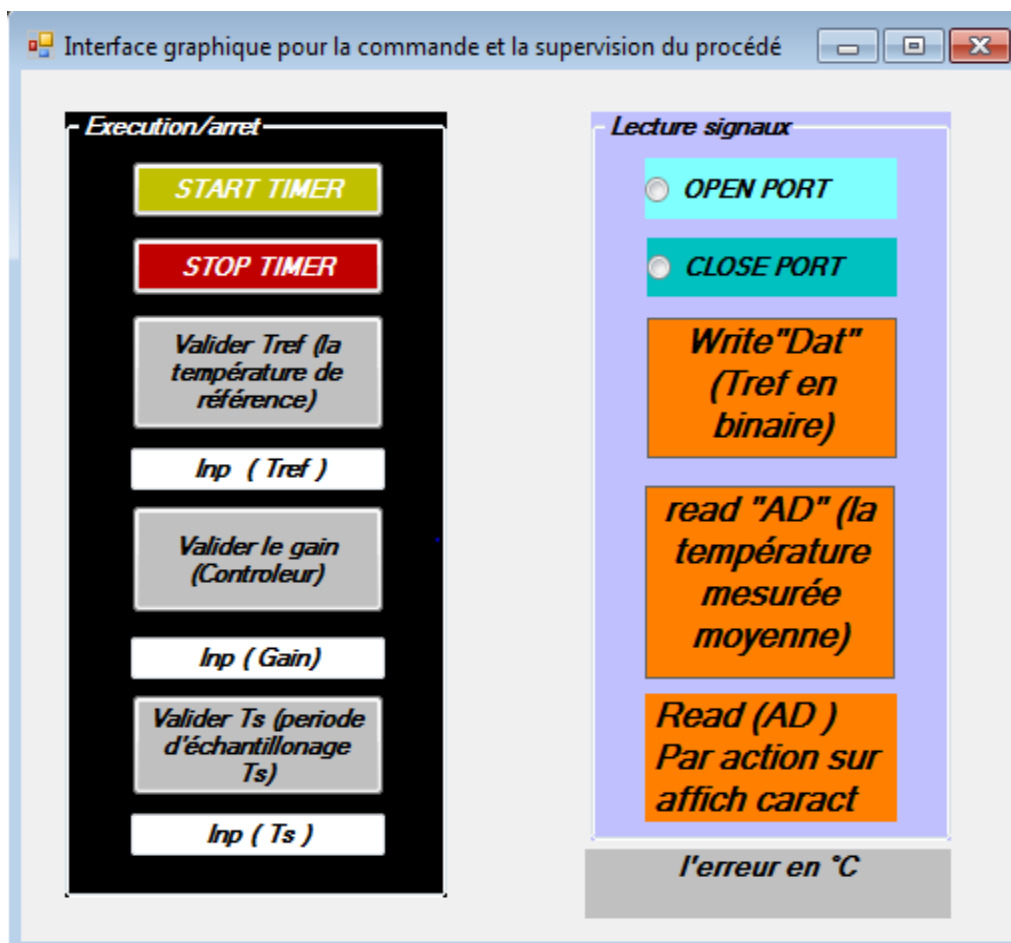


Figure IV.11 L'interface graphique (E/S) contrôle de température avec un régulateur P non linéaire



➤ L'interface graphique (E/S) pour un régulateur PI

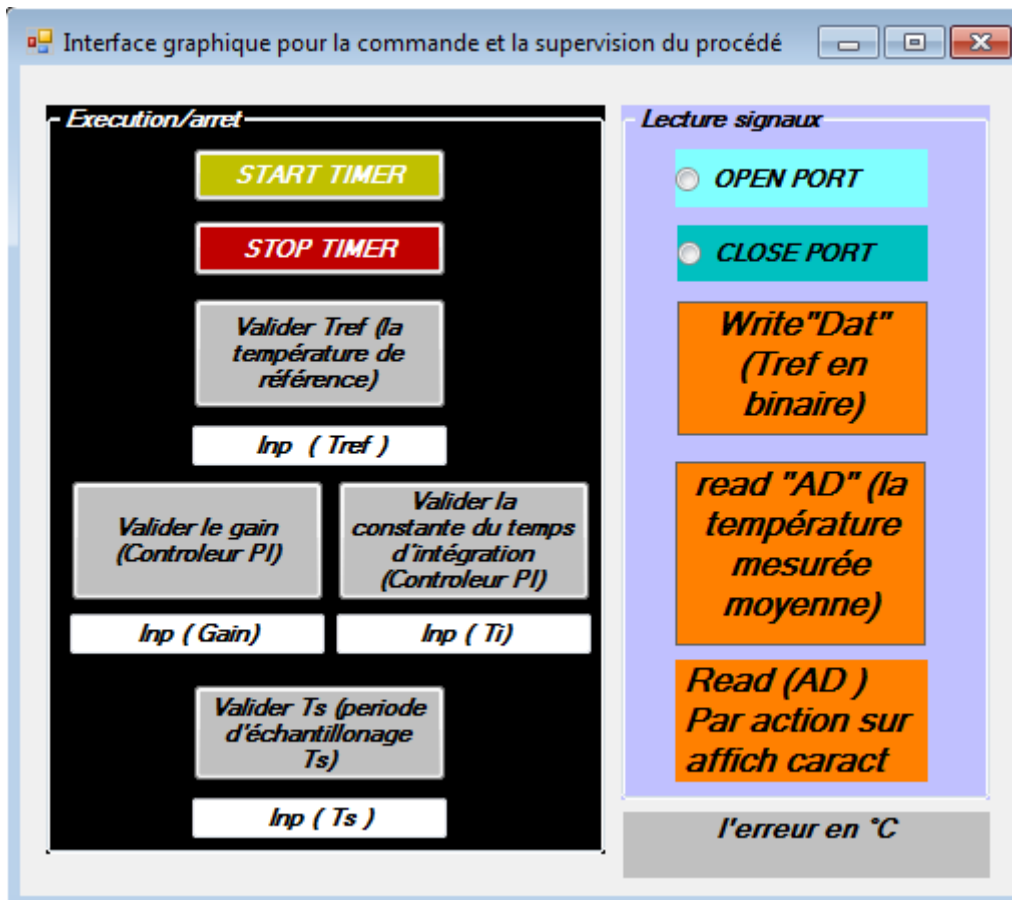
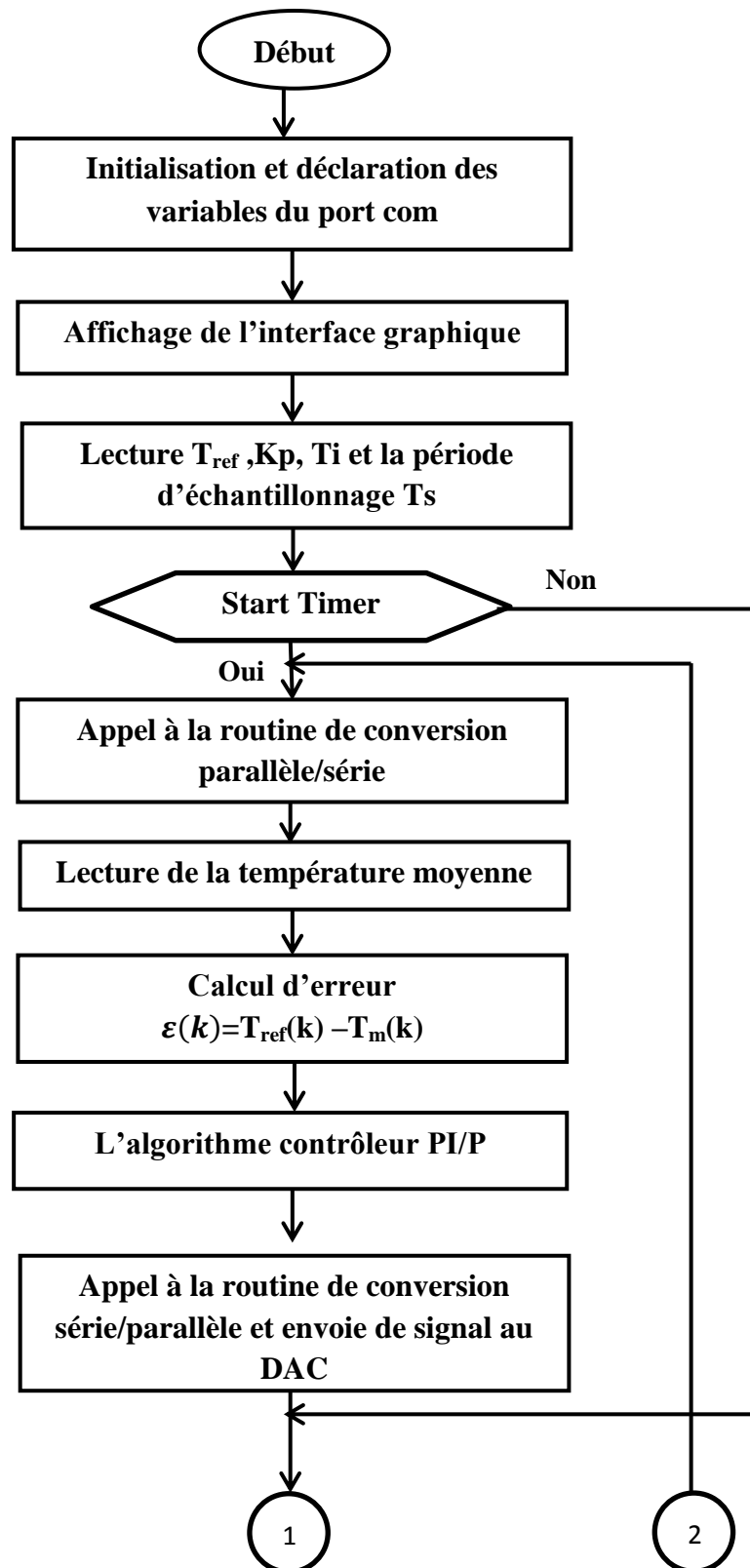


Figure IV.12 L'interface graphique (E/S) contrôle de température avec un régulateur PI non linéaire

#### IV.2.2.3 L'organigramme simplifié de software sous visual C#

Le software que nous proposons permettra un contrôle de la carte électronique (E/S) analogique. Ce programme informatique a pour fonction de générer le signal de commande de température. Cette dernière est comprise entre 0 et 10Volts pour une valeur de température comprise entre 0 et 100°C. Il permet également une mesure et affichage de température réelle  $T_r(k)$  à des périodes d'échantillonnage que nous fixons à travers le software.

Pour illustrer cette partie software, la figure (IV.13) présente l'organigramme simplifié décrivant le fonctionnement de four avec ce type de commande.



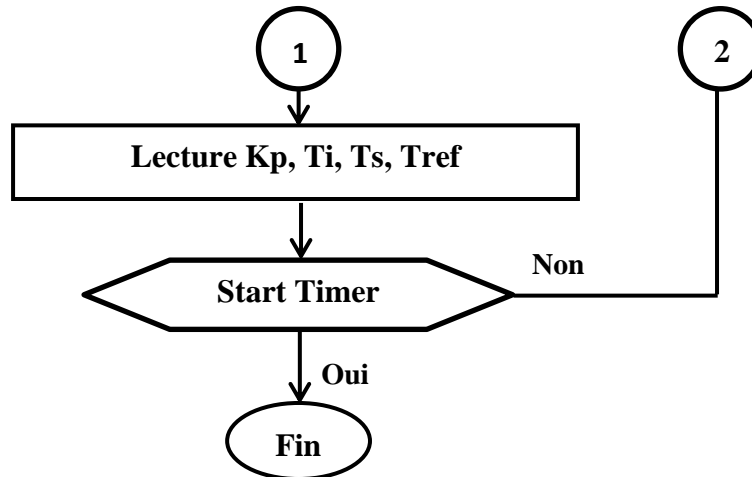


Figure IV.13 L'organigramme simplifié de software sous visual C#

Pour les routines de commande, nous avons deux variantes :

- pour le régulateur P :** Routine de commande P

$$U_p(k) = k_p \times e(k)$$

$$U_{out} = U_p(k)$$
- Pour le régulateur PI :** Routine de commande PI

$$U_p(k) = k_p \times e(k)$$

$$U_i(k+1) = U_i(k) + K_i \times e(k+1)$$

$$U_{out} = U_p + U_i$$

### IV.3 Evaluation des performances du système

#### IV.3.1 Evaluation de la commande

Dans cette partie, nous allons tester notre système de commande en ce qui concerne la régulation de la température. L'exécution du programme que nous avons implémenté sur le micro-ordinateur nous a permis de relever les différentes courbes pour un cycle de

température donnée et de visualiser le comportement de notre système de commande que nous avons développé.

La base du temps utilisée est de 100s/div et la courbe de température est donnée sous l'échelle de 2V/div et celle de commande est donnée sous l'échelle de 5V/div.

L'oscilloscope ne nous donne pas précisément les mesures et ce n'est pas possible de visualiser l'erreur statique, il nous faut un oscilloscope plus précis que celui que nous avons utilisé, mais l'utilisation de l'afficheur alphanumérique nous a aidé d'avoir les mesures de la température des trois capteurs et leur valeur moyenne ainsi que la valeur de la commande.

#### IV.3.1.1 Commande par un régulateur P avec un élément non linéaire

La figure (IV.15) représente l'évolution de la température du four pour une valeur aléatoire de gain ( $K_p=3$ ), température de référence ( $T_{ref}=70^\circ\text{C}$ ) et une période d'échantillonnage ( $T_s=1000$  ms).

L'oscillogramme montre la réaction du contrôleur (signal de sortie de régulateur) avec la variation du signal de mesure.

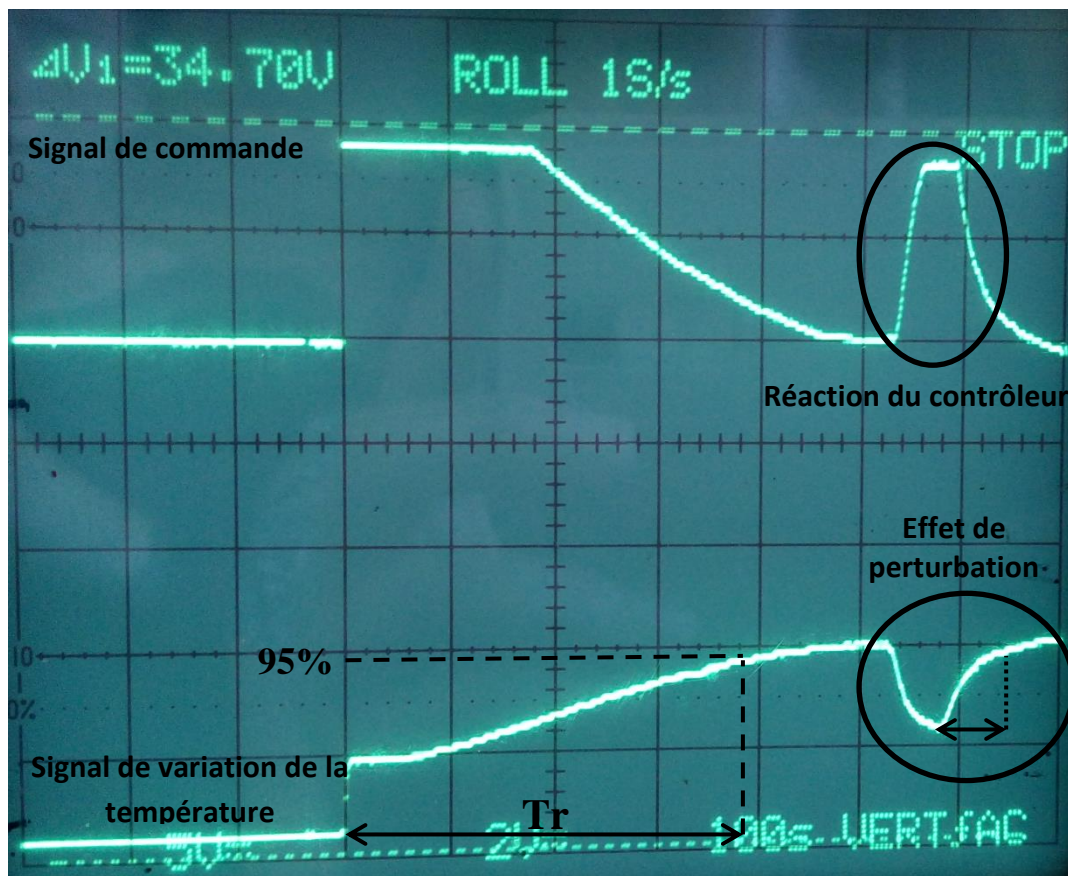


Figure IV.14 Courbe de réponse du four et le signal du contrôleur P non linéaire

L'allure (1) de la figure (IV.15) représente le signal de commande via le circuit de conversion digital/ analogique, ce signal correspond au fonctionnement du contrôleur P associé à un élément non linéaire.

L'allure (2) de la figure représente le signal de la température moyenne via le diviseur de tension.

### ➤ Interprétation

Après l'exécution de notre programme implémenté dans le micro-ordinateur, nous voyons bien sur l'oscilloscope une température qui se stabilise à 69°C après 400s, ensuite pour confirmer, nous avons testé la commande face à une perturbation, Cette action perturbatrice a été générée en éloignant la sonde du capteur de température de la résistance du four, nous remarquons que la température diminue et le signal de commande remonte à sa valeur maximale, le système prend 50s pour revenir à son état stable, où le signal de commande reviendra à nouveau à 0 volt, nous constatons par la suite que le régulateur réagit au moment de perturbation d'une façon rapide. Ce qui nous assure le bon fonctionnement de notre commande.

Nous voyons que le contrôleur P assure une réaction instantanée du signal d'erreur ; dans ce sens, sa commande ne dépend pas du passé, ni d'une tendance, mais simplement de ce qui se passe à l'instant présent.

Avec un tel contrôleur, il sera tentant d'augmenter  $K_p$  de manière à réduire l'erreur statique et rendre le système plus rapide. Néanmoins, on est limité dans cette voie pour des raisons pratiques concernant la capacité du régulateur à fournir des signaux de commande d'amplitude suffisante.

#### IV.3.1.2 Commande par un régulateur PI avec un élément non linéaire

La figure (IV.16) représente l'évolution de la température du four pour une valeur de gain ( $K_p=3$ ), température de référence ( $T_{ref}=70^\circ\text{C}$ ), une constante d'intégration  $T_i=1\text{s}$  et une période d'échantillonnage ( $T_s=1000\text{ ms}$ ).

L'oscillogramme montre la réaction du contrôleur (signal de sortie de régulateur) avec la variation du signal de mesure.

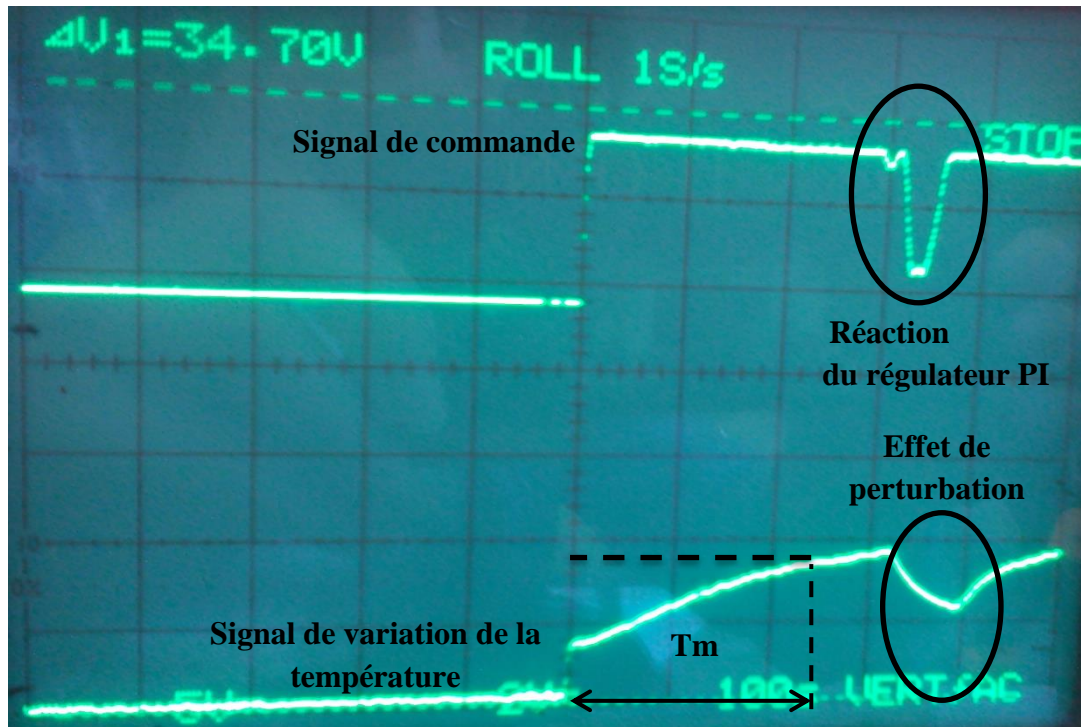


Figure IV.15 Courbe de réponse du four et le signal de contrôleur PI non linéaire

### ➤ Interprétation

L'analyse de cet oscillogramme, nous montre que le signal de commande reste constant et prend sa valeur maximale jusqu'au moment où la température atteint  $70^{\circ}\text{C}$  environ et cela après 400s, ensuite nous avons effectué une perturbation en éloignant la sonde du capteur de la résistance chauffante du four. La température diminue et le contrôleur réagit et le signal remonte brusquement à sa valeur maximale après 50s. L'erreur entre la consigne et la température réelle diminue sans atteindre la valeur zéro.

Le régulateur PI fonctionne d'une manière réursive en utilisant des données précédente donc tant que l'écart statique n'est pas proche de 0, le signal de commande prends toujours sa valeur maximale.

### IV.3.2 Réglages des paramètres du régulateur [18]

Le problème de la détermination des paramètres des régulateurs est connu par la synthèse des systèmes bouclés. Les méthodes de synthèse sont très nombreuses et une classification rigoureuse n'est pas une tâche facile.

#### On distingue deux types de méthodes

- Les méthodes basées sur la connaissance du modèle du système sous forme de fonction de transfert par exemple. Les actions du régulateur seront calculées de façon à obtenir la fonction de transfert souhaitée en boucle ouverte ou en boucle fermée.

- Les méthodes dites empiriques ne nécessitant pas une connaissance parfaite du modèle du procédé à commander. Les paramètres du régulateur seront calculés à partir des observations expérimentales sur le procédé (Relevé de la réponse indicielle par exemple). L'intérêt majeur de ces méthodes réside dans leur simplicité. Elles sont largement utilisées dans le domaine industriel et elles sont dans la plus part des cas suffisants mais ne permettent pas un réglage fin.

#### IV.3.2.1 Méthodes empiriques de Ziegler & Nichols

Ziegler et Nichols ont proposé deux approches expérimentales destinées à fixer rapidement les paramètres des régulateurs P, PI et PID. La première nécessite l'enregistrement de la réponse indicielle du système à régler, alors que la deuxième exige d'amener le système en boucle fermée à sa limite de stabilité.

Pour notre cas, on va utiliser **la méthode de Ziegler et Nichols en boucle ouverte**.

##### a. Mode opératoire

- 1) Nous sélectionnons le bouton « Port Série Open » de l'interface graphique pour ouvrir la boucle.
- 2) Nous envoyons une variation constante du signal d'entrée du procédé et nous enregistrons la variation du signal de mesure à la sortie du procédé.

Il s'agit donc de l'enregistrement de la réponse indicielle du procédé seul.

##### b. Exploitation de la réponse indicielle

Sur l'enregistrement de la réponse indicielle, nous traçons le mieux possible la tangente au point d'inflexion Q de la courbe. Nous mesurons ensuite le temps  $T_u$  correspondant au point d'intersection entre l'axe des abscisses et la tangente ainsi que le temps  $T_a$  « temps de montée de la tangente ».

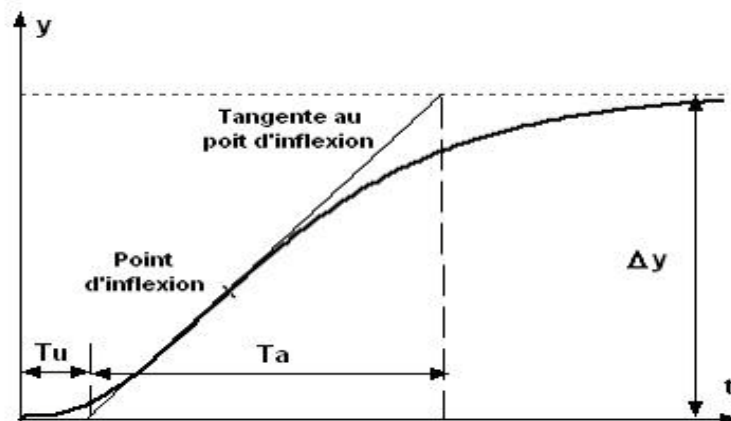


Figure IV.16 Schéma expliquant l'exploitation de la réponse indicielle

### c. Réglage des régulateurs

Ziegler & Nichols proposent de calculer les paramètres du régulateur P, PI ou PID à l'aide des recommandations suivantes présentées en Tableau (IV.2)

**Tableau IV.2 Calcul des paramètres par la méthode de Ziegler & Nichols**

Réglage des paramètres du régulateur	Kp	Ti	Td
P : $C(p) = Kp$	$\frac{Ta}{Tu * K}$	*	*
PI : $C(p) = Kp(1 + \frac{1}{sTi})$	$\frac{0.9Ta}{Tu * K}$	3.33Tu	*
PID : $C(p) = Kp(1 + sTd + \frac{1}{sTi})$	$\frac{1.2Ta}{Tu * K}$	2.0Tu	0.5Tu

En utilisant l'approche expérimentale de Ziegler & Nichols en boucle ouverte sur la réponse indicielle du four électrique, on a relevé la valeur de Tu et Ta.

**Tu=100s      Ta=392s.      K=0.98**

**Pour le régulateur proportionnel :**

Le gain proportionnel

$$Kp = \frac{392}{0.98 * 100} = 4$$

**(IV.7)**

Pour le régulateur PI :

$$Kp = \frac{0.9 * 392}{100 * 0.98} = 3.6$$

**(IV.8)**

$$Ti = 3.33 * 100 = 333ms = 0.333s$$

**(IV.9)**

**Tableau IV.3 Résultats de la synthèse**

Réglage des paramètres du régulateur	Kp	Ti	Td
P : $C(p) = Kp$	4	*	*
PI : $C(p) = Kp(1 + \frac{1}{sTi})$	3.6	0.333	*



Après la détermination des paramètres des régulateurs nous exécutons notre programme de commande sur le four électrique, pour le régulateur PI déterminé et cela pour une référence de 65°C et une période d'échantillonnage de 1000 ms.

La courbe de réponse et le signal de commande sont illustrés en figure (IV.17).



**Figure IV.17 Courbe de réponse du four et le signal de contrôleur PI non linéaire déterminé par la synthèse**

### ➤ Interprétation des résultats

L'interprétation des résultats obtenus se fait en analysant trois critères cités ci-dessous :

#### **La précision**

L'écart entre la valeur de consigne et la valeur effectivement réglée est presque nulle, cela est dû à l'utilisation d'une action intégrale **I** qui permet de rendre le système plus précis statiquement.

#### **La rapidité**

Un système a une rapidité satisfaisante s'il se stabilise à son niveau constant en un temps jugé satisfaisant. Elle est qualifiée en analysant le temps de montée en température qui correspond au temps nécessaire pour atteindre 95% de la valeur finale à partir de l'instant d'application de la consigne.

95% de la valeur finale correspond à 62°C

De la figure(IV.17) nous déduisons le temps nécessaire pour atteindre  $62^{\circ}\text{C}$  est d'environ 720s , donc nous constatons que notre système est lent.

### **La stabilité**

Le système constitué du procédé et de la boucle de régulation est dit stable s'il est soumis à une perturbation ou une variation de consigne, la grandeur réglée retrouve un état stable.

Dans notre cas le système est soumis à une perturbation et la grandeur réglée retrouve son état stable comme il est illustré en figure (IV.17) donc notre système est stable.

Après l'analyse et l'interprétation des résultats, nous constatons que le régulateur numérique PI avec les paramètres trouvés après la synthèse, arrive à annuler l'erreur statique donc rends le système plus précis statiquement mais avec un grand temps de réponse.

### **CONCLUSION**

Pour le contrôle de température du four électrique nous avons implémenté des programmes software en utilisant deux langages de programmation, le visual C# pour la réalisation des interfaces homme-machine et le mikro C pour la programmation du microcontrôleur qui permet l'acquisition des données et l'affichage sur l'afficheur alphanumérique.

Deux types de régulateurs sont utilisés, un régulateur à action proportionnelle et un régulateur à action proportionnelle et intégrale. Afin d'éviter la saturation du DAC avec son amplificateur nous ajoutons un élément non linéaire à la sortie de chaque régulateur.

Après l'analyse des courbes de réponse du système en boucle fermée pour les deux régulateurs nous concluons que notre système de commande fonctionne et réagisse correctement.

## CONCLUSION GENERALE

L'objectif de ce travail est tourné vers la mise en œuvre d'une stratégie de régulation qui permet de contrôler la température du four électrique de laboratoire avec des faibles coûts en utilisant des composants électroniques simples et pas onéreux.

Ce travail présente une étude expérimentale d'une commande numérique hybride pour le contrôle d'un four électrique disponible au niveau de laboratoire, cette commande est une combinaison d'une commande par micro-ordinateur et par microcontrôleur.

Notre étude se divise en deux parties :

La première partie consiste à présenter et identifier le four électrique ainsi que la validation du modèle identifié en tenant compte du non linéarité du système, tout en basant sur des données expérimentales.

La deuxième partie constitue en premier lieu la réalisation d'une carte d'interface électronique à base d'un microcontrôleur programmé par le langage de programmation MikroC.

Cette carte qui permet une jonction entre le processus et le PC via le port série du Micro-ordinateur a été testée sur une plaque d'essai afin de s'assurer son bon fonctionnement et de sa fiabilité avant son branchement avec le calculateur.

En deuxième lieu l'implémentation et l'élaboration du software sous forme d'interface graphique à l'aide de Visual C#2010 qui permet le pilotage de l'ensemble des paramètres du processus de manière visuelle par un simple CLICK de la souris. Ce programme informatique a comme fonction de générer la référence de température, la comparaison avec la grandeur réelle et enfin l'élaboration des algorithmes de réglage que nous avons retenus. Nous avons élaboré deux types de commande, un régulateur P et un régulateur PI, avec un élément non linéaire afin d'éviter la saturation du DAC et son amplificateur.

Au cours de notre travail qui est purement expérimental nous avons pu créer un programme de commande qui peut avoir de nombreuses applications dans le domaine de l'industrie et qui peut être facilement adapté pour répondre à d'autres paramètres à contrôler tel que la pression, le débit et l'humidité.

Pour des développements futurs nous proposons une régulation anti Wind up pour annuler le problème causé par l'ajout de l'élément non linéaire, ainsi que l'amélioration de la

carte afin de réguler d'autres grandeurs physiques en parallèle, en ajoutant d'autre registre à décalage et d'autre DAC.

Nous proposons aussi une commande adaptative qui permet une sélection du gain  $K_p$  et de la constante d'intégration  $K_i$  en fonction de la température de référence.

# Annexe (A)

Programme software pour un contrôleur p sous visual C#

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using System.Threading;

namespace WindowsFormsApplication21
{
    public partial class Form1 : Form
    {
        string str = "COM1: baud=115200 parity=N data=8 stop=0";
        [DllImport("rscom")]
        static extern void OPENCOM(string str);
        [DllImport("rscom")]
        static extern void CLOSECOM();
        [DllImport("rscom")]
        static extern void TXD(int bite);
        [DllImport("rscom")]
        static extern void DTR(int bite);
        [DllImport("rscom")]
        static extern void RTS(int bite);
        [DllImport("rscom")]
        static extern int DSR();
        [DllImport("rscom")]
        static extern int DCD();
        public int AD;//DIN
        public int VT;//Tc
        public int ts;//Te
        public int dat=0;
        public int n1;
        public int n2;
        public int AD1;//DIN1

        public int dsrn;
        public int m = 0;
        public int AD2 = 0;
        public int kp = 0;
        public int delT1 = 0;
        public int delT = 0;
        public int delTbin = 0;
        public int ecart;
        public int dat1 = 0;
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            this.timer1.Enabled = true;
            timer1.Interval = ts;
            button2.BackColor = Color.GreenYellow;
        }
    }
}
```

```

        button1.Text = "EXECUTION EN COURS";
    }

private void button2_Click(object sender, EventArgs e)
{
    this.timer1.Enabled = false;
    button1.BackColor = Color.Red;
    button2.Text = "PROGRAMME A L ARRET";
    button1.Text = "EXECUTION INTEROMPUE";
}

private void button3_Click(object sender, EventArgs e)
{
    VT = int.Parse(textBox1.Text);
    n2 = 255*VT/100;
}

private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    OPENCOM(str);
    radioButton1.Text = "SERIAL PORT OUVERT";
}

private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    CLOSECOM();
    radioButton2.Text = "SERIAL PORT FERMEE";
}

//tempo pour LCD et clock avec DTR
public void del1()
{
    for (int k = 1; k <= 5; k++)
    {
        label4.Text = "INTERRUPTION 2MS";
        label4.Refresh();
    }
}

//
public int DA( int dat,int dat1,int AD)
{
    // ROUTINE DE CONVERSION SUR 24 BITS AVEC CMOS 4094
    int bit1 = 1;
    dsrn=0;
    RTS(0);
    for (int i = 1; i <= 8; i++)
    {
        if ((dat+dat1 & bit1) > 0)
        {
            TXD(1);
        }
        else
        {
            TXD(0);
        }
    }
    dsrn = DSR();

    if (dsrn == 1)
    {
        AD = AD + bit1;// VARIABLE TEMPERATURE EN BINAIRE
    }
    DTR(1);
    del1();
}

```

```

        DTR(0);
        bit1 = bit1 * 2;
    }
    RTS(1);
    Thread.Sleep(1);
    RTS(0);
    AD1 = AD;
    return AD1;
}
private void timer1_Tick(object sender, EventArgs e)
{
    AD1 = AD;
    //envoi carac T *****
    DA(dat , dat1, AD);
    dat = 0;
    dat = dat + n1;
    DA(dat,dat1, AD);
    AD2 = AD1 * 100 / 255; //VARIABLE TEMPERATURE EN °C DU ADC 0804
    //ROUTINE CONTOLEUR
    delT1 = n2 - AD1; //ECART DE REGLAGE EN BINAIRE
    ecart = VT - AD2;+-
    label5.Text = "ECART=" + ecart.ToString()+"°CELSIUS";
    label5.Refresh();
    //LIMITATION DE L'ECART DE REGLAGE SIGNAL EXCLUSIVEMENT POSITIF
    if (delT1 > 0)
    {
        delT = delT1;
    }
    else if (delT1 < 0)
    {
        delT = 0;
    }
    else
    {
    }
    // LIMITATION DU SIGNAL DE SORTIE VERS DAC ENTRE 0 A 255 ( 8 bits)
    if (delT * kp <= 255)
    {
        dat1 = delT * kp;
    }
    else if (delT * kp > 255)
    {
        dat1 = 255;
    }
    else if (delT * kp < 0)
    {
        dat1 = 0;
    }
    else
    {
    }
    //fin program cont
    DA(dat, dat1, AD);
    label1.Text = "N2=" + n2.ToString() ;
    label1.Refresh();
    label2.Text = "TM=" + AD2.ToString() + "Degres";
    label2.Refresh();
    progressBar2.Value = AD2;

    // label2.BackColor = Color.
    label3.Text= "AD1(Bin)=" + AD1.ToString();
}

```

```

        label3.Refresh();
    }

private void timer2_Tick(object sender, EventArgs e)
{
}

private void button4_Click(object sender, EventArgs e)
{

}

public void writedata(int y1)
{
}

private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
}

private void button5_Click(object sender, EventArgs e)
{
    ts = int.Parse(textBox2.Text);
    timer1.Interval = ts;
}

private void button6_Click(object sender, EventArgs e)
{
}

private void textBox4_TextChanged(object sender, EventArgs e)
{
}

private void button7_Click(object sender, EventArgs e)
{
    // gain du controleur
    kp = int.Parse(textBox4.Text);
}

private void Form1_Load(object sender, EventArgs e)
{ }

private void label5_Click(object sender, EventArgs e)
{ }

private void label2_Click(object sender, EventArgs e)
{ }

private void label1_Click(object sender, EventArgs e)
{
}

private void groupBox1_Enter(object sender, EventArgs e)
{ }
}}

```

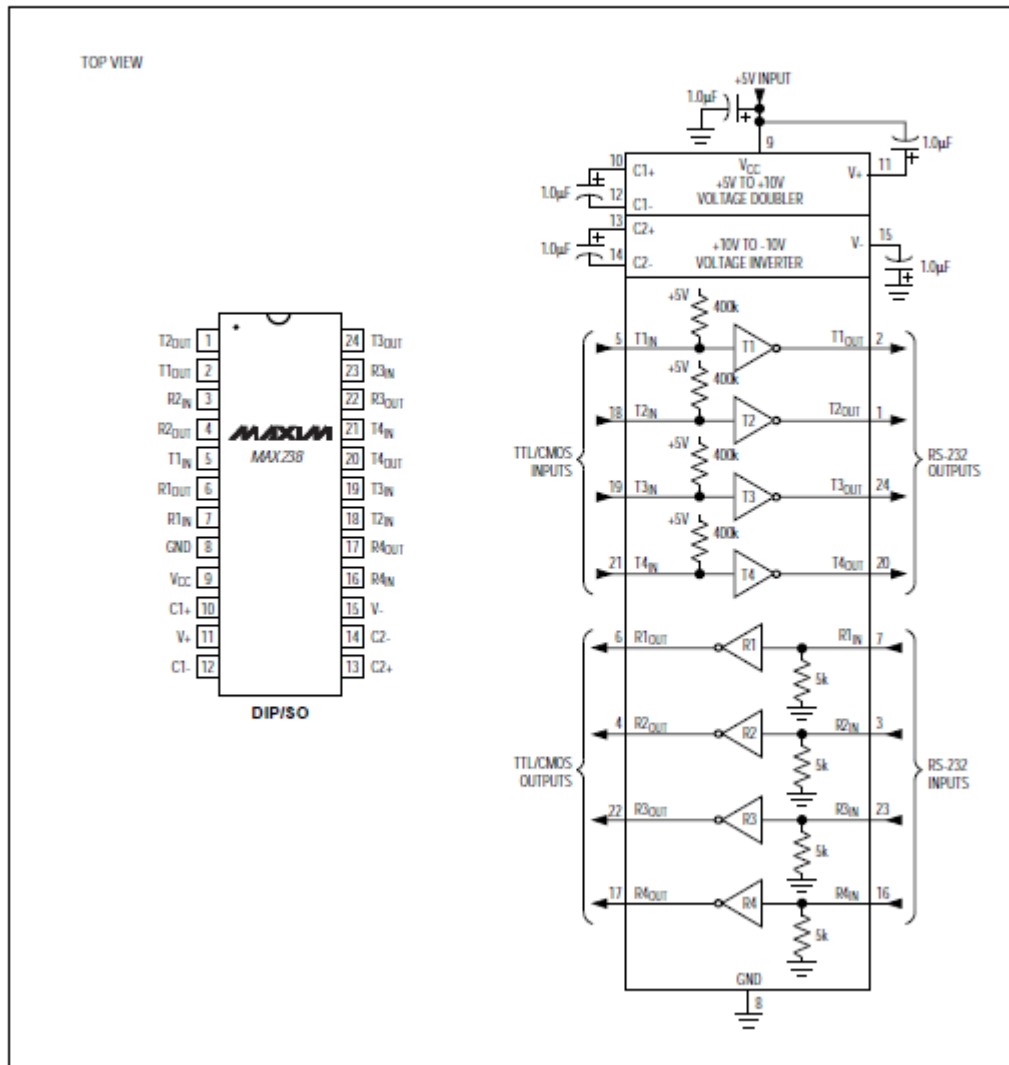


# Annexe (B)

Extrait de datasheet

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

MAX220-MAX249



## BIBLIOGRAPHIE

- [1] Document de laboratoire d'automatique appliqué, Année 2003.
- [2] Jean-François BOURGEOIS, Richard JAUME, Marianne le boulche, EDF Division Recherche & Développement, Claude OBERLIN, Ingénieur Senior SE, <Fours électriques à résistances-Technologies de mise en œuvre>,12 Septembre 2 014.
- [3] [www.wikipédia.com](http://www.wikipédia.com),<Triacs> , un article de wikipédia, l'encyclopédie libre.
- [4] Paul Klounowski <<Use of the AD590 temperature transducer in a remote sensing application >>, ANALOG DEVICE.
- [5] Prof.Belkacem OULD BOUAMAMA, <Identification des systèmes>, Laboratoire d'automatique, Génie informatique et signal (LAGIS-UMR CNRS 8219).
- [6] H.FEKHAR, KEBIR.H, <Réalisation d'une carte d'interface PC avec implémentation du SOFTWARE sous Windows. Application à un asservissement de température>.
- [7] [www.wikipédia.com](http://www.wikipédia.com), <Filtre passe bas>, Un article de Wikipédia, l'encyclopédie libre.
- [8] [Electronique.aop.free.fr](http://Electronique.aop.free.fr),<Filtre Sallen and Key>.
- [9] Datasheet MAX238, <+5V-Powered, Muttichannel RS-232 Drivers/Receivers>par MAXIM.2006 Maxim Integrated Products.
- [10] Martins Bates,<<Pic Microcontrôleur an introduction to Microelectronics>>ELSEVIER 2007
- [11] C.Tavevov,<<Microcontrôleurs PIC18>> édition Dunod 2008
- [12] Y.Rajasree,<<advanced microprocessors>> new-age international publishers Delhi 2006
- [13] L.D jasio, T-Wilmshort ,D.Mortor ,M.P.Bates <<Pic microcontrollers>> Elsevier News 2008.
- [14] H.FEKHAR, M.BOUAZDIA,M.DJOUDI<Conception et réalisation d'une commande de type système embarqué avec utilisation de la technique M.L.I.Application aux contrôles de température et pression>.
- [15] Datasheet HEF4094B, < 8-stage shift-and-store register>2013.
- [16] W.Kretiz, <Digital electronics> ,Prentice Hall, INC 2000.
- [17] C.Cimeli –R.Bourgeron. Guide du technicien en électronique.Hachette Technique.
- [18] Dr.A.KHELASSI <<Cours sur l'automatisation des systèmes industriels>>UMBB-Boumerdes 2015.
- [19] User's Manual, <<COMPILER FOR MICROCHIP PIC MICROCONTROLLERS>> ,mikroElektronika.
- [20] Manual ISIS V6 (traduction multipower 1999-2006 V6-avril 2006).
- [21] « Microcontrôleur :utilisation de PICKIT2 ».
- [22] <https://msdn.microsoft.com/site> officiel du microsoft.

