

N°Ordre...../Faculté/UMBB/2016

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



Faculté des Hydrocarbures et de la Chimie

**Mémoire de fin d'études
En vue de l'obtention du diplôme**

MASTER 2

Présenté par

BOUROUIS THABET

Spécialité : Automatisation des procédés industriels

Option : Commande automatique

THEME

**Conception et réalisation d'un Système de supervision et de
contrôle d'un four électrique avec un software sous Visual C# et
une carte électronique à base de microcontrôleur**

Devant les membres de jury :

**KIDOUCHE
HABBI
FEKHAR**

**Madjid
Hacène
Hacène**

**Prof
Prof
MA/A**

**UMBB Président
UMBB Examineur
UMBB Encadreur**

Année universitaire : 2015/2016

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



Faculté des Hydrocarbures et de la Chimie

**Mémoire de fin d'études
En vue de l'obtention du diplôme**

MASTER 2

Présenté par

BOUROUIS THABET

Spécialité : Automatisation des procédés industriels

Option : Commande automatique

THEME

**Conception et réalisation d'un Système de supervision et de
contrôle d'un four électrique avec un software sous Visual C# et
une carte électronique à base de microcontrôleur**

Avis favorable de l'encadreur :

Dr. FEKHAR.H

Avis favorable du Président du jury :

.....

Remerciements

Je tiens à remercier en premier lieu ALLAH, le tout puissant, qui m'a donné le courage et la volonté pour bien mener ce travail.

Je remercie, mon encadreur Monsieur H.FEKHAR pour sa disponibilité, son sérieux et ses conseils valeureux.

Je remercie tous les enseignants du département automatisation et électrification des procédés industriels, qui ont assurés ma formation durant tout le cycle d'étude.

Je remercie également l'ensemble des membres de notre club scientifique « AEPI New Vision » avec lesquels j'ai passé de formidables années.

Je tiens par cette occasion à présenter mon vif remerciement à tous ceux qui ont collaboré de près ou de loin à la réalisation de ce modeste travail.

Bourouis Thabet

DEDICACE

J'ai le grand honneur de dédier ce

Modeste travail

A ceux qui m'ont fait de moi un homme

*"MA MERE" et "MON PERE" qui m'ont aidé et
m'ont encouragé beaucoup durant toutes mes études.*

A mes chers frères Adel, Nouredine, Abdessalam et Fouad.

A toute ma famille

A tous mes amis et mes proches

A tout ceux qui m'ont aidé de loin ou de près durant

Les moments difficiles

Sommaire

INTRODUCTION GENERALE	15
Chapitre I : Description et présentation du process.	
I-1. Introduction	18
I-2. Description du four électrique à résistance	18
I-2.1 Généralités sur les fours électriques à résistance	18
I-2.2 Constituants du four	18
I-2-1.a Une enceinte (chambre de chauffe)	18
I-2-1.b Résistances électriques	18
I-2-1.c La charge à traiter	18
I-2.3 Principe de fonctionnement.....	19
I-3. Présentation du four de laboratoire	19
I-3.1 Description du four de laboratoire	19
I-3.2 Son circuit de puissance	20
I-3.3 Présentation de capteur de température	22
I-3-3.a Généralités sur les capteurs de température	22
I-3-3.b Capteur de température à semi-conducteur AD590	22
I-3.4 Circuit de conditionnement de capteur	23
I-3-4.1 Conversion courant - voltage	24
I-3-4.2 Filtre de second d'ordre	24
I-3-4.3 Un diviseur de tension	25
I-4. Conclusion	25
Chapitre II : Identification de la dynamique du four.	
II-1. Introduction	27
II-2. Mise en œuvre de l'identification	27
II-3. Identification par la méthode graphique de STREJC	29
II-3.1 Méthodologie	29
II-3.2 Validation du modèle obtenue	30
II-4. Identification par le toolbox « ident » sous MATLAB	31
II-4.1 Méthode d'utilisation du toolbox	32
II-4.2 Valider le modèle	36

II-5. Conclusion.....	36
Chapitre III : Description de la carte de commande réalisée.	
III-1. Introduction	38
III-2. Schéma bloc de la carte électronique de communication PC-PIC	38
III-3. Schéma réel de la carte électronique pilote par USB-UART.....	40
III-3.1 Organe de commande	40
III-3.2 Organe d'acquisition de données	41
III-3.3 Organe de visualisation (LCD 2 x 16)	42
III-3.4 Organe de communication	42
III-3.5 Organe de conversion ADC et DAC	43
III-3-5.a Le convertisseur analogique numérique ADC	43
III-3-5.b Le convertisseur numérique analogique DAC0800.....	44
III-3.6 Simulation et réalisation de la carte de commande	46
III-4. Présentation de port de communication UART	48
III-4.1 Introduction	48
III-4.2 Constituants d'une trame UART	48
III-4.3 Vitesse de transmission	48
III-4.4 Principe de transmission par UART.....	49
III-4-4.1 Transmission sérielle asynchrone	49
III-4-4.2 Fonctionnement d'UART	49
III-4.5 Caractéristiques d'une UART	50
III.5 Présentation de MCU PIC18F4550 avec son architecture interne.....	50
III-5.1 Généralités sur les microcontrôleurs	50
III-5.2 Classifications des PICs de MICROCHIP.....	51
III-5.3 Choix du PIC	51
III-5.4 Microcontrôleur PIC18F4550.....	51
III-5.5 Fonction des différentes broches du PIC184550.....	52
III-5.6 Architecture interne du PIC18F4550.....	53

III.6 Conclusion	54
Chapitre IV : Supervision et acquisition des données.	
IV-1. Introduction	56
IV-2. Présentation des logiciels utilisés	56
IV-2.1 Visual studio	56
IV-2.2 Introduction sur les langages orientés objet	57
IV-2.3 Langage de programmation C#	58
IV-2.4 La plateforme .NET	58
IV-2.5 MIKROC	59
IV-3. Présentation générale sur la supervision industrielle	59
IV-3.1 Introduction	59
IV-3.2 Avantage de la supervision	59
IV-3.3 Fonctionnalités d'un système de supervision	60
IV-3-3.a La commande	60
IV-3-3.b La surveillance	60
IV-3.4 Architecture d'un réseau de supervision	61
IV-4. Présentation de l'application réalisée	61
IV-4.1 Acquisition des données	62
IV-4-1.a Introduction	62
IV-4-1.b Description d'étapes d'acquisition des données	63
IV-4.2 Supervision	66
IV-4-2.1 Synoptique	67
IV-4-2.2 Courbes	68
IV-4-2.3 Alarme	69
IV-4-2.4 Historisation du procédé	70
IV-4-2.5 Gestion des paramètres de Contrôle	70
IV-4-2.6 Gestion de base des données	71
IV.5 Résultats de simulation du projet	72
IV-5.1 Simulation de la synoptique du logiciel	72

IV-5.2 Simulation d'un déclenchement de système d'alarme	74
IV.6 Conclusion	75
Chapitre V : Implémentation du programme de commande.	
V-1. Introduction	77
V-2. Schéma bloc de la boucle de régulation du four électrique	77
V-3. Commande en boucle ouvert avec feedback de mesure	78
V-3.1 La structure de la commande	78
V-3.2 Implémentation de l'algorithme de contrôle.....	78
V-3.3 Résultats et interprétations.....	79
V-4. Commande en boucle ferme.....	80
V-4.1 Contrôleur ON-OFF.....	81
V-4-1.1 La structure de commande	81
V-4-1.2 Implémentation de l'algorithme de contrôle	81
V-4-1.3 Résultats et interprétations	83
V-4.2 Contrôleur ON-OFF à hystérésis	83
V-4-2.1 La structure de commande	83
V-4-2.2 Implémentation de l'algorithme de contrôle	83
V-4-2.3 Résultats et interprétations	85
V-4.3 Contrôleur PI non-linéaire	85
V-4-3.1 La structure de commande	85
V-4-3.2 Implémentation de l'algorithme de contrôle	86
V-4-3.3 Résultats et interprétations	87
V.5 Conclusion	88
CONCLUSION GENERALE.....	90
Références bibliographique	93
ANNEXE	95

Liste des figures

Figure I.1 : photo réelle du four du laboratoire	19
Figure I.2 : Constitutions du four du laboratoire	20
Figure I.3 : Schéma simplifié d'une tension module à l'aide d'un triac	21
Figure I.4 : Schéma interne du circuit de puissance du four.....	21
Figure I.5 : Différents régimes de fonctionnement du Triac.....	21
Figure I.6 : schéma interne du capteur AD590.....	23
Figure I.7 : La structure du circuit de conditionnement	24
Figure I.8 : schéma de circuit de conditionnement de capture.....	24
Figure I.9 : Schéma d'un filtre du second d'ordre	25
Figure I.10 : schéma d'un diviseur de tension.....	25
Figure II.1 : réponse du four relevé par oscilloscope après filtrage	28
Figure II.2 : la réponse du four relevé par notre interface graphique.....	28
Figure II.3 : comparaison de la réponse réelle et avec celle identifiée	30
Figure II.4 : Erreur d'identification	31
Figure II.5 : représente les sections principales de l'outil « ident ».....	32
Figure II.6 : Schéma synoptique de l'identification par le toolbox « ident ».....	33
Figure II.7 : introduction des données entrées/sorties sur le toolbox	33
Figure II.8 : choix de structure de modèle.....	34
Figure II.9 : Courbes du système réel	34
Figure II.10: Courbes de réponse du système réel et celle du modèle mathématique ...	35
Figure II.11: Réponse indiciaire du modèle identifié	36
Figure III.1: Architecture matérielle et logicielle du dispositif de contrôle	38
Figure III.2 : Schéma bloc de la carte électronique	39
Figure III.3 : Répartition des broches du PIC vers les périphéries entrées/sorties.....	40
Figure III.4 : photo réelle du capteur de température AD590.....	41

Figure III.5 : Schéma électronique du circuit de conditionnement du capteur AD590...	41
Figure III.6 : une photo réelle d'afficheur LCD	42
Figure III.7 : schéma de brochage minimal d'afficheur LCD	42
Figure III.8 : photo réelle du circuit USB-UART	43
Figure III.9 : photo réelle du DAC0800.....	44
Figure III.10 : Structure interne du DAC0800.....	45
Figure III.11 : Schéma de branchement réel du DAC08EN	45
Figure III.12 : Schéma de brochage final de la carte de commande	46
Figure III.13 : Photo réelle de la carte de commande	47
Figure III.14 : schéma représente les constituants d'une trame UART	48
Figure III.15 : Transmission sérielle en mode asynchrone	49
Figure III.16 : Désignation standard des pins du PIC 18F4550.....	52
Figure III.17 : Architecture interne du PIC18F4550.....	53
Figure IV.1 : logo de Visual Studio	56
Figure IV.2 : la fenêtre de l'environnement du développement VISUAL STUDIO.....	57
Figure IV.3 : représente le dialogue qui permet de créer un nouveau projet	57
Figure IV.4 : Représente le logo de langage C#.....	58
Figure IV.5 : représente le logo de la plateforme .NET	58
Figure IV.6 : représente le logo de MIKROC	59
Figure IV.7 : l'architecture d'un système de supervision	61
Figure IV.8 : la structure générale du software de commande et supervision d'un four électrique.....	62
Figure IV.9 : Constituants d'un système d'acquisition des données.....	63
Figure IV.10 : la configuration logicielle des registres associe aux ports A et D	63
Figure IV.11 : représente la trame des donnees transmit a l'ordinateur	64
Figure IV.12 : représente la trame de signal de commande	65
Figure IV.13 : représente la trame des donnees transmit a l'ordinateur	65
Figure IV.14 : représente le décodage de la trame	66
Figure IV.15 : représente la structure d'un tableau TEMP dans la base des données Sql Server.....	66
Figure IV.16 : la structure logicielle de notre système de supervision	67

Figure IV.17 : la synoptique de logiciel de supervision avec ces parties	67
Figure IV.18 : représente la fenêtre « courbes ».....	68
Figure IV.19 : représente la fenêtre « alarme » et ses parties	69
Figure IV.20 : La fenêtre « historique ».....	70
Figure IV.21 : l’interface graphique « gestion des paramètres ».....	71
Figure IV.22 : représente la structure de base de données utilise dans ce projet	71
Figure IV.23 : la fenêtre « portConf ».....	72
Figure IV.24 : le synoptique à l’état initial	72
Figure IV.25 : Le système de supervision en état marche.....	73
Figure IV.26.a : le système de supervision avec le régulateur TOR	73
Figure IV.26.b : le système de supervision avec le régulateur a hystérésis	74
Figure IV.26.c : le système de supervision avec le régulateur PID	74
Figure IV.27 : la section de configuration des seuils d’alarme	75
Figure IV.28 : l’état de synoptique en cas le déclenchement d’alarme	75
Figure V.1 : Schéma simplifié de la boucle de régulation	77
Figure V.2 : présente la structure de la commande en boucle ouvert	78
Figure V.3 : Organigramme simplifier du programme de commande en BO.....	79
Figure V.4 : L’évolution de la température du four en fonction du temps	80
Figure V.5 : Schéma synoptique simplifié du command en boucle fermée	80
Figure V.6 : présente le schéma de régulation ON-OFF	81
Figure V.7 : Algorithme de réglage ON-OFF	82
Figure V.8 : Courbe de signal de la commande et de la réponse du four avec contrôleur ON-OFF	83
Figure V.9 : Schéma fonctionnel de la commande de la plaque chauffante avec contrôleur ON/OFF a hystérésis	83
Figure V.10 : Organigramme de commande avec contrôleur ON-OFF a hystérésis.....	84
Figure V.11 : Courbe de signal de la commande et courbe de réponse du four avec un contrôleur ON-OFF a hystérésis	85
Figure V.12 : Schéma fonctionnel du contrôleur PI avec anti-windup	86
Figure IV.13 : Organigramme globale du la régulation	87
Figure IV.14.a : la commande et la réponse du four avec un contrôleur PI non linéaire	88

Liste des tableaux

Tableau II.1 données de la méthode STRETCH	7
Tableau III.1 bits de configuration du registre ADCON0.....	17
Tableau III.2 bits de configuration du registre ADCON1.....	35
Tableau III.3 : bits de configuration du registre ADCON2.....	61

Introduction générale

Introduction générale :

L'automatisation et l'informatisation apparaissent comme deux des plus importants facteurs de développement des grands procédés industriels continus. La part des calculateurs devient de plus en plus prépondérante et comprend l'acquisition des mesures, parfois à des techniques de traitement de signal et la presque totalité des commandes, effectuées suivant des critères d'optimisation parfois complexes. Les calculateurs sont connus pour leurs capacités de précision, de mémorisation, de rapidité, de répétabilité.

La demande croissante en qualité, sécurité, disponibilité et optimisation des coûts dans les procédés industriels rend nécessaire l'utilisation de systèmes de supervision avancées dans ces procédés.

Dans ces systèmes, une grande partie du développement a été mis sur la conception des interfaces. On peut avoir une vision globale de l'architecture matérielle du procédé grâce à un synoptique, qui reproduit les schémas d'installation. On peut aussi accéder à la vision des historiques des variables, ou de plusieurs variables à la fois s'il est intéressant de corrélérer leur évolution.

La supervision des boucles de commande locales est la tâche la plus simple. Les commandes ont été calculées grâce à un certain nombre d'hypothèses simplificatrices sur le fonctionnement du procédé, la plus fréquente étant la linéarité autour d'un point de fonctionnement.

Notre travail consiste à l'élaboration d'un système de supervision développé par le langage de programmation C#, ayant pour le but de superviser et de contrôler le procédé à commander. Le système est utilisé une carte électronique à base de microcontrôleur couplé à un PC.

Vu l'importance d'un système de supervision et de commande dans un procédé industriel, nous avons opté pour l'implémentation d'un software complet qui sert à superviser, enregistrer les données du procédé, avec un contrôle de la température du four.

Au niveau de ce présent mémoire, notre travail consistera à effectuer cinq tâches essentielles:

- Présentation du four électrique de laboratoire.
- Identification de la dynamique du four, en exploitant les courbes de réponses.

Introduction générale

- Elaboration de la carte électronique qui permettra une communication entre l'ordinateur et le four électrique.
- Elaboration d'un software à l'aide du langage de programmation Visuel C# pour la supervision et le contrôle du four.
- Implémentation d'un programme informatique contenant différentes techniques de contrôle assurant la stabilité avec les meilleures performances possibles.
- Présentation des courbes de réponse et évaluation de la commande.

Le premier chapitre sera consacré à la description de système à commander, le four avec les trois éléments qui le composent à savoir l'organe de puissance à Triac, la résistance chauffante et le capteur à base de semi-conducteur.

Le deuxième chapitre concerne l'identification de la dynamique du four électrique en utilisant deux méthodes. La première par la méthode graphique de Strejc et la deuxième par le toolbox de MATLAB« **ident** ».

Dans le troisième chapitre, nous nous intéresserons essentiellement à développer une carte de commande électronique à base de microcontrôleur permettant l'adaptation entre le procédé et le software de commande. Cette carte électronique sera constituée des éléments suivants:

- Des circuits d'adaptation et de filtrage.
- Circuit de puissance pour la plaque chauffante.
- Un microcontrôleur.
- Une interface d'affichage (LCD, Baregraphe).
- Un module de communication USB-UART.

Le quatrième chapitre concerne l'implémentation d'un système de supervision pour superviser et Controller le procédé, Ce logiciel est constitué par les trois parties suivants :

- Acquisition des données.
- Supervision.
- Commande.

En fin le cinquième chapitre, nous avons branché la carte de commande et testés notre système, pour cela nous développerons plusieurs algorithmes de commande.

Chapitre I

Description et présentation du process

Chapitre I : Description et présentation du process

I-1. Introduction :

Un four est une enceinte maçonnée ou un appareil, muni d'un système de chauffage puissant, qui transforme, par la chaleur, les produits et les objets. Dans un processus de production d'objet, le four permet le traitement d'objets par la chaleur [1].

Dans ce chapitre on s'intéresse à la description des fours électriques à résistance en générale, et du four disponible au niveau du laboratoire en particulier, avec les trois éléments qui le compose, à savoir son circuit de puissance à triacs (solid state relay), son bac chauffant, son circuit de conditionnement et ses capteur de températures.

I-2. Description du four électrique à résistance :

I-2-1. Généralités sur les fours électriques à résistance :

Les fours électriques à résistances constituent les équipements électrothermiques industriels les plus connus et les plus répandus. Ils continuent à faire l'objet de nombreux perfectionnements, fruits d'efforts soutenus de recherche et développement pour les faire évoluer afin de mieux répondre aux besoins exprimés par les industriels utilisateurs [2].

I-2-2. Constitutions du four :

Un four à résistances est essentiellement constitué des éléments suivants :

I-2-2-a. Une enceinte (chambre de chauffe) :

Réalisée à partir de matériaux à caractère isolant ou réfractaire dont le rôle est de réduire les pertes thermiques par transmission vers l'extérieur. Elle varie par sa forme, par le moyen de manutention de la charge ou son type d'exploitation.

I-2-2-b. Résistances électriques :

Disposées en voûte, en sole ou sur les parois latérales (pariétales) de l'enceinte. Leur alimentation est assurée par un circuit de puissance (solid state relay) à base de triacs. Un système de modulation de la puissance y est associé, piloté par un système de régulation de température.

I-2-2-c. La charge à traiter :

Caractérisée par sa forme, sa nature et sa température initiale, dans notre cas la charge à chauffé est un liquide.

I-2-3. Principe de fonctionnement :

La puissance électrique $P(W)$ dissipée dans l'élément chauffant de résistance $R(\Omega)$ est transformée en chaleur par effet joule.

Elle s'exprime par la loi de Joule

$$P = UI = RI^2 = \frac{U^2}{R} \quad (I.1)$$

Avec $U(V)$ et $I(A)$ respectivement les valeurs efficaces de la tension aux bornes de la résistance et du courant électrique qui la traverse, générer par le module de puissance.

Dans la pratique, les résistances électriques utilisées de manière classique ont une réactance faible devant R .

Les fours à résistances procèdent par chauffage indirect, la chaleur produite par effet Joule par les résistances est transmise à la charge par rayonnement et par convection [3].

I-3. Présentation du four de laboratoire :

I-3-1. Description du four de laboratoire :

Les fours électriques industriels sont très utilisés dans le secteur industriel, la chaleur est produite par des résistances. La figure I.1 représente la photo réelle du four disponible au niveau du laboratoire.

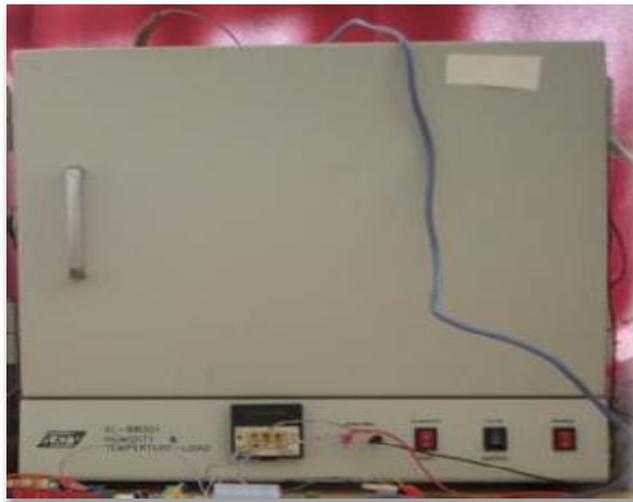


Figure I.1 : photo réelle du four du laboratoire.

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

Dans le mode manuel la consigne température est introduite par le clavier alors que dans l'autre mode le signal de commande est une tension continue comprise entre 0 et 10 V.

Il est constitué d'une résistance dont la tension d'alimentation est contrôlée par un triac, d'un banc ou on peut mettre le liquide et de deux capteurs de température.

L'un d'eux est relié à l'afficheur digital, l'autre au circuit de transduction.

La **figure I.2** présente les constituants du four du laboratoire.

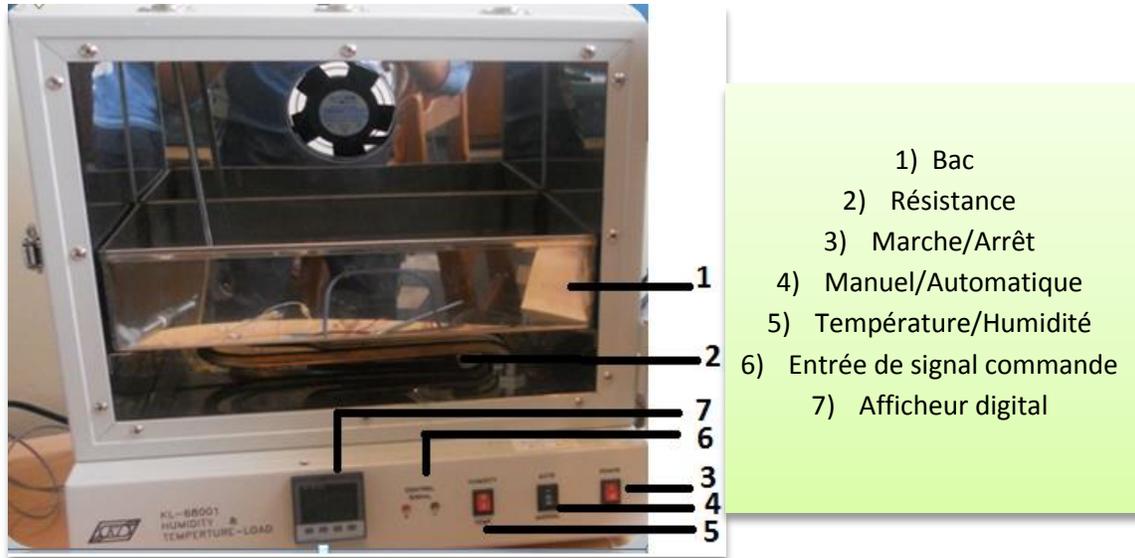


Figure I.2 : Constitutions du four du laboratoire.

I-3-2. Son circuit de puissance :

La structure du four avec son circuit de puissance est représentée par la **figure I.3**.

Elle est constituée essentiellement d'un circuit de puissance à Triacs (bloc 3) et un circuit électronique qui commande ces Triacs (bloc 1). La tension aux bornes de la résistance chauffante (bloc 2) est modulée par le signal d'entrée de la gâchette **U_g**.

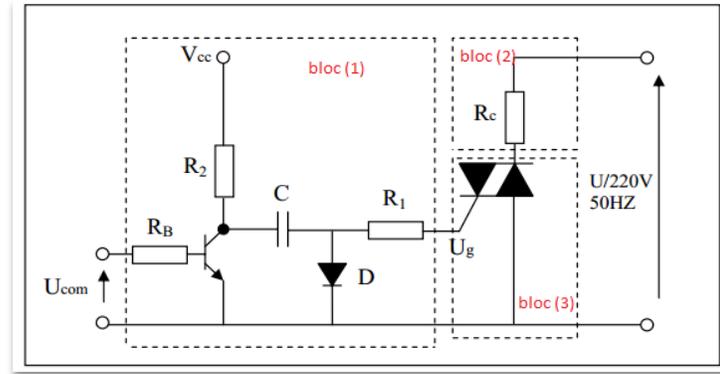


Figure 1.3 : Schéma simplifié d'une tension module à l'aide d'un triac

La Figure I.4 présente le schéma interne du circuit de puissance du four.

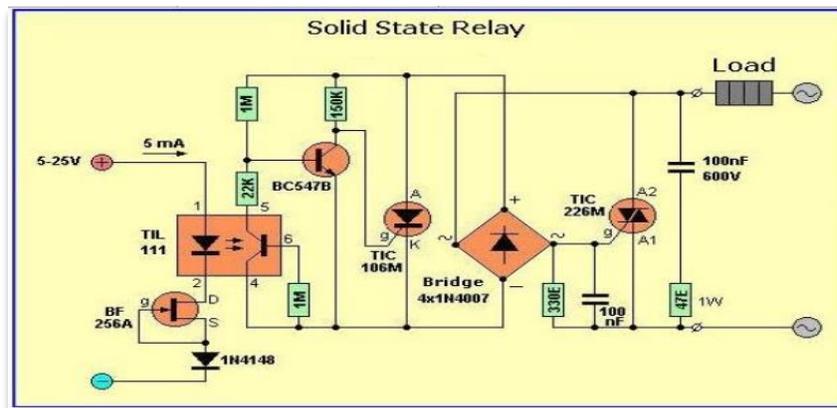


Figure I.4- Schéma interne du circuit de puissance du four

Les différents régimes de conduction des Triacs se divisent en quatre quadrants, qui sont montrés par la figure I.5. 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.

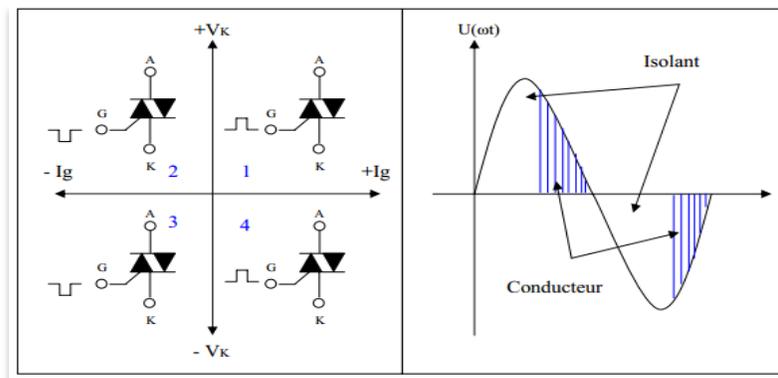


Figure.I.5- Différents régimes de fonctionnement du Triac

I-3-3. Présentation de capteur de température**I-3-3-a. Généralités sur les capteurs de température :**

Les sondes de température (ou capteurs de température) sont des dispositifs permettant de transformer l'effet du réchauffement ou du refroidissement sur leurs composants en signal électrique.

On compte deux grandes familles de sondes de température :

- Thermocouples
- Les sondes à résistance de platine

I-3-3-b. Capteur de température a semi-conducteur AD590 : [4]

Les capteurs AD590 sont des éléments sensibles qui convertissent la mesure de température en un courant de sortie proportionnel. La technologie de ces capteurs convient particulièrement aux applications de mesure et de régulation entre -55 et 150°C , là où la fiabilité, la linéarité et la précision des circuits électroniques est nécessaire.

Les AD590 peuvent servir à déterminer des valeurs minima, moyennes et différentielles de température ainsi que pour la compensation de soudure froide des thermocouples. Grâce à leur petite taille et leur réponse rapide, ils s'appliquent particulièrement aux circuits électroniques et aux radiateurs des systèmes électroniques. Il suffit de les alimenter pour mesurer la température absolue: ni linéarisation, ni amplification ni compensation de soudure froide ne sont nécessaires.

Caractéristiques:

- Tension maximale directe (E+ à E-): +44 V
- Tension inverse (E+ à E-): -20 V
- Tension par rapport la masse (boîtier et E+ ou E-): ± 200 V
- Température des bras: 300°C
- Alimentation : Plage de tension: +4 à +30 Vcc
- Courant nominal à 25°C ($298,2^{\circ}\text{K}$): $298,2 \mu\text{A}$

L'AD590 est un capteur à base de transistors, il exploite le changement de la tension avec la température. Si deux transistors opèrent à rapport constant de courant de collecteur alors la tension aux bornes de leur émetteur sera (kT/q) .

Avec :

$$I_T = \frac{2 k.T}{R. q} \log 8 \quad (\text{I.2})$$

On obtient la relation suivante :

$$\frac{I_T}{T} = 1\mu\text{A} / ^\circ\text{K} \quad (\text{I.3})$$

K : est la constante de Boltzmann.

q : la charge des électrons.

R : la résistance du pont égale à 368 Ω .

La tension résultante est directement proportionnelle à la température absolue. Grâce à des résistances cette tension est convertit en courant. La figure suivante montre le schéma électrique de l'AD590.

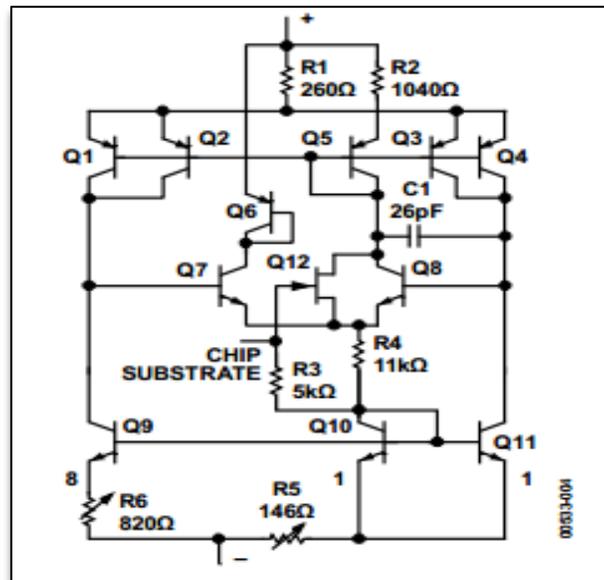


Figure I.6 : schéma interne du capteur AD590

Q8 et Q11 sont les transistors qui produisent la tension proportionnelle à la température, et les résistances R5 et R6 convertit cette tension en un courant.

Cependant, il est nécessaire d'utiliser un circuit de conditionnement.

I-3-4. Circuit de conditionnement de Capture

C'est un circuit de conditionnement du capteur de température AD590, il compose de 3 étages sont:

- Conversion Courant-Voltage.
- Filtre du second d'ordre.
- Un diviseur de tension.

La Figure I-7 présente la structure du circuit de conditionnement.

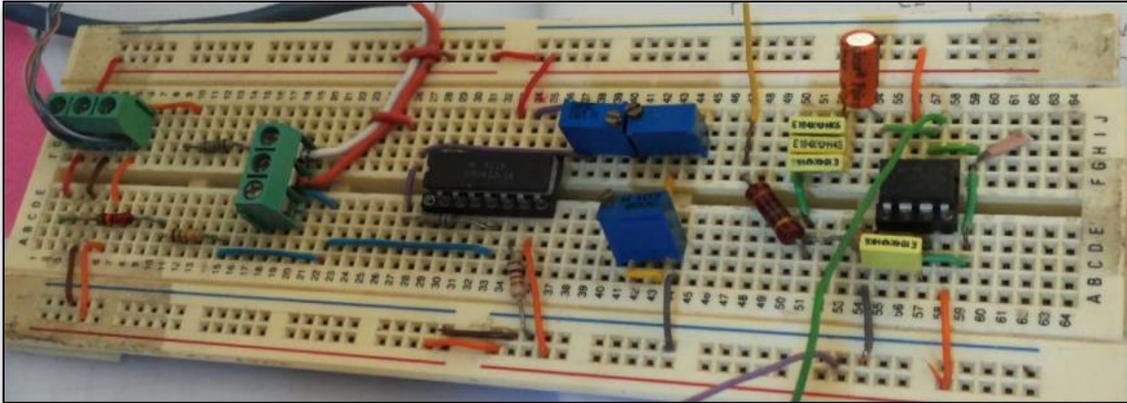


Figure I.7 : La structure du circuit de conditionnement

I-3-4-1. Conversion Courant –Voltage:

C'est un circuit à base d'amplificateur opérationnelle, il permet de convertir le courant délivré par le capteur AD590 à une tension avec un gain en sortie de (1mV/kelvin)

La figure I.8 présente le schéma de circuit de conditionnement du capteur AD590 :

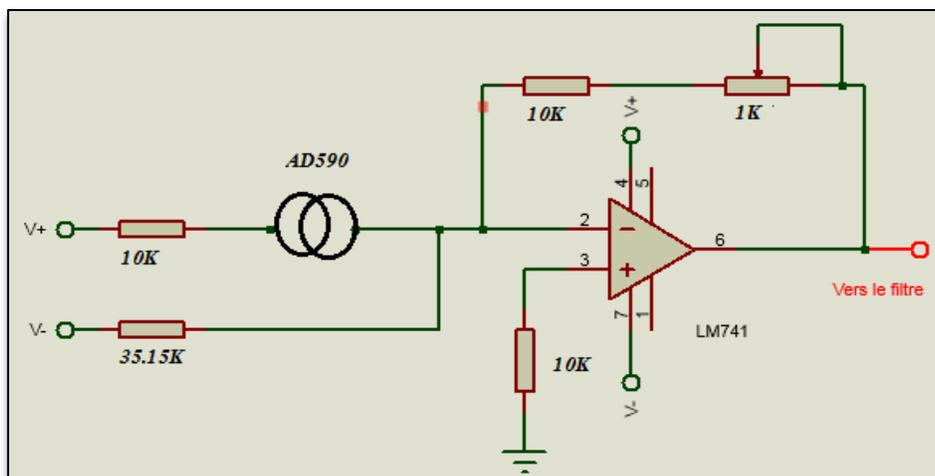


Figure I.8 : schéma de circuit de conditionnement de capture

I-3-4-2. Filtre de second d'ordre:

Le signal délivré par le capteur est fortement bruité il faut donc le filtré. Plusieurs solutions sont possibles, la plus simple consiste à mettre un filtre de second ordre.

La figure I.9 présente le schéma d'un filtre du second d'ordre.

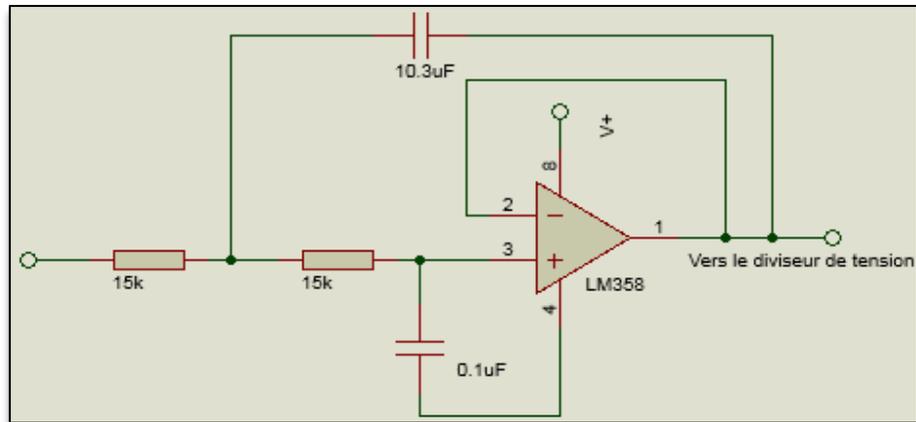


Figure I.9 : Schéma d'un filtre du second d'ordre

I-3-4-3. Un diviseur de tension:

C'est un montage diviseur de tension à base amplificateur opérationnelle qui permet d'assurer une gamme de tension entre 0 et 5v.

La figure I.10 présente le schéma d'un diviseur de tension.

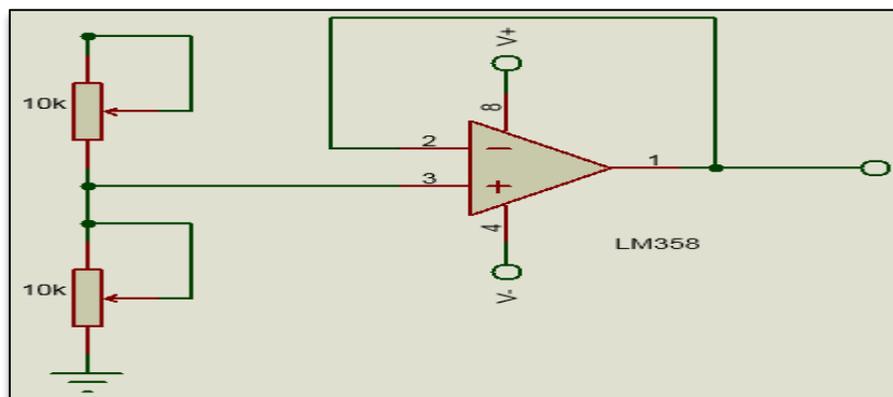


Figure 1.10 : schéma d'un diviseur de tension

I-4. Conclusion :

Dans ce chapitre, nous avons effectué un aperçu général sur les fours électriques à résistance ainsi que son principale de fonctionnement, après cela nous avons élaboré une présentation sur le four du laboratoire, son circuit de puissance, son capteur et son circuit de conditionnement avec filtrage.

Chapitre II

Identification de la dynamique du four

Chapitre II: Identification de la dynamique du four

II-1. Introduction :

L'identification d'un procédé est définie comme la détermination, basée sur la connaissance des entrées et des sorties du procédé, d'un modèle appartenant à une classe spécifiée, équivalente au procédé.

Le modèle auquel nous nous intéressons est un modèle dynamique non linéaire de type fonction de transfert (modèle de représentation) et qui, souvent, décrit le comportement du procédé autour d'un point de fonctionnement particulier, il ne prend en compte que les petites variations autour de ce point. [5]

II-2. Mise en œuvre de l'identification :

La conception et la commande des fours électriques nécessitent le calcul de certains modèles mathématiques. On fait la modélisation en utilisant MATLAB via la fonction «IDENT» qui fait l'apparition d'une interface à plusieurs paramètres et des listes des formes de systèmes à identifier, il nécessite seulement d'introduire le vecteur des sorties inspirés d'une expérience réelles sur notre four du laboratoire, et de sélectionner la structure du modèle selon sa réponse réel.

Les étapes d'identification :

- a) Au premier lieu on met le système de commande en mode manuel (en boucle ouvert), On attend que le système soit bien stabilise.
- b) On place un oscilloscope à la sortie de circuit de conditionnement du capteur AD590 afin de relever soigneusement et enregistrer les réponses à ces signaux.
- c) On choisir l'emplacement de fichier pour l'enregistrement des données de mesure afin d'exploiter facilement par le MATLAB.
- d) On attaque le système par un échelon (signal d'entrée 'tension en voltage' de 5V qui est équivalent en température de 50°C).
- e) Le four va réagir à cette entrée en changeant sa valeur de température, durant son fonctionnement en boucle ouvert notre logiciel reçoit la valeur de la température mesure (en degré Celsius) à chaque (1 s).
- f) Ces valeurs mesurée va enregistrer dans un fichier « .m » de manière qu'on obtenir chaque valeur dans une ligne.
- g) chaque nouvelle mesure va représenter dans une courbe pour faciliter l'exploitation en temps réel.

- h) A la fin d'expérience, on obtient un fichier **data.m**.
- i) Introduction des données sur le MATLAB :

```
clear all;
clc;
%% pour charger le vecteur de sortie
load data.m
%% enregistrer ces données dans un vecteur
Temp=data;
```

- j) Tracer les résultats par la fonction « plot ».

```
t=0:1:(length(Temp)-1);
plot(t,Temp);
```

Les courbes obtenues :



Figure II-1: réponse du four relevé par oscilloscope après filtrage

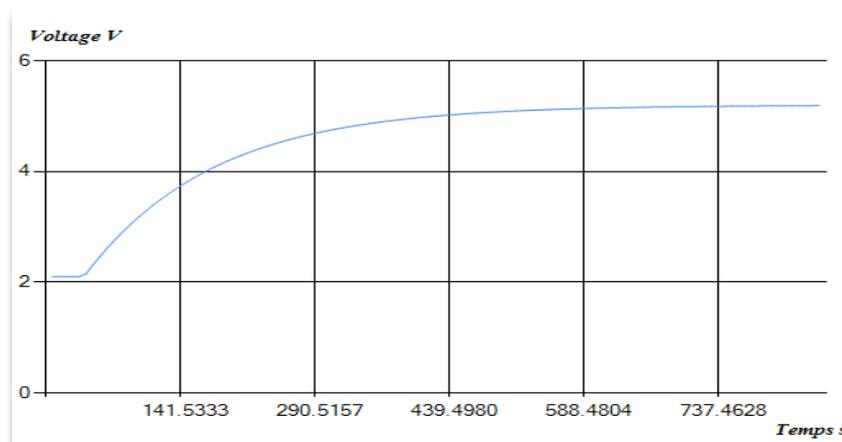


Figure II-2: la réponse du four relevé par notre interface graphique.

- k) Interopération la forme des réponses pour choisir une structure de modèle (ordre, retard)
- l) Faire l'identification par l'un des méthodes proposées.
- m) Valider le modèle.

II-3. Identification par la méthode graphique de STREJC :

II-3-1. Méthodologie : [6]

Cette méthode peut s'appliquer aux systèmes dont la réponse indicielle ne présente pas de dépassement. On identifie le système à une fonction de la forme.

$$G(s) = \frac{K}{(1+Ts)^n} * e^{-\tau s} \quad (\text{II.1})$$

Les paramètres à identifier sont donc :

- le gain statique **K**
- le retard τ
- la constante de temps **T**
- et l'ordre **n**

Et pour faire l'identification nous avons besoin de Tableau de Stretch suivant :

n	Tu/T	Ta/T	Tu/Ta
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

Tableau II-1 : données de la méthode STRETCH

Pour identifier le système, la méthode peut se décomposer de la manière suivante :

- Premièrement selon l'allure de la réponse, on choisit la structure puis les paramètres du modèle.
- On commence par déterminer le gain statique en mesurant la valeur finale de la sortie et en la divisant par la hauteur de l'échelon d'entrée :

K (sans unité) est donné par :

$$K = \frac{\Delta Y}{\Delta X} = \frac{5.2-2.1}{5} = 0.62 \quad (\text{II.2})$$

- On trace le mieux possible la tangente au point d'inflexion de la réponse indicielle.
- La tangente découpe un segment Ta sur l'axe des temps au bout d'un certain temps Tu, et on relève du graphe les valeurs : Tu=40s Ta=295
- On calcule le rapport Tu/Ta et on choisira du tableau II-1, la valeur de n entier immédiatement inférieur. On a Tu/Ta=0.119 alors la valeur de n entier le plus proche inférieur sera égale à n=2.

- Connaissant la valeur de n , T_a ou T_u on détermine la constante de temps T à l'aide de l'une des deux dernières colonnes du tableau II-1. Dans notre cas, pour $n=2$ $T_a/T=2.72$ alors $T = T_a/2.72=108.46s$.
- D'après le tableau II-1 $T_u/T=0.28$ ce qui donnerait une valeur de $T=142.85s$ de la valeur mesurée réel est $T_u=108.46s$ On peut donc déduire le retard

$$t = T_{\text{Réal}} - T_{\text{théorique}} = 34.69s$$

La fonction de transfert du modèle de stretch est la suivante :

$$G(s) = \frac{0.62}{(1 + 108.46s)^2} * e^{-34.69s}$$

II-3-2. Validation du modèle obtenue :

Après l'obtention du modèle on procède à la validation du modèle en comparant la réponse expérimentale du système et la réponse simulée du modèle. Et pour cela on calcule l'erreur d'identification en déterminant la différence entre les deux modèles.

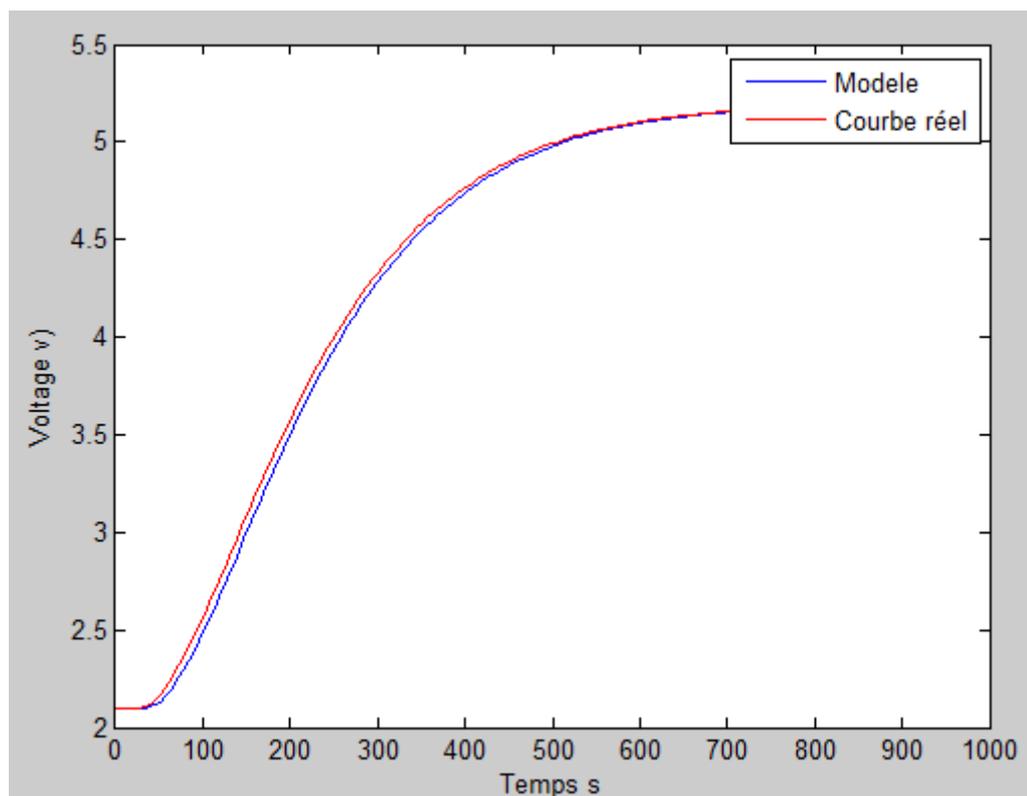


Figure II-3: comparaison de la réponse réelle et avec celle identifiée

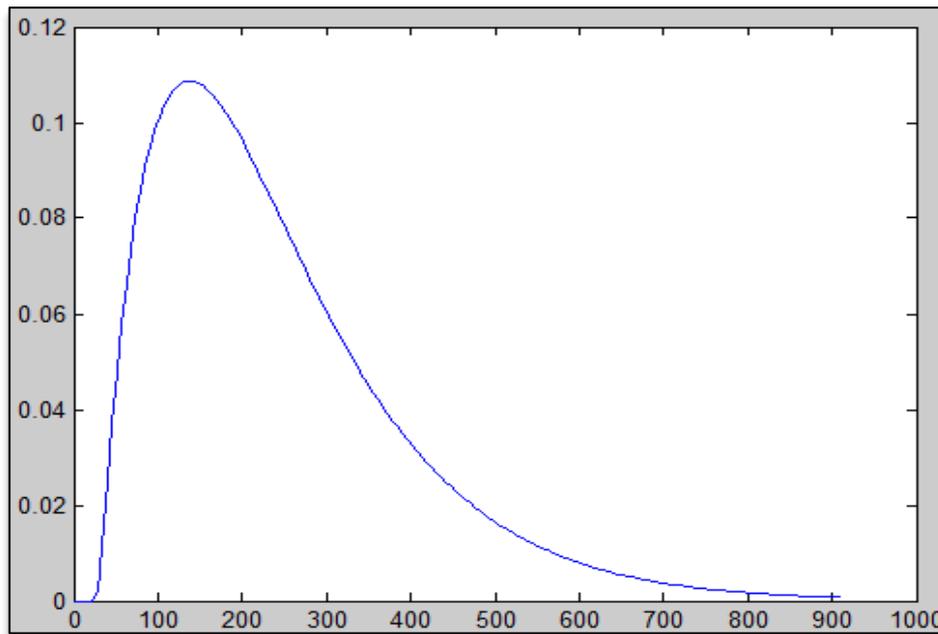


Figure II-4 : Erreur d'identification

On détermine alors l'erreur relative maximale qui doit être inférieure à 10%. L'erreur relative d'identification est donnée par la relation suivante :

$$Er = \frac{E_{\max}}{\Delta Y_{\max}} * 100 \quad (\text{II.3})$$

Application :

$$Er\% = \frac{0.011}{3.1} * 100\% = 3.54\%$$

Puisque l'erreur obtenue $Er < 10\%$ nous considérons que le modèle est relativement acceptable.

II-4. Identification par le Toolbox « ident » sous MATLAB : [7] [8] [9]

Le toolbox « système d'identification » de MATLAB est une interface utilisateur graphique pour les méthodes d'identification implémentées sous MATLAB.

- Permet d'élaborer des modèles simples à partir de données de mesure réalisées sur le système physique.
- Permet de comparer différentes méthodes d'identification.
- Permet d'identifier des systèmes multi-variables.
- Permet la construction de modèles structures ou définis par l'utilisateur.
- Détermination de structures de modèles.
- Estimation de spectres.
- Algorithmes adaptatifs et récursifs.
- Utilisation de SIMULINK et modèles continus.

La figure suivant représente les sections principales d' « ident » :

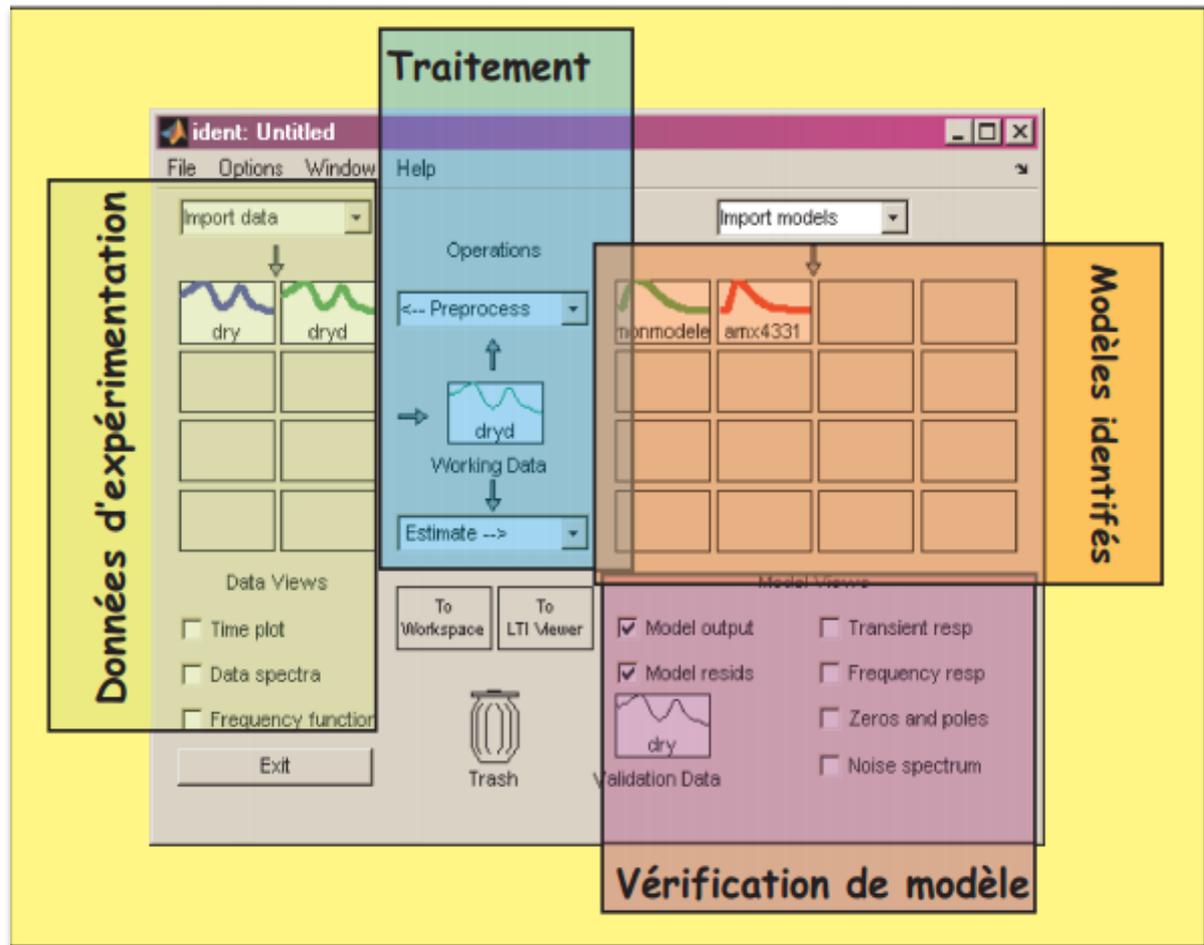


Figure II-5 : représente les sections principales de l'outil « ident ».

II-4-1. Méthode d'utilisation du toolbox :

Après avoir recueilli les données nécessaires à l'identification, on les définit sous forme de deux vecteurs : celui de l'entrée $u(k)$ et celui de la sortie $y(k)$ dans le workspace de MATLAB. On lance alors le Toolbox « système identification ». on pourra résumer la méthode d'utilisation du toolbox en trois étapes principales qui sont montrées par les trois figures suivantes.

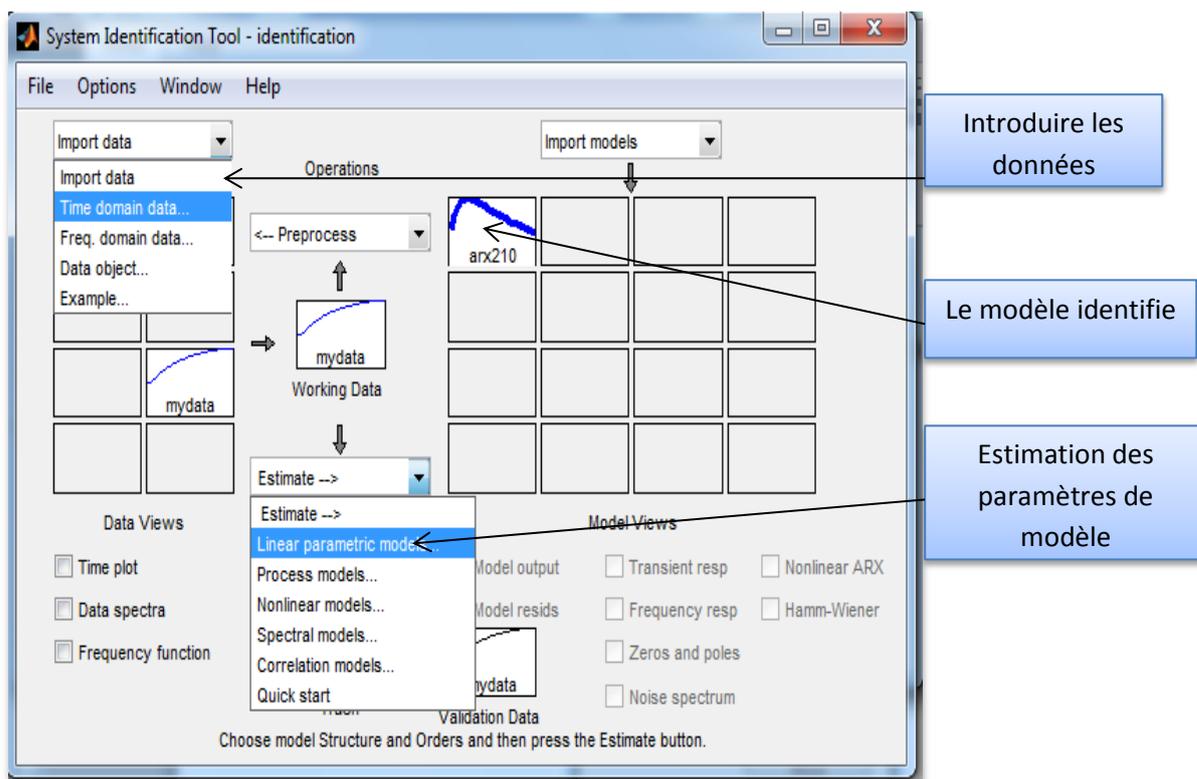


Figure II-6 : Schéma synoptique de l'identification par le toolbox « ident ».

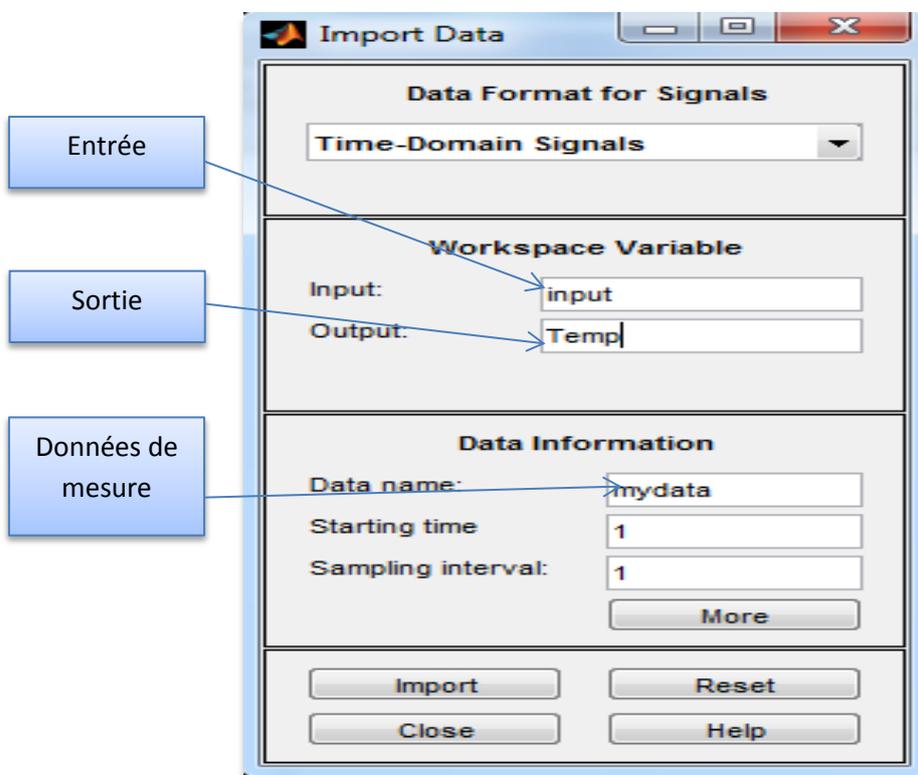


Figure II-7: introduction des données entrées/sorties sur le toolbox

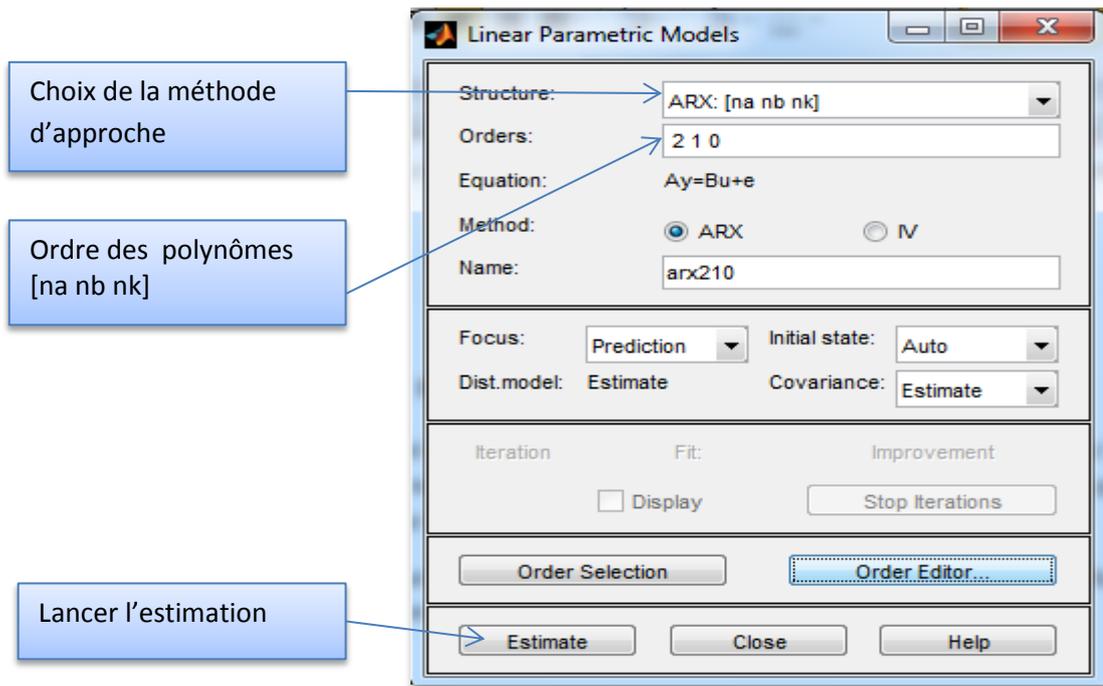


Figure II-8 : choix de structure de modèle

Etape 1 : importer les données définies précédemment dans le workspace avec l'onglet « import data » cela nous permettra aussi de visualiser les courbes du système réel $U(t)$ et $Y(t)$ (voir **figure IV-9**).

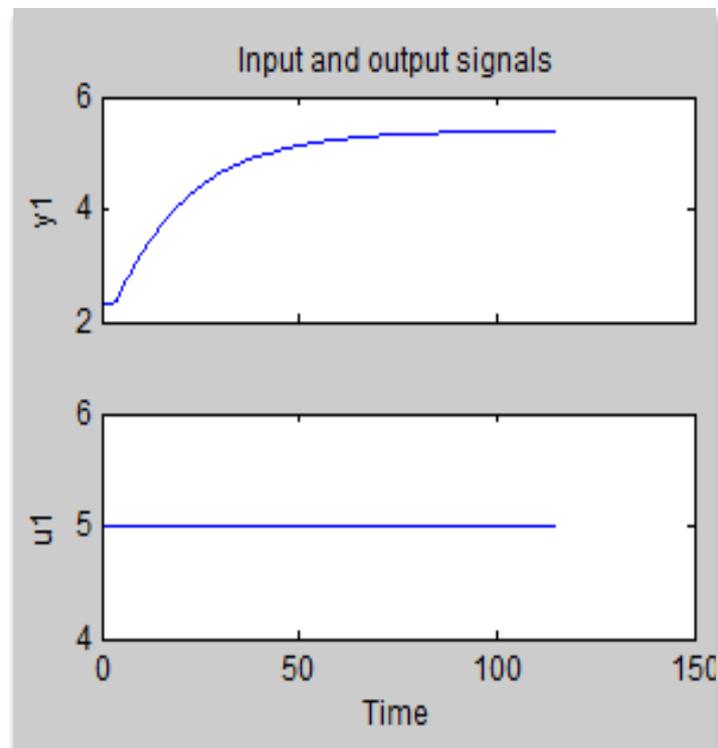


Figure II-9 : Courbes du système réel.

Etape 2 : après avoir choisi la structure d'identification et l'ordre du modèle dans l'onglet « linear parametric models ». Dans notre cas on a choisi donc la structure ARX et l'ordre des polynômes $A=2$, $B=1$ et le retard $=0$, on lance l'estimation du modèle.

Après l'estimation nous pourrions visualiser les courbes $Y(t)$ réel et $Y_m(t)$ de l'identification **figure IV-10**.

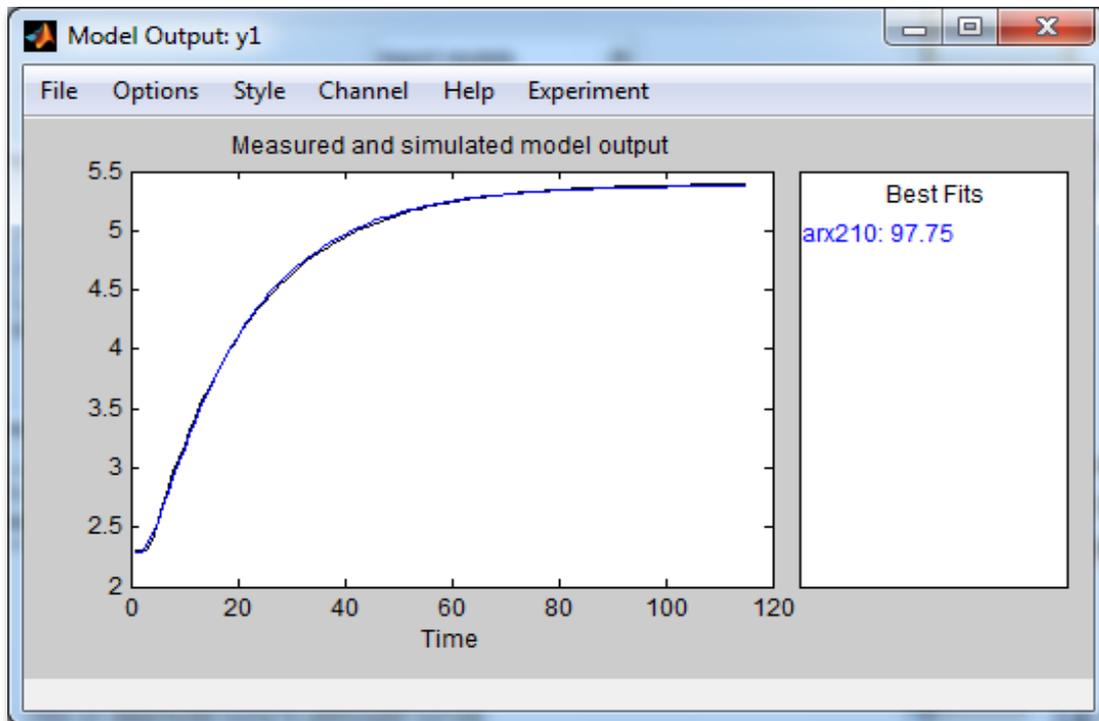


Figure II-10 : Courbes de réponse du système réel et celle du modèle mathématique.

Etape 3 : après avoir identifié le système, on importe les résultats dans le workspace de MATLAB pour avoir accès aux polynômes A , B du modèle mathématique et on trouve dans notre cas :

$$A(q) = 1 - 1.622q^{-1} + 0.6394q^{-2}$$

$$B(q) = 0.01913q^{-1}$$

Après on obtient :

$$G(z) = \frac{0.01913z^2}{z^2 - 1.622z + 0.6394}$$

La **figure II-11** montre la réponse indicielle du modèle mathématique $G(s)$

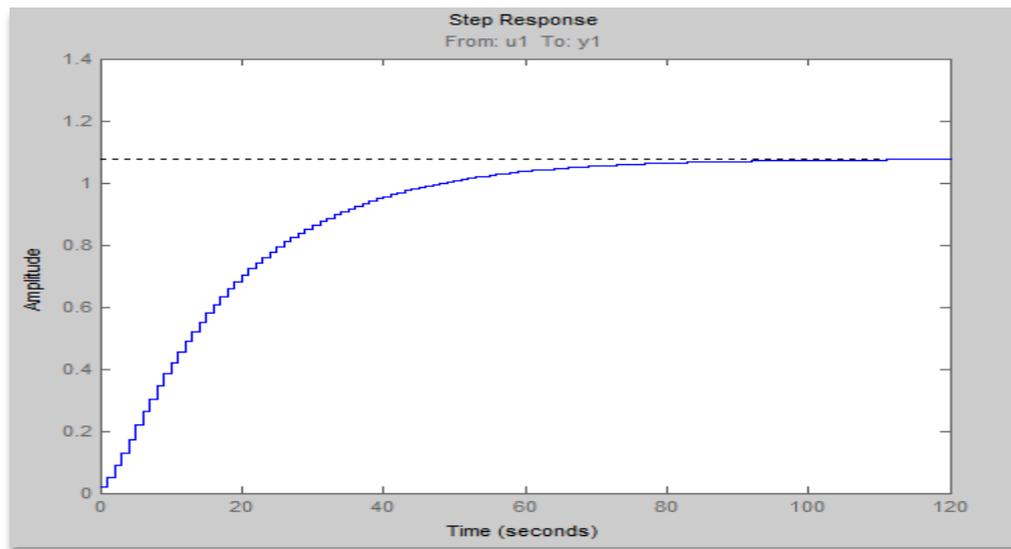


Figure II-11 : Réponse indicielle du model identifie

II-4-2. Valider le modèle :

A la lumière de la figure N° 10, On conclut que le modèle identifié est plus ou moins représentatif du système étudié.

II-5. Conclusion :

Dans ce chapitre, nous avons présentés des méthodes d'identification du modèle utilise, et on à vue que chaque méthode donne des résultats acceptables. Mais parmi ces méthodes nous avons retenu la méthode de STREJC qui semble assez bien fonctionne car la réponse indicielle est proche de celle du système et l'erreur calculée est acceptable d'une part et d'autre part sa simplicité, et sa précision.

Chapitre III

Structure de la carte électronique

Chapitre III: Description de la carte de commande réalisée.

III-1. Introduction :

Dans le chapitre précédent, nous avons élaboré une description de notre processus représenté par le four électrique à résistance de laboratoire avec ses constituants (circuit de puissance, circuit de conditionnement et ses capteurs).

Dans ce chapitre nous allons étudier la partie Hardware du projet en présentant la carte électronique de commande, Nous exposons en détaille les éléments principaux de notre carte et leurs schémas de brochages.

III-2. Schéma bloc de la carte électronique de communication PC-PIC :

Le principe de la commande avec boucle de mesure est présenté en **figureIII.1**. En effet nous retrouvons le calculateur numérique assuré par un ordinateur qui fournit de la consigne température (T_c) ainsi de visualiser et enregistrer la valeur de température de four.

Le microcontrôleur a pour le but d'assurer deux taches, l'un pour transférer les mesures de capteur (T_m) vers l'ordinateur, et l'autre pour affecter le signal de commande U_{com} .

Le four électrique équipé par des résistances électriques disposées en sole de l'enceinte apportant l'énergie calorifique, La puissance nominale des résistances est modulée grâce à un bloc de puissance de type gradateur, Celle-ci est contrôlée par une tension de commande continue (U_{com}) comprise entre 0 et 10Volts, par conséquence, la variation de la tension efficace (U_{eff}) appliquée aux bornes des résistances électriques.

Enfin, Un capteur transmetteur de type AD590 et son circuit de conditionnement permettent de mesurer la température réelle du four (T_m) travers son circuit de conditionnement.

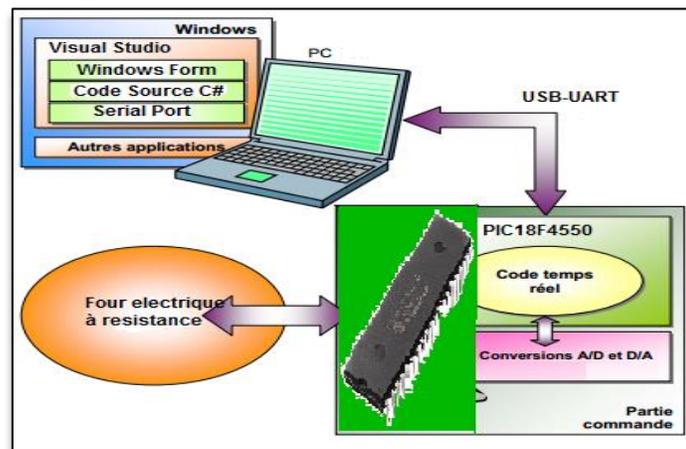


Figure III.1 : Architecture matérielle et logicielle du dispositif de contrôle

Le schéma bloc de la carte électronique est représentée en **figure III.2** comporte essentiellement les éléments suivants :

- PC Calculateur numérique type LAPTOP (1) : C'est l'organe principal qui assure la supervision et la commande du système.
- Microcontrôleur PIC 18F4550 (3) : C'est l'organe responsable à l'acquisition des données, ainsi que l'application de signal de commande.
- Module de communication USB-UART (2) : Il représente le lien entre le calculateur numérique et le microcontrôleur.
- Circuit de conditionnement + filtre second ordre (4) :
- Circuit de puissance (5) :
- Capteur de température (6).
- Four électrique (7).
- Affichage LCD (8).
- Baregraphe (9).

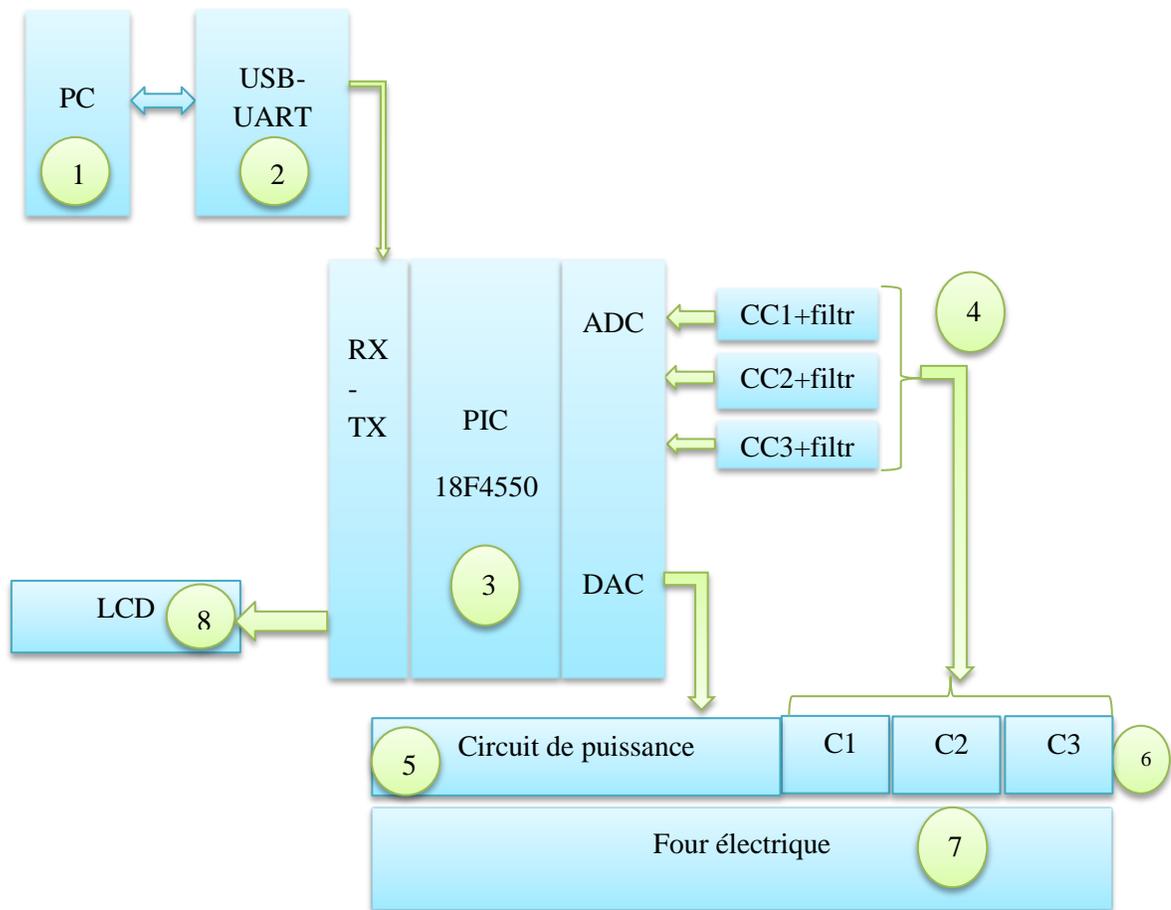


Figure III.2 : Schéma bloc de la carte électronique

III-3. Schéma réel de la carte électronique piloté par USB-UART :

Notre carte de commande Entrées-sorties analogique est constituée essentiellement par les organes suivants (voir **figure III-3**) :

- L'organe d'**Acquisition** de données (circuits de conditionnement des capteurs).
- L'organe de **Commande** (Le microcontrôleur PIC 18F4550).
- L'organe de **Visualisation** (Un afficheur LCD 2x16).
- L'organe de **Communication** avec l'ordinateur (USB-UART).
- L'organe de **Conversion ADC /DAC** (ADC inclut dans le PIC).

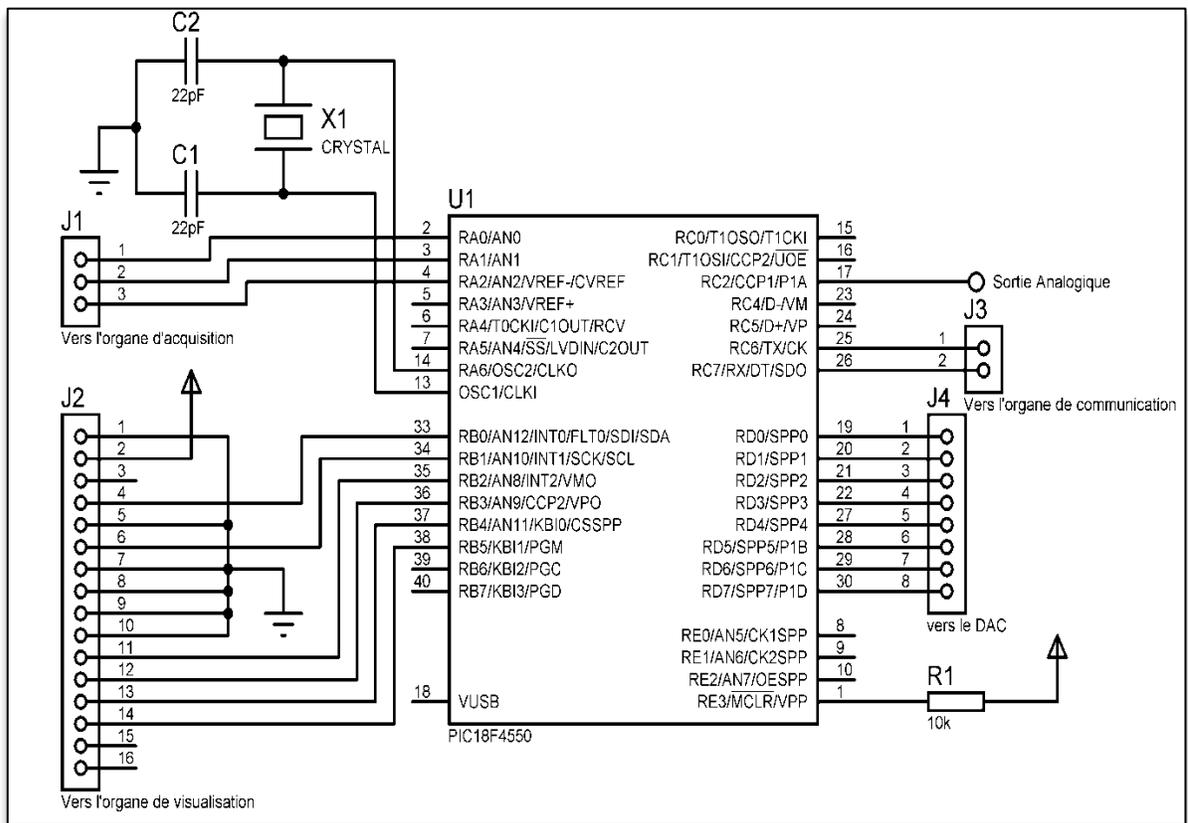


Figure III.3 : Répartition des broches du PIC vers les périphéries entrées/sorties.

III-3-1. Organe de commande :

C'est le noyau de calcul temps réel lui permettent d'exécuter le programme développer sur l'ordinateur, le microcontrôleur PIC18F4550 est physiquement connecté au procédé à contrôler par biais des organes de conversion ADC /DAC. [10]

III-3-2. Organe d'Acquisition de données :

Il représente les circuits de conditionnement des capteurs utilisés pour transmettre le signal obtenu au PIC en temps réel, comme on a vu dans le premier chapitre, on a utilisé trois capteurs de type AD590.

La Figure III-4 présente le capteur de température AD590.



Figure III.4 : photo réelle du capteur de température AD590.

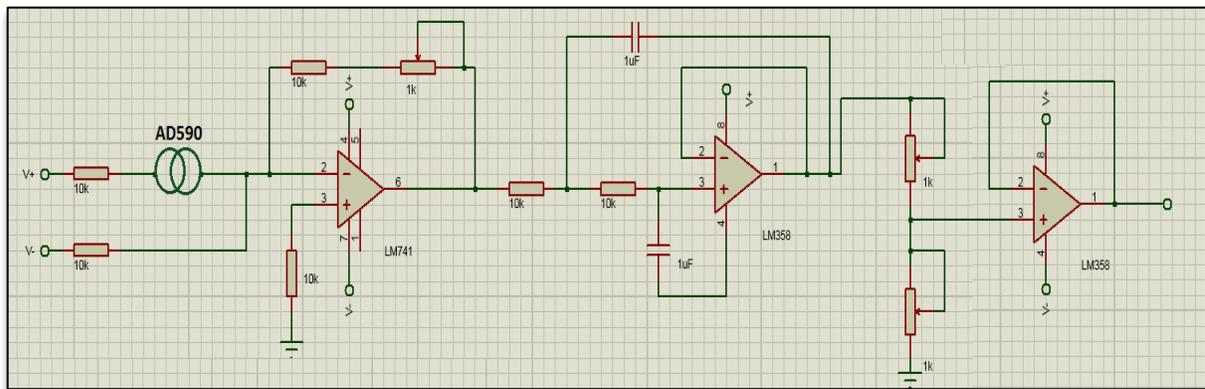


Figure III.5 : Schéma électronique du circuit de conditionnement du capteur AD590.

Le schéma réel du circuit de conditionnement est présenté en **figure III-5**.

On remarque que notre circuit de conditionnement se divise en trois blocs :

- Convertisseur courant tension qui convertie le courant en tension et la normalise on ajustant la résistance variable RV1 afin d'obtenir un pas de $0.1V/^{\circ}C$.
- Filtre passe bas du 1^{er} ordre qui sert à filtrer les parasites.
- Un diviseur de tension 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 incluse dans la marge de sécurité de notre PIC.

III-3-3. Organe de visualisation (LCD 2 x 16) : [11]

L'un des organes connectés avec le PIC est un afficheur LCD (liquid cristal display) alphanumérique, le but de son utilisation est de visualiser les valeurs en temps réel de la température mesurée par les capteurs ainsi que la température de référence et même les valeurs des paramètres de nos régulateurs telle que l'action proportionnelle, l'action intégrale...etc. Ces composants sont des modules assez complexes, et permettent d'afficher les textes et les symboles avec une relative simplicité.

En **figure III.6** et **III.7** nous présentons les photos de ces afficheurs alphanumérique :



Figure III.6 : une photo réelle d'afficheur LCD

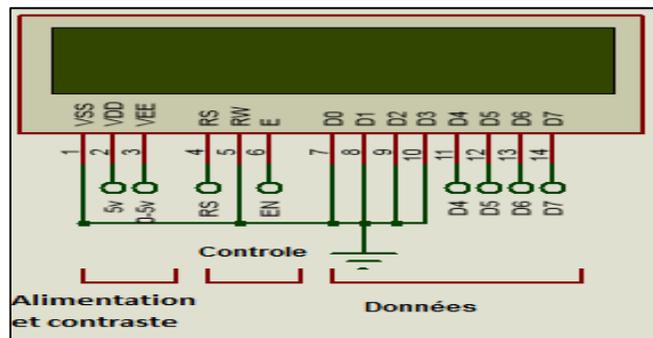


Figure III.7 : schéma de brochage minimal d'afficheur LCD.

III-3-4. Organe de communication:[10]

La présence d'un USB-UART permet la communication entre le PC et le microcontrôleur .la liaison crée est non seulement utilise pour charger sur le programme de contrôle développé avec VISUAL STUDIO sur le PC mais également pour interagir en permanence avec le microcontrôleur et récupérer toutes les donnes enregistrées.

La figure III.8 montre l'aspect de ce circuit USB-UART

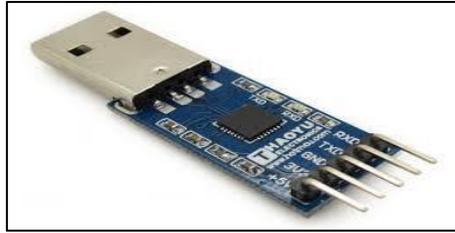


Figure III.8: photo réelle du circuit USB-UART.

III-3-5. Organe de conversion ADC et DAC :

III-3-5-a. Le convertisseur analogique numérique ADC [12]:

Le signal venant de l'environnement du procédé à travers ses capteurs est un signal analogique, alors il devra être converti en un signal digital pour qu'il puisse exciter l'organe de commande (PIC).

Ce convertisseur analogique-numérique se trouve par défaut dans la structure interne du PIC et plus précisément dans les ports A, E, B.

Le PIC18F4550 dispose de **13** canaux de conversion analogique/numérique **PORTA**, **PORTE** et **PORTB**, qui permettent un échantillonnage sur **10 Bits**. Les pins de ces ports peuvent être utilisés comme de simples entrées-sorties digitales ou bien comme entrées des unités analogiques AN0, AN1, ..., AN12. L'utilisation de ces pins comme entrées analogiques avec réglage de sa fréquence propre de conversion s'obtient à l'aide des registres situés en RAM suivants:

- ADCON0.
- ADCON1.
- ADCON2.
- ADRESL.
- ADRESH.

➤ **ADCON0:**

Ce registre en BANK0 permet de configurer les paramètres de fonctionnement des convertisseurs comme l'indique le tableau III.1.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
Read as "0"		Analog Channel Select bits				A/D Conversion Status bit	A/D enable bit

Tableau III.1: bits de configuration du registre ADCON0

➤ **Registre ADC0N1:**

Le registre situé en **BANK1** permet de configurer les broches des PORTA, PORTE et PORTB en entrées digitales ou analogique et de sélectionner la source de voltage du référence, Tableau III.2.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
--	--	VCFG0	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0
Read as "0"		V _{REF-} source	VREF+ source	A/D Port Configuration Control bits:			

Tableau III.2 : bits de configuration du registre ADCON1

➤ **Registre ADCON2:**

Le registre situé en **BANK1** permet de sélectionner le temps et la fréquence d'échantillonnage pour le convertisseur et de sélectionner les registres où doivent être sauvegardés des résultats sur 10bits, Tableau III.3.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADFM	--	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0
A/D Result Format Select bit	Read as "0"	A/D Acquisition Time Select bits			A/D Conversion Clock Select bits		

Tableau III.3 : bits de configuration du registre ADCON2

III-3-5-b. Le convertisseur numérique analogique DAC0800 [13] :

Après étude et élaboration du circuit de sortie binaire, nous étudions maintenant le circuit de conversion DAC (la **figure III.9**). Celui-ci nous permettra de générer une tension au moyen de données binaires.

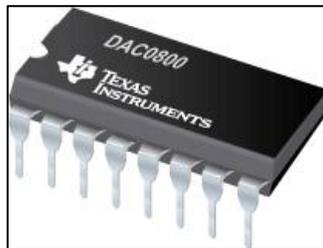


Figure III.9 : photo réelle du DAC0800

Les applications de ce dernier sont multiples. Ce DAC que nous avons utilisé est référencier sous la dénomination DAC0800 par la société TEXAS INSTRUMENTS.

Ce boîtier dispose d'une maille dans laquelle l'utilisateur fait circuler un courant de référence, que l'on notera I_{ref} (figure III.10). Ce courant doit être égale a 2 mA selon les donnes de fabricant.

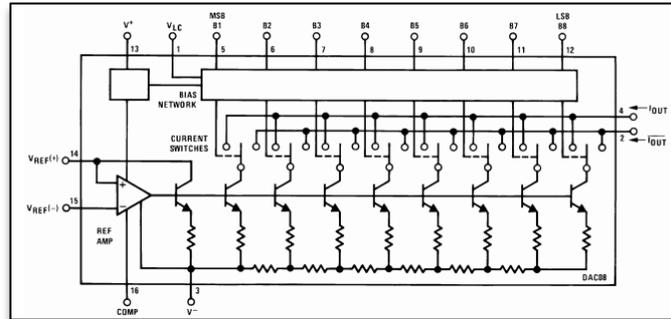


Figure III.10 : Structure interne du DAC0800

Les signaux logiques (B0 à B7) font commuter les interrupteurs soit sur I_{out} soit sur \bar{I}_{out} (\bar{I}_{out} est le complément de I_{out}) ce qui nous fournit à la sortie deux courants (I_{out} , \bar{I}_{out}).

De manière générale, ce circuit de conversion permet de générer un courant I_{out} en fonction des données binaires (B0 à B7) a son entrée. Le courant maximum est de 2mA pour une entrée pleine échelle sur 8 bits.

Le brochage ainsi que l'ensemble des composants nécessaires à son fonctionnement sont présentes en figure (II.11).

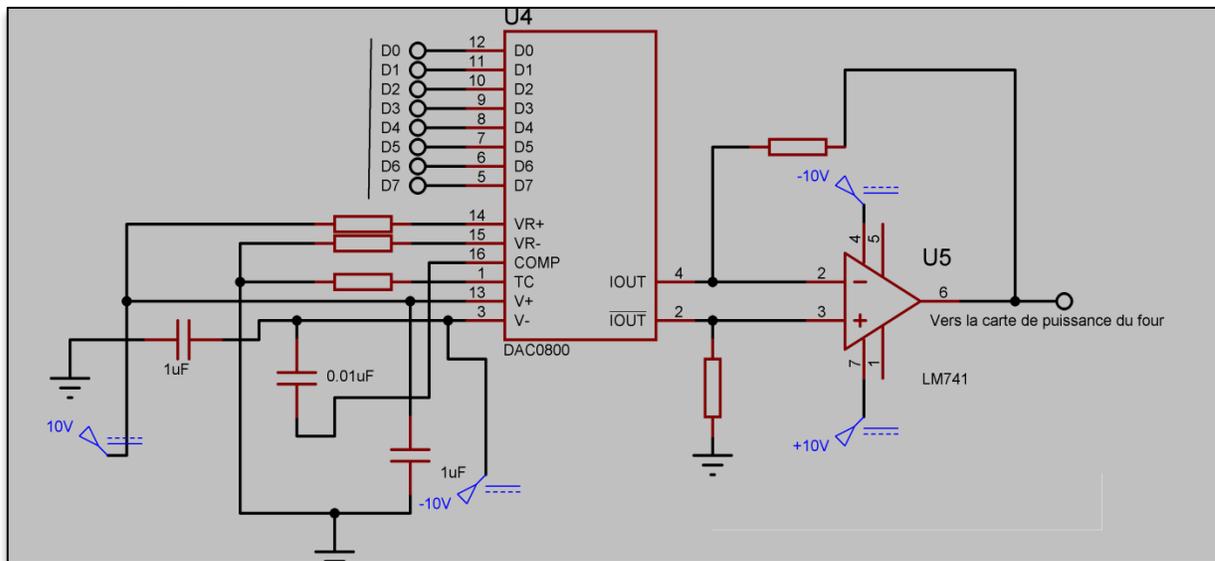


Figure.II.11 : Schéma de branchement réel du DAC0800 avec l'ensemble Des composants nécessaires.

Ce circuit converti une donnée numérique de 8 bits a une grandeur analogique en 150ns, il nécessite des tensions de référence de +10Volts et -10Volts ainsi qu'un amplificateur opérationnel LM741 afin de transformer le courant I_{out} en voltage exploitable par le process.

Les valeurs des composants passifs (résistances, capacités) sont fixées pour un courant de référence de 2mA selon la recommandation du fabricant [14].

III-3-6. Simulation et réalisation de la carte de commande :

La figure suivante représente le schéma électronique global de la carte de commande réalisée sous PROTEUS-ISIS.

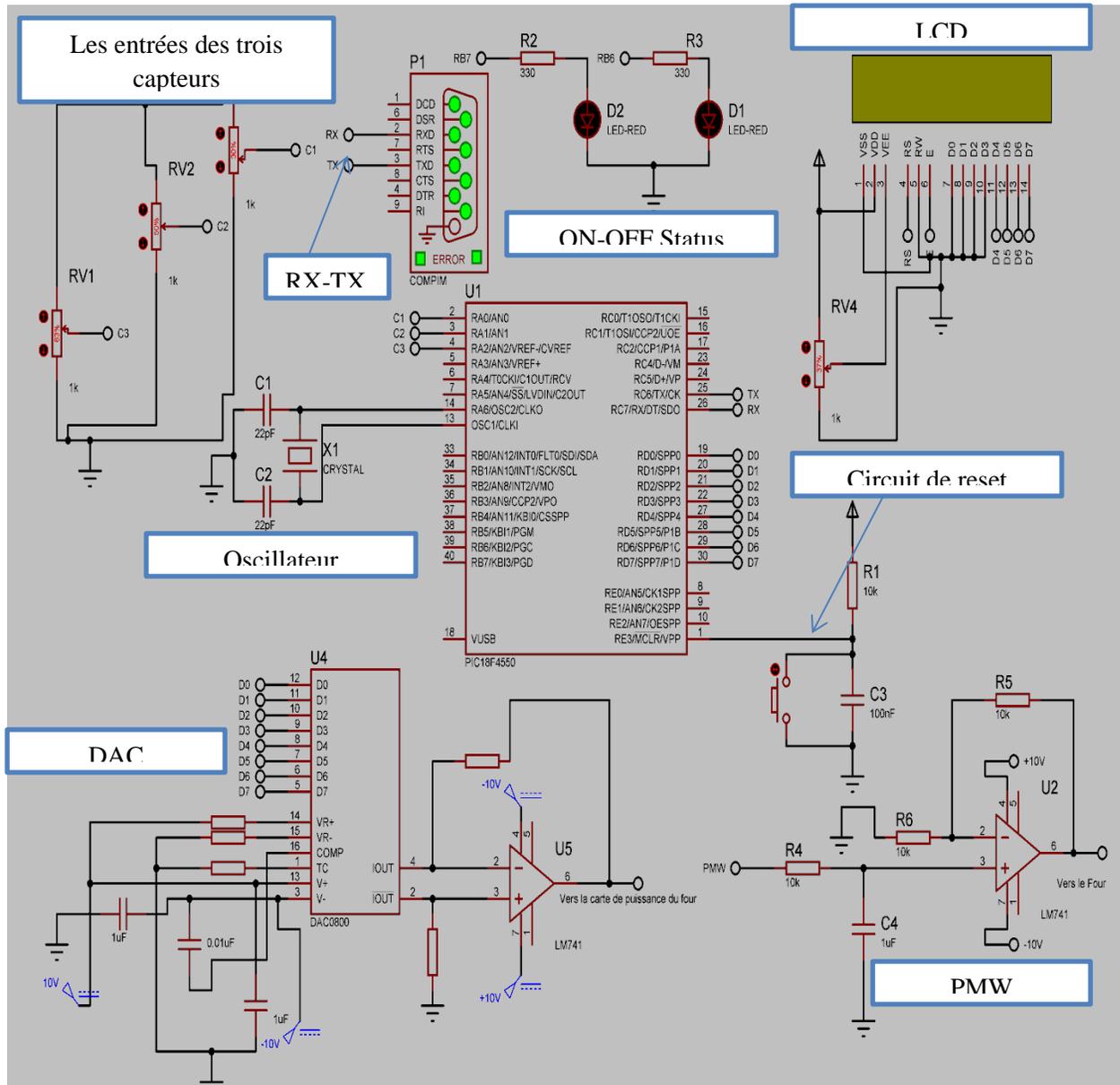
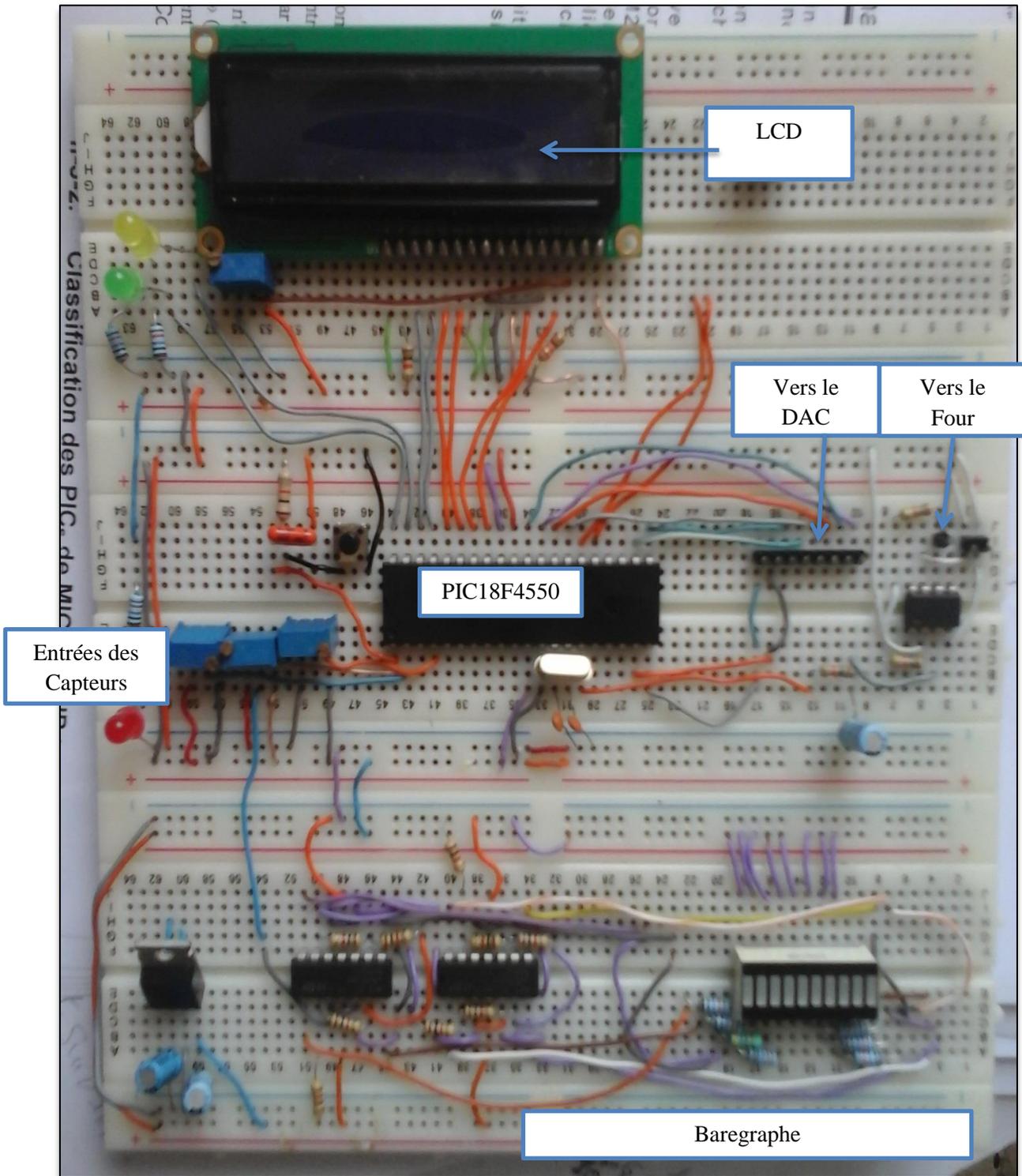


Figure III.12 : Schéma de brochage final de la carte de commande.



La figure III-13 présente la photo réelle de la carte de commande

III-4. Présentation de port de communication UART :

III-4-1. Introduction : [15]

Un UART, pour *Universal Asynchronous Receiver Transmitter*, est un émetteur-récepteur asynchrone universel de fabricant chinois [18].

En langage courant, c'est le composant utilisé pour faire la liaison entre l'ordinateur et le port série. L'ordinateur envoie les données en parallèle (autant de fils que de bits de données). Il faut donc transformer ces données pour les faire passer à travers une liaison série qui utilise un seul fil pour faire passer tous les bits de données.

Les UART sont généralement intégrés dans des composants comme des microcontrôleurs. Ils ne sont dans ce cas plus un composant à proprement parler, mais une fonction périphérique du composant.

III-4-2. Constituants d'une trame UART : [16]

Une trame UART est constituée des bits suivants :

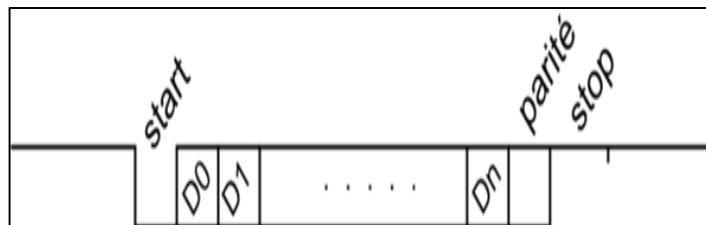


Figure III.14: schéma représente les constituants d'une trame UART.

- un bit de start toujours à 0 : servant à la synchronisation du récepteur
- les données : la taille peut varier (généralement entre 5 et 9 bits)
- éventuellement un bit de parité paire ou impaire
- et un bit de stop toujours à 1 (la durée peut varier entre 1, 1,5 et 2 temps bit)

Le niveau logique de repos est le 1.

III-4-3. Vitesse de transmission [16]:

Afin de faciliter l'interopérabilité entre périphériques (PC, microcontrôleur, modem, ...) des vitesses de transmission sont normalisées par multiples et sous-multiples de 9600 baud, l'unité baud correspondant à un bit par seconde :

- 110 baud
- 300 baud
- 1 200 baud
- 2 400 baud
- 4 800 baud
- 9 600 baud

- 19 200 baud
- 38 400 baud
- 57 600 baud
- 115 200 baud

III-4-4. Principe de transmission par UART:

III-4-4-1. Transmission sérielle asynchrone

La figure III.15 illustre le principe de la transmission sérielle en mode asynchrone. Chaque caractère transmis est précédé par un bit de départ (Start) dont l'état logique est toujours 0 et terminé par un ou deux bits d'arrêt (Stop) dont l'état logique est 1. En l'absence de transmission la ligne se trouve à l'état logique 1. En transmission asynchrone, le temps qui sépare deux caractères est indéfini puisque le récepteur doit attendre l'arrivée du bit de départ pour réagir à l'arrivée d'un nouveau caractère.

Ainsi, une erreur de transmission détruit un seul caractère et non pas un message complet comme c'est le cas en transmission synchrone.

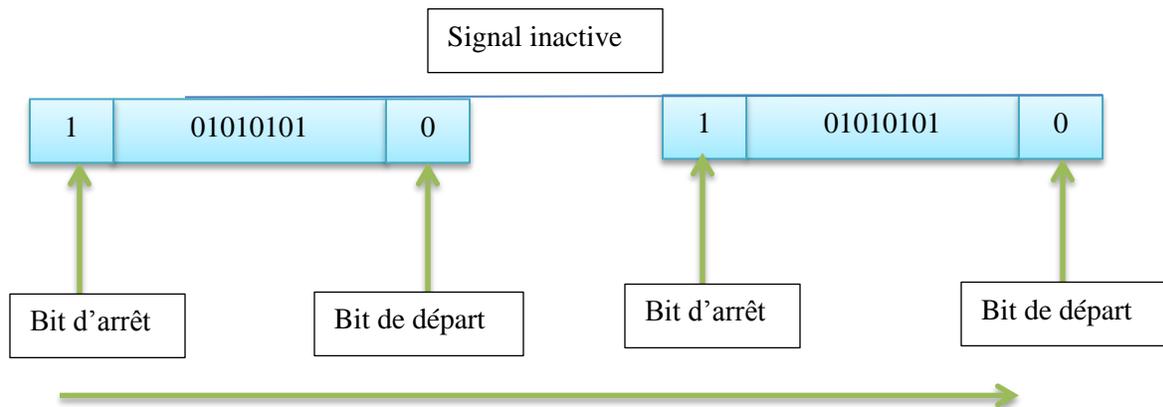


Figure III.15 : Transmission sérielle en mode asynchrone.

III-4-4-2. Fonctionnement d'UART : [16]

Le port série permet de communiquer des données entre 2 microprocesseurs. Le circuit périphérique qui gère cette fonction s'appelle un UART (pour Universal Asynchronous Receiver Transmitter). La transmission est réalisée un octet à la fois, le logiciel copie l'octet dans l'UART et ce dernier se charge de le transférer bit par bit vers le récepteur.

Du côté du récepteur, les bits sont capturés par un autre UART et lorsqu'un octet complet est reçu, il est disponible pour le software du récepteur, chaque UART dispose d'une sortie TX et d'une entrée RX.

La transmission est asynchrone, cela signifie qu'il n'y a pas de signal d'horloge (clock) entre l'émetteur et le récepteur.

Le récepteur, pour identifier les différents bits d'un octet, se base sur la durée de ses bits qui doit être fixe et connue de l'émetteur et du récepteur.

La durée d'un bit est définie par le BAUDRATE, nombre de bits par secondes.

Une liaison série de type UART utilise donc 2 fils sans compter la masse qui doit relier les deux circuits.

III-4-5. Caractéristiques d'une UART : [16]

La plupart des microcontrôleurs modernes disposent de 1 ou plusieurs UART. C'est une technique assez simple à mettre en œuvre.

Sa principale limitation est la vitesse de transmission maximum assez faible due à l'absence de clock entre le récepteur et l'émetteur

Par contre la distance entre les 2 appareils peut être assez importante comparée à des techniques plus rapides tels que USB.

Les niveaux électriques 0V et 5V ou 3.3V fournis par les sorties d'un microcontrôleur ne sont pas ceux utilisés par le port série des (anciens) PCs COM1 et COM2, il faut pour cela utiliser un convertisseur de niveaux de type MAX232 ou équivalent pour passer du RS232 au TTL.

Pour relier 2 microcontrôleurs il n'est pas nécessaire d'utiliser un tel circuit, il suffit de brancher le TX de l'un dans le RX de l'autre et vice versa.

Le circuit USB-UART permet de convertir les signaux RX/TX d'un microcontrôleur vers un signal (et un protocole) USB.

III-5. Présentation de MCU PIC18F4550 avec son architecture interne :

III-5-1. Généralités sur les microcontrôleurs [17] :

Les microcontrôleurs PIC (ou PICMICRO dans la terminologie du fabricant) forment une famille de microcontrôleurs de la société MICROCHIP. Ces microcontrôleurs sont dérivés du PIC1650 développé à l'origine par la division microélectronique de General Instrument.

Le nom PIC n'est pas officiellement un acronyme, bien que la traduction en « *Peripheral Interface Controller* » (« contrôleur d'interface périphérique ») soit généralement admise. Cependant, à l'époque du développement du PIC1650 par General Instrument, PIC était un acronyme de « *Programmable Intelligent Computer* » ou « *Programmable Integrated Circuit* ».

III-5-2. Classification des PIC_s de MICROCHIP :

Actuellement les modèles MICROCHIP, sont classés en 3 grandes familles, comportant chacune plusieurs références. Ces familles sont :

- **Base-line** : les instructions sont codées sur 12 bits.
- **mid-line** : les instructions sont codées sur 14 bits.
- **High-End** : les instructions sont codées sur 16 bits

III-5-3. Choix du pic :

Le choix du PIC est directement lié à l'application envisagée.

- Il faut, dans un premier temps, déterminer le nombre d'entrées/sorties nécessaires pour l'application.
- Il faut ensuite déterminer si l'application nécessite des convertisseurs A/D.
- La rapidité d'exécution est un élément important, il faut consulter les DATA-SHEET pour vérifier la compatibilité entre la vitesse maximale du PIC choisit et la vitesse max nécessaire au montage.
- La taille de la RAM interne et la présence ou non d'une EEPROM pour mémoriser des données.
- La taille du programme de l'application détermine la taille de la mémoire flash Programme du PIC recherché.
- Nombre de cycle «écriture».

Pour notre travail on a opté pour le PIC18F4550 de **Microchip_PIC** (Programable Interface Controller) pour les raisons suivantes :

- La disponibilité.
- Fréquemment utilisé actuellement.
- peu coûteux.
- Performant.

III-5-4. Microcontrôleur PIC18F4550 : [12]

Le **18F4550** est un microcontrôleur du MICROCHIP qui a les caractéristiques principales suivantes:

- **40** pins avec un package **DIP**.
- **5** PORT **I/O** (A, B, C, D, E);
- **13** entrées analogiques de **10** bits pour l'**ADC** ;
- Mémoire programme = 32 KBytes.
- Mémoire programme = 16384 instructions ;
- Mémoire de données = 2304 bytes , sur 15 bank.
 - Static RAM = 2048 bytes;
 - EEPROM = 256 bytes. ;

- Limite maximale du nombre d'écriture ou d'effacement de la mémoire programme est de 100.000 fois et celle de l'EEPROM est de 1.000.000 fois ;
- Existence d'une possibilité à protéger le code source ;
- 4 Timers ;
- **75 (jusqu'à 83)** instructions assembleur différente.
- **20** sources d'interruption;
- 48 MHZ (fréquence maximale).
- UART.
- USB (Universal Serial Bus).

III-5-5. Fonction des différentes broches du PIC18F4550 : [12]

Le diagramme des pins correspond au PIC 18F4550 avec le package DIP est représenté dans la **figure III.16** ainsi que la description des fonctionnalités des pins cités juste au-dessous:

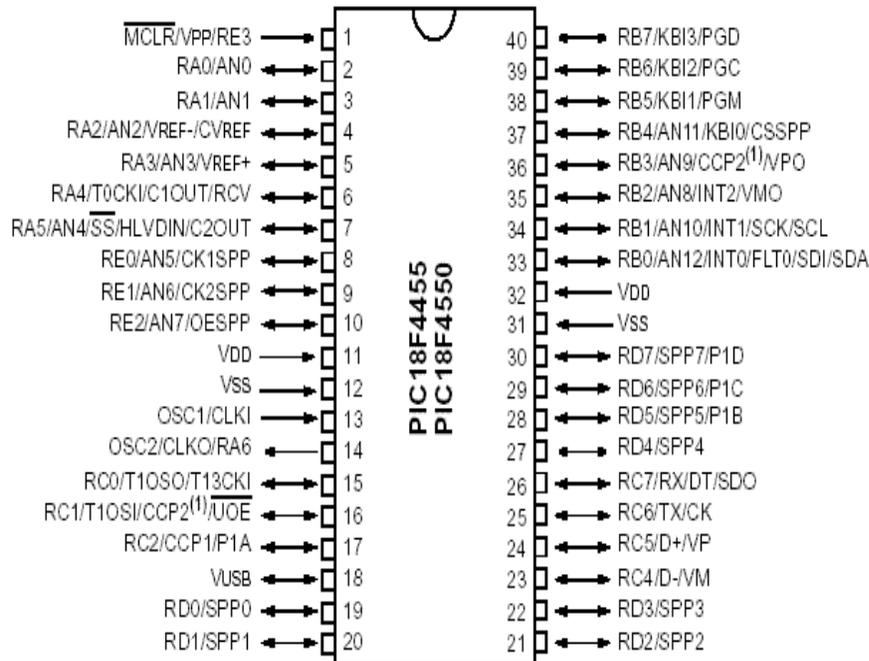


Figure III.16 : Désignation standard des pins du PIC 18F4550.

Les ports d'entrée/sortie :

Le PIC 18F4550 contient les 5 ports suivants :

- Port A : 6 pins I/O numérotées de RA0 à RA5
- Port B : 8 pins I/O numérotées de RB0 à RB7
- Port C : 8 pins I/O numérotées de RC0 à RC7
- Port D : 8 pins I/O numérotées de RD0 à RD7
- Port E : 3 pins I/O numérotées de RE0 à RE2

III-5-6. Architecture interne du PIC 18F4550: [12]

La figure III.17 présente l'architecture du PIC 18F4550, commune à la majorité des microcontrôleurs PIC. Le PIC 18F4550 est un microcontrôleur 8 bits d'architecture de type RISC (*Reduced Instructions Set Computer*), signifie « calculateur à jeu réduit d'instructions ». Les instructions sont en effet au nombre de 75 ou 83 avec l'activation du mode d'instruction élargie.

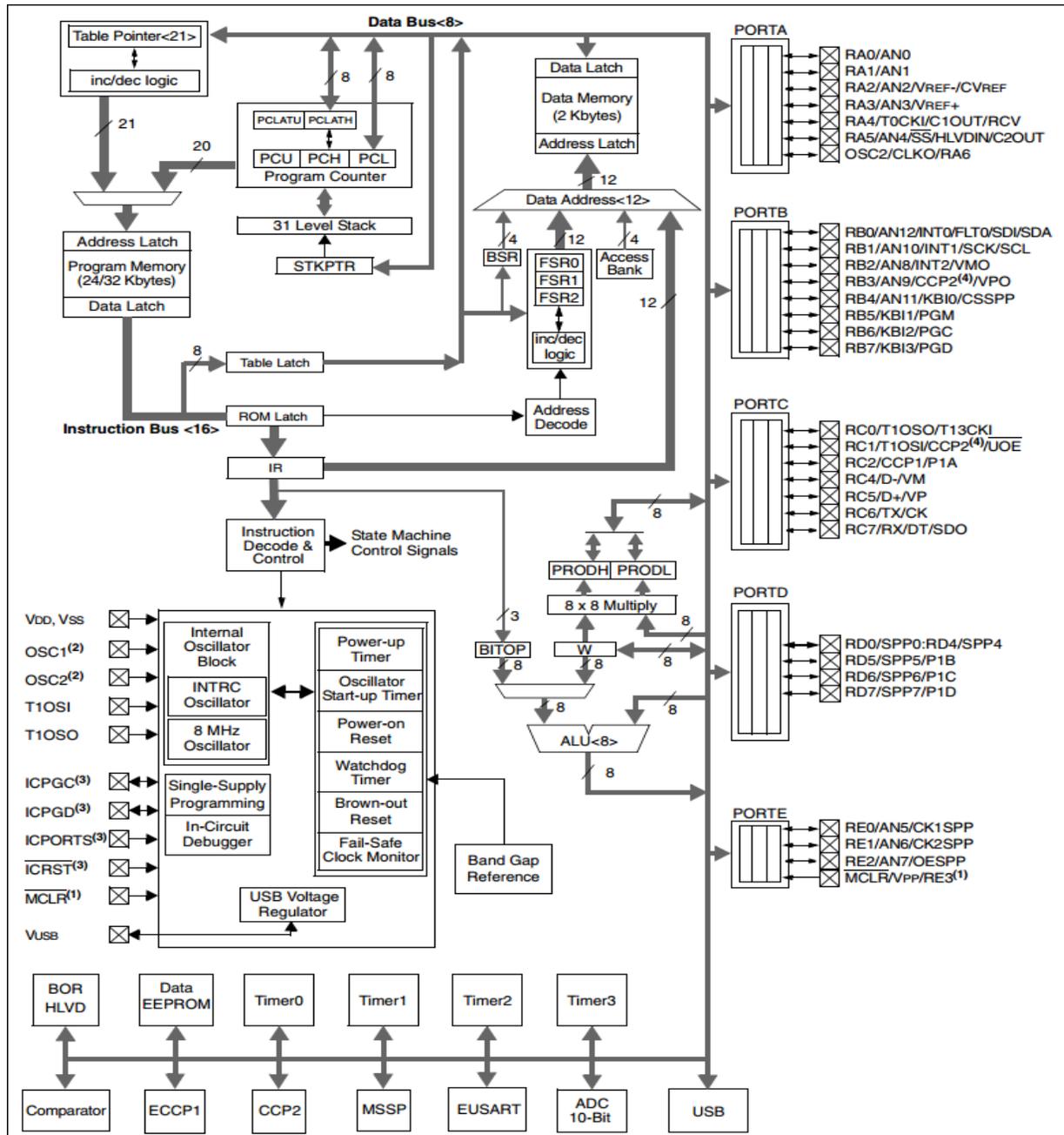


Figure III.17 : Architecture interne du PIC18F4550.

Les PIC se conforment à l'architecture Harvard : ils possèdent une mémoire de programme et une mémoire de données séparées. La plupart des instructions occupent un mot de la mémoire de programme. La taille de ces mots dépend du modèle de PIC, tandis que la mémoire de données est organisée en octets.

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

Comme la plupart des instructions sont exécutées en un seul cycle, hormis les sauts, cela donne une puissance de l'ordre de 1 MIPS par MHz (1 million d'instructions par seconde).

III-6. Conclusion :

Dans ce chapitre, nous avons exploré les solutions matérielles possibles pour réaliser une carte électronique adoptée pour le Contrôle et le pilotage du four électrique disponible au niveau du laboratoire.

Cependant, au préalable nous avons testé et vérifié individuellement l'ensemble des 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 four où le logiciel **PROTEUS** a été utilisé lors de la simulation des composants. La description de ce dernier ainsi que les autres logiciels utilisés seront présentés dans le prochain chapitre.

Chapitre IV

Supervision et acquisition des données

Chapitre VI: Supervision et acquisition des données.

IV-1. Introduction :

Après avoir étudié et réalisé la carte d'interface électronique, assurant la jonction entre le processus à commander et l'ordinateur, nous abordons dans cette section la partie software permettant la supervision de notre procédé et l'acquisition des données entre la carte électronique et notre software.

Notre software est basé principalement sur la plateforme .NET sous Windows, utilisant la programmation orientée objet avec le langage c #.

Notre objectif dans ce chapitre est de réaliser un système de supervision pour le four électrique du laboratoire afin de surveiller, contrôler et de détecter en temps réel des problèmes qui peuvent survenir en cours du fonctionnement de l'installation.

IV-2. Présentation des softwares utilisés :

IV-2-1. Visual studio C#: [1]

Microsoft Visual Studio est une suite de logiciels de développement conçue par Microsoft pour le système d'exploitation Windows et dernièrement pour linux et mac. La dernière version s'appelle Visual Studio 2015.

La **figure IV-1** présente le logo du Visual Studio.



Figure IV-1 : logo de VISUAL STUDIO

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles. Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du plateforme .NET, qui fournit un accès à des technologies clés simplifiant le développement d'applications web ASP et de services web XML grâce à Visual Web Developer.

La figure IV-2 présente la fenêtre principale de Visual studio :

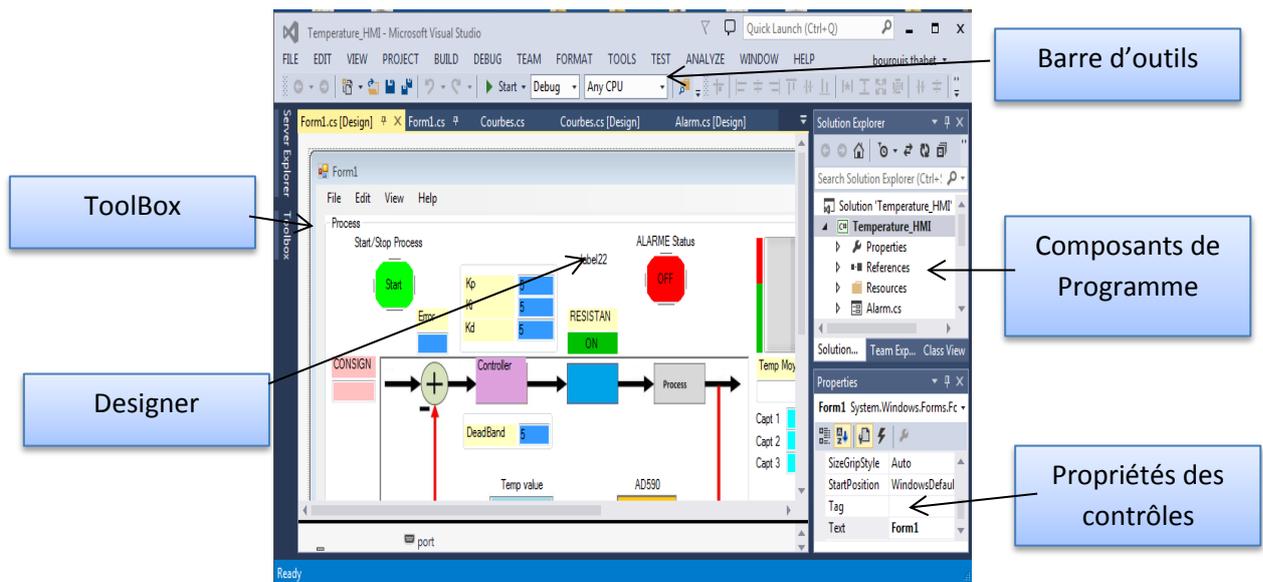


Figure IV-2 : la fenêtre de l'environnement du développement VISUAL STUDIO.

On peut créer un nouveau projet utilisant le dialogue « New Project » comme la figure IV-3 :

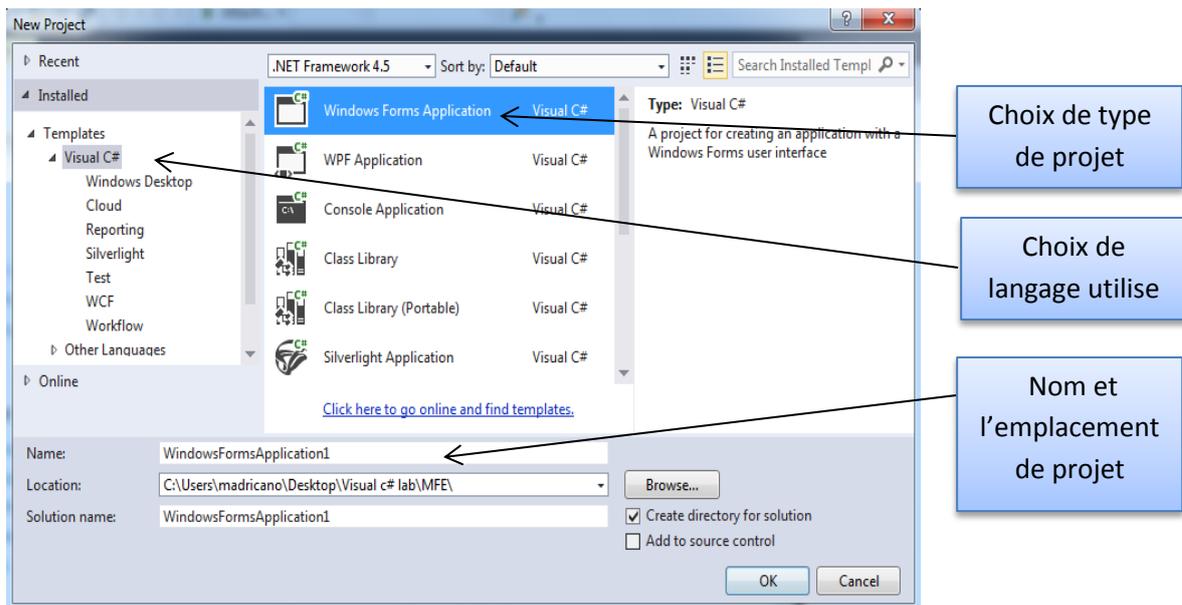


Figure IV-3 : le dialogue qui permet de créer un nouveau projet.

IV-2-2. Introduction sur les langages orientées objet : [2]

La programmation orientée objet (POO) est un type de programmation qui a pour avantage de posséder une meilleure organisation, surtout dans le gros programme.

Ces derniers seront agencés de façon plus logique et seront donc plus facilement modifiables. La POO est dirigée par trois principes fondamentaux qu'il convient de toujours garder à l'esprit : encapsulation, héritage, polymorphisme.

IV-2-3. Langage de programmation C# : [3]

Le C# (C Sharp) est un langage de programmation orientée objet, fortement organisé, dérivé du C et C++ et qui ressemble au langage JAVA. Il est utilisé pour développer des applications web, ainsi que des applications de bureau, de services web, de commandes ou de bibliothèques de classes.

En C# une application est un lot de classes où l'une des classes comporte une méthode 'Main', comme cela se fait en java.

C# est destinée à développer sur la plateforme .NET, une pile technologique créée par Microsoft. Cette plateforme .net est gratuite au niveau du site de web du Microsoft.

La **figure IV-4** représente le logo du langage C#.



Figure IV-4 : le logo de langage C#.

IV-2-4. La plateforme .NET : [4]

La plate-forme est un environnement de développement et d'exécution, qui reprend les conceptions de la machine virtuelle de JAVA, et qui utilise le CLR (Common Language Runtime).

La compilation du code source génère un objet intermédiaire dans le langage MSIL (Microsoft Intermediate Language).

La plateforme est indépendante de toute architecture de processeur et de tout système d'exploitation.

La **figure IV-5** présente le logo de plateforme dot Net.



Figure IV-5 : le logo de la plateforme .NET.

IV-2-5. MIKROC : [5]

MIKROC PRO est un compilateur intégré qui utilise le langage C standard, il est dédié pour la conception des systèmes embarqués, il dispose d'un environnement de développement très riche, et puissant avec des optimisations avancées, beaucoup de bibliothèques matérielles et logicielles et d'autres outils (UART Terminal, USB HID terminal ...etc).

La **figure IV-6** présente le logo de MikroC pour PICs.



Figure IV-6 : le logo de MIKROC

IV-3. Présentation générale sur La supervision industrielle:**IV-3-1. Introduction : [8]**

La supervision est une technique industrielle de suivi et de pilotage informatique de procédés de fabrication automatisés.

La supervision concerne l'acquisition de données (mesures, alarmes, l'état du fonctionnement) et des paramètres de commande des processus généralement confiés à des automates programmables.

IV-3-2. Avantage de la supervision :

Les logiciels de supervision sont une classe de programmes dédiés à la production, dont les buts sont les suivants :

- ✓ Assistance de l'opérateur dans ses actions de commande du processus de production (interface IHM dynamique...).
- ✓ la visualisation de l'état et de l'évolution d'une installation automatisée de contrôle de processus, avec une mise en évidence des anomalies (alarmes).
- ✓ la collecte d'informations en temps réel sur des processus depuis des sites distants (machines, ateliers, usines...) et leur archivage.
- ✓ Apporter de l'aide à l'opérateur dans son travail (séquence d'actions, recette) et dans ses décisions (propositions des paramètres, signalisation des valeurs en défaut, aide à la résolution d'un problème ...).
- ✓ Fournir des données pour atteindre les objectifs de production (quantité, qualité, traçabilité, sécurité...).

IV-3-3. Fonctionnalités d'un système de Supervision:[9]

Un système de supervision comprend deux sous-ensembles fonctionnels:

- ❖ la commande.
- ❖ la surveillance.

IV-3-3-a. La Commande :

Le rôle de la commande est de faire exécuter un ensemble d'opérations au procédé en fixant des consignes de fonctionnement en réponse à des ordres d'exécution. Il s'agit de réaliser généralement une séquence d'opérations constituant une gamme de fabrication dans le but de fabriquer un produit en réponse à une demande d'un client.

La commande regroupe toutes les fonctions qui agissent directement sur les actionneurs du procédé qui permettent d'assurer :

- ✓ le fonctionnement en l'absence de défaillance.
- ✓ la reprise ou gestion des modes.
- ✓ les traitements d'urgence.
- ✓ une partie de la maintenance corrective.

Les fonctions de commande en marche normale sont:

- L'envoi de consignes vers le procédé dans le but de provoquer son évolution.
- L'acquisition de mesures ou d'informations permettant de reconstituer l'état réel du procédé et/ou du produit.
- L'envoi vers le procédé d'ordres prioritaires permettant de déclencher des procédures de sécurité (arrêts d'urgence par exemple).

IV-3-3-b. La Surveillance :

La partie surveillance d'un superviseur a pour objectifs :

- recueille en permanence tous les signaux en provenance du procédé et de la commande.
- reconstitue l'état réel du système commandé.
- fait toutes les inférences nécessaires pour produire les données utilisées pour dresser des historiques de fonctionnement.
- met en œuvre un processus de traitement de défaillance dans le cas échéant.

Dans cette définition, la surveillance est limitée aux fonctions qui collectent des informations, les archivent, font des inférences, etc. sans agir réellement ni sur le procédé ni sur la commande. La surveillance a donc un rôle passif vis-à-vis du système de commande et du procédé.

IV-3-4. Architecture d'un réseau de supervision : [10]

En vue de la réalisation d'une communication entre un système embarqué et un PC, des mécanismes d'échange ont été développés dans ce sens pour assurer l'échange de données entre le PC de supervision et un microcontrôleur.

Le choix d'un réseau de communication dépend principalement des besoins en termes de couverture géographique, de qualité de données et de nombre d'abonnés.

Le PC de supervision n'échange pas directement les données avec les capteurs ou les actionneurs du procédé à superviser, mais à travers le microcontrôleur qui gère l'ensemble du processus.

Un réseau de supervision est souvent constitué de :

- ❖ Un PC utilisé comme poste opérateur, permet l'acquisition des données, l'affichage des synoptiques et la conduite de l'unité.
- ❖ Un réseau d'acquisition, reliant le poste opérateur avec les organes de commande.

La **figure IV-7** est présenté l'architecture d'un système de la supervision :

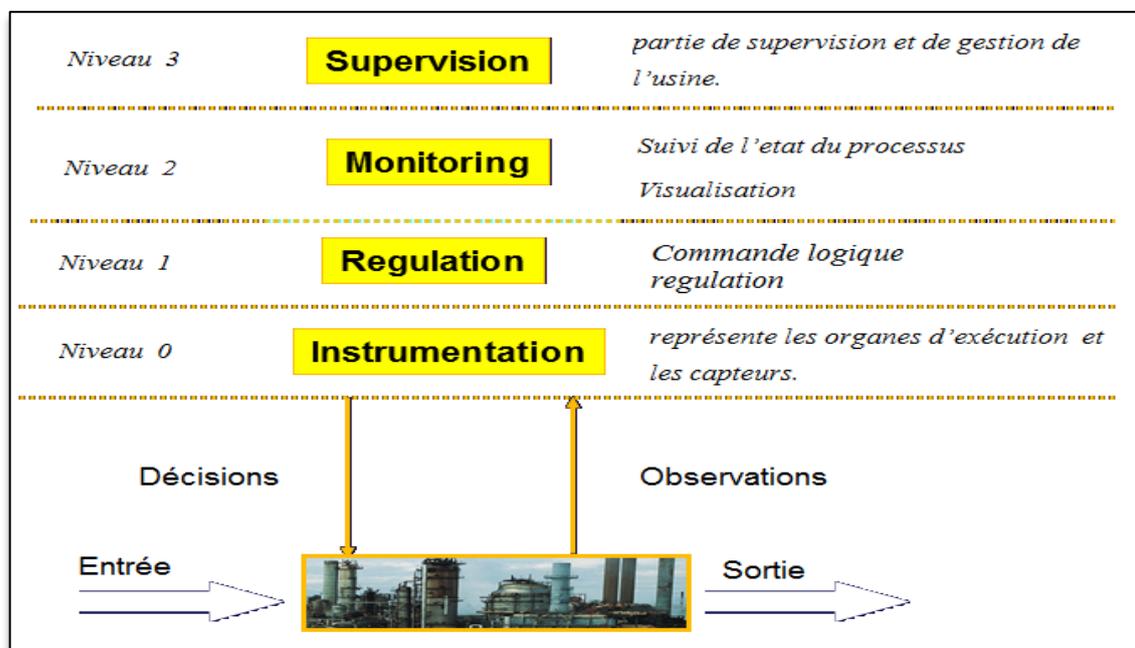


Figure IV-7 : l'architecture d'un système de supervision.

IV-4. Présentation de l'application réalisée :

Le logiciel de supervision et contrôle d'un four électrique est basé sur la plateforme .NET sous le système d'exploitation Windows, il est développé avec le langage orienté objet C#(C Sharp) qui permet la flexibilité et la structuration de code source, ainsi que la rapidité d'exécution.

Ce logiciel a pour but de contrôler et superviser le four électrique du laboratoire.

Pour la supervision, le software contient plusieurs fenêtres (interface graphique) chacune pour assurer une tâche spécifique tel que : la gestion des alarmes, archivage, synoptique ...etc.

Enfin, l'acquisition des données est une tâche principale dans notre software, il permet d'acquérir les valeurs de la température du four et assurer l'envoi de signal de commande.

La **figure IV-8** représente la structure générale du software de commande et supervision d'un four électrique :

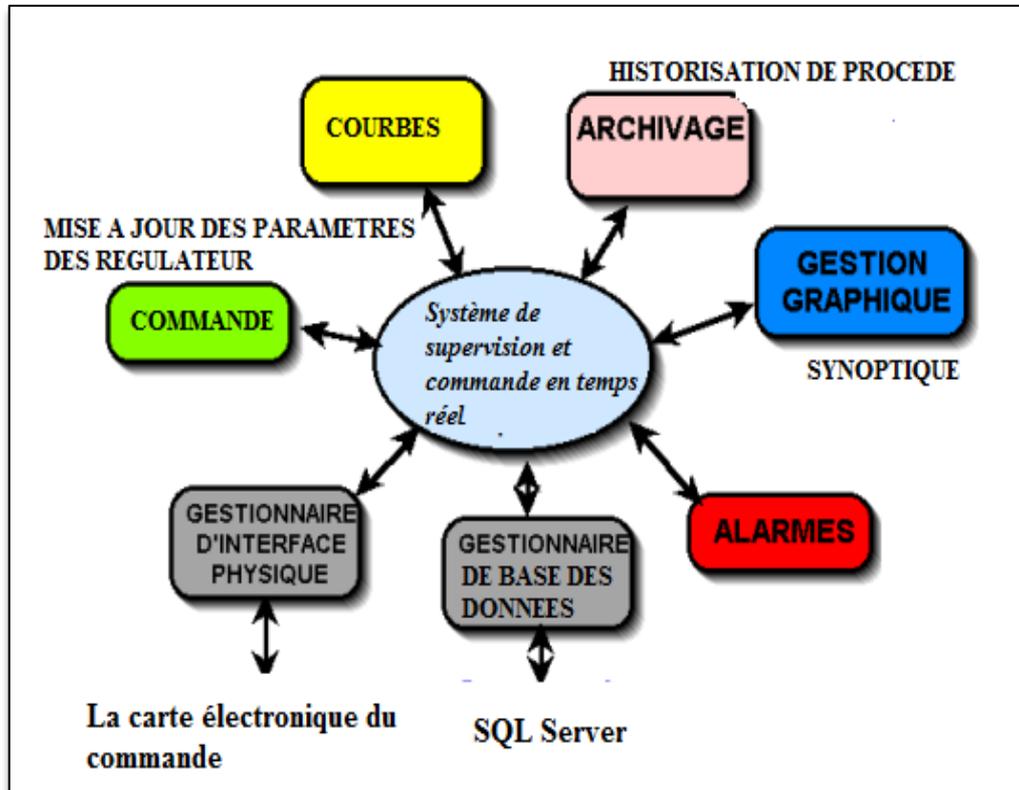


Figure IV-8 : la structure générale du software de commande et supervision d'un four électrique

IV-4-1. Acquisition des données :

IV-4-1-a. Introduction:

Le principe d'un système d'acquisition des données est la mesure d'une grandeur physique ou un signal électrique.

L'acquisition des données sur PC est présentée par l'interface d'une carte électronique et un logiciel sur l'ordinateur, cette relation est assurée par un protocole de communication qui permet l'envoi des données réel entre les deux.

En résumé, un système d'acquisition des données assurée les tâches suivants :

- Recueillir des informations et les mesures de capteur.
- Enregistrer la température du four.
- Créer un lien entre le four électrique et l'interface graphique de l'ordinateur.
- Définir un canal de sortie analogique ou numérique en temps réel

Les constituants d'un système d'acquisition des données sont :

- Acquisition des données (Capteurs, conditionneurs)
- Transformations des données (CAN/CNA)
- Traitement des données (calculateur numérique)

Elles sont illustrées par la figure IV-9 :

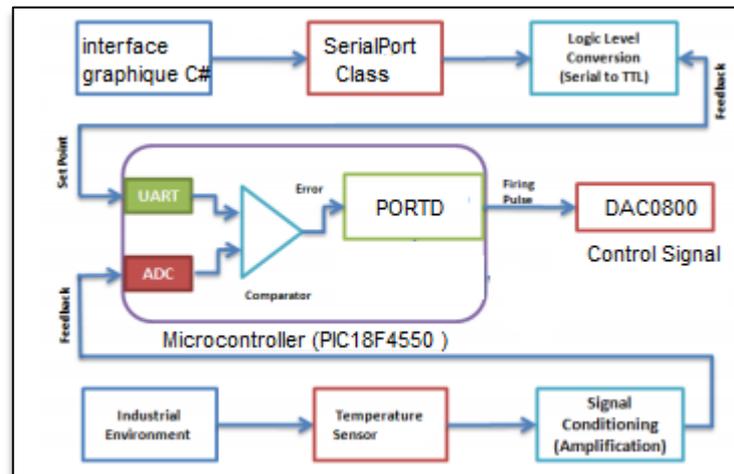


Figure IV-9: Constituants d'un système d'acquisition des données

IV-4-1-b. Description d'étapes d'acquisition des données:

Cette section définit la procédure permettant l'acquisition des données, cette procédure est répartie en 4 étapes, Ses étapes sont illustrées par la figure IV-10:

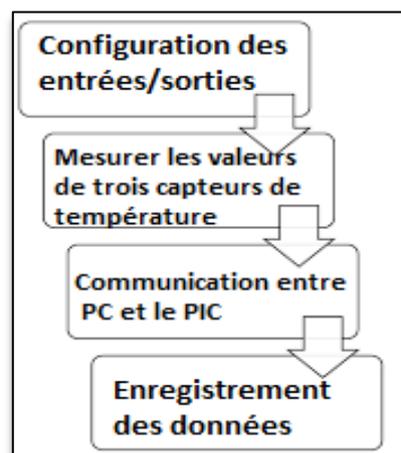


Figure IV-10 : représente les étapes d'acquisition des données.

1. Configuration des entrées/sorties :

Toutes les pins de microcontrôleur pic18f4550 sont disponible à la fois en lecture et en écriture, si ils sont utilisés dans le mode logique par défaut entrées/sorties (I/O), les pins de PIC sont reconnus sous formes de lettres PORTA, PORTB, PORTC, PORTD, et PORTE.

Chaque port est un mot de 8bits et chaque bit a généralement sa propre broche du microcontrôleur correspondante .Pour toutes les directions du bus, il y a deux registres séparés, en plus de ca il y a un registre qui décide de la direction réelle du bus, ou ma valeur du bit a 0 marque le bus comme une sortie puis en entrée lorsque la valeur est à 1.

Dans notre cas, le PORTA est configuré comme une entrée analogique pour mesurer les valeurs de capteur, et le PORTD comme sortie digitale pour l'affectation de signal de commande vers le procédé.

La figure IV-11 représentée la configuration logicielle de registres associes aux PORTA et PORTD :

```
//Configure ADC
ADC_Init();
ADCON1=0b00001011; // AN0--AN3 ==analog AN4--AN12== digital
ADCON2=0b00101101; // 12 Tad / Fosc/16
//PORTA 0--3 analog input
TRISA=0b000000111;
CMCON=0b000001111; //disable comparators
//PORTD as digital output
TRISD=0x00;
PORTD=0x00;
```

Figure IV-11: la configuration logicielle des registres associe aux ports A et D.

2. Mesurer les valeurs de trois capteurs de température

La chaine d'acquisition a pour fonction de recueillir et transformer la grandeur à mesurer sous une forme adaptée à son exploitation.

La conversion analogique numérique consiste à transformer la tension analogique (délivre à la sortie de circuit de conditionnement du capteur) en un code binaire adapte à son exploitation dans un processus de régulation.

Dans notre cas, l'ADC intégré dans le microcontrôleur pic18F4550 transformé le signal de capteur de la température AD590 en une valeur binaire stockée dans un variable entier de 0 à 1023, soit sur 10 bits.

3. Communication entre l'ordinateur et le microcontrôleur.

3.1 Transférer les donnes vers l'ordinateur :

La transmission d'une série d'octets par le port série est facilement réalisable entre le microcontrôleur et le PC. Cependant, la réception d'une série d'octets ne permet pas de suivre l'évolution de différents variables du programme du microcontrôleur. Une couche de protocole supplémentaire est nécessaire .celle-ci va définir pour chaque donnée envoyée un canal et le type de donnée transmise (octet ou mot) .il est alors possible de

visualiser les différents variables du microcontrôleur en tracent leurs évolutions respectives au cours de temps.

Le microcontrôleur transmet les mesures des capteurs avec le protocole suivant :

- Byte1 : un caractère de début d'envoi ':'(Header).
- Byte2-4 : la première mesure variée entre 0-255.
- Byte5 et 9 : un caractère de séparation '*'.
- Byte6-8 : la deuxième mesure.
- Byte10-12: la troisième mesure.
- Byte13 : un caractère de fin ';'.

La trame des données est présentée sur la figure IV-12.

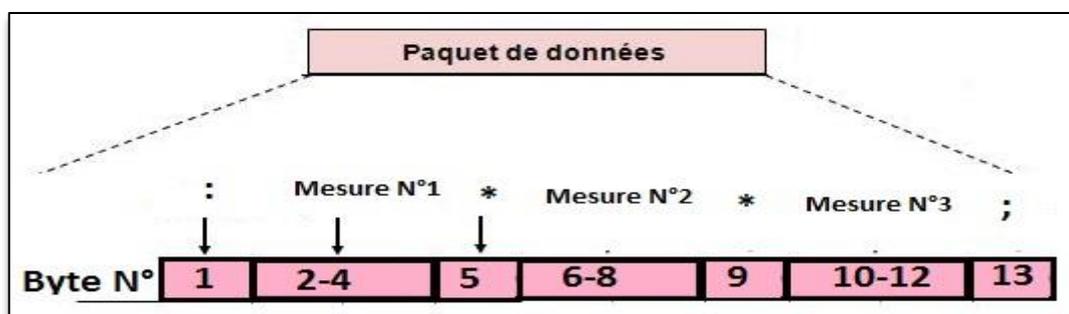


Figure IV-12: représente la trame des données transmise à l'ordinateur.

3.2 Emission de données du l'ordinateur vers le microcontrôleur :

Le microcontrôleur reçoit la commande délivrée par l'ordinateur sous la structure suivante :

- Byte1 : un caractère de début '*'.
- Byte2-4 : le signal de commande.
- Byte5 : un caractère de fin ';'.

Cette structure est illustrée dans la figure IV-13:

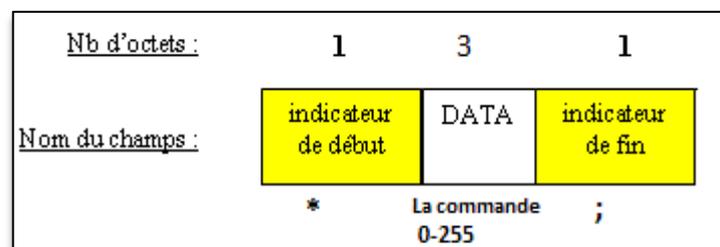


Figure IV-13: représente la trame de signal de commande.

Après le décodage, le microcontrôleur sauvegarde la valeur obtenue dans un variable, ensuite cette valeur doit être affectée sur le port D pour obtenir un signal binaire pour attaquer l'entrée de DAC.

3.3 Réception des données sous Visual C# :

Le programme sous Visual C# acquiert pendant un laps de temps de une seconde la trame de données en provenance du microcontrôleur. Le décodage optimal de la trame est ensuite séparé les trois mesures. Chaque variable est affichée et sauvegardée dans une base de données, puis la moyenne de ces trois mesures est utilisée pour les différents algorithmes de commande. La figure IV-14 illustre la méthode de décodage de la trame.

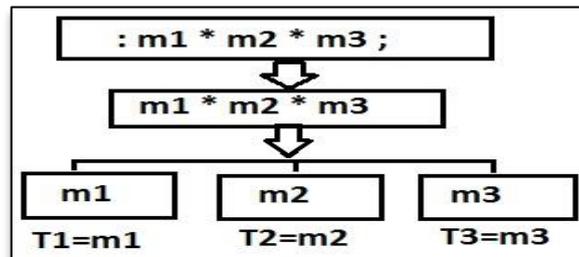


Figure IV-14: représente le décodage de la trame.

4. Enregistrement des données :

Après la réception des mesures, le programme que nous proposons est assuré de la sauvegarde de ces mesures à une base de données de type SQL Server.

L'enregistrement de la valeur moyenne de température du four se fait sous la forme d'un tableau. Le tableau est illustré dans la figure IV-15:

	id	Temp	Time	Date	RegType
▶	1	23	12:00:00	2015-05-17	PI

Figure IV-15: représente la structure d'un tableau TEMP dans la base des données SQL SERVER

IV-4-2. Supervision :

La partie de logiciel dans notre système de supervision est assurée par une interface graphique développée à l'aide du langage C#. Cette interface permettant d'exécuter des algorithmes de contrôle et aussi la visualisation de l'évolution des paramètres du processus.

Notre application est composée de parties suivantes :

- Synoptique.
- Courbes.
- Alarmes.
- Historisation de procédé.
- Gestion des paramètres de contrôle.
- Gestion de base de données.

La figure IV-16 montre la structure logicielle de notre système de supervision ainsi que ces différentes parties :

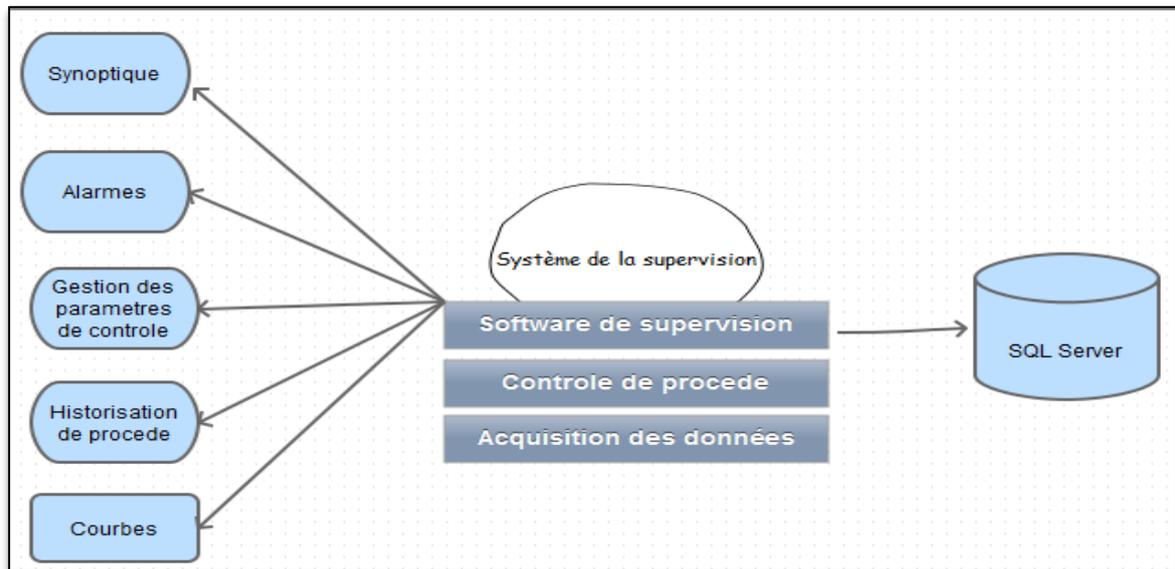


Figure IV-16 : la structure logicielle de notre système de supervision.

IV-4-2-1. Synoptique :

C'est la fonction essentielle de notre software, elle fournit une représentation synthétique, dynamique et instantanée de l'ensemble des paramètres de procédé.

Permet à l'opérateur d'interagir avec le processus et de visualiser la température de four en temps réel.

La figure IV-17 présente la synoptique qui est l'interface graphique globale de notre software :

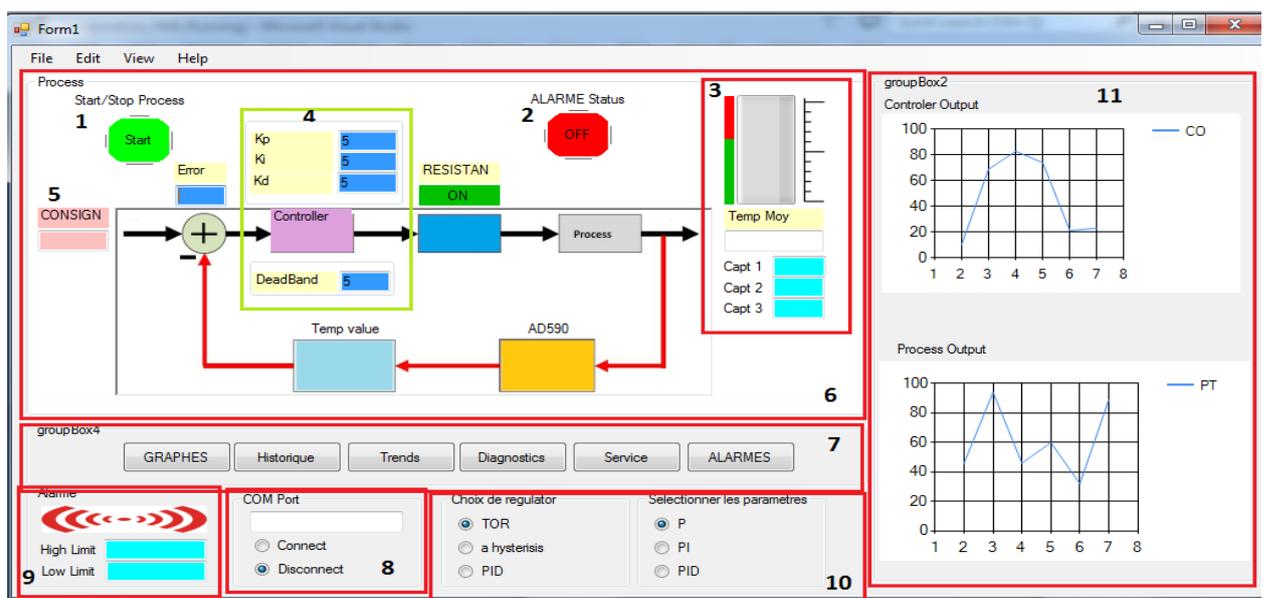


Figure IV-17: la synoptique de logiciel de supervision avec ces parties.

La synoptique comporte les parties suivants :

- **Etat de procédé (1):** Démarrer ou arrêté le procédé.
- **Etat d'alarme(2) :** affichage de statuts d'alarme.
- **La température désirée (5) :** Edition et l'affichage de la consigne.
- **Courbes (11):** visualisation en temps réel des variables entrées/sorties (commande/réponse du four).
- **Communication port série (8):** configuration de port série, ouvrir et ferme une connexion.
- **Choix de régulateur (10):** choix de régulateur utilise.
- **Configuration du seuil d'alarme (9):** préciser le seuil minimal et maximal d'alarme.
- **Boucle de régulation (6):** représente le schéma fonctionnel de procédé.
- **Affichage des paramètres de procédé (3) (4):** affichage de la valeur moyen de température du four, ainsi que les paramètres des régulateurs et la valeur de l'écart entre la consigne et valeur réelle.
- **Barre d'outils (7):** permet l'accès à différentes fenêtres du logiciel.

IV-4-2-2. Courbes :

C'est la partie responsable de la présentation graphique en temps réel de signal de commande et de la valeur de température réelle du four. Elle Donne des outils utilisés pour la modification sur les propriétés des courbes.

La figure IV-18 présente l'interface graphique « courbes » :

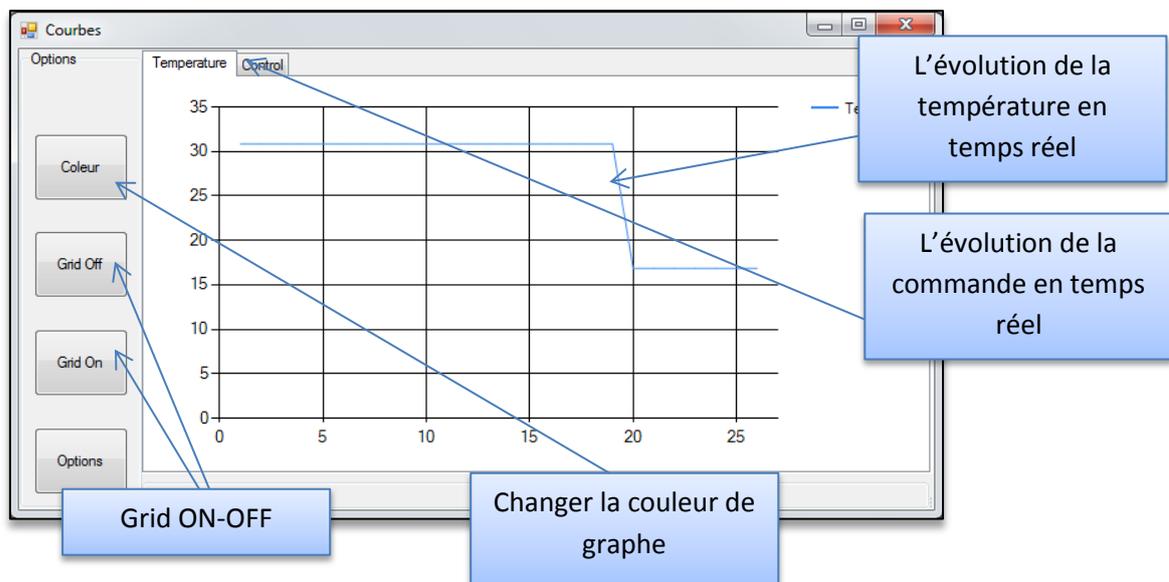


Figure IV-18 : représente la fenêtre « courbes ».

IV-4-2-3. Alarme :

Cette partie est responsable de fonctions suivantes :

- surveillance en temps réel sur le fonctionnement de processus.
- Configuration des seuils de déclenchement d'alarme.
- Affiche l'ensemble des alarmes selon des règles de priorité.
- Filtrer l'ensemble des alarmes selon des conditions sélectionné.
- Assure l'enregistrement de toutes les étapes de traitement de l'alarme

La figure IV-19 présente la fenêtre « alarme » et ses parties :

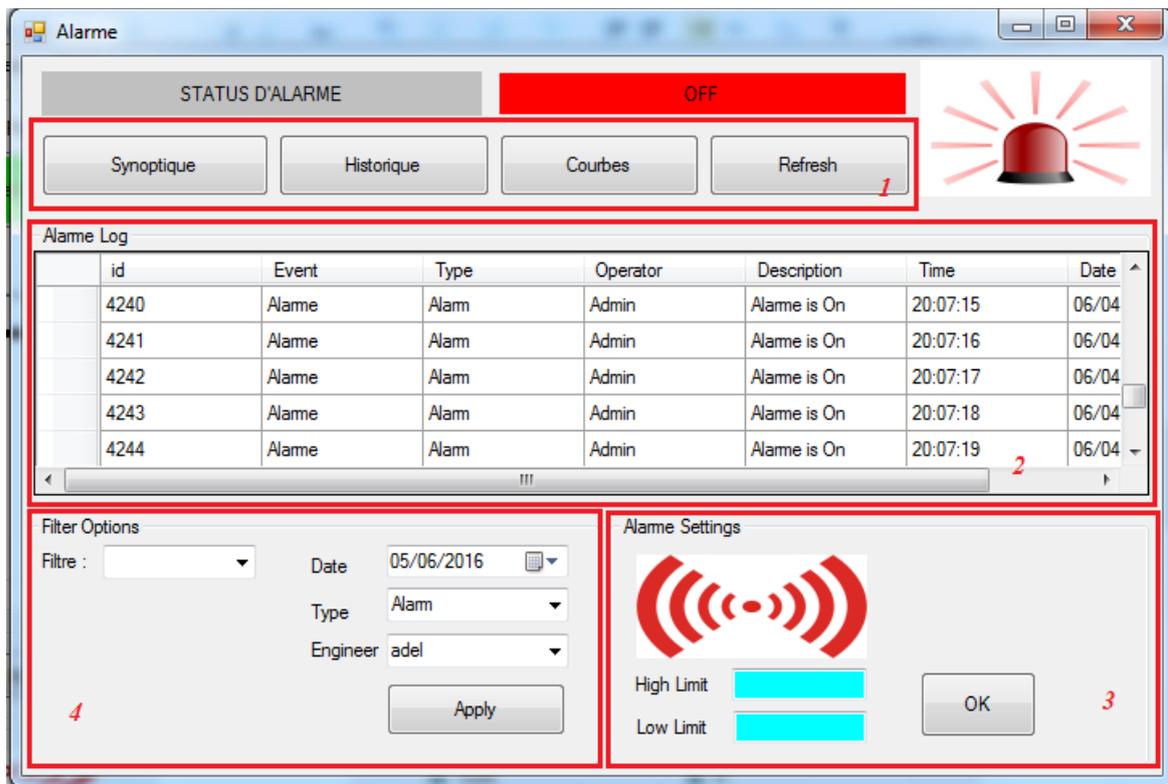


Figure IV-19: représente la fenêtre « alarme » et ses parties

La fenêtre « Alarme » comporte les parties suivantes :

- **Barre d'outils(1).**
- **Options de filtrage des résultats(4) :** spécifié les règles de filtrage pour un affichage sélectif des alarmes enregistré dans la base de données.
- **Affichage(2) :** Afficher tous les événements d'alarme selon les règles de filtrage sélectionné.
- **Configuration des alarmes(3) :** modification des seuils maximal et minimal d'alarme.

IV-4-2-4. Historisation du procédé :

La fenêtre « historique » est permet de :

- la sauvegarde périodique de la valeur de température.
- la sauvegarde d'événements horodatés (archivage sélectif).
- Fournit les outils de recherche dans les données archivées.
- sauvegarder une série de donnes(les valeurs de température ou la commande) dans un fichier sélectionné.
- tracer une courbe pour une série de donnes archivées.

La **figure IV-20** présente la deuxième partie de la fenêtre « historique » qui est permet de sauvegarder les donnes archives de procédé dans un fichier ou bien de visualiser dans un graphe.

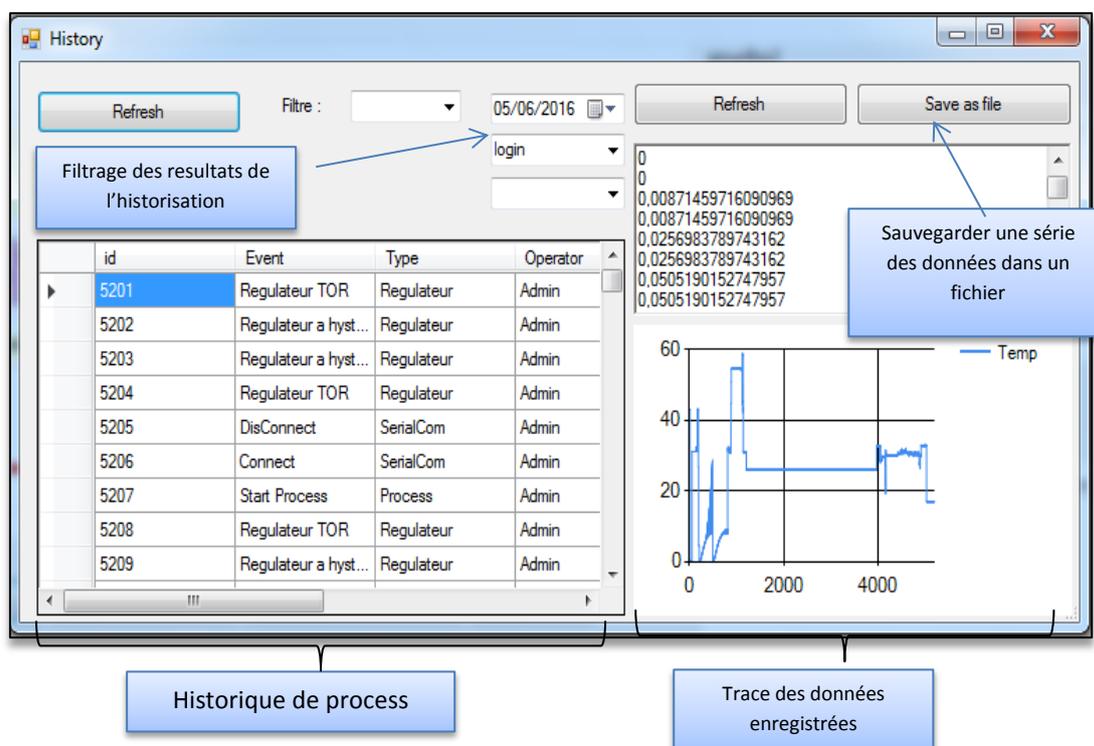


Figure IV-20 : La fenêtre « historique ».

IV-4-2-5. Gestion des paramètres de Contrôle :

C'est la partie responsable de fonctions suivantes :

- permet de configurer les paramètres de régulateur.
- modification de la consigne.
- sélectionner le mode de fonctionnement Auto/Manuel.
- configuration des paramètres de connexion de la base de donnes SQL SERVER.
- permet de sauvegarder un backup de la base de donnes.

La **figure IV-21** présente l'interface graphique « Gestion des paramètres» et ses parties :

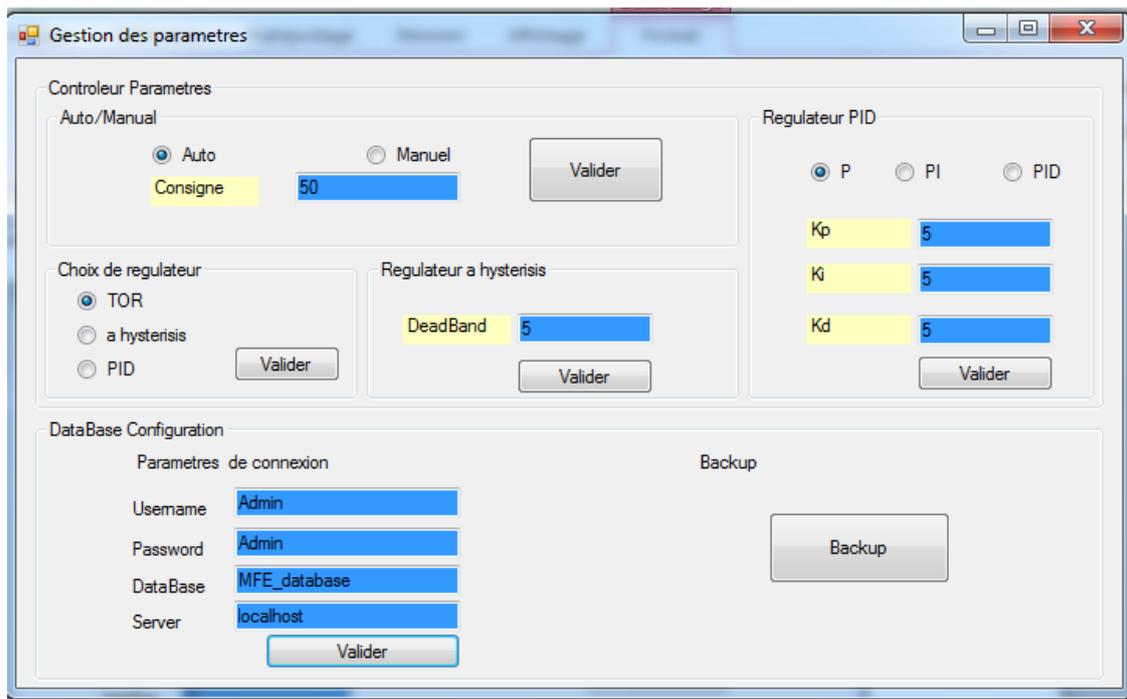


Figure IV-21: l'interface graphique « gestion des paramètres »

IV-4-2-6. Gestion de base des données :

Dans les systèmes de supervision industrielle, la base de données est la partie indispensable qui relie toutes ces éléments, elle permet de :

- sauvegarder la valeur moyenne de température dans un tableau à la base de données SQL server.
- l'enregistrement de toutes les tâches et configurations manipulées durant le fonctionnement du système.

La figure IV-22 présente la structure de base de données utilisée dans notre système de supervision :

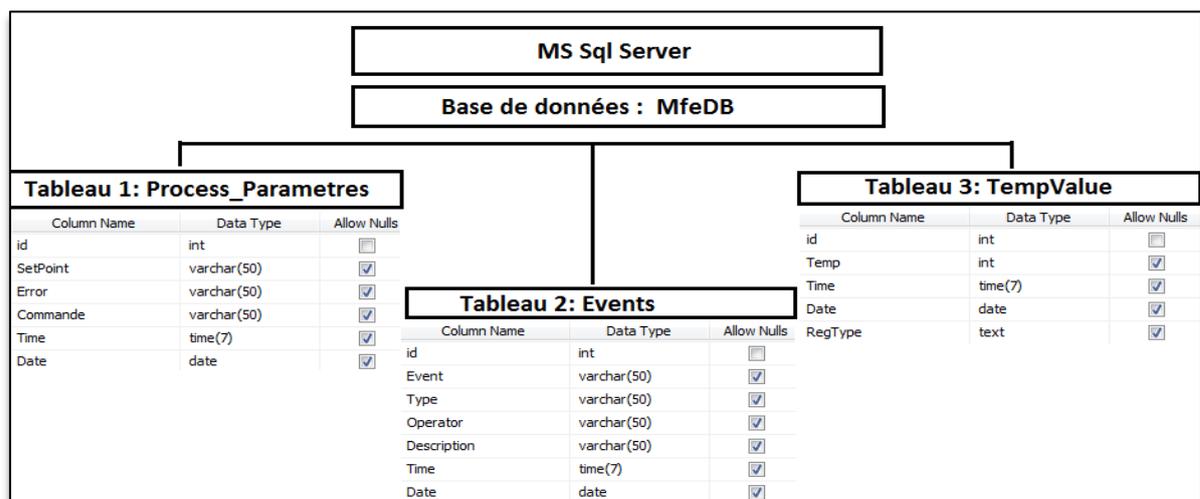


Figure IV-22: représente la structure de base de données utilisée dans ce projet.

IV-5. Résultats de simulation du projet :

IV-5-1. Simulation du la synoptique du software :

La commande et la supervision du four électrique au sein de laboratoire est représenté par le cycle constitué des étapes suivantes :

Etape initial :

Au chaque démarrage de notre software, on doit configurer les paramètres de port série (voir la figure IV-23).

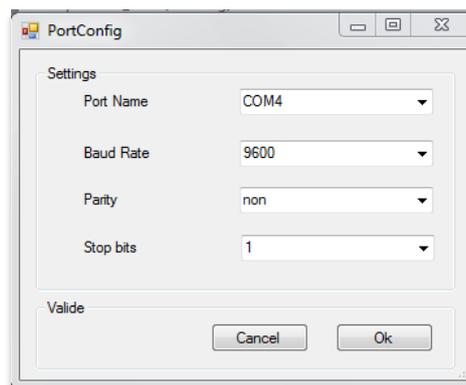


Figure IV-23 : la fenêtre « portConf ».

A l'état initial les composants de système sont comme suit : (voir figure IV-24)

- Régulateur en mode “TOR”.
- Procédé en d'état Arrêt.
- La communication série en état “déconnecté”.
- Alarme en état « OFF »
- Les courbes en état initiale.

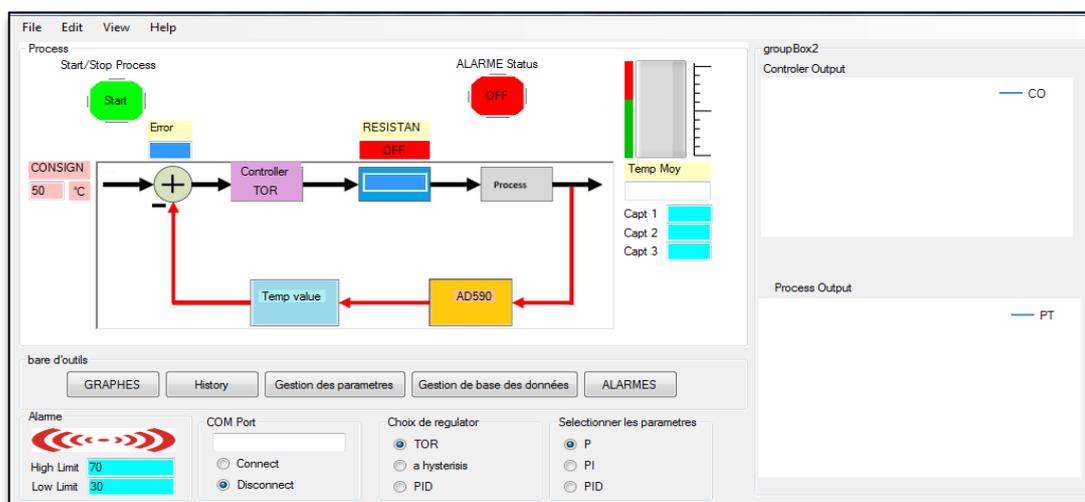


Figure IV-24 : le synoptique à l'état initial.

Etape 2 : Après la sélection de régulateur désiré et initialisation de la valeur de consigne, on doit ouvrir le port série et démarrer le procédé (voir figure IV-25).

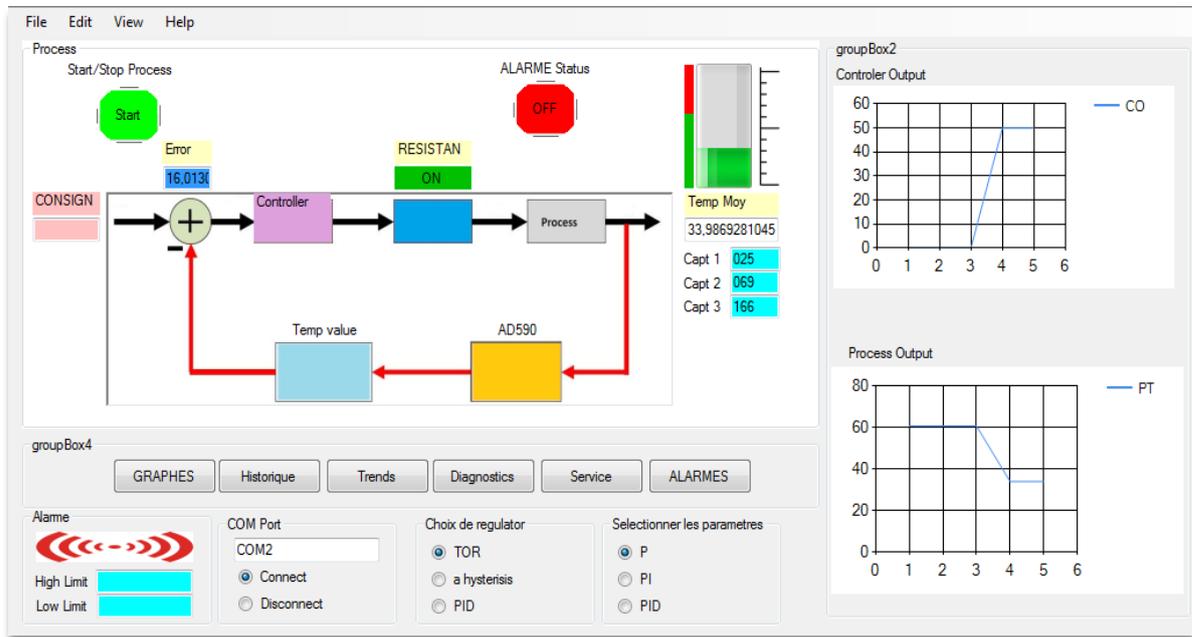


Figure IV-25 : Le système de supervision en état marche.

Etape 3 : Après avoir terminé l’initialisation de système de la supervision, le software va réagir avec le procédé selon le régulateur choisi.

Les figures (IV-26-a), (IV-26-b) et (IV-26-c) présente le fonctionnement en temps réel de notre software avec le procédé selon les trois modes de régulation.

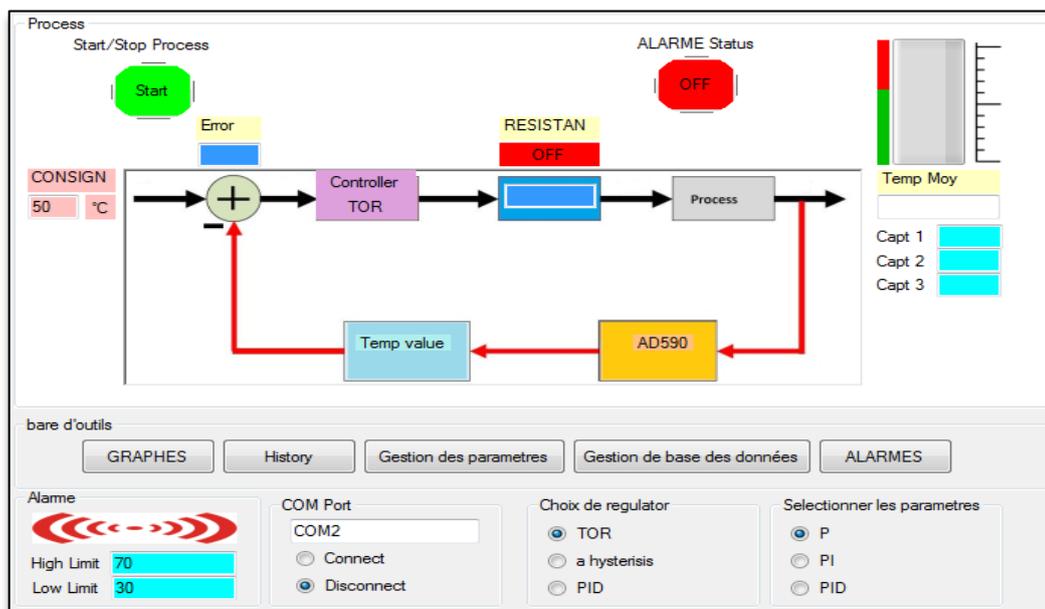


Figure IV-26-a : le système de supervision avec le régulateur TOR.

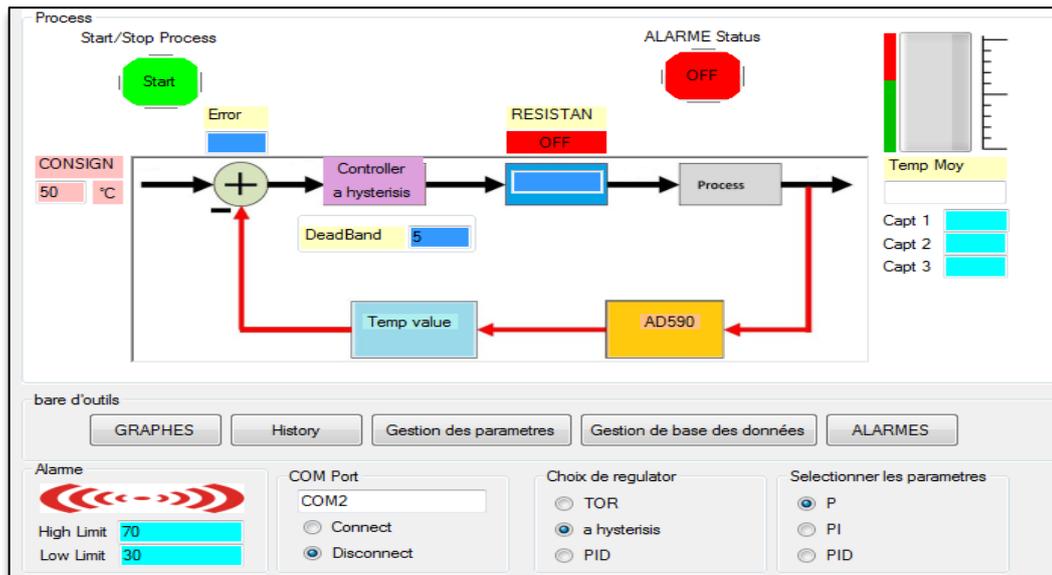


Figure IV-26-b : le système de supervision avec le régulateur a hystérésis.

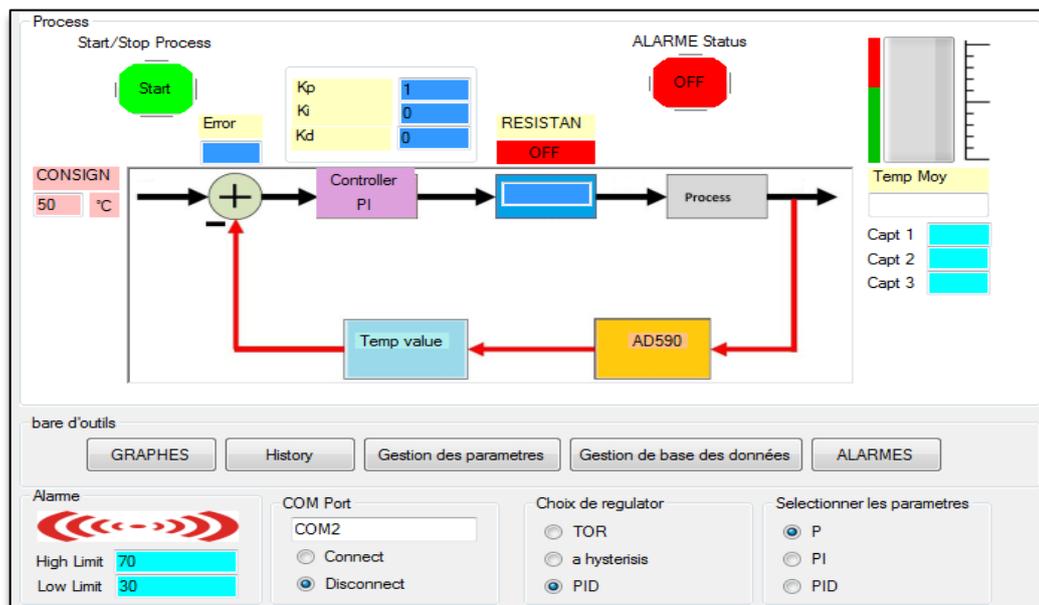


Figure IV-26-c : le système de supervision avec le régulateur PID.

Etape 4 : Dans le fonctionnement du système, toutes les donnes sont enregistrées dans la base des données.

IV-5-2. Simulation d'un déclenchement de système d'alarme :

Suite à une très basse ou très haute valeur de température, une alarme se déclenche qu'on la visualise dans la liste des alarmes comme suit (voir figure IV-27).

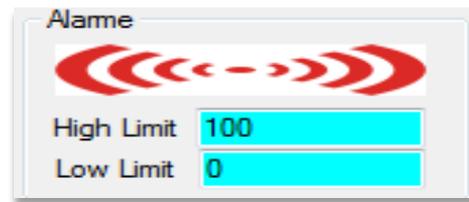


Figure IV-27 : la section de configuration des seuils d'alarme.

Le déclenchement de l'alarme provoque (voir la figure IV-28) :

- La fermeture de port série.
- L'arrêt du système.



Figure IV-28 : l'état de synoptique en cas le déclenchement d'alarme.

IV-6. Conclusion :

Le software que nous avons implémenté à l'aide d'un langage orienté objet sous Windows ont été appliques à la supervision, et le contrôle de la température d'un four électrique de laboratoire.

Dans ce chapitre, nous avons introduit une description détaillée de chaque partie de notre software.

Chapitre V

Implémentation du programme de commande

Chapitre V: Implémentation du programme de commande.

V-1. INTRODUCTION :

Dans cette partie, nous allons examiner une commande par PC, ou l'utilisateur introduit la consigne de température au niveau du software. La variation de la température en temps réel est relevée et affichée sur l'interface graphique que nous proposons. La température de four du laboratoire est contrôlée grâce à l'utilisation des plusieurs types de régulateur en boucle fermée, on doit présenter le contrôleur ON-OFF, ON-OFF à hystérésis et un contrôleur PI.

V-2. Schéma bloc de la boucle de régulation du four électrique :

Une chaîne de régulation est formée généralement d'une cascade d'appareils qui doivent être compatibles entre eux, Dans ce but, notre chaîne de régulation est constituée principalement par un software GUI (Graphical User Interface) qui génère les algorithmes de différents contrôleurs, une carte de communication entre le four et l'ordinateur, et le four électrique du laboratoire.

Le schéma bloc simplifié de la boucle de régulation est présenté dans la **figure V-1**:

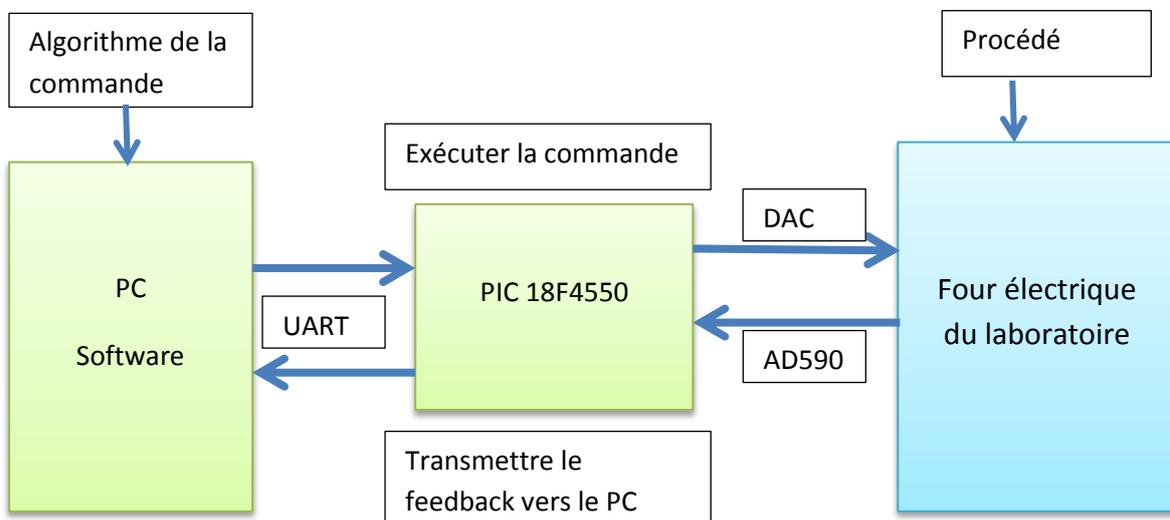


Figure V-1 : Schéma simplifié de la boucle de régulation

V-3. Commande en boucle ouvert avec feedback de mesure (Mode Manuel):

V-3-1. La structure de la commande :

Dans cette première phase, nous allons examiner une commande par PC, ou l'utilisateur introduit la consigne de température au niveau du software. La variation de la température en temps réel est relevée et affichée sur l'interface graphique que nous proposons.

Le principe de la commande avec boucle de mesure est présenté en **figure V-2**. En effet nous retrouvons le software sur l'ordinateur fournit la valeur de la consigne température T_{ref} et affiche la température réelle du four.

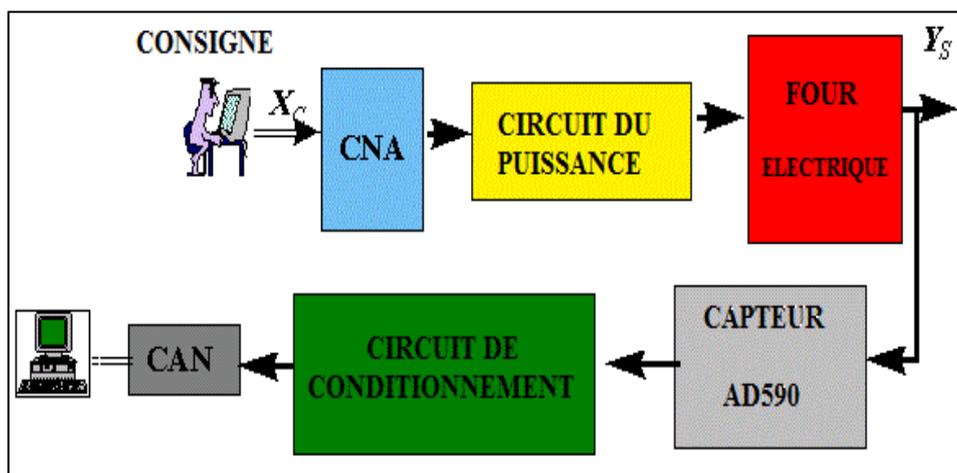


Figure V-2: présente la structure de la commande en boucle ouvert.

V-3-2. Implémentation de l'algorithme de contrôle :

L'algorithme que nous proposons permettra un contrôle de four en mode Manuel (commande en boucle ouvert). Ce programme informatique a pour fonction de générer le signal de référence de température T_{ref} . 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 chaque une période d'échantillonnage ($t=1s$). Pour illustrer cette partie software, la **figure V-3** présente l'organigramme simplifié décrivant la logique de cette commande.

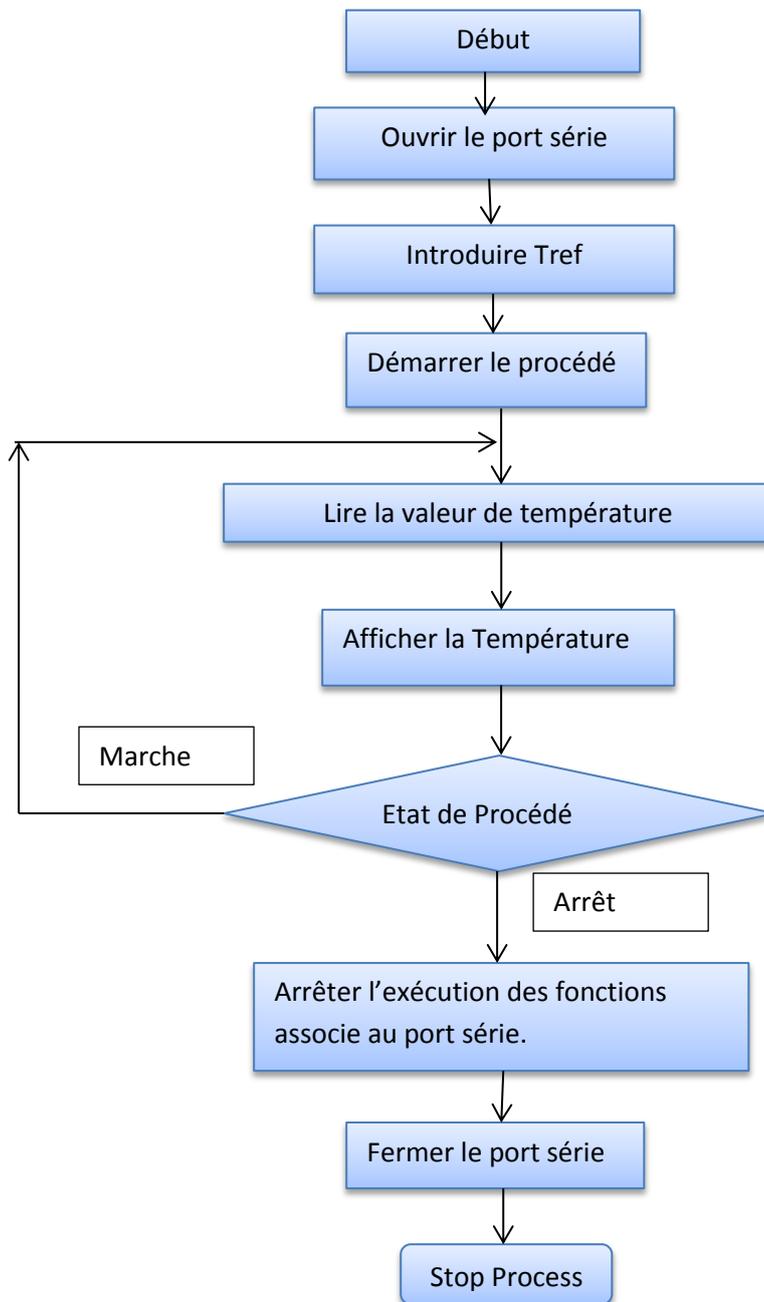


Figure V-3 : Organigramme simplifié du programme de commande en boucle ouverte.

V-3-3. Résultats et interprétations:

L'exécution du programme, nous a permis de relever la courbe de réponse T_r pour une trajectoire T_{ref} . Nous constatons que le four électrique possède une dynamique relativement lente soit environ de 100 secondes par échelon comme l'indiquée la figure V-4 ci-dessous.

Pendant cet essai nous avons positionné le capteur de température à environ par rapport à la résistance chauffante du four.

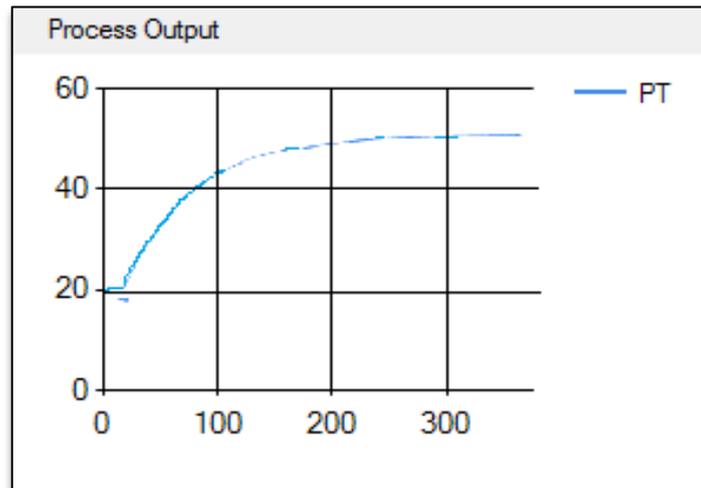


Figure.V-4 : L'évolution de la température du four en fonction du temps.

V-4. Commande en boucle fermée (Mode Automatique) :

Après l'analyse des performances de la structure de contrôle en boucle ouverte, nous constatons qu'il est nécessaire de concevoir une commande avec boucle de réglage afin d'améliorer la dynamique du four.

Dans cette partie nous étudions les stratégies de commande suivantes :

- Une commande avec contrôleur ON-OFF.
- Une commande avec contrôleur à hystérésis ON-OFF avec zone morte.
- Une commande avec contrôleur proportionnel intégrale (PI) non linéaire.

Le schéma synoptique simplifié de cette structure est présenté en **figure V-5**.

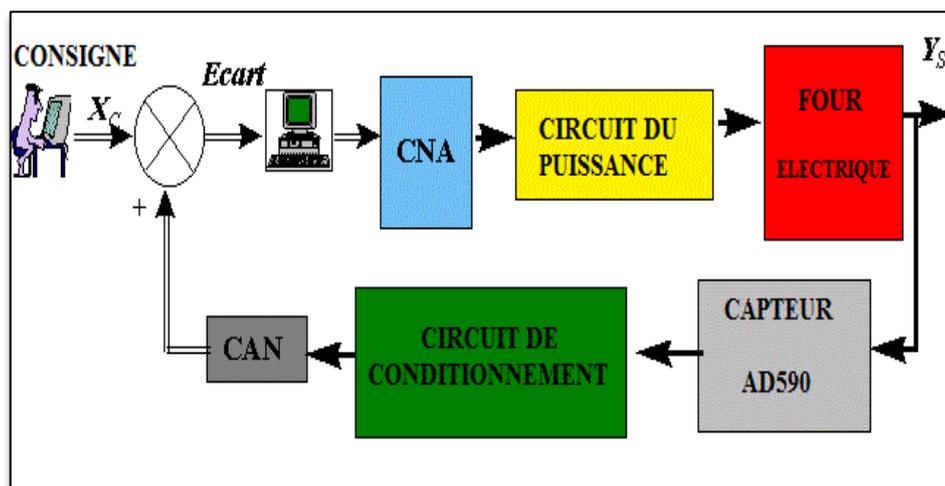


Figure.V-5 : Schéma synoptique simplifié du command en boucle fermée.

V-4-1. Contrôleur ON-OFF :**V-4-1-1. La structure de commande :**

L'algorithme de réglage ON-OFF, consiste à délivrer une commande max ou min pour minimiser l'erreur entre la consigne et la mesure. La commande U est donnée par :

$$U = \begin{cases} 0 & \text{Si } e \geq 0 \\ 10 \text{ V} & \text{Si } e < 0 \end{cases}$$

La **figure V-6** présente le schéma de régulation tout ou rien :

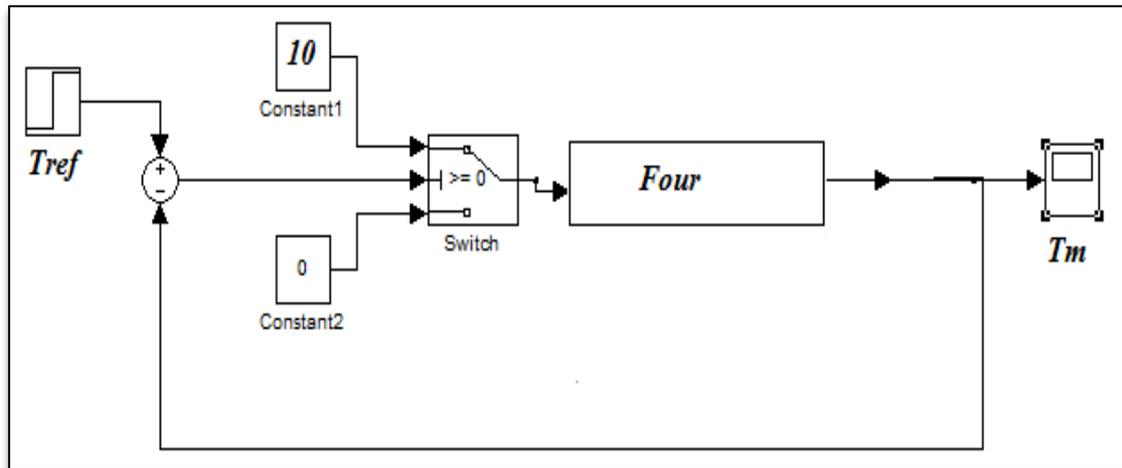


Figure V-6: présente le schéma de régulation ON-OFF.

V-4-1-2. Implémentation de l'algorithme de contrôle :

L'algorithme de calcul de la commande ON-OFF est donné dans la **figure V-7** ci-dessous :

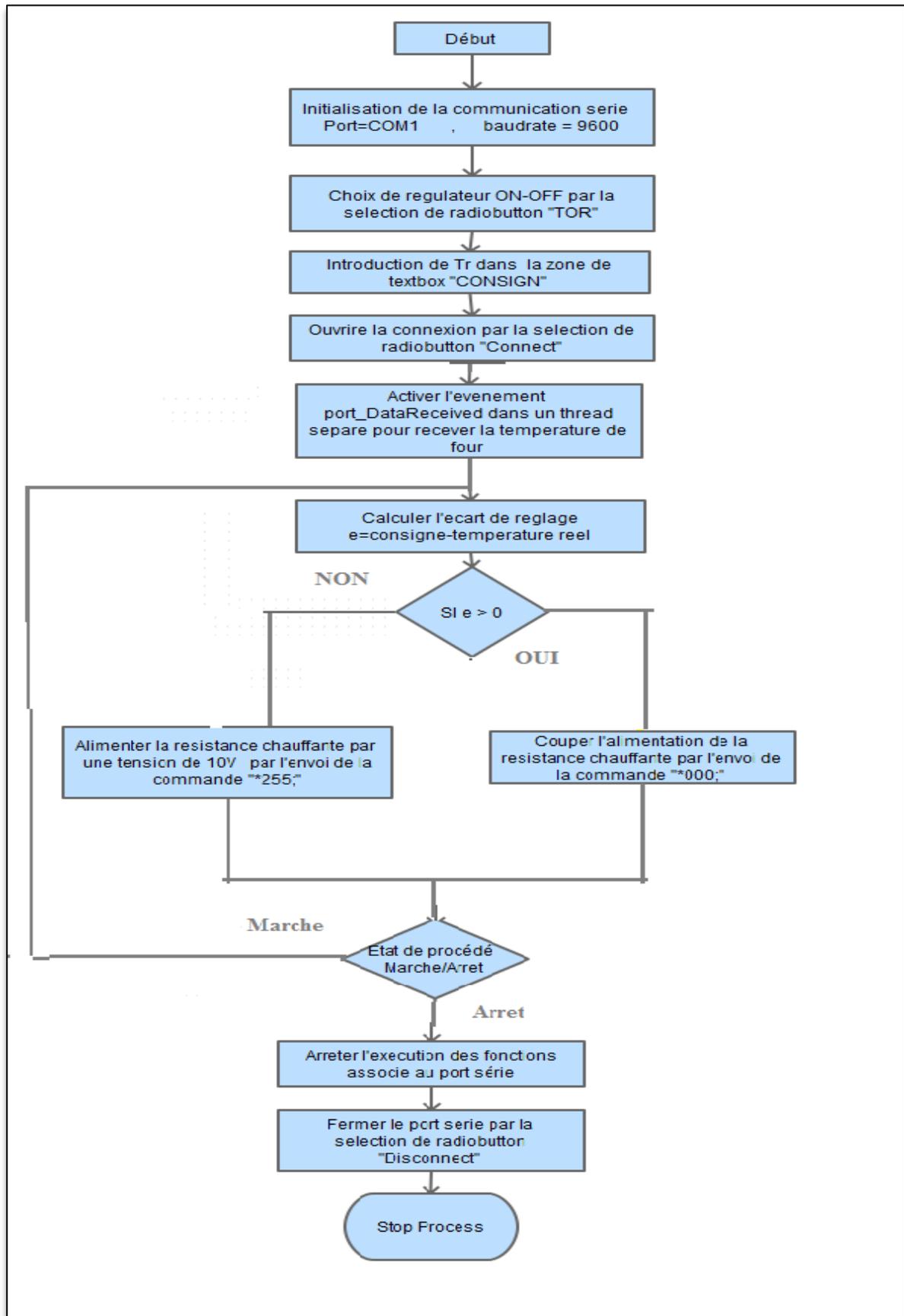


Figure V-7:Algorithme de réglage ON-OFF.

V-4-1-3. Résultats et interprétation :

Après implémentation du programme informatique et exécution de ce dernier, nous avons enregistré les courbes de réponse pour un échelon de $T_{ref}(k) = 50$. Les courbes relatives à la four et la commande sont portées en **figure V-8**.

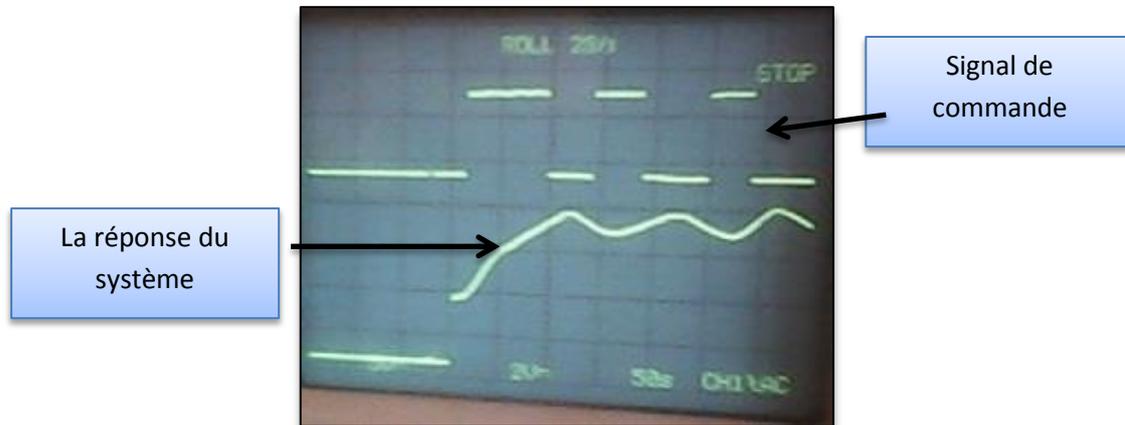


Figure.V-8 : Courbe de signal de la commande et de la réponse du four avec contrôleur ON-OFF.

V-4-2. Contrôleur ON-OFF à hystérésis :

V-4-2-1. La structure de commande :

Cette configuration utilise un régulateur non linéaire de type ON/OFF avec zone morte (Dead band). Ce type de réglage est utilisé dans le cas du four électrique afin de maintenir la température entre deux valeurs $\pm \Delta t$, qui sera fixée par l'utilisateur.

Le schéma bloc de la commande est présenté en **figure V-9**.

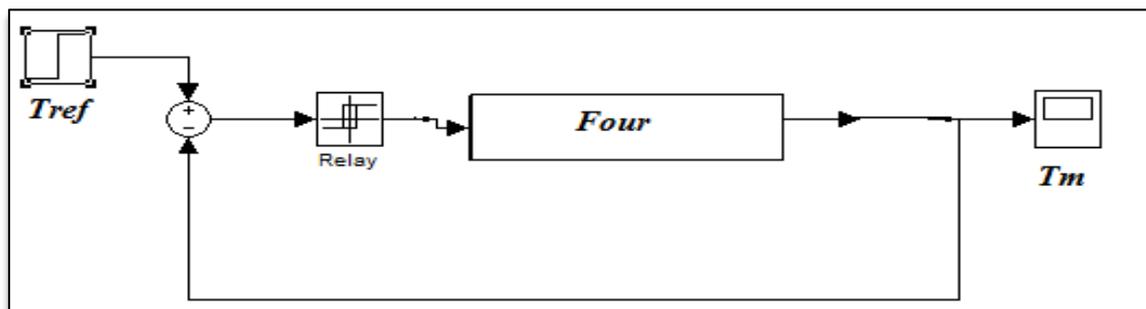


Figure.V.9 : Schéma fonctionnel de la commande de la plaque chauffante avec contrôleur ON/OFF a hystérésis.

V-4-2-2. Implémentation de l'algorithme de contrôle :

L'algorithme du contrôleur a hystérésis travaillant en mode poursuite s'effectue selon les étapes suivantes :

1. Calcul de l'erreur de réglage e .
2. Comparaison de $\varepsilon(k)$ par rapport à la zone morte $\pm \Delta t$.

3. Si $e < 0$ et $e > \Delta t$ alors la sortie du contrôleur est on mode OFF ($U=0$).
4. Si $e < 0$ et $e \leq \Delta t$ alors la sortie du contrôleur garde l'état précédent.
5. Si $e > 0$ et $e \leq +\Delta t$ alors la sortie du contrôleur est on mode ON ($U=1$).
6. Si $e > 0$ et $e > +\Delta t$ alors la sortie du contrôleur garde on l'état précédent.

La **figure V-10** illustre l'organigramme décrivant la logique de contrôle avec ce type de contrôleur non linéaire.

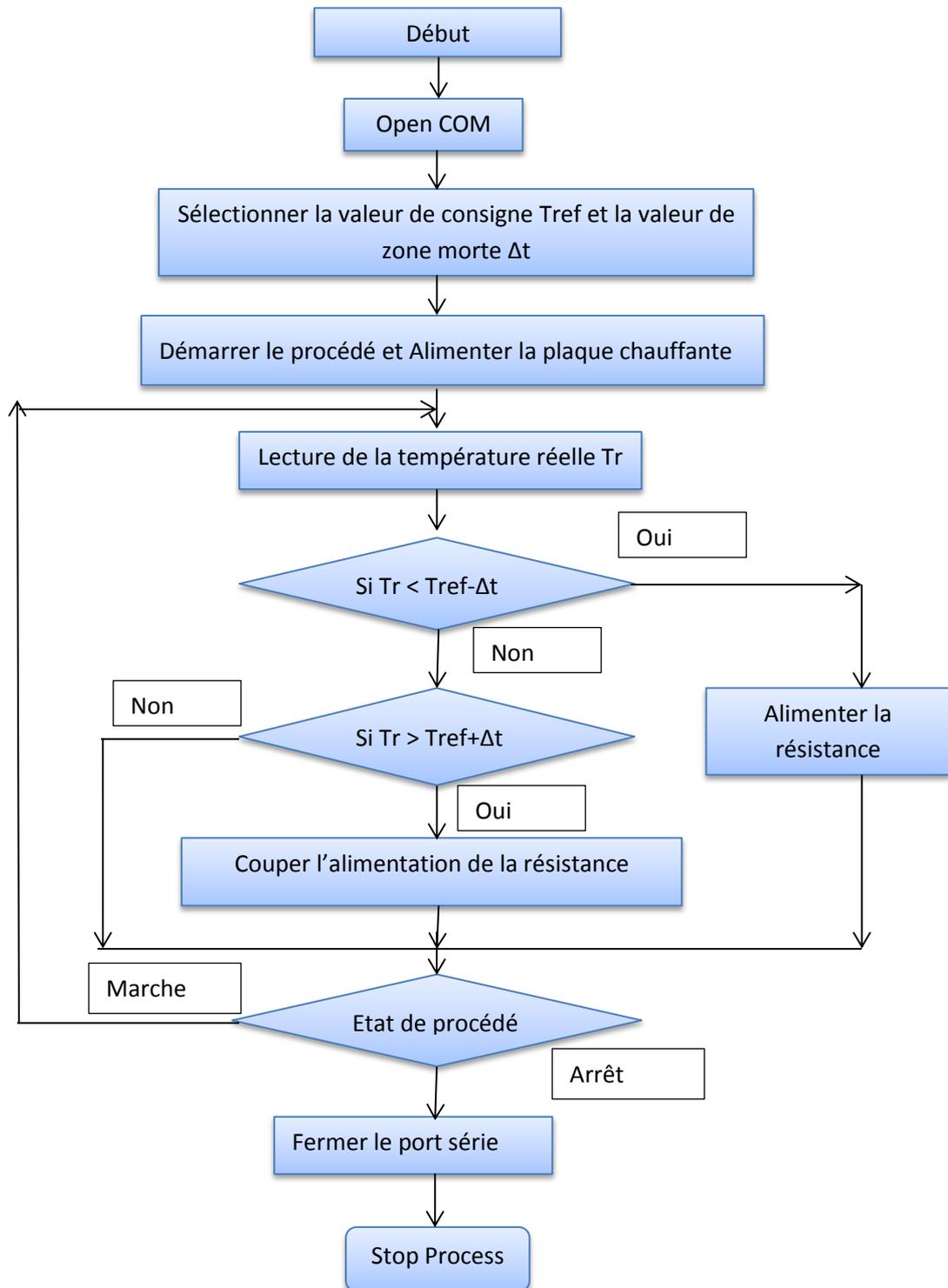


Figure.V.10 : Organigramme de commande avec contrôleur ON-OFF à hystérésis.

V-4-2-3. Résultats et interprétation :

Après implémentation du programme informatique et exécution de ce dernier, nous avons enregistré la courbe de réponse pour un échelon de $T_{ref} = 50$ degrés et $\pm \Delta t = 5$ degrés. Les courbes de réponses relatives à le four et le signal de commande sont portées en figure V-11.

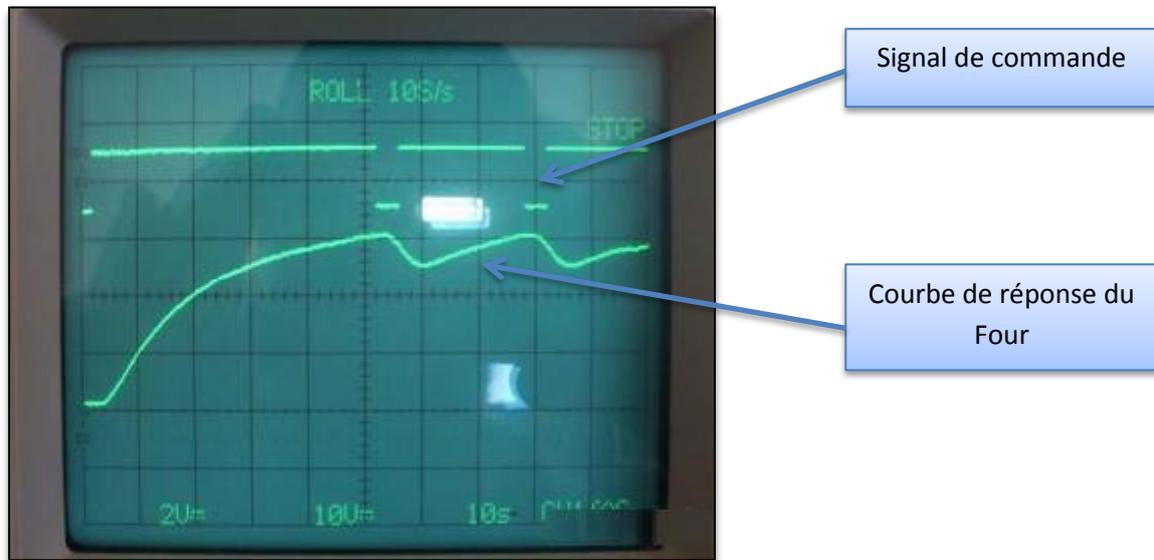


Figure.V.11: Courbe de signal de la commande et courbe de réponse du four avec un contrôleur ON-OFF a hystérésis.

V-4-3. Contrôleur PI non-linéaire :

V-4-3-1. La structure de la commande :

Le régulateur PI est un régulateur numérique associée à une caractéristique non linéaire afin d'éviter la saturation de l'unité de conversion DAC avec son amplificateur, donc le régulateur fonctionne d'une façon non linéaire. Les fonctions de consigne T_{ref} , comparaison et l'algorithme de la régulation PI, sont pris en charge par l'ordinateur

L'intérêt principal de ce correcteur est d'ajouter dans la chaîne de commande une intégration. On sait bien que la présence d'une intégration permet d'éliminer l'erreur de réglage. La **Figure V-12** illustre la structure d'un régulateur PI non linéaire.

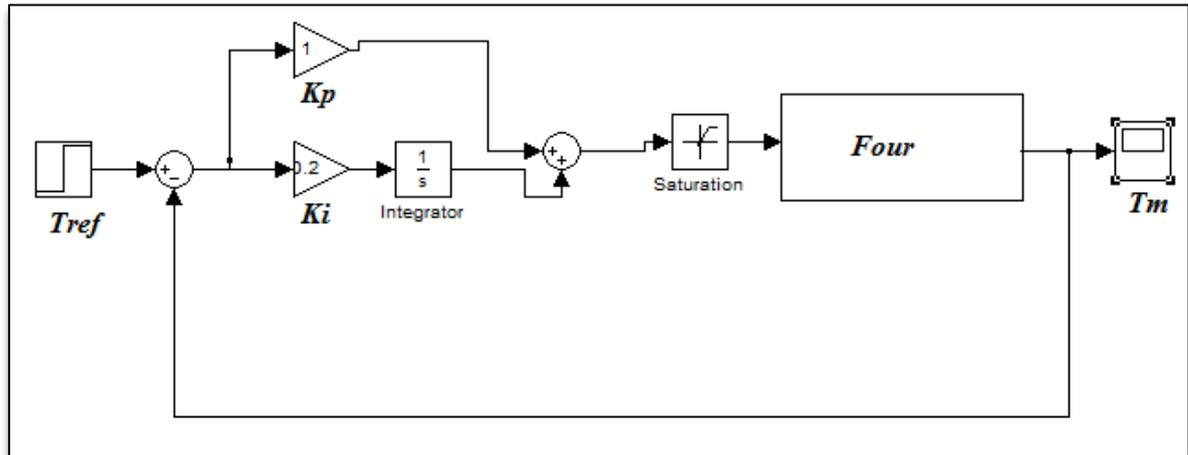


Figure V-12 : Schéma fonctionnel du contrôleur PI avec anti-windup.

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

$$C(s) = Kp[1 + Ti * \frac{1}{s}] \quad (V-1)$$

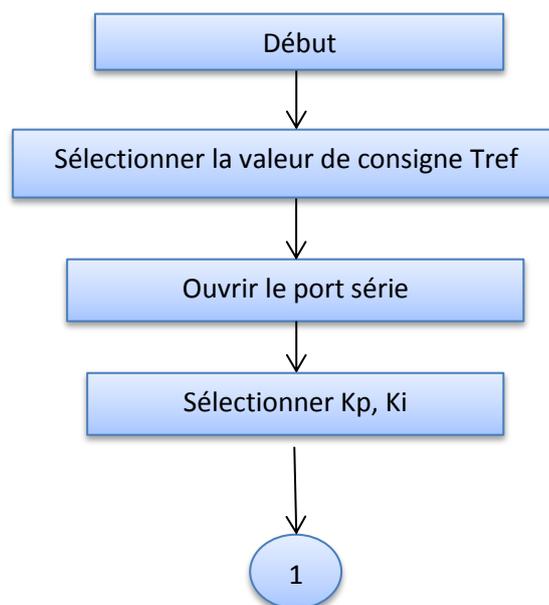
L'algorithme de ce correcteur est donné par l'expression récurrente suivante :

$$u(k) = Kp * e(k) + Ki \sum_{i=0}^k e(i) \quad (V-2)$$

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

V-4-3-2. Implémentation de la logique de commande

Pour illustrer la structure de cette commande, la figure V-13 présente l'organigramme simplifié décrivant le fonctionnement du four avec ce type de commande.



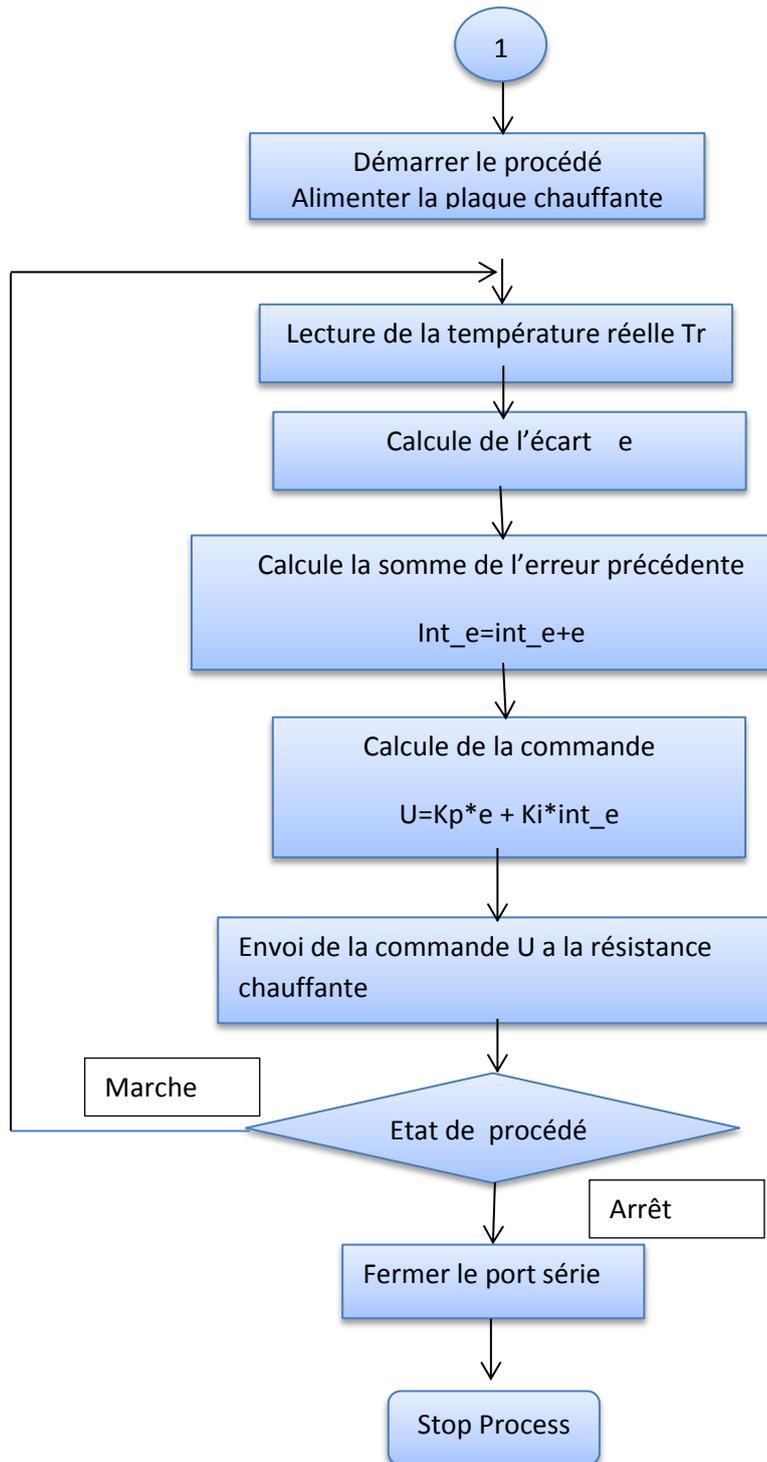


Figure V.13 : Organigramme globale de la régulation

V-4-3-3. Résultats et interprétation

L'exécution du programme nous a permis de construire la courbe de réponse du four pour une référence de 50°C avec un gain $K_p = 0.22$ et $T_i = 0.02$ et une période d'échantillonnage de 1000ms.

Les courbes de la réponse de four et de la commande sont illustrées dans la figure V.14.

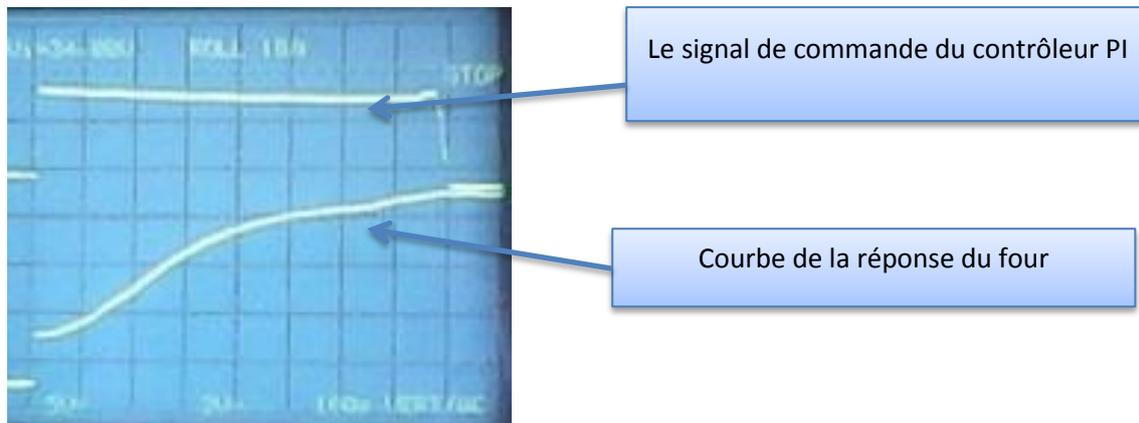


Figure V-14 : la commande et la réponse du four avec un contrôleur PI non linéaire.

L'analyse de cette courbe obtenue sur notre software, nous montre que l'erreur entre la consigne « Tc » et la température réelle (via le capteur AD590) diminue sans atteindre la valeur zéro. Ceci est dû aux non linéarité de la dynamique du four et aux caractéristiques des amplificateurs convertisseurs utilisés aux niveaux de la carte d'interface que nous avons développée et utilisée. Nous constatons également que la courbe de réponse du four change avec la référence proposée. C'est pourquoi il est nécessaire d'implémenter d'autres algorithmes de commande afin d'améliorer les performances de notre asservissement numérique.

V-5. Conclusion :

L'ensemble des programmes que nous avons implémentés à l'aide d'un langage orienté objet tel que Visual C# sous Windows ont été appliqués au contrôle de la température d'un four électrique au niveau du laboratoire de recherche.

La régulation de notre système consiste à implémenter un programme de régulation utilisant un contrôleur ON-OFF, un contrôleur ON-OFF à hystérésis, et un contrôleur PI avec élément non linéaire pour éviter la saturation du DAC.

Enfin, en analysant les résultats des stratégies de commande que nous avons traitées dans ce chapitre, la régulation avec le régulateur PI avec l'élément non linéaire présente des meilleures performances par rapport à la régulation avec un contrôleur ON-OFF. Cependant, compte tenu de la non-linéarité du processus, il est nécessaire d'implanter un contrôleur PI adaptatif ou les paramètres seront ajustés selon le point de fonctionnement.

Conclusion générale

Conclusion générale:

Dans ce travail, nous avons étudié et réalisé un système de supervision sous PC pour le contrôle d'un four électrique de laboratoire. La partie matérielle est constituée d'une carte d'interface électronique à base de microcontrôleur permet une jonction entre le processus et le software via le port USB du Micro-ordinateur. D'autre part, nous avons contribué à améliorer les stratégies de commande du four toute en utilisant trois types de régulateurs. Nous avons développé un logiciel de supervision par un langage de programmation orienté objet afin d'améliorer le fonctionnement de procédé en utilisant plusieurs interfaces pour visualiser les paramètres de ce procédé.

En premier lieu, nous avons identifié la dynamique du four avec ses trois organes (four, actionneur et capteur de température) en tenant compte des non linéarités du système. Ceci a conduit à établir son modèle identifié à partir de sa réponse indicelle toute en basant sur la méthode d'identification graphique et utilisant le toolbox «ident » de MATLAB.

Dans la deuxième phase de ce travail qui est importante, nous avons étudié et réalisé une carte d'interface électronique à base de microcontrôleur de type PIC18F4550, constituée essentiellement de organes d'affichage (LCD, baregraphe) et d'unités de conversion analogiques adaptés aux ports séries du PC par le module UART. Cette structure permet de générer des signaux (entrées – sorties) exploitable par les procédés industriels. Cette partie hardware a été testée sur banc d'essai afin de s'assurer du bon fonctionnement et de sa fiabilité avant son branchement avec le calculateur.

Dans la troisième phase de ce travail qui est la plus importante, nous avons étudié et réalisé un software de supervision sous Windows. Le software que nous avons élaboré et implémenté sous forme d'interface graphique à l'aide d'un langage orienté objet tel que Visual C#2013 permettant un pilotage de l'ensemble des paramètres du process de manière visuelle. Il permet d'effectuer plusieurs taches telles que l'historisation des variables du procédé, la gestion des alarmes, L'enregistrement des données, et l'exécution des algorithmes de commande approprié.

Le dernier chapitre a été consacré à l'implémentation des algorithmes de contrôle du four. Ces algorithmes est intégré dans notre software de supervision et de la commande, Ce Software à comme fonction de générer la référence de la température, la comparaison avec la grandeur réelle et enfin l'élaboration des algorithmes de réglage que nous avons retenu. Nous avons élaboré trois types de commande, un régulateur ON-OFF

Conclusion générale

a hystérésis, un régulateur ON-OFF, et un régulateur PI avec un élément non linéaire afin d'éviter la saturation du DAC et son amplificateur.

A l'issu des résultats que nous avons obtenu nous constatons que les paramètres de ce contrôleur doivent être modifiés en fonction du point de fonctionnement du four. Ceci nécessite l'implémentation d'un nouveau software de type PI adaptatif. Cet algorithme permet une sélection du gain (K_p) et de la constante de temps intégrale (K_i) en fonction de la référence de température. Cette dernière variante est devenue nécessaire compte- tenu de la non linéarité de l'ensemble four –circuiterie électronique. Nous avons également implémenté un contrôleur PI avec anti-windup.

A travers ce projet, l'un des objectifs que nous nous sommes fixé était de concevoir et mettre en œuvre une solution pour l'implémentation des régulateurs de température pour des faibles coûts. Notre projet peut avoir de nombreuses applications dans le domaine de l'industrie et peut être facilement adapté pour répondre à d'autres paramètres à contrôler tels que la pression, le débit, le niveau...etc.

Références bibliographiques

Références

- [1] *J. F. Bourgeois, A. Girault, R. Jaume, M. Le Boulch, C. Oberlin*, « Four électrique à résistances », *Technique de l'ingénieur D5901*.
- [2] *www.wikipédia.com*, « Four », *Un article de Wikipédia, l'encyclopédie libre*.
- [3] **Paul Klonowski** , « Use of the AD590 Temperature Transducer in a Remote Sensing Application », *ANALOGUE DEVICES*.
- [4] **DOTOU Samuel**, « Etude des fours électriques présenté », Un article de site web « www.academia.edu »
- [5] **Prof. Belkacem OULD BOUAMAMA**, « Identification des Systèmes », Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS - UMR CNRS 8219).
- [6] **Prof. BELKACEM OUELD BOUAMAMA**, <Methode d'identification de strejc>, Un article de site web « www.polytech-lille.fr ».
- [7] **Vincent MAHOUT**, « Introduction à l'identification Présentation rapide et incomplète de la toolbox Ident de Matlab » *INSA de Toulouse*, 2011.
- [8] **Dr.BENHALLA**, « Modelisation et identification », December 2012, Version 2.
- [9] **Benjamin Bradu**, « Identification paramétrique : la méthode des moindres carrés », 24 Novembre 2006.
- [10] **Ludovic Blanc, Vincent Raimbault**, « contrôle en température d'un four cylindrique a trois zones de chauffe », *Departement de genie electrique, Ecole polytechnique Montreal*.
- [11] « www.labcenter.com », « Proteus », Une Description generale de logiciel Proteus, Labcenter Electronics.
- [12] **CHRISTIAN TAVENNOR**, « Microcontrolleur PIC18F, description et mise en œuvre », *DUNOD*, 2008.

- [13] *Datasheet*, « *DAC0831 ADC0801/ADC0802/ADC0803/ADC0804/ADC08058-Bit μ P Compatible A/D Converters* », *Novembre 1999, page 1.*
- [14] **C.E.spark**, « **data acquisition device** », **Database national semiconductor corporation, 1989.**
- [15] **www.wikipédia.com**, < UART >, Un article de Wikipédia, l'encyclopédie libre.
- [16] **Prof Jacques Lepot**, <le port serie, Uart>, Un article de site web « **www.reality.be/elo/labos2** ».
- [17] **www.wikipédia.com**, < PIC microcontrôleur >, Un article de Wikipédia, l'encyclopédie libre.
- [18] **www.prolific.com.tw**, <USB to UART/RS232/Serial>, Prolific Technology Inc.
- [19] **www.wikipédia.com**, < *Microsoft Visual Studio* >, *Un article de Wikipédia, l'encyclopédie libre.*
- [20] **Herbert Schildt**, <c# 4.0 The complete reference >, **McGrawHill, 2010.**
- [21] **www.wikipédia.com**, < *C Sharp* >, *Un article de Wikipédia, l'encyclopédie libre.*
- [22] **https://msdn.microsoft.com**, site officiel du Microsoft.
- [23] **https://www.mikroe.com/mikroc/pic**, site officiel de MikroC.
- [24] **www.wikipédia.com**, < *Supervision* >, *Un article de Wikipédia, l'encyclopédie*
- [25] **Pierre bonnet**, « introduction à la supervision », Université de Lille, **Novembre 2010.**
- [26] **Jean-François PETIN**, « Modèles, méthodes et outils pour l'analyse, la conception et l'implantation des Systèmes de supervision industrielle ».

Annexe

```

// LCD module connections
sbit LCD_RS at RB0_bit;
sbit LCD_EN at RB1_bit;
sbit LCD_D4 at RB2_bit;
sbit LCD_D5 at RB3_bit;
sbit LCD_D6 at RB4_bit;
sbit LCD_D7 at RB5_bit;

sbit LCD_RS_Direction at TRISB0_bit;
sbit LCD_EN_Direction at TRISB1_bit;
sbit LCD_D4_Direction at TRISB2_bit;
sbit LCD_D5_Direction at TRISB3_bit;
sbit LCD_D6_Direction at TRISB4_bit;
sbit LCD_D7_Direction at TRISB5_bit;
// End LCD module connections

volatile unsigned char loopVar = 0;
unsigned char uart_rd[4]={'0','0','0','\0'};
unsigned char str1[7];
unsigned char str_temp[7];
unsigned char str[13];
unsigned char output[7];
unsigned int i=0;
unsigned int j=0;
unsigned int ni=0;
unsigned int pot1;
unsigned int temp;
unsigned int start=0;
int temp_arr[4]={0,0,0,'\0'};
unsigned short port_val=0;
void sendData ()
{
    str[0]=': ';
    j=0;
    for(i=0;i<3;i++)
    {
        //lire la valeur delivre par le capteur i
        pot1=ADC_Read(i);
        pot1=pot1/4;
        IntToStr(pot1,str1);
        pot1=(pot1*100)/255;
        temp_arr[i]=pot1;

        if (pot1<1000 && pot1>99)
        {
            str[j+1]=str1[3];
            str[j+2]=str1[4];
            str[j+3]=str1[5];
        }
        else if (pot1<100 && pot1>9)
        {
            str[j+1]='0';
            str[j+2]=str1[4];
            str[j+3]=str1[5]; }
        else if (pot1<10)
        {
            str[j+1]='0';
            str[j+2]='0';
            str[j+3]=str1[5];
        }
        if(j<5){ str[j+4]='*'; }
    }
}

```

```

        j=j+4;
    }

    str[12]='\0';

    //calculer la moyenne
    temp=(temp_arr[0]+temp_arr[1]+temp_arr[2])/3;
    //enregistrer la valeur de temperature
    IntToStr(temp,str_temp);
    //envoi de ces valeurs vers le PC
    UART1_Write_Text(str);
}

void InitTimer0 () {
    TOCON      = 0x86;
    TMR0H      = 0x67;
    TMR0L      = 0x69;
    GIE_bit    = 1;
    TMR0IE_bit = 1;
}

void Interrupt () {
    if (TMR0IF_bit) {
        TMR0IF_bit = 0;
        TMR0H      = 0x67;
        TMR0L      = 0x69;
        SendData ();
    }
}

void main ()
{
    //initialiser les parametres de Timer0
    InitTimer0 ();
    //Configurer l'ADC
    ADC_Init ();
    ADCON1=0b00001011; // AN0--AN3 ==analog AN4--AN12== digital
    ADCON2=0b00101101; // 12 Tad / Fosc/16
    //PORTA 0--3 analog input
    TRISA=0b00001011;
    CMCON=0b00000111; //desactive les comparateurs
    //declarer le PORTD comme sortie
    TRISD=0x00;
    PORTD=0x00;
    TRISB.B7=0x00; //Declarer le pin 6 et 7 de PORTB comme sortie
    TRISB.B6=0x00;
    TRISC.B2=0x00; //Declarer le pin 2 de PORTC comme sortie
    TRISC.B1 = 0x00; //Declarer le pin 1 de PORTC comme sortie
    PWM1_Init(5000); //initializer Pwm a 5KHz
    delay_ms(250);
    // Configure UART
    UART1_Init(9600);
    //portb as digital output
    //TRISB=0x00;
    CVRCON.CVROE=0;
    CVRCON.CVREN=0; //disable comparator voltage reference
    Lcd_Init (); // Initialize LCD
    //Affichage au debut de demmarage
    Lcd_Cmd(_LCD_CLEAR); // Clear display
    Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off
    Lcd_Out(1,1,"Systeme de");
    Lcd_Out(2,1,"la supervision");
}

```

```

delay_ms(2000);
Lcd_Out(1,1,"Realisee Par:");
Lcd_Out(2,1,"BOUROUIS Thabet");
    PWM1_Start(); // start PWM1
    PWM1_Set_Duty(port_val);
    Delay_ms(2000);
    Lcd_Cmd(_LCD_CLEAR);
while(1)
{
    if(PORTA.B4==1)
    {
        start=1;
    }else{
        start=0;
    }
    //Si le procede mise en marche
    if(start=1){
        PORTB.B6=0; //led orange off
        PORTB.B7=1;//led vert on
        //Recevez Le signal de commande
        if (UART_Data_Ready() == 1) {
            UART1_Read_Text(output, ";", 6);
            for(ni=0;ni<7;ni++)
            {
                if(output[ni]=='*' && ni<3 &&
output[ni+4]==':') {
                    uart_rd[0]=output[ni+1];
                    uart_rd[1]=output[ni+2];
                    uart_rd[2]=output[ni+3];
                }
            }
            port_val=atoi(uart_rd);
            PORTD=port_val;
            PWM1_Set_Duty(port_val);
        }
        //Afficher la valeur du temperature
        Lcd_Out(1,1,"Temp=");
        Lcd_Chr(1, 7, str_temp[3]);
        Lcd_Chr(1, 8, str_temp[4]);
        Lcd_Chr(1, 9, str_temp[5]);
        //Afficher le signal de commande
        Lcd_Out(2,1,"Com=");
        Lcd_Out(2,8,uart_rd);
    }
    //si le procede en etat d'arret
    if(start=0){
        PORTB.B6=1; //led orange on
        PORTB.B7=0;//led vert off
    }
}
}
}

```

