



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER PROFESSIONNEL

Domaine : Sciences et Technologies

Filière : Génie électrique

Spécialité : **Automatique et informatique
industrielles**

Présenté par :

**Haffad salem.
Bourdjioua slimane.**

Mémoire dirigé par Mr **Takfarinas CHELLI** et codirigé par Mr **Nassim TADJINE**

Thème

Réalisation d'un régulateur PID à base d'un microcontrôleur.

Mémoire soutenu publiquement le 14/07/ 2015 Devant les jurys composé de :

M O.MEGHERBI

MAA, Dpt d'automatique, Président

M Takfarinas CHELLI

MAA, Dpt d'automatique, Encadreur

M Nassim TADJINE

Ingénieur, Cevital, Co-Encadreur

M F.BOUDJEMAA

MAA, Dpt d'automatique, Examineur

M O.BOUKEUNDOUR

MAB, Dpt d'automatique, Examineur

Promotion 2015

Remerciement

Le présent document couronne nos études universitaire très riches en enseignements et expériences tant sur le plan du savoir scientifique que de celui de la vie en collectivité.

Nous louons Dieu le tout puissant pour nous avoir prêté force et patience pour l'aboutissement de ce modeste travail.

Nous tenons à remercier très chaleureusement Mr T.CHELLI qui nous a permis de bénéficier de son encadrement. Les conseils qu'il nous a prodigué, la patience, la disponibilité, la confiance qu'il nous a témoigné ont été déterminantes dans la réalisation de notre travail.

Nous exprimons aussi nos meilleurs sentiments de gratitude aux membres du jury d'avoir accepté de faire part du jury et consacré leurs temps à la lecture et à la correction de ce manuel.

Nous ne terminerons pas ce mot sans gratifier de nos remerciements nos parents pour leurs contributions, leur soutien et leur patience durant tout notre parcours scolaire.

Nous souhaitant adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire, à tous nos proches et ami(e)s. qui nous ont toujours soutenus et encouragé au cours de notre formation.

Dédicace

Je dédie ce modeste travail à :

- *A mon Cher père, qui a toujours cru en moi et a mis à ma disposition tous les moyens nécessaires pour que je réussisse dans mes études.*
- *A ma chère mère, que je ne cesse de remercier pour tout ce qu'elle m'a donné. Que Dieu la récompense pour tous ses bienfaits.*
- *A mes frères YIDHIR, AHMED et à mes sœurs YAMINA, SALMA, FAYZA.*
- *A tous mes ami(e)s qui m'ont aidé de près ou de loin.*
- *A mon binôme BOURDJIOUA Slimane qui n'a pas cessé de m'encourager.*

HAFFAD SALEM

Dédicace

Je dédie ce modeste travail à :

- *A mon Cher père, qui a toujours cru en moi et a mis à ma disposition tous les moyens nécessaires pour que je réussisse dans mes études.*
- *A ma chère mère, que je ne cesse de remercier pour tout ce qu'elle m'a donné. Que Dieu la récompense pour tous ses bienfaits.*
- *A mes frères et à mes sœurs.*
- *A tous mes ami(e)s qui m'ont aidé de près ou de loin.*
- *A mon binôme HAFFAD Salem qui n'a pas cessé de m'encourager.*

BOURDJIOUA SLIMANE

Sommaire

Introduction générale.....	1
----------------------------	---

Chapitre I

La régulation

Introduction.....	2
I.1. Notion de système.....	2
I.1.1 Système physique	4
I.1.2. Notion de modèle	4
I.1.2.1. Modèle mathématique	4
I.1.2.2. Fonction de Transfert	5
I.1.3. Système de 1 ^{er} ordre	5
I.1.4. Système 2 ^{em} ordre	7
I.2. Historique de la régulation	12
I.3. Caractéristiques d'un système asservi	13
I.4. Principe de fonctionnement.....	13
I.4.1. Régulation en boucle ouverte (Manuelle).....	14
I.4.2. La régulation en boucle fermée.....	15
I.5. Représentation fonctionnelle d'une boucle de régulation	16
I.6. Classification des régulateurs	18
I.7. Les régulateurs à action P, I, D.....	19
I.7.1. Les actions P, I, D	19
I.7.2. Le régulateur PI.....	24
I.7.3. Le régulateur PID	24
I.8. Le PID numérique.....	27
Conclusion	29

Chapitre II

Réalisation d'une boucle de régulation à base d'un microcontrôleur

Introduction.....	30
II.1. Description de l'application	30
II.2. Conception du circuit	30
II.2.1. Première démarche : le choix du microcontrôleur	30
II.2.1.1. Le pic 16f877A caractéristique et organisation mémoire.....	31
II.2.1.1.1. Les principales caractéristiques du PIC 16F877A.....	31
II.2.1.1.2. Schéma fonctionnel du 16f877A	31
II.2.1.1.3. Organisation interne du PIC 16F877A	31
II.2.1.1.4. Organisation externe du PIC 16F877A.....	32
II.2.1.2. Le convertisseur analogique numérique du PIC 16F877A.....	34
II.2.1.2.1. Résolution d'un convertisseur analogique-numérique	34
II.2.1.2.2. L'organigramme d'acquisition de la mesure.....	35
II.2.1.2.3. Échantillonnage de l'information	35
II.2.1.2.4. Détail de fonctionnement du CAN contenu dans le 16F877A.....	36
II.2.2. Deuxième démarche : circuit de base pour le fonctionnement du PIC.....	40
II.2.3. Troisième démarche : la conversion numérique-analogique	43
II.2.4. Quatrième démarche : bloc d'affichage LCD et clavier (boutons poussoirs).	45
II.2.4.1. blocs d'affichage LCD.....	45
II.3. Conception software	48
II.3.1. Introduction	48
II.3.2. Présentation du MikroC.....	49
II.3.3. Procédures de création de programme sous mikroC PRO for PIC.....	50
II.3.4. Organigramme du Programme implémenté dans le PIC.....	52
Conclusion	53

Chapitre III

Simulation et tests

Introduction.....	54
III.1. Logiciel de simulation et conception de circuit imprimé	54
III.2. Test de bloc de la CNA, CAN et l'action proportionnelle	55
III.3. Teste de l'action intégrale.....	57
III.4. Teste de l'action dérivée.....	58
III.6. Teste de la régulation d'un système 1 ^{er} ordre	60
III.7. Teste de la régulation d'un système deuxième ordre	62
Conclusion	66
Conclusion générale	67

Liste des figures

Figure I.1 : Notion de système.	2
Figure I.2 : Types d'entrées d'un système.	3
Figure I.3 : Un moteur à courant continu.	4
Figure I.4 : Réponse impulsionnelle d'un système du 1^{er} ordre.	6
Figure I.5 : Réponse indicielle d'un système du 1^{er} ordre.	7
Figure I.6 : Réponse impulsionnelle en fonction de ξ pour w_0 fixé.	10
Figure I.7 : Réponse indicielle en fonction de ξ pour w_0 fixé.	11
Figure I.8 : Clepsydre de Ktesibios (-270 av. J.C., Alexandrie).	12
Figure I.9 : Système de régulation de manière générale.	14
Figure I.10 : Régulation en boucle ouverte	15
Figure I.11 : Régulation en boucle fermée.	16
Figure I.12 : Représentation fonctionnelle d'une boucle de régulation.	16
Figure I.13 : schéma fonctionnel d'un régulateur.	17

Figure I.14 : Influence de T_i sur la commande.	20
Figure I.15 : Influence de T_d sur la commande.	21
Figure I.16 : Schéma synoptique d'un régulateur PID.	25
Figure I.17 : Essai indiciel.	27
Figure II.1 : Brochage du 16f877A	33
Figure II.2 : Principe de conversion Analogique/Numérique	34
Figure II.3 : L'organigramme d'acquisition de la mesure.	35
Figure II.4 : Structure interne du CAN 10 bits des 16F877A	36
Figure II.5 : Déroulement d'une conversion analogique-numérique	40
Figure II.6 : Circuit d'alimentation	41
Figure II.7 : Circuit de l'horloge	42
Figure II.8 : Le circuit RESET du pic	42
Figure II.9 : Schéma synoptique de principe de déroulement de la CNA	43
Figure II.10 : Convertisseur à résistances pondérées	43
Figure II.11 : Brochage du DAC0808	44
Figure II.12 : Schéma du câblage du DAC0808 avec l'amplificateur LF351	45
Figure II.13 : Schéma fonctionnel d'un afficheur LCD	46
Figure II.14 : photo de LCD et son brochage	46
Figure II.15 : Disposition des boutons de réglages sur la carte.	48
Figure II.16 : Interface du logiciel mikroC PRO for PIC	49
Figure II.17 : Fenêtre de création d'un nouveau projet	50
Figure II.18 : Fenêtre pour le choix du pic	50
Figure II.19 : Fenêtre pour la configuration de l'horloge	51
Figure II.20 : Fenêtre pour le choix de l'emplacement de projets à créer	51
Figure III.1 : Interface de logiciel de simulation ISIS	54
Figure III.2 : Circuit de PIC et CNA de régulateur PID	55

Figure III.3 : Signaux d'entrée et de sortie pour $k_p=1$	56
Figure III.4 : Signaux d'entrée et de sortie pour $k_p=2$	56
Figure III.5 : Signaux d'entrée et de sortie pour $k_i=1$	57
Figure III.6 : Signaux d'entrée et de sortie pour $k_d=9$	58
Figure III.7 : Figure illustrant le fonctionnement de LCD avec les BP	59
Figure III.8 : Signaux de commande et de mesure d'une régulation PID d'un système 1 ^{er} ordre	60
Figure III.9 : le schéma de simulation d'une boucle de régulation.....	61
Figure III.10 : signale de mesure.....	61
Figure III.11 : signale de commande.	61
Figure III.12 : Signaux de commande et de mesure d'une régulation PID d'un système 2 ^{èm} ordre $\xi > 1$	62
Figure III.13 : signale de mesure $\xi > 1$	63
Figure III.14 : signale de commande $\xi > 1$	63
Figure III.15 : Signaux de commande et de mesure d'une régulation PID d'un système 2 ^{èm} ordre $\xi = 1$	64
Figure III.16 : signale de mesure $\xi = 1$	64
Figure III.17 : signale de commande $\xi = 1$	65
Figure III.18 : Signaux de commande et de mesure d'une régulation PID d'un système 2 ^{èm} ordre $\xi < 1$	65
Figure III.19 : signale de mesure $\xi < 1$	66
Figure III.20 : signale de commande $\xi < 1$	66

Liste des tableaux

Tableau I.1 : les racines de l'équation caractéristique du système de deuxième ordre.....	8
Tableau I.2 : résumé des actions des corrections P, I et D.	23
Tableau I.3 : Structures des régulateurs PID.	26
Tableau I.4 : Choix d'une méthode de correction en fonction de l'indice de réglabilité (Cas des systèmes stables).....	27
Tableau I.5 : Choix d'une méthode de correction en fonction de l'indice de réglabilité (Cas des systèmes instables).....	27

Tableau II.1 : la disposition des bits de résultats dans les 2 octets en fonction de la valeur du bit ADFM.....	37
Tableau II.2 : Configuration de la nature des broches analogique du pic.....	39
Tableau II.3 : Les valeurs limite d'un CNA de type DAC0808	44
Tableau II.4 : Rôle des différentes broches de l'afficheur LCD	47

Introduction générale

Introduction générale

Les techniques de régulation ne sont pas seulement un moyen de commander des processus mais aussi un moyen de réduire les pertes de production, d'augmenter la qualité et la quantité des produits, d'augmenter la disponibilité des unités et de déterminer les coûts marginaux de production.

Au vue de l'importance primordiale que tient la régulation dans le domaine industriel il nous a paru incontournable le fait de réaliser un régulateur qui nous permette d'appliquer les principes de la régulation acquis tous au long de nos études, et ce non seulement pour consolider nos acquis théoriques mais aussi pour découvrir d'autres aspects pratiques lié à la régulation.

Le régulateur standard le plus utilisé dans l'industrie est le régulateur PID (proportionnel intégral dérivé) car il permet de régler à l'aide de ses trois paramètres les performances d'une régulation d'un processus modélisé.

Le régulateur PID est bien adapté à la plupart des processus de type industriel et est relativement robuste par rapport aux variations des paramètres du procédé.

Notre choix pour la réalisation de ce régulateur s'est porté sur le microcontrôleur de type PIC pour sa simplicité de programmation et le large éventail de domaines d'applications qu'il nous offre sans oublier aussi son faible coût économique.

Nous pouvons qualifier ce projet de pluridisciplinaire. En effet, nous avons travaillé dans différents domaines tels que : l'électronique analogique, l'électronique numérique, l'automatique et l'informatique pour la programmation des microcontrôleurs.

Pour mettre au point notre régulateur on a opté en premiers lieux pour l'étude des différentes notions des systèmes, de la régulation et des régulateurs. Nous avons entamé la conception software et le circuiterie de notre carte électronique toutes en présentant tous les composants électroniques qui composent cette dernière, en seconds lieux. Enfin nous avons pu procéder aux tests de simulation de chaque bloc de notre carte.

Chapitre I

La régulation

Introduction

La régulation est l'action de régler automatiquement une grandeur de telle sorte que celle-ci garde constamment sa valeur ou reste proche de la valeur désirée, quelles que soient les perturbations qui peuvent subvenir.

Principe de fonctionnement de la régulation peut se résumer en ces trois mots clefs :

- **Mesurer.**
- **Comparer.**
- **Corriger.**

Cependant, chaque procédé possède ses exigences propres, chaque appareil possède ses propres conditions de fonctionnement. Il est donc indispensable que la régulation soit conçue pour satisfaire aux besoins particuliers liés à la sécurité, aux impératifs de production et aux matériels. [1]

I.1. Notion de système

Un système est un dispositif isolé, soumis aux lois de la physique, de la chimie, de la biologie, de l'Economie, etc....

Définition (Système) : Un système est un ensemble d'éléments interconnectés pour accomplir une tâche prédéfinie. Il est affecté par une (ou plusieurs) variable(s). Le résultat de l'action des entrées est la réponse du système caractérisée par l'évolution d'une ou plusieurs variables : les sorties. Voir figure I.1 ci-dessous. [2]

Remarque : Un système industriel est souvent appelé processus.



Figure I.1 : Notion de système.

Les systèmes que l'on abordera possèdent les caractéristiques suivantes :

Représentation : un système est généralement représenté par un schéma fonctionnel sous forme de rectangle. Les signaux d'entrée appliqués à ce rectangle sont caractérisés par des flèches entrantes et annotées. L'action de ces entrées produit de cette manière (causale) des effets, mesurés par des signaux de sortie, représentés par des flèches sortantes et annotées.

SISO/MIMO : les entrées que l'on décide d'appliquer au système (et les sorties de ce système) peuvent être multiples : on parlera de système MIMO (Multi-Input Multi-Output, multi-entrées multi-sorties), ou uniques : on parlera de système SISO (Single-Input Single-Output, mono-entrée mono-sortie). L'objectif de ce projet se limitera aux systèmes SISO.

Nature des entrées : les entrées affectant un système peuvent être de différentes natures (voir figure I.2) :

La commande $e(t)$: elle a pour but d'exercer une action entraînant le fonctionnement souhaité du système.

La perturbation $d(t)$: il s'agit d'une entrée particulière (car elle est indépendante de notre décision) qui trouble le fonctionnement désiré du système.

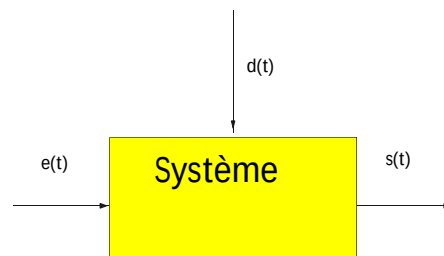


Figure I.2 : Types d'entrées d'un système.

Modèle : un système est caractérisé par des relations entrées/sorties exprimées sous la forme de lois mathématiques.

Système linéaire : les lois mathématiques entre l'entrée et la sortie sont des équations différentielles linéaires. Un système linéaire possède les propriétés de superposition et de proportionnalité. Le principe de superposition permet (entre autres) de décomposer l'étude de systèmes complexes en sous-systèmes plus simples à étudier.

Causalité : l'action (l'entrée) précède la réponse (la sortie).

Invariant dans le temps : le fonctionnement du système est le même quel que soit le moment ou l'instant où on l'utilise → les coefficients des équations différentielles sont constants et réels.

I.1.1 Système physique

D'après la définition, un système est excité (attaqué) par une entrée et délivre (répond par) une sortie. Dans un premier temps, on va faire figurer le processus physique l'intérieur d'un bloc auquel on va appliquer les entrées.

Exemple : Moteur

Considérons un moteur (voir figure 1.3) courant continu comme un système physique dont l'entrée de commande sera la tension U et la sortie, la vitesse (angulaire) de rotation de l'arbre moteur Ω .

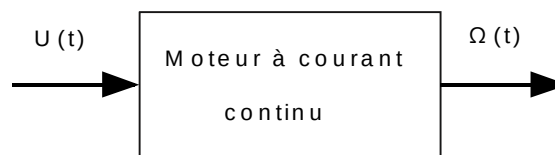


Figure I.3 : Un moteur à courant continu.

La relation mathématique entre $\Omega(t)$ et $U(t)$ caractérisera le moteur en tant que système.

I.1.2. Notion de modèle

I.1.2.1. Modèle mathématique [2]

Pour pouvoir utiliser les schémas précédents, il faut pouvoir décrire efficacement chaque processus physique situé à l'intérieur des blocs rectangulaires. Pour pouvoir analyser le comportement de ces processus, il est intéressant de pouvoir disposer d'un modèle mathématique de ces processus.

La phase de la modélisation est essentiel dans le processus de mise au point de régulateurs. Toutefois le but de ce projet n'est pas d'étudier en détail la modélisation des systèmes dynamiques. On se contentera de savoir modéliser rapidement et simplement.

Définition : on définit le modèle mathématique d'un système dynamique comme un ensemble d'équations qui représente le comportement dynamique du système avec la précision souhaitée. Il est obtenu en écrivant les lois de la physique qui régissent le comportement du système.

Obtention du modèle de comportement :

- Définir le système étudié et ses composants élémentaires.

- Formuler le modèle mathématique idéal (modèle de connaissance) et dresser la liste des hypothèses à retenir.
- Ecrire les lois de la physique (équations différentielle).
- Définir le modèle dédié à la régulation (fonction de transfert).

I.1.2.2. Fonction de Transfert

Pour un système, la fonction de transfert est l'expression mathématique plus au moins complexe qui indique le rapport entre une fonction du signal de sortie et une fonction du signal d'entrée.

Définition : La fonction de transfert d'un système continu est le rapport de la transformée de Laplace de sa sortie sur la transformée de Laplace de son entrée en considérant les conditions initiales nulles. $F(p) = \text{Sortie}(p) / \text{Entrée}(p)$.

L'ordre du système est le degré le plus élevé du polynôme du dénominateur appelé polynôme caractéristique.

I.1.3. Système de 1^{er} ordre

Par définition, ces systèmes obéissent à une équation différentielle du premier ordre à coefficients constants :

$$x(t) = a \frac{dy(t)}{dt} + by(t) \quad (1.1)$$

Où $x(t)$ et $y(t)$ représentent respectivement l'entrée et la sortie du système.

La transformée de Laplace de l'équation, où les conditions initiales sont considérées nulles, Permet d'obtenir facilement la forme générale de la fonction de transfert des systèmes du premier Ordre :

$$F(p) = \frac{y(p)}{x(p)} = \frac{1}{ap+b} = \frac{1/b}{1+\left(\frac{a}{b}\right)p} \quad (1.2)$$

En utilisant les changements de variables $K = 1/b$ et $T = a/b$ on obtient la forme (appelée forme Standard) d'une fonction de transfert d'un système du premier ordre :

$$F(p) = \frac{K}{1+\tau p} \quad (1.3)$$

$F(p)$ possède 2 paramètres essentiels : K et τ

— **Le gain statique** du système K s'exprime dans la même unité que le rapport $\frac{\Delta y(t)}{\Delta x(t)}$.

Pour un système stable (la sortie est constante en régime permanent), le gain statique représente la valeur prise par sa réponse à une entrée en échelon unitaire quand $t \rightarrow \infty$

_ **La constante de temps** τ du système s'exprime en secondes et caractérise la vitesse (dynamique) de réaction d'un système.

Ces deux grandeurs suffisent à caractériser tout système du premier ordre : ce sont ses grandeurs (Paramètres) caractéristiques.

➤ Réponses temporelles

Il est intéressant de connaître et de reconnaître les réponses des systèmes du premier ordre aux signaux temporels classiques : impulsion, échelon, rampe. Cela permettra en particulier une comparaison avec les signaux réponses du système physique sur lequel on travaille.

• Réponse impulsionnelle

La réponse impulsionnelle d'un système est la sortie de ce système suite à une sollicitation en impulsion de Dirac :

$$x(t) = \begin{cases} \infty & \text{pour } t = 0 \\ 0 & \text{pour } t \neq 0 \end{cases} \Rightarrow X(p) = 1 \quad (I.4)$$

Les équations (I.3) et (I.4) permettent d'écrire l'expression de la réponse impulsionnelle :

$$y(p) = \frac{K}{1+\tau p} x(p) = \frac{K}{\tau} \frac{1}{p+\frac{1}{\tau}} \xrightarrow{\mathcal{L}^{-1}} y(t) = \frac{K}{\tau} e^{-\frac{t}{\tau}} \quad (I.5)$$

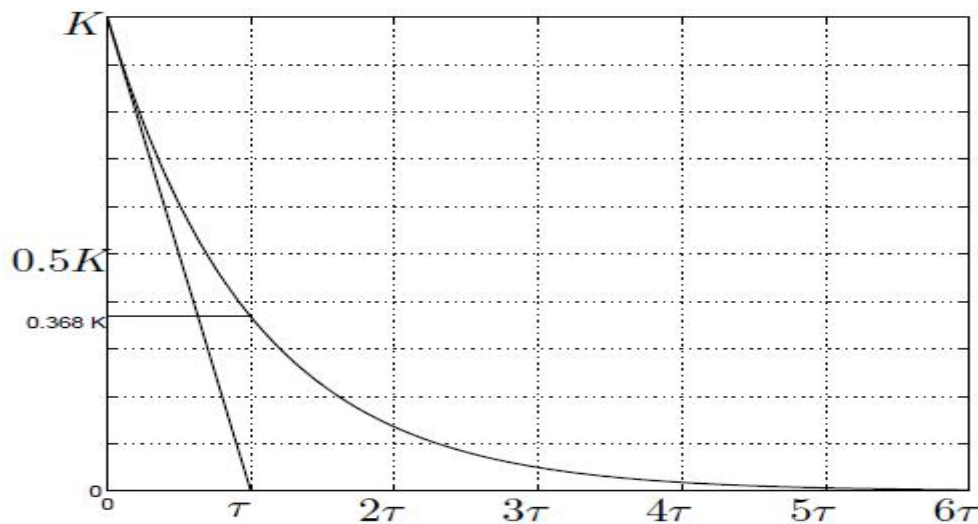


Figure I.4 : Réponse impulsionnelle d'un système du 1^{er} ordre.

• Réponse indicielle

La réponse indicielle d'un système est sa réponse à une entrée en échelon (entrée constante de valeur A). L'entrée du système est donc définie par l'équation :

$$x(t) = \begin{cases} A & \text{pour } t \geq 0 \\ 0 & \text{pour } t < 0 \end{cases} \Rightarrow X(p) = \frac{A}{p} \quad (I.6)$$

Les équations (I.29) et (I.32) permettent d'écrire l'expression de la réponse indicielle :

$$y(p) = \frac{K}{1+\tau p} \frac{A}{p} = AK \left(\frac{1}{p} - \frac{1}{p+\frac{1}{\tau}} \right) \xRightarrow{\mathcal{L}^{-1}} y(t) = AK(1 - e^{-\frac{t}{\tau}}) \quad (I.7)$$

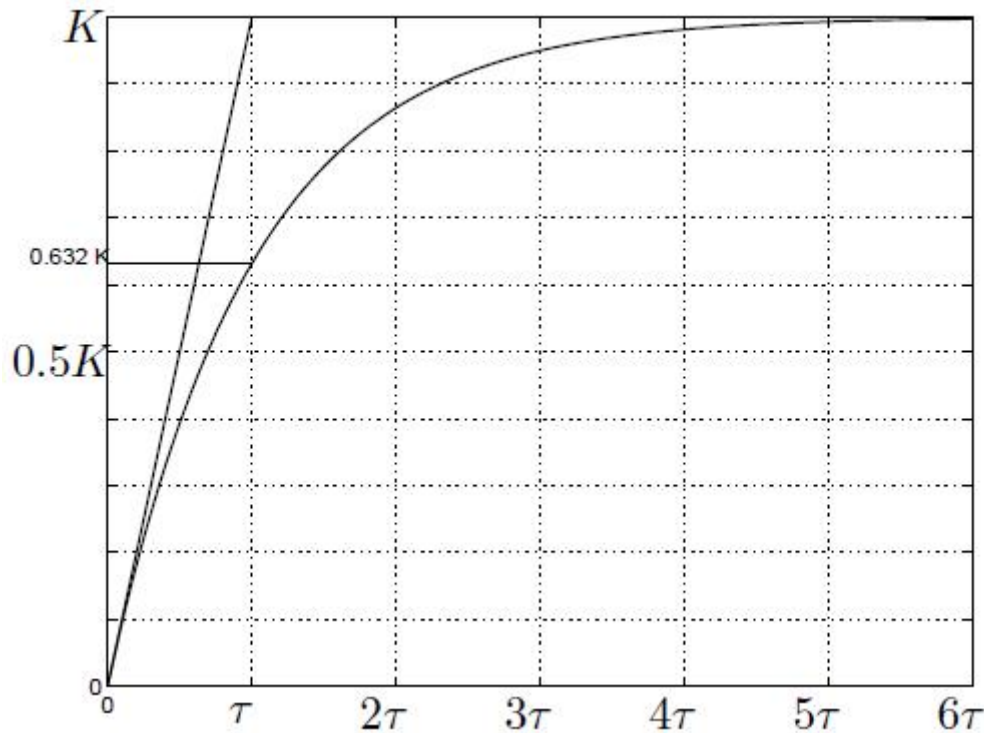


Figure I.5 : Réponse indicielle d'un système du 1^{er} ordre.

I.1.4. Système 2^{em} ordre

Par définition, ces systèmes obéissent à une équation différentielle du deuxième ordre coefficients constants :

$$x(t) = a \frac{d^2 y(t)}{dt^2} + b \frac{dy(t)}{dt} + cy(t) \quad (I.8)$$

Où $x(t)$ et $y(t)$ représentent respectivement l'entrée et la sortie du système.

La transformée de Laplace de l'équation 1.7, où les conditions initiales sont considérées nulles, permet d'obtenir facilement la forme générale de la fonction de transfert des systèmes du premier ordre :

$$G(p) = \frac{y(p)}{x(p)} = \frac{1}{ap^2 + bp + c} = \frac{Kw_0^2}{p^2 + 2\xi w_0 p + w_0^2} \quad (I.9)$$

_ K est appelé **gain statique** du système.

- _ w_0 est la **pulsation propre non amortie** du système (elle s'exprime en radians par seconde).
- _ ξ est le **facteur d'amortissement** du système.

Ces trois grandeurs suffisent à caractériser tout système du deuxième ordre : ce sont ses **grandeurs caractéristiques**.

➤ Etude des pôles d'un système du deuxième ordre

Les pôles d'une fonction de transfert $G(p)$ sont les valeurs de « p » qui annulent le dénominateur de $G(p)$. Ce sont donc les racines de l'équation caractéristique qui suit :

$$p^2 + \xi w_0 p + w_0^2 = 0 \quad (1.10)$$

Le discriminant réduit associé à cette équation est : $\Delta = (\xi^2 - 1)w_0^2$. Selon que ξ sera supérieur ou inférieur à 1, l'équation 1. Admettra des solutions réelles ou complexes conjuguées.

valeur de ζ	$\zeta > 1$	$\zeta = 1$	$\zeta < 1$
pôles p_{1-2}	$-\zeta\omega_0 \pm \omega_0\sqrt{\zeta^2 - 1} = \frac{-1}{\tau_{1-2}}$	$-\omega_0 = \frac{-1}{\tau_0}$	$-\zeta\omega_0 \pm j\omega_0\sqrt{1 - \zeta^2}$
transfert :	$F(p) = \frac{K}{(1+p\tau_1)(1+p\tau_2)}$	$F(p) = \frac{K}{(1+p\tau_0)^2}$	$F(p) = \frac{K\omega_0^2}{p^2 + 2\zeta\omega_0 p + \omega_0^2}$

Tableau I.1 : les racines de l'équation caractéristique du système de deuxième ordre

On voit donc que les pôles sont de natures différentes selon la valeur de ξ . Les réponses temporelles vont donc s'étudier en fonction de la valeur de ξ .

➤ Réponses temporelles

Il est intéressant de connaître et de reconnaître les réponses des systèmes du deuxième ordre aux signaux temporels classiques : impulsion, échelon, rampe. Cela permettra en particulier une comparaison avec les signaux réponses du système physique sur lequel on travaille.

• Réponse impulsionnelle

La réponse impulsionnelle d'un système est la sortie de ce système suite à une sollicitation en impulsion de Dirac :

$$x(t) = \begin{cases} \infty & \text{pour } t = 0 \\ 0 & \text{pour } t \neq 0 \end{cases} \Rightarrow X(p) = 1$$

✓ **Cas de $\xi > 1$:**

Dans ce cas on peut écrire la transformée de Laplace de la réponse impulsionnelle comme suit :

$$\begin{aligned}
 Y(p) &= F(p) \cdot X(p) = F(p) \\
 &= \frac{K}{(1 + \tau_1 p)(1 + \tau_2 p)} \\
 &= \frac{K}{\tau_1 - \tau_2} \left[\frac{\tau_1}{(1 + \tau_1 p)} - \frac{\tau_2}{(1 + \tau_2 p)} \right]
 \end{aligned}$$

Ou bien encore, dans le domaine temporel :

$$y(t) = \frac{K}{\tau_1 - \tau_2} \left[e^{-\frac{t}{\tau_1}} - e^{-\frac{t}{\tau_2}} \right] \quad (I.11)$$

✓ **Cas de $\xi = 1$:**

Dans ce cas la transformée de Laplace de la sortie d'un système du deuxième ordre soumis à une entrée en impulsion est donnée par l'équation suivante :

$$Y(p) = \frac{K}{(1 + \tau_1 p)^2} = \frac{K w_0^2}{(p + w_0)^2} \quad (I.12)$$

Donc :

$$y(t) = K w_0^2 t e^{-\frac{t}{\tau}} \quad (I.13)$$

✓ **Cas de $\xi < 1$:**

Dans ce cas la transformée de Laplace de la sortie d'un système du deuxième ordre soumis à une entrée en impulsion est donnée par l'équation suivante :

$$Y(p) = \frac{K w_0^2}{p^2 + 2\xi w_0 p + w_0^2} \quad (I.14)$$

Ou bien encore, dans le domaine temporel :

$$y(t) = \frac{K w_0}{\sqrt{1 - \xi^2}} e^{-\xi w_0 \cdot t} \sin\left(\sqrt{1 - \xi^2} \cdot w_0 \cdot t\right) \quad (I.15)$$

$\xi = \text{zetaeta}$

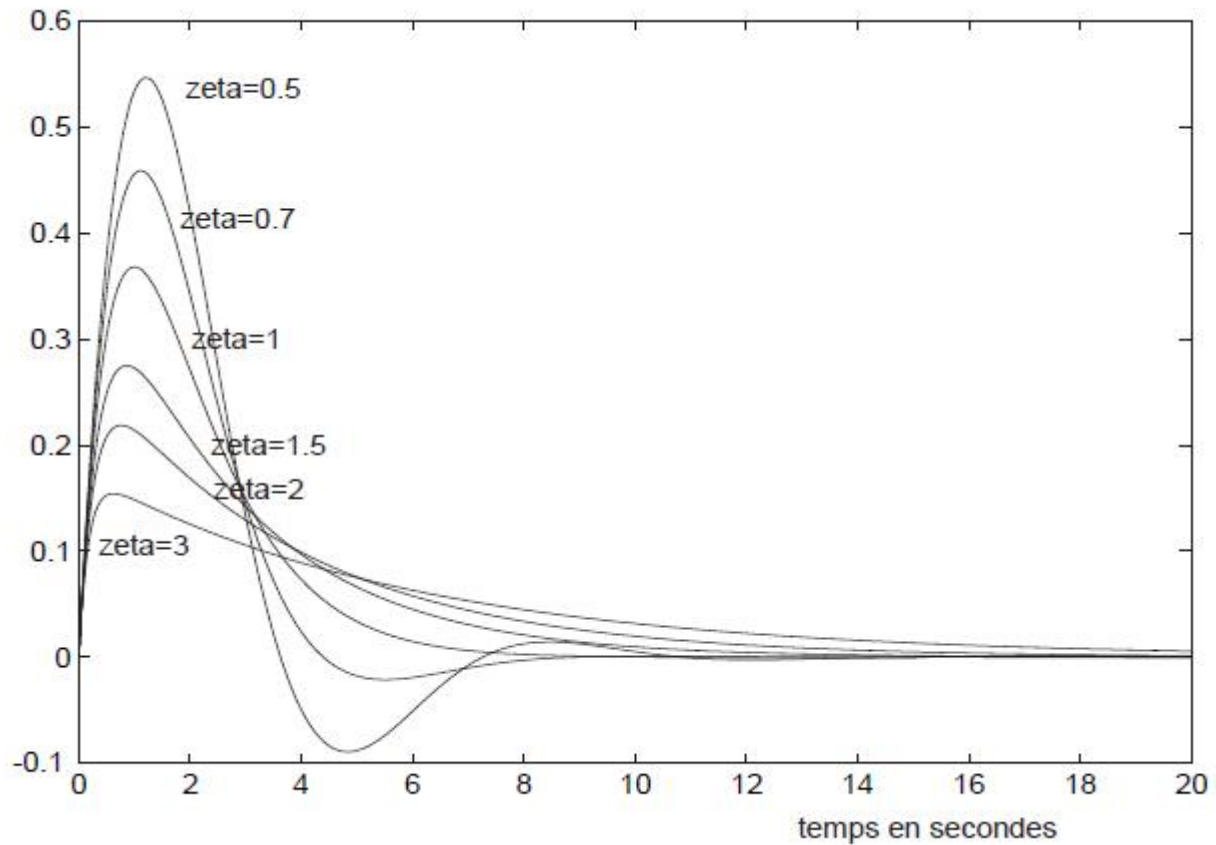


Figure I.6 : Réponse impulsionnelle en fonction de ξ pour w_0 fixé.

• Réponse indicielle

Il s'agit d'étudier la réponse du système à une entrée en échelon d'amplitude A . On peut donc écrire la transformée de Laplace de la sortie :

$$Y(p) = \frac{AKw_0^2}{p(p^2 + 2\xi w_0 p + w_0^2)} \quad (I.16)$$

✓ Cas de $\xi > 1$:

Dans ce cas, on obtient par transformation de Laplace inverse de $Y(p)$:

$$y(t) = KA \left(1 + \frac{(-\xi - \sqrt{\xi^2 - 1}) \cdot e^{\omega_0(-\xi + \sqrt{\xi^2 - 1}) \cdot t} - (-\xi + \sqrt{\xi^2 - 1}) \cdot e^{\omega_0(-\xi - \sqrt{\xi^2 - 1}) \cdot t}}{2\sqrt{\xi^2 - 1}} \right) \quad (I.17)$$

✓ Cas de $\xi = 1$:

Le signal temporel de sortie est donnée par :

$$y(t) = KA(1 - (1 + \omega_0 \cdot t) \cdot e^{-\omega_0 \cdot t}) \quad (I.18)$$

✓ Cas de $\xi < 1$:

$$y(t) = KA \left(1 - e^{-\xi \cdot \omega_0 \cdot t} \left(\cos(\sqrt{1 - \xi^2} \cdot \omega_0 \cdot t) + \frac{\xi}{\sqrt{1 - \xi^2}} \sin(\sqrt{1 - \xi^2} \cdot \omega_0 \cdot t) \right) \right) \quad (I.19)$$

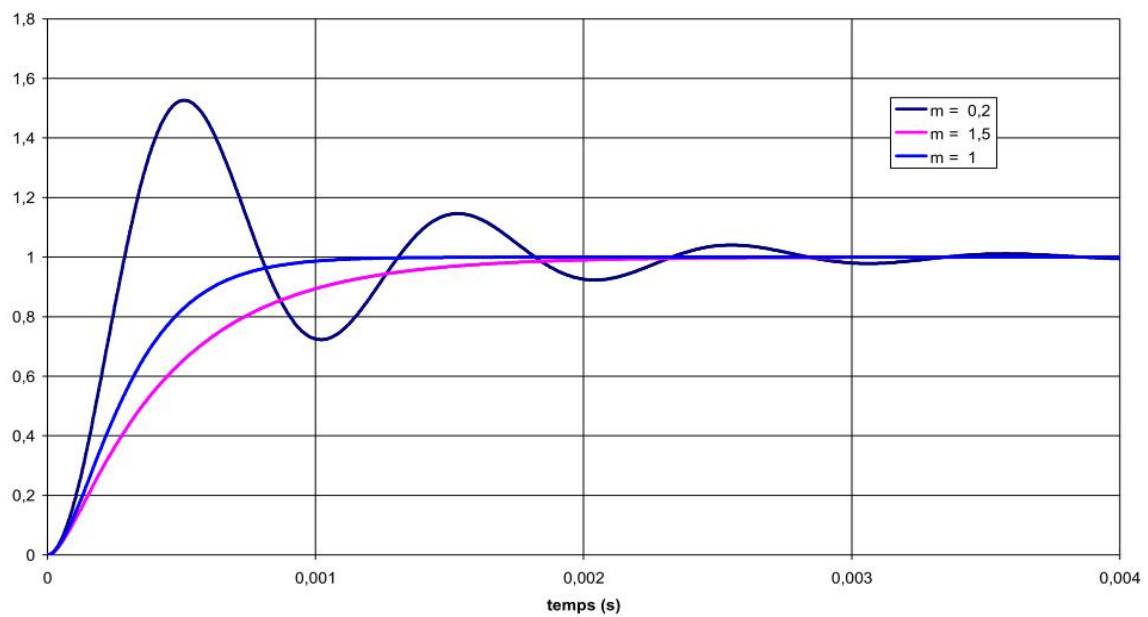


Figure I.7 : Réponse indicielle en fonction de ξ pour w_0 fixé.

I.2. Historique de la régulation

Le système souvent cité comme étant la première réalisation d'un correcteur automatique est la clepsydre de Ktesibios (-270 av. J.C) (figure 1.1). Afin d'améliorer le principe de l'horloge à eau, Ktesibios introduit un réservoir supplémentaire dans lequel le volume de liquide reste constant grâce à un mécanisme qui ferme l'entrée du réservoir lorsque celui-ci est trop plein.

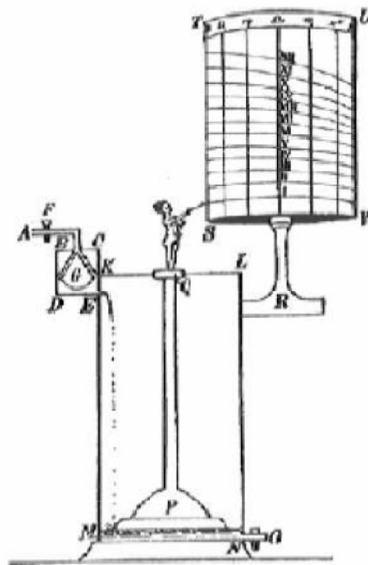


Figure 1.8 : Clepsydre de Ktesibios (-270 av. J.C., Alexandrie).

Les développements de l'horloge conduisent alors à la création d'automates extrêmement compliqués. Citons, à titre d'exemple, le canard de Vaucanson (1709-1782) qui pouvait boire, se nourrir, caqueter, nager, "digérer" et déféquer. L'idée de la programmation est née à peu près à la même époque. En 1728, Jean-Philippe Falcon crée le premier métier à tisser programmable par cartons perforés. Vaucanson réalise un métier à tisser programme en 1745, et en 1801 le célèbre métier à tisser automatique programmable de Joseph-Marie Jacquard.

A la même époque d'autres systèmes voient le jour, poussés par la révolution industrielle qui est en marche. Denis Papin développe la soupape de sécurité pour les systèmes fonctionnant à la vapeur et James Watt invente le régulateur de vitesse à boule.

Les réalisations précédentes ne manquent pas d'ingéniosité, néanmoins il faudra attendre l'arrivée de l'électricité puis de l'électronique pour voir apparaître les premiers correcteurs en tant que tels. Enfin, le développement des microprocesseurs puis celui des microcontrôleurs permet aujourd'hui de mettre en œuvre des principes de commande très élaborés. [3]

I.3. Caractéristiques d'un système asservi

- **Stabilité :**

Définition 1 : on dira qu'un système est stable si écarté de sa position d'équilibre, il y revient.

Dentition 2 : Un système est dit stable si sa réponse à toute entrée bornée est bornée.

- **Précision :**

C'est la capacité de l'asservissement à atteindre la consigne avec précision.

La précision statique (ou erreur de position) du système est caractérisée par l'erreur en régime permanent en réponse à un échelon :

$$\varepsilon_{\infty} = \lim_{t \rightarrow \infty} \varepsilon(t) \quad (I.20)$$

On parlera de précision dynamique dès que l'entrée du système évolue de manière continue dans le temps : par exemple on désigne par erreur de vitesse la valeur de l'erreur quand l'entrée du système est une rampe.

- **Temps de réponse :** c'est la vitesse à laquelle la valeur finale sera atteinte.

I.4. Principe de fonctionnement

Dans la plupart des appareils et installations industrielles, tertiaires et mêmes domestiques, il est nécessaire de maintenir les grandeurs physiques à des valeurs déterminées, en dépit des variations externes ou internes influant sur ces grandeurs. Le niveau d'un réservoir d'eau, la température d'une étuve, le débit d'une conduite de gaz, étant par nature variables, doivent donc être régulés par des actions convenables sur le processus considéré.

Si les perturbations influant sur la grandeur à contrôler sont lentes ou négligeables, un simple réglage (dit en boucle ouverte) permet d'obtenir et de maintenir la valeur demandée (par exemple : action sur un robinet d'eau), dans la majorité des cas, cependant, ce type de réglage n'est pas suffisant, parce que trop grossier ou instable. Il faut alors comparer, en permanence, la valeur mesurée de la grandeur réglée à celle que l'on souhaite obtenir et agir en conséquence sur la grandeur d'action, dite grandeur régulante, on a dans ce cas, constitué une boucle de régulation et plus généralement une boucle d'asservissement.

Cette boucle nécessite la mise en œuvre d'un ensemble de moyens de mesure, de traitement de signal ou de calcul, d'amplification et de commande d'actionneur, constituant une chaîne d'éléments associés à la chaîne de régulation (ou d'asservissement).[4]

Pour réguler un système physique, il faut :

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

On peut représenter une régulation de la manière suivante :

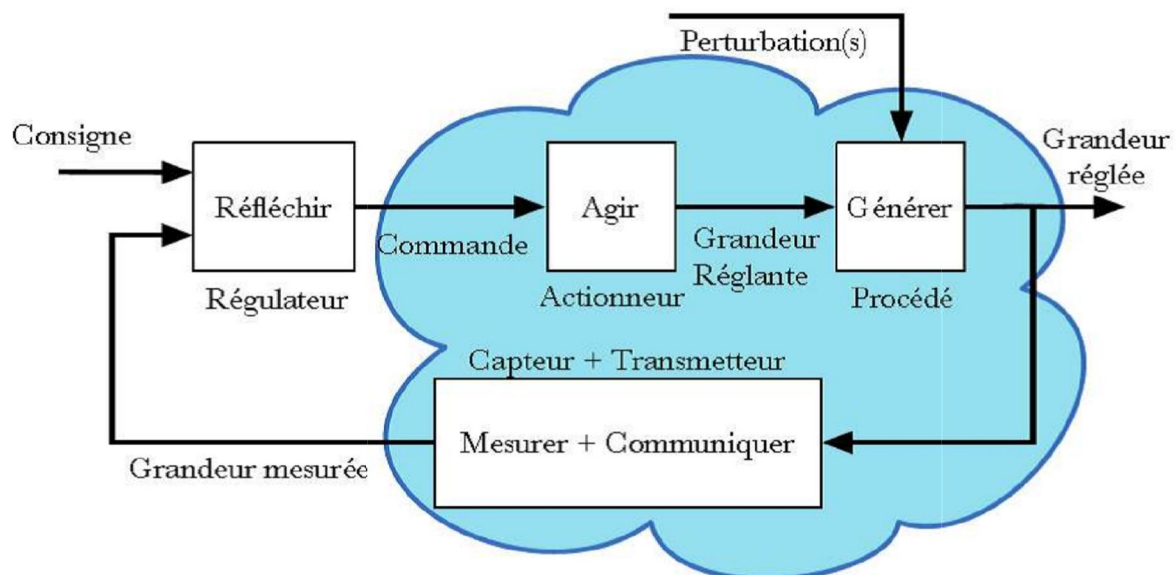


Figure I.9 : Système de régulation de manière générale

I.4.1. Régulation en boucle ouverte (Manuelle)

Dans un asservissement en boucle ouverte, l'organe de contrôle ne réagit pas à travers le processus sur la grandeur mesurée (celle-ci n'est pas contrôlée). Une régulation en boucle ouverte ne peut être mise en œuvre que si l'on connaît la loi régissant le fonctionnement du processus.

Contrairement à un asservissement en boucle fermée, un asservissement en boucle ouverte permet d'anticiper les phénomènes et d'obtenir des temps de réponse très courts. De plus, il n'y a pas d'oscillation à craindre (car il s'agit d'un système dynamiquement stable). Enfin,

l'asservissement en boucle ouverte est la seule solution envisageable lorsqu'il n'y a pas de contrôle final possible.

Au niveau des inconvénients, la régulation en boucle ouverte impose de connaître la loi régissant le fonctionnement du processus, et il est très fréquent que l'on ne connaisse pas la loi en question. Autre inconvénient sérieux, il n'y a aucun moyen de contrôler, à plus forte raison de compenser, les erreurs, les dérives, les accidents qui peuvent intervenir à l'intérieur de la boucle ; autrement dit, il n'y a pas de précision ni surtout de fidélité qui dépendent de la qualité intrinsèque des composants. Enfin, la régulation en boucle ouverte ne compense pas les facteurs perturbateurs.[5]

- **Fonctionnement en boucle ouverte :**

On parle de fonctionnement en boucle ouverte quand c'est l'opérateur qui contrôle l'organe de réglage.

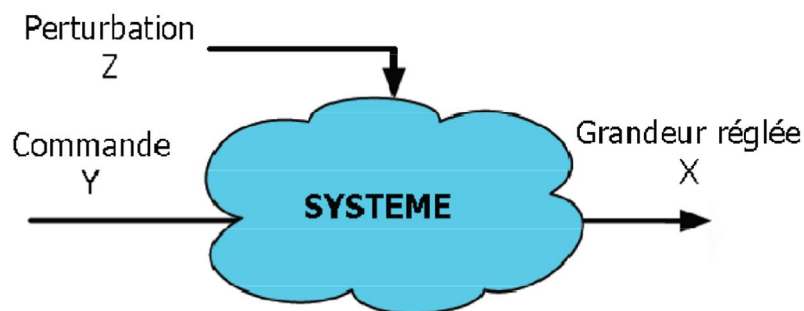


Figure I.10 : Régulation en boucle ouverte

I.4.2. La régulation en boucle fermée

Dans une régulation en boucle fermée, une bonne partie des facteurs perturbateurs sont automatiquement compensés par la contre-réaction à travers le procédé. Autre avantage, il n'est pas nécessaire de connaître avec précision les lois, le comportement des différents composants de la boucle, et notamment du processus, bien que la connaissance des allures statistiques et dynamiques des divers phénomènes rencontrés soit utile pour le choix des composants.

Parmi les inconvénients d'une régulation en boucle fermée, il faut citer le fait que la précision et la fidélité de la régulation dépend de la fidélité et de la précision sur les valeurs mesurées et sur la consigne.

Autre inconvénient, sans doute plus important, le comportement dynamique de la boucle dépend des caractéristiques des différents composants de la boucle, et notamment du processus, en fait un mauvais choix de certains composants peut amener la boucle à entrer en oscillation.

Enfin, la régulation en boucle fermée n'anticipe pas. Pour que la régulation envoie une commande à l'organe de contrôle, il faut que les perturbations ou les éventuelles variations de la valeur de consigne se manifestent sur la sortie du processus : ceci peut exiger un délai parfois gênant. [5]

• Fonctionnement en boucle fermée

C'est le fonctionnement normal d'une régulation. Le régulateur compare la mesure de la grandeur réglée et la consigne et agit en conséquence pour s'en rapprocher

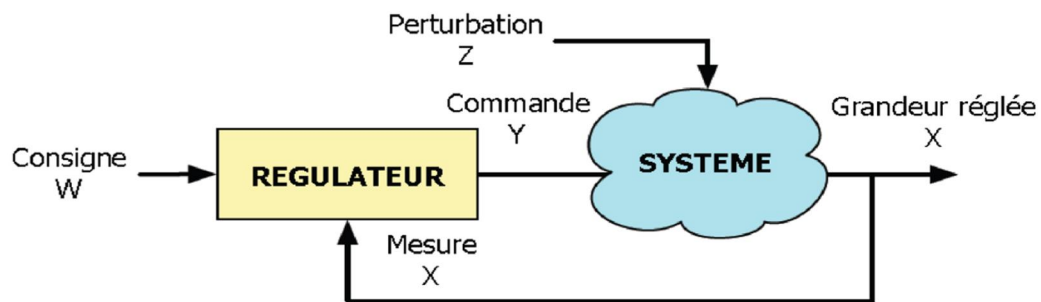


Figure I.11 : Régulation en boucle fermée.

I.5.Représentation fonctionnelle d'une boucle de régulation

D'une manière générale, une boucle de régulation peut être représentée de la manière suivante :

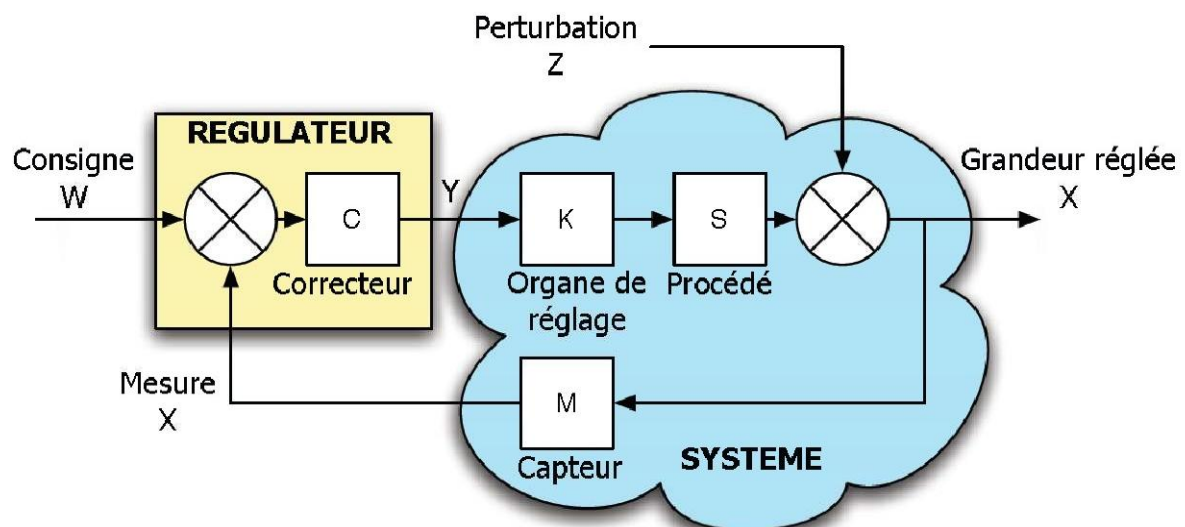


Figure I.12 : Représentation fonctionnelle d'une boucle de régulation.

- **Éléments constitutifs d'une boucle de régulation :**

Une boucle de régulation doit comporter au minimum les éléments suivants :

- un capteur de mesure ;
- un transmetteur souvent intégré au capteur ;
- un régulateur ;
- un actionneur (organe de réglage).

✓ Le **régulateur** reçoit deux informations :

- Le signal de mesure (**X**) provenant du capteur,
- la consigne (**W**) (qui peut être locale ou externe).

En fonction de l'écart entre ces deux valeurs et de l'algorithme de calcul pour lequel il a été configuré, il délivre un signal de sortie (**Y**) dirigé vers l'actionneur (Organe de réglage) afin d'annuler cet écart et de ramener la mesure vers la valeur de consigne.

- **Schéma fonctionnel d'un régulateur**

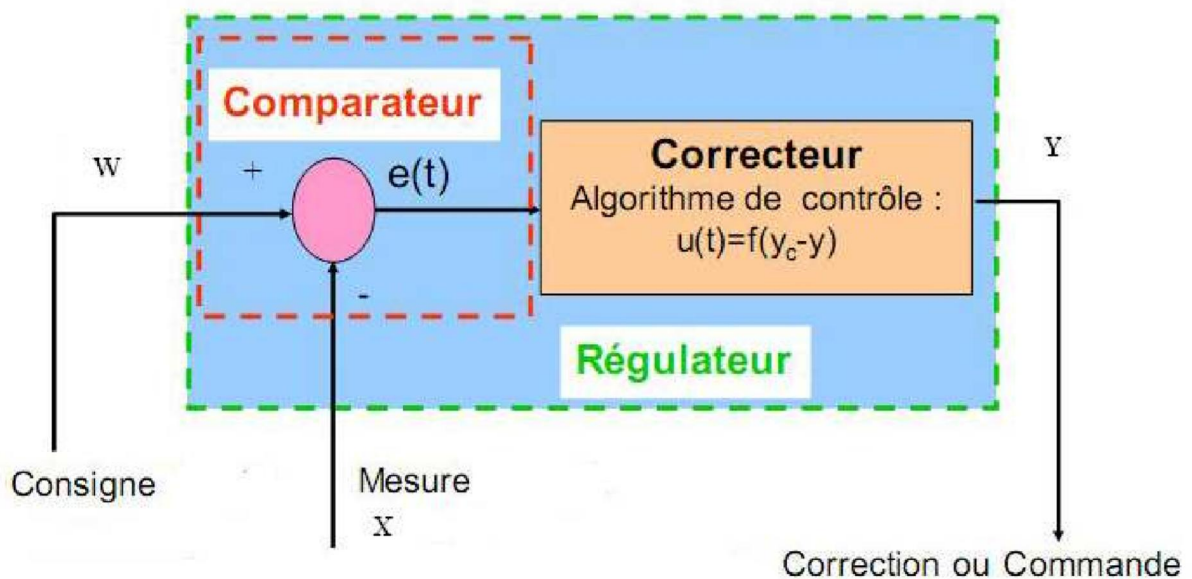


Figure I.13 : schéma fonctionnel d'un régulateur.

I.6. Classification des régulateurs

Les régulateurs sont classés selon la nature de l'énergie qu'ils utilisent, le sens d'action et selon le type d'action :

- **Pneumatique** : sortie 0.2 à 1 bar. Ils sont utilisés dans l'industrie chimique du gaz, ne présentent pas de danger d'explosion, de moins au moins utilisés car lents et encombrants.

Les régulateurs pneumatiques, toujours présents sur le marché, ont un aspect similaire et avec des possibilités identiques à celles des régulateurs électroniques.

Les signaux d'entrée et de sortie, sont des pressions d'air modulées, dont l'échelle est généralement 0.2 à 1 bar. La consigne externe est souvent en option.

- **Electrique** : sortie 4-20 mA utilisent des signaux analogiques à base d'amplificateurs opérationnels.

Ces régulateurs utilisent une électronique analogique, à base d'amplificateurs opérationnels, ils cèdent le pas à la technologie numérique, mais sont encore nombreux dans l'industrie.

- **Electronique (numérique)** : Sortie sous forme numérique. La technologie numérique permet d'avoir une grande souplesse : opération arithmétique, auto ajustage des coefficients, possibilité d'émettre ou de recevoir des données.

La différence fondamentale dans la présentation de ces régulateurs est un clavier opérateur, intégré ou indépendant (micro console), permettant d'émettre ou de recevoir des données.

La technologie numérique permet avec une grande souplesse, une extension des possibilités du régulateur :

- Traitement du signal d'entrée (extraction de racine carré, filtrage, linéarisation,...)
- Mise à l'échelle (valeur et format) des indicateurs.
- Choix du type d'alarme, soit sur la mesure, soit sur l'écart,
- Affichage précis des données telles qu'actions, limites, ...
- Choix du mode de dérivée, soit sur la mesure soit sur l'écart,
- .

La technologie numérique permet, dans certains cas, au régulateur de calculer lui-même, les actions, on distingue :

- Les régulateurs auto réglant qui calculent les actions PID à un point de fonctionnement donné, et ce à partir d'une intervention humaine.
- Les régulateurs auto-adaptatifs qui calculent et ajustent en permanence les paramètres de leur algorithme (PID ou autre) en fonction de l'évolution du procédé.

- **Le sens d'action direct** - l'augmentation de la mesure provoque l'augmentation de la commande.
- **Le sens d'action inverse** - l'augmentation de la mesure provoque la diminution de la commande.

- Régulateur proportionnel (P).
- Régulateur proportionnel intégrale (PI).
- Régulateur proportionnel dérivée (PD).
- Régulateur proportionnel intégrale dérivée (PID).
- Tout ou rien.

I.7. Les régulateurs à action P, I, D

I.7.1. Les actions P, I, D

En pratique, à une catégorie donnée de systèmes à asservir correspond un type de correcteur adopté. Pour effectuer un choix judicieux, il faut connaître les effets des différentes actions : Proportionnelle, Intégrale et Dérivée.

- **L'action proportionnelle**

La sortie $u(t)$ du régulateur proportionnel est donnée en fonction de son entrée $\varepsilon(t)$ qui représente l'écart entre la consigne et la mesure par la relation :

$$u(t) = K_p \cdot \varepsilon(t) \quad (I.21)$$

Pour le cas discret, cette relation reste la même telle que :

$$u(k) = K_p \cdot \varepsilon(k) \quad (I.22)$$

Le rôle de l'action proportionnelle est de minimiser l'écart $\mathcal{E}$ entre la consigne et la mesure et elle réduit le temps de monter et le temps de réponse.

On constate qu'une augmentation du gain K_p du régulateur entraîne une diminution de l'erreur statique et permet d'accélérer le comportement global de la boucle fermée. On serait tenté de prendre des valeurs de gain élevées pour accélérer la réponse du procédé mais on est limité par la stabilité de la boucle fermée. En effet, une valeur trop élevée du gain augmente l'instabilité du système et donne lieu à des oscillations.

• L'action intégrale

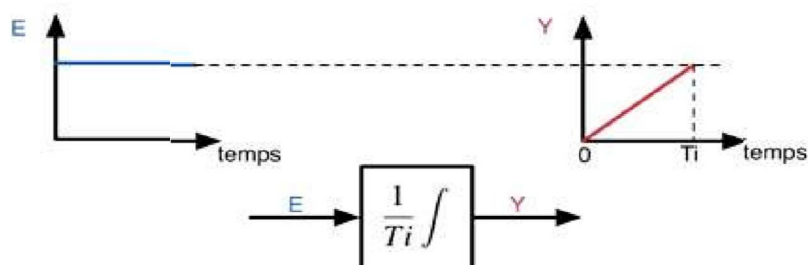
L'action intégrale agit proportionnellement à la surface de l'écart entre la consigne et la mesure, et elle poursuit son action tant que cet écart n'est pas nul. On dit que l'action intégrale donne la précision statique (Elle annule l'erreur statique). L'action intégrale est conditionnée par le temps d'intégrale T_i

$$u(t) = \frac{1}{T_i} \int_0^t \mathcal{E}(t) \cdot dt \quad (I.23)$$

On obtient en discret

$$u_i[k] = \frac{T_e}{T_i} \cdot \sum_{k=0}^n e[k] \quad (I.24)$$

❖ **Fonctionnement** : Pour étudier l'influence de l'action intégrale, on s'intéressera à la réponse du module intégral à un échelon. Plus K_i est grand (T_i petit), plus la valeur de la sortie Y augmente rapidement. Le temps T_i est le temps pour que la commande Y augmente de la valeur



de l'entrée $E=W-X$.

Figure I.14 : Influence de T_i sur la commande.

Comme dans le cas de l'action proportionnelle, un dosage trop important de l'action intégrale engendre une instabilité de la boucle de régulation. Pour son réglage, il faut la aussi trouver un compromis entre la stabilité et la rapidité.

L'ajout du terme intégral permet d'améliorer la précision mais en contrepartie, il introduit malheureusement un déphasage de $(-\frac{\pi}{2})$ ce qui risque de rendre le système instable du fait de la diminution de la marge de phase.

L'action intégrale est utilisée lorsqu'on désire avoir en régime permanent, une précision parfaite, en outre, elle permet de filtrer la variable à régler.[6]

• L'action dérivée

Elle est une action qui tient compte de la vitesse de variation de l'écart entre la consigne et la mesure, c'est une action qui amplifie les variations brusques de la consigne, elle joue aussi un rôle stabilisateur, contrairement à l'action intégrale. En effet, elle délivre une sortie variant proportionnellement à la vitesse de variation de l'écart ε :

$$u(t) = T_e \frac{d\varepsilon(t)}{dt} \quad (I.25)$$

❖ Fonctionnement :

Pour étudier l'influence de l'action dérivée, on s'intéressera à la réponse du module dérivé à une rampe. Plus **Td** est grand, plus la valeur de la sortie **Y** sera importante.

Le temps **Td** est le temps pour que l'entrée E augmente de la valeur de la sortie **Y**.

Avec T_d le dosage de l'action dérivée, exprime en minutes ou en secondes.

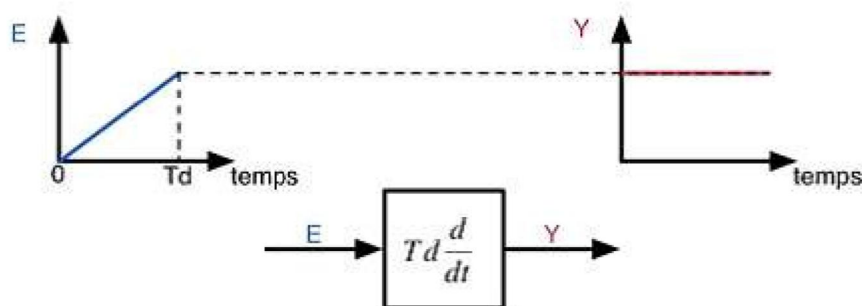


Figure I.15 : Influence de T_d sur la commande.

L'action dérivée va ainsi intervenir uniquement sur la variation de l'erreur ce qui augmente la rapidité du système (diminution des temps de repos). L'action dérivée permet aussi d'augmenter la stabilité du système par apport de phase $\left(+\frac{\pi}{2}\right)$ ce qui augmente la marge de phase). L'annulation de cette action en régime statique impose donc de ne jamais l'utiliser seule, l'action dérivée n'exerce qu'un complément à l'action proportionnelle. [6]

En pratique, il est souhaitable de limiter l'action dérivée afin de ne pas amplifier les bruits haute fréquence et de limiter l'amplitude des impulsions dues aux discontinuités de l'écart. Lorsque la période d'échantillonnage T_e est petite, la différence vers l'arrière (Approximation d'Euler rétrograde) nous permet d'approcher la dérivée d'un signal à temps continu par :

$$\frac{df(t)}{dt} = \frac{f(t) - f(t - T_e)}{T_e} \quad (1.26)$$

L'opération de dérivation se traduisait par une multiplication par la variable de

Laplace p en continu. Dans le cas discret, en appliquant la transformée en z on obtient :

$$Z\left\{\frac{f(t) - f(t - T_e)}{T_e}\right\} = \frac{1 - z^{-1}}{T_e} \quad (1.27)$$

Ceci conduit à établir l'équivalence linéaire entre la variable de Laplace p et la variable z

$$p \longleftrightarrow \frac{1}{T_e} (1 - z^{-1}) \quad (1.28)$$

Donc en discret, le terme dérivé peut être remplacé par :

$$u(k) = \frac{T_d}{T_e} \cdot (\varepsilon(k) - \varepsilon(k - 1)) \quad (1.29)$$

- **Résumé des actions des corrections P, I et D**

Action	Rôle et domaine d'utilisation
P	L'action Proportionnelle corrige de manière instantanée, donc rapide, tout écart de la grandeur à régler, elle permet de vaincre les grandes inerties du système. Afin de diminuer l'écart de réglage et rendre le système plus rapide, on augmente le gain (on diminue la bande proportionnelle) mais, on est limité par la stabilité du système. Le régulateur P est utilisé lorsqu'on désire régler un paramètre dont la précision n'est pas importante, exemple : régler le niveau dans un bac de stockage
I	L'action intégrale complète l'action proportionnelle. Elle permet d'éliminer l'erreur résiduelle en régime permanent. Afin de rendre le système plus dynamique (diminuer le temps de réponse), on diminue l'action intégrale mais, ceci peut provoquer l'instabilité en état fermé. L'action intégrale est utilisée lorsqu'on désire avoir en régime permanent, une précision parfaite, en outre, elle permet de filtrer la variable à régler d'où l'utilité pour le réglage des variables bruitées telles que la pression.
D	L'action Dérivée, en compensant les inerties dues au temps mort, accélère la réponse du système et améliore la stabilité de la boucle, en permettant notamment un amortissement rapide des oscillations dues à l'apparition d'une perturbation ou à une variation subite de la consigne. Dans la pratique, l'action dérivée est appliquée aux variations de la grandeur à régler seule et non de l'écart mesure-consigne afin d'éviter les à-coups dus à une variation subite de la consigne. L'action D est utilisée dans l'industrie pour le réglage des variables lentes telles que la température, elle n'est pas recommandée pour le réglage d'une variable bruitée ou trop dynamique (la pression). En dérivant un bruit, son amplitude risque de devenir plus importante que celle du signal utile.

Tableau I.2 : résumé des actions des corrections P, I et D.

I.7.2. Le régulateur PI

Le correcteur intégral est en général associé au correcteur proportionnel, il élabore alors une commande qui peut être donnée par la relation suivante :

$$u(t) = Kp(\varepsilon(t) + \frac{1}{Ti} \int_0^t \varepsilon(t) \cdot dt \quad (I.30)$$

La fonction de transfert du correcteur est alors donnée par :

$$C(p) = Kp \cdot \frac{1+T_i p}{T_i p} \quad (I.31)$$

Pour un régulateur intégral pur, le régime dynamique est relativement long. D'un autre côté, le régulateur proportionnel réagit immédiatement aux écarts de réglage mais il n'est pas en mesure de supprimer totalement l'erreur statique.

La combinaison des actions proportionnelle et intégrale permet d'associer l'avantage de l'action Proportionnel (c'est-à-dire la réaction rapide à un écart de réglage) à l'avantage de l'action Intégrale qui est la compensation exacte de la grandeur pilotée.

La transposition de correcteurs continus consiste à discrétiser un correcteur continu afin de l'utiliser dans une commande numérique.

En utilisant cette équivalence on obtient le correcteur PI discret :

$$C(z) = \frac{U(z)}{\varepsilon(z)} = \frac{(Kp+Ki) - Kp \cdot z^{-1}}{(1 - z^{-1})} \quad (I.32)$$

D'où on obtient l'algorithme de commande du régulateur PI :

$$u(k) = u(k-1) + (Kp + Ki) \cdot \varepsilon(k) - Kp \varepsilon(k-1) \quad (I.33)$$

I.7.3. Le régulateur PID

Le régulateur standard le plus utilisé dans l'industrie est le régulateur PID (proportionnel intégral dérive), car il permet de régler à l'aide de ses trois paramètres les performances (amortissement, temps de réponse, ...) d'un processus modélisé par un deuxième ordre.

Nombreux sont les systèmes physiques qui, même en étant complexes, ont un comportement voisin de celui d'un deuxième ordre. Par conséquent, le régulateur PID est bien adapté à la plupart des processus de type industriel et est relativement robuste par rapport aux variations des paramètres du procédé, quand on n'est pas trop exigeant sur les performances de la boucle fermée par rapport à celles de la boucle ouverte (par exemple, accélération très importante de la réponse ou augmentation très importante de l'amortissement en boucle fermée).

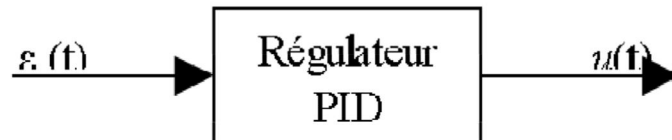


Figure I.16 : Schéma synoptique d'un régulateur PID.

Un régulateur PID est obtenu par l'association de ces trois actions et il remplit essentiellement les trois fonctions suivantes :

- Il fournit un signal de commande en tenant compte de l'évolution du signal de sortie par rapport à la consigne.
- Il élimine l'erreur statique grâce au terme intégrateur.
- Il anticipe les variations de la sortie grâce au terme dérivateur.

La commande $U(t)$ donnée par le régulateur PID, dans sa forme Classique est décrite par :

$$U(t) = Kp[\mathcal{E}(t) + \frac{1}{T_i} \int_0^t \mathcal{E}(t) \cdot dt + T_d \frac{d\mathcal{E}(t)}{dt}] \quad (I.34)$$

Elle est la somme de trois termes :

- le terme proportionnel

$$p = Kp \cdot \mathcal{E}(t) \quad (I.35)$$

- le terme intégral

$$I = Kp \frac{1}{T_i} \int_0^t \mathcal{E}(t) \cdot dt \quad (I.36)$$

- le terme dérivatif

$$D = Kp \cdot T_d \cdot \frac{d\mathcal{E}(t)}{dt} \quad (I.37)$$

Les paramètres du régulateur PID sont le gain proportionnel K_p , le temps intégral T_i et le temps dérivatif T_d , les temps étant exprimés en secondes.

➤ Structures des régulateurs PID

Les trois corrections, proportionnelle, intégrale et dérivée, permettent de définir trois structures de régulateur différentes.

	$Y(t) = K_p \times E + \frac{1}{T_i} \int_0^t E \times dt + T_d \times \frac{dE}{dt}$
	$Y(t) = K_p \times \left\{ E + \frac{1}{T_i} \int_0^t E \times dt + T_d \times \frac{dE}{dt} \right\}$
	$Y(t) = K_p \times \left\{ \left(1 + \frac{T_d}{T_i}\right) E + \frac{1}{T_i} \int_0^t E \times dt + T_d \times \frac{dE}{dt} \right\}$

Tableau I.3 : Structures des régulateurs PID.

➤ P, PI, ou PID ?

Le choix est toujours un peu ambigu. Le principe de choix communément admis, qui s'applique d'ailleurs à tous les choix de régulateur et correcteurs, est le principe du minimum : si un correcteur P donne les résultats escomptés, on ne met pas un PI, moins encore un PID ! Les tableaux suivants donnent un critère de choix en fonction de la réponse temporelle du système à un échelon d'entrée.

❖ Essai indiciel :

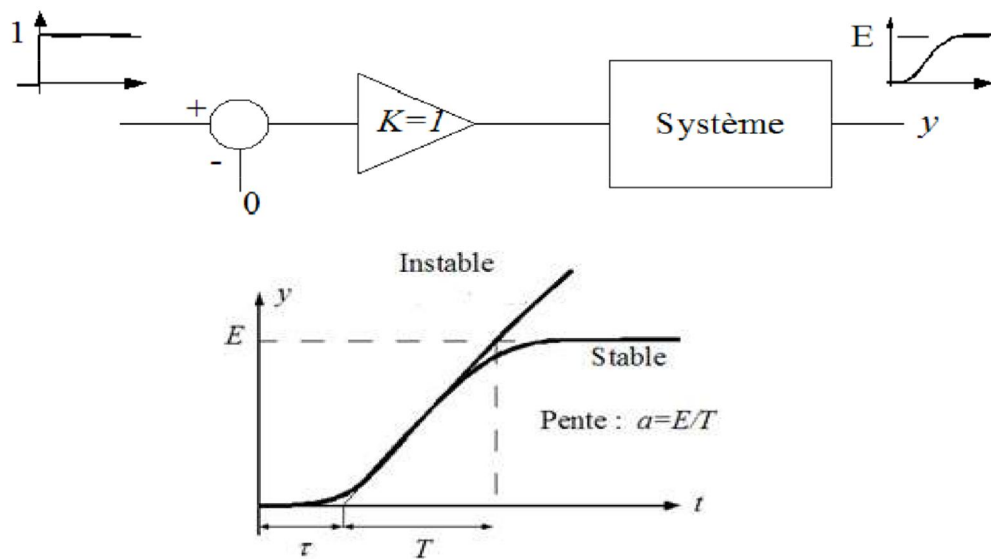


Figure I.17 : Essai indiciel.

Réglabilité T/τ	> 20	10 à 20	5 à 10	2 à 5	< 2
Régulateur	Tout ou rien	P	PI	PID	limite du PID ¹

Tableau I.4 : Choix d'une méthode de correction en fonction de l'indice de réglabilité
(Cas des systèmes stables)

Réglabilité $a\tau$	$0.05 < a\tau$	$0.05 < a\tau < 0.1$	$0.1 < a\tau < 0.2$	$0.2 < a\tau < 0.5$	$a\tau > 0.5$
Régulateur	Tout ou rien	P	PI	PID	limite du PID ¹

Tableau I.5 : Choix d'une méthode de correction en fonction de l'indice de réglabilité
(Cas des systèmes instables)

I.8. Le PID numérique

Pour la mise en œuvre des régulateurs, on peut se contenter de remplacer l'équation différentielle du régulateur PID par une équation aux différences prenant en compte le temps d'échantillonnage T_e .

➤ **Terme intégrale :**

L'intégrale du terme I est remplacée par la somme des écarts et la différentielle dt , par la période d'échantillonnage T_e . On a donc :

$$u_i[n] = \frac{T_e}{T_i} \cdot \sum_{k=0}^n e[k] = u_i[n-1] + \frac{T_e}{T_i} \cdot e[n] \quad (I.38)$$

➤ **Terme dérivé :**

Pour trouver l'équivalent numérique du terme dérivé, il faut partir de l'équation différentielle représentant la fonction de transfert $G_d(s)$:

$$G_d(s) = \frac{U_d(s)}{E_d(s)} = \frac{s \cdot T_d}{1 + s \cdot T_d / N} \quad (I.39)$$

En effectuant les produits croisés, on obtient :

$$U_d(s) \cdot \left(1 + s \cdot \frac{T_d}{N}\right) = E_d(s) \cdot s T_d \quad (I.40)$$

A cette expression correspond l'équation différentielle suivante :

$$U_d(t) = T_d \cdot \frac{de_d(t)}{dt} - \frac{T_d}{N} \cdot \frac{du_d(t)}{dt} \quad (I.41)$$

Dont l'équivalent numérique est :

$$U_d[n] = T_d \frac{e_d[n] - e_d[n-1]}{T_e} - \frac{T_d}{N} \cdot \frac{u_d[n] - u_d[n-1]}{T_e} \quad (I.42)$$

En regroupant les termes, il vient :

$$U_d[n] \cdot \left(1 + \frac{T_d}{NT_e}\right) = \frac{T_d}{T_e} (e_d[n] - e_d[n-1]) + \frac{T_d}{NT_e} \cdot u_d[n-1] \quad (I.43)$$

On en déduit alors l'expression permettant de calculer le terme dérivé :

$$U_d[n] = \left(\frac{NT_d}{T_d + NT_e}\right) (e_d[n] - e_d[n-1]) + \left(\frac{NT_d}{T_d + NT_e}\right) u_d[n-1] \quad (I.44)$$

➤ **Signal de commande (structure PID mixte).**

Le signal de commande est la somme des 3 composantes multipliée par K_p :

$$u[n] = K_p(e_p[n] + u_i[n] + u_d[n]) \quad (I.45)$$

Conclusion

Dans ce chapitre nous avons abordé quelques notions et caractéristiques des systèmes de premier et deuxième ordre, ainsi quelques définitions et principes qui sont les bases de la régulation des systèmes, nous avons mis en évidence les différents types de régulations principalement la régulation PID et cela tout en mettant en relief les différentes actions et structures du régulateur PID.

Chapitre II

Réalisation d'une boucle de régulation PID à base d'un microcontrôleur

Introduction

Dans ce chapitre nous allons entamer la réalisation dans notre carte toute en expliquant la partie hardware et software qui compose notre régulateur.

II.1. Description de l'application

L'objectif de ce projet est de concevoir un module effectuant les fonctions de corrections numériques P.I.D. d'un signal d'entrée analogique qui génère un signal de commande vers un système dont le modèle est connu (fonctions de transfert), cela nous impose d'utiliser certains composants. Ils ont été choisis pour leurs caractéristiques techniques et les performances nécessaires à la bonne exécution du système.

Les microcontrôleurs sont le cœur du système. Le premier doit effectuer en permanence le calcul de la correction P.I.D. et le deuxième doit affecter les paramètres du modèle, mais ils doivent aussi gérer l'envoi de données à l'afficheur LCD pour un affichage cohérent des paramètres.

Le PIC doit également prendre en compte les interruptions engendrées par les modifications possibles des paramètres à l'aide des boutons-poussoirs. Il est de plus équipé d'un convertisseur analogique-numérique (CAN) 10 bits en entrée.

En sortie de chacun des deux Pics nous avons choisi d'utiliser un convertisseur numérique analogique (CNA) 8 bits de référence DAC0808.

L'utilisation de ce PIC et du CAN impose quelques contraintes. Il ne peut recevoir en entrée qu'un signal compris entre 0 et 5 Volts.

II.2. Conception du circuit

Le schéma électrique générale obtenue à l'issu du montage des différents circuits Constituants la carte est représentée dans l'annexe 2.

II.2.1. Première démarche : le choix du microcontrôleur

Le choix d'un microcontrôleur est fait sur des critères bien défini comme ses performances, sa taille, la facilité de son utilisation, le prix du montage, le nombre d'entrées et sorties, les registres, sa fréquence...etc.

II.2.1.1. Le pic 16f877A caractéristique et organisation mémoire

II.2.1.1.1. Les principales caractéristiques du PIC 16F877A

Le PIC 16F877A est caractérisé par :

- Une Fréquence de fonctionnement élevée, jusqu'à 20 MHz Une mémoire vive de 368 octets.
- Une mémoire EEPROM pour sauver des paramètres de 256 octets
- Une mémoire de masse de type FLASH de 8K mots (1mot = 14 bits), elle est réinscriptible à volonté
- Chien de garde WDT
- 33 Entrées et sorties
- Chaque sortie peut sortir un courant maximum de 25 mA
- 03 Temporisateurs : TIMER0 (8 bits avec pré diviseur), TIMER1 (16 bits avec prédiviseur avec possibilité d'utiliser une horloge externe réseau RC ou QUARTZ) et TIMER2 (8 bits avec prédiviseur et post diviseur)
- 02 entrées de captures et de comparaison avec PWM (Modulation de largeur d'impulsions)
- Convertisseur analogique numérique 10 bits avec 8 entrées multiplexées maximum
- Une interface de communication série asynchrone et synchrone (USART/SCI)
- Une interface de communication série synchrone (SSP/SPI et I2 C)
- Une tension d'alimentation entre 2 et 5.5 V
- Jeux de 35 instructions [8]

II.2.1.1.2. Schéma fonctionnel du 16f877A (Voir l'annexe 1)

II.2.1.1.3. Organisation interne du PIC 16F877A [9]

Les mémoires de 16f877A :

La mémoire du 16f877a est divisée en 3 parties :

La mémoire programme :

La mémoire programme est constituée de 8Kbits. C'est dans cette zone que nous allons écrire notre programme.

La mémoire EEPROM :

La mémoire EEPROM (Electrical Erasable Programmable Read Only Memory), est constituée de 256 octets que nous pouvons lire et écrire depuis notre programme. Ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres

La mémoire RAM :

C'est de la mémoire d'accès aléatoire, mais labile (c'est-à-dire qu'elle s'efface lorsqu'elle n'est plus sous tension) ; cette mémoire contient les registres de configuration du PIC ainsi que les différents registres de données. Elle contient également les variables utilisées par le programme.

Cette mémoire RAM disponible sur le 16F877A est de 368 octets.

Watchdog :

Cette fonction est capable de surveiller le bon fonctionnement du programme que le microcontrôleur exécute.

Le TIMER :

Un timer est un registre interne au microcontrôleur, celui-ci s'incrémente au grès d'une horloge, ce registre peut servir par exemple pour réaliser des temporisations, ou bien encore pour faire du comptage (par l'intermédiaire d'une broche spécifique : RA4/TOKI). Le PIC 16F877A possède 3 timers.

II.2.1.1.4. Organisation externe du PIC 16F877A**•Brochage de 16f877A :**

Le boîtier du PIC 16F877A décrit par la figure II-4 comprend 40 pins : 33 pins d'entrées/sorties, 4 pins pour l'alimentation, 2 pins pour l'oscillateur et un pin pour le reset (MCLR) qui sert à initialiser le μ C en cas de la mise sous tension, de remise à zéro externe, de chien de garde et en cas de la baisse de tension d'alimentation et cela bien sur on le raccordant à un circuit externe bien définie.

Les broches VDD et Vss servent à alimenter le PIC, on remarque qu'on a 2 connections « VSS » et 2 connections « VDD ».

La présence de ces 2 pins s'explique pour une raison de dissipation thermique. Les courants véhiculés dans le pic sont loin d'être négligeables du fait des nombreuses entrées/sorties disponibles, Le constructeur a donc décidé de répartir les courants en plaçant 2 pins pour l'alimentation VSS, bien évidemment, pour les mêmes raisons, ces pins sont situées de part et d'autre du PIC, et en positions relativement centrales. [10]

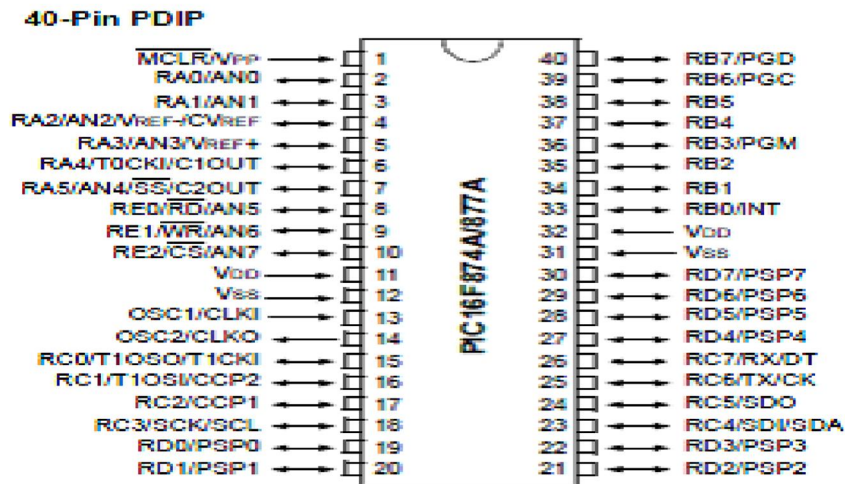


Figure II.1 : Brochage du 16f877A

Les µc 16F877A 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

Particularité du port A :

Le port A est formé de six pins donc six entrées/sorties numérotées de RA0 à RA5 qui peuvent être utilisé comme des entrées pour le convertisseur analogique numérique.

On peut utiliser ce port, soit pour la conversion analogique /numérique, soit en mode (I/O), dans notre projet on a utilisé RA0 et RA1 comme entrée pour le CAN.

Particularité du port B :

Il possède 8 pins d'entrée/sortie classique numérotées de RB0 à RB7.

Particularité du port C :

C'est un PORT comportant 8 pins de RC0 à RC7, dont on peut les utilisés comme des entrée /sortie numérique ou les utilisant pour des opérations bien définie selon la configuration de chaque pin exemple les pins RC7/RX/DT et RC6/TX/CK on les utilise pour la liaison série.

Particularité du port D :

C'est un PORT comportant 8 pins de RD0 à RD7, dont on peut les utilisé comme des entrée /sortie numérique.

Particularité du port E :

Le port E possède trois pins donc trois entrées/sorties, RE0 à RE2, il est utilisé comme entrées au convertisseur analogique numérique ou numérique selon sa configuration faites lors de sa programmation.

L'oscillateur :

Les broches OSC1 et OSC2 ou CLKIN et CLOUT permettent de faire fonctionner l'oscillateur interne du PIC qui peut être un quartz, un résonateur céramique, un oscillateur externe ou un réseau RC, dont le rôle est de créer des impulsions de fréquences élevées. La fréquence maximale d'utilisation va dépendre du microcontrôleur utilisé. Le suffixe indiqué sur le boîtier donne la nature de l'horloge à utiliser et sa fréquence maximale. Notre PIC utilisé est limité à 20Mhz.

MCLR :

La broche MCLR permet de réaliser un Reset du circuit quand elle est placée à 0V

II.2.1.2. Le convertisseur analogique numérique du PIC 16F877A

La fonction conversion analogique-numérique consiste à transformer une grandeur électrique en une grandeur numérique exprimée sur N bits. Cette grandeur de sortie représente, dans le système de codage qui lui est affecté, un nombre proportionnel à la grandeur analogique d'entrée.

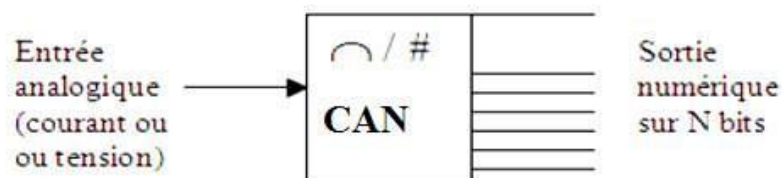


Figure II.2 : Principe de conversion Analogique/Numérique

II.2.1.2.1. Résolution d'un convertisseur analogique-numérique

La résolution d'un C.A.N. est la valeur de la variation de la tension d'entrée qui provoque un changement d'1 LSB sur le nombre N en sortie. C'est donc la largeur d'un palier de la caractéristique de transfert. Plus la résolution est petite, plus la conversion est précise.

D'une façon générale, la résolution est égale à la plage de tension de la mesure divisée par le nombre de bits du CAN :

$$\text{Résolution} = \frac{\text{plage de mesure}}{\text{nombre de bits du CAN}}$$

Par conséquent, pour augmenter la résolution d'un système, on a 2 possibilités :

- soit augmenter le nombre bits du CAN utilisé
- soit diminuer la plage de mesure

Par exemple : Le CAN intégré dans le 16F877A est un CAN 10 bits qui donne une précision en 5V de 5mv environ, ce qui est une précision tout à fait intéressante. La résolution 10 bits du PIC permet d'attribuer 1024 valeurs numérique à notre signal d'entrée.

II.2.1.2.2. L'organigramme d'acquisition de la mesure

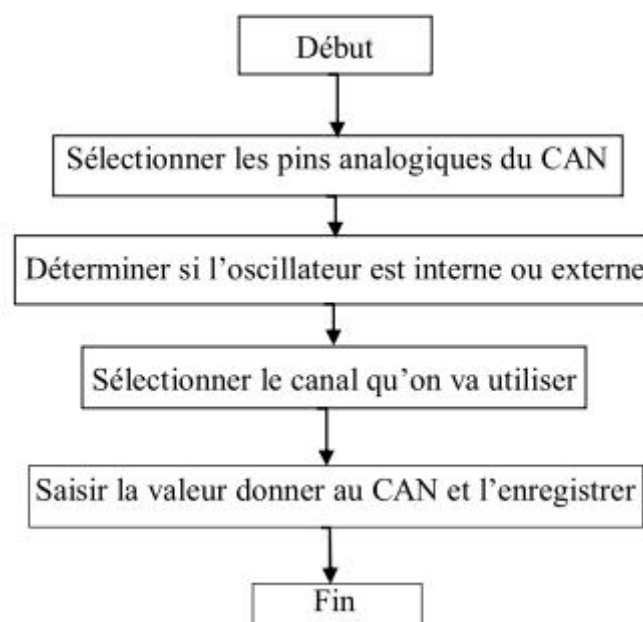


Figure II.3 : L'organigramme d'acquisition de la mesure

Cet organigramme représente les étapes d'acquisition d'un signal analogique appliqué sur les pins du CAN.

II.2.1.2.3. Échantillonnage de l'information

La première étape du processus de numérisation est l'échantillonnage. Cela consiste à passer d'un signal à temps continu (un signal électrique en l'occurrence), en une suite discrète de valeurs (valeurs mesurées à intervalles réguliers). Cet échantillonnage est de l'ordre de la dizaine de microsecondes. Une relation existe entre la fréquence maximale du signal d'entrée et la fréquence échantillonnage. En effet, pour éviter le phénomène de recouvrement spectral il faut respecter le théorème de Shannon : $f_e > 2.f_{max}$.

II.2.1.2.4. Détail de fonctionnement du CAN contenu dans le 16F877A

Le microcontrôleur 16F877A intègre un module de conversion analogique numérique qui est associé à un multiplexeur analogique multivoie des entrées. Autrement dit on peut réaliser des mesures sur plusieurs voies d'entrée analogiques (jusqu'à 5). Ces voies analogiques correspondent aux broches du port A.

Le module CAN 10 bits implique que le résultat de la conversion est un nombre de 10 bits, c'est-à-dire compris entre 000H et 3FFH (11 1111 1111 en binaire ou 1024 en décimal).

Ceci correspond donc à 1024 niveaux de mesure, ce qui pour 5V donne une résolution de 4,8mV, précision largement suffisante.

Le temps de conversion de ce module est de l'ordre de : 40µs ce qui permet en pratique de réaliser jusqu'à 25000 mesures par seconde.

Structure interne du CAN 10 bits des 16F877A

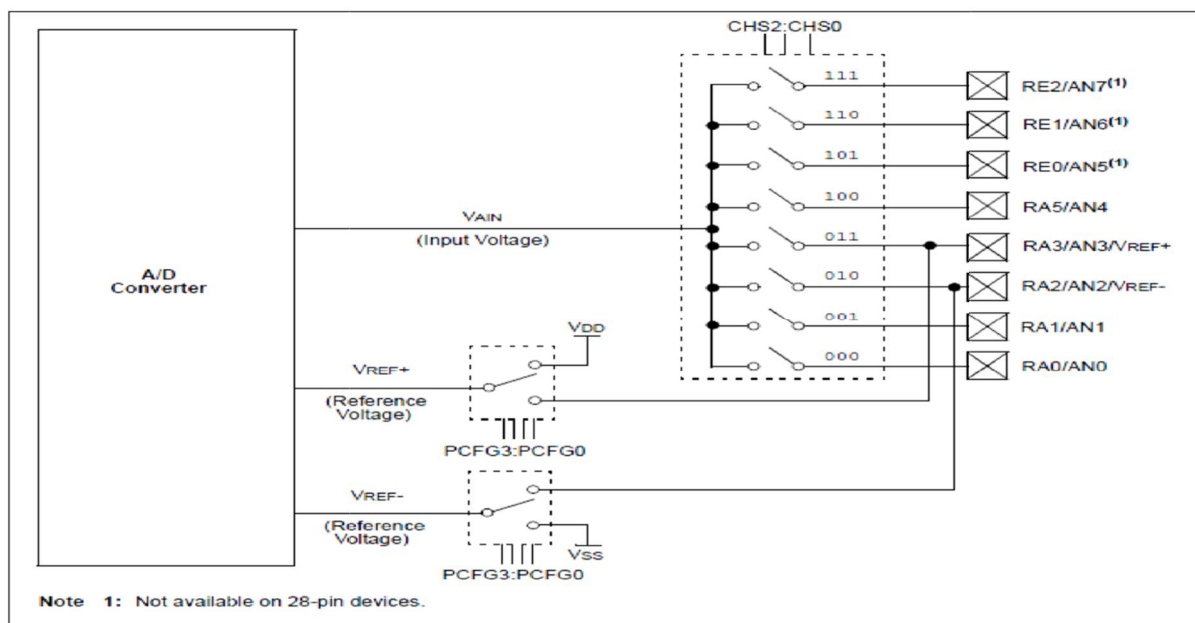


Figure II.4 : Structure interne du CAN 10 bits des 16F877A

Il y a 4 registres associés au module CAN du 16F877A :

ADRESH : le registre de résultat "Haut"

ADRESL : le registre de résultat "Bas"

ADCON0 : premier registre de contrôle du module CAN

ADCON1 : seconde registre de contrôle du module CAN

Les registres ADRESH et ADRESL

Le convertisseur Analogique Numérique est 10 bits et par conséquent 1 octet (=8 bits) ne suffit pas pour contenir le résultat de la mesure. C'est pour cela que 2 registres sont nécessaires et sont donc utilisés. De fait 6 bits restent inutilisés sur les 16 disponibles.

Il est possible de définir la « justification » du résultat au sein de ces 2 octets à l'aide d'un bit du registre ADCON1 : le bit ADFM.

Voici la disposition des bits de résultats dans les 2 octets en fonction de la valeur du bit ADFM :

	ADRESH								ADRESL							
ADFM	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	X9	X	X	X	X	X	X	X	X	X0
0	X9	X	X	X	X	X	X	X	X	X0	0	0	0	0	0	0

Tableau II.1 : la disposition des bits de résultats dans les 2 octets en fonction de la valeur du bit ADFM

Les cases vertes représentent le résultat sur 10 bits, X0 correspond au bit de poids faible et X9 au bit de poids fort.

Le registre ADCON0 :

Ce premier registre de contrôle contient plusieurs bits qui configurent le fonctionnement du module CAN :

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

- Bits ADCS1, ADCS0 : bits de sélection de l'horloge utilisée pour le module CAN.

ADCS1: ADCS0: A/D Conversion Clock Select bits

00 = Fosc/2

01 = Fosc/8

10 = Fosc/32

11 = FRC (clock derived from the internal A/D module RC oscillator)

- Bits CHS2, CHS1, CHS0 : les bits de sélection du canal analogique à mesurer. Avant toute conversion, il faut définir quel est le canal qui va être utilisé pour cette conversion. Ceci se fait grâce à ces 3 bits selon le tableau suivant :

CHS2 : CHS0 : Analogue Channel Select bits

000 = Channel 0. (RA0/AN0)
 001 = Channel 1. (RA1/AN1)
 010 = Channel 2. (RA2/AN2)
 011 = Channel 3. (RA3/AN3)
 100 = Channel 4. (RA4/AN4)
 101 = Channel 5. (RA5/AN5) ⁽¹⁾
 110 = Channel 6. (RA6/AN6) ⁽¹⁾
 111 = Channel 7. (RA7/AN7) ⁽¹⁾

- Bit GO/DONE : Le bit de lancement de la conversion et témoin de la fin de conversion.

Ce bit n'est actif que si le bit ADON=1, c'est-à-dire si le module CAN est actif. Il sert à lancer la conversion : en mettant ce bit à 1, on lance la conversion à partir de la voie analogique sélectionnée par les bits CHS2, CHS1 et CHS0. Une fois que la conversion est terminée, ce bit passe à 0 automatiquement. Il sera donc possible d'utiliser ce bit pour attendre la fin de la conversion en cours.

- Bit ADON : Le bit "d'allumage" du module CAN.

Ce bit sert à "allumer" le module CAN. Si ADON=0, le module CAN est inactif. Si le bit ADON=1, le module est opérationnel.

Le registre ADCON1 :

Ce second registre de contrôle contient plusieurs bits qui configurent le fonctionnement du module CAN :

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

- Bit ADFM : Configuration du format du résultat.

Ce bit sert à configurer le résultat au sein des 2 registres ADRESL et ADRESH selon le tableau II.2.

- Bits PCFG0, CFG1, PCFG2 et PCFG3 : Bits de configuration des ports A (en analogique ou numérique).

Lorsqu'une broche de ce port est configurée en analogique, elle devient un des canaux de mesure utilisables par le CAN. Lorsqu'elle est configurée en numérique, elle est une broche E/S comme les autres broches des autres ports.

La configuration de la nature des broches du port A est décrite dans le tableau suivant :

PCFG3: PCFG0	AN7 ⁽¹⁾ RE2	AN6 ⁽¹⁾ RE1	AN5 ⁽¹⁾ RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs ⁽²⁾
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011X	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog Input D = Digital I/O

Tableau II.2 : Configuration de la nature des broches analogique du pic

Déroulement d'une conversion analogique-numérique :

Étudions comment le module CAN 10 bits réalise une conversion analogique numérique. Le schéma suivant résume la séquence d'une conversion analogique-numérique par le module utilisé après avoir expliqué tous les bits et les registres nécessaires pour le déroulement de cette conversion :

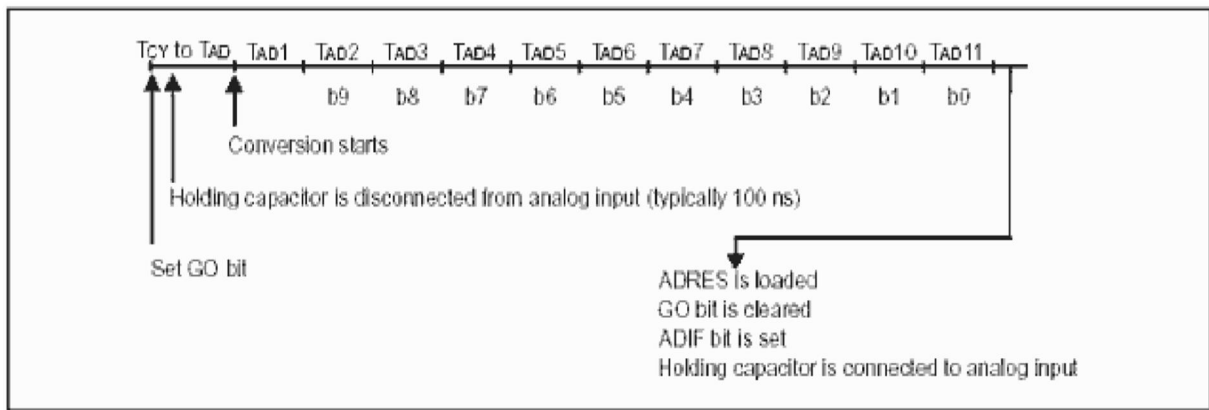


Figure II.5 : Déroulement d'une conversion analogique-numérique

L'ensemble de la conversion est lancé lorsque le bit GO/Done est mis à 1 par l'utilisateur. S'en suit un temps de latence interne lié à la présence d'un condensateur interne. Puis la conversion commence avec le "calcul" des 10 bits 4. Lorsque la conversion analogique-numérique est terminée :

- le résultat est stocké dans les registres ADRESL et ADREH
- le bit GO/Done est mis à 0 automatiquement témoignant de la fin de la conversion
- le drapeau de l'interruption "fin de conversion" est mis à 1.

II.2.2. Deuxième démarche : circuit de base pour le fonctionnement du PIC

Dans cette partie on va présenter les différents circuits de base pour le fonctionnement de PIC

Circuits d'alimentation :

Comme tout circuit intégré, un microcontrôleur a besoin d'être alimenté et c'est le rôle dévolu à VSS qui est la patte de masse, et à VDD qui est l'alimentation positive.

Notre montage nécessite une alimentation 5V pour alimenter le PIC et ces périphériques, mais aussi une tension de 12 V pour alimenter les amplificateurs opérationnels.

Autre contrainte présente dans ce circuit est qu'il peut être, dans certains cas, alimenté par une batterie 12 V dans des zones qui ne sont pas couverts par le réseau électrique. Pour ces raisons nous avons optés pour une alimentation externe, qui sera régulé et stabilisé vers la tension nécessaire au fonctionnement des composants qui illustrés ci-dessous :

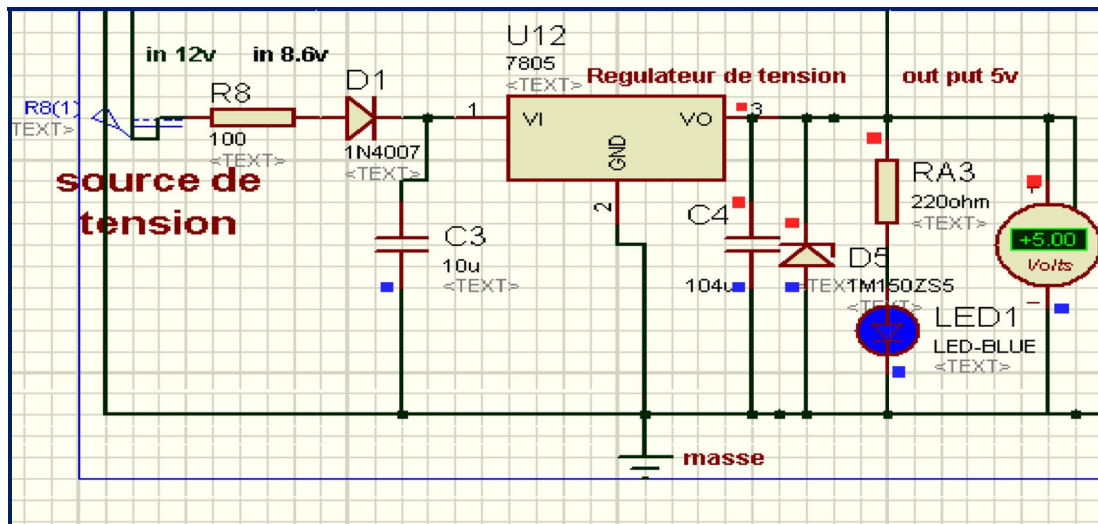


Figure II.6 : Circuit d'alimentation

Pour alimenter le circuit, une source continue d'environ 9V fera parfaitement l'affaire (d'ailleurs c'est le rôle de la résistance R8 d'amener la tension de 12v à 9v). La régulation et l'adaptation à un niveau de 5V requis par le PIC incombent au régulateur 7805.

Remarque : pour que celui-ci fonctionne correctement, la tension d'alimentation doit être supérieure d'au moins 2 ou 3 volts à la tension régulée,

La diode D1 protège le circuit contre les inversions accidentelles de polarité (diode type 1N4001 ou 1N4007 par exemple).

Le condensateur C4 sert de « réservoir » au régulateur. Le condensateur C5 nous permet de découpler le régulateur du reste du circuit : il absorbe les variations de tension parasite qui pourraient remonter vers le régulateur.

Le couple RA3 – Led1 sert de témoin de mise sous tension, et accessoirement décharge le condensateur C4 lorsque l'alimentation est coupée.

Circuit de l'horloge :

Sur tous les microcontrôleurs actuels le générateur d'horloge est intégré. Il ne reste plus qu'à placer un composant externe pour en fixer la fréquence d'utilisation. Il faut donc se reporter à la documentation technique de constructeur pour connaître la plage d'utilisation de circuit.

L'horloge est un élément essentiel dans un microcontrôleur ou microprocesseur car c'est elle qui rythme le fonctionnement de toute la logique interne et qui cadence donc l'exécution des instructions du programme.

Dans notre réalisation on a opté pour un oscillateur à quartz « parallèle » dont la fréquence d'oscillation nominale peut aller jusqu'à 20MHz (laquelle on a utilisé), et des condensateurs C1 et C2 pour lequel on prendra une valeur de 20pF, conformément aux indications de la documentation de Microchip.

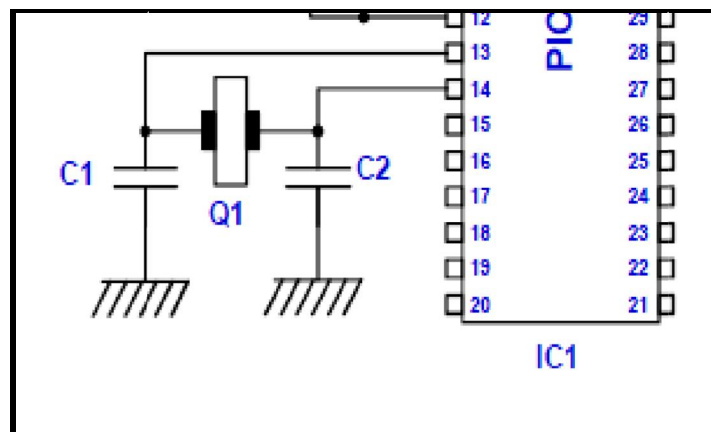


Figure II.7 : Circuit de l'horloge

Circuit Reset :

Le circuit Reset de pic (le pin 1 du PIC) est constitué d'une résistance qui est relié à la tension d'alimentation de 5v, d'une diode et d'un bouton poussoir qui est relié à la masse. Ainsi, pour redémarrer le PIC, on coupe l'alimentation et on réalimente.

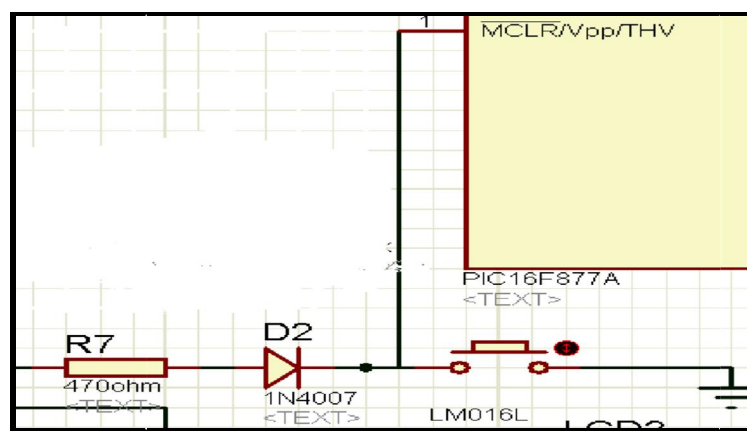


Figure II.8 : Le circuit RESET du pic

II.2.3. Troisième démarche : la conversion numérique-analogique

✓ But de CNA

Un CNA doit convertir une grandeur numérique (variable à temps discret et à valeurs discrètes) en une grandeur analogique (variable à temps continu et à valeurs continues).

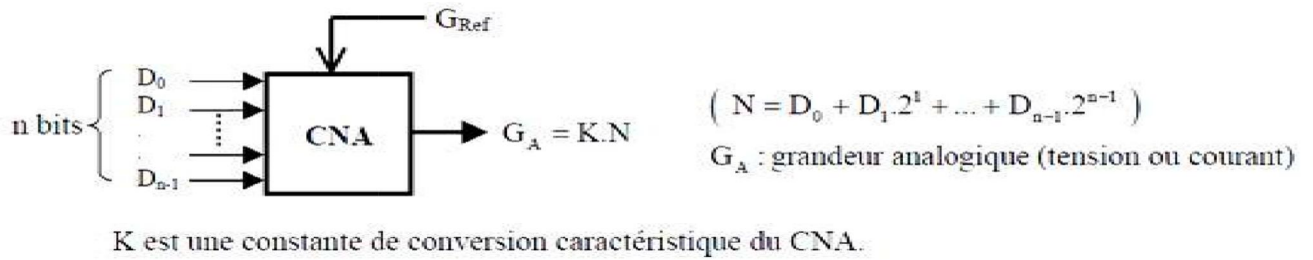


Figure II.9 : Schéma synoptique de principe de déroulement de la CNA

Principe de conversion :

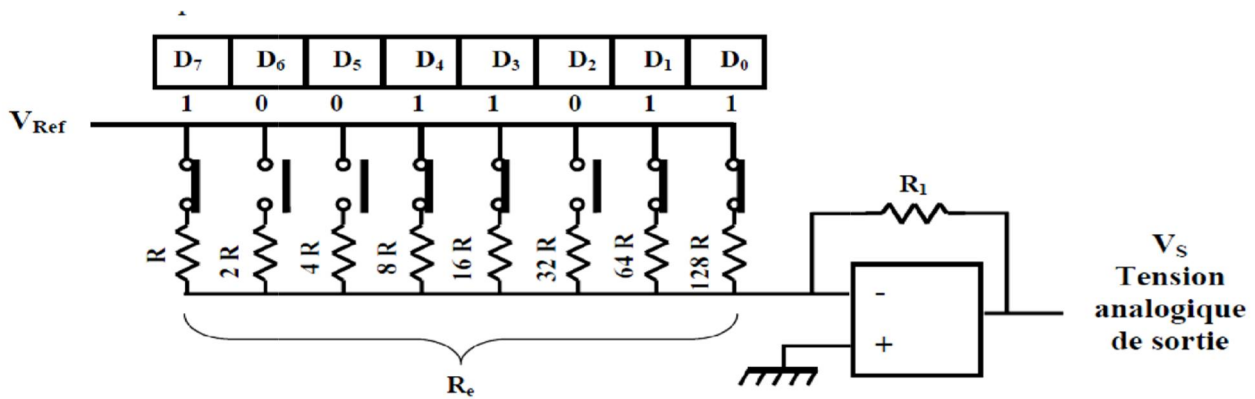


Figure II.10 : Convertisseur à résistances pondérées

La relation qui régit le schéma de conversion ci-dessous est :

$$V_s = -V_{Ref} \cdot \frac{R_1}{R} \cdot \left(D_7 + \frac{D_6}{2} + \frac{D_5}{4} + \frac{D_4}{8} + \frac{D_3}{16} + \frac{D_2}{32} + \frac{D_1}{64} + \frac{D_0}{128} \right)$$

$$= -\frac{V_{Ref}}{128} \cdot (D_7 \cdot 2^7 + D_6 \cdot 2^6 + D_5 \cdot 2^5 + D_4 \cdot 2^4 + D_3 \cdot 2^3 + D_2 \cdot 2^2 + D_1 \cdot 2^1 + D_0 \cdot 2^0) \quad \text{si } R_1 = R$$

Dans notre projet nous avons utilisé un convertisseur analogique numérique de courant "le DAC0808" suivi d'un convertisseur courant/tension à base d'un amplificateur opérationnel.

DAC 0808

Les DAC0808 sont des circuits monolithiques convertisseurs D/A 8 bits fournissant un courant pleine échelle de 150 mA et ne dissipant que 33 mW avec une alimentation égale à 5 V.

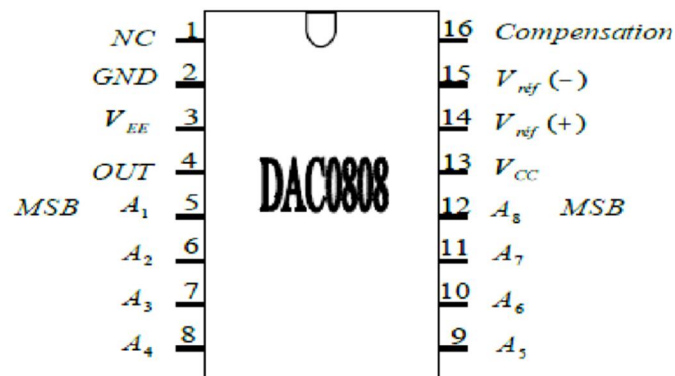


Figure II.11 : Brochage du DAC0808

Le tableau ci-dessous donne un aperçu sur les valeurs limites d'un CNA de type DAC0808 :

V_{CC}	+18V DC
V_{EE}	-18V DC
Tension d'une entrée digitale, V5-V12	-10 VDC à +18 VDC
Tension de sortie, Vo	-11 VDC à +18 VDC
Courant de référence, I14	5 mA
Tension de référence V14, V15	V_{CC} et V_{EE}
Puissance dissipée	1000 mW
Décote au-dessus de 25 °C	6.7 mW/°C
T° de fonctionnement DAC0808L	$-55^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
T° de fonctionnement DAC0808LC séries	$0 \leq T_A \leq +75^{\circ}\text{C}$
Plage de températures de stockage	$-65^{\circ}\text{C} \text{ à } +150^{\circ}\text{C}$

Tableau II.3 : Les valeurs limite d'un CNA de type DAC0808

Utilisation du DAC 0808

L'application typique du DAC0808 utilisée pour notre carte est de convertir le courant issu du convertisseur en une tension proportionnelle en utilisant le circuit TL084 qui est un amplificateur. La figure 2.12 illustre ce fonctionnement :

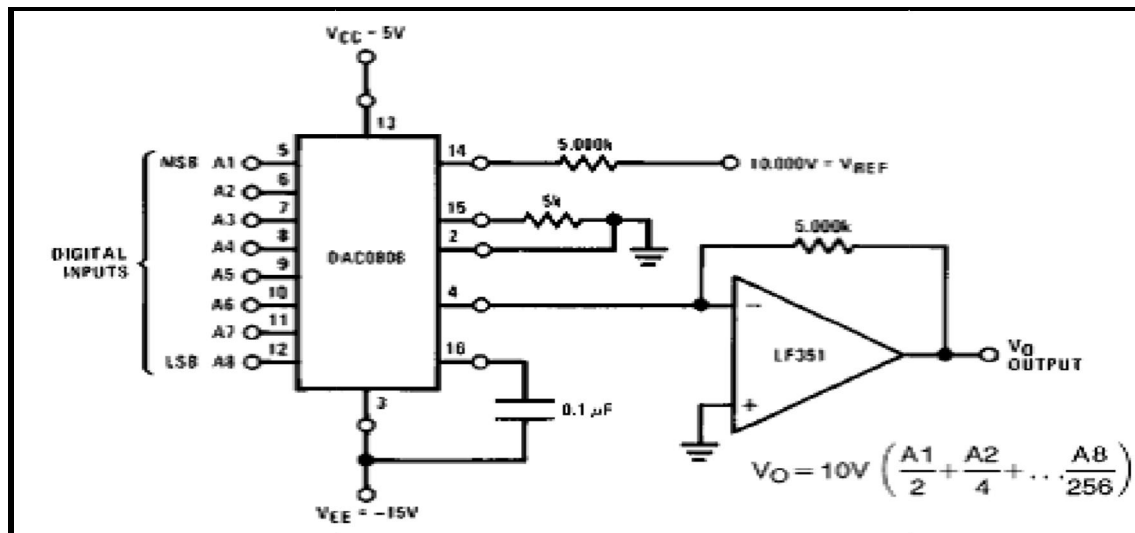


Figure II.12 : Schéma du câblage du DAC0808 avec l'amplificateur LF351

II.2.4. Quatrième démarche : bloc d'affichage LCD et clavier (buttons poussoirs).

II.2.4.1. blocs d'affichage LCD

Introduction :

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

Les afficheurs à cristaux liquides, appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA).

Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), et aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétro-éclairage. Cette fonction fait appel à des LED montées derrière l'écran du module.

Principe des cristaux liquides :

L'afficheur est constitué de deux lames de verre, distantes de 20 µm environ, sur lesquelles sont dessinées les mantisses formant les caractères. L'espace entre elles est rempli de cristal liquide normalement réfléchissant (pour les modèles réflexifs).

L'application entre les deux faces d'une tension alternative basse fréquence de quelques volts (3 à 5 V) le rend absorbant. Les caractères apparaissent sombres sur fond clair, ce qui nous permet de visualiser les caractères voulu.

Schéma fonctionnel :

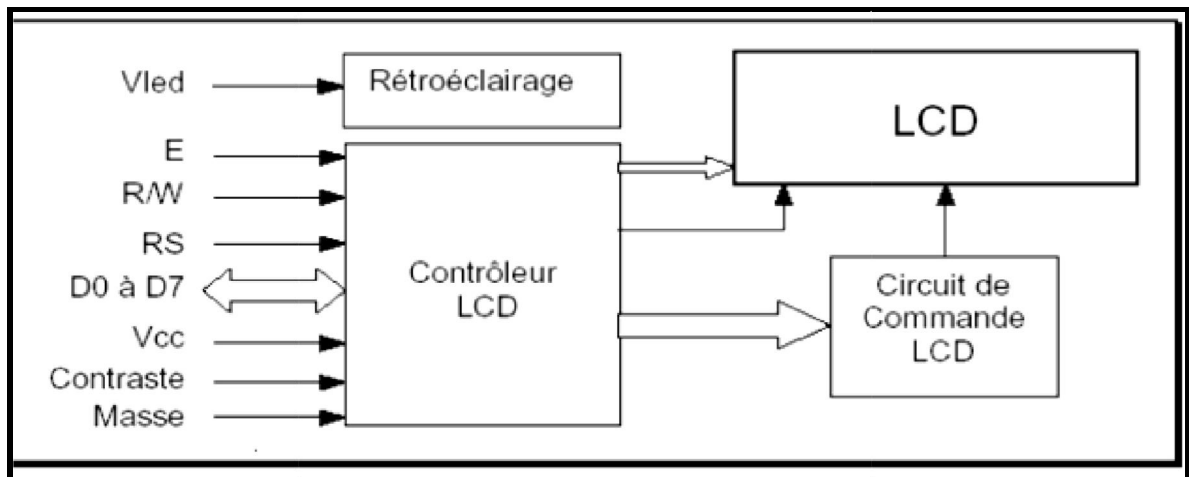


Figure II.13 : Schéma fonctionnel d'un afficheur LCD

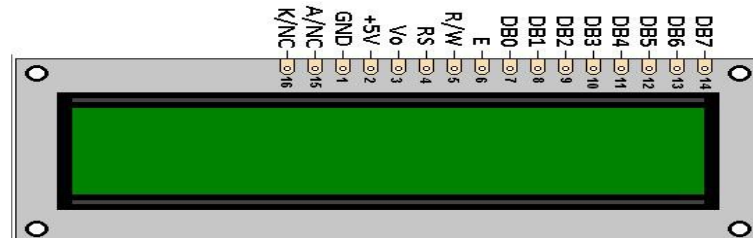


Figure II.14 : photo de LCD et son brochage

Rôle des différentes broches de l'afficheur LCD

Broche	Nom	Fonction
1	VSS	Masse
2	VDD	Alimentation positive (+5V).
3	VEE	Cette tension permet, en la faisant varier entre 0 et +5V, le réglage du contraste de l'afficheur.
4	RS	Sélection du registre (Register Select) Grâce à cette broche, l'afficheur est capable de faire la différence entre une commande et une donnée. Un niveau bas indique une commande et un niveau haut indique une donnée.
5	RW	Lecture ou écriture (Read/Write) L : Écriture H : Lecture
6	E	Entrée de validation (Enable) active sur front descendant. Le niveau haut doit être maintenue pendant au moins 450 ns à l'état haut.
7-14	D0-D7	Bus de données bidirectionnel 3 états (haute impédance lorsque E=0)
15	A	Anode rétro éclairage (+5V)
16	K	Cathode rétro éclairage (masse)

Tableau II.4 : Rôle des différentes broches de l'afficheur LCD

Dans notre projet on a utilisé l'afficheur LCD pour afficher les différentes valeurs de Kp, Ki et Kd qui seront choisies par des boutons poussoirs

2.4.4.2 Claviers (boutons poussoirs) :

Le clavier doit permettre la saisie et la modification des paramètres du correcteur PID : Kp, Ki, Kd et consigne en cas de consigne interne et ils sont gérés par un programme qu'on va implémenter dans le pic et la disposition des six boutons dans la carte qui sont connectés au PIC sur les pins RA4, RA5, RC0, RC1, RC2, RC3 respectivement pour les boutons de consigne, kp, ki, kd, bouton d'incrément de 0.1 et bouton de décrémentation sont montrés dans la figure suivante.

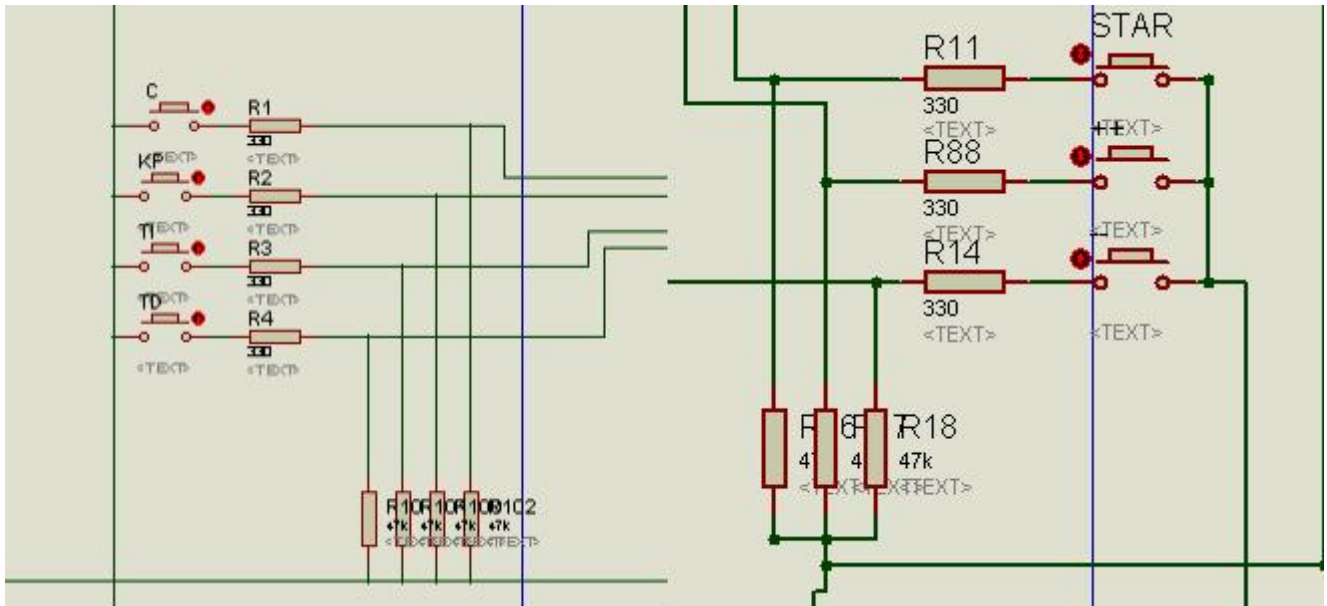


Figure II.15 : Disposition des boutons de réglages sur la carte.

II.3. Conception software

II.3.1. Introduction

La programmation des PIC se fait par le langage assembleur qui est un langage de bas niveau qui représente le langage machine sous une forme lisible par un humain. Les combinaisons de bits du langage machine sont représentées par des symboles dits « mnémoniques » (du grec mnêmonikos, relatif à la mémoire), c'est-à-dire faciles à retenir.

Le programme assembleur convertit ces mnémoniques en langage machine en vue de créer par exemple un fichier exécutable.

Le développement des environnements de programmation, a permis de voir naître de nouveaux compilateurs qui permettent de programmer avec les langages haut niveau tels que le C, PASCAL, BASIC etc...

Ces environnements comportent aussi des bibliothèques qui permettent de faciliter le développement. Il existe plusieurs outils de développement, les uns sont gratuits, les autres sont payants.

Dans notre recherche de l'outil que nous allons utiliser pour programmer notre PIC afin de réaliser la fonction PID et l'implémenter dans le PIC, nous avons optés pour le langage C. Ce choix est à la fois un choix personnel et un choix technologique.

D'une part le langage C est utilisé dans différents systèmes et domaines de développement, ce qui nous permettra une évolution future, d'autre part le langage C est l'un des langages les plus puissants.

Avantages du C

Pour de la programmation de base, le C est intéressant. Il permet rapidement, sans gros effort, de développer des programmes fonctionnels. Il permet aussi de s'affranchir de connaissances complexes sur l'architecture des PIC. Il a l'avantage de gérer facilement les boucles, les choix, ainsi que l'affichage.

Inconvénients du C

Le C n'est pas le langage naturel du microcontrôleur. Il permet de programmer plus intuitivement. Les logiciels de programmation en C transforment alors les lignes en C en lignes assembleurs directement compréhensibles par le microcontrôleur. Pour programmer efficacement, il est souvent nécessaire d'aller voir le code assembleur, il est donc conseillé d'avoir des bases solides en assembleur, nous avons fait le choix d'utiliser l'environnement de développement MikroC de Mikroelektronika.

II.3.2. Présentation du MikroC

Le « MikroC » est un compilateur pour PIC Conçu par la société « Mikroelektronika », le compilateur C nouvelle génération "MikroC" pour microcontrôleurs PIC bénéficie d'une prise en main très facile. Il comporte plusieurs outils intégrés (mode simulateur, terminal de communication, gestionnaire 7 segments, analyseur statistique, correcteur d'erreur, explorateur de code...) ; Il a une capacité à pouvoir gérer la plupart des périphériques rencontrés dans l'industrie (Bus I2C, 1Wire, SPI, RS485, Bus CAN, cartes compact Flash, signaux PWM, afficheurs LCD et 7 segments...) ; de ce fait il est un des outils de développement incontournable et puissant

Il est conçu pour fournir les solutions les plus faciles que possibles pour des applications se développant pour les systèmes à microcontrôleur. Il contient un large ensemble de bibliothèques de matériel, de composant et la documentation complète.

Le compilateur MikroC nous permet de développer rapidement des applications complexes.

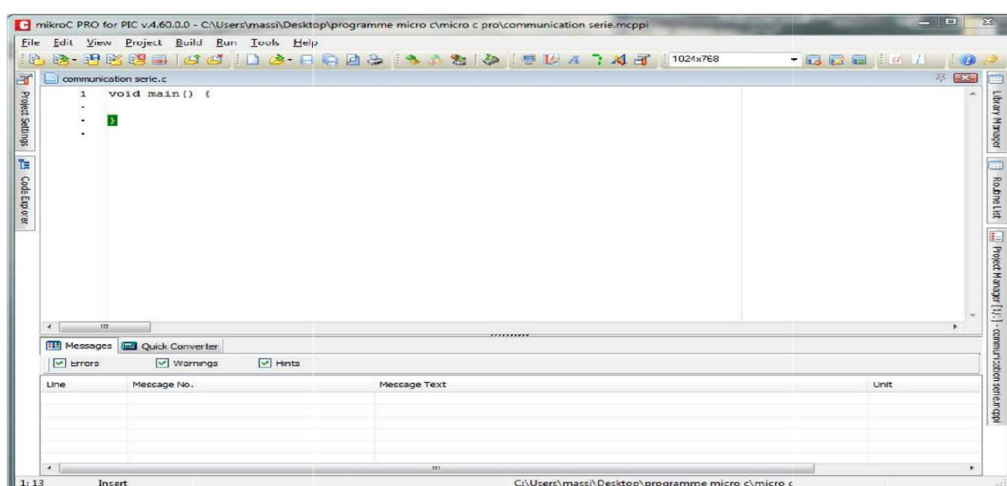


Figure II.16 : Interface du logiciel mikroC PRO for PIC

II.3.3. Procédures de création de programme sous mikroC PRO for PIC

On clique tous d'abord sur le bouton « File » en suite « New Project »

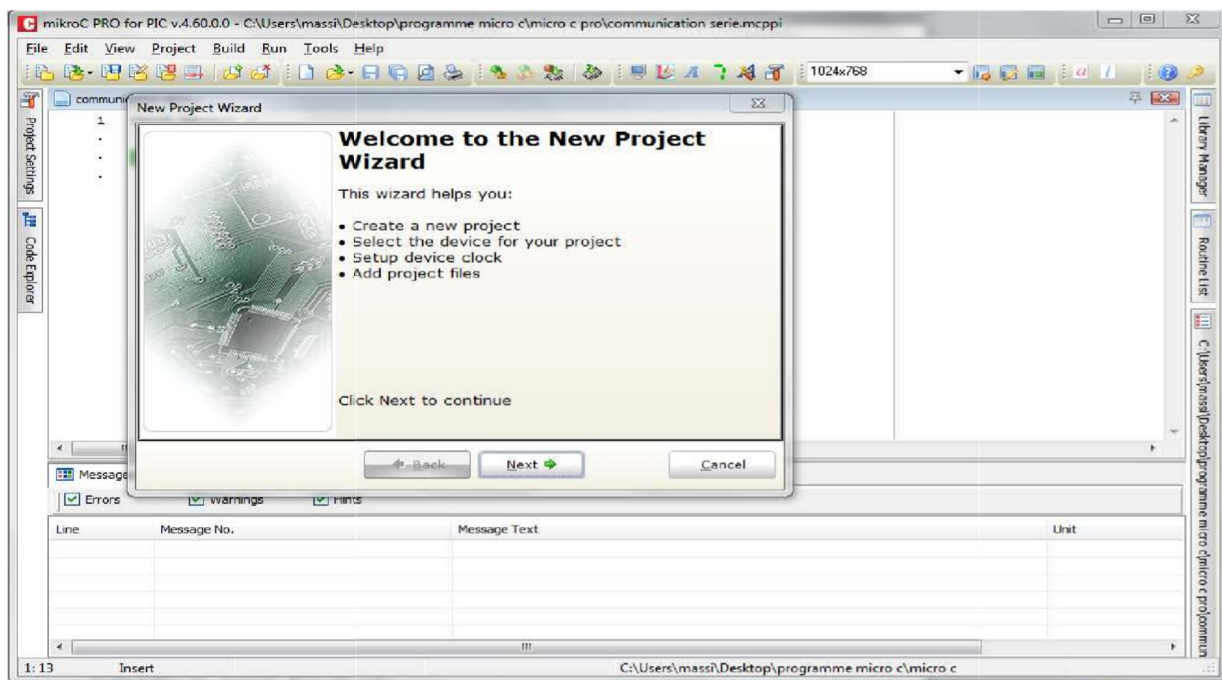


Figure II.17 : Fenêtre de création d'un nouveau projet

Ensuite on choisira notre PIC

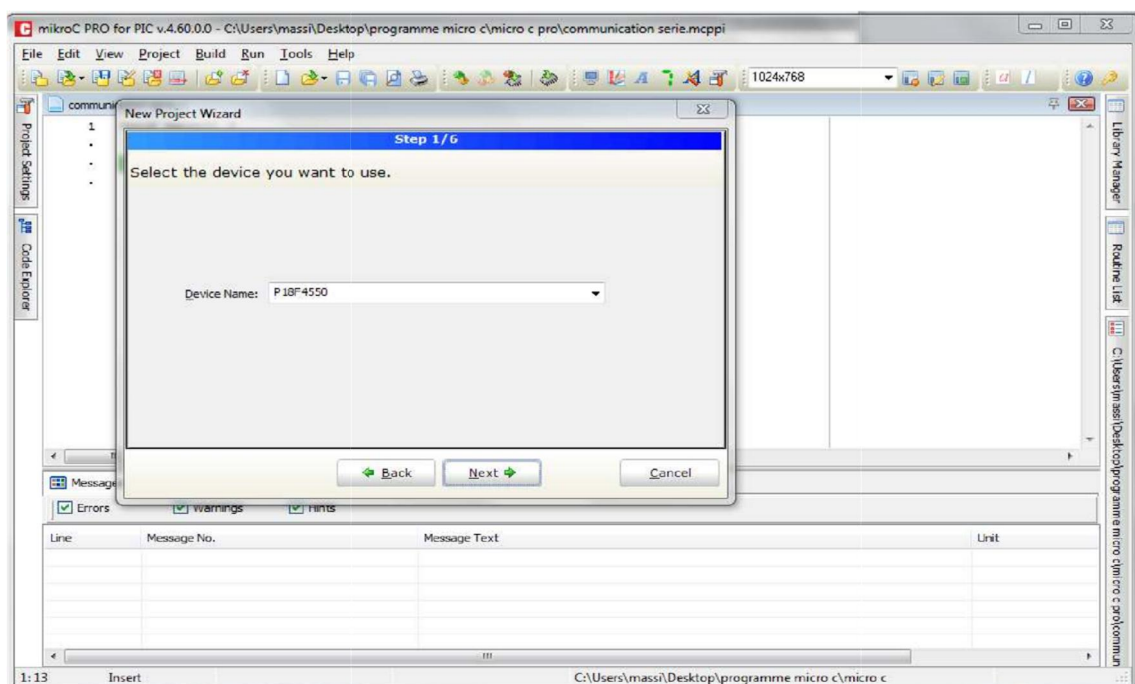


Figure II.18 : Fenêtre pour le choix du pic

On choisit la fréquence avec laquelle on va travailler

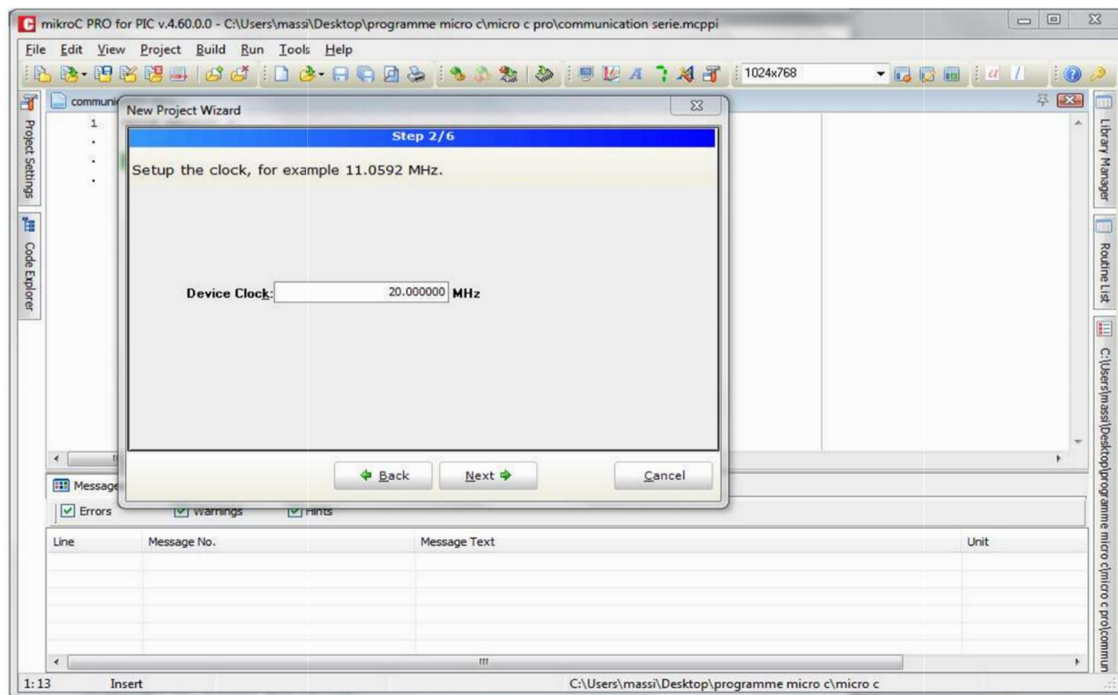


Figure II.19 : Fenêtre pour la configuration de l'horloge

On donne un nom à notre projet

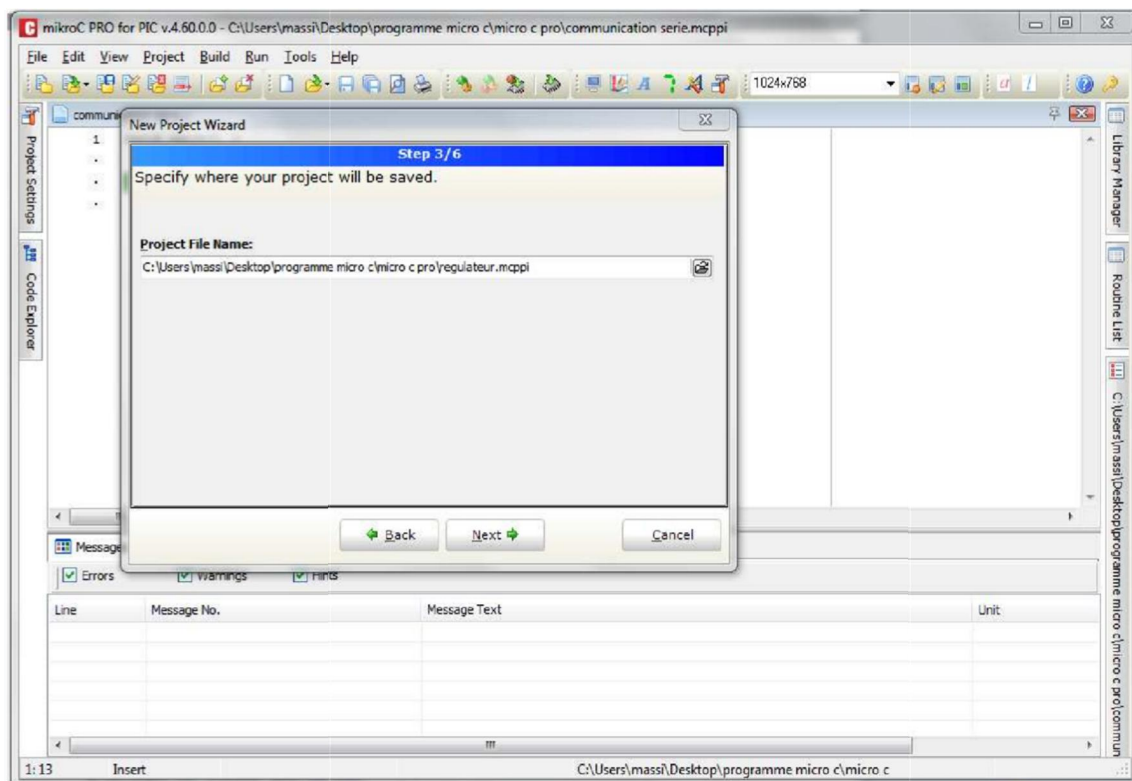
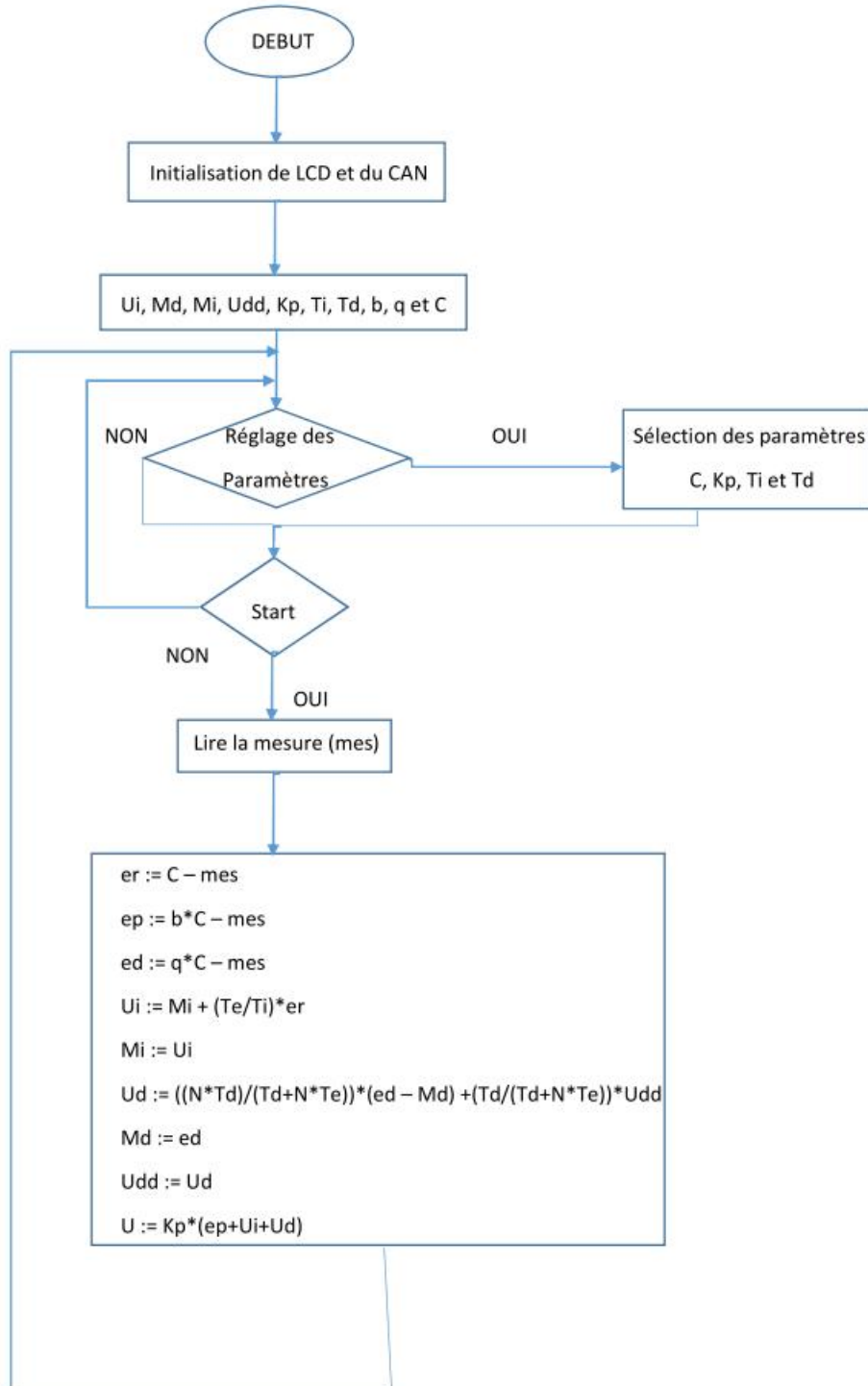


Figure II.20 : Fenêtre pour le choix de l'emplacement de projets à créer

II.3.4. Organigramme du Programme implémenté dans le PIC

- Cas d' régulateur



Explication des termes utilisés l'organigramme :

Mi : Mémoire du terme intégral précédent.

ep : erreur proportionnelle.

ed : erreur dérivée

Ma : Mémoire de l'erreur dérivée précédente.

Udd : Mémoire de du terme dérivé précédent.

Kp : Gain de l'action proportionnel.

Ti : Temps intégrale.

Td : Temps dérivée.

er := **C-M** : calcule de l'erreur entre la consigne et la mesure

ep := **b*C – mes** : calcule de l'erreur proportionnelle

ed := **q*C – mes** : calcule de l'erreur dérivée

Ui := **Mi +(Te/Ti)*er** : calcule du terme intégrale

Mi := **Ui** : mémorisation du terme intégrale

Ud := **((N*Td)/(Td+N*Te))*(ed – Md)+(Td/(Td+N*Te))*Udd** : calcule du terme Dérivée

Ma := **ed** : mémorisation de l'erreur dérivée

Udd := **Ud** : mémorisation du terme dérivé

U := **(E +Ui+Ud)*Kp** : calcule de la commande pour une structure d'un PID mixte.

Conclusion

Dans ce chapitre on a opté pour l'étude des microcontrôleurs de type PIC en détaillant ses différentes caractéristiques, on s'est concentré plus précisément sur le PIC16f877A et cela pour mettre en évidence les raisons qui poussent notre choix à se porter sur ce PIC. On s'est intéressé aussi à la circuiterie et la conception software de notre boucle de régulation, tout d'abord la conception du circuit dans laquelle nous avons détaillé chaque bloc composant notre carte électronique ensuite la conception software dans laquelle nous avons présenté la manière avec laquelle nous avons développé le programme qu'on a implémenté dans notre microcontrôleur ainsi que la présentation du logiciel de programmation des microcontrôleurs mikroC for PIC PRO.

Chapitre III

Simulation et tests

Introduction

Dans ce chapitre on va procéder au teste des trois actions de notre régulateur pour vérifier le fonctionnement de chacune, le fonctionnement de bloc de conversion analogique-numérique (CAN) et numérique-analogique (CNA), le bloc d'affichage (LCD) et les buttons poussoir ainsi que les résultats de la régulation des systèmes premier et deuxième ordre.

III.1. Logiciel de simulation et conception de circuit imprimé

Avant de passer à la réalisation pratique de notre système nous avons simulé d'abord les différentes parties du système.

Pour cela on a utilisé le logiciel ISIS qui est un très bon logiciel de simulation en électronique.

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

La simulation permet d'ajuster et de modifier le circuit comme si on manipulait un montage réel. Ceci permet d'accélérer le prototypage et de réduire son coût.

Il faut toujours prendre en considération que les résultats obtenus de la simulation sont un peu différents de celles du monde réel, et cela dépend de la précision des modèles, des composants et de la complication des montages.

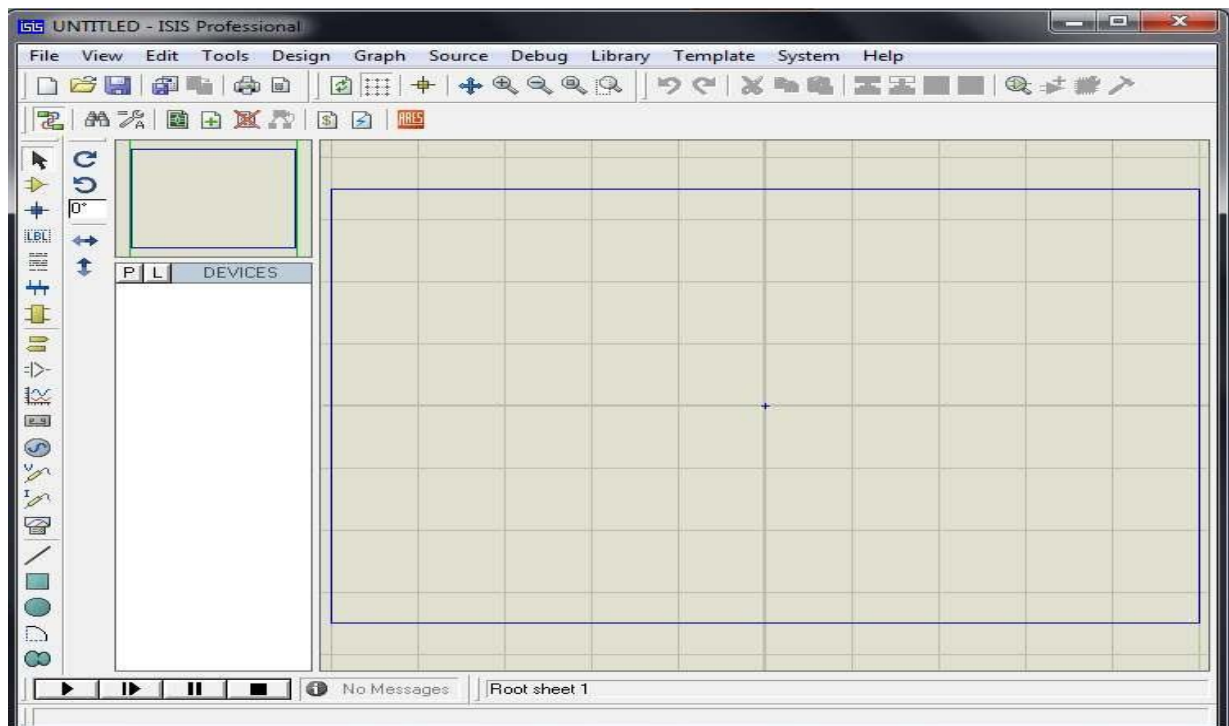


Figure III.1 : Interface de logiciel de simulation ISIS.

III.2. Test de bloc de la CNA, CAN et l'action proportionnelle

Pour tester le bon fonctionnement du CAN, on lui injecte un signal quelconque, puis on lit le résultat de la conversion en sortie du PIC à l'aide de 8 diodes placées sur chacun des bits du bus de sortie puis cette sortie est utilisée pour le test du CNA qui se fait en même temps que le test du correcteur proportionnel, quand K vaut 1. Nous injectons en entrée un signal sinusoïdal et nous devons visualiser sur l'oscilloscope une sinusoïde similaire de même amplitude. Nous effectuons le même test avec cette fois-ci K égal à 2. Nous devons pouvoir visualiser une sinusoïde ayant une amplitude 2 fois plus grande que celle injectée en entrée.

❖ Circuit d'essai :

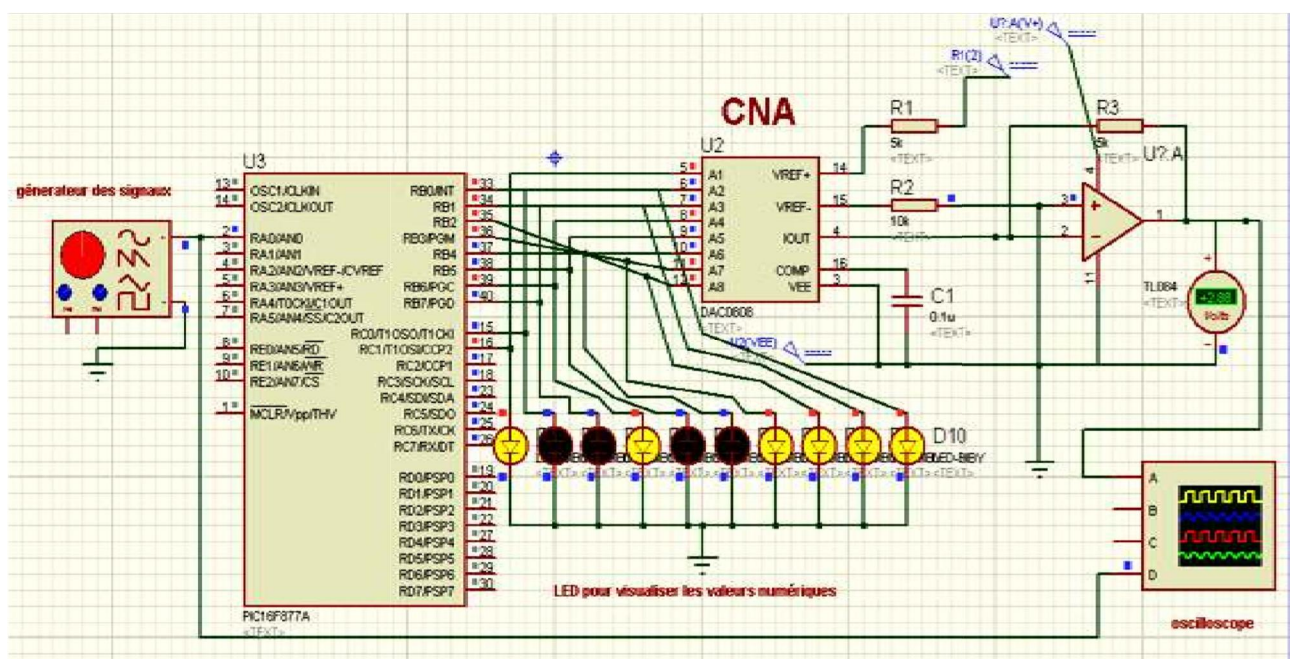


Figure III.2 : Circuit de PIC et CNA de régulateur PID

❖ Réponse visualisé dans l'oscilloscope :

- ✓ L'équation programmé qui réalise la fonction proportionnelle est :

$$u(k) = K_p * \varepsilon(k) \quad (\text{III.1})$$

- ✓ Le signal A en vert représente le signal d'entrée qui va être numérisé et qu'on va lui appliquer l'action proportionnel avec un K_p différent.
- ✓ Le signal B en jaune représente le signal de sortie.

1^{ère} cas $K_p=1$: c'est pour vérifier le bon fonctionnement de conversion analogique-numérique dans le PIC ainsi que la conversion numérique-analogique avec le DAC 0808 :

- ✓ échelle de l'oscilloscope : 0.2 V/ div pour l'axe verticale et 50ms/div pour l'axe des abscisses.

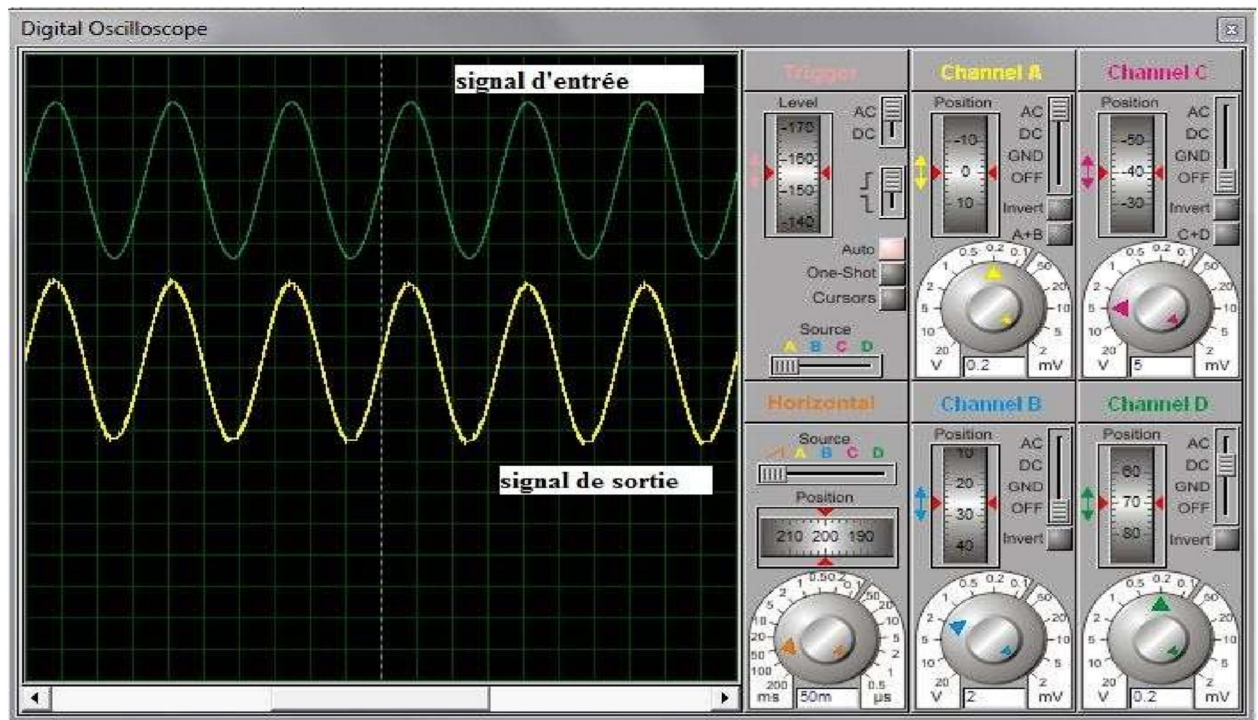


Figure III.3 : Signaux d'entrée et de sortie pour $k_p=1$

2^{ème} cas $K_p=2$: c'est pour tester le fonctionnement de l'action proportionnelle implémenté dans le PIC :

- ✓ Echelle de l'oscilloscope : 0.2 V/ div pour l'axe verticale et 14.08ms/div pour l'axe des abscisses.

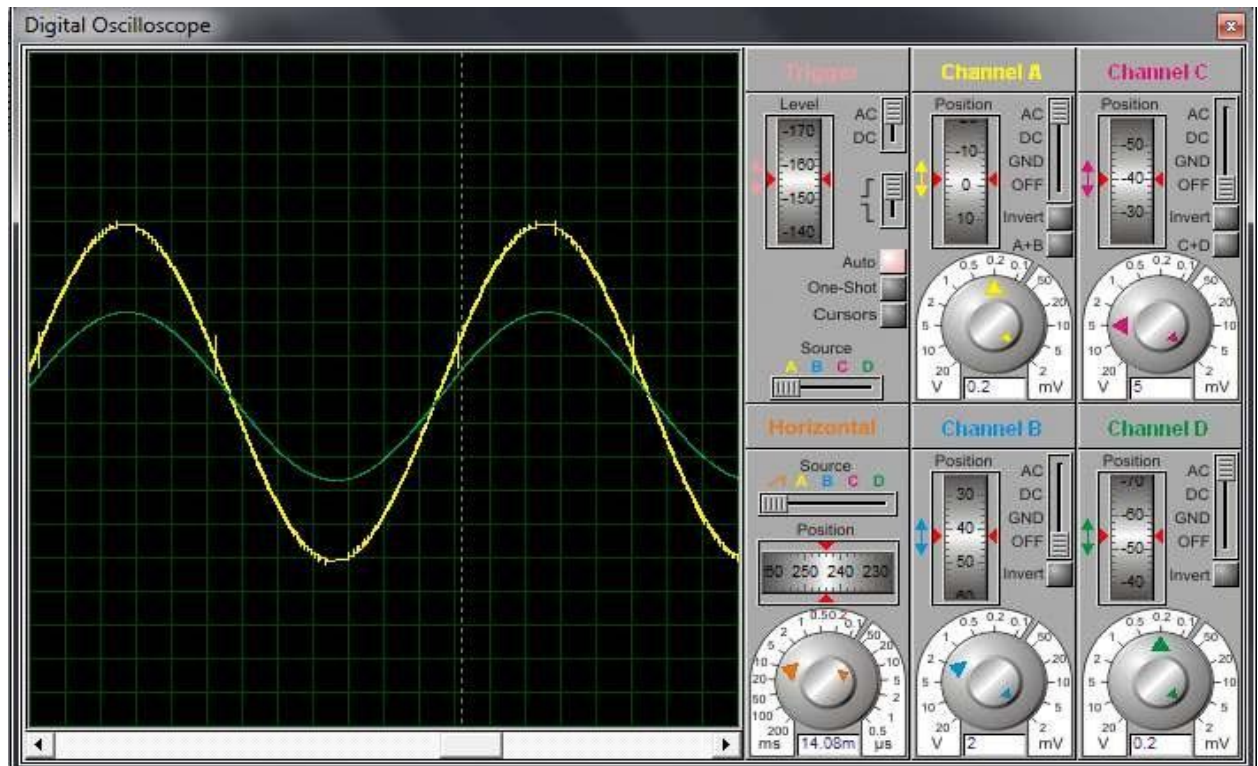


Figure III.4 : Signaux d'entrée et de sortie pour $k_p=2$

Interprétation :

❖ Dans la première simulation pour $K_p=1$ on a pu vérifier le bon fonctionnement et la bonne mise en œuvre de programme de la conversion analogique-numérique et de circuit de conversion analogique-numérique avec le DAC 0808.

❖ Dans la deuxième simulation pour le $K_p=2$ on a pu tester le bon fonctionnement de l'action proportionnelle tel qu'on a obtenu en sortie un signal d'amplitude de 3v pour un signal en entrée de 1v.

III.3. Teste de l'action intégrale

Pour tester l'action intégrale du correcteur on injecte un signal carré en entrée et nous devons visualiser un signal triangulaire sur l'oscilloscope.

✓ L'équation programmée qui réalise la fonction intégrale est :

$$u_i[n] = \frac{T_e}{T_i} \cdot \sum_{k=0}^n e[k] = u_i[n-1] + \frac{T_e}{T_i} \cdot e[n] \quad (\text{III.2})$$

✓ Le signal A en jaune représente le signal d'entrée qui va être numérisé et qu'on va lui appliquer l'action intégrale avec un $K_i=1$. Le signal B en vert représente le signal de sortie.

✓ Échelle de l'oscilloscope : 1 V/ div pour le signal d'entrée et 0.16 V/ div pour le signal de sortie et 25ms/div pour l'axe des abscisses.

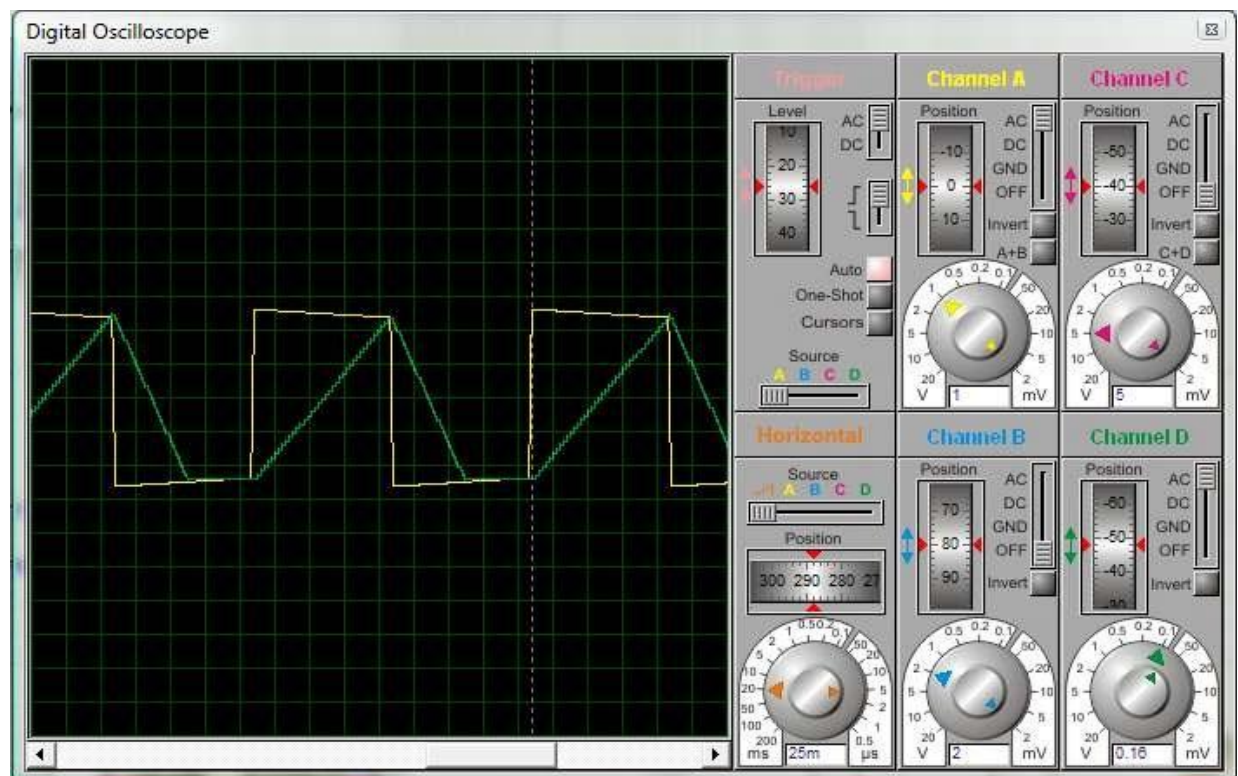


Figure III.5 : Signaux d'entrée et de sortie pour $k_i=1$

Interprétation

D'après le signal obtenu en sortie de l'oscilloscope nous pouvons conclure par dire que la fonction intégrale implémentée dans le PIC est vérifiée tout en sachant que l'intégrale d'une constante est une rampe dont la constante va représenter la pente de cette rampe et c'est bien ça qu'on a obtenu dans la sortie de l'oscilloscope.

III.4. Teste de l'action dérivée

Pour tester l'action dérivée il faut injecter un signal triangulaire en entrée et nous devons récupérer en sortie un signal carré.

✓ L'équation programmée qui réalise la fonction dérivée est :

$$U_d[n] = \left(\frac{NT_d}{T_d + NT_e} \right) (e_d[n] - e_d[n - 1]) + \left(\frac{NT_d}{T_d + NT_e} \right) u_d[n - 1] \quad (\text{III.3})$$

✓ Le signal A en jaune représente le signal d'entrée qui va être numérisé et qu'on va lui appliquer l'action dérivée avec un $k_d = 9$.

✓ Le signal B en vert représente le signal de sortie.

✓ Échelle de l'oscilloscope : 1 V/div pour le signal d'entrée et 15.67 ms/div pour l'axe des abscisses.

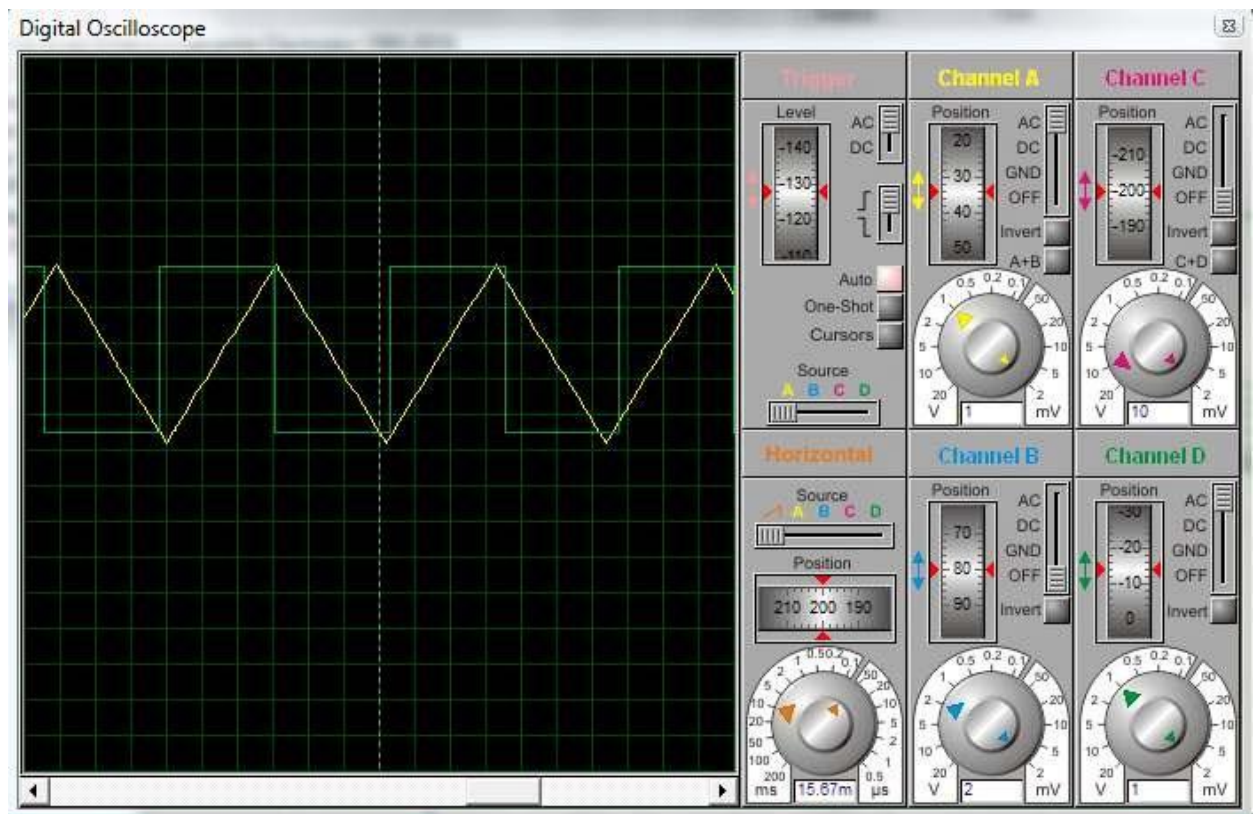


Figure III.6 : Signaux d'entrée et de sortie pour $k_d = 9$

Interprétation

D'après le signal obtenu de l'oscilloscope nous pouvons conclure par dire que la fonction dérivée implémentée dans le PIC est vérifiée connaissant que la dérivée d'une rampe est une constante dont la pente de la rampe va représenter la valeur de cette constante et c'est bien ça qu'on a obtenu dans la sortie de l'oscilloscope.

III.5. Test des boutons poussoirs et le bloc d'affichage (LCD)

Pour vérifier le bon fonctionnement de l'afficheur LCD et les boutons poussoirs il suffit d'afficher différentes valeurs afin d'essayer toutes les possibilités d'affichage. Ainsi que les mots qu'on désire afficher indiquant l'état de régulateur.

Et à titre d'exemple nous avons pris cette figure illustrant le réglage de la valeur k_p et c'est la même chose pour les autres actions et quand il n'y a pas de réglage des valeurs k_p , k_d , k_i l'afficheur indiquera l'état de régulateur.

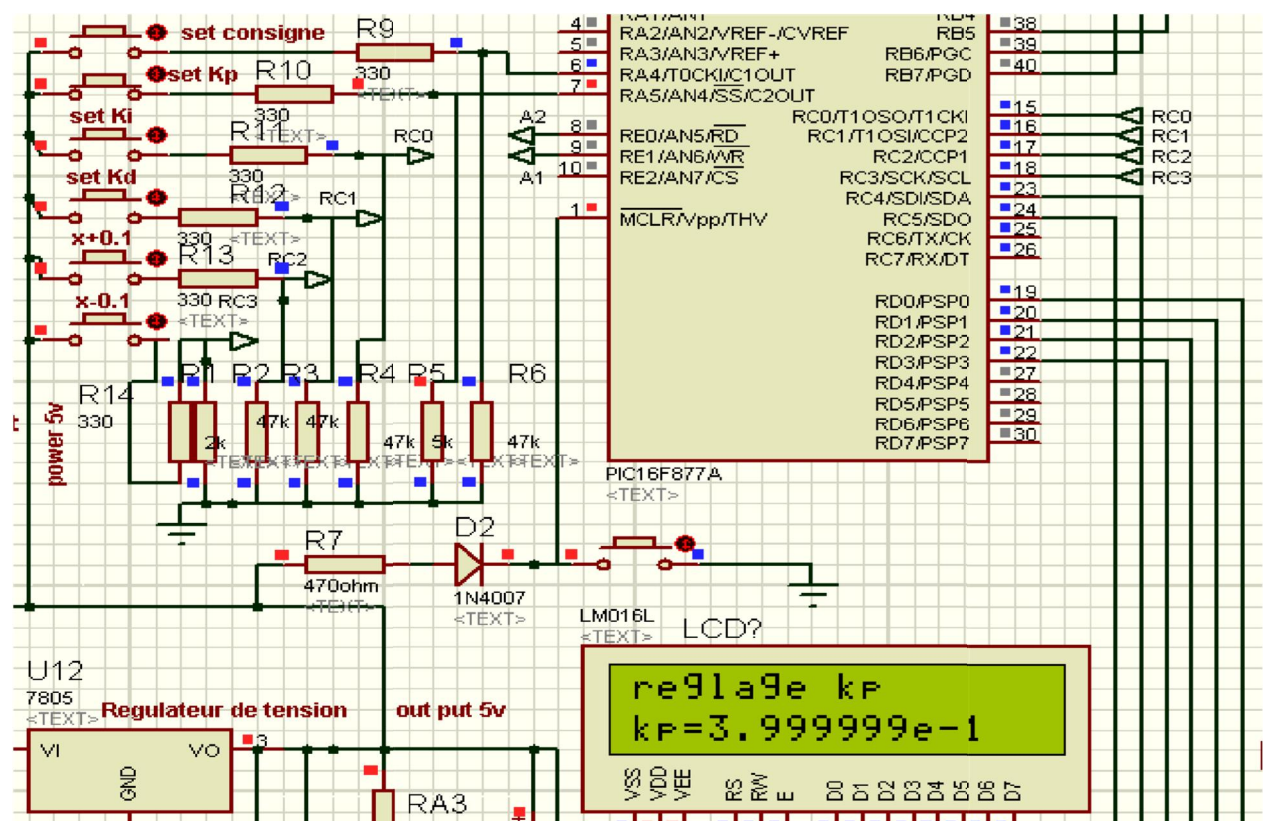


Figure III.7 : Figure illustrant le fonctionnement de LCD avec les BP

III.6. Teste de la régulation d'un système 1^{er} ordre

Pour tester cette régulation, nous essayons d'abord d'injecter les paramètres qui caractérisent le système (dans le pic qui s'occupe du modèle) que nous voulons réguler et de lui appliquer un signal de commande en calculant ses paramètres de PID et les injecter dans notre régulateur tout en essayant de le perturber et nous devons récupérer un signal de sortie qui, après un certain temps, rejoint notre signal d'entrée (consigne).

- ✓ L'équation programmée qui caractérise le système de premier ordre est :

$$y(k) = AK(1 - e^{-\frac{k \cdot T_e}{\tau}}) \quad (\text{III.4})$$

- ✓ Le signal A en bleu est le signal de sortie (mesure) qui est la réponse indicielle du système de 1^{er} ordre avec un gain statique $K=1$ et d'une constante de temps $\tau=2$ subi à une perturbation de 2V.
- ✓ Le signal B en jaune est le signal de commande fourni par le régulateur d'un $K_p=0.7$, $T_i=0.6$, $T_d=0.4$ et $N=0.005$.
- ✓ La consigne C=3.5V.

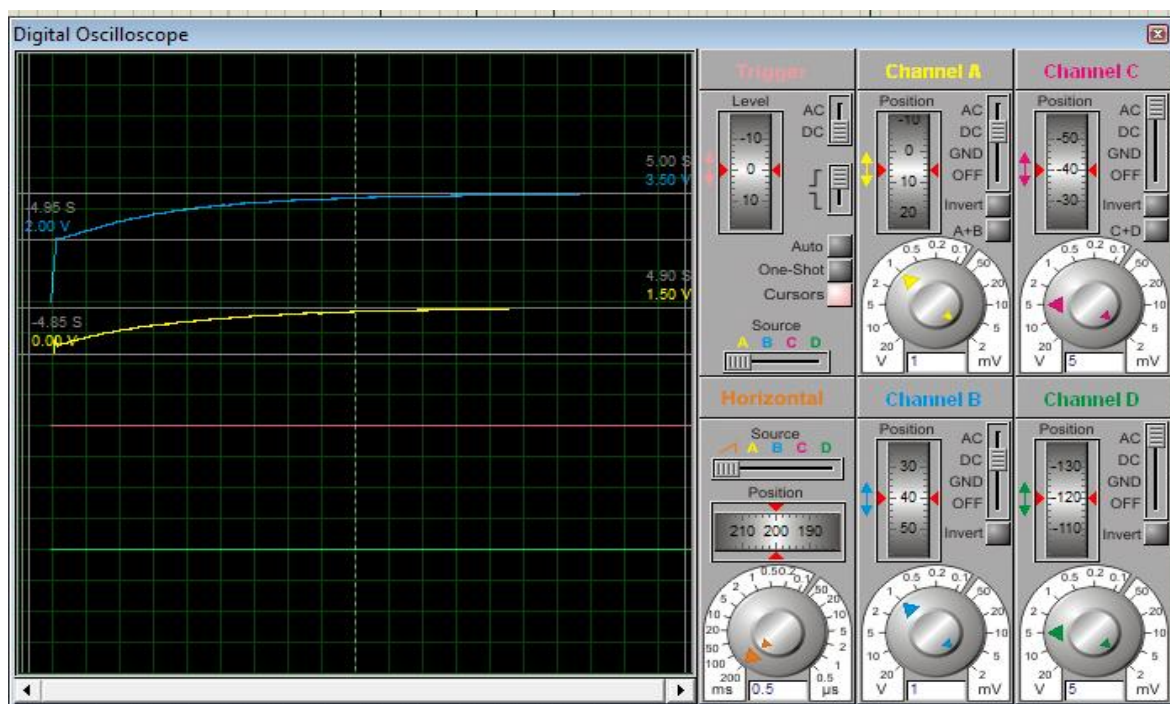


Figure III.8 : Signaux de commande et de mesure d'une régulation PID d'un système 1^{er} ordre

Interprétation

D'après le signal obtenu de l'oscilloscope nous pouvons conclure par dire que la régulation s'est déroulée correctement en comparant ces deux signaux à ceux de MATLAB.

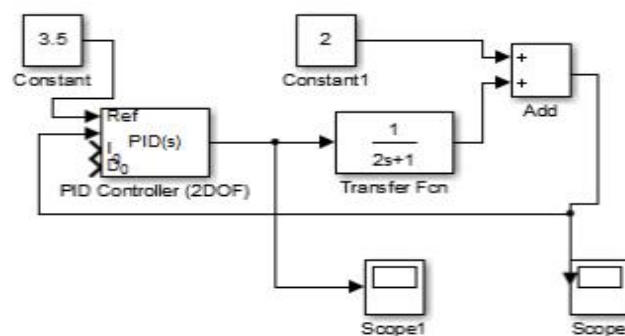


Figure III.9 : le schéma de simulation d'une boucle de régulation

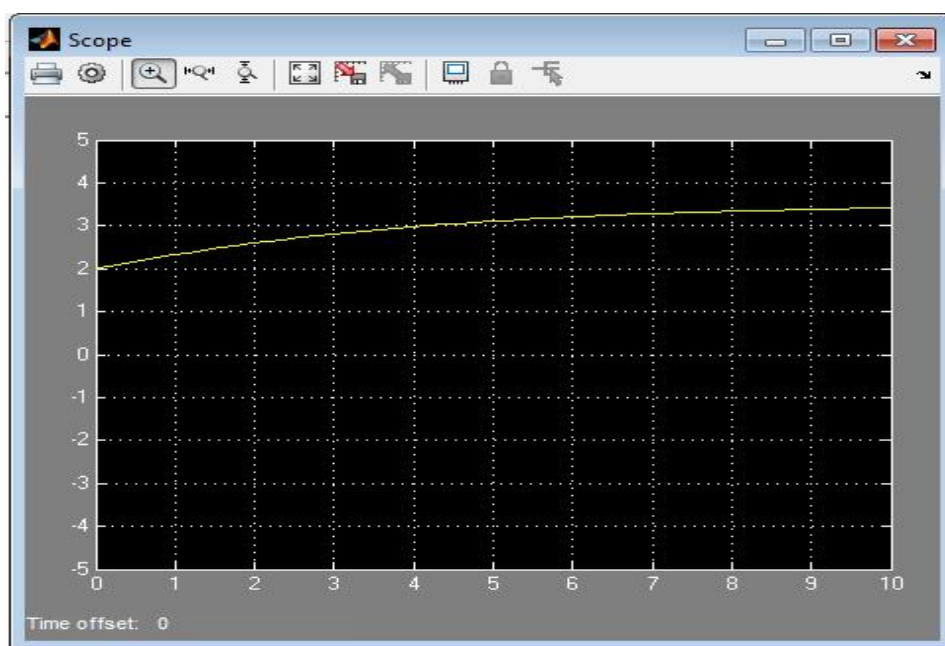


Figure III.10 : signal de mesure.

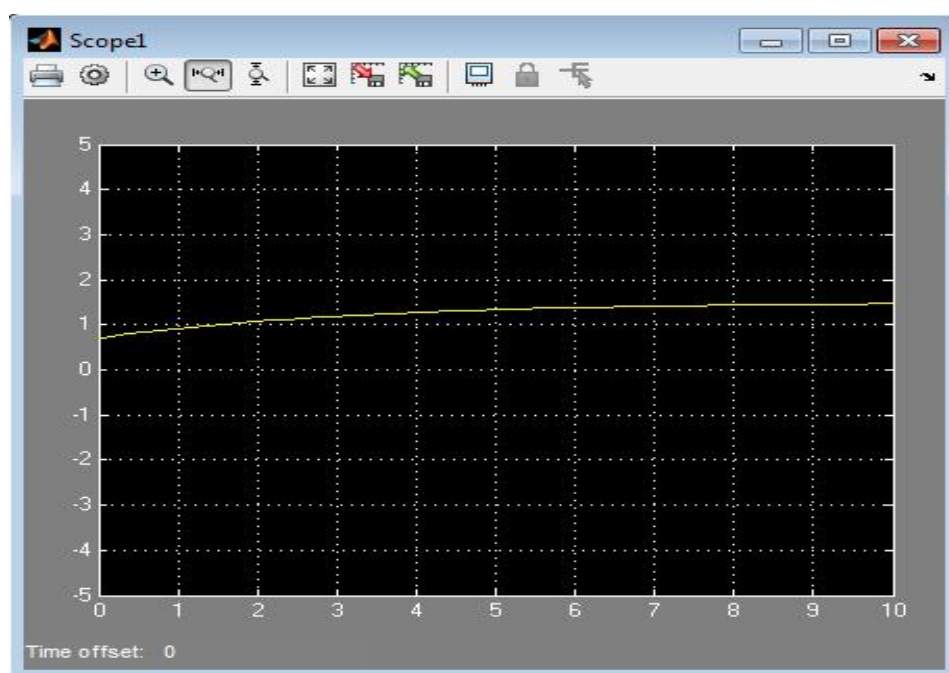


Figure III.11 : signal de commande.

III.7. Teste de la régulation d'un système deuxième ordre

On suit la même procédure que celle du premier ordre sauf que dans les systèmes de deuxième ordre il existe trois cas et cela dépend du coefficient d'amortissement :

- Régime apériodique $\xi > 1$.
- Régime apériodique critique $\xi = 1$.
- Régime oscillatoire $\xi < 1$.

➤ **Cas de $\xi > 1$:**

L'équation programmée qui caractérise le système de deuxième ordre avec $\xi > 1$ est :

$$y(k) = KA \left(1 + \frac{(-\xi - \sqrt{\xi^2 - 1}) \cdot e^{\omega_0(-\xi + \sqrt{\xi^2 - 1})kT_e} - (-\xi + \sqrt{\xi^2 - 1}) \cdot e^{\omega_0(-\xi - \sqrt{\xi^2 - 1})kT_e}}{2\sqrt{\xi^2 - 1}} \right) \quad (\text{III.5})$$

- ✓ Le signal A en bleu est le signal de sortie (mesure) qui est la réponse indicielle du système de deuxième ordre avec un gain statique $K=1$, de pulsation propre $\omega_0=2$ et coefficient d'amortissement $\xi=2$ subi à une perturbation de 2V.
- ✓ Le signal B en jaune est le signal de commande fourni par le régulateur d'un $K_p=0.8$, $T_i=0.6$, $T_d=0.4$ et $N=0.005$.
- ✓ La consigne C=3.5V.

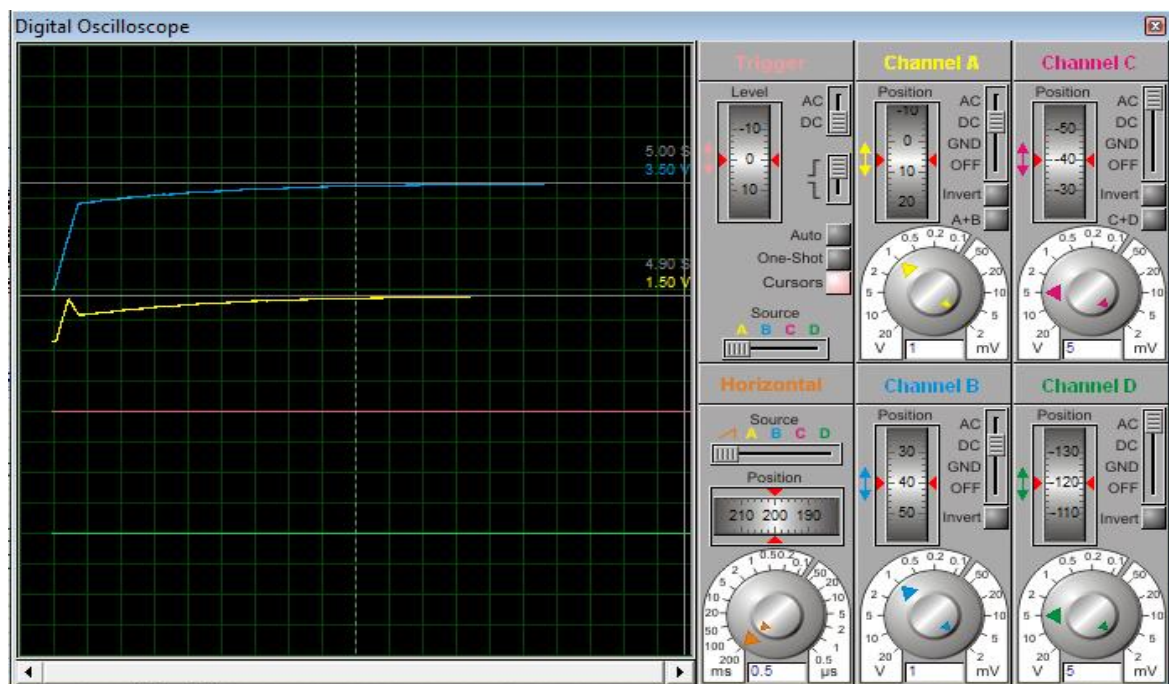


Figure III.12 : Signaux de commande et de mesure d'une régulation PID d'un système 2^{em} ordre $\xi > 1$.

Interprétation

D'après le signal obtenu de l'oscilloscope nous pouvons conclure par dire que la régulation s'est déroulée correctement en comparant ces deux signaux à ceux de MATLAB.

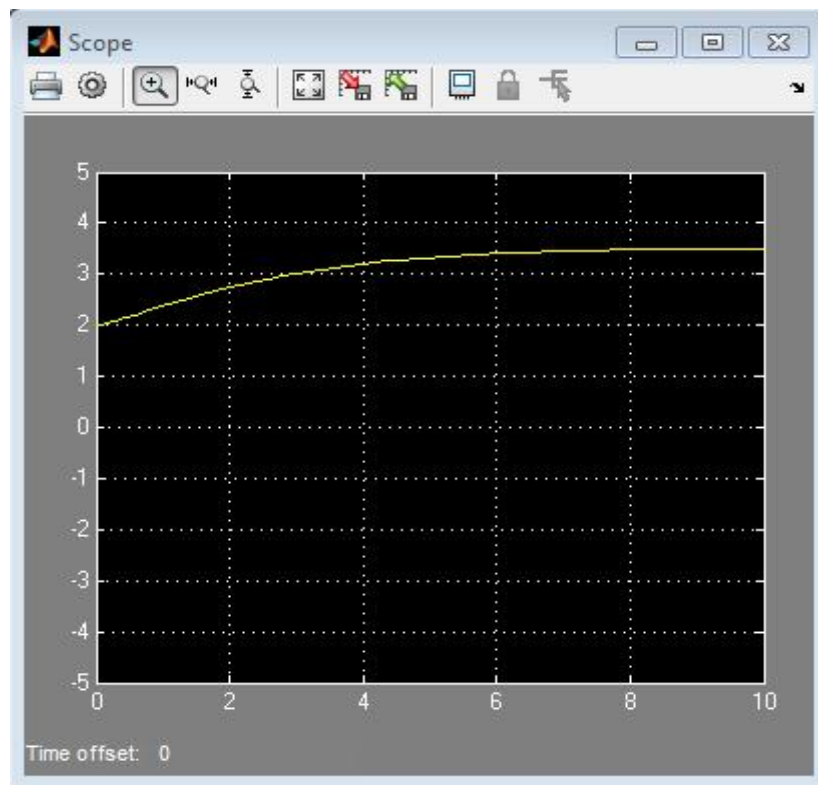


Figure III.13 : signal de mesure $\xi > 1$.

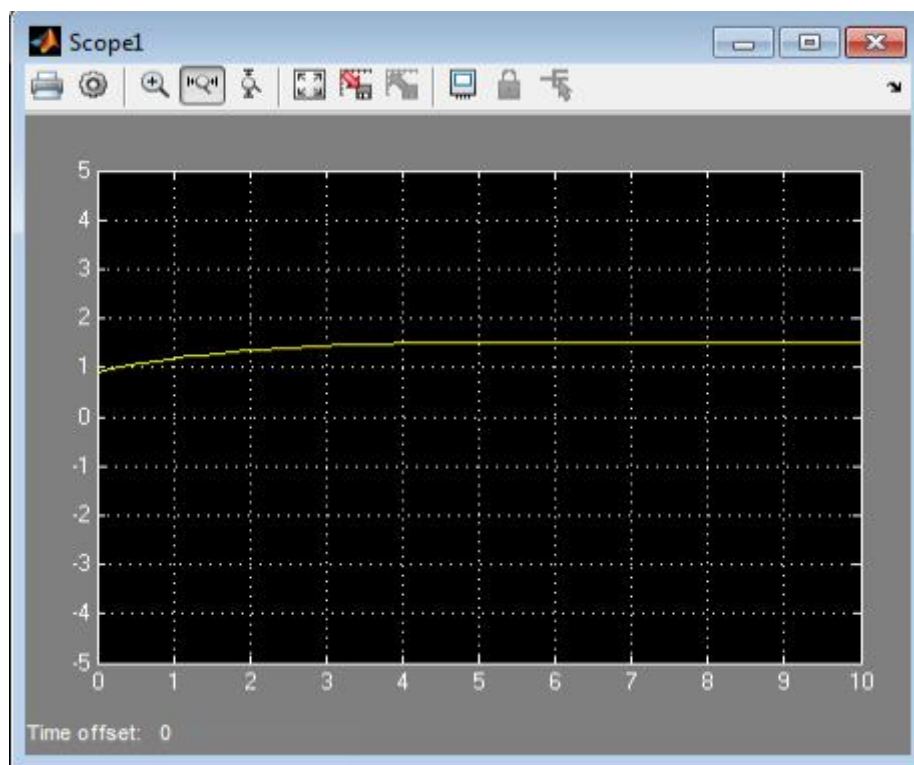


Figure III.14 : signal de commande $\xi > 1$.

➤ **Cas de $\xi = 1$:**

L'équation programmée qui caractérise le système de deuxième ordre avec $\xi = 1$ est :

$$y(k) = KA(1 - (1 + \omega_0 \cdot k \cdot Te) \cdot e^{-\omega_0 \cdot k \cdot Te}) \quad (\text{III.6})$$

- ✓ Le signal A en bleu est le signal de sortie (mesure) qui est la réponse indicielle du système de deuxième ordre avec un gain statique $K=1$, de pulsation propre $\omega_0=2$ et coefficient d'amortissement $\xi=1$ subi à une perturbation de 2V.
- ✓ Le signal B en jaune est le signal de commande fourni par le régulateur d'un $K_p=0.8$, $T_i=0.6$, $T_d=0.4$ et $N=0.005$.
- ✓ La consigne C=3.5V.

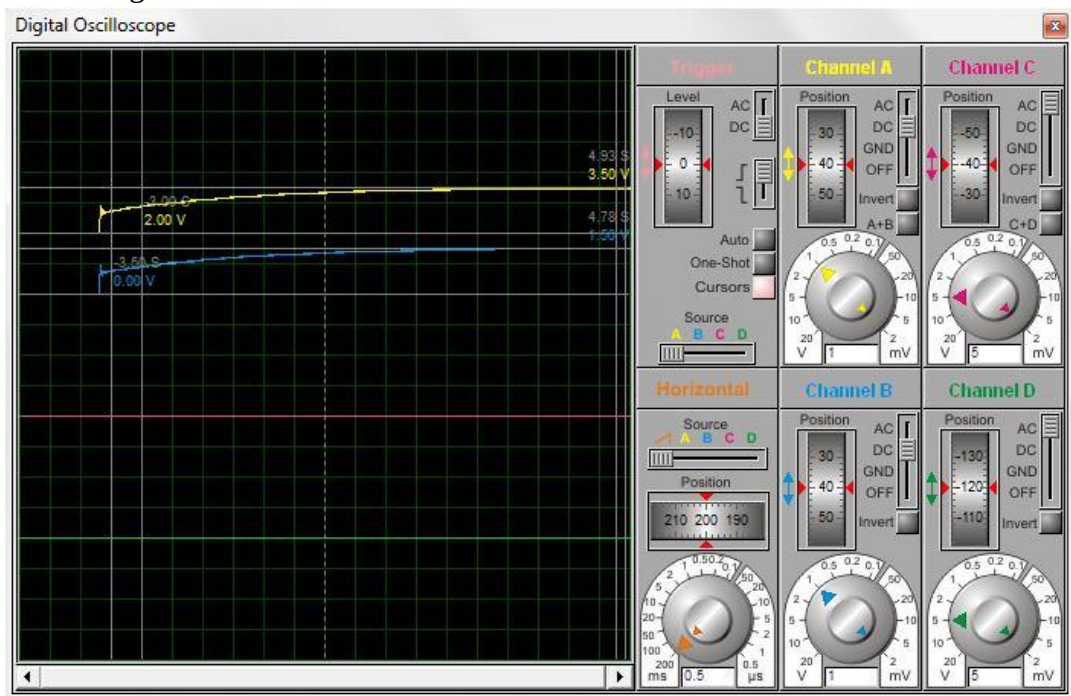


Figure III.15 : Signaux de commande et de mesure d'une régulation PID d'un système 2^{em} ordre $\xi=1$

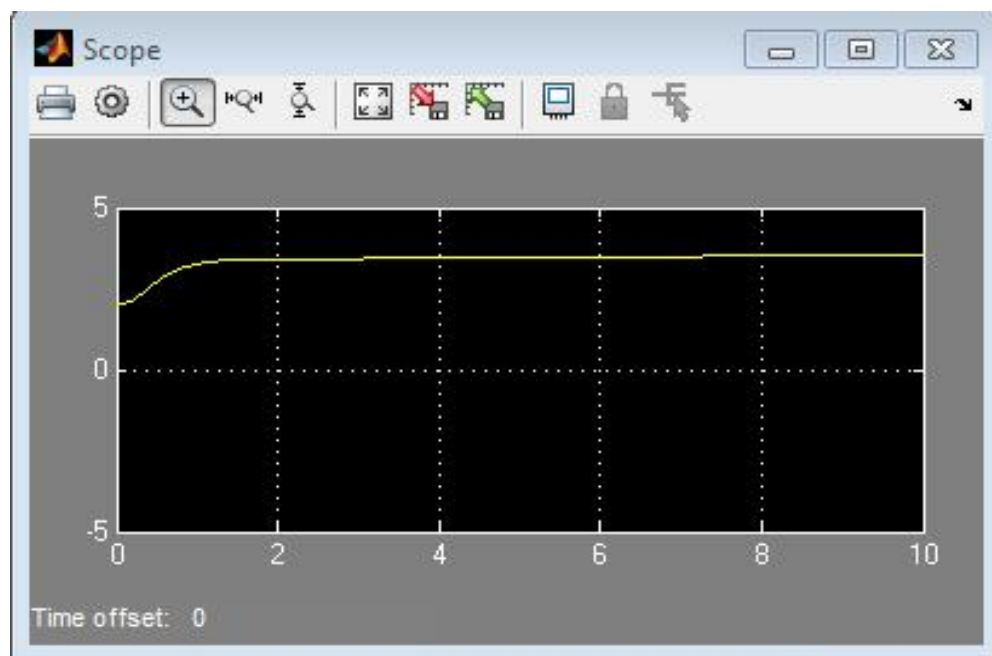


Figure III.16 : signale de mesure $\xi=1$.

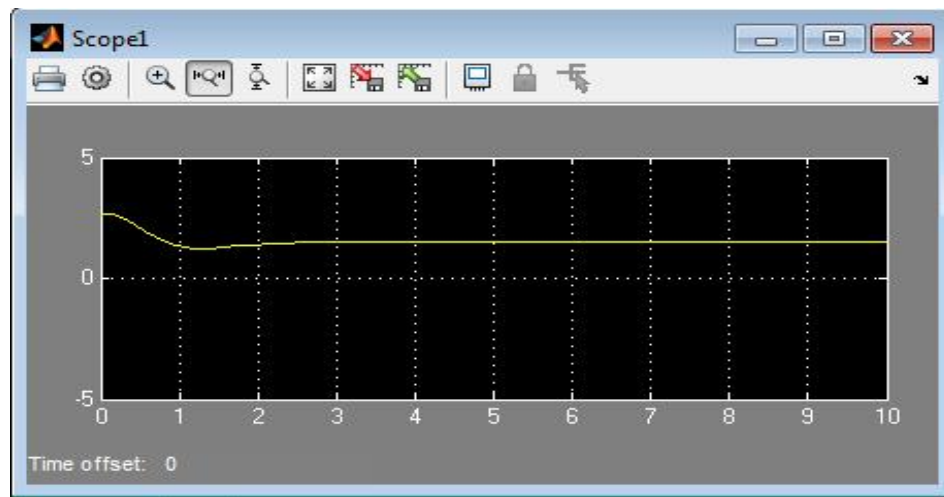


Figure III.17 : signal de commande $\xi = 1$.

➤ Cas de $\xi < 1$:

L'équation programmée qui caractérise le système de deuxième ordre avec $\xi < 1$ est :

- ✓ Le signal A en bleu est le signal de sortie (mesure) qui est la réponse indicielle du système de deuxième ordre avec un gain statique $K=1$, de pulsation propre $\omega_0=2$ et coefficient d'amortissement $\xi = 0.7$ subi à une perturbation de 2V.
- ✓ Le signal B en jaune est le signal de commande fourni par le régulateur d'un $K_p=0.5$, $T_i=0.7$, $T_d=0.1$ et $N=2.2$.
- ✓ La consigne C=3.5V.

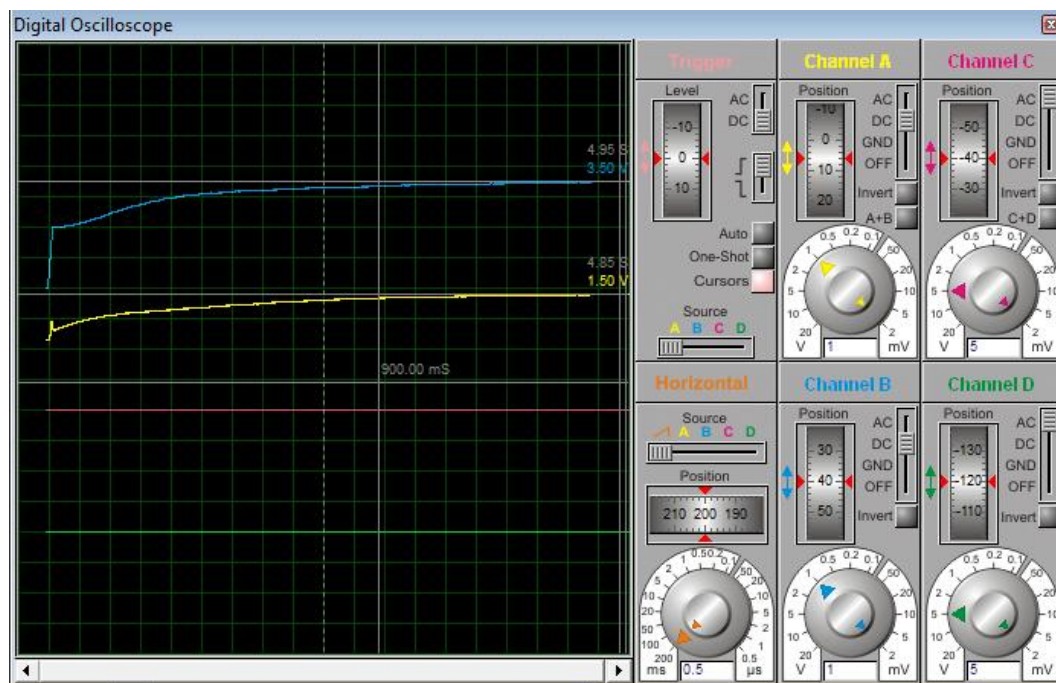


Figure III.18 : Signaux de commande et de mesure d'une régulation PID d'un système 2^{ème} ordre $\xi < 1$.

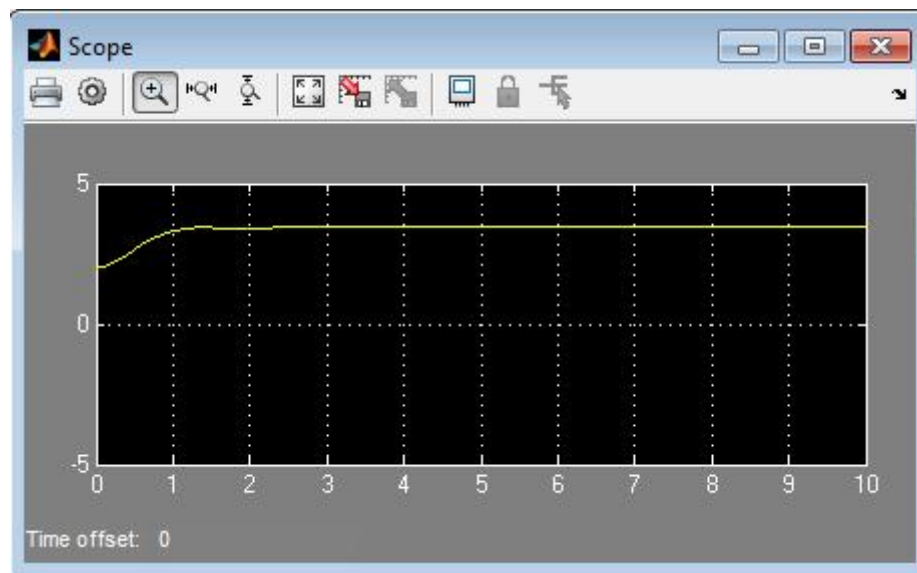


Figure III.19 : signal de mesure $\xi < 1$.

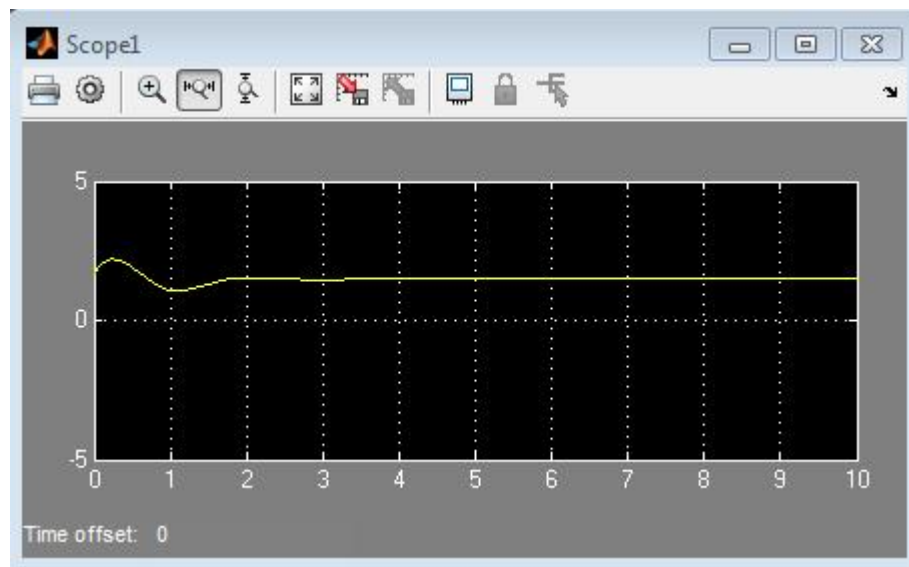


Figure III.20 : signal de commande $\xi < 1$.

Conclusion

Nous avons pu voir à travers ce chapitre l'application de concepts théoriques de régulation des systèmes par un régulateur PID numérique d'une structure mixte tout en obtenant des résultats satisfaisants et ceci à l'aide des outils de programmation offerts par le mikroC ainsi que la supervision des processus à travers l'interfaçage graphique réalisé à l'aide de ISIS.

Conclusion générale

Conclusion générale

Avec ce projet nous avons certes consolidé nos acquis théoriques dans le domaine de la régulation mais nous avons entrouvert une nouvelle porte vers un autre domaine aussi passionnant qui est la réalisation pratique de nos études théoriques.

Nous avons acquis durant la réalisation de notre projet d'innombrables et précieuses connaissances dans divers domaines aussi variés telle que la régulation, l'électronique, la programmation....etc.

Ainsi on a opté pour l'étude dans notre premier chapitre les différentes notions et principes de la régulation qui nous permettent de mettre au point une régulation adaptée au système que nous voulons réguler et ainsi choisir le type de régulateur adéquat

Puis dans le deuxième chapitre on a opté pour l'étude des microcontrôleurs de type PIC en détaillant ses différentes caractéristiques, on s'est concentré plus précisément sur le PIC16f877A et cela pour mettre en évidence les raisons qui poussent notre choix à se porter sur ce PIC. On s'est intéressé aussi à la circuiterie et la conception software de notre boucle de régulation, tout d'abord la conception du circuit dans laquelle nous avons détaillé chaque bloc composant notre carte électronique ensuite la conception software dans laquelle nous avons présenté la manière avec laquelle nous avons développé le programme qu'on a implémenté dans notre microcontrôleur.

Enfin dans le troisième chapitre nous avons simulé le fonctionnement de chaque bloc de notre boucle de régulation.

Notre confrontation a divers difficultés pratiques puis en les résolvant nous a permis d'apprendre à porter un autre angle de réflexion sur la régulation des systèmes tout en prenant en compte les exigences liées à la réalisation pratique.

Notre incursion dans le domaine de la réalisation pratique nous fait prendre conscience du vaste domaine dans lequel nous nous sommes initié et de l'ampleur du travail qu'il nous reste à faire pour maîtriser au moins une petite partie de ce domaine, néanmoins nous espérons qu'avec notre modeste travail nous avons fait un pas vers la maîtrise de quelque subtilité de ce vaste domaine.

Notre réalisation comme tous travail humain est susceptible d'être amélioré et développer, et c'est pour cela nous proposons certaines améliorations envisagées tel que :

- Implémenter dans le PIC d'autre structure PID (Parallèle et sérié) ainsi que la structure RST.
- Le doté de l'option auto-tunning c.-à-d. auto identification des paramètres PID.
- Le mettre en communication avec le système réel qu'on veut réguler.

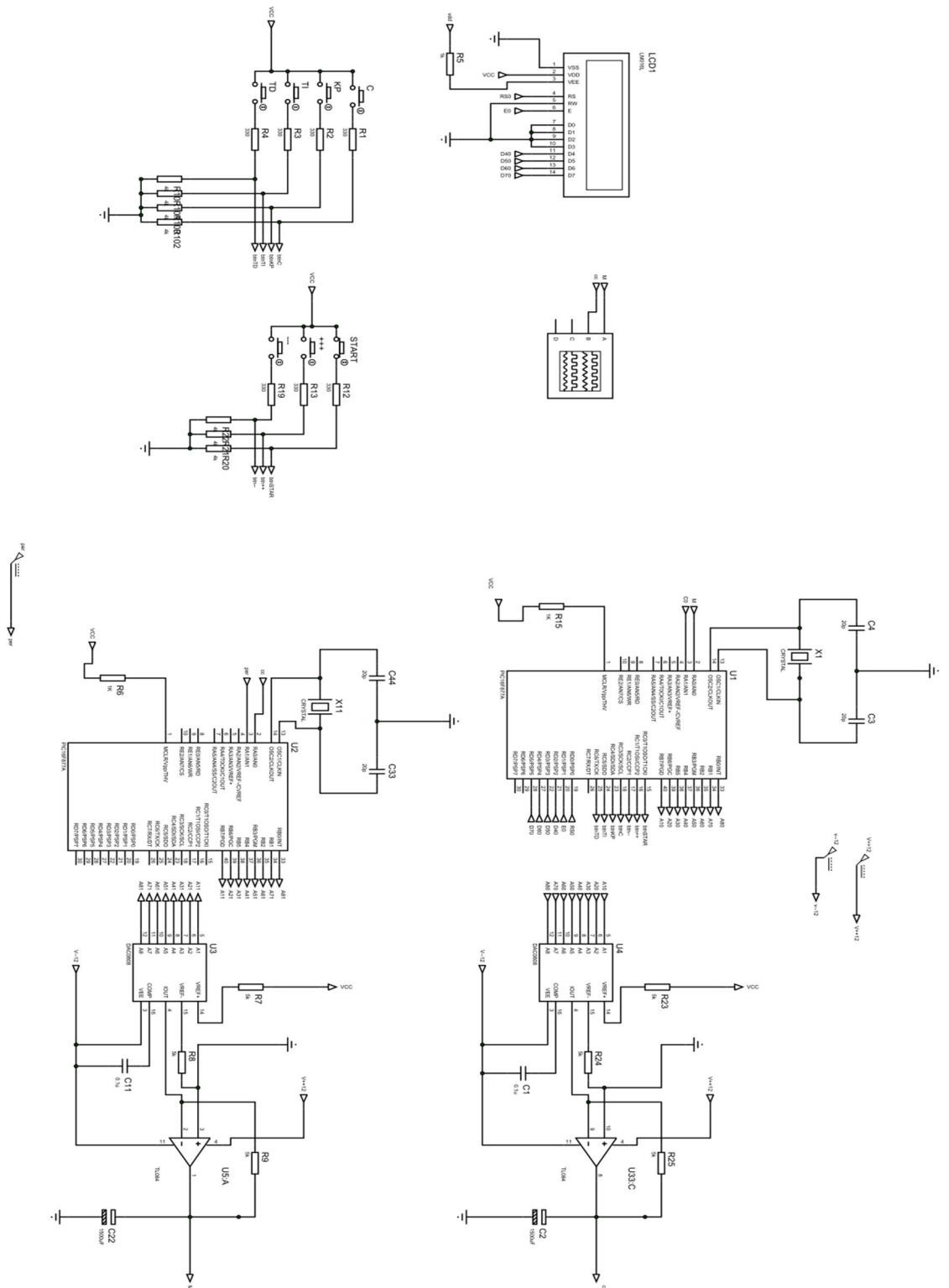
Bibliographie

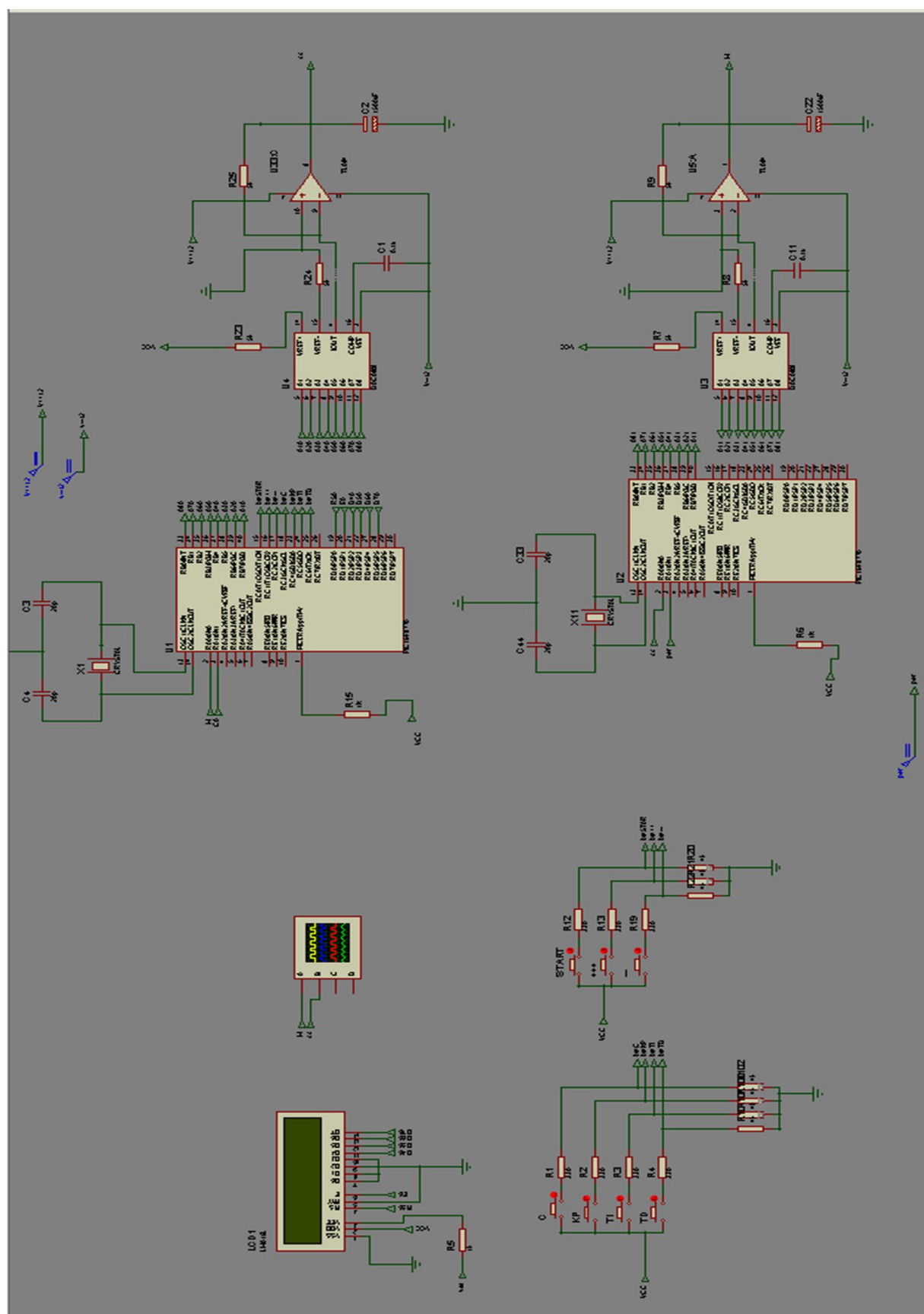
Bibliographie

- [1] : Cours d'Automatique 2005-2006
Bernard Bayle
Ecole Nationale supérieure de Physique de Strasbourg
- [2] BELKACEM OULD-BOUAMAMA
Régulation automatique (le 15 juin 1998)
www.eudil.fr/eudil/belk/sc00a.htm
- [3] Régulation industrielle –Notion de Base
Article PDF
- [4] R.RHODE ENSEMBLE SCOLAIRE PRADEAU LA SEDE
Régulation analogique
Article PDF
- [5] ALINA BESACON- VODA et SYLVIANNE GENTIL
Régulateurs PID analogique et numérique
Technique de l'ingénieur : R7416 (03/1999)
- [6] ERIC MAGAROTTO
Cour de régulation (version septembre 2004)
Article PDF
- [7] <http://pagesperso-orange.fr/fabrice.sincere/>
- [8] M.ROBERT
Introduction au microcontrôleur
- [9] <http://freeelektronik.free.fr/LEKTRONIK/C9A.htm>
- [10] Data sheet Microchip
PIC16F87X28/40-pin 8-Bit CMOS FLASH Microcontrollers
-

Annexe

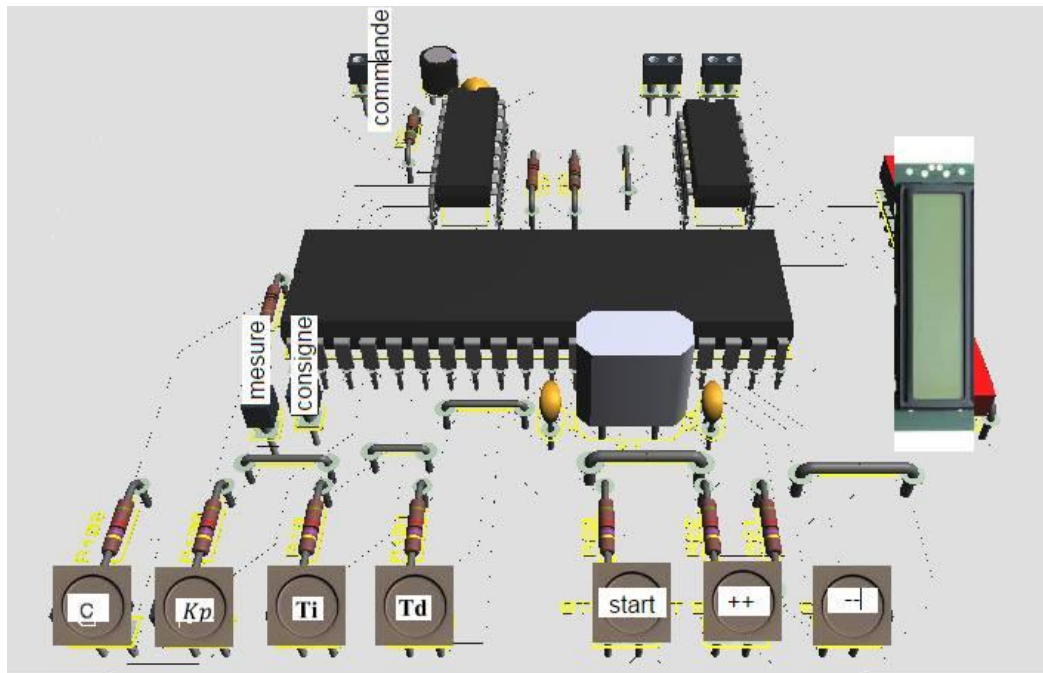
Le schéma électrique de notre carte



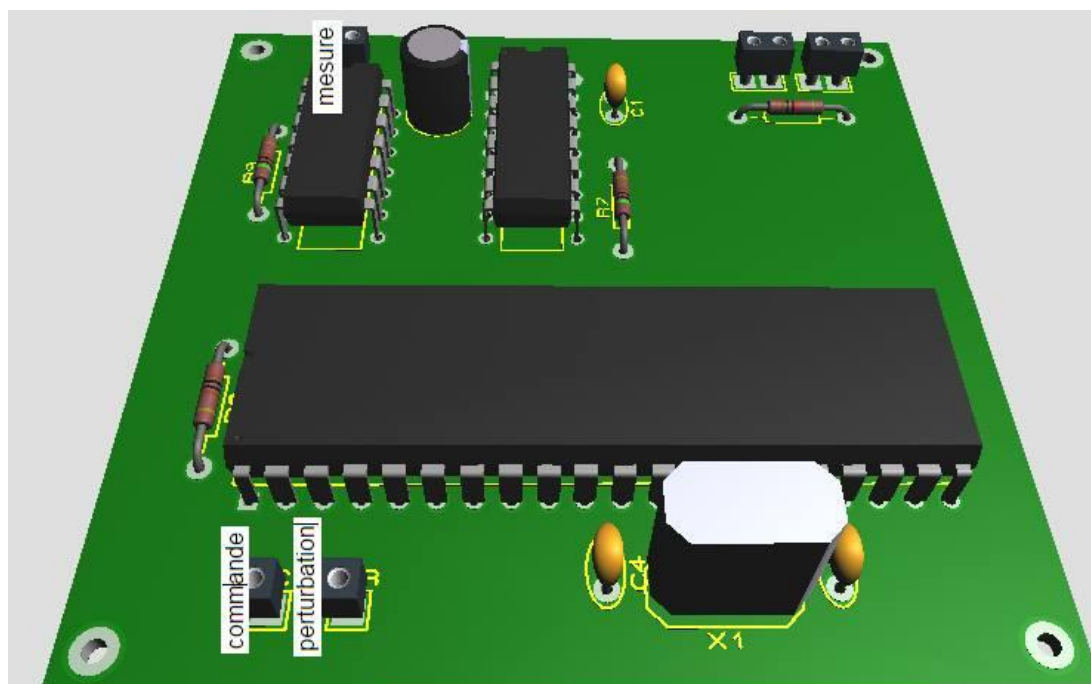


Vue en trois dimensions de notre carte

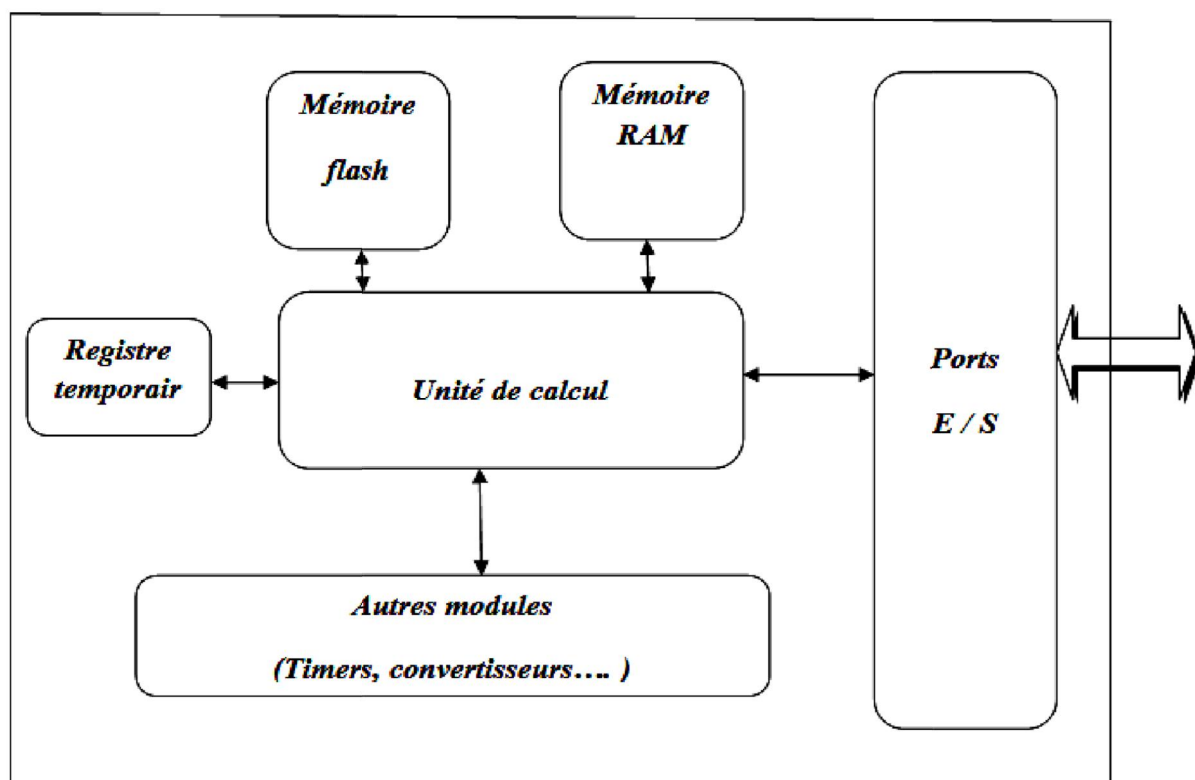
Parti régulateur



Parti système



Structure interne d'un microcontrôleur



Principe de fonctionnement de l'unité centrale

1. Le PC (Compteur Programme) registre de 13 bits contient l'adresse de la prochaine instruction à exécuter dans la mémoire programme.

Le PC peut être stocké, si nécessaire, dans une pile qui peut contenir 8 valeurs aux maximum (8 niveaux).

Au début du programme qui commence à l'adresse 0, PC=0.

2. La mémoire programme contient l'ensemble des instructions du programme.

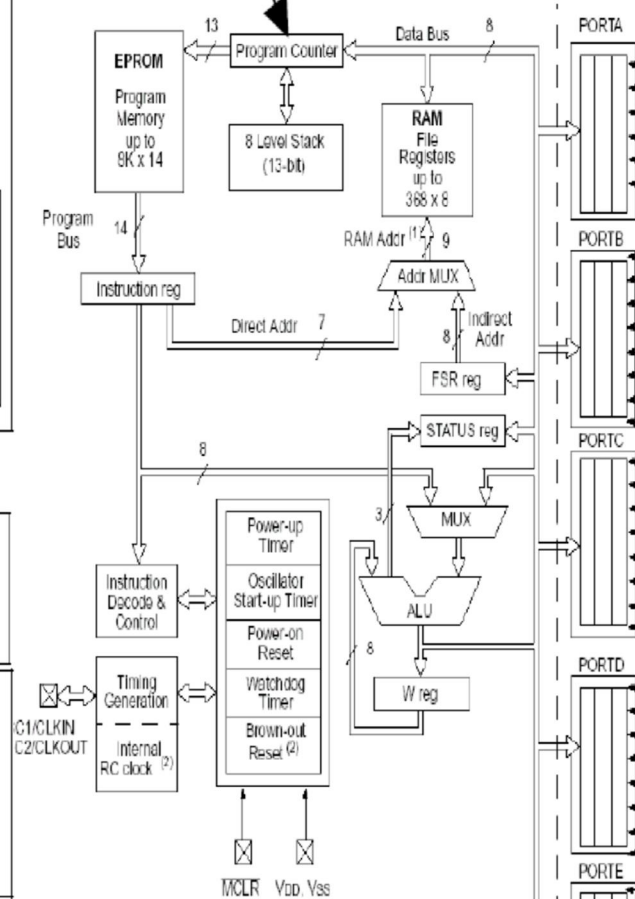
Par exemple en adressage direct une instruction se compose d'un mot de 14 bits qui a la structure suivante :

Opération
à exécuter
sur 7 bits

Opérande de 7
bits qui est
l'adresse de la
donnée dans la
RAM.

3. Le registre d'instruction permet de mémoriser la prochaine instruction qui sera exécutée.

4.A. L'opération va être décodée pour pouvoir indiquer à l'ALU les calculs qui devront être exécutés.



4.B. L'opérande fournit l'adresse qui va permettre d'accéder à la donnée dans la RAM.

5. La donnée et l'opération décodée arrivent simultanément dans un **multiplexeur** (structure de Harvard).

6.A. L'ALU (Unité Arithmétique et Logique) va exécuter les calculs ou tests logiques requis par l'instruction.

7.A. L'ALU pour effectuer ses opérations a besoin d'une variable, ici un registre de travail **W** (Working register).

Schéma fonctionnel du 16f877A

