République Algérienne Démocratique et Populaire
Ministre de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Bejaia
Faculté de Technologie
Département Automatique Télécommunication et Électronique

Mémoire de fin de cycle

En vue de l'obtention du diplôme de Master en Automatique

Thème

Implémentation d'un contrôleur PID sur un dsPIC 33F

Réalisé par :

- ***Kaci samir***
- ***Youcef khodja lakhdar***

Encadré par :

M^{me} : S.MEZZAH

Promotion 2014-2015

Remerciement

Nos remerciements vont tout d'abord au Dieu le tout puissant pour la santé et la patience qu'il nous a donné.

Le travail présenté dans ce manuscrit a été effectué dans le cadre du projet de Master, sou la direction de Mme Mezzah, au quelle nous tenons à adresser nos plus vifs remerciements pour d'avoir aidé à diriger cette étude.

Nos vifs remerciements au membre de jurys de bien vouloir accepter d'évaluer notre travail.

Et enfin, à toutes les personnes qui ont contribuées, de près ou de loin à la réalisation de ce modeste travail sans oublier Mr C.Mourad

DÉDICACES

Je dédie ce mémoire à :

A mes chers parents pour leur dévouement et leur immense sacrifice, dans eux il m'aurait été difficile de mener à bien mes études.

A mes grand parents pour leurs générosité et leur soutient moral au cours de mes études sans oublier ma tante ZAHOUA.

A mes chers frères et sœurs.

A mes beaux- frères et mes chers neveux YANIS et AGHILAS.

A tous les miens qui me sont très chers.

A tous mes amis qui ont bien voulu m'apporter leur soutien moral en particulier.

A tous les enseignants qui m'ont aidé de proche ou de loin pour avoir un jour mon diplôme de master.

A toutes les personnes que je n'ai pas pu cité j'adresse toute ma reconnaissance et mon indéfectible attachement.

SAMIR

Dédicaces

Aux être les plus chères, mes parents, mes frères et ma sœur qui ont étaient à mes côtés et m'ont toujours soutenu tout au long de ces langues années d'études. En signe de reconnaissance, qu'ils se trouvent ici, l'expression de ma profonde gratitude pour tout ce qu'ils ont consenti d'efforts et de moyens pour me voir réussir dans mes études.

A tous mes amis avec qui j'ai partagé beaucoup de moments inoubliables en particulier mon chère KIKI.

Lakhdar

Sommaire

Chapitre III : programmation et simulation

Liste des figures

Liste des tableaux

INTRODUCTION GENERALE

Introduction générale

L'automatique est à la fois une science et une technique. Son rôle est la maitrise du comportement d'un système (traduit par ses grandeurs de sortie) en agissant de manière adéquate sur ses grandeurs d'entrée. L'automatique est une discipline qui traite également de la modélisation, la simulation, l'analyse, la commande et la régulation des systèmes dynamiques.

La régulation est primordiale dans le domaine industriel. Il nous a donc paru incontournable de réaliser un régulateur qui puisse nous permettre d'appliquer les principes de régulation acquis tout au long de nos études, et ce, non seulement pour consolider nos acquis théoriques, mais aussi pour découvrir d'autres aspects pratiques liés à la régulation. Plus précisément, notre étude a porté sur le contrôle par PID de la température d'un four électrique.

Notre choix pour la réalisation de ce régulateur s'est porté sur le microcontrôleur ***dsPIC.*** La raison en est sa simplicité de programmation et le large éventail de domaines d'applications qu'il offre, mais aussi son faible coût économique. Les dsPIC offrent la possiblité de manipuler des composants de hautes performances avec la même simplicité d'utilisation des microcontrôleurs.

La démarche suivie dans l'élaboration de notre étude commence par une délimitation concise des contenus des différentes parties constitutives de ce mémoire. Au premier chapitre, vient un aperçu des différentes notions relatives à la régulation par ***PID***. Le chapitre suivant aborde une description générale des caractéristiques de la famille de dsPIC30/33 de Microchip. Au troisième et dernier chapitre, sera présentée la programmation et la simulation du programme avec le logiciel ***Proteus-Isis***.

Une conclusion générale scelle les bouts des grandes lignes de notre étude par un autre regard rétrospectif à fin de récapitulation.

Chapitre I

Le contrôleur PID : étude et simulation

I.1 Introduction

Le contrôleur **PID** (Proportionnel, intégral et dérivé) est une méthode de régulation souvent employée pour les asservissements dans différents processus industriels. Ces paramètres ont besoin d'être ajustés en fonction du procédé de commande, tout comme ils doivent rester sans changement pendant son activité régulière.

Pas toujours simple, le démarrage du contrôleur PID exige un minutieux travail d'ajustement de ces paramètres, sans compter l'existence de quelques méthodes, telles que celle de ASTROM K. J. et Alexandre DRIVE. Le principe de ces méthodes consiste dans le calcul approché pour les valeurs des paramètres. Cependant, pour les calculer, une période d'observation sera nécessaire, avec une plus grande certitude d'exécution du contrôleur.

Il existe d'autres cas, plus complexes, en raison de leurs particularités, où il y a de petits changements de procédure compromettant l'exécution du contrôleur PID. Ces situations sont observées par les cartes de tendance analysées par des opérateurs de processus, amenant la nécessité d'un rajustement de paramètres de contrôleur. Il est difficile d'en définir la raison, encore moins l'expliquer, étant donné que souvent, les aspects des procédures s'interprètent de diverses manières.

I.2 Contrôle par PID [1]

Un contrôleur, ou régulateur, est un instrument utilisé en conjonction avec d'autres dispositifs pour contrôler automatiquement une variable de procédé à un point de consigne prédéterminé.

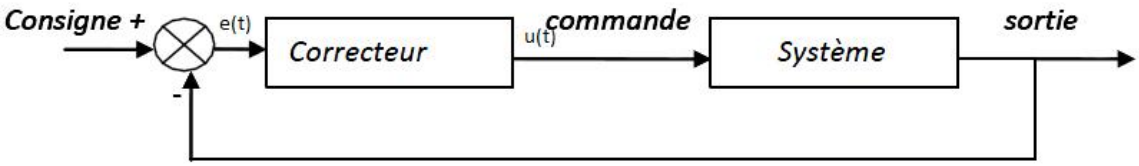


Figure I.1. Représentation générale du système

Le contrôleur compare la valeur de la variable du procédé avec la valeur de la consigne et délivre un signal d'erreur e(t), puis produit un signal de sortie (la commande u(t)) qui est une fonction du signal d'erreur d'entrée. La relation entre le signal de sortie du contrôleur et le signal d'entrée du contrôleur définit le mode de contrôle.

Les modes de contrôle couramment utilisés ou communs sont les suivants :

1- Contrôleur de type proportionnel ;

2- Contrôleur de type intégral ;

3- Contrôleur de type dérivé.

À ce titre, le régulateur le plus complet est donc le régulateur PID. Ses initiales signifient *Proportionnel*,*IntégraletDérivé* et indiquent que toutes ces actions sont présentes dans ce régulateur. À partir de ce régulateur, on a des régulateurs plus spécifiques, le régulateur PI (Proportionnel Intégral), le régulateur PD (Proportionnel Dérivé) ou simplement le régulateur P (Proportionnel).

Il existe plusieurs architectures possibles de combinaison des trois effets (série, parallèle ou mixte). Ici on présente une architecture parallèle :

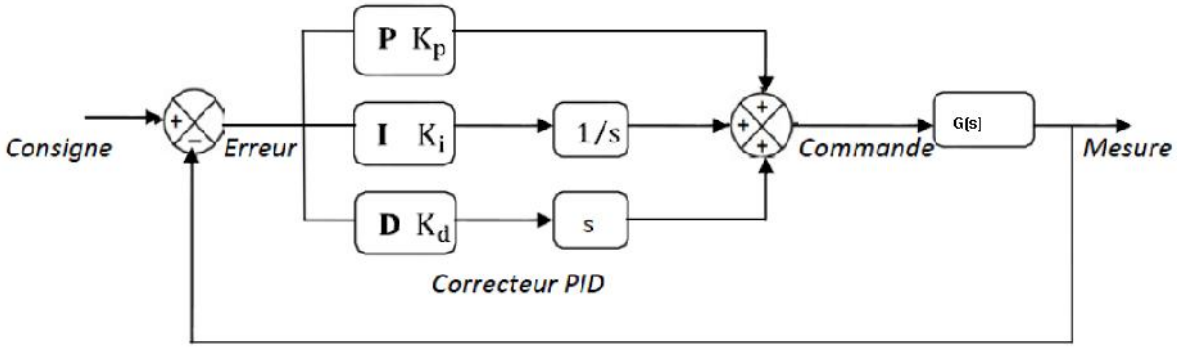


Figure I.2. Asservissement par correcteur PID

I.3 Principe général d'un correcteur PID [2]

L'erreur observée est la différence entre la consigne et la mesure. Le PID permet trois actions en fonction de cette erreur :

- Une action **Proportionnelle** : l'erreur est multipliée par un gain $\mathbf{K_p}$;
- Une action **Intégrale** : l'erreur est intégrée sur un intervalle de temps ***s***, puis multipliée par un gain K_i ;
- Une action **Dérivée** : l'erreur est drivée suivant un temps **s**, puis multipliée par un gain K_d. Les actions dérivées et intégrales ne s'emploient jamais seules mais en combinaison avec l'action proportionnelle.

I.3.1 Synthèse sur les actions PID

Sous forme de tableau récapitulatif, on résume les avantages et les limitations des actions de base des régulateurs PID :

Action	Points forts	Points faibles
P	Action instantanée	Ne permet pas d'annuler une erreur statique mais permet de la réduire
I	Annule l'erreur statique	Action lente Ralentit le système (effet déstabilisant)
D	Action très dynamique Améliore la rapidité Apporte un effet stabilisant	Sensibilité aux bruits Forte sollicitation de l'organe de commande

Tableau I.1 Avantages et inconvénients des actions du PID

I.4 Réglage d'un PID

Le réglage d'un PID consiste à déterminer les coefficients Kp, Ki et kd afin d'obtenir une réponse adéquate du procédé et de sa régulation. L'objectif en est d'être **stable**, **rapide** et **précis**. Pour cela, il faut limiter tout **dépassement** éventuel comme il est expliqué (*Cf.* figure I.3).

- **La stabilité,** doit toujours converger vers un point d'équilibre stable, et ne doit pas osciller autour du point de consigne.
- **La rapidité** d'un système asymptotiquement stable se mesure par la durée de son régime transitoire. Il est pratique de la mesurer grâce à la réponse indicielle. Pour un système asymptotiquement stable, plus la réponse indicielle converge rapidement vers la valeur finale, plus le système est rapide.
- **La précision** en régime établi, la grandeur régulée, doit être maintenue en permanence au plus près de la consigne.

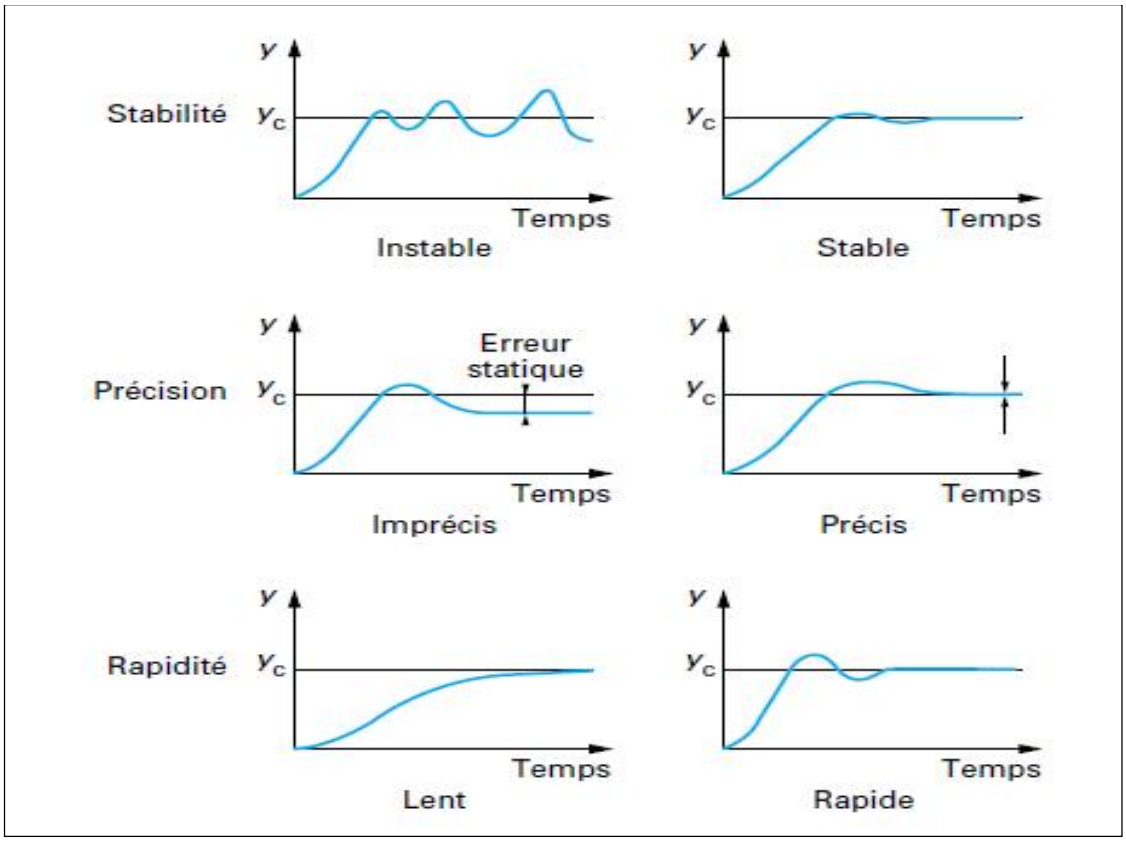


Figure I.3. Qualités de la régulation

I.5 Influence des paramètres du PID sur le système [2]

Les paramètres du PID influent sur la réponse du système de la manière suivante :

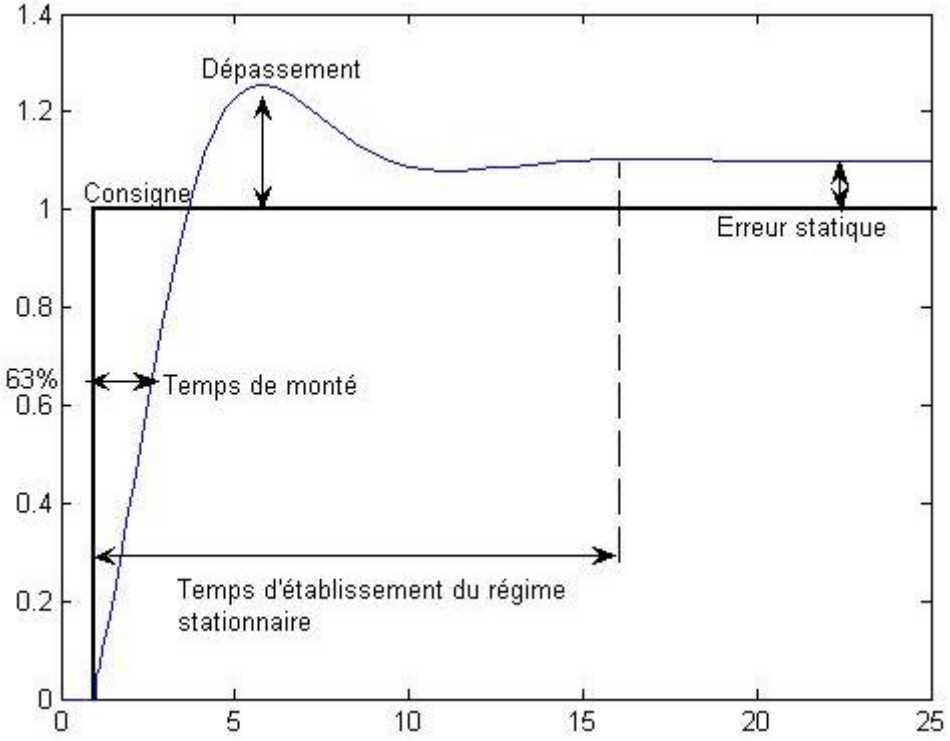


Figure I.4. Réponse d'un système de second ordre

I.5.1 L'influence de Kp

Lorsque Kp augmente, le temps de montée (*Rize time*) est plus court mais il y a un dépassement plus important. Le temps d'établissement du régime varie peu et l'erreur statique se trouve diminue.

I.5.2 L'influence de Ki

Lorsque *Ki* augmente, le temps de montée est plus court ; cependant on y constate un dépassement plus important. Le temps d'établissement au régime stationnaire s'allonge, mais dans ce cas on assure une erreur statique réduite. Donc plus ce paramètre est élevé, moins l'erreur statique est grande, moins prompte sera la réponse.

I.5.3 L'influence de Kd

Lorsque Kd augmente, le temps de montée varie peu mais le dépassement diminue. Le temps d'établissement au régime stationnaire est meilleur. Aucune influence n'est apparue sur l'erreur statique. Si ce paramètre est trop élevé, le système anticipe trop et la consigne n'est pas atteinte dans les délais appropriés. Pour ces trois paramètres, le réglage au-delà

d'un seuil trop élevé a pour effet d'engendrer une oscillation du système de plus en plus importante, menaçant d'instabilité.

I.6 Aspects fonctionnels du contrôleur PID

La réalisation d'une boucle d'asservissement par un PID comporte deux aspects essentiels

- Le réglage du régulateur PID, pour lequel la connaissance d'un modèle dynamique du procédé d'une part, et les performances désirées, d'autre part, déterminent le choix de la méthode de synthèse ;
- L'implantation du régulateur dans une version analogique ou numérique et dans une configuration série, parallèle ou mixte.

I.6.1 Contrôleur proportionnel(P)

Le contrôleur P est simple n'est qu'un gain K_p, comme expliqué à la figure suivante :

Figure I.5. Contrôleur proportionnel

Dans le cas d'un contrôle proportionnel, l'erreur est virtuellement amplifiée d'un certain gain constant qu'il conviendra de déterminer en fonction du système.

$$u(t) = \text{[illegible]} \qquad (01)$$

Ce qui, dans la méthode de Laplace, donne :

$$U(p) = \text{[illegible]} \qquad (02)$$

L'idée étant d'augmenter l'effet de l'erreur sur le système afin que celui-ci réagisse plus rapidement aux changements de consigne. Plus la valeur de $\mathbf{K_p}$ est grande, plus la réponse est moins. En revanche, la stabilité du système s'en trouve détériorée. Dans le cas d'un $\mathbf{K_p}$ démesuré, le système peut même diverger. Donc un système avec un contrôleur **P** ressemblerait au système rapporté ci-après :

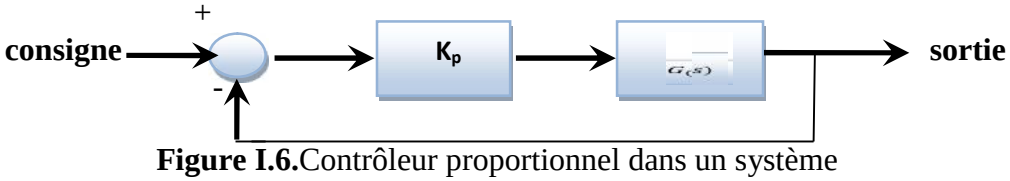


Figure I.6. Contrôleur proportionnel dans un système

I.6.2 Contrôleur intégrale(I)

Au contrôle proportionnel, il peut être ajouté l'intégration de l'erreur. Dans ce cas, nous obtenons une régulation **PI** (Proportionnelle et intégrée). L'erreur entre la consigne et la mesure est ici intégrée par rapport au temps et multipliée par une constante qu'il faudra aussi déterminer en fonction du système.

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)\,d\tau \qquad (03)$$

Ce qui, suivant la méthode de Laplace, donne :

$$U(p) = K_p \cdot E(p) + K_i \cdot \frac{E(p)}{p} \qquad (04)$$

Pourquoi a-t'on besoin de rajouter cette fonctionnalité à notre organe de contrôle ? C'est pour la simple raison qu'en contrôle proportionnel, il subsiste une erreur statique. Lorsque le système s'approche de sa consigne, l'erreur n'est plus assez importante pour faire avancer le système. Le terme intégral permet ainsi de compenser l'erreur statique et fournit, par conséquent, un système plus stable en régime permanent. Plus K_i est élevé, plus l'erreur statique se corrige.

I.6.3 Contrôleur dérivé(D)

Pour obtenir un contrôle en PID, il nous faut encore rajouter un terme. Il consiste à dériver l'erreur entre la consigne et la mesure par rapport au temps et à le multiplier lui aussi par une constante.

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)\,d\tau + K_d \cdot \frac{d}{dt} e(t) \qquad (05)$$

Ce qui, dans la méthode de Laplace, donne :

$$U(p) = K_p \cdot E(p) + K_i \cdot \frac{E(p)}{p} + K_d \cdot p \cdot E(p) = E(p) \cdot \left[K_p + K_i \cdot \frac{1}{p} + K_d \cdot p\right] \qquad (06)$$

Pourquoi est-il besoin d'avoir un terme dérivé ? La raison en est que le contrôle **PI** peut amener à un dépassement de la consigne, ce qui n'est pas toujours très souhaitable

(exemple d'inversion de polarité dans le cas de moteurs électriques). Le terme dérivé permet justement de limiter cela. Lorsque le système s'approche de la consigne, ce terme freine le système en y appliquant une action dans le sens opposé permettant ainsi une stabilisation plus immédiate.

I.7 Méthodes de réglage des actions

Il existe différentes méthodes de réglage des actions d'un régulateur **PID**. Quelques méthodes assez connues sont illustrées ci-après.

I.7.1 Méthode de Ziegler et Nichols

Elle nécessite l'observation de la réponse du procédé et la connaissance de la structure du régulateur. Son avantage est d'être une méthode pouvant calculer des actions sans passer nécessairement passer par la détermination des paramètres du procédé.

I.7.2 Méthode par approches successives

Elle consiste à modifier les actions du régulateur et à observer les effets sur la mesure enregistrée, jusqu'à obtenir la réponse optimale. On règle l'action d'abord proportionnelle, puis l'action dérivée et l'intégrale.

Cette technique s'illustre par son intérêt et sa simplicité, utilisable sur n'importe quel type de système. Néanmoins, du fait de son caractère itératif, son application devient longue sur des procédés à grande inertie.

I.7.3 Méthode nécessitant identification de procédé

Si l'on connaît les paramètres du procédé, suite à une modélisation de sa fonction de transfert réglable, et si l'on est en possession de la structure du régulateur, il est alors possible de calculer rapidement les paramètres de réglage qu'on pourra affiner**,** suite à des essais, afin d'obtenir la réponse souhaitée.

Ce que nécessite cette méthode, c'est un enregistreur à déroulement rapide. Elle est de préférence utilisée sur des procédés à grande inertie.

I.8 Exemple de simulation sur *Matlab/Simulink*

On a illustré (*supra,* titre I.5) l'influence des paramètres du **PID** tels que **Kp**, **Ki** et **Kd** sur les systèmes asservis. Néanmoins, nous allons simuler la fonction **G(s)**d'un système de 1er ordre avec *Simulink* pour faire ressortir la variation des paramètres du PID.

$$G(s) = \frac{4}{8s+1} \qquad (07)$$

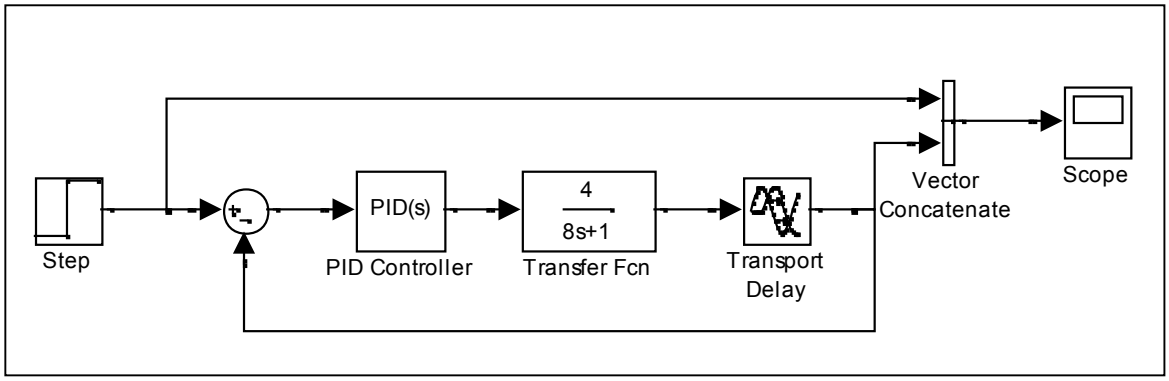


Figure I.7. Simulation pour la fonction de transfert G(s) avec le contrôleur PID

I.8.1 Cas de contrôleur PID manuel

Les paramètres de **PID** s'introduisent de sorte que : Kp=0,576, Ki=0,0653 et Kd=0,3388. Les résultats de la simulation obtenue sont rapportés dans la figure suivante :

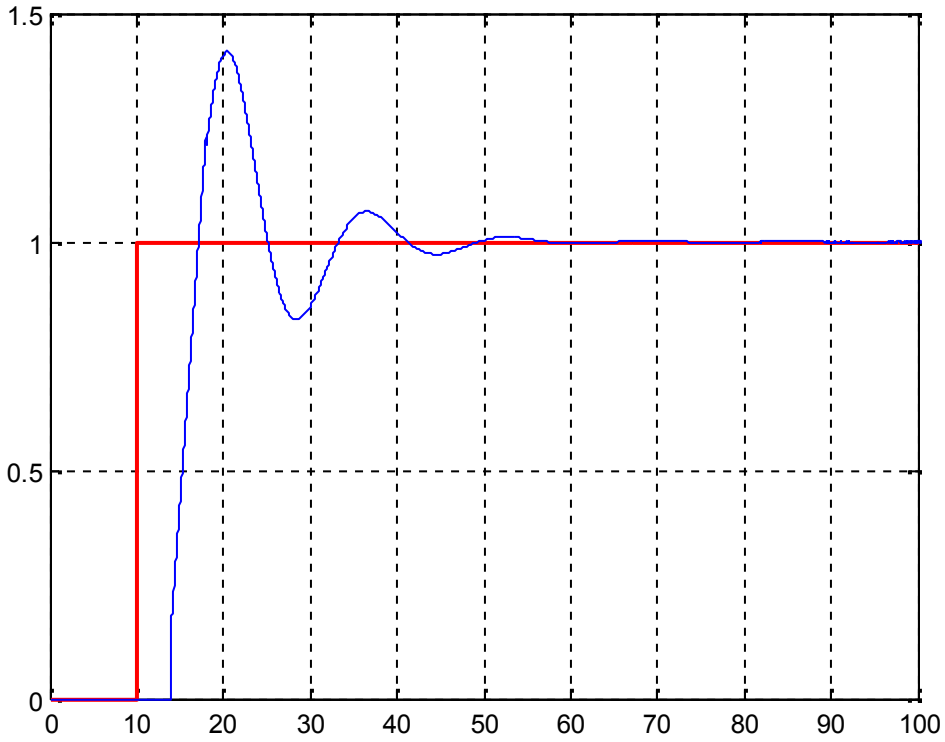


Figure I.8. Simulation pour KP=0.576, Ki=0.0653 et Kd=0.3388

I.8.1.1 Amélioration du temps de montée

Pour améliorer le temps de montée on augmente **Kp** à 0.7, ce qui nous donne le résultat illustré dans la figure suivante :

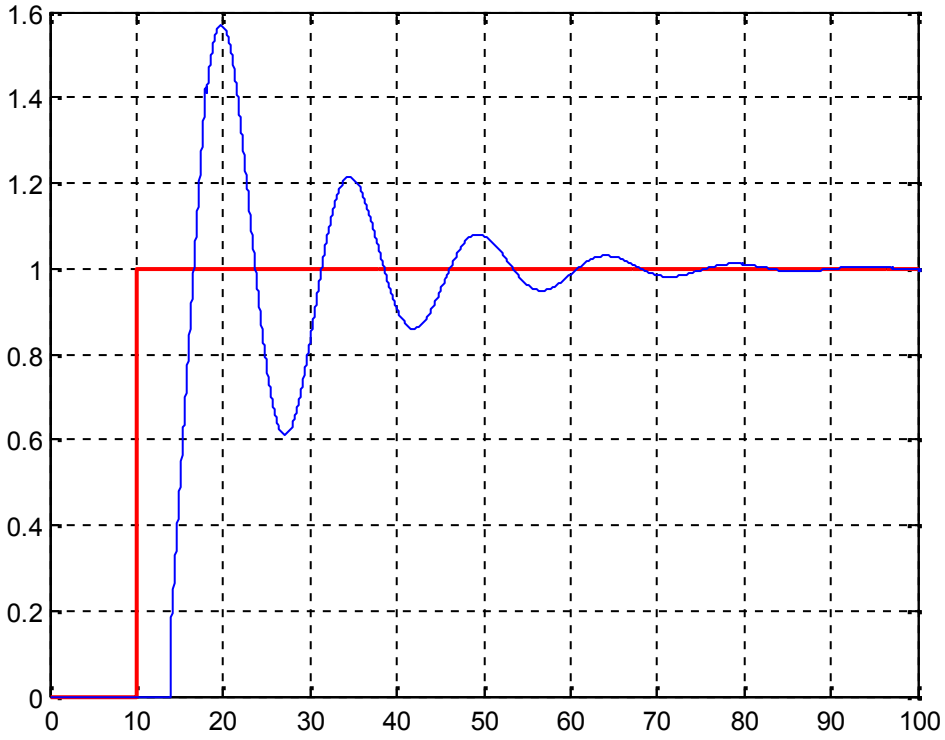


Figure I.9. Simulation pour **Kp**=0,7

La figure nous montre que pour un **Kp** supérieur à 0.567 on obtient un temps de montée inférieur à celui du précédent. Autrement, le dépassement devient plus grand.

I.8.1.2 Amélioration de l'erreur statique

À cet effet, la simulation a dû être refaite cette fois-ci. Le **Ki** est augmenté à **0.1**, suite à quoi nous avons obtenu le résultat ci-dessous (figure I. 10) :

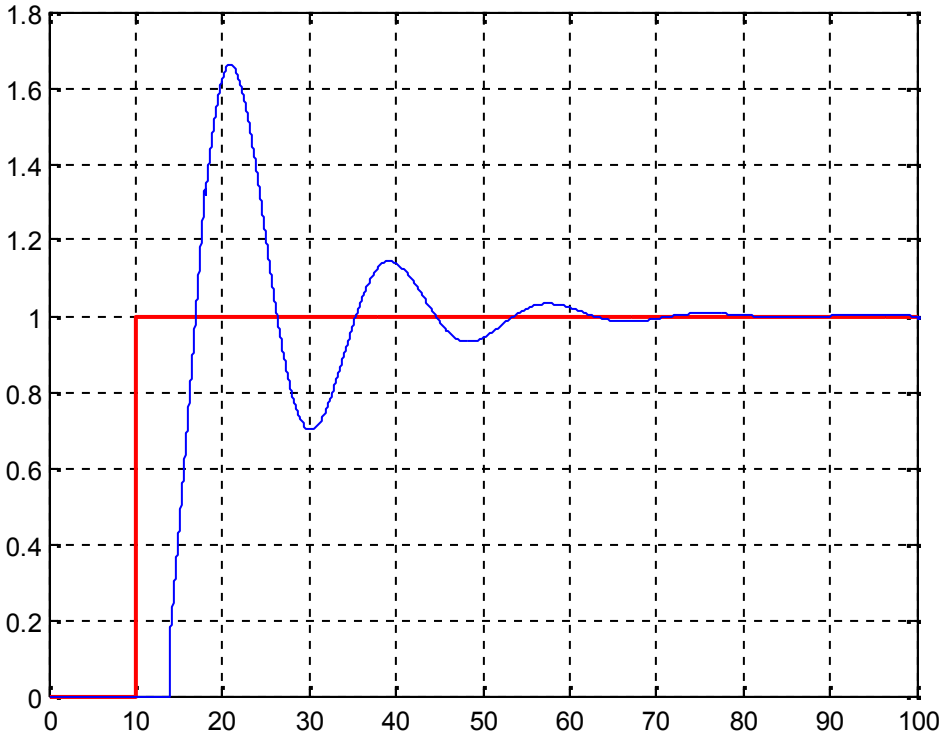


Figure I.10. Simulation pour Ki=0.1

Il est à remarquer dans cette figure que le temps de montée est légèrement réduit par rapport à celui de la figure7.3, cependant il est d'un dépassement plus grand. Toutefois, ce qui est beaucoup plus intéressant, c'est l'amélioration de l'erreur statique, afin de réaliser un système plus précis.

I.8.1.3 Amélioration de la précision

Enfin, pour obtenir un système plus précis, il faut augmenter le **Kd** à 0.5, aboutissant ainsi aux résultats qu'affiche la figure suivante :

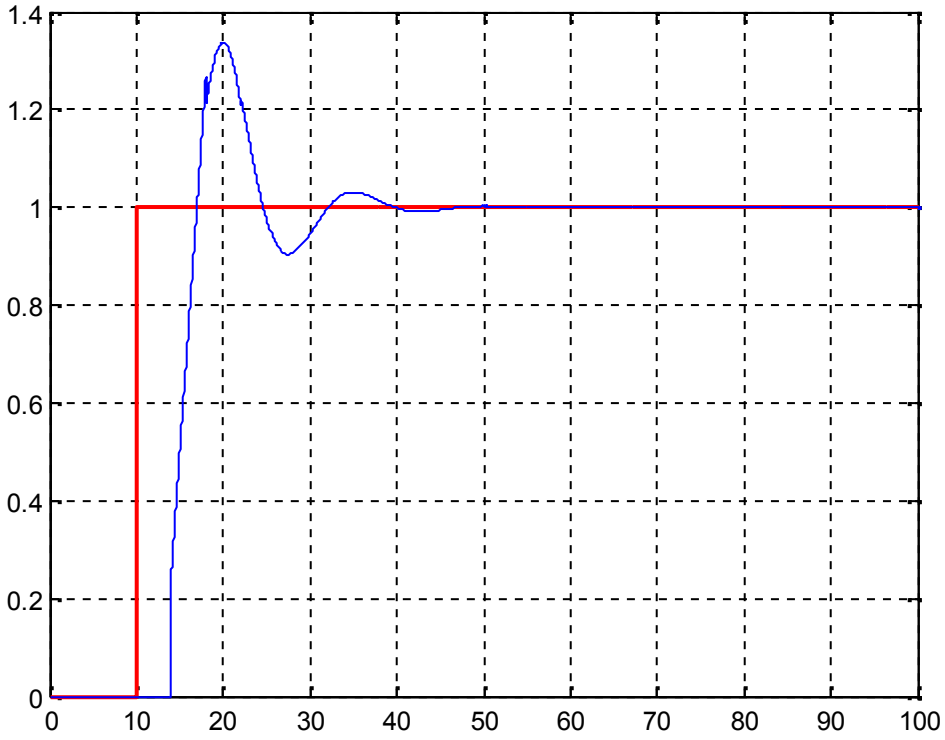


Figure I.11. Simulation pour Kd=0.5

Lorsqu'on augmente **Kd,** le système gagne en précision, en amélioration de temps de montée et en temps de dépassement. L'inconvénient est un déficit de stabilité qu'entraine cette augmentation de **Kd**.

I.8.2 Cas de contrôleur PID automatique

Dans cette expérience, le choix de commande est laissé au contrôleur automatique. En un petit moment, le contrôleur automatique à fini l'exécution, nous livrant les paramètres suivants: **Kp=0.2814, Ki=0.1279, Kd=0.4833**

Après la simulation, le contrôleur automatique nous a donné ce signal de référence rapporté dans la figure suivante :

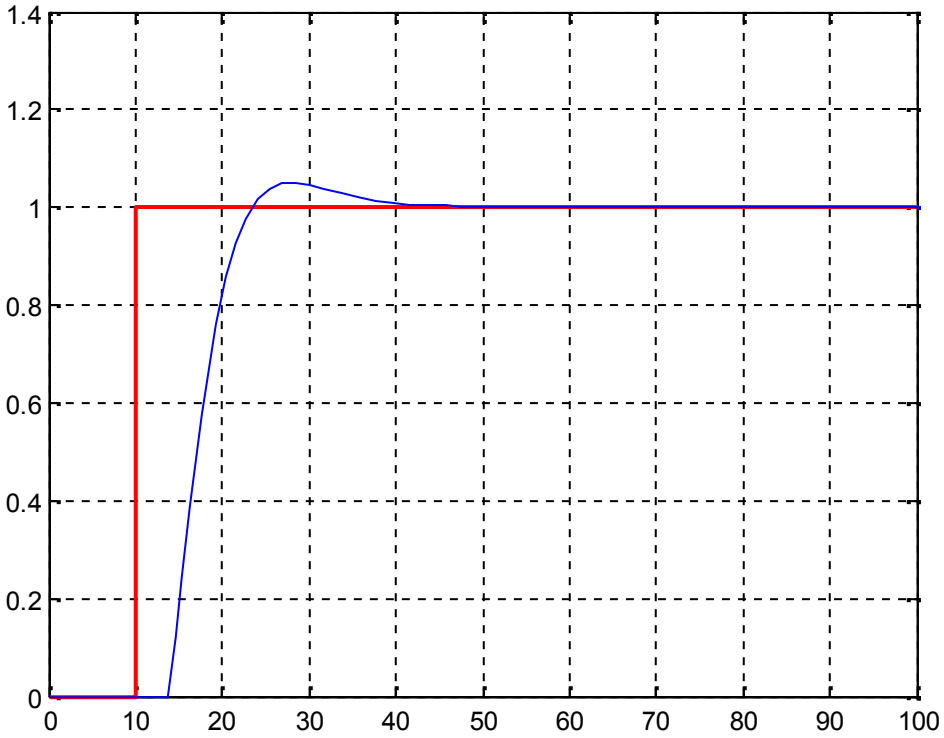


Figure I.12. Simulation par un PID automatique

Le contrôleur automatique a choisi pour le système un **Kp** inférieur à celui qu'on a opéré manuellement, ce qui enregistre un temps de montée inférieur à celui de notre commande. Certes, il rend le système lent, mais il diminue le dépassement qui fait moins osciller le système. En revanche, le contrôleur automatique augmente **Ki** et **Kd** pour venir à bout de l'erreur statique du système, ce qui fait de lui un système plus précis et d'une stabilité remarquable pendant tout le laps de fonctionnement du système.

Le contrôleur automatique est un programme accordant une importance primordiale à la précision et la stabilité du système, ce qui influe sur la rapidité ; il prend aussi un temps plus bref à calculer les paramètres, prouvant ainsi nettement son état de contrôleur rapide et facile à manier.

I.9 Conclusion

Le rappel des principes de base de la régulation PID des systèmes, abordés dans ce chapitre, ont conduit à la mise en évidence des différents types de contrôleur**,** mettant ainsi en relief les différentes méthodes de réglage des actions. En traitant les différents types de contrôleur par PID, il nous a également été possible de mettre en relief les différentes méthodes de réglage des actions.

On a clos ce chapitre par l'exemple d'une fonction de transfert de premier ordre, simulée par deux **PID** contrôleurs, l'un manuel et l'autre automatique, tout en illustrant les différentes caractéristiques du système avec des figures de simulation avec *Matlab/Simulink.*

Chapitre II

Les contrôleurs de signaux numériques dsPICs

II.1 Introduction

Parallèlement aux microcontrôleurs (MicroController Unit MCU), les processeurs de traitement numérique du signal (Digital Signal Processor DSP) ont bénéficié des énormes progrès en rapidité (grâce au faible temps de commutation) et en puissance de calculs (grâce au-nombre de bits des bus internes). Un dispositif multicœur MCU/DSP combinants les caractéristiques des 2 types de processeurs offrera le meilleur des deux composants.

En effet, la plupart des jeux d'instructions des microcontrôleurs sont très adaptés à des tâches de contrôle, mais pas nécessairement au traitement des données qui engendrent un calcul plus ou moins complexe. Modifier ce jeu d'instruction (et donc l'architecture sous-jacente) pour ajouter des instructions centrées sur le traitement de données est devenue plus courant et a donné lieu à ce qu'on appelle parfois les architectures unifiées, mais plus communément connu comme Contrôleurs de Signaux Numériques (Digital Signal Controller: DSC).

Les DSCs représentent la première étape vers un microcontrôleur multi cœur complet. Les applications qu'ils abordent reflètent généralement un besoin croissant de plus de contrôle ou de traitement intelligent dans des applications où les microcontrôleurs sont traditionnellement utilisés.

Les DSCs de Microchip appelés dsPICs sont des microcontrôleurs 16 bits renforcés de blocs de calcul (DSP Engine) conçus pour les applications à fort contenu algorithmique (commande de moteurs, régulation, control adaptatif, chaîne numérique de puissance, filtrage, etc.).Les microcontrleurs dsPIC existent en 2 grandes familles 30F et 33F/E qui se déclinent encore en plusieurs gammes de composants programmables spécialisées selon le type et les performances de la périphérie intégrée.

II.2Architecture générale des dsPICs [3]

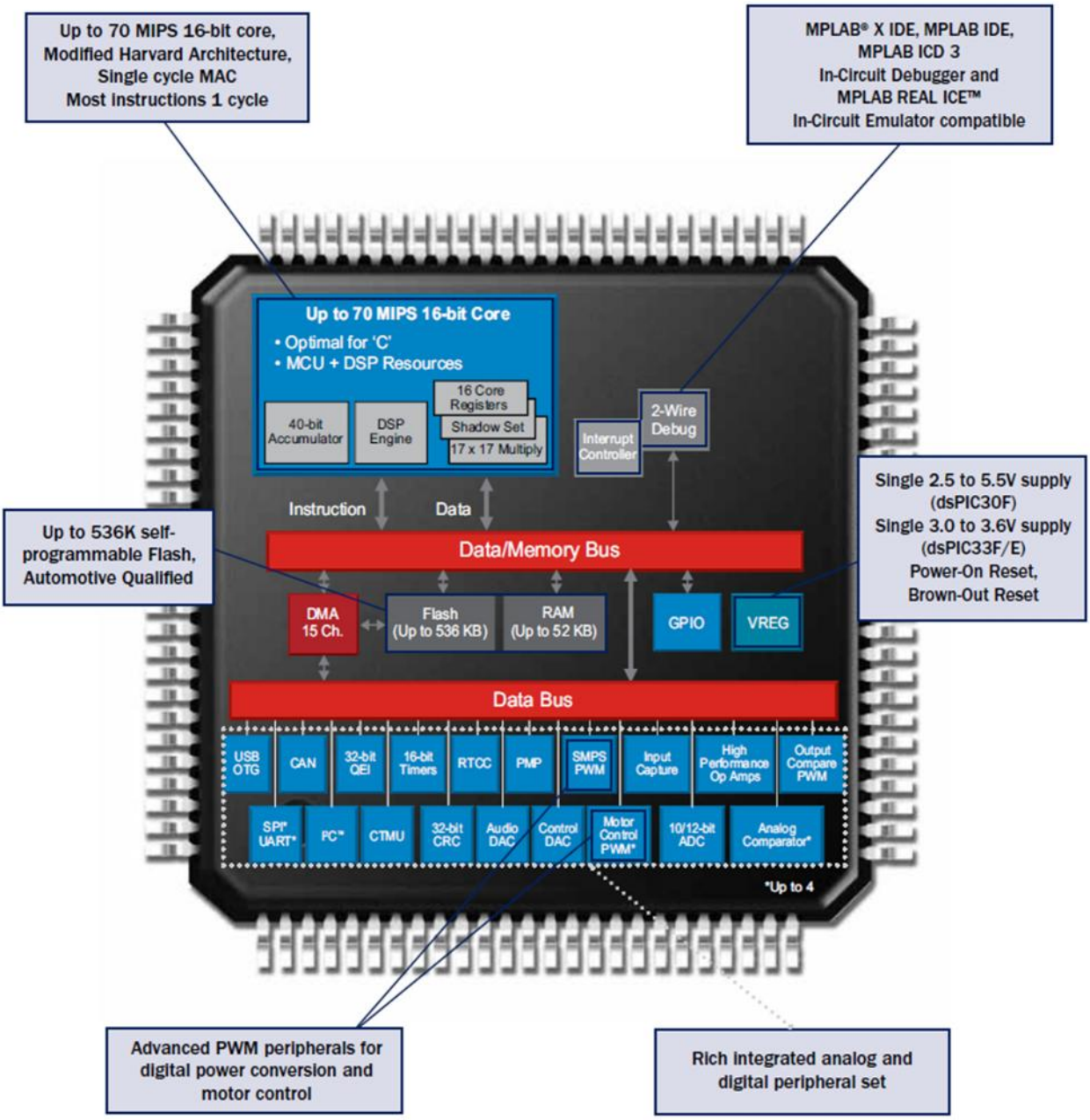
La figure II.1 présente l’architecture générale d’un dsPIC dont les caractèristiques sont les suivantes:

- Architecture Harvard avec des bus programme et données séparés: soit 16 bits pour les données et 24 bits pour les instructions. Cette organisation permet de réduire considérablement les temps d'exécution car, pendant que le processeur lit la prochaine instruction dans la mémoire programme, il exécute l'actuelle qui manipule des données en RAM.
- Un cœur DSP qui est le processeur annexe dédié aux calculs mathématiques. Ce ci confère au dsPIC la possibilité de réaliser rapidement les multiplications, division, calculs

trigonométriques et d'utliserun langage de programmation de haut niveau (tel que).Ce cœur comporte :

- un multiplieur câblé 17x17 bits ;
- des accumulateurs de 40 bits ;
- un "barrel shifter" capable de réaliser jusqu'à 15 décalages sur une donnée en un seul cycle ;
- Supporte l'instruction de calcul MAC (Multiply and Accumulate).

-



FigureII.1. Architecture générale des dsPIC

- Memoires de programme de type flash et 2 mémoies de donnée (RAM et EEPROM) dont les tailles varie selon le type du composant.
- Un Controleur d'interruption gérant un grand nombre de sources d'interruption internes et externes de priorité configurable.
- Rapidité d'éxcution variant entre 30MIPS et 70MIPS (Million d'instructions Par Seconde) et plus.
- Un large éventail de périphériques intégrés, nous citons principalement :
    - Des temporisateur/compteur 16 bits hautement configurable avec une multitude de modes de fonctionnement (Timers).
    - Des convertisseurs analogiques numériques rapides à 10bit et 12bits (ADC).
    - Des générateurs de signaux à largeur d'impulsions modulée (Pulse Width Modulation PWM) avec plusieurs modes de fonctionnement pour différentes application : contrôle d'alimentaions à découpage (Switch Mode Power Supply SMPS, contrôle de moteur…)
    - Des comparateurs digitales (Compare/Capture) et analogiques rapides (Analog Comparator) et des des amplificateurs opérationnels (OP Amps) .
    - des interfaces de communications série I2C™, SPI et UART, USB.
    - des périphériques de communication CAN (Control Area Network)
    - une horloge/calendrier temps réel RTCC.

II.3 Différents types de dsPICs [4]

- **Les dsPIC à usage général (General Propose DSCs):**

Les DSCs GP sont dotés de fonctionnalités spécialisées et de périphériques optimisés pour les applications à usage général permettent aux développeurs de créer des produits hautes performances à moindre coût :

Product	CPU Speed (MIPS)	Program Memory (KB)	RAM (B)	Pin Count	XLP	A/D Ch	UART	SPI	I2C	USB	CAN	PWM outputs	16 Bit Timers	32 Bit Timers
dsPIC30F3014	30	24	2048	40	No	13	2	1	1	0	0	2	3	1
dsPIC30F4011	30	48	2048	40	No	9	2	1	1	0	1	10	5	2
dsPIC30F6014A	30	144	8192	80	No	16	2	2	1	0	2	8	5	2
dsPIC30F6015	30	144	8192	64	No	16	2	2	1	0	1	16	5	2
dsPIC33FJ128GP802	40	128	16384	28	No	10	2	2	1	0	1	4	5	2
dsPIC33FJ128GP804	40	128	16384	44	No	13	2	2	1	0	1	4	5	2

Tableau II.1 Quelques dsPIC à usage général

- **Les dsPICs Orientés capteurs :**

Les boîtiers sont particulièrement petitsse distingue par sa capacité à fonctionner sous 5 V pour une connexion facilitée à des capteurset pour une meilleure robustesse et immunité aux bruits dans des environnements difficiles (électroménager, industriel, automobile). Certains dsPIC sont dotés de mémoire flash avec codage ECC (Error Correcting Code) pour une fiabilité et une sûreté améliorée. On y trouve également un contrôle de redondance cyclique CRC (Cyclic Redundancy Check), un temporisateur de sécurité DMT (Deadman Timer) et un temporisateur de chien de garde à fenêtre WWDT (Windowed Watchdog Timer). La majorité de ces DSCs comportent des controleurs bus de communication multiples tels queles CAN (Control Area Network) et SENT (Single Edge Nibble Transmission).

Product	Pins	Flash Memory Kbytes	RAM Bytes	EEPROM Bytes	Timer 16-bit	Input Capture	Output Compare/Standard PWM	A/D 12-bit 200 ksps	UART	SPI	I²C™	I/O Pins (Max.)†	Package Code
dsPIC30F2011	18	12	1024	—	3	2	2	8 ch, 1 S/H	1	1	1	12	P, SO, 28-pin ML
dsPIC30F3012	18/44	24	2048	1024	3	2	2	8 ch, 1 S/H	1	1	1	12	P, SO, 44-pin ML
dsPIC30F2012	28	12	1024	—	3	2	2	10 ch, 1 S/H	1	1	1	20	SP, SO, 28-pin ML
dsPIC30F3013	28/44	24	2048	1024	3	2	2	10 ch, 1 S/H	2	1	1	20	SP, SO, 44-pin ML

Tableau II.2 Quelques dsPICs orientés capteurs

- **Les dsPIC de contrôle de moteurs (Motor Control MC DSCs):**intègrent desgénérateurs PWM à plusieurs canaux avec sorties synchronisées pour une utilisation en courant triphasé, ce qui les rend compatibles avec un vaste éventail d'algorithmes et d'applications de commande de moteurs, depuis le simple moteur à capteurs jusqu'au contrôle de flux sinusoïdal avancé en passant par les moteurs brushless DC, les moteurs à aimant permanent (PMSM) et les moteurs à induction AC synchrones (ACIM). Les développeurs peuvent ainsi obtenir de meilleures performances : efficacité améliorée, fonctionnement plus silencieux, couple plus régulier, meilleure fiabilité.

Product	Pins	Flash Memory Kbytes	RAM Bytes	EEPROM Bytes	Timer 16-bit	Input Capture	Output Compare/ Standard PWM	Motor Control PWM	Quadrature Encoder	A/D 10-bit 1 Msps	UART	SPI	I²C™	CAN	Package Code
dsPIC30F2010	28	12	512	1024	3	4	2	6 ch	Yes	6 ch, 4 S/H	1	1	1	—	SP, SO, MM
dsPIC30F3010	28/44	24	1024	1024	5	4	2	6 ch	Yes	6 ch, 4 S/H	1	1	1	—	SP, SO, 44-pin ML
dsPIC30F4012	28/44	48	2048	1024	5	4	2	6 ch	Yes	6 ch, 4 S/H	1	1	1	1	SP, SO, 44-pin ML
dsPIC30F3011	40/44	24	1024	1024	5	4	4	6 ch	Yes	9 ch, 4 S/H	2	1	1	—	P, PT, ML
dsPIC30F4011	40/44	48	2048	1024	5	4	4	6 ch	Yes	9 ch, 4 S/H	2	1	1	1	P, PT, ML
dsPIC30F5015	64	66	2048	1024	5	4	4	8 ch	Yes	16 ch, 4 S/H	1	2	1	1	PT
dsPIC30F6015	64	144	8192	4096	5	8	8	8 ch	Yes	16 ch, 4 S/H	2	2	1	1	PT
dsPIC30F5016	80	66	2048	1024	5	4	4	8 ch	Yes	16 ch, 4 S/H	1	2	1	1	PT
dsPIC30F6010A	80	144	8192	4096	5	8	8	8 ch	Yes	16 ch, 4 S/H	2	2	1	2	PT

Tableau II.3 Quelques dspics pour Contrôle de moteurs

- **Contrôle d'alimentation:**Ils sont caractérisés par plusieurs modules PWM à plusieurs cannaux à haute résolution (jusqu'à 1nS), des convertiseurs A/N 10 bits et 12bits rapides et des comparateurs analogiques.

Product	CPU Speed (MIPS)	Program Memory (KB)	RAM (B)	Pin Count	XLP	A/D Ch	UART	SPI	I2C	USB	CAN	PWM outputs	16 Bit Timers	32 Bit Timers
dsPIC30F1010	30	6	256	28	No	6	1	1	1	0	0	5	2	0
dsPIC30F2010	30	12	512	28	No	6	1	1	1	0	0	8	3	1
dsPIC30F2020	30	12	512	28	No	8	1	1	1	0	0	10	3	1
dsPIC30F2023	30	12	512	44	No	12	1	1	1	0	0	10	3	1

Tableau II.4 Quelques dsPICsde contrôle d'alimentation

II.4 Les applications des dsPICs

Les applications des dspics sont multiples dans les domaines suivants :

- télécommunications : modem, multiplexeurs, récepteurs de numérotation DTMF, télécopieurs, codeurs de parole GMS, …),
- interfaces vocales : codeur vocaux pour répondeurs, reconnaissance automatiquede la parole, synthèse vocale …
- militaire : guidage missiles, navigation, communications cryptée, radar, …
- multimédias et grand public : compression des signaux audio (CD), compressiondes images, cartes multimédias pour PC, synthèse musicale, jeux, …
- médical : compression d'image médicale (IRM, échographie…), traitements dessignaux biophysiques (ECG, EEG,…), implants cochléaires, équipement demonitoring.
- électronique automobile : équipement de contrôle moteur, aide à la navigation, commande vocale, détection de cliquetis pour avance à l'allumage, …
- automatisation et contrôle de processus : surveillance et commande de machines, contrôle de moteurs, robots, servomécanisme, …

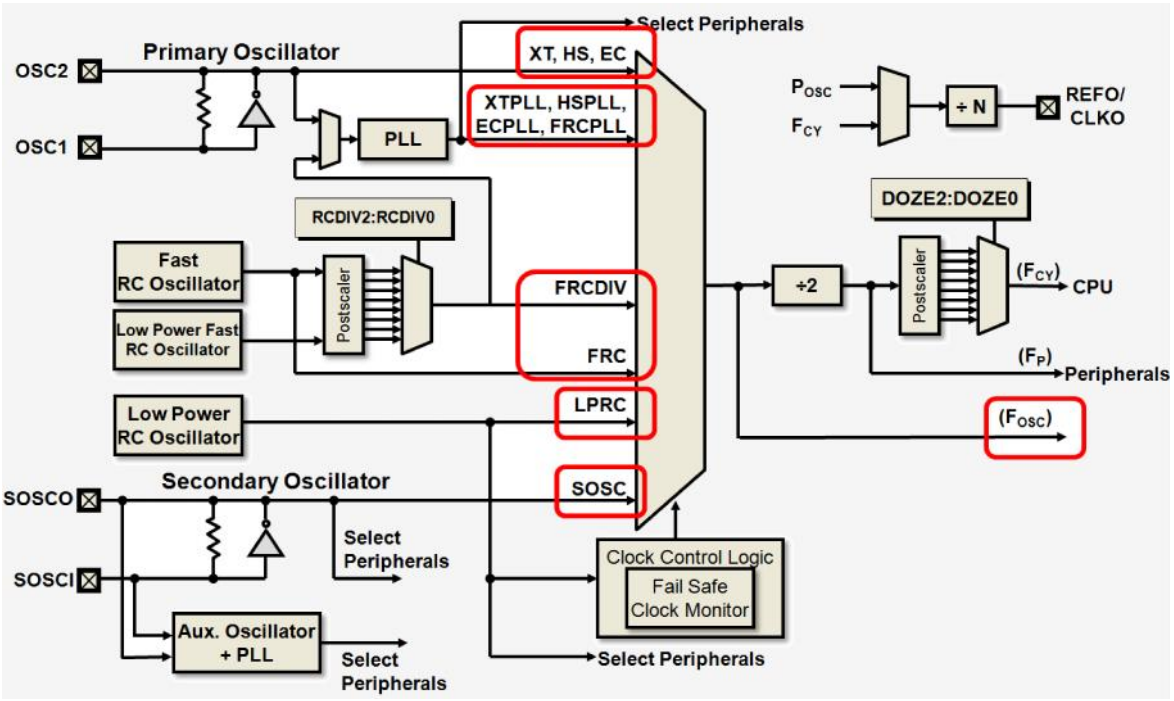
- instrumentation : analyseur de spectre, générateurs de fonction, interprétation designaux sismiques, …

II.5 Principes de fonctionnement des principaux modules de dsPICs

II.5.1 Le système d'horloge

Les dsPICs possèdent un système d'horloge largement configurable assez complexe avec plusieurs sources d'horloges dérivées de sources d'horloge interne ou externe. La fréquence de l'horloge peut être divisée et/ou multiplié afin d'obtenir la fréquence requise. La PLL (boucle à verrouillage de phase) permet d'augmenter la fréquence de l'oscillateur utlisé en la multipliant par un facteur configurable, tandis que les prédiviseurs programmables permetnt de la réduire afin d'adapter la fréquence d'entrée en fonction de l'application.

Un mécanisme Fail-Safe moniteur Horloge (FSCM) détecte une défaillance de l'horloge et permet la reprise sécuriséde l'exécution du programmed'application ou l'arrêt. La configuration par défaut (hardware power-up) après la mise sous tension est déterminée par les paramètres de bits de configuration. Une configuration supplémentaire peut être réglée lors de l'exécution du programme.

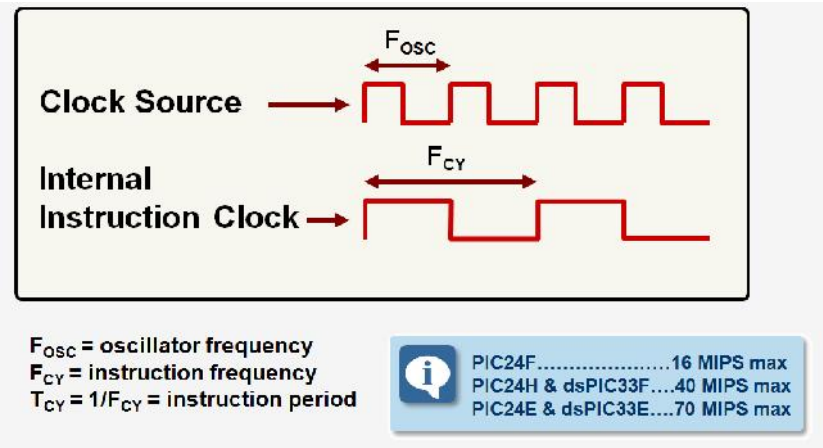


FigureII.2. Le système d'horloge du dsPIC

La source de l'horloge (F_{OSC}) du système est configurable parmi les 5 sources suivantes:

- ***Primary Oscillator*** (**POSC**) appliqué aux pins OSC1 and OSC2 .ins
- ***Internal Phase-Locked Loop*** (**PLL**)
- ***Internal Fast RC Oscillator*** (**FRC**)
- ***Internal Low-Power RC Oscillator*** (**LPRC**)
- ***Secondary Oscillator*** (**SOSC**) appliqué aux pins SOSCI and SOSCO pins

L'horloge d'exécution des instruction (Frésuence Cycle) Fcy (égale aussi à la féquence de fonctionnement des périphériques Fp) est égale à Fosc/2.



FigureII.3. Relation entre Fosc et Fcy

II.5.2 Ports d'entrée/sortie (E/S)

Les portes d'entrée/sortie sont les parties les plus utilisées dans le dsPIC**.** Elles sont très variables en nombre et en type. Comme les PIC MCU, Les broches d'E/S sont multiplexées est partagés par les périphériques tels que les entrées analogiques, sorties PWM, interfaces UART, broches d'interruption externesetc. Le registre TRISx est utilisé pour déterminer la direction de la broche (Entrée / Sortie). La configuration des broches du porten entrées analogiques se fait en utilisantle registre de configuration correspondant.

Une caractéristique particulière des dsPICs est la fonction PPS (Peripheral Pin Select) qui permet l'affectation de broches spécifiques appelé RPx à des périphériques au choix(UART, SPI, PWM, entrée d'horloge, entrée interruption…). Cette fonctionnalité rend possible le routage simple des pistes lors de la réalisation matérielle.

II.5.3 Le circuit d'initialisation (Reset)

Le Reset se définit comme l'événement qui réinitialise le programme. Pour les dsPIC, il existe plusieurs évenements qui peuvent engendé un reset :

- POR: Power-on Reset;
- MCLR :Reset during normal operation;
- MCLR :Reset during Sleep;
- WDT: Watchdog Timer (WDT) Reset (during normal operation);
- BOR: Programmable Brown-out Reset (BOR);
- Reset Instruction;
- Reset cause par une exception (trap);

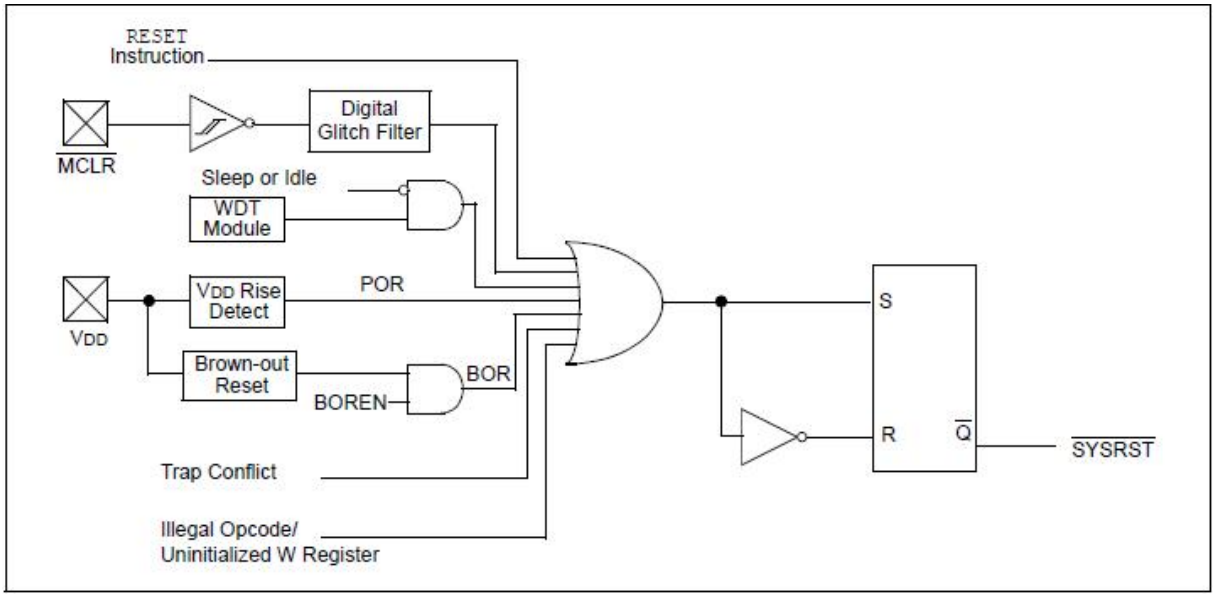


Figure II.4. Bloc diagramme de système Reset.

II.5.4. Le convertisseur Analogique Numérique

La structure et les modes de fonctionnement varient d'un composant à un autre, néanmoins les principes de base de fonctionnement restent identiques. La figure II.7 montre un convertisseur 10-bit à approximations succéssives avec 4 échontillonneurs/bloqueurs dédiée (dedicated S&H (Sample & Hold) et 2 échontillonneurs/bloqueurs partagés (shared S&H). Se convertisseur contient 2 SARs (Succesive Approximations Registers) ce qui veut dire que ce convertisseur comporte 2 modules de conversion. Les résultats de conversion sont mémorisés temporairement dans des buffers dédiés (ADCBUFx)

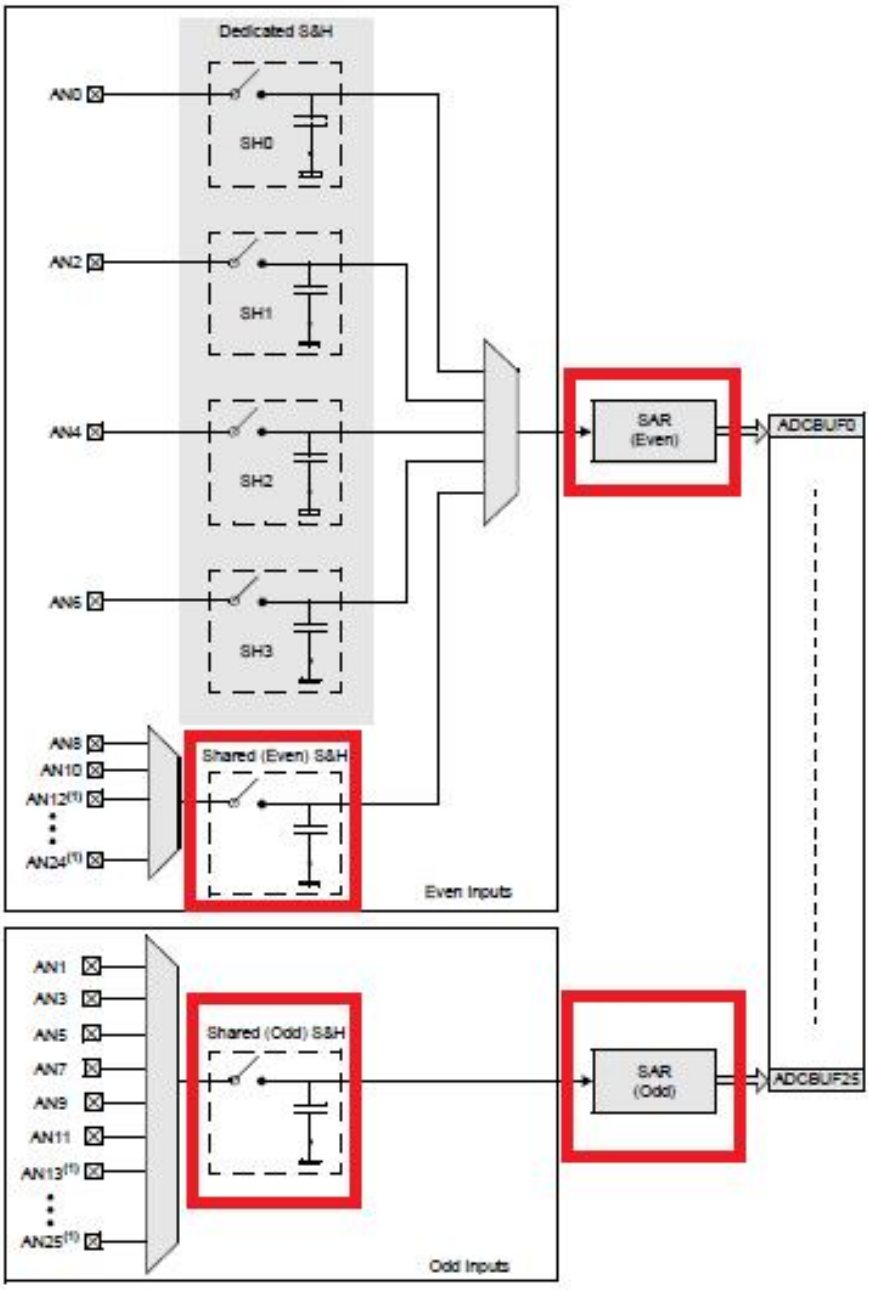


Figure II.5 Exemple de convertisseur A/N d'un dsPIC33.

Les convertisseurs A/N sont largement configurables. Les modes de fonctionnement peuvant être configurés sont multiples :

- Le lancement de la conversion peut être automatique après l'échantillonnage, ou déclenchable par un évenement interne ou externe d'une trentaine de sources.
- La conversion simple ou multi cannaux permet de choisir entre l'utilisation du convertisseur pour une seule entrée, ou scanner séquentiellement quelques ou toutes les entrées analogiques.

- Les tensions de référence positive et négatives du convertisseur peuvent être choisies avec une multitude d'option: tension d'alimentation du dsPIC, tension appliquée aux pins d'entrées …

II.5.6 Le générateur PWM

Le principe de fonctionnement d'un générateur de signal PWM des dsPIC est articulé sur l'utlisation de trois registres de bases :

- Le registre de comptage (counter)
- Le registre de période (Period register)
- Le registre de la largeur d'implusion (Duty register)

Des comparateurs sont aussi utlisés pour controler le signal PWM (figure II.8): L'impulsion commence lorsque la valeur du compteur est égale la valeur contenue dans le registre de la période et se termine lorsque la valeur du compteur est égale à la valeur du registre de la largeur d'impulsion.

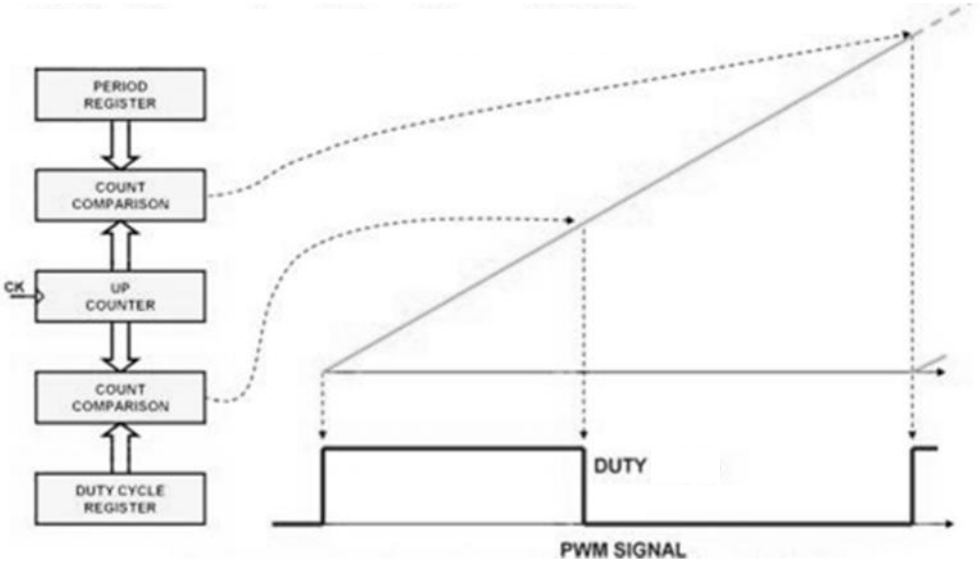


Figure II.6. Le principe de fonctionnement d'un générateurPWM.

* Les modes d'opérations :

Pour couvrir un large spectre d'applications mettant en œuvre un ou plusieurs Signaux PWM, le générateur PWM des dspic est conçu pour délivrer une grande variété de formesde signaux. En général, ce module opère selon les modes configurables suivants :

a) **Mode complémentaire** : le générateur délivre avec une pair de pins (notées L et H) deux signaux compémentaires (l'un est l'inverse de l'autre) (figureII.9). pour éviter les

problèmes de transions simultanés des 2 signaux pour certaines applications, des retards configurables peuvent être insérés (Dead time).

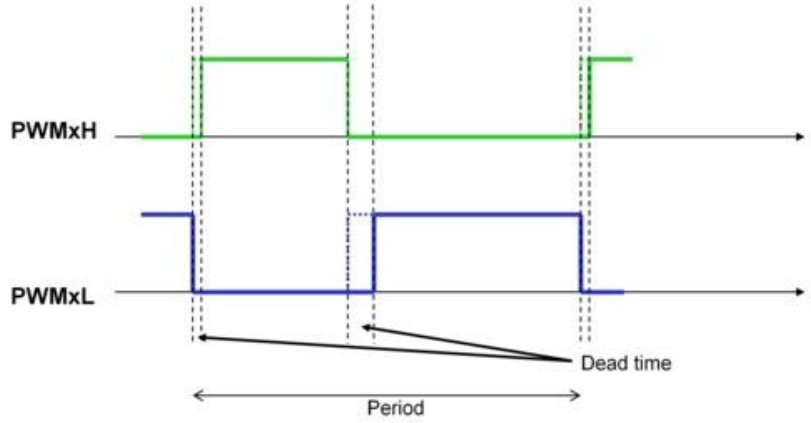


Figure II.7. Mode d'opération complémentaire

b) **Mode redondant** : le générateur délivre avec une pair de pins L et H le même signal PWM.

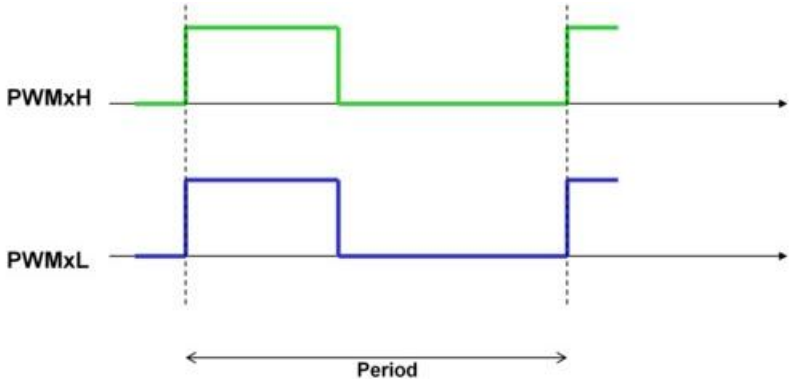


Figure II.8. Mode d'opération redondant

c) **Mode indépendant** : le générateur délivre deux signaux PMW différents à travers la pair de pins L et H.

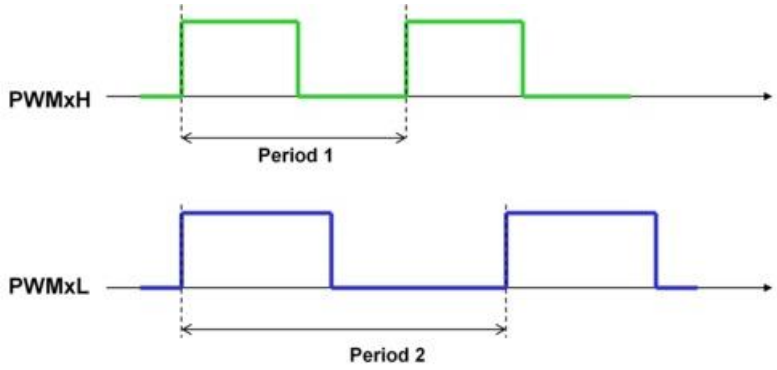


Figure II.9. Mode d'opération indépendant

d) **Mode multiphases** : le générateur délivre sur plusieurs pairs des signaux PMW identiques et déphasés.

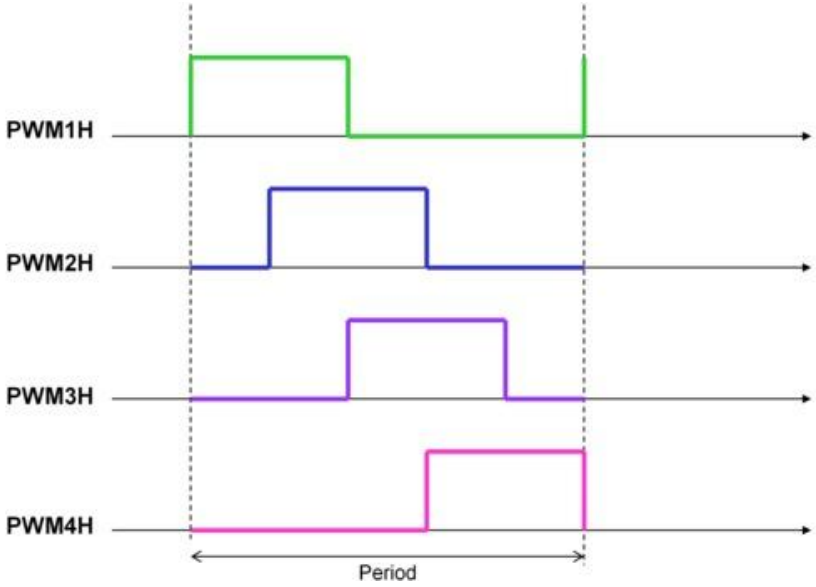


Figure II.10. Mode d'opération multiphases

e) **Mode à phase variable** : le générateur délivre sur une paire de broches 2 signaux PMW identiques et dépahsés. La phase étant configurable.

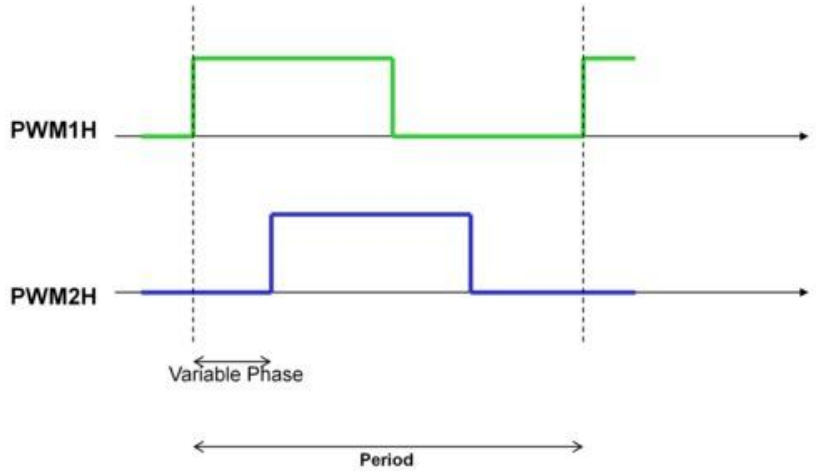


Figure II.11. Mode à phase variable

II.6 Conclusion

Dans ce chapitre, et loin des détails des fiches techniques, nous avons présentés les caractéristiques générales de la la famille dsPIc de Microchip afin d'exposer leurs domaines d'applications potentiels. Leurs principes d'utlisation simples et similaires à ceux des microcontrôleurs couplés à leurs potentiels de calcul rapide et de périphérie intégrée rendent leurs exploitation incontournables spécialement dans des applications suscitant un calcul plus au moins complexe, tels que les controleurs PID.

Chapitre III

Programmation et simulation

III.1 Introduction

Dans ce chapitre, nous présentons l'implémentation d'un contrôleur PID sur un microcontrôleur dsPIC, dans le but de contrôler la température d'un four électrique. Le programme d'application est écrit en C en utilisant l'environnement de développement MikroC Pro for dsPIC. Le système développé est simulé en utilisant Proteus-Isis.

III.2 Description de l'application

Pour notre application, nous nous proposons d'implémenter un contrôleur PID pour régler la température d'un four autour d'une consigne Tc. La température est alors mesurée et convertie en valeur numérique pour être traitée par le dsPIC. Le commande PID est ensuite calculée en fonctionne de l'erreur entre la valeur mesurée et la consigne. La commande calculée est utilisé pour modulé la durée d'impulsion d'une PWM. Toutes les fonctions de cette chaine de régulation (Acquisition + conversion A/N + Calcul PID + Génération PWM) sont implémentées sur ledsPIC.

Le schéma de simulation réalisé avec Proteus-Isis utilise les 2 principaux composants suivants :

- Le dsPIC33FJ32MC204
- Un four doté de capteur de température

III.2.1 Modèle de simulation du four

Sous Isis, le modèle de simulation du four est régi parles paramètres fonctionnels suivants (valeurs par défaut) :

- La température ambiante (°C) = 25
- Résistance thermique à l'air ambiant (°C/W) = 0.7 ;
- Constante de temps du four (ESA) = 10 ;
- Constante de temps de réchauffeur (ESA) = 1 ;
- Coefficient de température (V/°C) = 1 ;
- Puissance de chauffage (W)= 120.

Le modèle schématique du four contient une borne sensorielle *T* délivrant une tension proportionnelle à la température du système. Conformément aux paramètres précédemment établis, ce terminal devrait délivrer une tension de 1V/°C. Pour ainsi dire, en mesurant une température de 100° C, le terminal T donna 100 V.

III.2.1 Fonction de transfert du four

Le four utilisé est un système du 1er ordre dont la fonction de transfert est donnée par l'expression suivante :

$$G(s) = \frac{k_0 e^{-s\tau_0}}{1+\gamma_0 s} \quad (8)$$

Afin d'identifier les paramètres de cette fonction de transfert, il faut établir la réponse du système en boucle ouverte à une entrée échelon (courbe de réaction). Pour cela, nous utilisons l'analyse interactive de Proteus-ISIS (figure III.1).La réponse du système à un échelon d'entrée de 0V à 2V est collectée par la sonde OV1(T).

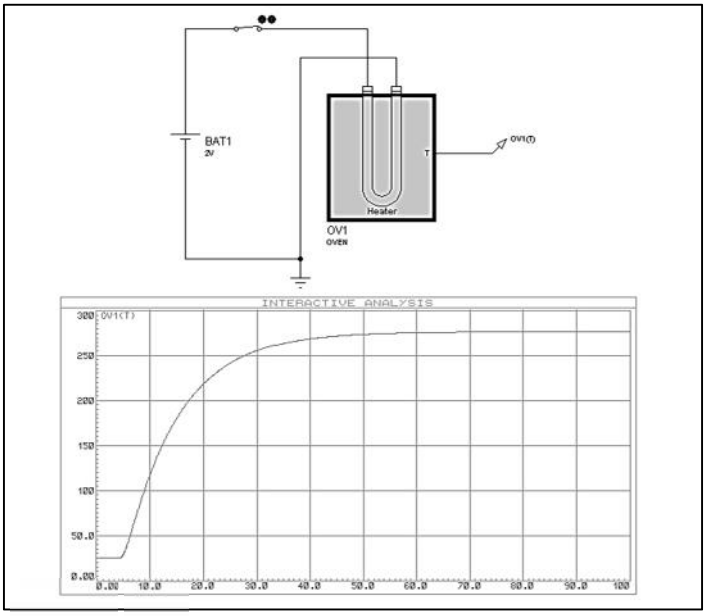


Figure III.1. Schéma d'analyse de la réponse du système

Les coefficients τ_0 ,γ_0 et k_0 sont déduits à partir de la ligne droite de la pente maximum de la manière suivante :

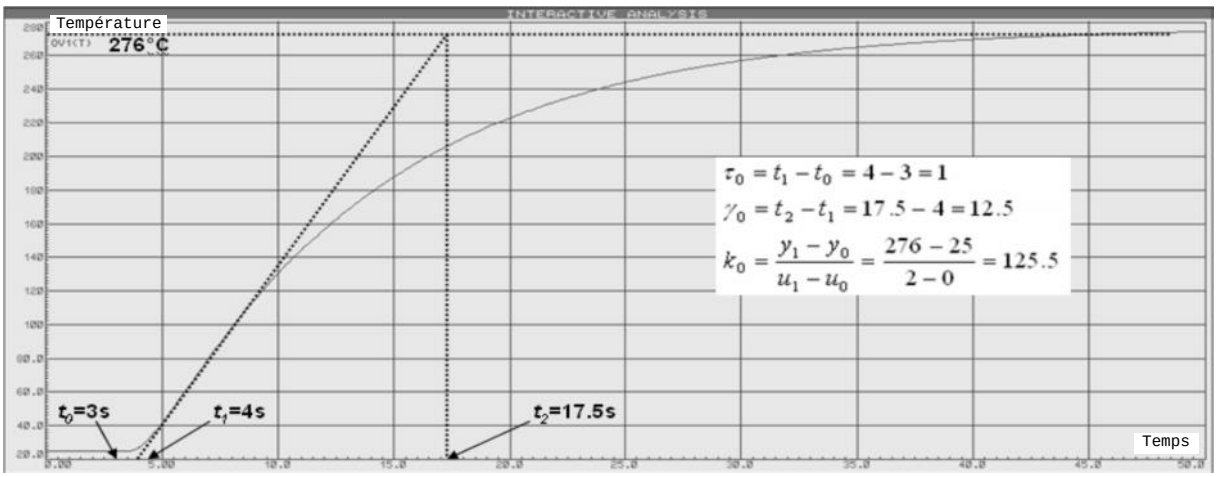


Figure III.2. Détermination des coefficients du système à partir de la courbe de réaction.

Par conséquent, le modèle du système de chauffage est défini ainsi :

$$G(s) = \frac{k_0 e^{-s\tau_0}}{1+\gamma_0 s} = 125.5\,\frac{e^{-s}}{1+12.5s} \quad (9)$$

III.3 Calcul des paramètres de PID [6]

Rappelons l'expression de la commande à appliquer à l'entrée du processus, générée par un contrôleur PID :

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)d\tau + K_d \cdot \frac{d}{dt}e(t) \quad (10)$$

Pour l'implémentation de cette commande sur un dsPIC, nous utilisons l'expression de calcul du contrôleur PID numérique donnée par :

$$U(i) = K_p \cdot e(i) + K_i \cdot Ts \sum e(i) + K_d \cdot \frac{e(i)-e(i-1)}{Ts} \quad (11)$$

Avec Ts période d'échantillonnage.

Les paramètres K_p , K_i et K_d sont calculés selon la règle de l'accord de Ziegler-Nichols basée sur la réponse du système :

$$K_p = \frac{\gamma_0}{k_0\tau_0} = 0.1243, K_i = \frac{K_p}{2\tau_0} = 0{,}06215, K_d = \frac{2Kp}{\tau_0} = 0{,}2486 \quad (12)$$

III.4. Implémentation du PID sur dsPIC

Comme déjà décrit, selon les fonctions mise en œuvre dans cette application, nous exploitons principalement un convertisseur A/N et un générateur PWM parmi les périphériques intégrés (figure III.3). Le régulateur PID est conçu à base d'un dsPIC33FJ32MC204 dont les principales caractéristiques sont données par le tableau III.1.

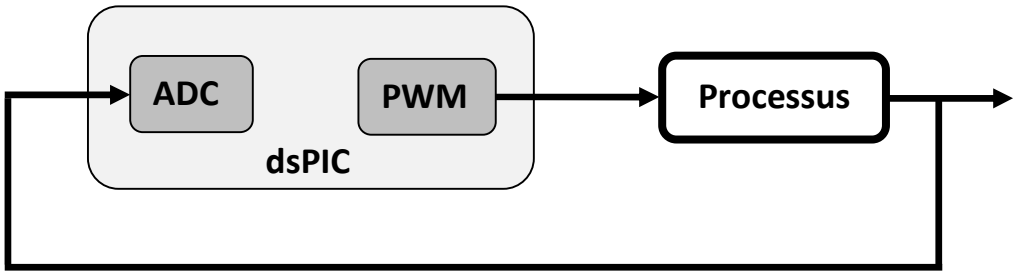


Figure III.3. Schéma bloc du système

Program Memory (KB)	RAM Bytes	I/O Pins	Digital Communication Peripherals	Analog Peripherals	Capture/Compare/PWM Peripherals	PWM Resolution bits	Motor Control PWM Channels	Timers
32	2,048	35	1-UART 1-SPI 1-I2C	1-A/D 9x12-bit 500(Ksample/s)	4/2	16	8	3 x 16-bit 1 x 32-bit

Tableau III.1 : Principales caractéristiques du dsPIC33FJ32MC204

III.4.1. Programme d'application

L'organigramme décrivant les différentes étapes de la régulation PID implémentée est le suivant :

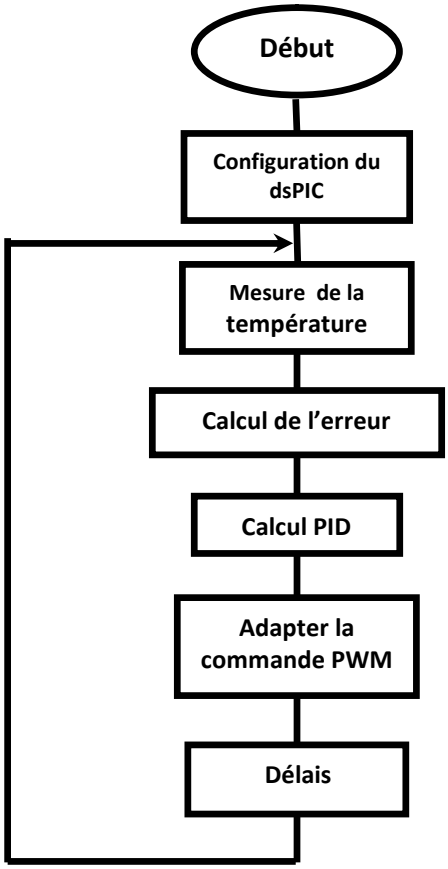


Figure III.4. Organigramme

III.4.2 Configuration des périphériques

III.4.2.1Convertisseur analogique numérique (ADC)

Le dsPIC33FJ32MC204 utilisé possède un convertisseur analogique/numérique à 9 entrées, capable d'offrir une précision de 10 bits. Ce convertisseur est capable

d'échantillonner jusqu'à quatre canaux à la fois [5].Sa configuration est effectuée en attribuant des valeurs spécifiques à plusieurs registres. La fonction `Init()`de `ADC Library` du compilateur mikroC pro for dsPICa été utilisée afin de faciliter la procédure.

**** Configuration par défaut du convertisseur:***

L'appel de la fonction `ADC1_Init()`de `ADC library` permet de configurer le convertisseur A/N avec les paramètres par défaut suivants:

- Conversion single Channel (pas de Channels scan);
- Résolution de conversion de 10 bits ;
- Les données sont codées en Entier non signé ;
- Les tensions de référence positive et négative VReF+ , VReF- sont égales à Vdd et Vss respectivement ;
- Horloge du convertisseur réglée à 32*Tcy (Tcy est la période du cycle du dsPIC) ;
- Conversion automatique (Lancement automatique après l'échantillonnage).

Le complément de la configuration nécessaire consiste à affecter la broche que nous avons sélectionné (AN2) au canal du convertisseur choisi:

- `AD1CHS0bits.CH0SB=2;`

Il faut noter ici que la mesure de la température s'effectue avec la fonction `ADC1_Read` de la même bibliothèque.

III.4.2.2 Le générateur PWM

La fréquence de ce signal doit être significativement plus élevée par rapport à la fréquence d'échantillonnage afin que le niveau de tension puisse être perçu. Le module PWM du dsPIC33FJ32MC204offre une multitude d'options configurées par l'intermédiaire de plusieurs registres de configuration [5]. La difficulté de configuration peut être contournée en utilisant les fonctions de `PWM_MC Library`. Voici la configuration du module PWM faisant appel à la fonction `PWM2_MC_Init` :

- La fréquence de PWM réglé à 1Khz.
- Activation de la sortie PWM2L1 du générateur PWM2(en mode indépendant)
- Désactivation du prédiviseur.

On note qu'au cours de la régulation, on utilise la fonction `PWM2_MC_Set_Duty` pour changer la valeur de la durée d'impulsion selon la valeur de la température mesurée.

III.4.3. Programmation

Pour écrire notre programme nous avons utilisé l'environnement de développement mikroC Pro for dsPIC (figure III.5).

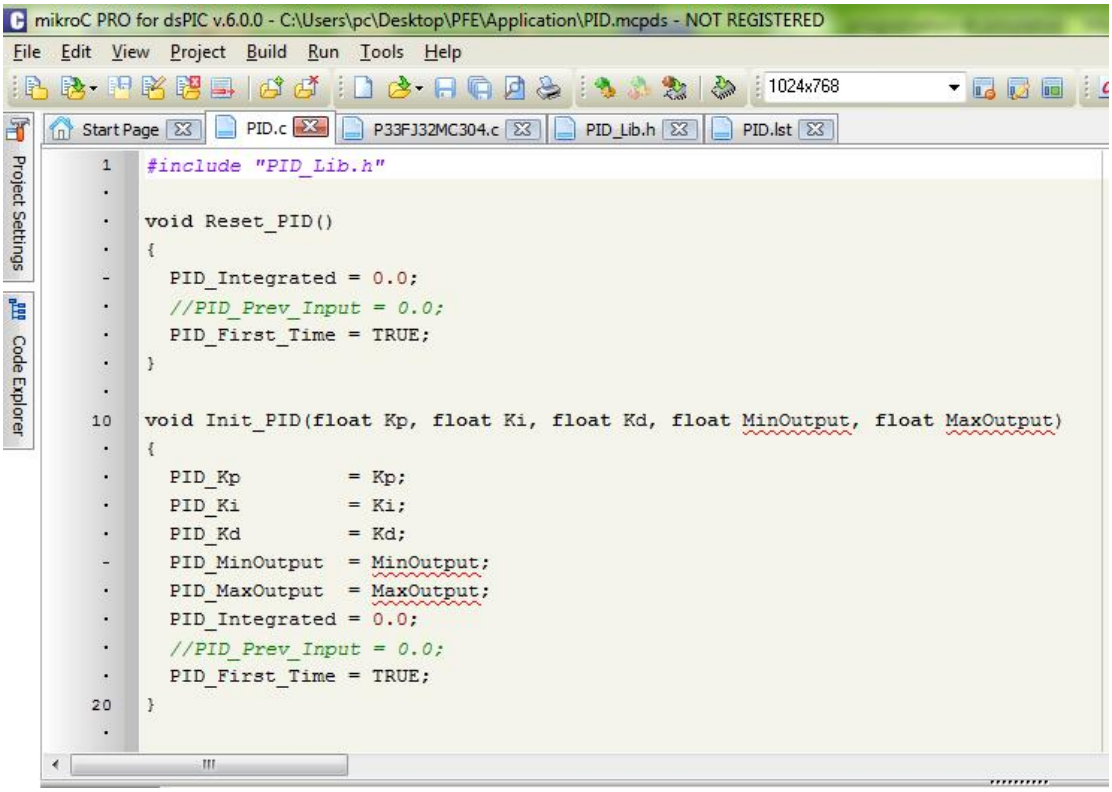


Figure III.5. mikroC Pro for dsPIC

Le programme:

```
#include "PID_Lib.h"
/*Le fichier PID_Lib.h comportment
  La déclération variables globales
  et les prototypes de fonctions */

void Reset_PID()
{
  PID_Integrated = 0.0;
  //PID_Prev_Input = 0.0;
  PID_First_Time = TRUE;
}
void Init_PID(float Kp, float Ki, float Kd, float MinOutput, float MaxOutput)
{
  PID_Kp         = Kp;
  PID_Ki         = Ki;
  PID_Kd         = Kd;
  PID_MinOutput  = MinOutput;
  PID_MaxOutput  = MaxOutput;
```

```
    PID_Integrated = 0.0;
    //PID_Prev_Input = 0.0;
    PID_First_Time = TRUE;
  }

  float PID_Calculate(float Setpoint, float InputValue)
  {
    float result;
    ErrValue = SetPoint - InputValue; // --- calcul de l'erreur
    ErrValue  = ErrValue * PID_Kp; // --- calcul de l'action proportionnele
    PID_Integrated = PID_Integrated + (ErrValue * PID_Ki); // l'action intégrale

    if (PID_First_Time)
    {
      PID_First_Time = FALSE;
      OldErrValue = 0;
    }
    DiffValue = (ErrValue - OldErrValue) * PID_Kd; // l'action dérivé
    OldErrValue = ErrValue;
    // --- calcul de l'action combinée ---
    Result = ErrValue + PID_Integrated + DiffValue;
    if (Result < PID_MinOutput) // limiter la sortie
      Result = PID_MinOutput;
    if (Result > PID_MaxOutput)
      Result = PID_MaxOutput;
    return (Result);
  }
  void Init() {
    ADPCFG=   0xFFFFFFFB;      // RB0 en analogique
    TRISB.B0 = 1;             // RB0 en entrée
    TRISC = 0;                // RC7 ensortie (et tout les autre pins du port)
  }
  float   SetPointM=90, InputValueM, commande, pwm_period, current_duty ;
  float   Kp=0.1243, Ki, Kd, Ts=0.1;    // Ts=0.1s

  void main() {
       Init();
      //ADC1_Init_Advanced(_ADC_12bit, _ADC_INTERNAL_REF);
      ADC1_Init();                 // Initialiser ADC

      AD1CHS0bits.CH0SB=2;
      //initialiser le controleur PID
       Ki=Kp*Ts/2; Kd=Kp*0.5/Ts;
       Init_PID(Kp, Ki, Kd, 0, 2);
      //configurer, initialiser et lancer le générateur PWM
     pwm_period = PWM2_MC_Init(1000, 1, 0b00100010, 0);
  PWM2CON1.PEN1L= 1;
  PWM2_MC_Set_Duty(0);
  PWM2_MC_Start();
    do {
       InputValueM = ADC1_Read(2);          // mesurer la température
       InputValueM=500*InputValueM/1024;    // quantification
       commande=PID_Calculate(SetpointM, InputValueM); // calcul de la commande
       current_duty=(commande/2)*pwm_period;//Modulationde lalargeur d'impulsion
       PWM2_MC_Set_Duty(current_duty);
      Delay_ms(100);           //Ts=0.1s
    } while(1);
  }
```

III.4.4 Simulation dans l'ISIS PROTEUS

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.

Après la compilation du programme, on passe à la simulation. La figure ci-après montre le schéma de simulation du programme dans ISIS pour une consigne=90°C.

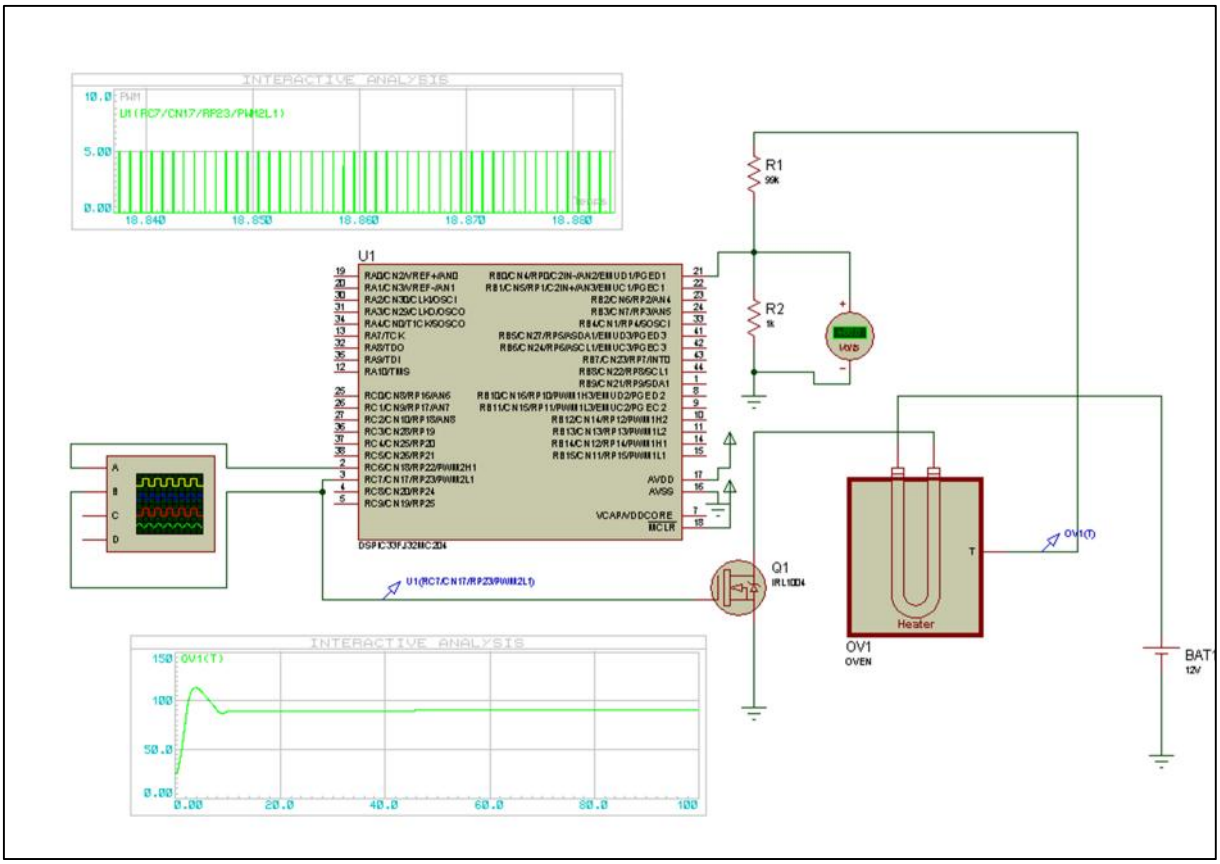


Figure III.6. Schéma de simulation ISIS

III.4.4.1 Résultats de la simulation

La simulation donne le résultat de la figure III.8, on constate bien que le signal obtenue affiche un dépassement au départ. Après quelques secondes la courbe diminue, mais, le système reste oscillant pendant un petit moment, finalement le système se stabilise autour du point désiré(température de 90°C).

Figure III.7. Résultat de la simulation

On résume le fonctionnement du système en 2 phases essentielles :

1ère phase : L'augmentation rapide de la résistance du four a provoqué un dépassement du système, on remarque que le PWM commence à diminuer pour rendre le système stable les figures suivantes montre l'état du four et celui du PWM.

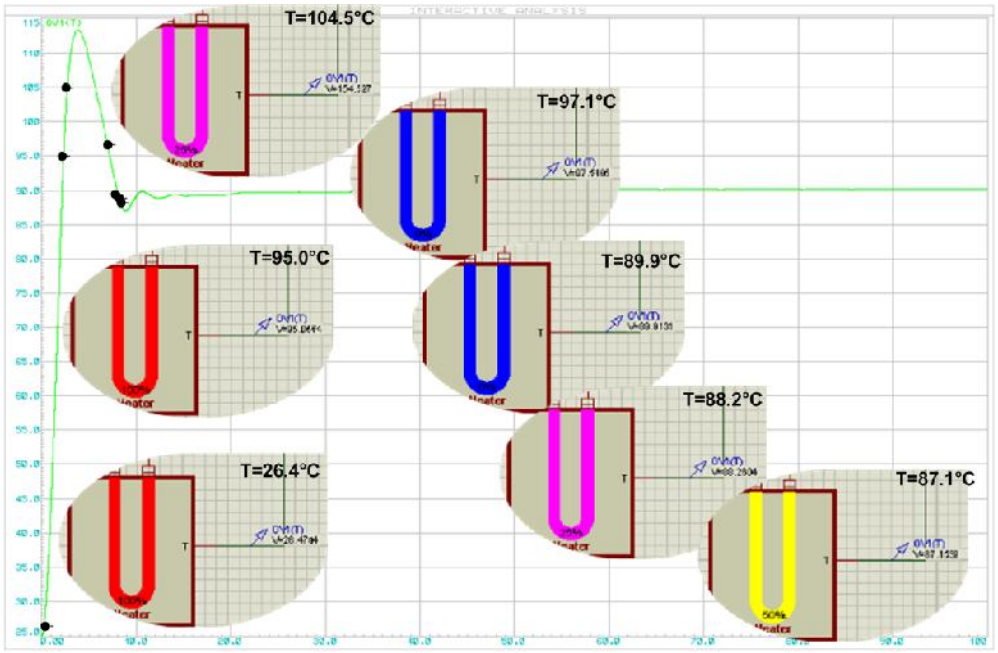


Figure III.8. 1ère phase : état du four.

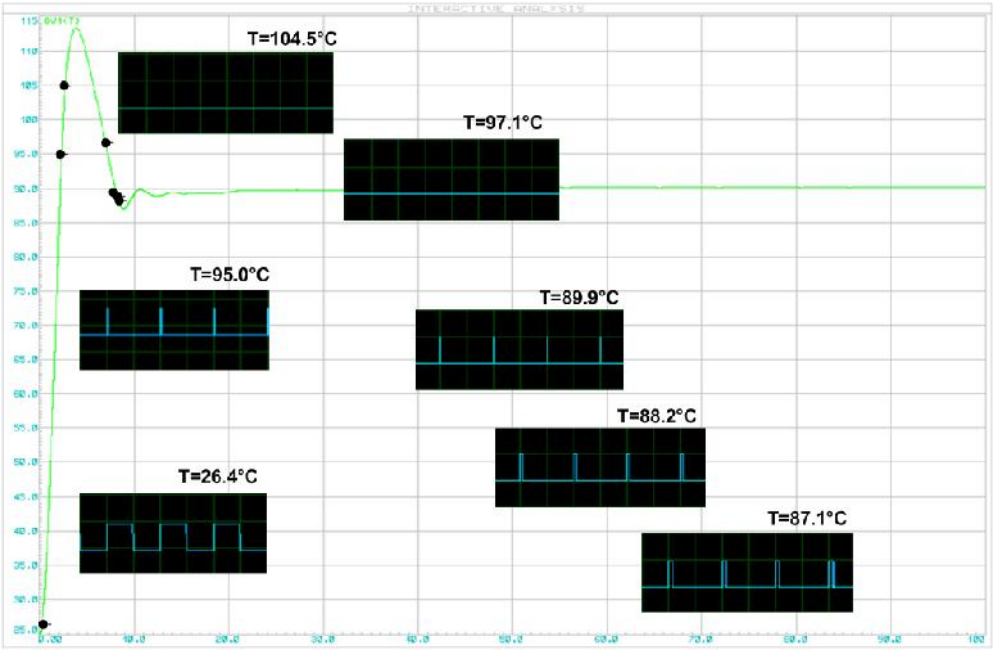


Figure III.9. 1ère phase : état du PMW.

2ème phase : le contrôleur PID a trouvé les paramètres convenables au système stable, le PWM stabilise la fréquence de consigne, ce qui a fait une résistance invariable et une température fixée sur tout le fonctionnement du système (Figure III.12 et 13)

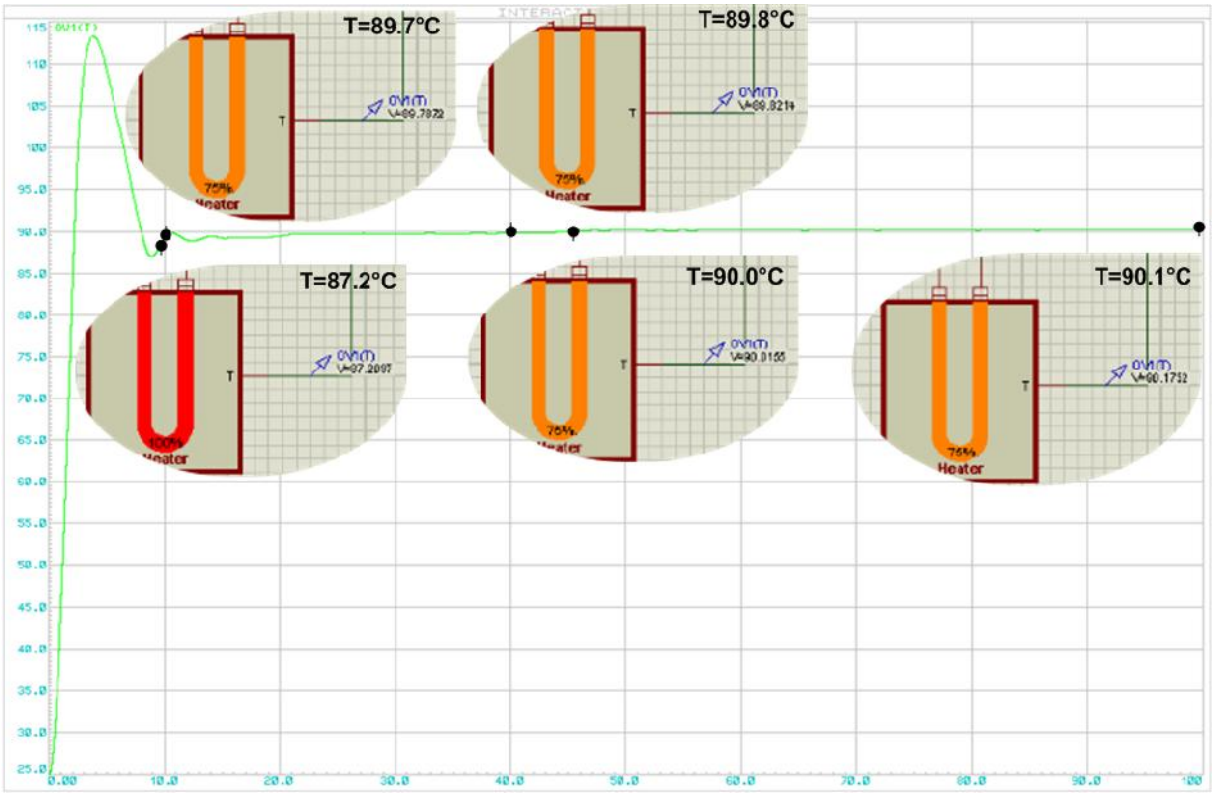


Figure III.10. 2ème phase : état du four.

Figure III.11. 2ème phase : état du PMW.

III.5. Exploitation de la mémoire du dspic

La programme élaboré a exploité seulement 3,7% de la mémoire DATA et 12% de la mémoire RAM (figure III.14), d'où le fait que le composant supporte encore l'implémentation de plus de fonctions de traitement.

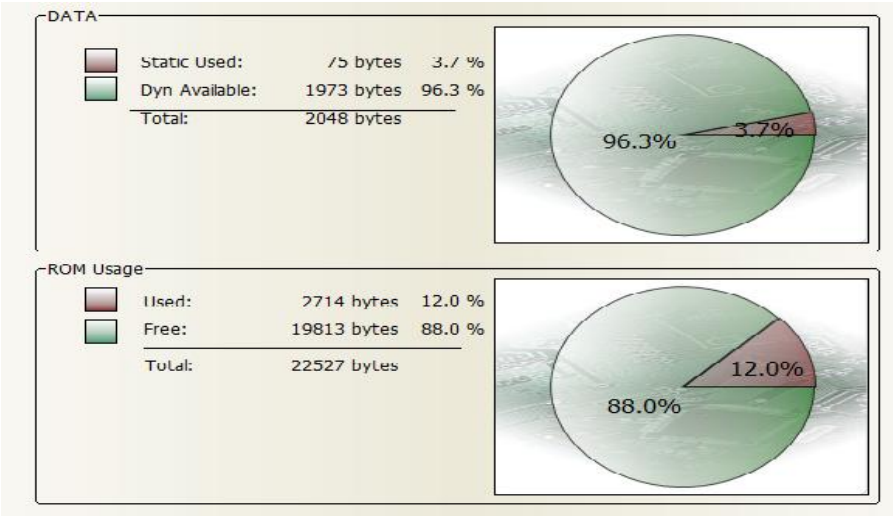


Figure III.12. Exploitation de la mémoire globale

Ainsi, dans la figure qui suit, on donnera le rétablissement de cette exploitation de mémoire par les différentes instructions utilisées:

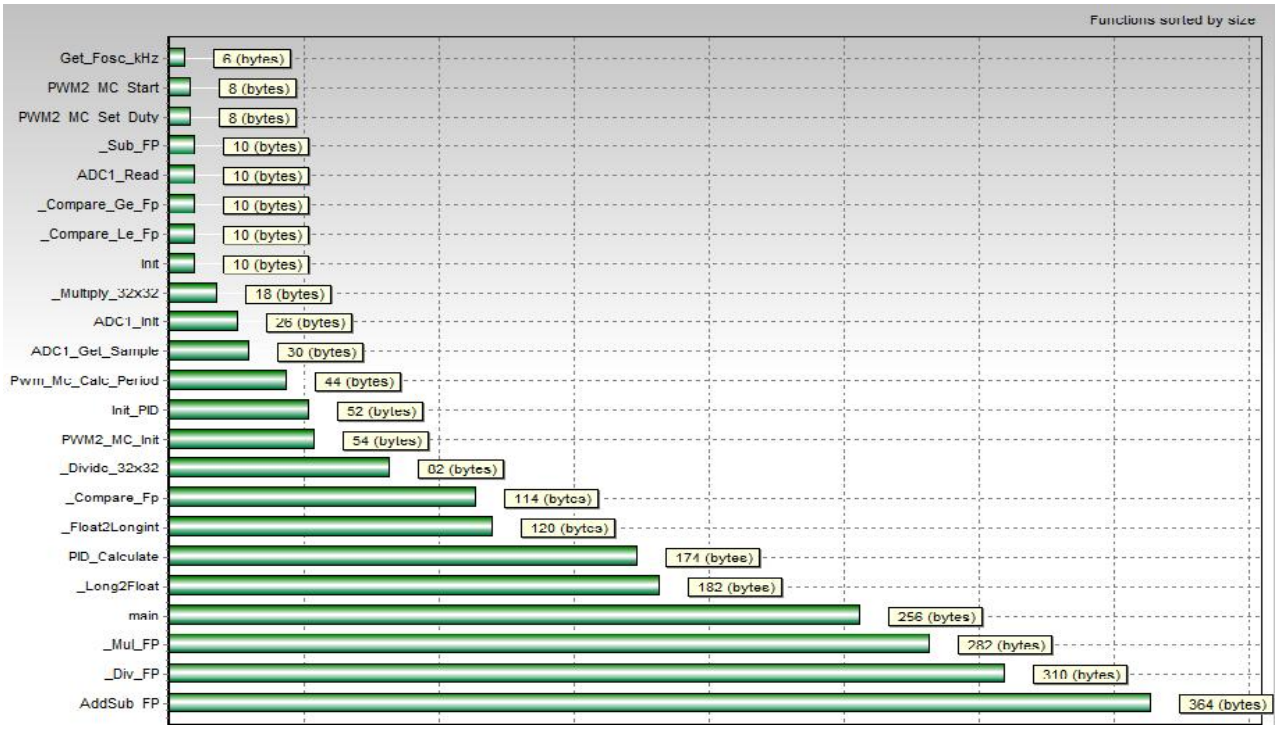


Figure III.13. Répartition de la mémoire selon les fonctions utilisés

III.6. Conclusion

Une description générale de notre système a été présentée dans ce chapitre, suite à quoi on a établi un programme selon le cahier de charge après avoir dressé l'organigramme de fonctionnement. Ce chapitre nous a permis aussi de voir le comportement du contrôleur automatique PID après avoir implémenté le programme sur Proteus-ISIS.

Conclusion général

Conclusion générale

L'automatique se vulgarise de plus en plus, son usage a presque investi tous les domaines. Science et technique clairement promise à remplacer les moteurs mécaniques à énergie fossile, hydrique ou éolienne, le DSPIC est omniprésent au même titre que l'électronique où il est intégré. Écologie et économie sont parmi les facteurs de son essor.

Entre autres systèmes électroniques vitaux sur le plan des commandes, le DSPIC est donc satisfaisant. Nous l'avons éprouvé sur un four électrique pour en parler à juste raison des résultats satisfaisants que nous rapportons ici.

Alliée au dsPIC, la commande PID s'est avéré un complément très appréciable, conjuguant simplicité de maniement et polyvalence. Maitrisant de façon optimale les signaux entrée/sortie, elle s'est avérée tout indiquée à une extension d'usage.

Le PID contrôleur est testé au cours de notre étude sous dsPIC programmé, le four (objet sur lequel porta notre expérience) a pu se stabiliser avec succès à une température quasiment constante de 90°C.

Un dernier argument à son actif, le dsPIC est d'un coût compétitif (à l'achat), fiable sur le plan de la sécurité et, enfin, d'une uniformité de fonctionnement appréciable.

Bibliographie

Référence bibliographique

[1] Daniel Ross & Etienne Deguine & Mickaël Camus, cour Asservissement par PID

[2] Asservissement de position et de vitesse d'une articulation Robotique, Institut Supérieur des sciences appliqués et de technologie De Sousse, disponible sur site www.université virtuelle du Tunis.com

[3] Introduction au dsPIC Digital Signal Controller. Disponible sur site. www.Microchip.com

[4] dsPIC30F High-Performance Digital Signal Controllers. Disponible sur site www.Microchip.com

[5] LOZEAU Mathieu « COMMANDE PAR SUPERVISION DE SYSTÈMES MÉCATRONIQUES VIA INTERNET » UNIVERSITÉ DE MONTRÉAL.

[6] A Simple PID Controller with Adaptive Parameter in a dsPIC; Case of Study. Instituto Politécnico de Setúbal/Campus do IPS, Estefanilha, 2914-508 Setúbal, Portugal