

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE



UNIVERSITE DE M'SILA  
FACULTE DES SCIENCES DE L'INGENIEUR  
DEPARTEMENT D'ELECTROTECHNIQUE

MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU DIPLÔME  
D'INGINIEUR D'ETAT EN GENIE ELECTROTECHNIQUE

**OPTION : ELECTROMECHANIQUE**

**THEME**

---

**REALISATION ET GUIDAGE D'UN AGV**

---

**Proposé et dirigé par :**

- Dr. CHAFAA KHEIREDDINE

**Présenté par :**

- BOUKHARI Samir  
- LAKEHAL Belabes Abdelatif

Année Universitaire : 2006 / 2007

# *Dédicaces*

*Je dédie ce modeste travail à mes très chers parents pour leurs suivis de tous le long de mes études.*

*Je dédie ce travail*

*A mes frères.*

*A Mes sœurs.*

*A Mes grands parents.*

*A toute la famille.*

*A mon encadreur Dr. Chafaa Kheireddine.*

*A mes amis (es), surtout mon binôme, Farid, Zine elaabiddine et*

*Ahmed.*

*A tous mes enseignants.*

*A tous mes collègues de la promotion 2007.*

*Boukhari Samir*

# *Dédicaces*

*Je dédie ce mémoire à mes très chers parents pour leurs soutient morale et financier dans les moments les plus ardu.*

*Je dédie ce travail A*

*Mes sœurs.*

*Mes grands parents.*

*Et a toute la famille, Lakehal et Bouricha.*

*A mon encadreur Dr C.H.A.F.A.A.*

*A mes amis (es), surtout Farid, et mon binôme.*

*A tous mes enseignants.*

*A tous mes collègues la promotion 2007 de l'Electromécanique.*

*Lakehal*

*Belabes Abd Elatif*

# Remerciements

*Nous tenons à remercier tout d'abord Dieu le tout puissant pour la volonté, la santé, la patience, et la force l'accomplir ce travail.*

*Ainsi, nous tenons également à exprimer nos vifs remerciements à notre encadreur Dr. CHAFAA KHEIREDDINE pour avoir d'abord proposée ce thème, pour suivi continuel tout le long de la réalisation de ce mémoire et qui n'a pas cessée de nous donner ses conseils, et la documentation qu'il a fournir.*

*Nos remerciements vont aussi à tous les enseignants ainsi que tous personnels administratif du département de l'électrotechnique et département de l'électronique, Nos remerciements à tous les membres du jury qui ont accepté de juger notre travail.*

*Nous tenons à remercier vivement toutes personnes qui nous ont aidé à élaborer et réaliser ce mémoire, ainsi à tous ceux qui nous ont aidés de près ou de loin à accomplir ce travail.*

*En fin, nous tenons à exprimer notre reconnaissance à tous nos amis et collègues pour le soutien moral et matériel...*

# SOMMAIRE

Introduction générale.....	8
----------------------------	---

## Chapitre I : Robot Mobile AGV

I-1 Introduction.....	11
I-2 Types des robots mobiles.....	12
I-2-1 Les robots mobiles à roues ou à chenilles.....	12
I-2-2 Les robots mobiles à pattes.....	12
I-2-3 Les corps articulés.....	13
I-3 Définition de l'AGV.....	13
I-4 Guidage de l'AGV.....	14
I-5 Utilisation de l'AGV.....	14
I-5-1 Utilisation industrielle.....	14
I-5-2 Utilisation non industrielle.....	15
I-6 Intérêts et avantages d'AGV.....	15
I-7 Les types d'AGV.....	16
I-8 Conclusion.....	17

## Chapitre II : Microcontrôleur Pic 16F84

II-1 Introduction.....	18
II-2 Caractéristiques principales d'un pic.....	19
II-3 Pic 16F84.....	20
II-3-1 Brochage et fonction des pattes.....	21
II-3-2 Architecture générale.....	22
II-4 Organisation de la mémoire.....	23
II-4-1 Mémoire de programme.....	23
II-4-2 Mémoire des données.....	24
II-4-2-1 Registres généraux.....	25
II-4-2-2 Registres spéciaux-SFRs.....	25

II-4-2-3 Mémoire EEPROM.....	28
II-5 Modes d'adressages.....	28
II-5-1 Adressage immédiat.....	28
II-5-2 Adressage direct.....	29
II-5-3 Adressage indirect.....	29
II-6 Ports d'entrées/sorties.....	30
II-6-1 Port A.....	30
II-6-2 Port B.....	31
II-7 Compteur.....	32
II-7-1 Registre TMRO.....	33
II-7-2 Choix de l'horloge.....	33
II-7-3 Pré-diviseur.....	33
II-7-4 Fin de comptage et interruption.....	34
II-7-5 Registres utiles à la gestion de timer0.....	34
II-8 Accès à la mémoire EEPROM.....	35
II-8-1 Registre utilisé.....	35
II-8-2 Lecture.....	36
II-8-3 Ecriture.....	36
II-9 Interruptions.....	37
II-9-1 Mécanisme.....	37
II-9-2 Différentes sources d'interruptions.....	38
II-9-3 Validation des interruptions.....	39
II-10 Chien de garde.....	40
II-10-1 Principe.....	40
II-10-2 Mise en service.....	40
II-10-3 Gestion.....	40
II-10-4 Choix de la durée.....	41
II-11 Conclusion.....	41

## Chapitre III : Moteur à Courant Continu.

III-1 Introduction.....	42
III-2 Généralités.....	43
III-2-1 Lois fondamentales.....	43
III-2-1-1 Loi de Laplace.....	44
III-2-1-2 Loi de Faraday.....	44
III-3 Présentation générale d'un moteur à courant continu.....	45
III-3-1 Constitution.....	45
III-3-2 Constituants électriques.....	46
III-3-2-1 Le stator.....	46
- à aimant permanent.....	46
- à inducteur bobiné.....	46
III-3-2-2 Le rotor.....	46
III-3-2-3 Collecteur.....	47
III-3-2-4 Balais.....	47
III-3-3 Principe de fonctionnement.....	48
III-3-3-1 Couple électromagnétique.....	48
III-3-3-2 Force électromotrice.....	48
III-4 Puissances mises en jeu.....	49
III-4-1 Puissance absorbée.....	49
III-4-2 Puissance électromagnétique.....	49
III-4-3 Puissance perdue.....	49
III-4-4 Puissance utile.....	50
III-4-5 Bilan des puissances.....	50
III-4-6 Rendement.....	51
III-4-7 Causes des pertes et solutions.....	51
III-5 Différents types d'excitation du Mcc.....	52
III-6 Moteur à courant continu de commande.....	52
III-7 Hacheur.....	56
III-7-1 Utilisation de l'hacheur.....	56

III-7-2 Hacheur à quatre quadrants.....	57
III-7-3 La commande de l'hacheur.....	57
III-8 Conclusion.....	58

## **Chapitre IV: Conception et Réalisation de l'AGV**

IV-1 Introduction.....	59
IV-2 Structure mécanique du véhicule.....	60
IV-2-1 Le moteur à courant continu utilisé.....	60
IV-2-2 Dimensions du véhicule et montage des moteurs.....	61
IV-2-3 Principe de guidage.....	62
IV-3 Conception et réalisation des cartes électroniques.....	63
IV-3-1 Modules d'alimentations.....	63
IV-3-2 Carte de commande.....	66
IV-3-3 Opto-coupleurs.....	67
IV-3-4 Hacheurs.....	68
IV-4 Circuits imprimés.....	71
IV-4-1 Circuit imprimé d'alimentation.....	71
IV-4-2 Circuit imprimé d'une carte d'isolation et d'un hacheur.....	71
IV-4-3 Circuit imprimé de la carte de commande.....	71
IV-5 Conclusion.....	73
Conclusion générale.....	74

# Introduction générale

L'homme a, de toute temps et par divers moyens, tenté d'accroître sa capacité de production et de libérer de tout travail monotone et musculaire. Dans cette optique, il a fabriqué des machines de plus en plus perfectionnées, qui permettent d'accomplir des fonctions complexes.

Le développement considérable de la technologie permet aujourd'hui de réaliser des véhicules "sans opérateur" appelés véhicules à guidage automatique ou AGV (en anglais, Automated Guided Vehicle).

En générale, l'automatisation de ces véhicules vise à limiter au maximum l'intervention de l'opérateur, particulièrement dans les milieux hostiles. C'est le cas par exemple des industries chimique et nucléaire. Ces véhicules présentent un avantage majeur en l'occurrence un fonctionnement fiable et un prix de revient relativement faible.

La grande variété des produits (assemblés, usines,...), et la réduction des délais de livraison imposent des changements de série encore plus fréquents. Il est alors indispensable de disposer d'ateliers flexibles ou FMS (flexible manufacturing système).

Les FMS permettent à l'industrie de maintenir sa compétitivité par l'accroissement de sa productivité, de sa rentabilité et de sa flexibilité.

Cette dernière propriété est d'autant plus accentuée que les relations entre éléments de ces systèmes sont développées.

Depuis un certain nombre d'années, les AGV sont devenus partie intégrante des ateliers flexibles. De part leur capacité de transporter des charges plus ou moins lourdes à travers un chemin souvent complexe, les AGV s'imposent comme éléments essentiels de ces ateliers. [7]

Notre objectif à travers ce travail et de nous introduire au monde de la robotique. Dans ce cadre nous avons réalisé un robot mobile (AGV) guidé par une carte à base de microcontrôleur PIC. Le véhicule réalisé fonctionne avec des moteurs à courant continu. Nous considérons ce travail multidisciplinaire, car il touche plusieurs domaines du génie électrique tel que : L'électronique, l'électrotechnique, la mécanique et l'informatique. Plusieurs cartes de commande ont été réalisées : Cartes d'alimentations, carte de commande, cartes de puissances et cartes d'isolation optique.

Le but initial a été de réaliser un AGV qui se corrige automatiquement lorsqu'il se déplace et cela en introduisant des données de sa position actuelle, donc nous pouvons dire qu'en faite, ce travail a été divisé en deux parties, partie acquisition et partie commande. Dans ce travail, nous avons réalisé la deuxième partie et malheureusement, la première partie n'a pas été réalisée.

Le plan du mémoire est comme suit :

Le premier chapitre est consacré à l'étude des véhicules AGV.

Les microcontrôleurs PIC ont été introduits dans le deuxième chapitre.

Dans le troisième chapitre nous avons présenté les moteurs à courant continu ainsi que la théorie des hacheurs.

La conception et la réalisation du véhicule sont présentées dans le chapitre quatre.

Enfin, nous terminons notre mémoire par une conclusion, une bibliographie et des annexes.

# Robots Mobiles AGV

### I-1 Introduction

L'utilisation des robots mobiles est aujourd'hui couramment envisagée pour l'automatisation de nombreuses tâches. Celles-ci sont particulièrement diversifiées: le nettoyage, le transport dans les ateliers automatisés, l'agriculture, l'exploitation des mines, l'assistance aux personnes handicapées et l'exploration de milieux hostiles en sont quelques exemples. L'architecture mécanique d'un robot mobile dépend très largement des spécificités de la mission à effectuer et de l'environnement de travail. Ces données conditionnent entre autres le choix d'un système de locomotion approprié. Trois configurations fondamentales peuvent être distinguées: les robots à roues ou à chenilles, les robots à pattes et les corps articulés. Des systèmes hybrides sont également possibles. [5]

## I-2 Types des robots mobiles

### I-2-1 Les robots à roues ou à chenilles

Sont les plus répandus; ils sont aisés à commander grâce au nombre restreint de degrés de libertés (le plus souvent deux) et sont très efficaces sur un sol plat ou modérément accidenté dans le cas des chenilles. Ils sont largement utilisés dans l'industrie pour le transport et la manutention automatisée dans les ateliers et les entrepôts (voir Figs. I.1, I.2).

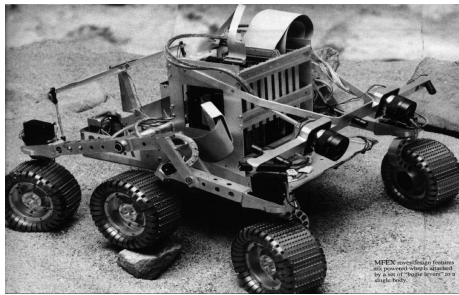


Fig.I.1 Robot mobile à roues

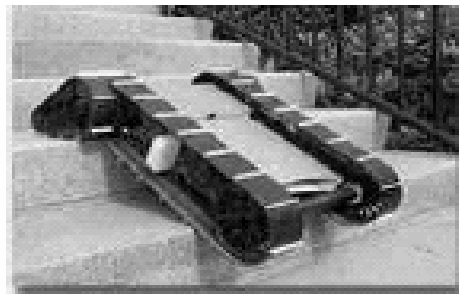


Fig.I.2 Robot mobile à chenilles

### I-2-2 Les robots à pattes

Sont considérés pour leur mobilité supérieure obtenue grâce à un nombre élevé de degrés de liberté (voir Fig.I.3). Le contact avec le sol est discret, ce qui permet une sélection des points d'appui en fonction des conditions locales du terrain. La charge utile est relativement faible, essentiellement à cause d'une puissance installée intrinsèquement élevée. Divers prototypes expérimentaux ont été mis au point, la plupart du temps, pour des activités d'inspection ou d'exploration de milieux peu accessibles. Néanmoins, malgré leurs possibilités attrayantes en matière de mobilité, l'intérêt de l'industrie pour ce type de véhicule se révèle être, à ce jour, encore très limité.

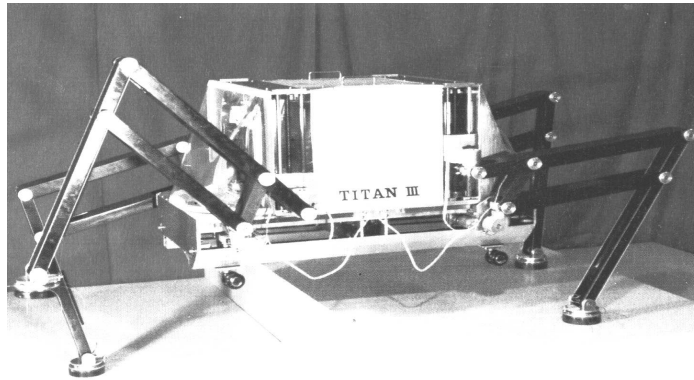


Fig.I.3 Robot à pattes

### **I-2-3 Les corps articulés**

Sont constitués de plusieurs modules élémentaires disposant de plusieurs degrés de liberté l'un par rapport à l'autre. Certaines configurations de ce type permettent une mobilité analogue à celle du serpent (mobilité péristaltique). Parmi diverses applications possibles de ce type de robot, on peut citer la maintenance des sites nucléaires et l'inspection des tuyauteries. [5]

### **I-3 Définition de l'AGV**

Un AGV est un véhicule guidé automatiquement. Il est doté d'un équipement de guidage automatique qu'il soit électromagnétique ou optique. Ce type de véhicule est capable de suivre des chemins prédéfinis selon le besoin.

L'association des industries robotique définit le robot comme un manipulateur multifonctionnel programmable mis en place pour faire bouger des matériaux, des parties des outils ou des spécialisés à travers de véritables mouvements programmés pour l'accomplissement d'une variété de tâche.

L'AGV étant programmable et dirigé par ordinateur peut paraître similaire à un robot, mais le fait que l'AGV doit suivre un chemin de guidage le différencie du robot industriel conventionnel.

L'AGV peut être considéré comme un robot qui a un nombre limité de degrés de liberté (1 à 3). La plupart des robots industriels utilisent 5 à 6 degrés de liberté. Le

fait que l'AGV ne peut pas être considéré comme un manipulateur (sauf si le véhicule a un bras attaché) le distingue aussi du robot industriel.[1]

#### **I-4 guidage de l'AGV**

Le guidage du véhicule est réalisé de trois manières :

- Guidage mécanique (rail).
- Guidage par marques.
- Apprentissage des trajectoires.
- Guidage programmé

#### **I-5 Utilisations de l'AGV**

##### **I-5-1 Utilisation industrielle**

Les AGVs sont utiles dans toute industrie où il y a nécessité de transporter de la matière, dans la manutention et en cours de la fabrication. Ils se distinguent en plus par la possibilité de leur utilisation dans des situations où l'être humain n'est pas bien adapté.

- Ils sont utilisés pour remplacer l'homme dans l'exécution de tâches dangereuses. Par exemple, dans l'industrie nucléaire où des usines automatiques mettent en œuvre des AGVs pour le transport du plutonium retraité, substance de haute toxicité.

Jusqu'à l'avènement de la robotique, ces tâches étaient accomplies par des spécialistes de la manutention munis de combinaisons de protection très élaborées.

- Les AGVs industriels s'acquittent parfaitement des tâches gênantes ou polluantes pour l'individu.

- Les tâches répétitives, considérées comme ennuyeuses ou épuisantes pour l'homme, constituent un champ idéal d'application pour les AGVs.

### **I-5-2 Utilisation non industrielle**

Les AGV sont aussi utilisés dans d'autres activités citons par exemple :

- applications à caractère social : aide pour les handicapés, distribution des repas ou de documents, transport public, etc ...
- application agricoles : arrosage, labour, ...
- applications domestiques : aspirateurs, fauchage de pelouses, ...
- applications militaires et spatiales : exploration ... [1]

### **I-6 Intérêts et avantages des AGV**

L'avènement des AGV dans l'industrie n'est pas le fait du hasard. Dans beaucoup de situations de manutention de matériels, ils se sont avérés comme le moyen de transport le plus économique. Relativement à d'autres techniques traditionnelles de transport, les AGV présentent les avantages suivants :

- **Flexibilité** : Aptitude d'être adaptable à toute situation (changement du produit à concevoir, modification ou extension de l'unité de fabrication).
- **Réduction des frais** : Possibilité d'opérer dans des conditions défavorables à l'homme (réduction de l'énergie gaspillée pour l'éclairage et la chaleur, élimination du travail de groupe...).
- **Productivité et efficacité élevées** : Dues en particulier à un transport à flux régulier, sans arrêt.
- **Meilleures opérations de stocks, contrôle et inventaire.**
- **Fiabilité élevée** : La disponibilité d'un certain nombre d'AGVs en réserve, permet d'assurer le relais pour une éventuelle maintenance.
- **Moins de produits endommagés** : Du fait que les AGVs suivent un chemin prédéfini, minimisant grandement les collisions.

- **Conservations de l'énergie et de l'environnement** : les AGVs n'exigent pas beaucoup d'énergie et fonctionnent sans polluer l'environnement (bruit, fumés).
- **Sécurité améliorée [8]**

### I-7 Types d'AGVs

Suivant l'application envisagée et le domaine d'action, on rencontre (fig.I-4) :

- AGV tracteurs.
- AGV pour charges unitaires.
- AGV tracteur de palettes.
- AGV élévateur à fourche.
- AGV pour charge légère.
- AGV pour ligne d'assemblage.

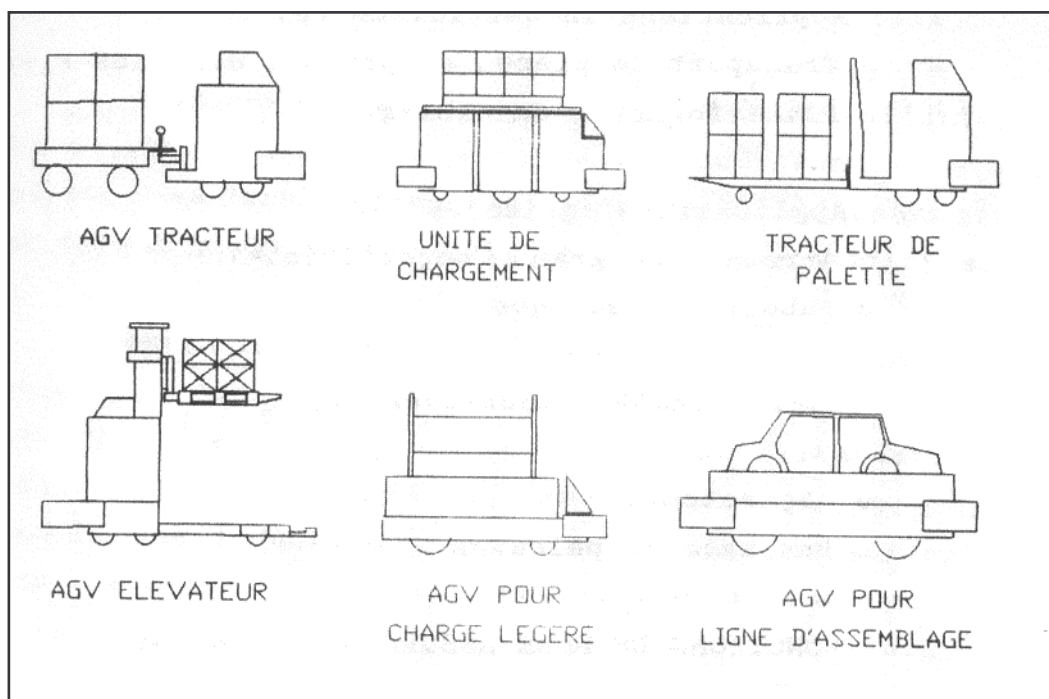


Fig.I-4 Différents types d'AGV

## **I-8 Conclusion**

Les notions des véhicules guidés automatiquement ont été introduites dans ce chapitre. Nous avons exposé leurs intérêts dans le domaine industriel ainsi que dans le domaine domestique, nous avons également cité leurs types et leurs applications.

# Micro contrôleur pic16F84

### II-1 Introduction

Depuis l'arrivée du premier microprocesseur dans les années 70 (déjà plus de 30 ans), conçu par INTEL<sup>™</sup> les progrès en intégration de composants n'ont cessé d'évoluer, faisant de ce fait augmenter la puissance de calcul et la rapidité des microprocesseurs.

Un microprocesseur peut être assimilé à un circuit logique complexe exécutant une à une des ordres (instructions) enregistrés dans une mémoire de programme externe.

Un microprocesseur ne fonctionne jamais seul, on lui associe toujours des périphériques d'entrées-sorties afin de pouvoir exécuter une fonction particulière et dialoguer avec "l'extérieure" (prise en compte de l'état d'un capteur, commande d'un relais, mémoire programme de l'application, etc...). Le grand avantage de cette logique programmable est que la modification d'une fonction ou d'une tâche

ne nécessite pas de câblage supplémentaire, mais uniquement un nouveau programme à loger en mémoire.

L'arrivée de micro contrôleur qui est un microprocesseur auquel on a intégré les périphérique d'entrées sorties, va faire que les montages et applications deviennent encore plus simple à mettre en œuvre, avec un gain de temps, de vitesse et un coût réduit.

Le pic 16F84 de la société Microchip est un micro contrôleur faisant partie de la famille RISC (reduced instruction set computer) dont les caractéristiques sont la vitesse d'excitation et le jeu d'instruction réduit (le pic 16F84 ne comporte que 35 instructions ).

## **II-2 Caractéristiques principales d'un pic**

Un PIC est un microcontrôleur de la société Microchip. Ses caractéristiques principales sont :

### **Séparation des mémoires de programme et de données (architecture Harvard)**

: On obtient ainsi une meilleure bande passante, les instructions et les données pas forcément codées sur le même nombre de bits.

Communication avec l'extérieur seulement par des ports : il ne possède pas de bus d'adresses, de bus de données et de bus de contrôle comme la plupart des microprocesseurs.

**Utilisation d'un jeu d'instructions réduit**, d'où le nom de son architecture : RISC (Reduced Instructions Set Construction). Les instructions sont ainsi codées sur un nombre réduit de bits, ce qui accélère l'exécution (1 cycle machine par instruction sauf pour les sauts qui requièrent 2 cycles). En revanche, leur nombre limité oblige à se restreindre à des instructions basiques, contrairement aux systèmes d'architecture CISC (Complex Instructions Set Construction) qui proposent plus d'instructions donc codées sur plus de bits mais réalisant des traitements plus complexes. [4]

Il existe trois familles de PIC :

- Base-Line : Les instructions sont codées sur 12 bits
- Mid-Line : Les instructions sont codées sur 14 bits
- High-End : Les instructions sont codées sur 16 bits

Un PIC est identifié par un numéro de la forme suivant : xx(L)XXyy –zz

- xx : Famille du composant (12, 14, 16, 17, 18)
- L : Tolérance plus importante de la plage de tension
- XX : Type de mémoire de programme

C - EPROM ou EEPROM

CR - PROM

F - FLASH

- yy : Identification

- zz : Vitesse maximum du quartz

Nous utiliserons un PIC 16F84 –10, soit :

- 16 : Mid-Line
- F : FLASH
- 84 : Type
- 10 : Quartz à 10MHz au maximum

### II-3 PIC 16F84

Device	Program Memory (words)	Data RAM (bytes)	Data EEPROM (bytes)	Max. Freq (MHz)
PIC16F83	512 Flash	36	64	10
PIC16F84	1 K Flash	68	64	10
PIC16CR83	512 ROM	36	64	10
PIC16CR84	1 K ROM	68	64	10

Fig.II.1: Liste des composants.

## Principales caractéristiques

- 35 instructions
- Instructions codées sur 14 bits
- Données sur 8 bits
- 1 cycle machine par instruction, sauf pour les sauts (2 cycles machine)
- Vitesse maximum 10 MHz soit une instruction en 400 ns (1 cycle machine = 4 cycles d'horloge)
- 4 sources d'interruption
- 1000 cycles d'effacement/écriture pour la mémoire flash, 10.000.000 pour la mémoire de donnée EEPROM

### II-3-1 Brochage et fonction des pattes

La Figure.II.2 montre le brochage du pic 16F84. Les fonctions des pattes sont les suivantes :

- $V_{SS}$ ,  $V_{DD}$  : Alimentation
- OSC1,2 : Horloge
- RA0-4 : Port A
- RB0-7 : Port B
- T0CKL : Entrée de comptage
- INT : Entrée d'interruption
- MCLR : Reset : 0V

Choix du mode programmation : 12V - 14V

exécution : 4.5V - 5.5V

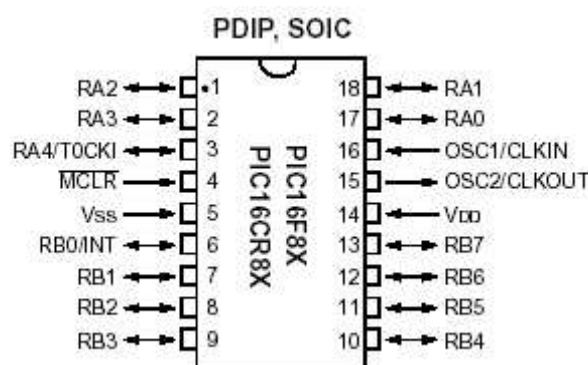


Fig. II.2: Brochage du pic.

### II-3-2 Architecture générale

La Figure.II.3 présente l'architecture générale du circuit. Il est constitué des éléments suivants :

- un système d'initialisation à la mise sous tension (power-up timer, ...)
- un système de génération d'horloge à partir du quartz externe (timing génération)
- une unité arithmétique et logique (ALU)
- une mémoire flash de programme de 1k "mots" de 14 bits.
- un compteur de programme (program counter) et une pile.
- un bus spécifique pour le programme.
- un registre contenant le code de l'instruction à exécuter
- un bus spécifique pour les données.
- une mémoire RAM contenant
- les SFR.
- 68 octets de données.
- une mémoire EEPROM de 64 octets de données.
- 2 ports d'entrées/sorties.
- un compteur (timer).
- un chien de garde (watchdog)

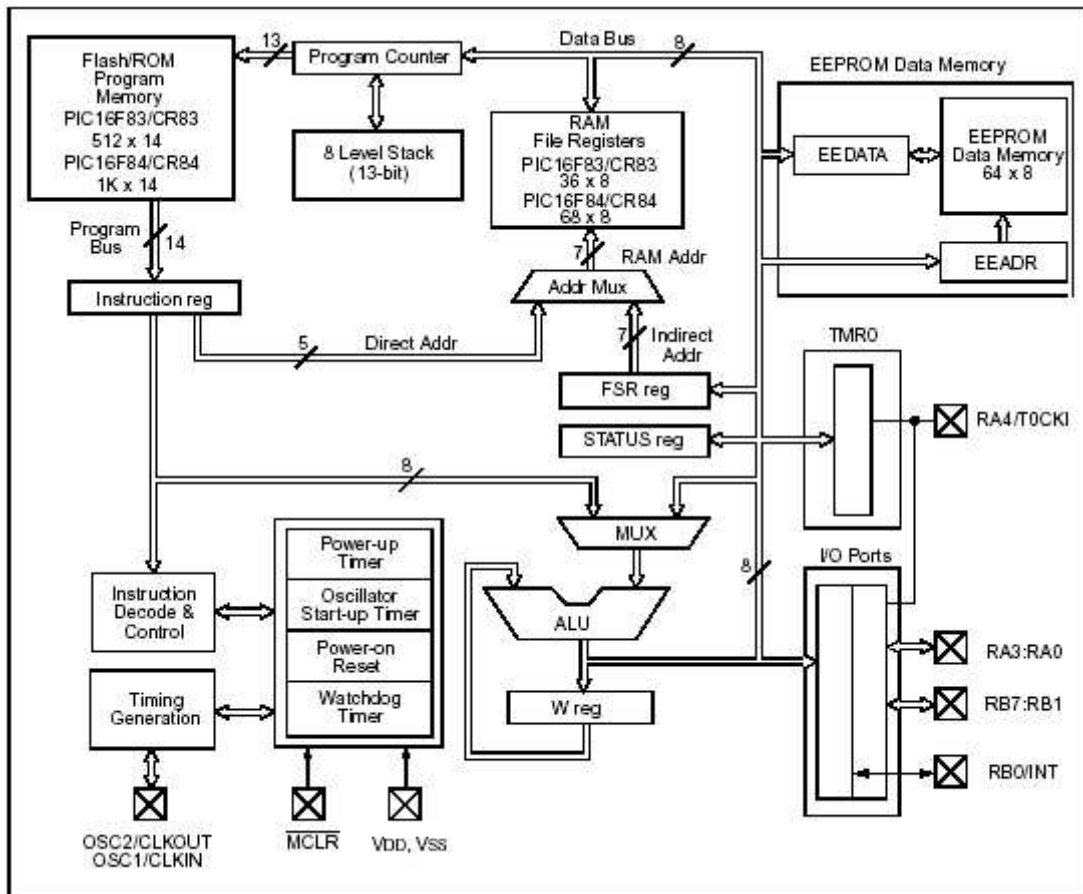


Fig.II.3: Architecture générale du PIC 16F8X.

## II-4 Organisation de la mémoire

Le PIC contient une mémoire de programme et une mémoire de données. La structure Harvard des PICs fournit un accès séparé à chacune. Ainsi, un accès aux deux est possible pendant le même cycle machine.

### II-4-1 Mémoire de programme

C'est elle qui contient le programme à exécuter. Ce dernier est téléchargé par liaison série. La Figure.III.1 montre l'organisation de cette mémoire. Elle contient 1k "mots" de 14 bits dans le cas du PIC 16F84, même si le compteur de programme (PC) de 13 bits peut en adresser 8k. Il faut se méfier des adresses images ! L'adresse 0000h contient le vecteur du reset, l'adresse 0004h l'unique vecteur d'interruption du PIC.

La pile contient 8 valeurs. Comme le compteur de programme, elle n'a pas d'adresse dans la plage de mémoire. Ce sont des zones réservées par le système.

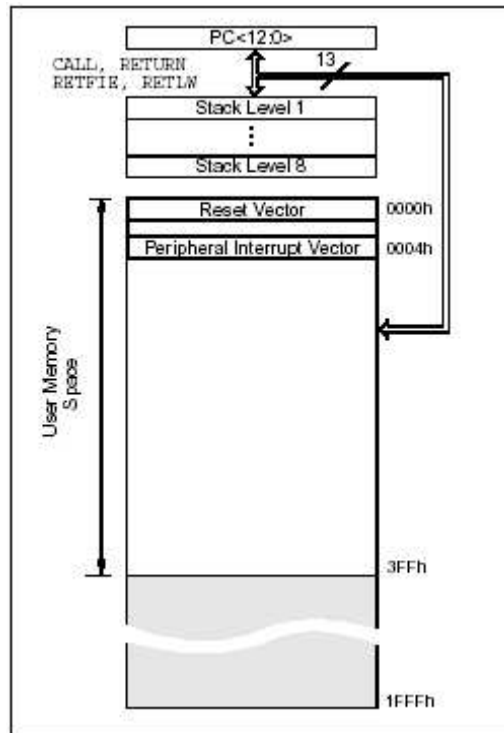


Fig.II.4 : Organisation de la mémoire de programme et de la pile.

#### II-4-2 Mémoire de données

Elle se décompose en deux parties de RAM (Figure.III.2) et une zone EEPROM. La première contient les SFRs (Special Function Registers) qui permettent de contrôler les opérations sur le circuit. La seconde contient des registres généraux, libres pour l'utilisateur. La dernière contient 64 octets.

les instructions orientées octets ou bits contiennent une adresse sur 7 bits pour désigner l'octet avec lequel l'instruction doit travailler. D'après la Figure.III.2, l'accès au registre TRISA d'adresse 85h, par exemple, est impossible avec une adresse sur 7 bits. C'est pourquoi le constructeur a défini deux banques. Le bit RP0 du registre d'état (STATUS.5) permet de choisir entre les deux. Ainsi, une adresse sur 8 bits est composée de RP0 en poids fort et des 7 bits provenant de l'instruction à exécuter.

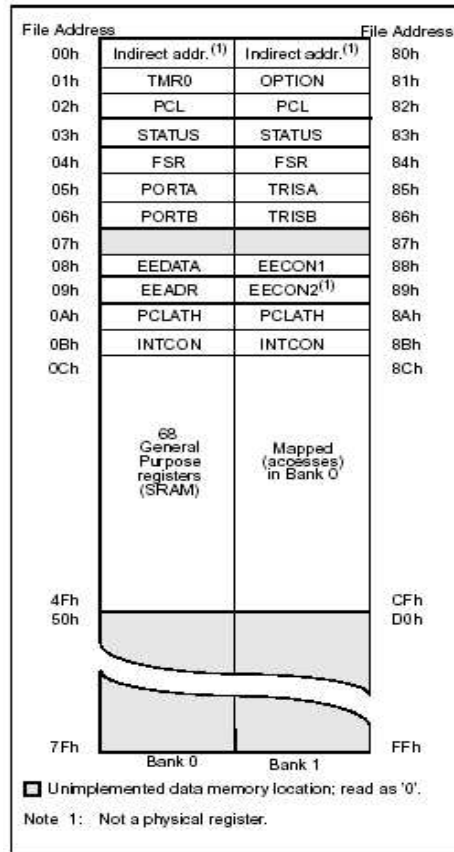


Fig.II.5 : Organisation de la mémoire de données.

### II-4-2-1 Registres généraux

Ils sont accessibles soit directement soit indirectement à travers les registres FSR et INDF.

### II-4-2-2 Registres spéciaux - SFRs

Ils permettent la gestion du circuit. Certains ont une fonction générale, d'autres une fonction spécifique attachée à un périphérique donné. La Figure.III.3 donne la fonction de chacun des bits de ces registres. Ils sont situés de l'adresse 00h à l'adresse 0Bh dans la banque 0 et de l'adresse 80h à l'adresse 8Bh dans la banque 1. Les registres 07h et 87h n'existent pas.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets (Note 3)
Bank 0											
00h	INDF	Uses contents of FSR to address data memory (not a physical register)								----	----
01h	TMR0	8-bit real-time clock/counter								xxxx xxxx	uuuu uuuu
02h	PCL	Low order 8 bits of the Program Counter (PC)								0000 0000	0000 0000
03h	STATUS <sup>(2)</sup>	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
04h	FSR	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
05h	PORTA	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	---u uuuu
06h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	uuuu uuuu
07h		Unimplemented location, read as '0'								-----	-----
08h	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC <sup>(1)</sup>			---	0000	---	0000
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
Bank 1											
80h	INDF	Uses contents of FSR to address data memory (not a physical register)								----	----
81h	OPTION_REG	REPO	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
82h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	0000 0000
83h	STATUS <sup>(2)</sup>	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	000q quuu
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	uuuu uuuu
85h	TRISA	—	—	—	PORTA data direction register			---	1111	---	1111
86h	TRISB	PORTB data direction register								1111 1111	1111 1111
87h		Unimplemented location, read as '0'								-----	-----
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 q000
89h	EECON2	EEPROM control register 2 (not a physical register)								-----	-----
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC <sup>(1)</sup>			---	0000	---	0000
0Bh	INTCON	GIE	EEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = value depends on condition.

Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a slave register for PC<12:8>. The contents of PCLATH can be transferred to the upper byte of the program counter, but the contents of PC<12:8> is never transferred to PCLATH.

2: The  $\overline{TO}$  and  $\overline{PD}$  status bits in the STATUS register are not affected by a  $\overline{MCLR}$  reset.

3: Other (non power-up) resets include: external reset through  $\overline{MCLR}$  and the Watchdog Timer Reset.

Fig.II.6: Description des SFR.

INDF (00h - 80h) : Utilise le contenu de FSR pour l'accès indirect à la mémoire.

TMR0 (01h) : Registre lié au compteur.

PCL (02h - 82h) : Contient les poids faibles du compteur de programmes (PC). Le registre PCLATH (0Ah-8Ah) contient les poids forts.

STATUS (03h - 83h) : Il contient l'état de l'unité arithmétique et logique ainsi que les bits de sélection des banques.

FSR (04h - 84h) : Permet l'adressage indirect.

PORTA (05h) : Donne accès en lecture ou écriture au port A, 5 bits. Les sorties sont à drain ouvert. Le bit 4 peut être utilisé en entrée de comptage.

PORTB (06h) : Donne accès en lecture ou écriture au port B. Les sorties sont à drain ouvert. Le bit 0 peut être utilisé en entrée d'interruption.

EEDATA (08h) : Permet l'accès aux données dans la mémoire EEPROM.

EEADR (09h) : Permet l'accès aux adresses de la mémoire EEPROM.

PCLATCH (0Ah - 8Ah) : Donne accès en écriture aux bits de poids forts du compteur de programme.

INTCON (0Bh - 8Bh) : Masque d'interruptions.

OPTION\_REG (81h) : Contient des bits de configuration pour divers périphériques.

TRISA (85h) : Indique la direction (entrée ou sortie) du port A.

TRISB (86h) : Indique la direction (entrée ou sortie) du port B.

EECON1 (88h) : Permet le contrôle d'accès à la mémoire EEPROM.

EECON2 (89h) : Permet le contrôle d'accès à la mémoire EEPROM.

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	TO	PD	Z	DC	C
bit7							bit0
<p>bit 7: <b>IRP</b>: Register Bank Select bit (used for indirect addressing)            0 = Bank 0, 1 (00h - FFh)            1 = Bank 2, 3 (100h - 1FFh)            The IRP bit is not used by the PIC16F8X. IRP should be maintained clear.</p> <p>bit 6-5: <b>RP1:RP0</b>: Register Bank Select bits (used for direct addressing)            00 = Bank 0 (00h - 7Fh)            01 = Bank 1 (80h - FFh)            10 = Bank 2 (100h - 17Fh)            11 = Bank 3 (180h - 1FFh)            Each bank is 128 bytes. Only bit RP0 is used by the PIC16F8X. RP1 should be maintained clear.</p> <p>bit 4: <b>TO</b>: Time-out bit            1 = After power-up, CLRWDT instruction, or SLEEP instruction            0 = A WDT time-out occurred</p> <p>bit 3: <b>PD</b>: Power-down bit            1 = After power-up or by the CLRWDT instruction            0 = By execution of the SLEEP instruction</p> <p>bit 2: <b>Z</b>: Zero bit            1 = The result of an arithmetic or logic operation is zero            0 = The result of an arithmetic or logic operation is not zero</p> <p>bit 1: <b>DC</b>: Digit carry/borrow bit (for ADDWF and ADDLW instructions) (For borrow the polarity is reversed)            1 = A carry-out from the 4th low order bit of the result occurred            0 = No carry-out from the 4th low order bit of the result</p> <p>bit 0: <b>C</b>: Carry/borrow bit (for ADDWF and ADDLW instructions)            1 = A carry-out from the most significant bit of the result occurred            0 = No carry-out from the most significant bit of the result occurred</p> <p>Note: For borrow the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low order bit of the source register.</p>							
<p>R = Readable bit            W = Writable bit            U = Unimplemented bit, read as '0'            - n = Value at POR reset</p>							

Fig.II.7: Registre d'étai du PIC - STATUS.

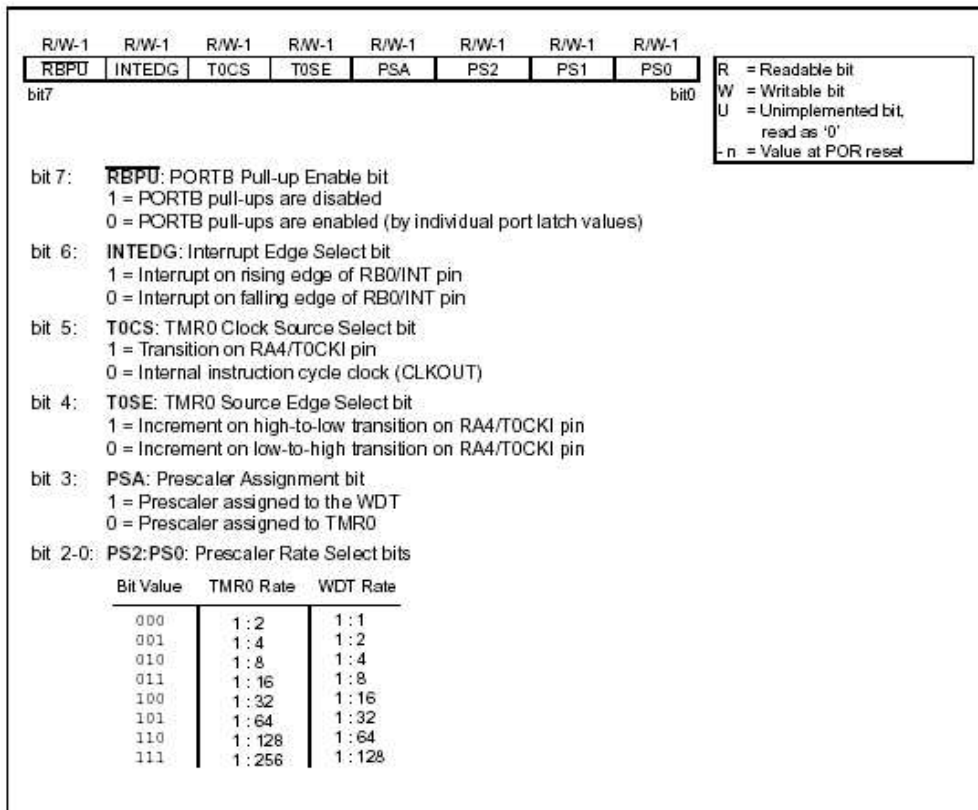


Fig.II.8: Registre de configuration de périphériques - OPTION\_REG.

### II-4-2-3 Mémoire EEPROM

Le PIC possède une zone EEPROM de 64 octets accessibles en lecture et en écriture par le programme. On peut y sauvegarder des valeurs, qui seront conservées même si l'alimentation est éteinte, et les récupérer lors de la mise sous tension. Leur accès est spécifique et requiert l'utilisation de registres dédiés.

## II-5 Modes d'adressages

On ne peut pas concevoir un programme qui ne manipule pas de données. Il existe trois grands types d'accès à une donnée ou modes d'adressage :

### II-5-1 Adressage immédiat

La donnée est contenue dans l'instruction.

Exemple : `movlw 0xC4` ; **Transfert la valeur 0xC4** dans W

## II-5-2 Adressage direct

La donnée est contenue dans un registre. Ce dernier peut être par un nom (par exemple W) ou une adresse mémoire.

Exemple : `movf 0x2B, 0` ; **Transfert dans W** la valeur **contenue à l'adresse 0x2B**.

L'adresse 0x2B peut correspondre à 2 registres en fonction de la banque choisie. Le bit RP0 permet ce choix, le bit RP1 étant réservé pour les futurs systèmes à 4 banques.

## II-5-3 Adressage indirect

L'adresse de la donnée est contenue dans un pointeur. Dans les PIC, un seul pointeur est disponible pour l'adressage indirect : FSR. Contenu à l'adresse 04h dans les deux banques, il est donc accessible indépendamment du numéro de banque. En utilisant l'adressage direct, on peut écrire dans FSR l'adresse du registre à atteindre. FSR contenant 8 bits, on peut atteindre les deux banques du PIC 16F84. Pour les PIC contenant quatre banques, il faut positionner le bit IRP du registre d'état qui sert alors de 9<sup>ème</sup> bit d'adresse (Fig.II.9).

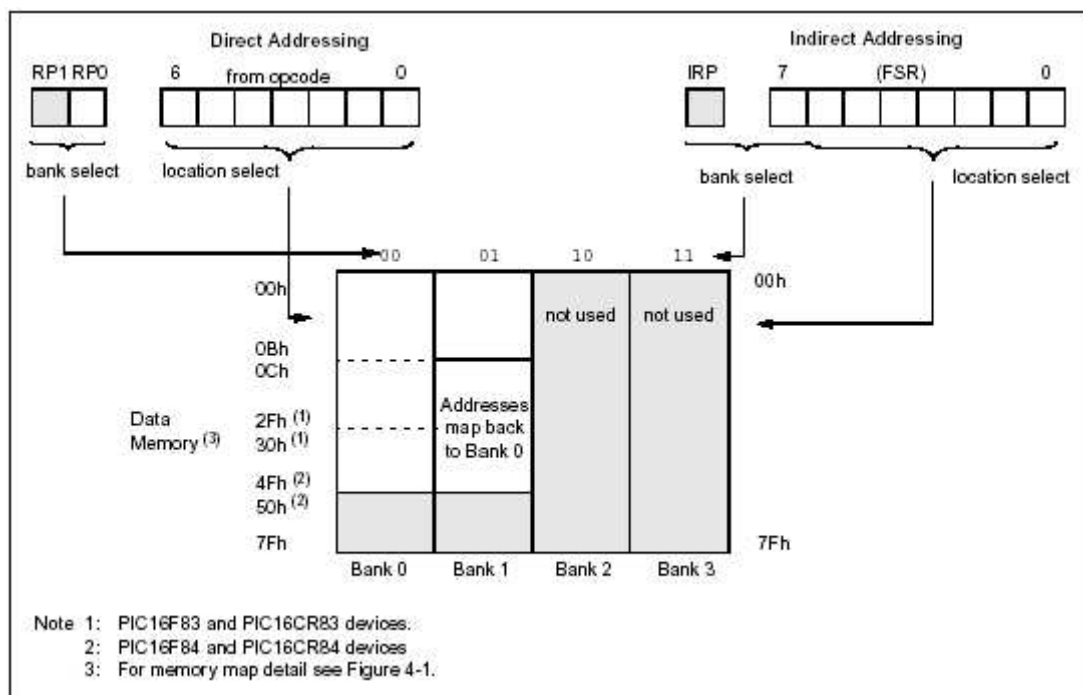


Fig.II.9: Adressages direct et indirect à la mémoire de données.

L'accès au registre d'adresse contenue dans FSR se fait en utilisant le registre INDF. Il se trouve à l'adresse 0 dans les deux banques. Il ne s'agit pas d'un registre physique. On peut le voir comme un autre nom de FSR, utilisé pour accéder à la donnée elle-même, FSR servant à choisir l'adresse.

Exemple : `movlw 0x1A ; Charge 1Ah dans W`

`movwf FSR ; Charge W, contenant 1Ah, dans FSR`

`movf INDF, 0 ; Charge la valeur contenue à l'adresse 1Ah dans W`

## II-6 Ports d'entrées/Sorties

Le PIC 16F84 est doté de deux ports d'entrées/Sorties appelés PortA et PortB.

### II-6-1 Port A

Il comporte 5 pattes d'entrée/sortie bidirectionnelles, notées RAX avec  $x=\{0,1,2,3,4\}$  sur le brochage du pic (Fig.II.2). Le registre PORTA, d'adresse 05h dans la banque 0, permet d'y accéder en lecture ou en écriture. Le registre TRISA, d'adresse 85h dans la banque 1, permet de choisir le sens de chaque patte (entrée ou sortie) : un bit à 1 positionne le port en entrée, un bit à 0 positionne le port en sortie.

La Figure.II.10 donne le câblage interne d'une patte du port A :

- "Data Latch" : Mémorisation de la valeur écrite quand le port est en sortie.
- "TRIS Latch" : Mémorisation du sens (entrée ou sortie) de la patte.
- "TTL input buffer" : Buffer de lecture de la valeur du port. La lecture est toujours réalisée sur la patte, pas à la sortie de la bascule d'écriture.
- Transistor N : En écriture : Saturé ou bloqué suivant la valeur écrite.

En lecture : Bloqué.

- Transistor P : Permet d'alimenter la sortie.

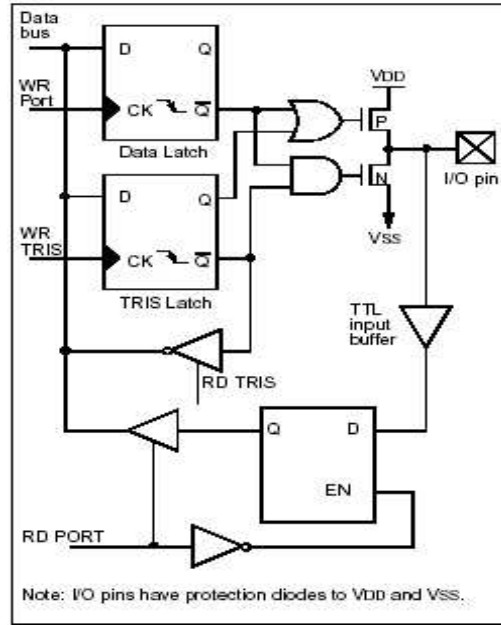


Fig.II.10: Câblage interne d'une patte du port A.

La patte RA4 peut aussi servir d'entrée de comptage pour le timer0.

## II-6-2 Port B

Il comporte 8 pattes d'entrée/sortie bidirectionnelles, notées RBx avec  $x=\{0,1,2,3,4,5,6,7\}$  sur le brochage du pic (Figure.II.2). Le registre PORTB, d'adresse 06h dans la banque 0, permet d'y accéder en lecture ou en écriture.

Le registre TRISB, d'adresse 86h dans la banque 1, permet de choisir le sens de chaque patte (entrée ou sortie) : un bit à 1 positionne le port en entrée, un bit à 0 positionne le port en sortie.

Le câblage interne d'une porte du port B ressemble beaucoup à celui du port A (Fig.II.11). On peut noter la fonction particulière pilotée par le bit RBPU (OPTION\_REG.7) qui permet d'alimenter (RBPU=0) ou non (RBPU=1) les sorties.

Les quatre bits de poids fort (RB7-RB4) peuvent être utilisés pour déclencher une interruption sur changement d'état.

RB0 peut aussi servir d'entrée d'interruption externe.

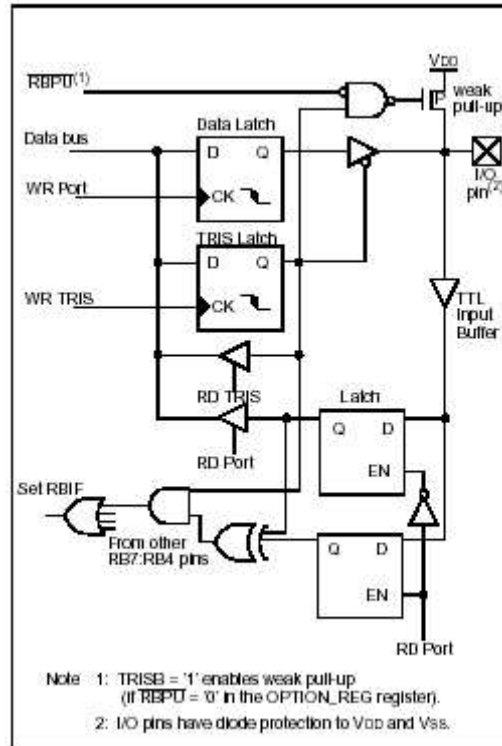


Fig.II.11: Câblage interne d'une patte du port B.

## II-7 Compteur

Le PIC 16F84 est doté d'un compteur 8 bits. La Figure.II.12 en donne l'organigramme.

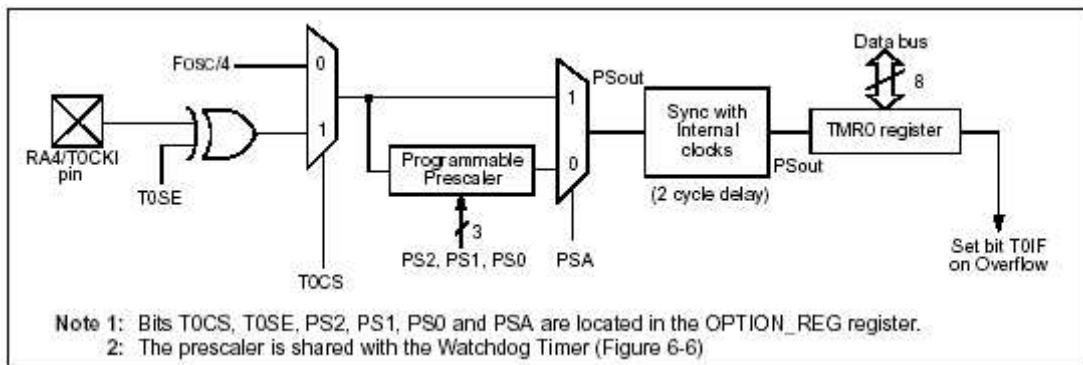


Fig.II.12: Organigramme du Timer0.

### II-7-1 Registre TMR0

C'est le registre de 8 bits qui donne la valeur du comptage réalisé. Il est accessible en lecture et en écriture à l'adresse 01h dans la banque 0.

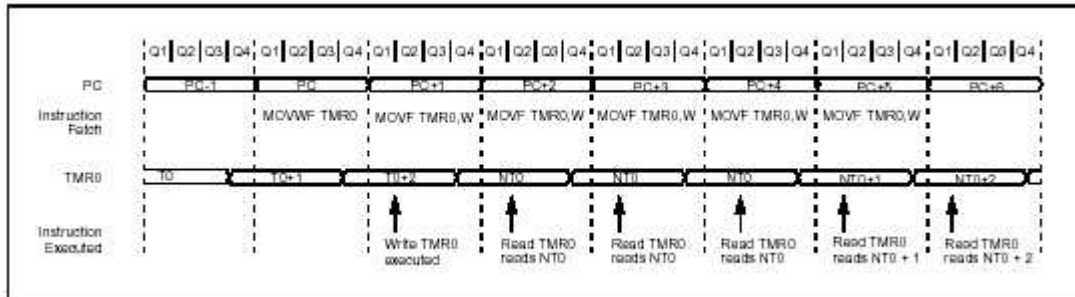


Fig.II.13: Prise en compte de l'écriture dans le registre TMR0.

Lors d'une écriture dans TMR0, le comptage est inhibé pendant deux cycles machine (Fig.II.13). Si l'on veut déterminer un temps avec précision, il faut tenir compte de ce retard au démarrage.

### II-7-2 Choix de l'horloge

Le timer0 peut fonctionner suivant deux modes en fonction du bit T0CS (OPTION\_REG.5). En mode timer (T0CS=0), le registre TMR0 est incrémenté à chaque cycle machine (si le pré-diviseur n'est pas sélectionné).

En mode compteur (T0CS=1), le registre TMR0 est incrémenté sur chaque front montant ou chaque front descendant du signal reçu sur la broche RA4/T0CK1 en fonction du bit T0SE (OPTION\_REG.4). Si T0SE=0, les fronts montants sont comptés, T0SE=1, les fronts descendants sont comptés.

### II-7-3 Pré-diviseur

En plus des deux horloges, un pré-diviseur, partagé avec le chien de garde, est disponible. La période de l'horloge d'entrée est divisée par une valeur comprise entre 2 et 256 suivant les bits PS2, PS1 et PS0 (respectivement OPTION\_REG.2, .1 et .0) (Figure VI.3). Le bit PSA (OPTION\_REG.3) permet de choisir entre la pré-division de timer0 (PSA=0) ou du chien de garde (PSA=1).

PSA	PS2	PS1	PS0	/tmr0	/WD
0	0	0	0	2	1
0	0	0	1	4	1
0	0	1	0	8	1
0	0	1	1	16	1
0	1	0	0	32	1
0	1	0	1	64	1
0	1	1	0	128	1
0	1	1	1	256	1
1	0	0	0	1	1
1	0	0	1	1	2
1	0	1	0	1	4
1	0	1	1	1	8
1	1	0	0	1	16
1	1	0	1	1	32
1	1	1	0	1	64
1	1	1	1	1	128

Fig.II.14: Valeurs du pré-diviseur en fonction de PSA, PS2, PS2 et PS0.

#### II-7-4 Fin de comptage et interruption

Le bit TOIF (INTCON.2) est mis à 1 chaque fois que le registre TMR0 passe de FFh à 00h. On peut donc tester ce bit pour connaître la fin de comptage. Pour compter 50 événements, il faut donc charger TMR0 avec la valeur  $256-50=206$  et attendre le passage de TOIF à 1. Cette méthode est simple mais bloque le processeur dans une boucle d'attente.

On peut aussi repérer la fin du comptage grâce à l'interruption que peut générer TOIF en passant à 1. Le processeur est ainsi libre de travailler en attendant cet événement.

#### II-7-5 Registres utiles à la gestion de timer0

Plusieurs registres ont été évoqués dans ce paragraphe. Ils sont synthétisés dans la Figure.II.15.

Address	Name	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	Value on Power-on Reset	Value on all other resets
0th	TMR0	Timer0 module's register								xxxx xxxx	uuuu uuuu
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 0000
8th	OPTION_REG	PSPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
85h	TRISA	-	-	-	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	---1 1111	---1 1111

Legend: x = unknown, u = unchanged, - = unimplemented read as '0'. Shaded cells are not associated with Timer0.

Fig.II.15: Registres utiles à la gestion de timer0.

## **II-8 Accès à la mémoire EEPROM**

Le PIC possède une zone EEPROM de 64 octets accessibles en lecture et en écriture par le programme. On peut y sauvegarder des valeurs, qui seront conservées même si l'alimentation est éteinte, et les récupérer lors de la mise sous tension. Leur accès est spécifique et requiert l'utilisation de registres dédiés. La lecture et l'écriture ne peut s'exécuter que selon des séquences particulières.

### **II-8-1 Registres utilisés**

Quatre registres sont utilisés pour l'accès à la mémoire EEPROM du PIC :

- EEDATA contient la donnée.
- EEADR contient l'adresse.
- EECON1 (Fig.II.16) est le registre de contrôle de l'accès à l'eeprom. Cinq bits permettent un cet accès :
- RD et WR initient la lecture ou l'écriture. Ils sont mis à 1 par le programme pour initier l'accès et mis à zéro par le système à la fin de l'accès.
- WREN autorise (1) ou non (0) l'accès en écriture.
- WRERR est mis à 1 par le système quand une opération d'écriture est interrompue par MCLR, reset ou le chien de garde.
- EEIF est un drapeau d'interruption signalant la fin de l'écriture physique dans la mémoire eeprom. Il doit être mis à 0 par programme.

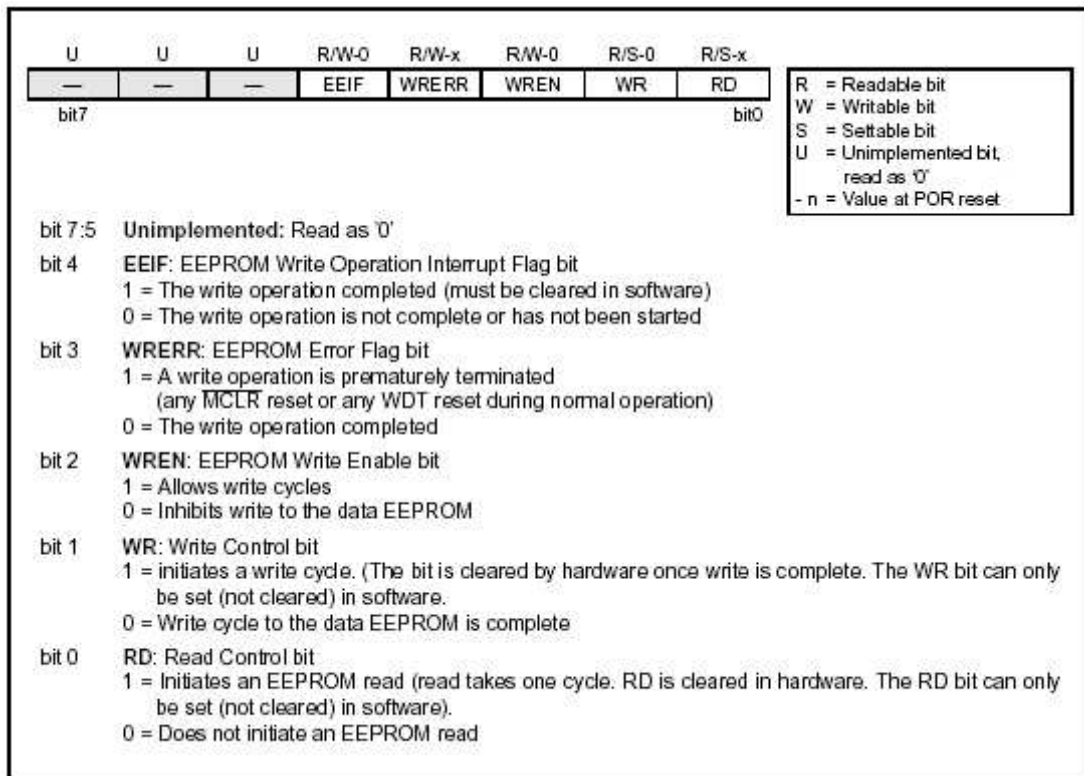


Fig.II.16: Registre EECON1.

- EECON2 joue un rôle spécifique lors de l'écriture.

### II-8-2 Lecture

Pour lire une donnée dans la mémoire EEPROM, il faut mettre l'adresse dans EEADR et positionner RD à 1. La valeur lue est alors disponible dans EEDATA au cycle machine suivant.

### II-8-3 Ecriture

Pour écrire une donnée dans la mémoire EEPROM, il faut d'abord mettre l'adresse dans EEADR et la donnée dans EEDATA. Un cycle bien spécifique doit ensuite être respecté pour que l'écriture ait lieu.

## II-9 Interruptions

### II-9-1 Mécanisme

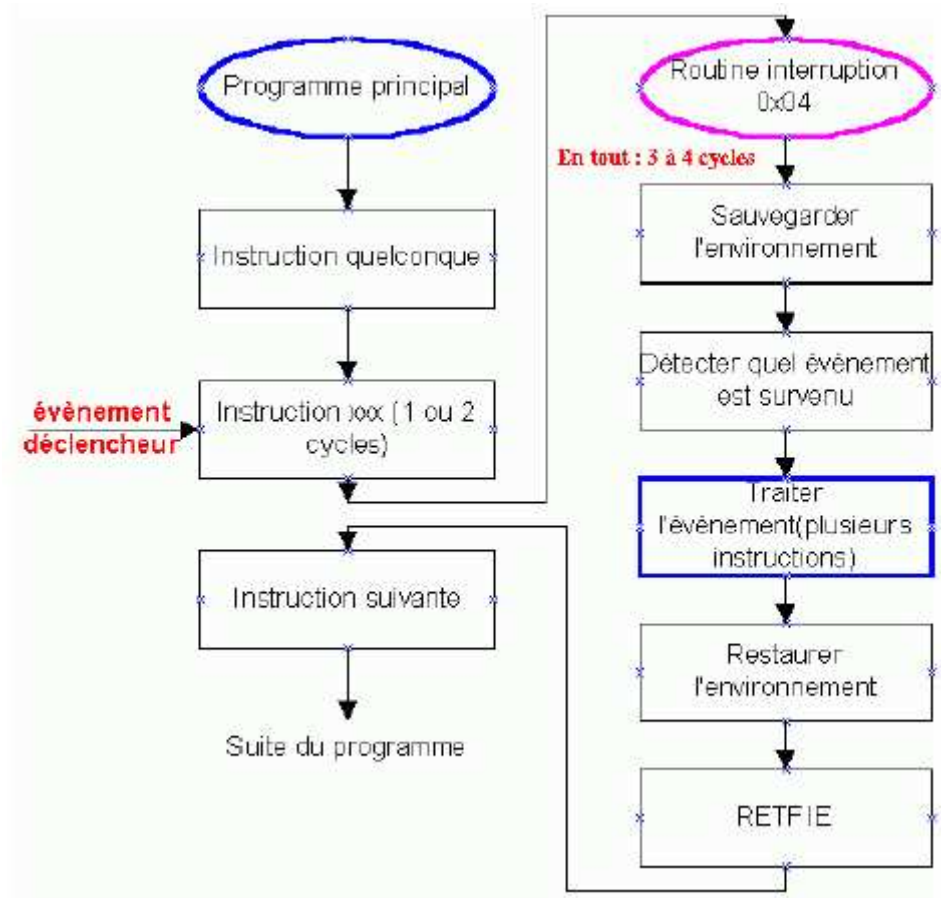


Fig. II.17: Déroulement d'un programme lors d'une interruption.

L'interruption est un mécanisme fondamental de tout processeur. Il permet de prendre en compte des événements extérieurs au processeur et de leur associer un traitement spécifique. La Figure.II.17 donne le déroulement du programme lors d'une interruption. Il faut noter que l'exécution d'une instruction n'est jamais interrompue ; c'est à la fin de l'instruction en cours lors de l'arrivée de l'événement que le sous-programme d'interruption est exécuté.

La séquence classique de fonctionnement d'une interruption est la suivante :

- 1- Détection de l'événement déclencheur
- 2- Fin de l'instruction en cours
- 3- Sauvegarde de l'adresse de retour

- 4- Déroutement vers la routine d'interruption
- 5- Sauvegarde du contexte
- 6- Identification de l'événement survenu
- 7- Traitement de l'interruption correspondante
- 8- Restauration du contexte
- 9- Retour au programme initial

## II-9-2 Différentes sources d'interruption

Dans le cas du PIC 16F84, il existe 4 sources d'interruption (Fig.II.18) :

- INT : Interruption externe, broche RB0/INT
- TMR0 : Fin de comptage
- PORTB : Changement d'état du port B (RB7-RB4)
- EEPROM : Fin d'écriture en EEPROM

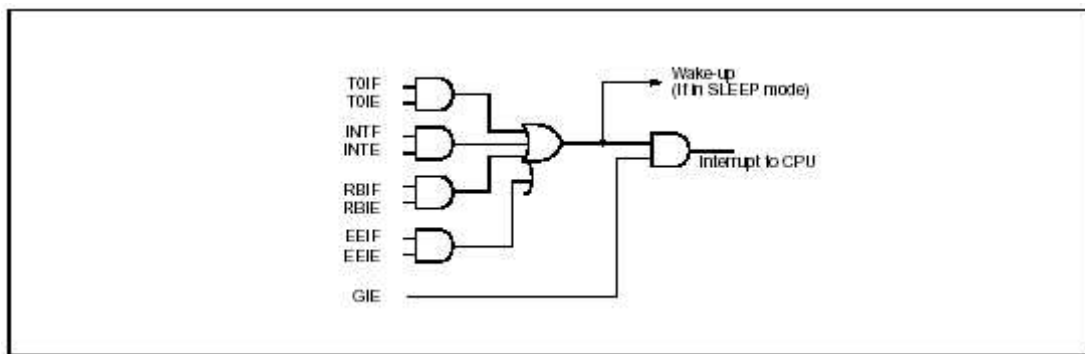


Fig.II.18: Logique des événements associés aux interruptions.

## II-9-3 Validation des interruptions

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
							bit0
bit7							
<p><b>bit 7:</b> <b>GIE:</b> Global Interrupt Enable bit            1 = Enables all un-masked interrupts            0 = Disables all interrupts            Note: For the operation of the interrupt structure, please refer to Section 8.5.</p> <p><b>bit 6:</b> <b>EEIE:</b> EE Write Complete Interrupt Enable bit            1 = Enables the EE write complete interrupt            0 = Disables the EE write complete interrupt</p> <p><b>bit 5:</b> <b>TOIE:</b> TMR0 Overflow Interrupt Enable bit            1 = Enables the TMR0 interrupt            0 = Disables the TMR0 interrupt</p> <p><b>bit 4:</b> <b>INTE:</b> RB0/INT Interrupt Enable bit            1 = Enables the RB0/INT interrupt            0 = Disables the RB0/INT interrupt</p> <p><b>bit 3:</b> <b>RBIE:</b> RB Port Change Interrupt Enable bit            1 = Enables the RB port change interrupt            0 = Disables the RB port change interrupt</p> <p><b>bit 2:</b> <b>TOIF:</b> TMR0 overflow interrupt flag bit            1 = TMR0 has overflowed (must be cleared in software)            0 = TMR0 did not overflow</p> <p><b>bit 1:</b> <b>INTF:</b> RB0/INT Interrupt Flag bit            1 = The RB0/INT interrupt occurred            0 = The RB0/INT interrupt did not occur</p> <p><b>bit 0:</b> <b>RBIF:</b> RB Port Change Interrupt Flag bit            1 = When at least one of the RB7:RB4 pins changed state (must be cleared in software)            0 = None of the RB7:RB4 pins have changed state</p>							

R = Readable bit  
 W = Writable bit  
 U = Unimplemented bit, read as '0'  
 -n = Value at POR reset

Fig.II.19: Registre de contrôle d'interruption du PIC – INTCON

Chacune de ses sources peut être validée indépendamment grâce aux bits 3 à 6 du registre INTCON (Fig.II.19). Le bit GIE de ce même registre permet une validation générale des interruptions. Ainsi, pour que le déroutement du programme en cours soit déclenché, il faut qu'un des événements extérieurs soit détecté, que l'interruption correspondante soit validée et que la validation générale soit activée.

## **II-10 Chien de garde**

### **II-10-1 Principe**

C'est un système de protection contre un blocage du programme. Par exemple, si le programme attend le résultat d'un système extérieur (conversion analogique numérique par exemple) et qu'il n'y a pas de réponse, il peut rester bloquer. Pour en sortir on utilise un chien de garde. Il s'agit d'un compteur qui, lorsqu'il arrive en fin de comptage, permet de redémarrer le programme. Il est lancé au début du programme. En fonctionnement normal, il est remis à zéro régulièrement dans une branche du programme qui s'exécute régulièrement. Si le programme est bloqué, il ne passe plus dans la branche de remise à zéro et le comptage va jusqu'au bout, déclenche le chien de garde qui relance le programme.

### **II-10-2 Mise en service**

Elle se décide lors de la programmation physique du PIC. Elle ne peut pas être suspendue pendant l'exécution d'un programme. Elle est définitive jusqu'à une nouvelle programmation de la puce.

La directive de programmation `_CONFIG` permet de valider (option `_WDT_ON`) ou non (option `_WDT_OFF`). La mise en service peut aussi être réalisée directement par le programmeur. L'inconvénient de cette seconde solution est que le code du programme ne contient pas l'information ; la mise en service du chien de garde peut être oubliée lors du téléchargement et générer un fonctionnement incorrect du programme en cas de blocage.

### **II-10-3 Gestion**

Une fois le chien de garde mis en service, il faut remettre le comptage à zéro régulièrement. Cette opération est réalisée par l'instruction `clrwtd`. Tant que le programme se déroule normalement, cette instruction est exécutée régulièrement et le chien de garde ne s'active pas. Si un blocage apparaît, la remise à zéro n'a pas lieu et le chien de garde est activé. Le PIC redémarre alors à l'adresse 0000h et le bit TO (STATUS.4) est mis 0. Le test de ce bit au début du programme permet de

savoir si le système vient d'être mis sous tension (TO=1) ou si le chien de garde vient de s'activer (TO=0).

#### **II-10-4 Choix de la durée**

Le chien de garde possède sa propre horloge. Sa période de base est de 18ms. Le pré-diviseur de fréquence utilisé par le compteur est partagé avec le chien de garde. Si le bit PSA (OPTION\_REG.3) est à 1, le pré-diviseur est assigné au chien de garde. 8 valeurs de 1 à 128 sont disponibles, ce qui permet d'aller jusqu'à  $128 * 18\text{ms} = 2.3\text{s}$  avant le déclenchement du chien de garde.

#### **II-11 Conclusion**

Dans ce chapitre, les notions de base du microcontrôleur (16F84) ont été données, où nous avons essayé de toucher tous les points concernant ce microcontrôleur et surtout sa structure interne.

# Moteur à courant continu

## III-1 Introduction

La machine à courant continu est un convertisseur d'énergie qui fonctionne dans les deux (2) sens :

Conversion de l'énergie électrique en énergie mécanique de rotation et réciproquement.

A partir d'une source électrique (forme électrique) MCC entraîne un axe en rotation (forme mécanique), et inversement, si on entraîne son axe en rotation (forme électrique), il fournira une tension (forme électrique) qui fera passer un courant dans un dispositif électrique.

Dans le 1<sup>er</sup> cas, MCC fonction en moteur, dans le 2<sup>ème</sup> cas MCC fonction en génératrice.

Actuellement, son utilisation la plus fréquente est en moteur, son utilisation en génératrice, se limite à délivrer une tension proportionnelle à la vitesse (dynamo tachymétrie).

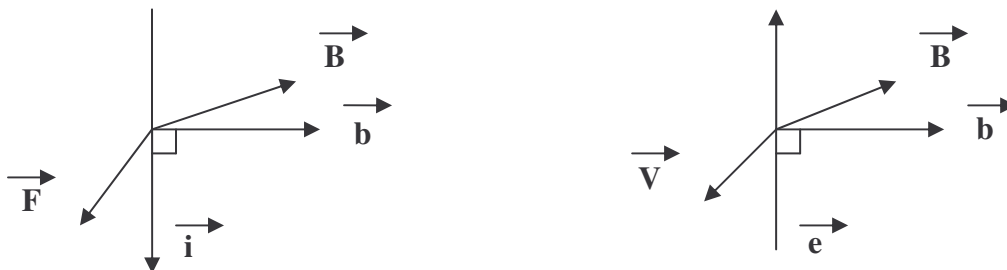
Les moteurs à courant continu sont utilisés dans les asservissements ou la facilité de réglage de la vitesse, du moment, du couple et du sens de rotation permet une mise en œuvre aisée.

Aujourd'hui les machines à courant continu sont très utilisées surtout pour la motorisation de véhicules ferroviaires, automobiles et aussi très largement dans le domaine des processus industriels. Dans les puissances les plus réduites (cas des moteurs à courant continu à aimant permanent), elles sont très répandues dans les automatismes et en robotique.

### III-2 Généralités

#### III-2-1 Lois fondamentales

Considérons un élément de conducteur rectiligne de longueur  $l$ , ici vertical, placé dans un champ magnétique  $B$  supposé ici uniforme.  $B$  est la projection de  $B$  dans un plan normal à l'élément de conducteur.



Ce conducteur peut être indépendamment le siège de deux types de phénomène :

<u>Laplace</u>	<u>Faraday</u>
$\vec{F} = i \cdot l \wedge \vec{B}$	$\vec{e} = v \cdot l \wedge \vec{B}$
$F = i \cdot l \cdot b$	$e = v \cdot l \cdot b = \frac{d\phi}{dt}$
$\nabla v$	$\nabla i$

Fig.III.1 : Représentation vectorielle des lois fondamentales.

### III-2-1-1 Loi de Laplace

Si le conducteur est parcouru par un courant  $i$ , le champ  $B$  exerce sur lui une force  $F$  d'intensité :

$$\vec{F} = i \cdot l \cdot \vec{B} \quad (\text{III.1})$$
$$F = i \cdot l \cdot b$$

$F$ , conséquence de l'existence de  $I$  et de  $B$ , est orientée de telle manière que  $i$ ,  $B$ ,  $F$  forment un trièdre direct.

Le fait que la conducteur, parcouru par un courant  $i$ , soit immobile ou non n'intervient pas.

### III-2-1-2 Loi de faraday

Un conducteur mobile de longueur  $l$ , animé d'une vitesse  $v$ , placé dans le champ  $B$  de projection  $b$  normale à  $l$ , est le siège d'une f.é.m induit  $e$ , qu'il soit parcouru par un courant ou non :

$$e = -\frac{d\Phi}{dt} \quad (\text{III.2})$$

$\Phi$  : étant le flux couple par le conducteur

On peut aussi écrire :

$$\vec{e} = \vec{v} \cdot l \wedge \vec{b} \quad (\text{III.3})$$
$$e = v \cdot l \cdot b$$

$e$  : est orientée de telle manière que  $v$ ,  $b$ ,  $e$  forment un trièdre direct. Cette orientation se déduit de celle du courant induit que  $e$  tend à circuler (si le circuit est fermé).

Les énergies échange sont égales et de signe contraire :

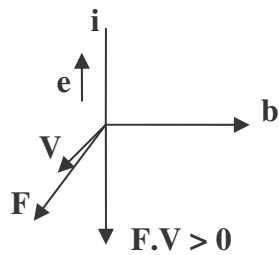
$dx$  étant le déplacement de l'élément de conducteur dans la direction de la force  $F$ .

$$dW_{\text{méca}} = F \cdot dx = F \cdot v \cdot dt = +(i \cdot l \cdot b) \cdot (v \cdot dt)$$

$$dW_{\text{élec}} = i \cdot d\Phi = -i \cdot b \cdot dS = -i \cdot b \cdot l \cdot dx = -i \cdot (b \cdot l \cdot v \cdot dt)$$

Le fonctionnement en moteur et génératrice est résumé par le diagramme suivant:

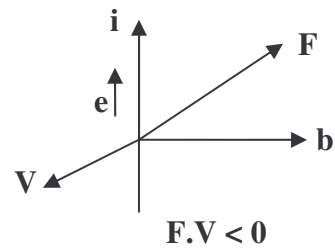
F et v de même sens



$$e.i < 0 \quad |F.v| = |e.i|$$

réceptrice

f et V de sens contraire



$$e.i > 0 \quad |F.v| = |e.i|$$

génératrice

fig.III.2: le fonctionnement en réceptrice et génératrice

### III-3 Présentation générale d'un moteur courant continu

#### III-3-1 Constitution

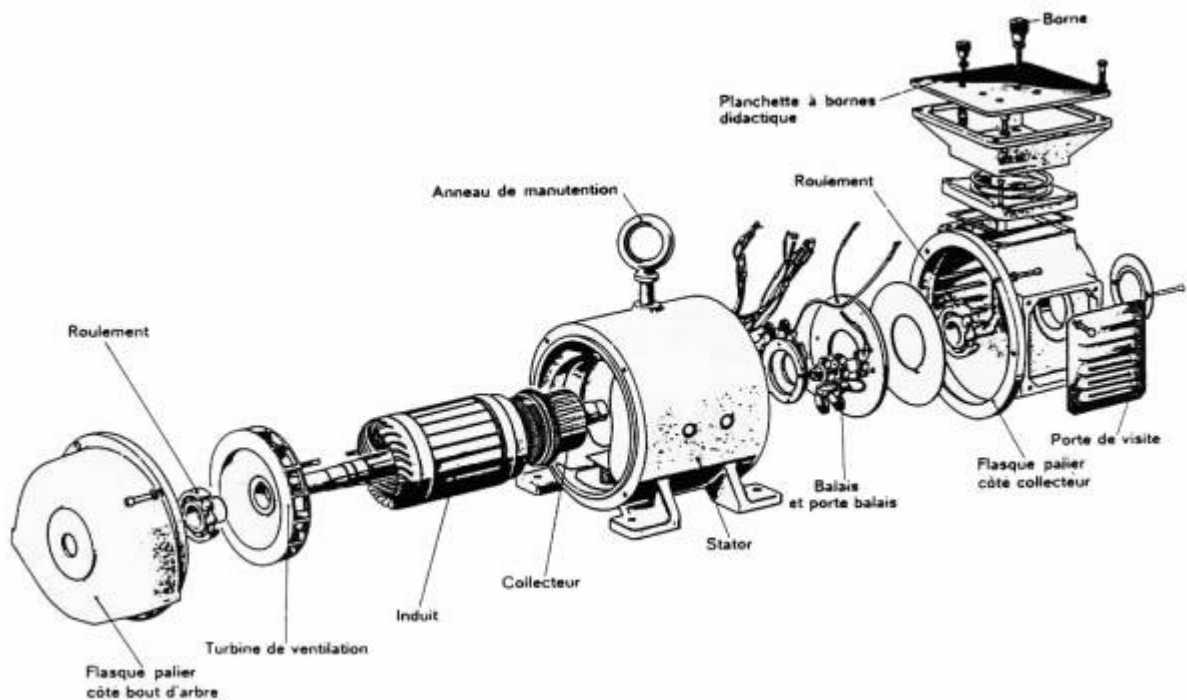


Fig.III.3: Constitution Mcc

### III-3-2 Constituants électriques

Dans un moteur à courant continu on peut distinguer les éléments principaux suivants :

#### III-3-2-1 Le stator

C'est la partie extérieure du moteur qui est fixe (statique). Il crée le champ magnétique dans lequel va se mouvoir le rotor. Pour créer ce champ, deux solutions sont possibles :

- **à aimant permanent** : cette solution est utilisée pour la réalisation de moteur de faible puissance (quelques watts), bon marché et d'encombrement réduite.
- **à inducteur bobiné** : il faut alors l'alimenter en énergie électrique (à partir de différences de potentiels de seulement 1,5 volt) pour créer le champ magnétique. Cette solution permet la réalisation des moteurs plus puissants que ceux à aimant permanent, mais également d'un coût nettement plus élevé.

#### III-3-2-2 Le rotor

C'est la partie intérieure du moteur qui tourne.

Dans un moteur à courant continu, le rotor est constitué par Un bobinage appelé induit. Le moteur à courant continu est alimenté par ce bobinage. Celui-ci étant mobile, il est muni des lames de cuivre : les collecteurs, ceux-ci, lors de la rotation, frottent sur des balais en graphique fixés sur le stator. C'est par leur intermédiaire que l'induit peut être alimenté en énergie électrique.

L'inconvénient de ce mode d'alimentation est l'assurance des balais dus aux frottements. (fig.III.4)

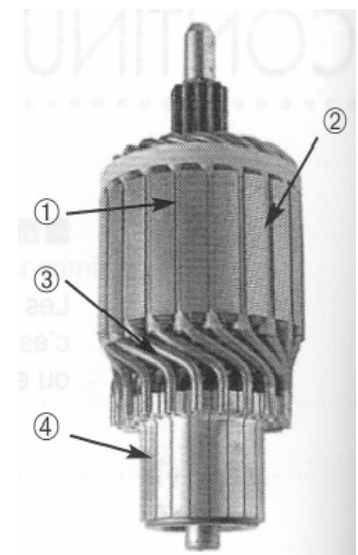


fig.III.4 Le rotor

### III-3-2-3 Collecteur

Il se compose de lames de cuivre isolées latéralement les unes des autres, disposées de sorte à former un cylindre et reliées aux conducteurs de l'induit en des points régulièrement espacés.

### III-3-2-4 Balais

Les balais permettent l'alimentation de l'induit (partie en rotation) grâce à un contact glissant entre les lames du collecteur reliées aux conducteurs de l'induit et le circuit électrique extérieur.

Ils sont constitués de petits cubes ayant une surface de contact de quelques mm<sup>2</sup> à quelques cm<sup>2</sup>, en graphite pur ou en alliage, qui doivent résister à des conditions d'utilisation sévères (courants élevés, températures élevées, frottements, arc, atmosphères chargées ou très sèches, ...). (fig.III.5)

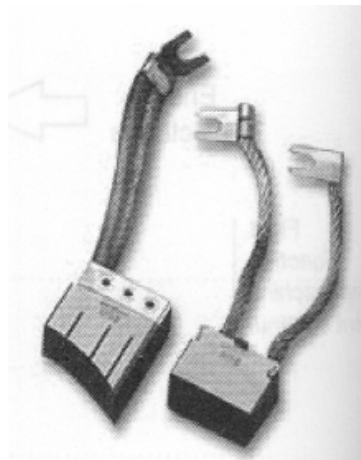


Fig.III.5 : Les balais

### Remarque

Généralement des pôles auxiliaires sont également présents dans les machines. Ils servent à réduire les étincelles au niveau des balais lors d'un changement de lame de collecteur. Ils sont constitués de bobinages placés perpendiculairement aux lignes de champ de l'inducteur et sont alimentés en série avec l'induit.

### III-3-3 Principe de fonctionnement

#### III-3-3-1 Couple électromagnétique

Lorsqu'on alimente l'inducteur par une source de tension continu, celui-ci produit un champ magnétique dont les lignes de champ sont canalisées par le circuit magnétique. On alimente ensuite l'induit par une autre source de tension continu. Chacun des conducteurs de l'induit est alors parcouru par un courant et placé dans le champ magnétique inducteur, est soumis à une force de Laplace telle que :  $F = B.I.l$ .

L'intensité de cette force est proportionnelle à l'intensité  $I$  du courant induit et à l'intensité  $B$  du champ magnétique créée par l'inducteur.

Deux conducteurs sont reliés pour former une spire. Il apparaît donc deux forces de même intensité  $F$  mais de sens opposés qui forment un couple électromagnétique d'intensité  $T = 2.r.F$ , ou  $r$  est le rayon du rotor.

On a donc :  $T = 2.r.F = 2.r.B.I.l = S.B.I = \Phi.I$

On montre que le moment du couple électromagnétique total  $T_{em}$  exercé sur l'induit vaut :

$$T = K.\Phi.I$$

- $T_{em}$  en newton mètre (N.m).
- $K$  : constante du moteur qui dépend que de sa constitution (nombre total de conducteurs  $N$ ... on montre que  $K = (p/2\pi a)$ .  $N$  avec  $p$  nombre paires de pôles et à nombre de paires de voies d'enroulement).
- $\Phi$  : flux créée par un pôle inducteur, en webers (Wb).
- $I$  : intensité du courant dans chaque conducteur de l'induit, en ampère (A).

#### III-3-3-2 Force électromotrice

Le flux magnétique embrassé par chaque spire de l'induit varie lorsque celle-ci est entraînée en rotation. La loi de Faraday,  $e = -d\Phi/dt$ , implique qu'une f.e.m.

Induit apparaît aux bornes de cette spire. Il en est de même pour chaque spire de l'induit.

On montre que la f.e.m. Induit total  $E$  qui apparaît aux bornes de l'induit vaut.

$$E = K \cdot \Phi \cdot \Omega$$

- $E$  en volts (v).
- $K$  est la même constante que précédemment.
- $\Phi$  : flux crée par un pole inducteur, en webers (Wb).
- $\Omega$  : vitesse de rotation du moteur en radians par seconde (rad/s).

### III-4 Puissance mises en jeu

#### III-4-1 Puissance absorbée

Le moteur absorbe la puissance électrique :  $P_A = P_{\text{induit}} + P_{\text{inducteur}} = U \cdot I + U_E \cdot I_E$

#### III-4-2 Puissance électromagnétique

De l'équation de l'induit :  $U = E + R \cdot I$ , on déduit l'équation des puissances de l'induit :  $U \cdot I = E \cdot I + R \cdot I^2$

- $P_{AI} = U \cdot I$  est la puissance absorbée par l'induit
- $P_{JI} = R \cdot I^2$  est la puissance dissipée par effet joule dans la résistance de l'induit
- $P_{EM} = E \cdot I$  est la puissance électromagnétique que l'induit transmet au rotor par l'action du couple électromagnétique  $T_{EM}$

Des équations fondamentales du moteur :  $E = K \cdot \Phi \cdot \Omega$  et  $T_{EM} = k \cdot \Phi \cdot I$ , on déduit que :  $P_{EM} = E \cdot I = T_{EM} \cdot \Phi$

Il y a bien conservation de l'énergie.

#### III-4-3 Puissance perdue

Lors de la transformation d'énergie, une partie de la puissance absorbée est perdue sous forme de chaleur :

- perte joule :  $P_J = P_{JI} + P_{JE} = R \cdot I^2 + R_E \cdot I_E^2 = R \cdot I^2 + U_E \cdot I_E$
- perte mécaniques (frottements, ventilation) :  $P_{MECA}$

- perte fer (courant induit dans le circuit magnétique et pertes dues au phénomène d'hystérésis) :  $P_{FER}$

On appelle pertes collectives  $P_c$ , la somme des pertes mécaniques et des pertes fer :

$$P_c = P_{MECA} + P_{FER}$$

Lorsque le moteur fonctionne à flux constant, les pertes collectives sont proportionnelles à la vitesse de rotation :  $P_c = a \cdot \Omega$  avec  $a$  constante.

On définit le moment du couple de pertes  $T_p$  par la relation :  $T_p = \frac{P_c}{\Omega}$

Conséquence, à flux constant, le moment du couple de perte  $T_p$  est une constante du moteur.

### III-4-4 Puissance utile

C'est la puissance mécanique  $P_u$  fournie à l'utilisateur. Elle correspond au

moment du couple utile tel que :  $T_u = \frac{P_u}{\Omega}$

### III-4-5 Bilan des puissances

Du graphe de la figure III.6, on déduit :

$$P_{EM} = P_c + P_u$$

$$D'où : T_{EM} = T_p + T_u$$

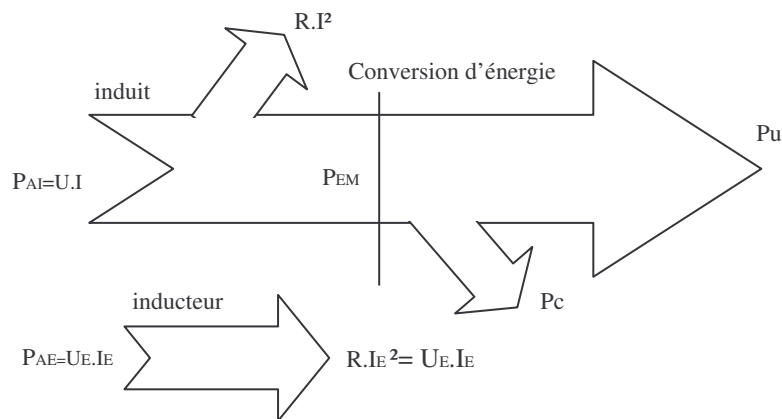


Fig.III.6 : Bilan des puissances

### III-4-6 Rendement

Le rendement d'un moteur à courant continu varie de 75 à 95%. Il est d'autant meilleur que le moteur est de forte puissance.

- Méthode directe :  $\eta = \frac{P_u}{P_A}$
- Méthode des pertes séparées :  $\eta = \frac{P_A - P_{JI} - P_{JE} - P_c}{P_A}$

Dans le cas du moteur à aimant permanent, l'inducteur est réalisé avec des aimants permanents : il n'y a donc pas d'alimentation pour l'inducteur, le flux est constant, et les résultats précédents se simplifient (fig.III.7).

$$\eta = \frac{P_u}{P_A}$$

$$\eta = \frac{P_u}{P_u + P_{JI} + P_c}$$

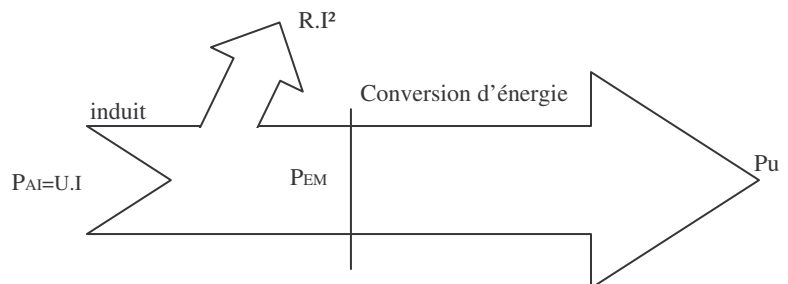


Fig.III.7: Bilan de puissance cas aimant permanent

### III-4-7 Causes des pertes et solutions

Pertes	Pertes fer $P_{fer}$	Pertes joules $P_j$	Pertes mécaniques $P_{méca}$
Causes	Elles sont dues à l'hystérésis (champ rémanent) et au courant de Foucault (courant d'induit dans le fer) et dépendent de B et de $\Omega$ .	Pertes dans l'induit et l'inducteur dues aux résistances des bobinages.	Elles sont dues aux frottements des diverses pièces en mouvement.
Solutions	Utilisation de matériaux à cycles étroits, comme le fer au silicium et le feuilletage de l'induit.	Il faut surtout éviter l'échauffement par ventilation.	Utilisation de roulement et de lubrifiant.

Tableau-1 causes des pertes et solutions

Le moteur que nous allons utiliser dans notre travail est un moteur à aimant permanent, puisque ce type de moteur est utilisé pour des applications de faible puissance (quelque watts).

### III-5 Différents types d'excitation du moteur à courant continu

Il existe plusieurs méthodes pour alimenter l'inducteur du moteur, chacune d'entre elles conduisant à des caractéristiques de fonctionnement différentes.

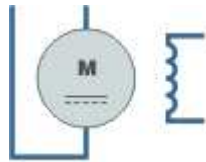


Fig.III.8 : Moteur à excitation indépendante

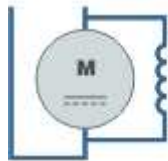


Fig.III.9 : Moteur à excitation shunt



Fig.III.10 : Moteur à excitation sérié

### III-6 Moteur à courant continue de commande

Le moteur à courant continu, avec excitation séparée est le moteur de base des asservissement de position, ainsi bien pour les petites que pour les grandes puissance.

Les moteurs à courant continu sont plus efficaces que les moteurs à courant alternatif à deux phases.

Les moteurs à courant continu sont excités par deux champs indépendants. Pour cela, on distingue deux types de moteurs à continu :

- Les moteurs commandés par la tension d'induit à flux constant.
- Les moteurs commandés par le flux à courant d'induit constant.

Comme exemple, les moteurs à courant continu utilisés dans l'instrumentation utilisent un aimant permanent comme source d'un champs magnétique (excitation), et le signal de commande est alors appliqué à l'induit (commandes par la tension d'induit).

Pour notre application, nous allons considérer le moteur à courant continu commandé par l'induit représenté dans la figure.III.11.

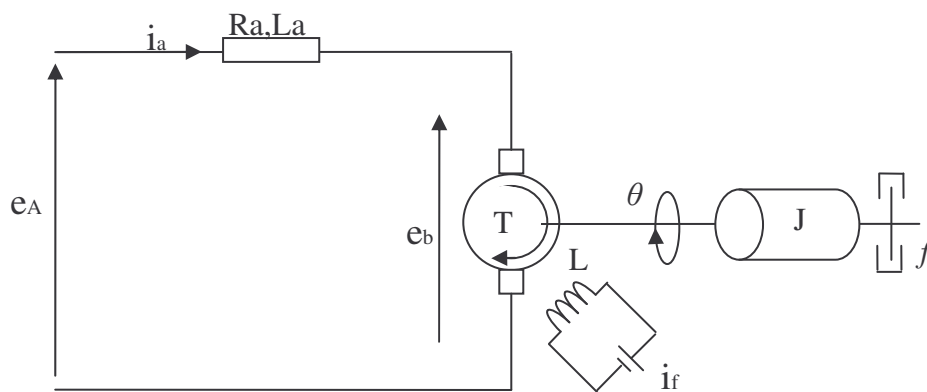


Fig.III.11: Représentation du moteur à courant continu

$R_a$  : Résistance d'induit, ohms ;

$L_a$  : Inductance d'induit, henry ;

$i_a$  : Courant d'induit, ampères ;

$i_f$  : Courant de champ d'excitation ;

$e_a$  : Tension de contrôle appliquée aux bornes de l'induit, volts ;

$e_b$  : f.e.m induit, volts ;

$\theta$  : Déplacement angulaire de l'axe du moteur, radians ;

T : couple délivré par le moteur, newton.mètre ;

J : moment d'inertie équivalent entre le moteur et la charge relative à l'axe du moteur, Ib.ft.sec<sup>2</sup> ;

f : coefficient de frottement équivalent entre le moteur et la charge à l'axe du moteur, N.m/rad/sec.

Le couple T délivré par le moteur est proportionnel au produit du courant d'induit  $i_a$  et au flux d'excitation  $\Psi$ .

$$\Psi = K_f . i_f \quad (\text{III.4})$$

Où  $K_f$  est une constante. Le couple T peut donc être écrit sous la forme :

$$T = K_f . i_f . K_1 . i_a \quad (\text{III.5})$$

Où  $K_1$  est une constante.

Dans ce type de moteur, le courant de champ  $i_f$  est maintenu constant, et pour un courant de champ constant, le flux devient constant, et par conséquent le couple devient proportionnel au courant d'induit :

$$T = K . i_a \quad (\text{III.6})$$

Où K dite constant du couple moteur. Quand le rotor tourne, une tension proportionnelle au produit du flux et la vitesse angulaire est induite au niveau de l'induit. Pour un flux constant, la tension induite  $e_b$  est donc proportionnelle à la

vitesse angulaire  $\frac{d\theta}{dt}$  :

$$e_b = K_b \frac{d\theta}{dt} \quad (\text{III.7})$$

Où  $K_b$  est la constante de la f.e.m induite.

La vitesse de ce type de moteurs est contrôlée par la tension appliquée aux bornes de l'induit  $e_a$ .

L'équation différentielle du circuit d'induit est donnée par :

$$L_a \frac{di_a}{dt} + R_a \cdot i_a + e_b = e_a \quad (\text{III.8})$$

Le courant  $i_a$  produit un couple qui sera appliqué à une inertie et un frottement, donc :

$$J \cdot \frac{d^2\theta}{dt^2} + f \cdot \frac{d\theta}{dt} = T = K \cdot i_a \quad (\text{III.9})$$

Supposons que les conditions initiales sont nulles, et prenons la transformée de Laplace des équations 7,8 et 9 :

$$K_b \cdot P \cdot \theta(P) = E_b(P) \quad (\text{III-10})$$

$$(L_a \cdot P + R_a) \cdot I_a(P) + E_b(P) = E_a(P) \quad (\text{III-11})$$

$$(J \cdot P^2 + f \cdot P) \cdot \theta(P) = T(P) = K \cdot I_a(P) \quad (\text{III-12})$$

Prenons  $E_a(P)$  comme entrée et  $\theta(P)$  comme sortie, on peut à partir des équations 10, 11 et 12 construire le diagramme de blocks correspondant à ce moteur comme le montre sur la figure.III.11.

La f.e.m induite ( $e_b$ ) est prise comme un signal de réaction qui est proportionnel à la vitesse du moteur.

La fonction de transfert du système est alors donnée par :

$$\frac{\theta(P)}{E_a(P)} = \frac{K}{(P[L_a \cdot J \cdot P^2 + (L_a \cdot f + R_a \cdot J) + R_a \cdot f + K \cdot K_b])} \quad (\text{III.13})$$

L'inductance  $L_a$  dans le circuit d'induit est généralement faible et peut être négligée. Si on la néglige, l'équation 13 devient :

$$\frac{\theta(P)}{E_a(P)} = \frac{K_m}{P \cdot (t_m \cdot p + 1)} \quad (\text{III.14})$$

Ou :  $K_m = \frac{K}{R_a \cdot f + K \cdot K_b}$  gain de moteur

$T_m = \frac{R_a \cdot J}{R_a \cdot f + K \cdot K_b}$  constant de temps du moteur

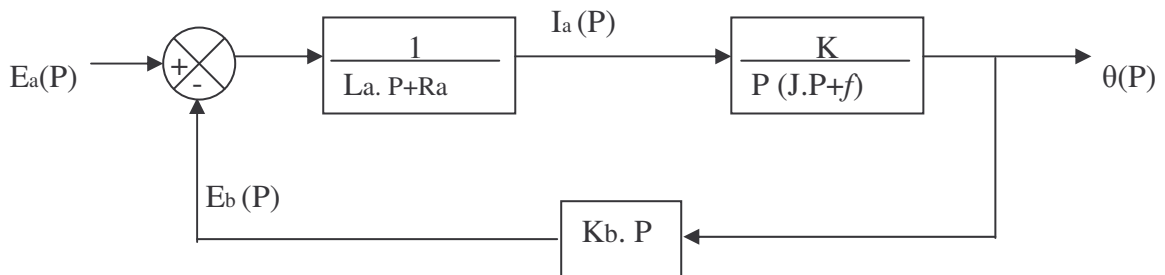


Fig.III.12: Schéma bloc du moteur à courant continu

### III-7 Hacheur

Les hacheurs sont des convertisseurs continu- continu ont pour fonction de fournir une tension continu variable à partir d'une tension fixe. La tension continue de départ être un réseau alternatif redressé et filtré, une batterie d'accumulateurs, une alimentation stabilisée...



Fig.III.13

#### III-7-1 Utilisation de l'hacheur

- commande des moteurs courant continu
- alimentation à découpage

### III-7-2 Hacheur à quatre quadrants (en pont H)

L'hacheur en pont est représenté par la fig.III.14

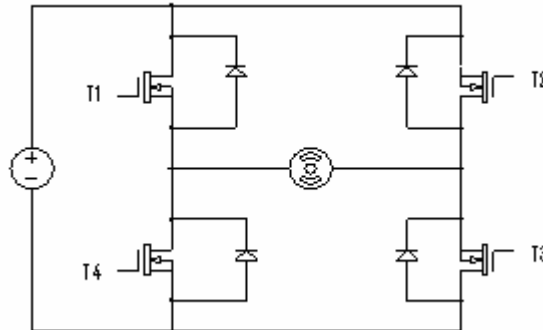


Fig.III.14: Schéma de principe du hacheur à quatre quadrants

### III-7-3 La commande

On commande simultanément la fermeture de T1 et T3 pendant  $\alpha T$ , T2 et T4 sont commandés simultanément pendant le reste de la période.

$$\left\{ \begin{array}{l} T1=1 \ \& \ T3=1 \quad \text{sur } [0, \alpha T] \\ T2=1 \ \& \ T4=1 \quad \text{sur } [\alpha T, T] \end{array} \right.$$

#### La tension aux bornes de la charge

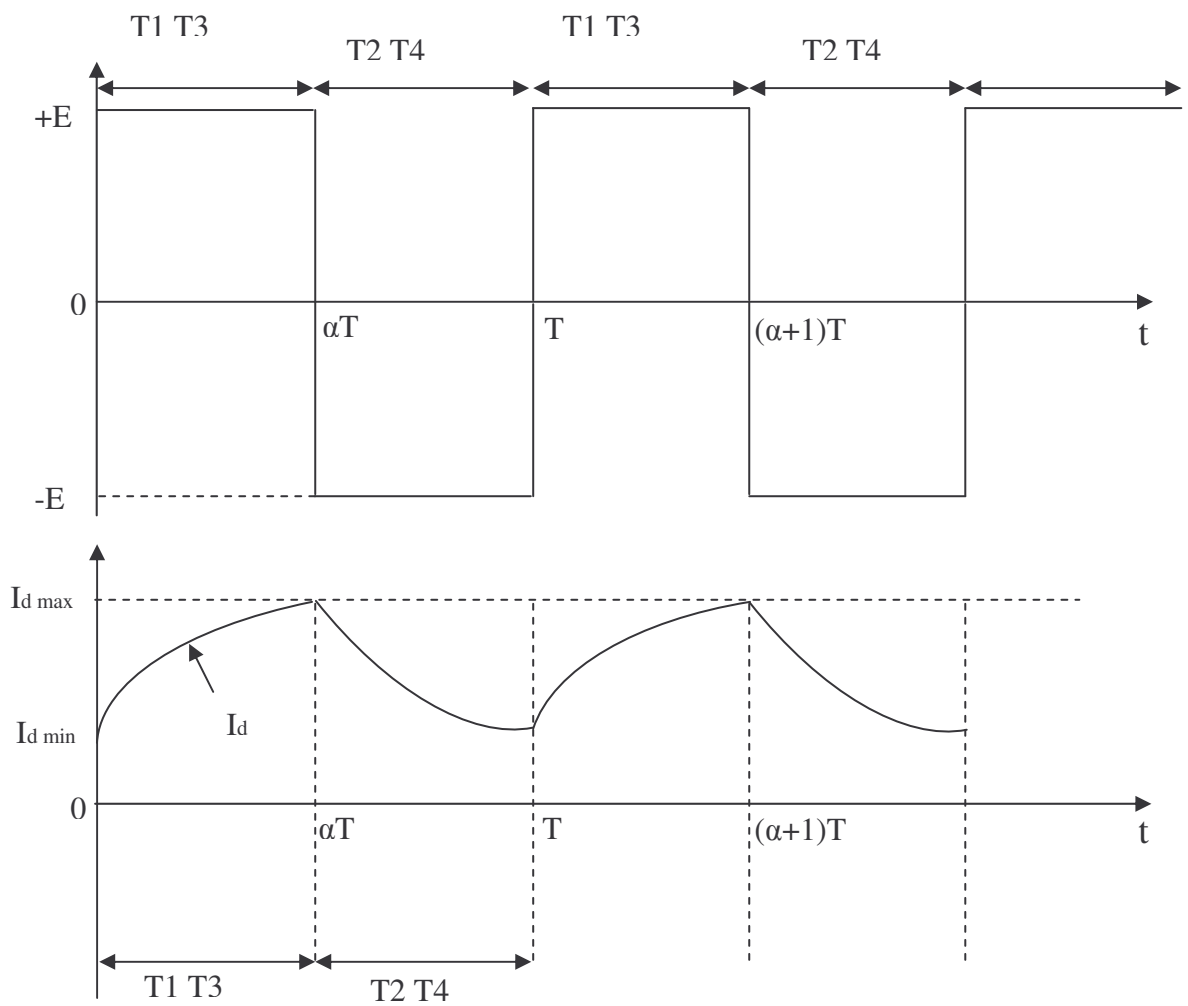
$$\left\{ \begin{array}{l} E \quad \text{si } T1=1, T3=1 \\ -E \quad \text{si } T2=1, T4=1 \end{array} \right.$$

#### La valeur moyenne de la tension

$$\bar{U}_d = \frac{1}{T} \int_0^{\alpha T} E dt - \int_{\alpha T}^T E dt \Rightarrow \bar{U}_d = (2\alpha - 1)E$$

$$\text{Si } \alpha < 1/2 \Rightarrow \bar{U}_d < 0$$

$$\text{Si } \alpha > 1/2 \Rightarrow \bar{U}_d > 0 \quad \text{le montage est réversible en tension}$$



## Conclusion

Dans ce chapitre nous avons présenté les moteurs à courant continu sous leurs aspects théorique et pratique ainsi que leur principe de fonctionnement et leur commande. Le principe de fonctionnement des hacheurs à quatre quadrants a été aussi présenté.



# Conception et Réalisation de l'AGV

### IV-1 Introduction

L'idée principale de ce projet été de réaliser un véhicule puis de le guider automatiquement en l'obligeant à suivre une trajectoire physique. La correction de la position du véhicule se fait en introduisant des données (acquisition) qui reflètent la position actuelle du véhicule, et selon ces données, la position du véhicule sera corrigée. Ce qui fait que ce travail a été divisé en deux parties, une partie acquisition et l'autre commande. Malheureusement, la première partie d'acquisition n'a pas été réalisée et vu la limitation du temps nous avons réalisé seulement la deuxième partie.

Notre projet consiste alors à réaliser un véhicule guidé automatiquement qui sera capable de suivre un chemin bien déterminé (chemin programmé). Cependant différentes techniques peuvent être élaborées afin de permettre le guidage du véhicule. Notre technique utilise le guidage du véhicule depuis un point de départ vers un point d'arrivée tout en suivant une trajectoire donnée choisie par l'utilisateur.[8]

La réalisation de ce véhicule requiert diverses notions dans plusieurs domaines : électronique, informatique, électrotechnique et mécanique, ce qu'on appelle brièvement **Robotique**. Le déplacement du véhicule est effectué par le moyen de deux moteurs à courant continu (à aimant permanent) dont les axes sont liés directement avec des roues motrices à l'arrière. Une roue castor (roue folle) est placée à l'avant du véhicule, qui sera orientée suivant la vitesse différentielle des deux roues motrices.

La commande de ce véhicule sera assurée principalement par trois cartes de commande, une carte numérique à base de microcontrôleur PIC, une carte d'isolation et en fin une carte de puissance contenant des hacheurs en pont.

## **IV-2 Structure mécanique du véhicule**

### **IV-2-1 Le moteur à courant continu utilisé**

Notre véhicule possède trois roues, deux roues motrices à l'arrière et une roue castor (folle) à l'avant. Il est mis en mouvement à l'aide de deux moteurs à courant continu.

Les moteurs sont disposés sur le véhicule d'une façon qu'ils ne supportent aucune charge, donc leur seule fonction est de faire tourner les roues, ce qui permet au véhicule de supporter des charges considérable.



Fig.IV.1: Photo du moteur utilisé

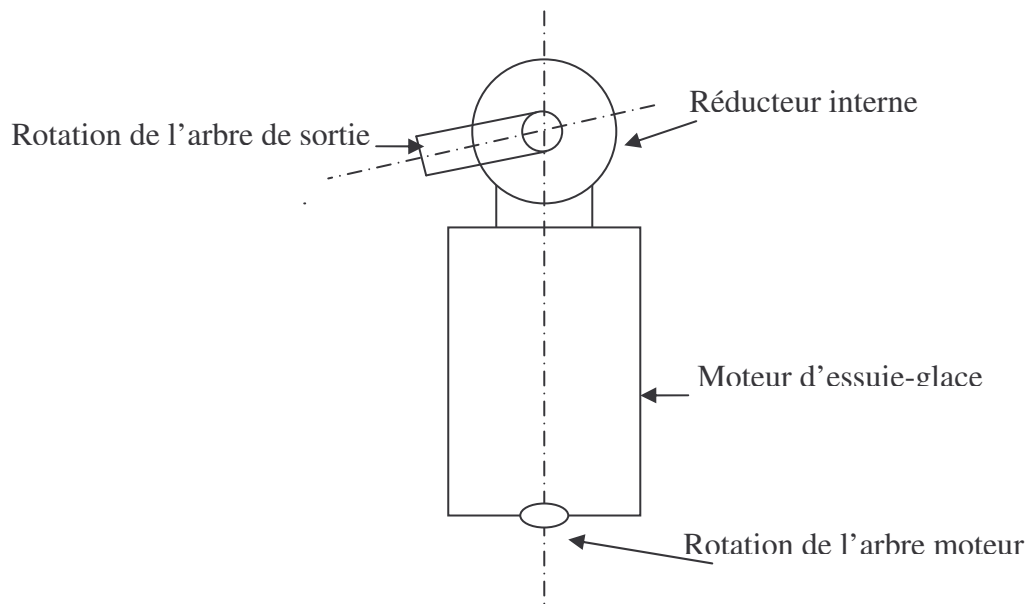


Fig.IV.2: Le moteur à courant continu utilisé

#### IV-2-2 Dimensions du véhicule et montage des moteurs

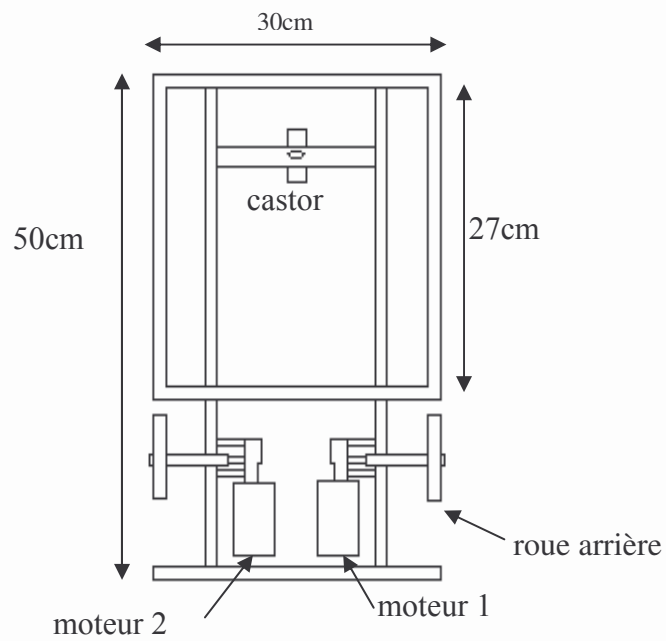
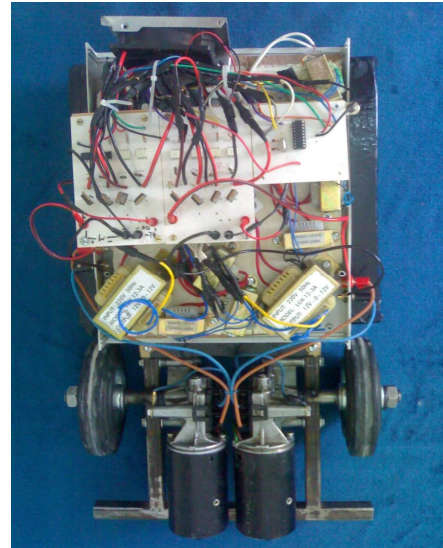


Fig.IV.3:Dimensions du véhicule et montage des moteurs



A: Avec couvert



B: Sans couvert

fig.IV.4: Photo du véhicule réalisée

### IV-2-3 Principe de guidage

Le déplacement du véhicule dépend des vitesses des deux roues motrices à l'arrière :

- Pour la marche avant du véhicule, les deux moteurs tourneront à la même vitesse et en même sens.
- Pour la marche arrière, les deux moteurs tourneront à la même vitesse mais au sens inverse.
- Pour tourner à droite lentement, on arrête le moteur droit et on laisse le moteur gauche en marche ; et pour tourner à droite rapidement, on tourne le moteur droit en sens inverse, et on laisse le moteur gauche en marche.
- Pour tourner à gauche lentement, on arrête le moteur gauche et on laisse le moteur droit en marche ; et pour tourner à gauche rapidement, on tourne le moteur gauche en sens inverse, et on laisse le moteur droit en marche.
- Pour arrêter le véhicule, on arrête les deux moteurs.

### IV-3 Conception et réalisation des cartes électroniques

Le schéma block de notre système est représenté dans la figure.IV.4 où nous remarquons que nous avons principalement cinq modules :

- ✓ Modules d'alimentations
- ✓ Module de la carte PIC
- ✓ Module des programmes contenant les trajectoires
- ✓ Module de la carte d'isolation optique
- ✓ Module des cartes hacheurs

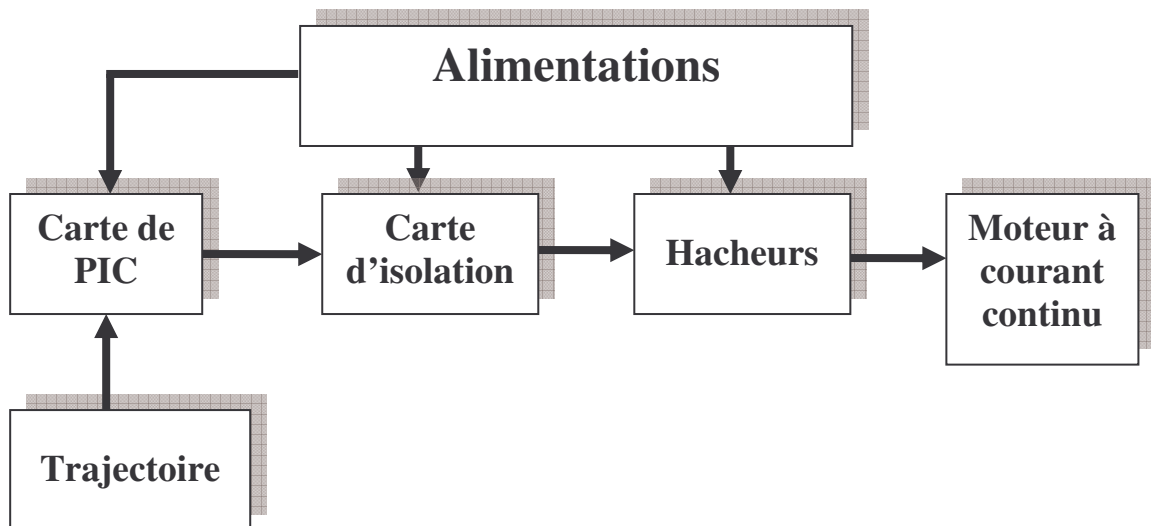


Fig.IV.5: Schéma block du système proposé

#### IV-3-1 Modules d'alimentations

L'alimentation électrique joue un rôle très important dans le domaine de l'électronique, car on ne peut pas faire fonctionner un montage électrique contenant des dispositifs à base de semi conducteurs (transistors, circuits intégrés, ...) sans l'alimenter.

Nous avons utilisé dans notre cas des alimentations classiques basées sur le principe suivant :

- ✓ Abaisser la tension alternative du réseau (220V) par transformateur abaisseur vers 6 ou 12 V alternatives.
- ✓ Redresser ensuite les tensions abaissées par le biais de ponts de diodes dans le but d'obtenir des tensions continues mais variables.
- ✓ Filtrer les tensions redressées par un filtre passe bas pour obtenir des tensions constantes continues légèrement ondulées.
- ✓ Réguler les tensions filtrées par des régulateurs dans le but d'éliminer les ondulations.

➤ **Alimentation du PIC (5V)**

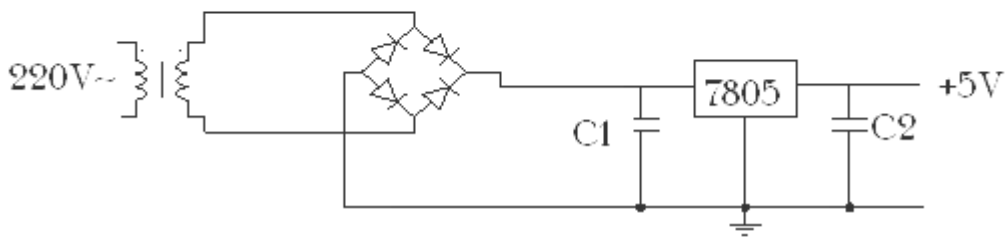


Fig.IV.6: Alimentation 5V

Cette alimentation délivrant 5 Volts a été réalisée par les composants suivants :

- Transformateur  
220 V/2\*6V-300mA
- Pont de diodes  
1A/50V
- Condensateur de filtrage  
C1= 470μF/35
- Condensateur de découplage  
C2= 470μF

### ➤ Alimentations des hacheurs (12V)

Le passage de la théorie vers la pratique n'est pas évident, cela peut être remarqué facilement dans la réalisation de la commande des hacheurs où nous avons rencontré des difficultés. Dans une telle conception, le problème rencontré est appelé Masses flottantes, ce qui rend difficile à commander un hacheur par une seule alimentation. Pour résoudre ce problème, nous avons opté d'utiliser plusieurs alimentations (6 alimentations) dans le but d'assurer des commandes qui peuvent activer et désactiver les transistors MOSFETs des hacheurs (voir fig.IV.18).

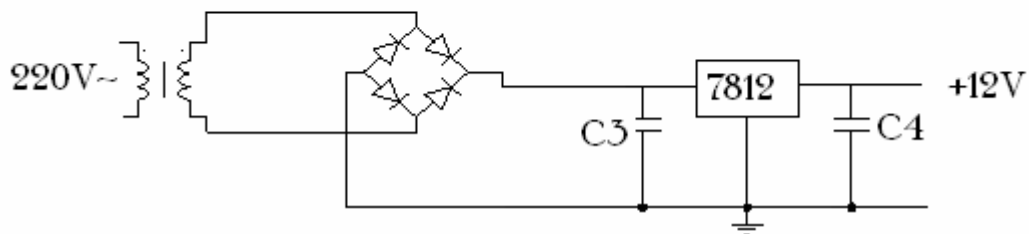


Fig.IV.7: Alimentation 12V

- Transformateur

220V/2\*6V-600mA

- Pont de diodes

1A/50V

- Condensateurs

1000 $\mu$ F/63V et 100nF

### ➤ Alimentations des Moteurs (12V)

Pour faire tourner les moteurs, deux alimentations ont été réalisées, où la seule différence par rapport aux alimentations des hacheurs est l'ampérage élevé :

220V/2\*6-3A

qui est nécessaire pour faire tourner les moteurs.

### IV-3-2 Carte de commande

Le pic 16F84 est le cœur de notre système, où on peut dire qu'il est le maître et les autres modules sont des esclaves. La description détaillée du PIC a été déjà vue au deuxième chapitre.

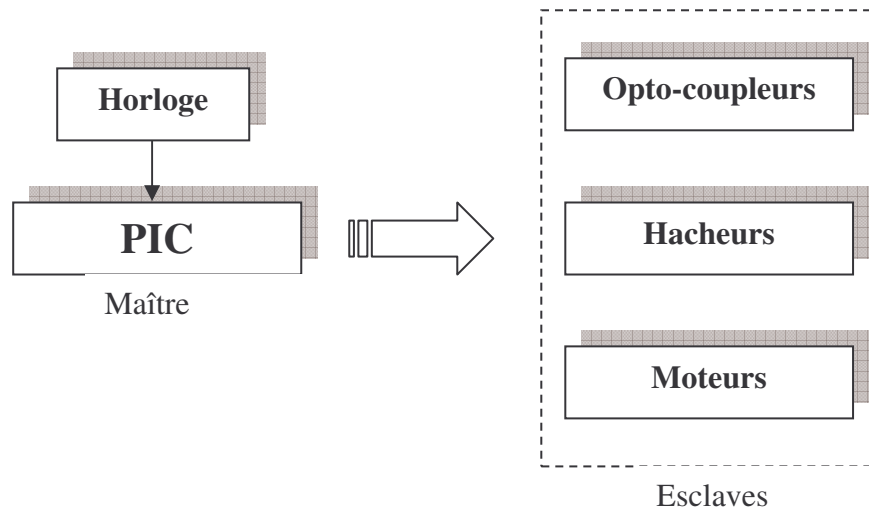


Fig.IV.8: Description Maître-Esclave

Le pic 16F84 est piloté par une horloge de 4MHz dont le circuit de base est le quartz Q lié avec deux condensateurs comme le montre la fig.IV.10.



Fig.IV.9: Pic 16F84

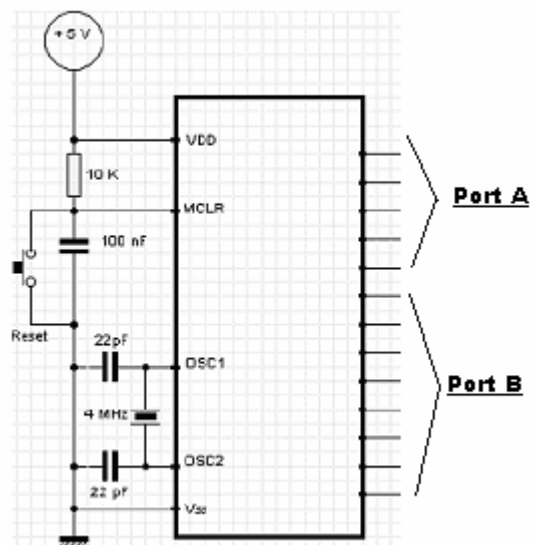


Fig.IV.10: Horloge du PIC16F84 à base du quartz

Puisque le véhicule est guidé par deux moteurs, ce qui fait deux hacheurs, chaque hacheur a besoin de quatre lignes de commandes, donc, en tout nous avons besoin de huit lignes de commande, c'est pour cette raison que nous avons choisi l'utilisation du port B formé de huit lignes qui seront programmées en sortie.

La trajectoire qui sera suivie par le véhicule est donnée sous forme d'un programme qui sera chargé dans la mémoire du pic. Le programme est écrit en langage C dans un environnement de programmation appelé HTPIC. L'HTPIC est un langage spécifique pour programmer les microcontrôleurs soit en langage évolué (C) soit en langage assembleur. Après compilation du code source, l'HTPIC nous donne un code machine Hexadécimale. Ce dernier code sera chargé dans le PIC par le biais d'un utilitaire (software) de programmation des pics appelé IC-PROG et d'un programmeur de pics.



Fig.IV.11: Programmeur de pics

### IV-3-3 Opto coupleurs :

L'utilisation de l'optocoupleur assure une isolation entre la carte de commande et la carte de puissance, afin d'éviter tout problème d retour de masse et bruits électriques. [3]

Lorsque les moteurs démarrent ou s'arrêtent, d'importants pics de courant peuvent se produire. Ceux ci pourraient être destructeurs pour l'électronique de commande. Nous utiliserons donc des optocoupleurs afin d'avoir une isolation

électrique parfaite entre les deux parties (commande et puissance). Les optocoupleurs utilisés ici sont des optocoupleurs classiques (type 4N25).

Le principe de l'optocoupleur est de transformer un signal électrique vers un signal optique, puis le retransformer vers sa forme électrique initiale, ce qui va nous donner une isolation physique qui matérialise réellement un vide (isolation).

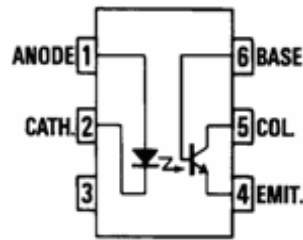


Fig.IV.12: Schéma de l'optocoupleur 4N25

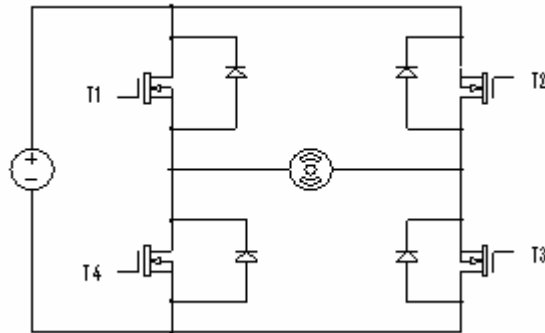
De cette façon, pour un faible signal commande en entrée, on a un fort signal commande en sortie (signal de commande dans la partie puissance).

Nous citons ici quelques caractéristiques techniques de l'optocoupleur 4N25 :

- $V_f$  max : 1,5-5 mA
- $V_{CEO}$  min (base ouverte) : 30V.
- $V_{CE}$  (sat) max : 0,5V-mA
- Taux de transfert : 2500V.
- Commutation ton ou fréquence : 300KHz.

#### IV-3-4 Hacheurs

Les moteurs utilisés sont des moteurs à courant continu, donc leur vitesse de rotation (donc la vitesse du robot) est proportionnelle à la tension d'alimentation. Il va donc falloir être capable de produire une tension variable et commandable. Pour cela il existe un montage appelé montage en pont H. Ce montage est constitué de 4 transistors avec 4 diodes montés en parallèle comme suit :



Nous avons donc maintenant un système qui à partir de quatre signaux In1, In2, In3 et In4 est capable de commander un moteur dans les deux sens et de l'arrêter.

In1	In2	In3	In4	T1	T2	T3	T4	Etat moteur	Schéma
0	0	0	0	bloqué	bloqué	bloqué	bloqué	Arrêt	A
1	0	1	0	Saturé	bloqué	saturé	Bloqué	Marche Avant	B
0	1	0	1	bloqué	saturé	bloqué	saturé	Marche arrière	C

Tab.IV.1: Tableau de fonctionnement de l'hacheur

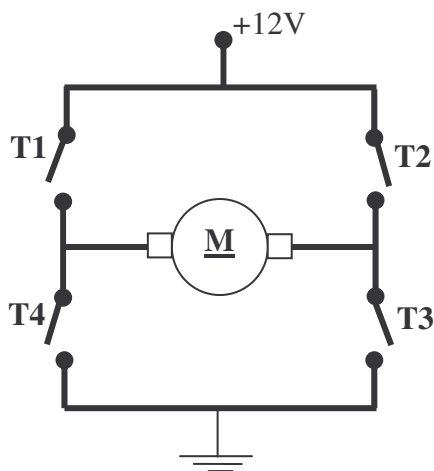


Schéma A

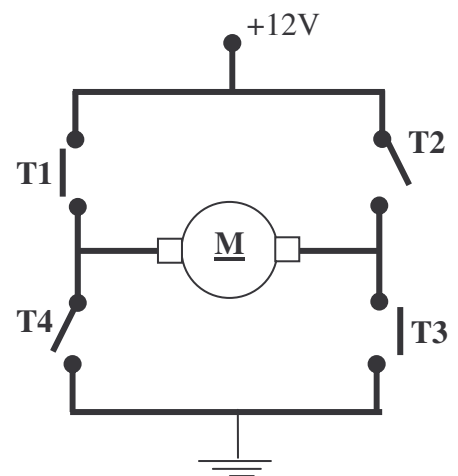


Schéma B

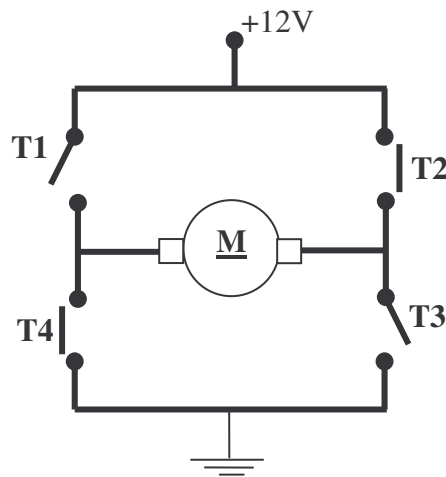


Schéma C

Nous avons donc maintenant un système qui à partir de quatre signaux  $In_1$ ,  $In_2$ ,  $In_3$  et  $In_4$  est capable de commander un moteur dans les deux sens et de l'arrêter.

Il va maintenant falloir faire varier la vitesse de celui ci. Pour cela, nous allons utiliser la technique du hachage. Qu'est ce que c'est ? Il s'agit d'une technique qui consiste à ne pas envoyer un signal qui soit un « 1 » pur ou un « 0 » pur mais un mélange des deux. En effet, si l'on envoi un signal qui bascule d'un état à un autre, à une fréquence suffisamment basse pour que les transistors puissent suivre les changements d'états, alors, si cette fréquence est suffisamment haute, les moteurs n'ont pas le temps de réagir à ce changement et ne voient qu'une tension moyenne. Ainsi pour un signal de rapport cyclique  $\alpha = T_H/T$ , les moteurs agiront comme si la tension a leurs bornes était de  $\Delta V_{cc}$ .

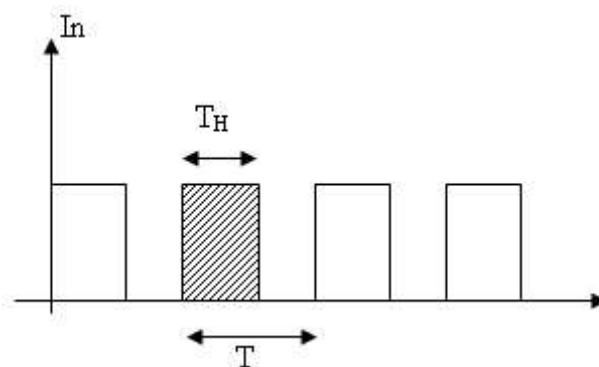


Fig.IV.14:

#### IV-4 Circuits imprimés

Nous avons réalisé trois circuits imprimés d'une seule face.

##### IV-4-1 Circuits imprimés des alimentations

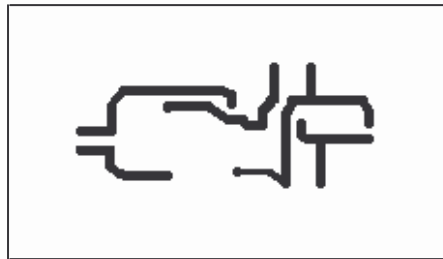


Fig.IV.15: Circuit imprimé de l'alimentation

##### IV-4-2 Circuit imprimé d'une carte d'isolation et d'un hacheur

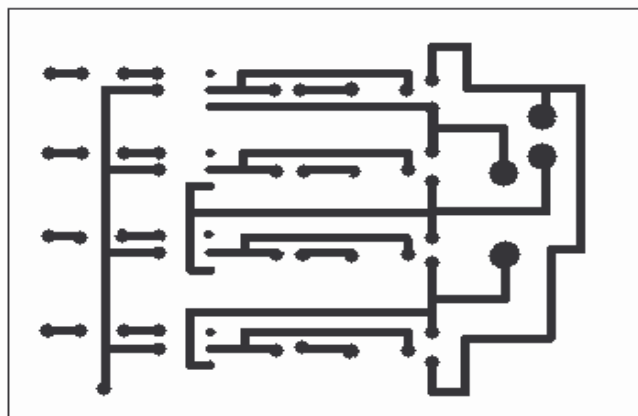


Fig.IV.16: Circuit imprimé d'isolation et hacheur

##### IV-4-3 Circuit imprimé de la carte de commande (pic 16F84)

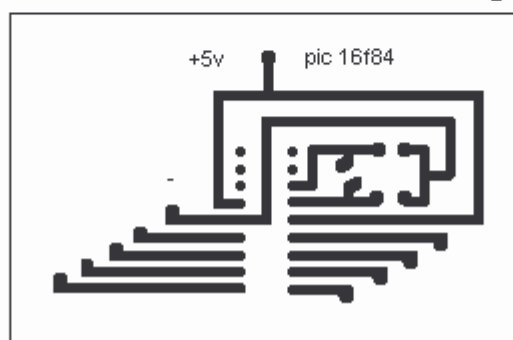


Fig.IV.17 Circuit imprimé de la carte de commande

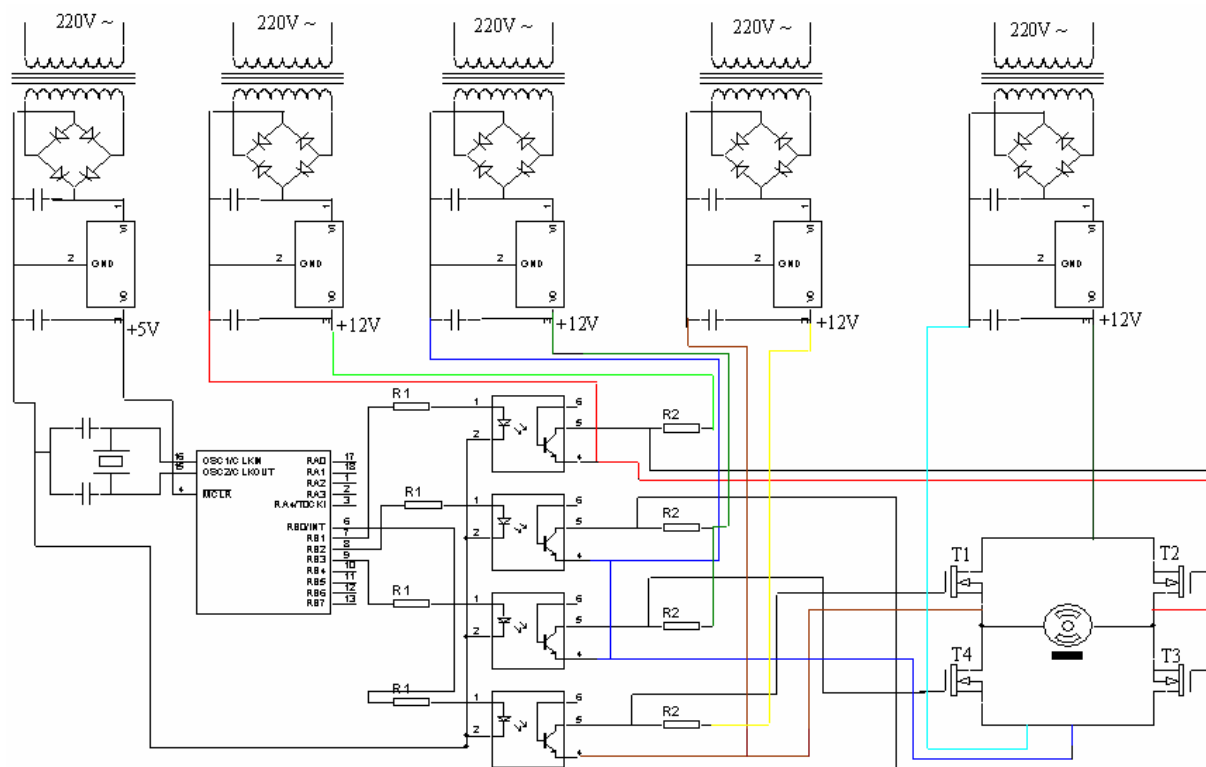


Fig.IV.18: Schéma électrique détaillé de la carte

#### **IV-5 Conclusion**

Dans ce chapitre nous avons présenté l'essentiel de notre travail qui a consisté la réalisation et le guidage de l'AGV.

Le but tracé a été atteint et la commande de l'AGV a donné une totale satisfaction et un grand succès.

## Conclusion générale

L'originalité de notre travail est de maîtriser le domaine de la robotique, car la robotique est une science qui touche plusieurs disciplines (Electriques et mécaniques).

Durant la réalisation de ce projet nous avons acquis beaucoup de connaissances concernant les véhicules à guidage automatique, l'interfaçage et comment utiliser les microcontrôleurs pour le contrôle des processus en temps réel ceci à travers des programmes.

La première partie réalisée été la structure mécanique du véhicule où nous avons appliqué nos connaissances de mécanique dans ce point où la précision et la symétrie sont très indispensables.

La deuxième partie réalisée est un ensemble de carte électroniques tel que :

- Carte d'isolation
- Carte de puissance (Hacheurs en pont)
- Carte de commande à base de PIC 16F84
- Cartes d'alimentation

La troisième partie est un ensemble de programmes de commande en langage C introduits par l'utilitaire HTPIC qui permet de compiler le fichier C puis fournir sa version en code objet (Hexadécimal). Les codes hexa sont ensuite chargés dans la mémoire du PIC par le biais d'IC-PROG et d'un programmeur.

Le système réalisé a donné entière satisfaction et fonctionne correctement, où plusieurs trajectoires ont été testées avec succès et la navigation du véhicule été excellente.

Rappelons que ce projet été divisé en deux partie, partie acquisition et partie commande, et que nous avons réalisé seulement la partie commande.

Ce projet est intéressant et mérite d'être continué. Nous proposons ici quelques améliorations qui peuvent être apportées pour une éventuelle continuation :

- Réalisation d'une carte d'acquisition qui nous détermine la position actuelle du véhicule en but de le repositionner.
- Installation de capteurs à ultrason pour la détection d'obstacles
- Installation d'une télécommande infrarouge
- Installation d'autres systèmes de guidage
- Installation de software de navigation et d'intelligence artificielle.