

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique
Laboratoire de Commande des Processus



Mémoire de Magistère

Spécialité : **Automatique**
Option : **Robotique et Productique**

Présenté par
OUENDI Fatima

Ingénieur d'Etat en Automatique de l'UMMTO

Thème

Etude et Conception des Servo-Entraînements dédiés à la Robotique

Soutenu devant le jury composé des membres :

Président : M.TADJINE.....Maître de conférences à l'ENP
Rapporteurs : O.STIHI.....Chargé de cours à l'ENP
T.ZEBBADJI.....Chargé de cours à l'ENP
Examineurs : H.CHEKIREB.....Maître de conférences à l'ENP
B.HEMMICI.....Chargé de cours à l'ENP
R.ILOUL.....Chargé de cours à l'ENP

المخلص

الهدف من هذا العمل هو إدخال و تجريب خوارزميات التحكم باستعمال بطاقة الكترونية مكونة أساسا من ميكرومراقب البطاقة هذه مصممة و منجزة من أجل التحكم في تحريك ذراع آلي مستوي مكون من 2 درجات حرية من النوع SCARA . هذا الآلي محرك بواسطة 3 محركات خطوة بخطوة. لقد قدمنا دارة تحكم و حققنا اتصال مع الكمبيوتر بواسطة الرابطة . USART

كلمات مفتاحية :

التحكم في الأذرع الآلية, الفعاليات خطوة بخطوة المكمرومراقب USART .

Résumé

L'objectif de ce travail est d'implémenter et de tester des algorithmes de commande en utilisant une carte électronique à base de microcontrôleur . Cette carte est conçue et réalisée pour piloter un bras manipulateur planaire à trois degrés de liberté de type SCARA.

Ce robot est actionné par trois moteurs pas à pas. Une interface graphique avec une liaison USART (PC-Carte électronique) a été développée pour le pilotage du robot à partir d'un PC.

Mots clés :

Commande des robots manipulateurs, actionneurs pas à pas, microcontrolleurs16f877, USART

Abstract

The aim of this work is to implement and test algorithms of control by using an electronic cart based on a microcontroller. This carte is conceived and realised for piloting a manipulator arm with three degrees of freedom of SCARA type.

This robot is actionned by three stepper motors. A graphical interface with a USART (PC-electronic carte) has been developed for the pilotage of the robot from a PC.

Key words:

Control of manipulator robots. Stepper motor actuators, microcontrollers. USART.

REMERCIEMENTS

Je tiens à remercier mes promoteurs M^r O.STIHI et M^r T.ZEBBADJI pour m'avoir dirigé durant ce travail.

Je remercie M^r H.CHEKIREB, M^r M.DAHDAH et M^{me} H.SAHRAOUI pour leur aide inestimable, qu'ils trouvent ici l'expression de ma reconnaissance à l'égard de tous les efforts qu'ils ont fourni tout au long de ces deux années.

Je remercie M^r M.TADJINE de m'avoir fait l'honneur d'accepter la présidence de ce jury.

Que messieurs H.CHEKIREB, B.HEMMICI, et R.ILLOUL trouvent l'expression de mes remerciements de m'avoir fait l'honneur d'accepter l'examen de ce travail.

Pour leurs encouragements, leur soutien moral et la patience qu'ils m'ont manifesté durant mes études, je remercie tous les membres de famille.

DÉDICACES

*Je dédie ce travail à tous ceux qui me sont chers :
Ma famille et mes amis.*

*« C'est parce qu'on imagine
simultanément tous les pas qu'on
devra faire qu'on se décourage,
alors qu'il s'agit de les aligner un
à un. »*

*Marcel Jouhandeau in « De la
grandeur »*

SOMMAIRE

SOMMAIRE

INTRODUCTION GENERALE.....	1
CHAPITRE I MOTIVATION ET POSITION DU PROBLEME.....	3
I. 1 INTRODUCTION.....	3
I.2 LES OBJECTIFS DE LA ROBOTISATION.....	3
I.3 LE ROBOT.....	4
I.4 FONCTIONS DE BASE.....	4
I.5 TYPE DE TACHE.....	5
I.6 CONCEPTION ET REALISATION DU ROBOT.....	5
I.6.1 ETUDE DU BRAS.....	5
I.6.2 FONCTIONNEMENT DE LA PINCE.....	6
I.6.3 CHOIX DES MOTEURS.....	7
I.6.4 ROLE DES TROIS MOTEURS PAS A PAS.....	8
I.7 MODELISATION DU ROBOT SCARA.....	9
I.7.1 MODELE GEOMETRIQUE DIRECT ET INVERSE DU ROBOT SCARA.....	9
I.7.2 ESPACE DE TRAVAIL.....	10
I.7.3 LIMITES ARTICULAIRES DU ROBOT SCARA.....	10
I.8 GENERATION DE MOUVEMENT.....	11
I.9 LA STRUCTURE DE COMMANDE DU ROBOT.....	12
I.10 CONCLUSION.....	12
CHAPITRE II SYSTEME D'ENTRAINEMENT DU ROBOT.....	13
II.1 INTRODUCTION.....	13
II.2 DOMAINE D'UTILISATION.....	13
II.3 TYPES DES MOTEURS PAS A PAS.....	14
II.3.1 MOTEURS PAS A PAS A RELUCTANCE VARIABLE.....	14
II.3.2 MOTEURS POLARISES.....	15
❖ LES MOTEURS A AIMANT PERMANENTS.....	15
❖ LES MOTEURS HYBRIDES.....	15
II.4 COMPARAISON DES MOTEURS PAS A PAS.....	16
❖ MOTEURS A AIMANT PERMANENT.....	16
❖ MOTEURS A RELUCTANCE VARIABLE.....	16
❖ MOTEURS HYBRIDES.....	17
II.5. COMMANDE DES MOTEURS PAS A PAS.....	17
II.5.1 FONCTIONNEMENT A PAS ENTIERS.....	17
❖ COMMANDE EN MODE 1.....	17
❖ COMMANDE EN MODE 2.....	19
II.5.2 FONCTIONNEMENT A DES PAS REDUITS.....	20
❖ COMMANDE EN MODE 3.....	20
❖ COMMANDE EN MODE 4.....	20
❖ COMMANDE EN MODE 5.....	21
II.6 ALIMENTATION DES MOTEURS PAS A PAS.....	22
II.6.1 OBJECTIFS D'UNE BONNE ALIMENTATION ET CLASSIFICATION.....	22
II.6.2 TYPES D'ALIMENTATION DES MOTEURS PAS A PAS.....	23
II.6.3 ALIMENTATION BIDIRECTIONNELLE.....	23
II.6.4 CONSTITUTION DU L298.....	24

II.7 CONCLUSION.....	24
CHAPITRE III ELECTRONIQUE DE COMMANDE.....	26
III.1 INTRODUCTION.....	26
III.2 LES COMPOSANTS CONSTITUANTS LA CARTE DE COMMANDE	26
III.3 STRUCTURE GLOBALE.....	28
III.3.1 LE MICROCONTROLEUR.....	28
III.3.1.1 IDENTIFICATION DES PIC.....	29
III.3.1.2 ETUDE DU MICROCONTROLEUR 16F877.....	29
III.3.1.3 CARACTERISTIQUES PRINCIPALES DU 16F877.....	30
III.3.1.4 LE PORT SERIE DU PIC 16F877 USART.....	31
III.3.2 LE MAX 232.....	31
III.3.3 COMMUNICATION SERIE RS232.....	32
III.3.3.1 BROCHAGE DU PORT SERIE	33
III.3.3.2 BROCHAGE DU PIC 16F877.....	33
III.3.4 BLOC DE VISUALISATION (L’AFFICHEUR LCD).....	34
III.3.4.1 LES PINS DE CONTROLE.....	35
III.3.5 LA PARTIE CONTROLE.....	35
III.3.6 BLOC D’ACQUISITION	35
III.3.7 L’ALIMENTATION DU CIRCUIT.....	35
❖ ALIMENTATION DE LA CARTE.....	35
III.4 CONCLUSION.....	37
CHAPITRE IV CONCEPTION LOGICIELLE DE LA COMMANDE	38
IV.1 INTRODUCTION.....	38
IV.2 LES OUTILS DE DEVELOPPEMENTS.....	38
IV.2.1 LE SIMULATEUR.....	38
IV.2.2 LE PROGRAMME A IMPLEMENTER.....	39
IV.3 FONCTIONNEMENT GENERAL DU SYSTEME DE COMMANDE.....	40
IV.3.1 DEROULEMENT DE L’ORGANIGRAMME.....	40
IV.3.1.1 MODIFICATION DU SENS DE ROTATION.....	42
IV.3.1.2 GESTION DE LA VITESSE	42
❖ PROCEDURE DE GESTION DE LA VITESSE	42
❖ GESTION DES INTERRUPTIONS DU TIMER0.....	42
❖ LES REGISTRES SPECIAUX DU PIC.....	44
a) LE REGISTRE STATUS.....	44
b) LE REGISTRE OPTION.....	44
c) LE SYSTEME D’INTERRUPTION	46
d) LA MEMOIRE EEPROM DATA.....	46
e) LES ENTREES SORTIES	48
f) LE TIMER2.....	48
IV.3.2 LES MODES DE FONCTIONNEMENT DU ROBOT SCARA.....	48
❖ LE MODE AUTO.....	48
❖ LE MODE STEP ANGLE.....	48
❖ LE MODE TACHE AUTOMATISEE.....	48
IV.4 INTERFACE LOGICIELLE SUR PC.....	49
IV.4.1 APERCU SUR DELPHI.....	49
IV.4.2 LE COMPOSANT C.PORT	50

IV.4.2.1 INSTALLATION D'UN NOUVEAU COMPOSANT.....	50
IV.4.3 LA CONFIGURATION DU PORT SERIE	51
IV.4.3.1PROTOCOLE DE COMMUNICATION.....	51
a) FORMAT DES TRAMES.....	51
b) DEBIT DES DONNEES.....	52
IV.4.3.3 TRANSMISSION DES DONNEES.....	52
IV.4.4 L'INTERFACE GRAPHIQUE.....	52
IV.4.4.1INITIALISATION DE LA POSITION DES MOTEURS.....	53
IV.4.4.2 LE DEROULEMENT DU PROGRAMME	53
IV.4.4.3 TEMPORISATION.....	54
LA FREQUENCE DE COMMUTATION DES BOBINES	54
IV.5 SELECTION DU MODE DE FONCTIONNEMENT.....	54
IV.5.1 FONCTIONNEMENT MANUEL.....	54
IV.5.2 FONCTIONNEMENT AUTOMATIQUE.....	54
IV.5.3 LE PROGRAMME	55
IV.6CONCLUSION.....	55
CONCLUSION GENERALE.....	56
REFERENCES BIBLIOGRAPHIQUES	57

INTRODUCTION GENERALE

Introduction Générale

La définition la plus précise du robot est un système automatique mécanisé capable d'effectuer une ou plusieurs tâches dans un environnement donné, de manière autonome et par l'exécution d'un programme. C'est une machine programmable capable d'exécuter diverses tâches répétitives, et en principe, d'adapter son comportement à certains aléas et événements perturbant le fonctionnement nominal prévu. Les tâches peuvent être de type point à point ou de type continu: suivie de courbes, suivi de surfaces. Les premières correspondent aux procédés les plus répandus dans l'industrie manufacturière dont le soudage par résistance, la palettisation, le perçage et la manutention. Tandis que les secondes, elles correspondent aux procédés de soudage à l'arc, collage, découpe, projection, peinture [1].

Les robots manipulateurs se substituent à l'homme ou prolongent son action en apportant précision, rapidité ou capacité à appliquer d'importants efforts [2]. Ils sont généralement constitués d'un bras manipulateur plus ou moins anthropomorphe mis en mouvement par des actionneurs électriques, hydrauliques ou pneumatiques. Par rapport à leurs concurrents hydrauliques et pneumatiques, les actionneurs électriques présentent un certain nombre d'avantages parmi lesquels :

- une énergie facilement disponible, soit à partir du secteur, soit à partir de batteries pour les engins autonomes.
- une adaptation aisée de l'actionneur et de sa commande du fait de la nature électrique de l'ensemble des grandeurs.

Les machines rencontrées dans le domaine de la robotique sont essentiellement de trois types :

- le moteur à courant continu représente la solution traditionnelle lorsqu'on a besoin de commander une vitesse ou une position, mais nécessite un asservissement.
- le moteur à courant continu sans balais est en fait une machine alternative associée à un commutateur électronique qui peut remplacer le moteur à courant continu classique avec des caractéristiques similaires.
- le moteur pas à pas est à la base un actionneur de positionnement ne nécessitant pas d'asservissement, mais peut être utilisé pour une commande de déplacement.

Grâce à sa capacité à contrôler la position et la vitesse, le moteur pas à pas est actuellement le principal élément intermédiaire entre les dispositifs de traitement d'information et le monde électromécanique extérieure. Cela est due au développement conjugué de l'électronique de puissance et, surtout, grâce à l'apparition de l'électronique numérique à forte intégration.

Vu ces qualités ces moteurs sont présents dans divers domaines comme :

- la traction des robots mobiles.
- le fonctionnement en moteur couple de grande puissance.
- l'indexage rotatif ou linéaire.

La commande numérique du moteur peut se faire de plusieurs façons. La méthode la plus performante permettant d'avoir de meilleures performances et une grande flexibilité est la commande électronique par microcontrôleur [3]. L'avènement des microcontrôleurs, qui associent au microprocesseur de base un programme intégré au circuit, ainsi que des périphériques et de la RAM, a permis de faire évoluer les montages vers plus de simplicité et de rapidité (les périphériques étant intégrés). Les microcontrôleurs sont aujourd'hui implantés dans la plupart des applications grand public ou professionnelles, il en existe plusieurs familles.

La société Américaine MICROCHIP Technologie a mis au point dans les années 90 un microcontrôleur CMOS : le PIC. Ce composant encore très utilisé à l'heure actuelle, est un compromis entre simplicité d'emploi, rapidité et prix de revient.

Dans ce sens nous décomposons notre travail en quatre chapitres

Dans le premier chapitre, nous nous intéressons aux méthodes de commande des manipulateurs ainsi que la présentation du bras utilisé puis nous avons justifié notre choix de la commande numérique à base de microcontrôleur. Un robot manipulateur d'application est un robot SCARA.

Le deuxième chapitre est dédié à l'étude des moteurs pas à pas. Pour ce faire nous passerons en revue les différents types de moteurs pas à pas ainsi que leurs principe de fonctionnement et leurs modes de commandes. La classification de ces moteurs, nous permettra ensuite de sélectionner la structure la mieux adaptée à l'application considérée.

Le troisième chapitre étudie l'optimisation de la commande du moteur pas à pas et du bras manipulateur en un temps minimum et traite de la conception et de la réalisation matérielle de la commande ainsi que la description des différents circuits qui constituent cette dernière.

Enfin, le quatrième chapitre est consacré à la conception logicielle du système, où les différents organigrammes régissant le fonctionnement de la commande ainsi que les outils de développements utilisés à cet effet sont décrits.

CHAPITRE I : MOTIVATION ET POSITION DU PROBLEME

Chapitre I

Position du Problème

I.1 Introduction

La conception, la fabrication et le pilotage des bras manipulateurs ont été rendus possible grâce aux connaissances scientifiques et techniques en mécanique et en électrotechnique. Ces machines sont destinées à augmenter les possibilités de l'homme pour la rapidité de ses déplacements et de ses actions, et pour l'amplification de sa force et de son champ d'action, notamment dans l'accomplissement de tâches pénibles, dangereuses et répétitives.

Les fonctions généralement concernées sont :

- le déplacement à plus ou moins longue distance sur terre, sur mer, dans l'air et sous l'eau.
- la manipulation au sens large du terme :
 - déplacer un outil pour saisir des objets ou de la matière, les transporter et les déposer,
 - exercer des efforts sur l'environnement ;
- la combinaison des deux fonctions précédentes.

Fonctionnellement un système robotisé se décompose en deux sous ensembles:

- a) **Le bras manipulateur:** qui agit sur l'environnement. Son état est mesuré par des capteurs, et modifié par des actionneurs.
- b) **Le système de commande:** il comprend les cartes électroniques qui réalisent l'asservissement des articulations aux positions de consignes de vitesses qui sont convertis et envoyées aux variateurs pilotant les actionneurs.

I.2 Les objectifs de la robotisation

Parmi les objectifs de la robotisation on peut citer :

L'adaptation à des environnements hostiles pour l'homme

L'adaptation à des tâches physiques ou intellectuelle pénibles pour l'homme.

L'augmentation de la sécurité.

L'adaptation à des contextes particuliers.

L'amélioration de la qualité du produit grâce à une meilleure répétitivité de la valeur ajoutée.

L'amélioration de la flexibilité du produit.

L'accroissement de la productivité du système sous forme d'une meilleure rentabilité et d'une meilleure compétitivité.

I.3 Le robot

Un robot manipulateur est généralement considéré comme un ensemble de corps rigides connectés en série par des articulations, avec une extrémité au sol, et l'autre libre (effecteur ou élément terminal). Il existe une multitude de robots manipulateurs qui sont utilisés actuellement. Nous citons par exemple, les robots anthropomorphes, les robots humanoïdes et les robots SCARA qui font l'objet de notre étude.

Le robot SCARA (Selective Compliance Assembly Robot Arm) est un robot série possédant quatre degrés de liberté dont trois sont rotoïdes et le quatrième degré est prismatique. Le robot SCARA a généralement pour tâche le déplacement d'objets d'un point à un autre appartenant tous deux à son espace de travail. Les deux premières articulations rotoïdes permettent le positionnement de l'organe terminal tandis que la troisième articulation rotoïde et l'articulation prismatique permettent respectivement l'orientation et le déplacement vertical de l'organe terminal.

Ce mémoire est consacré à l'application d'une commande numérique d'un robot SCARA à deux articulations rotoïdes montées sur une pince. Ce robot est actionné par trois moteurs pas à pas. La commande numérique est assurée par un microcontrôleur de la famille des PIC de Microchip.

I.4 Fonctions de base

Tout système robotisé comporte les fonctions suivantes :

- Acquérir les informations.
- Agir sur la matière d'œuvre.
- Dialoguer avec l'opérateur.
- Traiter les données.

On distingue classiquement 4 parties principales dans un robot manipulateur :

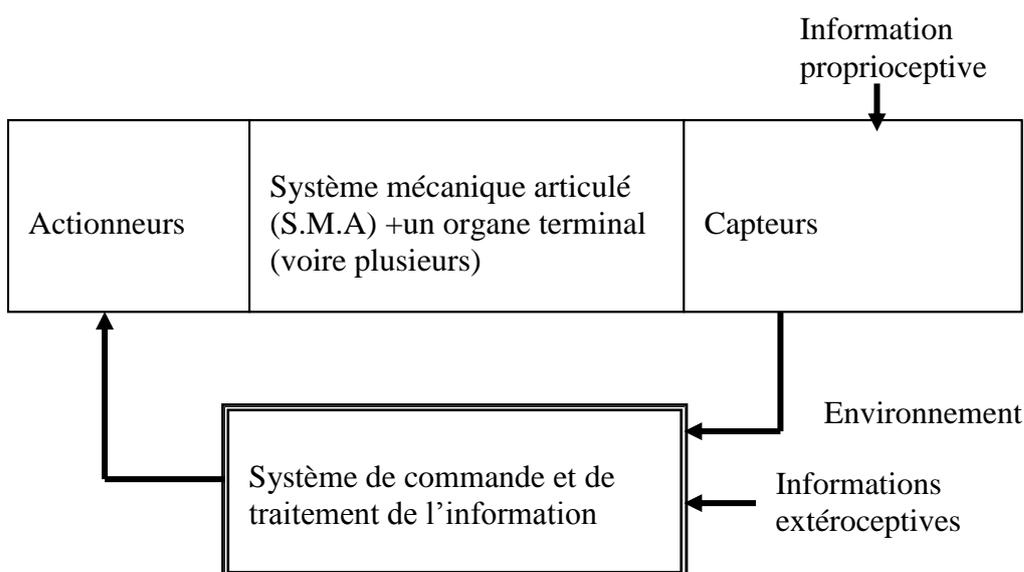


Fig.I.1 : schéma constituant un robot manipulateur

1.5 Type de tâche

Dans ce mémoire nous nous intéressons à une des tâches précitées qui est le perçage point par point des circuits imprimés et cela dans les deux sens avance et arrière grâce à des moteurs pas à pas de résolution de 24 pas par tours (comme le montre la fig.I.2)

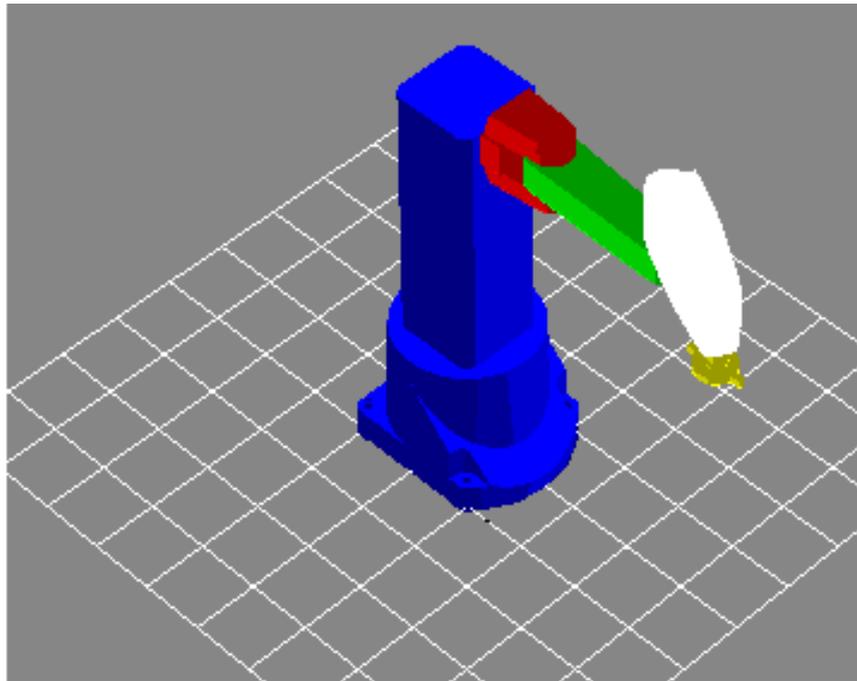


Fig.I.2 Robot de type SCARA.

I.6 Conception et réalisation du robot :

I.6.1 Etude du bras :

La fig.I.2 montre la structure mécanique du bras. Elle est réalisée pour satisfaire des critères de rigidité, d'économie et de fiabilité.

Le corps 1(voir la fig.I.2) constitue le support du bras. Il a une masse importante (de près de 10Kg) qui assure une inertie totale du robot lors de ses déplacements.

Les corps sont formés de 2 barres de longueur identique, assemblées par des boulons. Ces plaques sont en aluminium de faible épaisseur.

Les axes sont entraînés par des moteurs pas à pas à aimant permanent.

Le 1^{er} axe qui assure la translation du bras

Les paramètres du bras sont donnés par le tab.I.1.

Paramètres du robot	
L1	0.2m
L2	0.2m
M1	0.5Kg
M2	0.5Kg
h	0.5m
Poids de la pince	0.2Kg

Tab.I.1 : Paramètres du robot

I.6.2 Fonctionnement de la pince :

La pince forme un poignet qui sert au maintien d'un instrument quelconque. Le choix de la pince est effectué en fonction de la forme de la pièce à soulever, qui est cylindrique. C'est une pince à deux doigts à ouverture parallèle. Les ouvertures maximale et minimale dépendent du nombre de pas qu'effectue le moteur 3, compte tenu du diamètre de la pièce à soulever. La pince est actionnée par un moteur pas à pas 48 pas par tour de la marque MBB Minebea. Cette pince peut soulever une charge atteignant un kilogramme. Elle s'active dès que la position désignée par l'opérateur est atteinte.

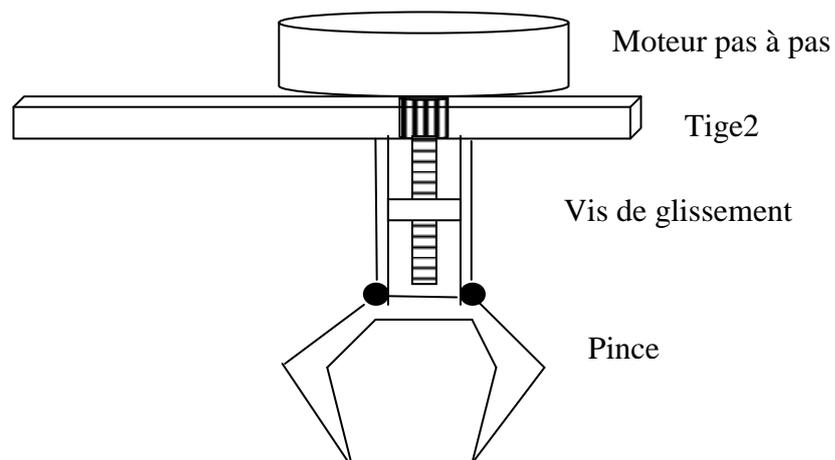


Fig.I.3: forme de la pince

I.6.3 Choix des moteurs :

Afin de mettre en mouvement notre robot, il était évident de le doter d'actionneurs électriques et plus précisément de moteurs électriques. Deux types de moteurs s'offraient à nous : les moteurs à courant continu et les moteurs pas à pas. Notre choix s'est porté sur l'utilisation des moteurs pas à pas pour les raisons suivantes :

Les moteurs pas à pas concurrencent dans certaines applications les servomoteurs à courant continu.

- Pour la précision, le mieux est le moteur pas à pas.
- Le pas à pas a la possibilité d'être monté derrière sur une réduction, et la précision de l'ensemble sera celle d'un pas plus le jeu de la réduction alors qu'avec un servomoteur, la précision sera celle du servomoteur plus celle de la réduction. D'ailleurs la plupart des servomoteurs sont limités à 180 ou 270 degrés alors qu'un pas à pas peut tourner indéfiniment. Par contre le servomoteur possède une position 0 et une position max, alors que le pas à pas ne sait jamais à quelle position il est, pour cela il faut rajouter un codeur en plus et faire les "0 position" à chaque démarrage du robot.
- Un des gros avantages du pas à pas est la durée de vie, le servomoteur contenant des pièces mécaniques (pignons, potentiomètre), et qui dit mécanique dit frottements et usure mécanique. Et pour le pas à pas, c'est entièrement électromagnétique, donc pas d'usure.
- Ensuite vient la différence de couple par rapport à la taille du moteur. Le gros avantage du servomoteur est d'offrir un grand couple pour sa toute petite taille, mais le gros inconvénient est qu'on est vite limité au niveau du couple à cause de la mécanique. Le moteur pas à pas, lui peut aller jusqu'à des couples incroyables, mais forcément la taille est plus grosse.
- Le servomoteur l'emporte largement sur le problème de l'encombrement et de poids. Par contre au niveau de la consommation, les servomoteurs consomment énormément de courant en pic, ce qui oblige à avoir des fils plus gros pour assurer le passage du courant, et des batteries permettant de forts pics de consommation.
- Au niveau de la programmation et utilisation en électronique, le servomoteur est très simple à piloter avec un signal PWM. Alors que le pas à pas nécessite un pont de puissance et un driver pour piloter les phases du moteur. Donc cela implique plus de travail et de complexité. C'est donc une des principales raisons à l'utilisation des servomoteurs en petite robotique avec leur petite taille.
- Mais avec le temps, l'utilisation se porte de plus en plus sur des moto réducteurs brushless avec codeur en quadrature. La plupart des utilisations ne nécessitent pas les intérêts du moteur pas à pas par rapport au couple du moteur à l'utilisation et surtout le couple de blocage.

- Alors que le moteur pas à pas se pilote pas par pas, c'est donc à l'utilisateur de commander chaque pas, ce qui permet de les compter et donc de faire facilement un positionnement relatif. Il n'y a par contre aucune information directe sur la position absolue du moteur.
- Les problèmes de glissement seront simplement dus à un manque de couple du moteur, donc un mauvais dimensionnement. Les moteurs pas à pas ont comme caractéristiques un couple classique et un couple de blocage. En blocage, ce couple monte très haut, donc dans le cas d'un bon dimensionnement, le moteur fonctionnera d'une manière optimale.
- Par contre le problème du pas à pas, c'est qu'en utilisation sans codeur, rien ne nous dit si le pas demandé a été exécuté ou non.
- Pour ce qui est de la vitesse minimale, il n'y en a pas. Un moteur pas à pas possède uniquement une vitesse maximale. Ceci est du au type de pilotage du moteur. Il ne se pilote pas en tension comme un moteur à courant continu (sauf certain pas à pas haute gamme), mais se commande pas par pas en faisant varier les différentes phases du moteur. Donc on peut aisément gérer une rampe d'accélération ou de décélération en faisant varier le temps entre chaque ordre d'exécution d'un pas.
- Pour la vitesse maximale, celle ci est due au temps de rotation d'un pas (ou demi pas), donc la fréquence de changement de phases ne doit pas être trop rapide par rapport aux caractéristiques du moteur, sinon cela pourrait immobiliser le moteur ou inverser son sens de rotation.

I.6.4 Rôle des trois moteurs pas à pas

Les 3 moteurs pas à pas montés sur le robot ont pour tâches successives :

Le moteur 1 assure le mouvement de l'ensemble constitué par les deux tiges et la pince.

Le moteur 2 assure le mouvement de la tige 2 et de la pince.

Le moteur 3 assure l'ouverture et la fermeture de la pince.

Les caractéristiques des trois moteurs pas à pas utilisés sont données au tab.I.2

Référence du moteur	PM55L-048
Nombre de pas par tour	48(7.5°/tour)
Méthode de commande	2-2 phases
Circuit de commande	Pont en h
Tension de service	24v
Courant /phase	800mA
Résistance par enroulement	30ohm
Résistance d'isolation	100M[ohm] MIN

Tab.I.2 : paramètres des moteurs utilisés

I.7 Modélisation du robot SCARA

I.7.1 Modèle géométrique direct et inverse du robot SCARA :

La modélisation géométrique étudie la géométrie du robot en vue de décrire sa position et son orientation, donc elle s'intéresse au mouvement sans tenir compte des forces qui le provoquent.

- **Modèle géométrique directe** : c'est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal en fonction des coordonnées articulaires. $X=f(q)$. Où X est le vecteur des coordonnées opérationnelles exprimées dans le repère de référence R_0 , et q les variables articulaires.
- **Modèle géométrique inverse** : il consiste en le calcul des coordonnées articulaires correspondantes à une situation donnée de l'organe terminal $q=g(x)$.

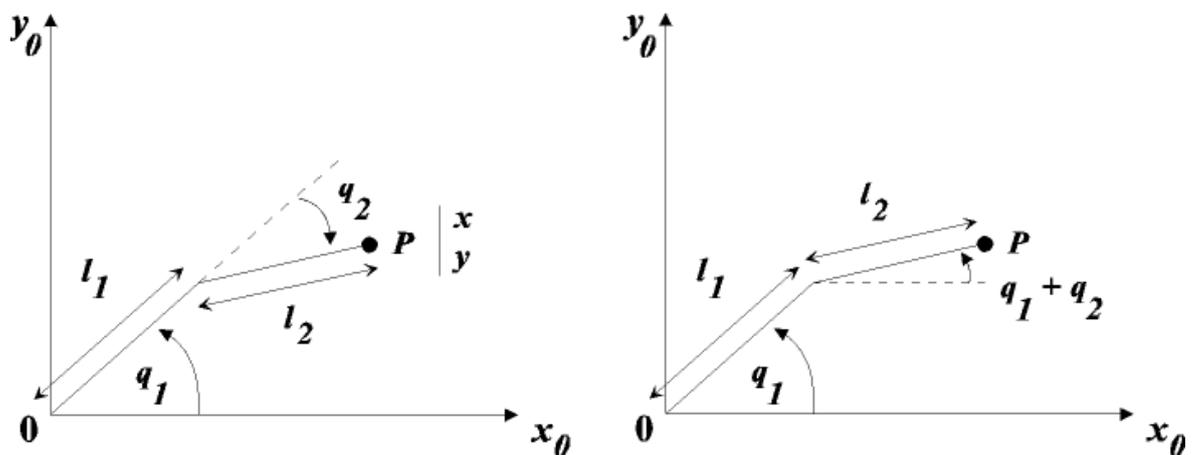


Fig.I.4 coordonnées opérationnelle et articulaires du robot

Le modèle géométrique est donné par la relation suivante :

$$\begin{cases} x = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) \\ y = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \end{cases}$$

C'est un système de 2 équations à 2 inconnues.

Une démarche analytique, procédant par substitution permet de déterminer le modèle géométrique inverse.

Il en résulte :

$$q_1 = \text{actg} \left(\frac{y(l_1 + l_2 \cos(q_2)) - x l_2 \sin(q_2)}{x(l_1 + l_2 \cos(q_2)) + y l_2 \sin(q_2)} \right)$$

$$q_2 = \pm \arccos \left(\frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2 l_1 l_2} \right)$$

I.7.2 Espace de travail :

Le domaine articulaire q admissible est en général un hyper parallélépipède défini par :

$$q_{i_{\min}} \leq q_i \leq q_{i_{\max}} \quad I=1 \text{ à } n.$$

Où $q_{i_{\min}}$ et $q_{i_{\max}}$ représentent les butées présentes au niveau de l'articulation I.

Dans notre cas, nous avons pris $l_1=l_2=l=0.2m$. Avec pour contrainte sur les articulations

$$-150^\circ \leq q_1 \leq 150^\circ, \quad -150^\circ \leq q_2 \leq 150^\circ$$

I.7.3 Limites articulaires du robot SCARA (butées mécaniques) :

Le robot SCARA présente des limitations en vitesses liées directement aux types de moteurs utilisés et de la commande appliquée, ainsi que des limitations mécaniques influençant directement l'espace de travail du robot appelées plus précisément butées mécaniques Fig.I.5, qui sont :

$$\begin{cases} -150 \leq q_1 \leq 150 ; & \text{Angle de rotation de la 1}^{\text{er}} \text{ articulation} \\ -150 \leq q_2 \leq 150 ; & \text{Angle de rotation de la 2}^{\text{ème}} \text{ articulation} \end{cases}$$

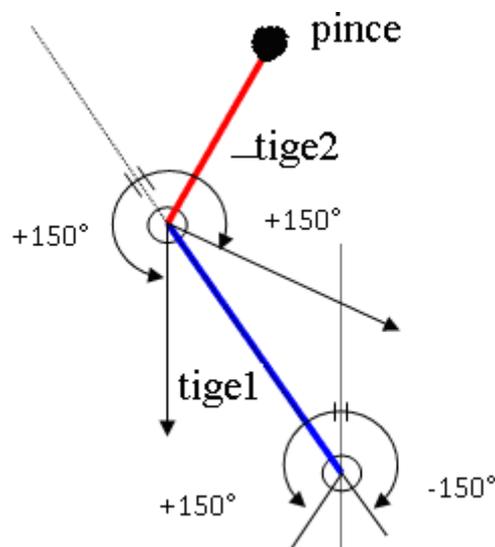


Fig.I.5 représentation des butées mécaniques

Les mouvements sont réalisés grâce à trois actionneurs pas à pas. Chacun des axes est équipé de deux fins de course permettant de déterminer l'arrivée de l'axe à la position maximale en évitant les butées mécaniques.

I.8 Génération de mouvement :

La tâche de déplacement d'un robot est spécifiée en définissant un chemin que le robot doit suivre. Un chemin est une séquence de points définis soit dans l'espace des tâches (opérationnel : afin de situer l'organe terminal), soit dans l'espace des configurations du robot (articulaire afin d'indiquer les valeurs des paramètres de liaison).

Ces points peuvent être :

- programmés par apprentissage.
- issus d'une base de données d'un système de CAO.

Le problème de la génération de mouvement est de calculer les séquences souhaitées (consigne) de variables articulaires ou de variables liées à l'organe terminal qui assurent le passage du robot par le chemin désiré.

Les trajectoires d'un robot peuvent être classifiées comme suit :

- les mouvements entre 2 points avec des mouvements libres entre les points,
- les mouvements entre 2 points via une séquence de points intermédiaires désirés, spécifiés notamment pour éviter les obstacles ; la trajectoire est libre entre les points intermédiaires,
- les mouvements entre 2 points, la trajectoire étant contrainte entre les points (trajectoire rectiligne par exemple),
- les mouvements entre 2 points via des points intermédiaires, la trajectoire étant contrainte entre les points intermédiaires.

Dans les deux premiers cas, la génération de mouvement peut se faire directement dans l'espace des configurations : elle se traduit par une séquence de positions articulaires constituant les consignes des asservissements.

Dans les deux derniers cas, la trajectoire étant fixée à tout instant dans l'espace opérationnel, il est préférable de raisonner dans cet espace. La loi de commande engendrée doit ensuite être transformée en consignes articulaires par le changeur de coordonnées.

Ces deux approches (génération de mouvement dans l'espace articulaire et génération de mouvement dans l'espace opérationnel) sont schématisées sur les figures suivantes.

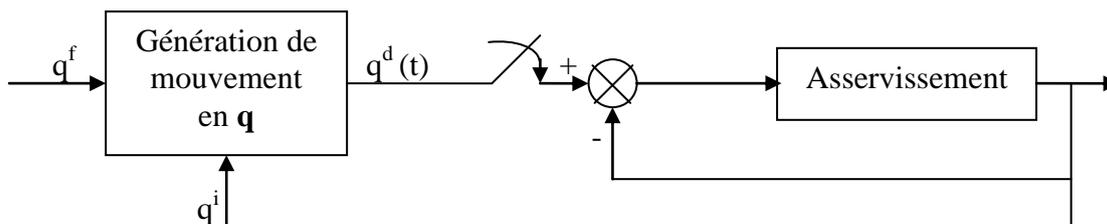


Fig.I.6. Génération de mouvement dans l'espace articulaire

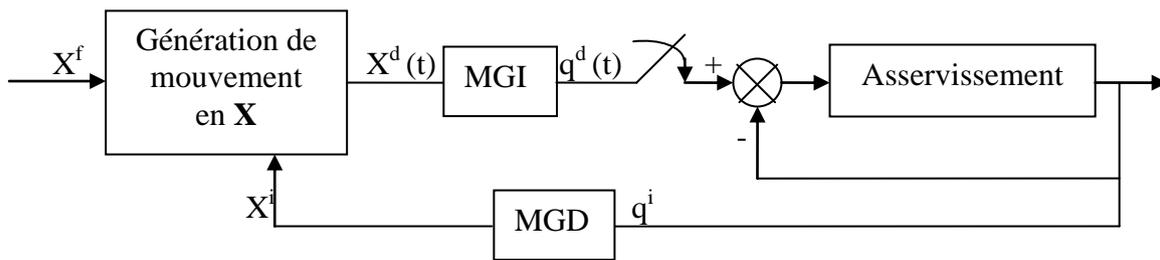


Fig.I.7. Génération de mouvement dans l'espace opérationnel

Plusieurs fonctions permettent le passage d'un point initial à un point final, parmi lesquelles on peut citer: l'interpolation polynomiale, la loi Bang –Bang et la loi bang –bang avec paliers de vitesse (loi trapèze).

I.9 La structure de commande du robot

La structure de commande est basée sur l'utilisation d'un microcontrôleur de type PIC. Le microcontrôleur communique avec le PC par l'intermédiaire d'une liaison série RS232.

Le PC utilisé possède 2 processeurs Intel Pentium 4 cadencé à une fréquence 3GHz. Le système d'exploitation est Windows XP.

Ce système de commande a pour fonction le contrôle des actionneurs du robot manipulateur, la génération de trajectoire et ce grâce au modèle géométrique inverse du robot à deux articulations rotoides. En supposant que les tiges et les roulements sont infiniment rigides, la position de l'outil sera connue par les matrices de transformation du robot.

I.10 Conclusion :

Nous avons présenté dans ce chapitre le robot utilisé : robot SCARA dont la troisième rotation est éliminée ($n=3$) ce qui l'amène à travailler dans le plan (x, y) ($m=2$). Nous avons présenté la commande et les principaux composants qui la constituent. Nous avons défini des tâches à suivre pour ce robot. Ces tâches sont la poursuite d'une trajectoire circulaire en position et en orientation dans l'espace opérationnel ainsi que l'atteinte de positions spécifiées par l'utilisateur toujours dans l'espace opérationnel. Les caractéristiques des actionneurs pas à pas utilisés pour effectuer ces tâches sont données dans le chapitre qui suit.

CHAPITRE II : SYSTEME D'ENTRAINEMENT DU ROBOT

Chapitre II

Système d'Entraînement du Robot

II.1 Introduction

Le moteur pas à pas constitue un convertisseur électromécanique. Son principe de fonctionnement consiste en une suite de déplacements discrets d'une amplitude uniforme plutôt qu'en un mouvement continu. Le moteur exécute un nombre de pas correspondant au nombre d'impulsions reçues à l'entrée lors de son fonctionnement. Chaque impulsion provoque un pas de déplacement de l'axe du moteur et l'accroche magnétiquement au point précis auquel celui-ci doit se mettre. Ce pas de déplacement, dit très souvent, le pas angulaire est par définition l'angle minimum entre deux positions voisines et correspond également au pas du moteur.

Ces moteurs constituent ainsi une alternative intéressante pour la résolution du problème de positionnement auquel nous sommes confrontés. En effet afin de motoriser un bras manipulateur à deux articulations rotoides devant effectuer des mouvements incrémentaux de rotation dans un plan, un cahier des charges a été dressé. L'objectif de cette application est de développer un système qui permet de générer un pas élémentaire de rotation de 15° correspondant à la distance séparant deux pins de circuits intégrés, ainsi qu'une force de démarrage d'environ 20 Newton sur chaque pas de déplacement. Ce système est prévu pour fonctionner à faible vitesse et sur une course utile d'environ 100 millimètres.

II.2 Domaine d'utilisation :

Un nombre croissant d'applications électronique et électrotechnique requièrent l'usage des moteurs pas à pas. Dans le domaine de l'automobile ainsi que dans le monde industriel, en passant par les environnements télécoms et médical, il ne manque pas d'exemples où l'adoption de tels moteurs se généralise. Parmi la multitude d'applications concernées, nous citerons : le positionnement automatique des phares de voiture, les robots dans les usines (positionnement des axes, vitesse variable des robots), les systèmes de surveillance vidéo, informatique (déplacement d'une tête de lecture d'un lecteur, déplacement de la tête d'impression, avance du papier pour les imprimantes), commande des axes pour les tables traçantes [4]. Le positionnement des antennes paraboliques...etc.

II.3 Types des moteurs pas à pas.

Les performances d'un système d'entraînement ou de positionnement dépendent simultanément du type de moteur, de la structure choisie, du système d'alimentation utilisé et de la stratégie de commande implémentée [5][6].

Les moteurs pas à pas peuvent être classés en fonction du phénomène physique qui est à l'origine de leurs mouvements. On distingue principalement, deux catégories de moteurs pas à pas : les moteurs à réluctance variable et les moteurs polarisés [7] [5].

II.3.1 Moteurs pas à pas à réluctance variable.

La rotation d'un moteur à réluctance variable est engendrée par la réaction entre un champ magnétique statorique et un rotor saillant, ce qui conduit à une disposition alignée de la partie saillante avec le pôle créé par le champ magnétique. Dans ces conditions, le flux est maximum et la réluctance du circuit magnétique est minimum [8]. Ce type de moteur est caractérisé par une structure dentée au niveau du rotor et du stator. Le nombre de bobines dans le stator et le type de connexion déterminent le nombre de phases du moteur. La fig.II.1 représente un moteur pas à pas réluctant comportant six plots (1,1', 2,2', 3,3') et quatre dents au rotor.

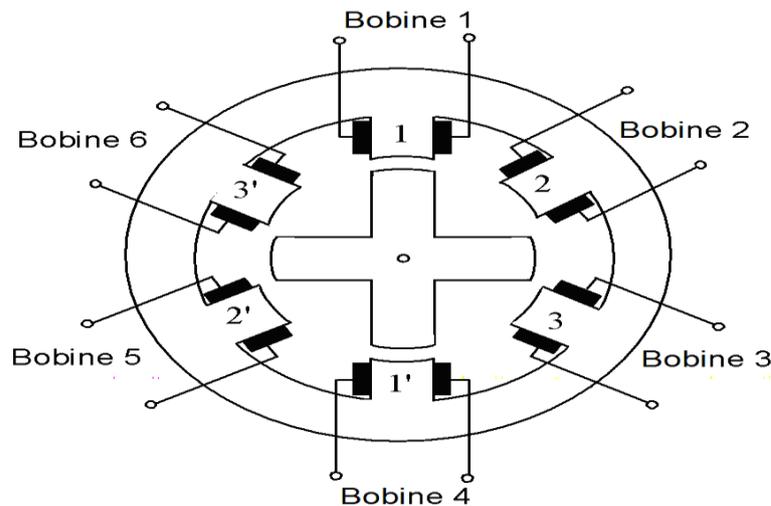


Fig.II.1: structure classique d'un moteur à réluctance variable rotatif à double saillance

Chaque couple de deux bobines statoriques diamétralement opposées et connectées en série, constitue une des trois phases de la machine considérée. L'alimentation d'une phase crée un couple permettant le déplacement du rotor vers une position d'équilibre qu'il garde tant que l'alimentation est maintenue. Cet état d'équilibre correspond à une position alignée entre les dents statoriques porteuses de la phase alimentée et les dents rotoriques. Chaque nouvelle séquence d'alimentation établit un nouvel équilibre. Le moteur se déplace donc avec un pas angulaire de 60° .

II.3.2 Moteurs polarisés

❖ Les moteurs à aimants permanents

Le fonctionnement des moteurs à aimants permanents est assuré par l'action d'un champ électromagnétique statorique sur un rotor composé d'aimants permanents [5] [9].

Le schéma (fig.II.2) illustre la structure élémentaire d'une machine à aimant permanent. Les enroulements sont bobinés sur des plots régulièrement répartis dans le stator.

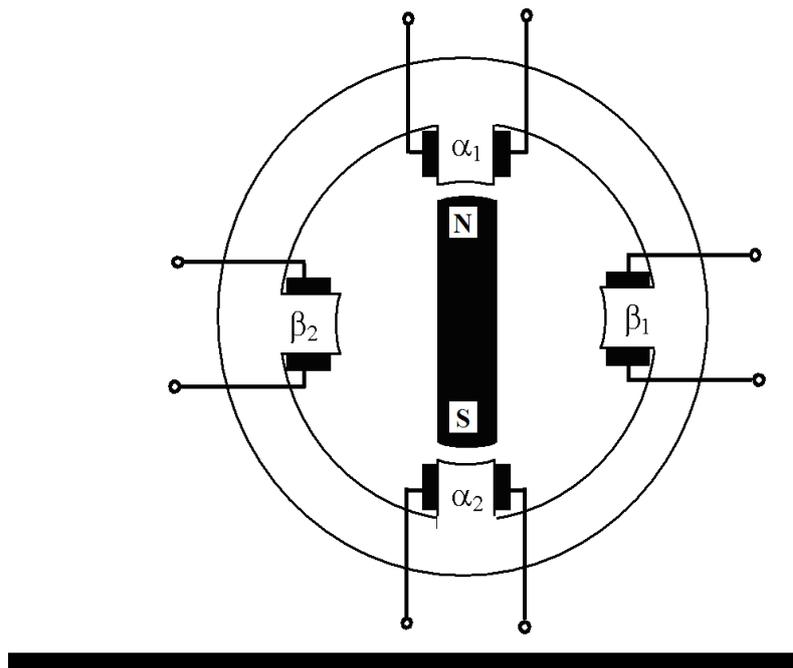


Fig.II.2: structure d'un moteur à aimant permanent rotatif

Les moteurs pas à pas à aimants permanents sont souvent alimentés en bipolaire ainsi les enroulements α_1 et α_2 (fig.1.3) sont mis en série pour ne former qu'une phase α ainsi que pour les enroulements β_1 et β_2 pour former la deuxième phase β décalée d'un angle de $\pi/2$. L'alimentation successive des phases (α) et (β) par des courants positifs et des courants négatifs peut donc provoquer un mouvement de rotation entre ces différentes positions d'équilibre.

❖ Les moteurs hybrides

Les moteurs pas à pas hybrides sont généralement constitués d'un rotor denté muni d'aimants permanents. Le rotor possède généralement deux disques polaires décalés d'un angle électrique de π . La fig.II.3 présente la structure d'un moteur pas à pas hybride à huit enroulements.

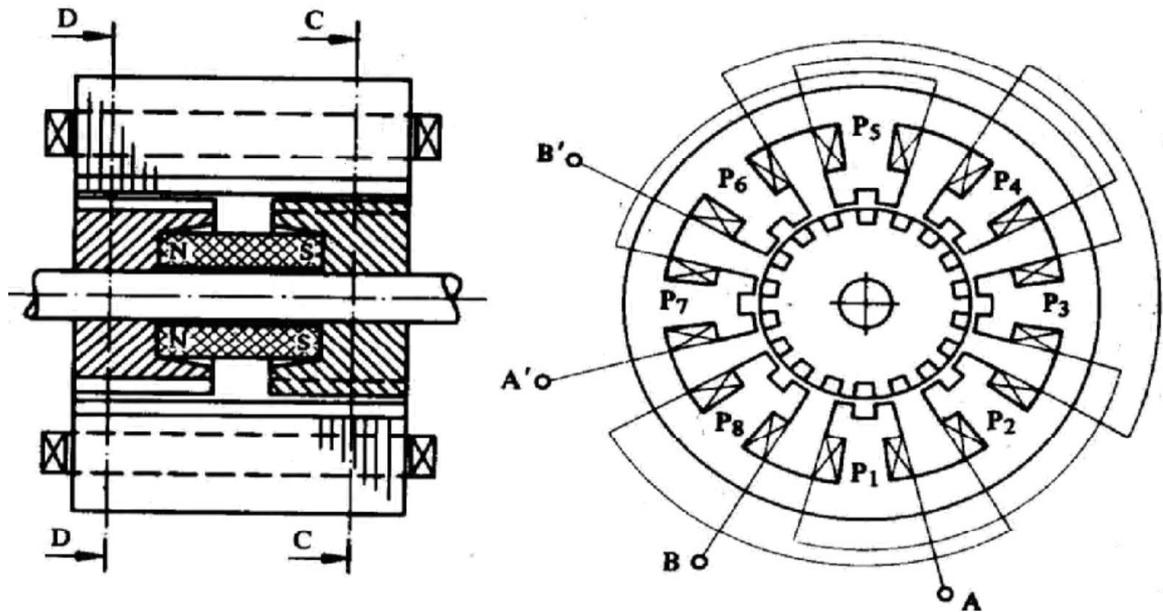


Fig.II.3 : structure d'un moteur hybride

Ce type de moteur présente à la fois les avantages du moteur à aimant permanent (possédant un couple élevé) et du moteur à réluctance variable (l'obtention d'un nombre important de pas par tour). Toutefois, l'inertie d'un tel rotor ainsi que les pertes fer sont relativement importantes et pénalisent donc cette structure [5] [10].

❖ Moteurs pas à pas linéaires :

Ils ont la même structure que les moteurs rotatifs c'est-à-dire ils se composent d'un stator portant les bobinages et d'une partie mobile (rotor) se déplaçant cette fois-ci linéairement. Cette partie mobile peut être réalisée à partir d'une pièce ferromagnétique dentée d'où la structure est dite à réluctance ou passive ; ou alors la partie mobile est munie d'aimants permanents et dans ce cas la structure est dite polarisée ou active.

II.4. Comparaison des moteurs pas à pas

• Moteurs à aimant permanent

- Nombre de pas par tour faible (2 à 48), donc faible fréquence de travail dû à la difficulté de loger les aimants du rotor.
- rendement élevé.
- couple moteur élevé, du à la puissance des pôles aimantés, le couple est proportionnel au courant.
- le sens de rotation dépend de l'ordre d'alimentation et du sens du courant dans les bobines.

• Moteurs à réluctance variable

- Nombre de pas par tour important, compris entre (12 à 72).
- permet d'atteindre des niveaux de fréquence élevés.

- le sens de rotation dépend de l'ordre d'alimentation des bobines (minimum 3 phases).
- construction assez facile.
- Rendement faible d'autant que sa puissance est faible.

- **Moteurs hybrides**

- Nombre de pas par tour plus élevé (24 à 400).
- Rendement bon en général.
- Couple moteur élevé du à l'alimentation du rotor, proportionnel au courant.
- Le sens de rotation dépend de l'ordre d'alimentation des bobines et du sens du courant.

II.5 Commande des moteurs pas à pas

Deux classes de déplacements peuvent être définies pour un moteur pas à pas : soit le moteur avance d'un pas entier, soit d'un fragment de pas selon un mode de commande adopté.

Dans cette partie, différentes stratégies de commande sont présentées : deux stratégies de commande en pas entier, dites commande en mode 1 et 2 ainsi que trois stratégies de commandes en pas fragmentés, dites commande en modes 3, 4 et 5 [7].

II.3.1 Fonctionnement à pas entiers

❖ Commande en mode 1

La commutation phase par phase de la structure quadriphasée de la fig.II.4 permet au mobile d'atteindre quatre positions d'équilibre sur une période de déplacement.

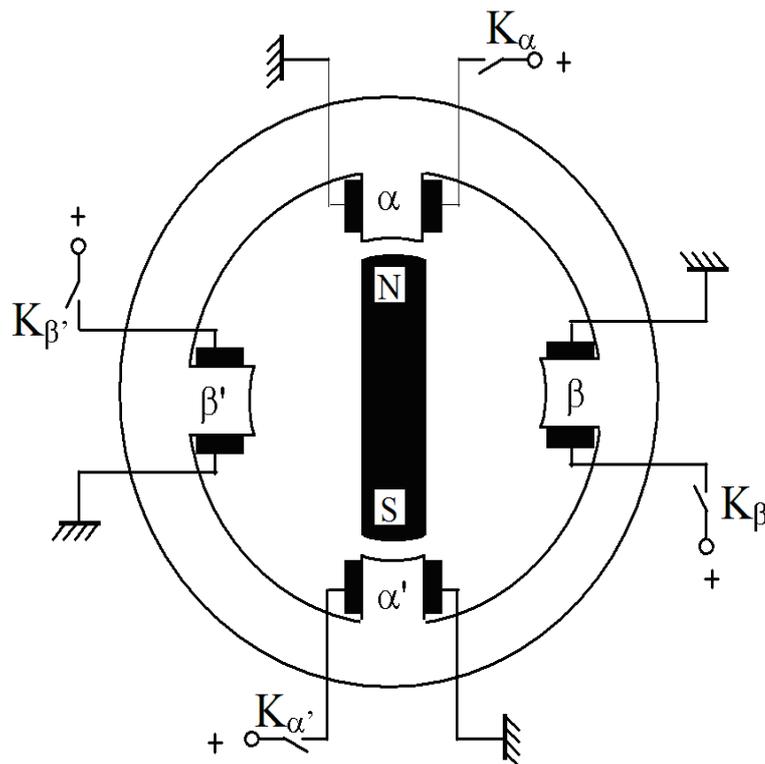


Fig.II.4 : structure quadriphasée à alimentation unipolaire

Les interrupteurs K_α , K_β , $K_{\alpha'}$, $K_{\beta'}$, de la fig.II.4 sont actionnés successivement au cyclogramme (fig.II.5.a). Ce qui conduit à quatre positions d'équilibre du mobile. La fig.II.5.b symbolise le mode de commande des différentes bobines et la position relative du mobile suivant l'alimentation.

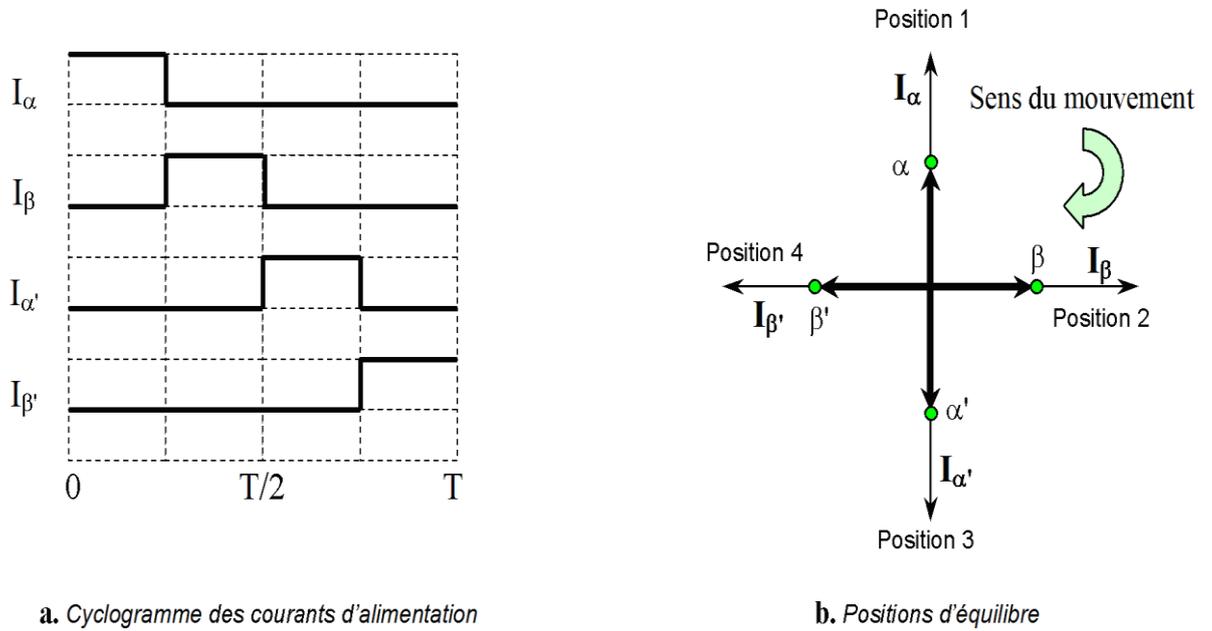


Fig.II.5 : principe d'alimentation en mode 1

Si les interrupteurs de la fig.II.4 sont actionnés dans l'ordre inverse K_α , $K_{\beta'}$, $K_{\alpha'}$, K_β le déplacement du champ et par conséquent celui de la partie mobile se fait dans le sens contraire du précédent. Ainsi, l'emplacement des positions d'équilibre est conservé, mais leur occurrence est inversée.

Les connexions sérielles de α et α' d'une part et de β et β' d'autre part permet d'obtenir un enroulement diphasé qu'il faut alimenter en bipolaire (fig.II.6).

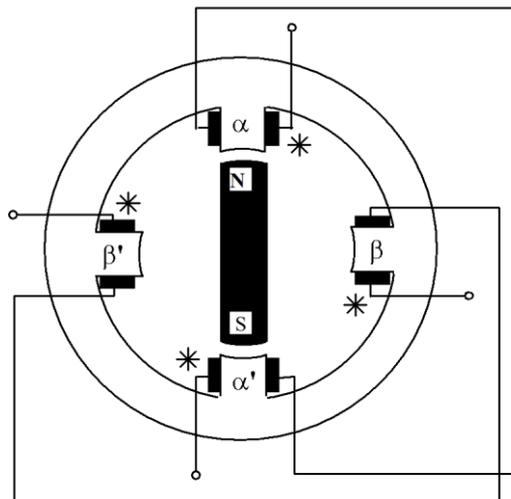


Fig.II.6 : principe de l'alimentation bipolaire en mode 1

L'excitation du stator ainsi connecté suivant le cyclogramme $(\alpha \alpha')$, $(\beta \beta')$, $(-\alpha \alpha')$, $(-\beta \beta')$ donne le même nombre de pas cycliques que l'alimentation unipolaire phase par phase, mais le couple est dans ce dernier cas plus important car deux bobines sont simultanément alimentées (fig.II.7).

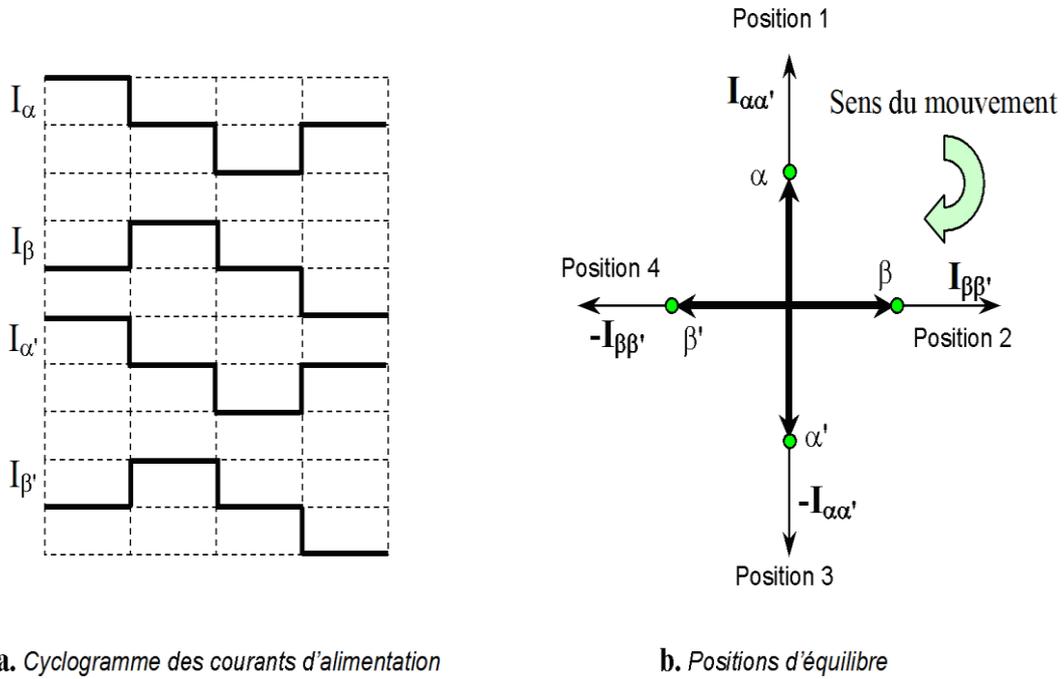


Fig.II.7 : principe de l'alimentation d'un bobinage diphasé en en mode 1

❖ **Commande en mode 2**

Dans la commande en mode 2, les doublets de phases successives de la machine (fig.2.8) sont alimentés simultanément suivant l'ordre (α, β) , (β, α') , (α', β') , (β', α) , conformément au cyclogramme (fig.II.8.a). Les positions d'équilibre correspondantes sont illustrées par la fig.II.8.b.

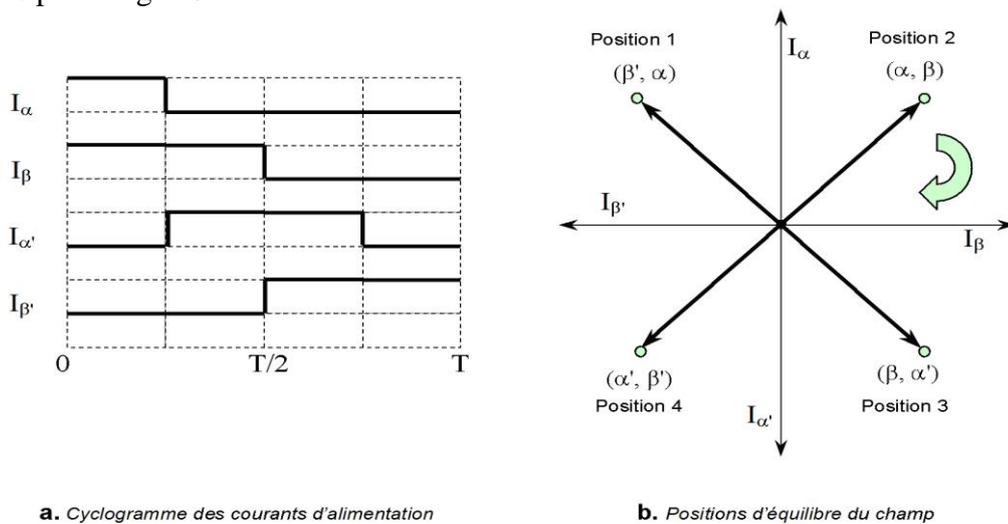


Fig.II.8 : principe d'alimentation en mode 2

L'avantage de ce mode de commande est qu'il optimise l'utilisation du cuivre de la machine et donc l'exploitation de sa puissance massique. Le couple développé dans ce cas est multiplié par $\sqrt{2}$ par rapport au couple développé par une commande en mode 1 unipolaire.

II.5.2 Fonctionnements à des pas réduits

❖ Commande en mode 3

Ce mode de fonctionnement est une combinaison en alternance des modes 1 et 2. Le nombre de positions d'équilibre est ainsi doublé et un déplacement en demi pas est généré. La fig.II.9 donne l'allure des courants de phase pour la machine à aimant permanent à alimentation unipolaire de la fig.II.4. Sur la fig.II.9.b, sont illustrées les huit positions d'équilibre correspondantes au fonctionnement en demi pas. L'inconvénient de ce mode de fonctionnement, est que le champ créé à chaque position n'a pas toujours la même amplitude. Il est $\sqrt{2}$ plus important à la position 2 comparativement à la position 1.

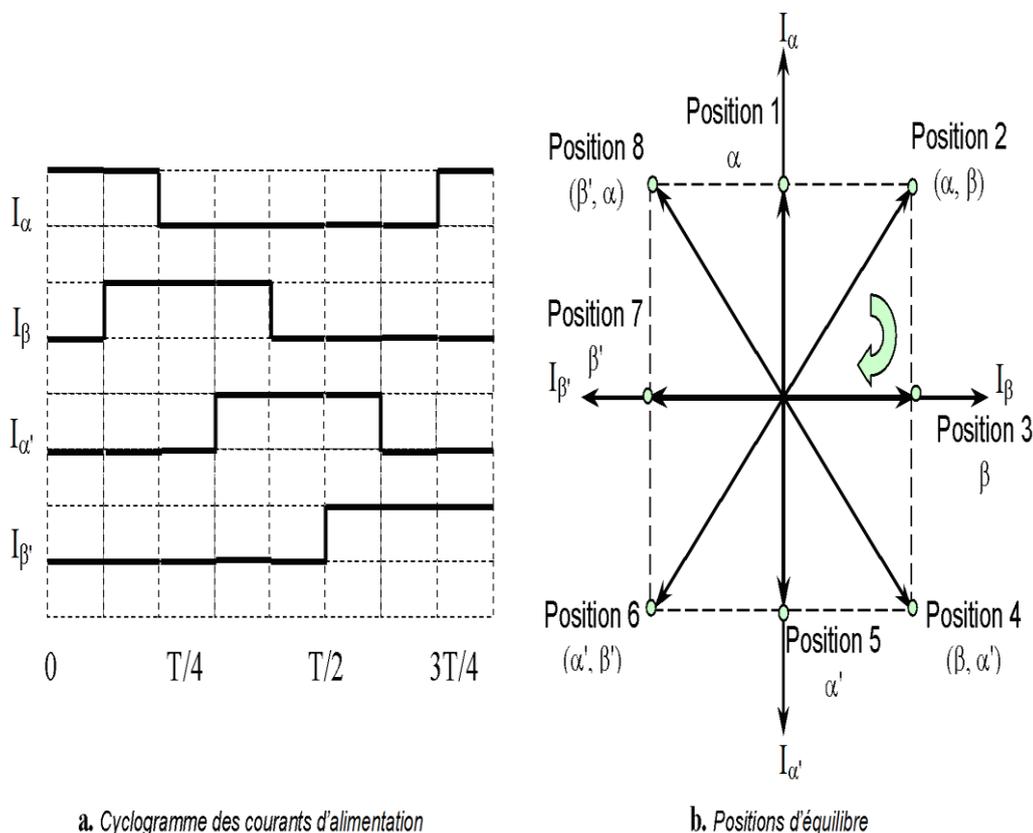


Fig.II.9: principe d'alimentation en mode 3

❖ Commande en mode 4.

La différence entre ce mode de fonctionnement et le mode 3, présenté précédemment, est que pour obtenir des vecteurs champs d'amplitudes identiques quelque soit la position, la consigne de courant est multipliée par un facteur $\sqrt{2}$ lorsqu'une seule phase est alimentée. Ce mode permet de conserver les pertes joule au cours du fonctionnement. Les allures des courants d'alimentation sont dans ce cas, données par le cyclogramme (fig.II.10.a) et les positions d'équilibre correspondantes (fig.II.10.b).

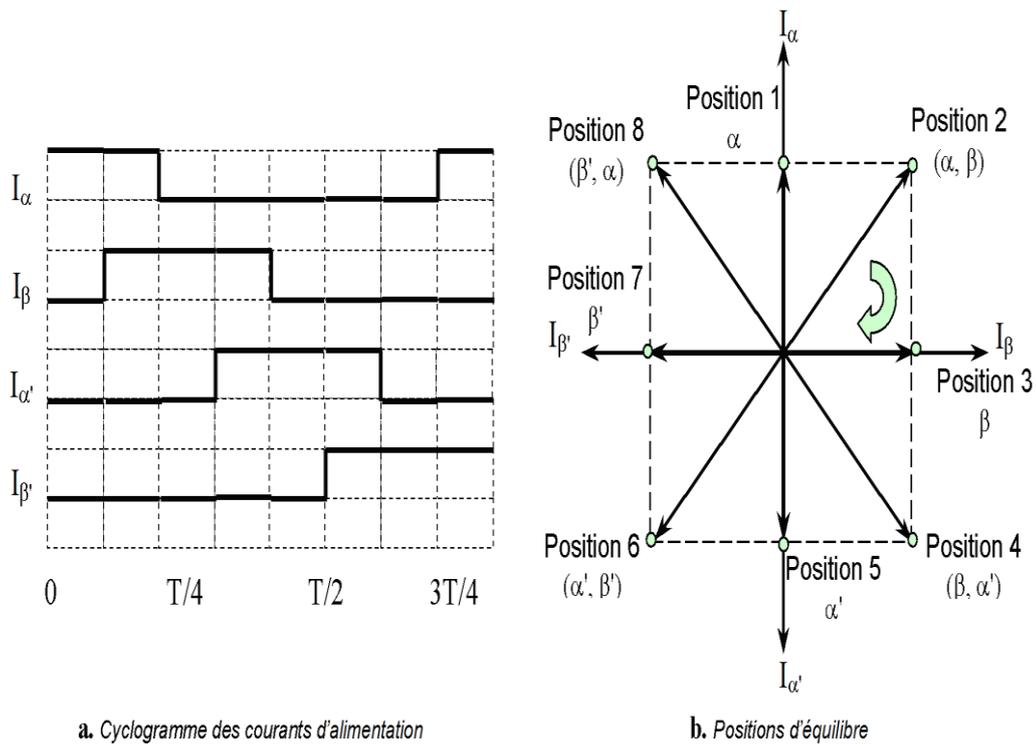


Fig.II.10 : principe d'alimentation en mode 4

❖ Commande en mode 5

Pour améliorer substantiellement la résolution angulaire d'un moteur pas à pas par action sur la commande, la technique de pilotage par micro pas, par variation de l'amplitude des courants peut être appliquée. Cette technique de fragmentation de pas consiste à alimenter simultanément deux phases de la machine par deux courants d'amplitudes différentes. Les amplitudes des courants d'alimentation varient séquentiellement en créneaux entraînant ainsi le rotor par fragments de pas [7] [11] [12].

Par ce mode d'excitation, la modulation des courants injectés dans les phases statoriques permet de positionner le rotor dans une position quelconque. Si les enroulements statoriques du moteur sont alimentés par des niveaux de courant à variations sinusoïdales (fig.II.11.a), alors les phases α et β doivent être alimentées simultanément par des courants $I_{\alpha\theta}$ et $I_{\beta\theta}$ et ce afin d'obtenir parmi les n positions celle repérée par θ (fig.II.11.b) [13].

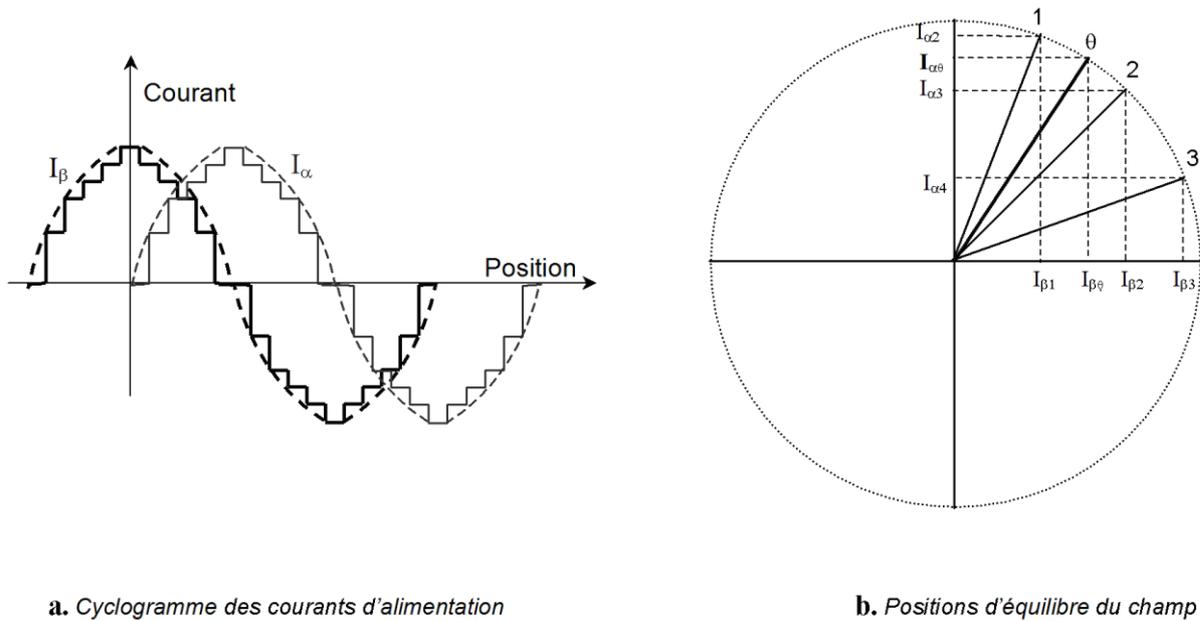


Fig.II.11: pilotage par modulation d'amplitude des courants.

II.6 Alimentation des moteurs pas à pas :

Les performances d'un entraînement à moteur pas à pas dépendent à la fois du moteur, de la structure du convertisseur d'alimentation et de la séquence d'excitation des phases [7]. Donc les moteurs pas à pas sont liés au concept de système contrairement aux performances d'autres types de moteurs, les performances d'un moteur pas à pas sont liées : commande, alimentation moteur –charge. Il n'est pas possible au niveau des performances de dissocier un élément d'un tel système.

II.6.1 Objectifs d'une bonne alimentation et classification :

Un enroulement de moteur pas à pas est un circuit électrique fortement inductif siège d'une force électromotrice variable ou présente parfois une inductance variant avec la position. Des éléments s'opposant à sa modification rapide du courant d'alimentation. Une bonne alimentation doit :

- ❖ Etablir rapidement un courant donné dans un circuit inductif ;
- ❖ Maintenir ce courant à sa valeur en dépit des fluctuations des paramètres du circuit, notamment ceux qui varient avec la vitesse ou la position.
- ❖ Interrompre rapidement ce courant
- ❖ Eviter l'apparition de courants parasites.

Selon les applications, les performances souhaitées et les moteurs utilisés, on rencontre différents types d'alimentation.

II.6.2 Types d'alimentation des moteurs pas à pas

L'alimentation peut être de type tension ou courant : l'alimentation tension est généralement utilisée pour les moteurs à faible puissance, et l'alimentation courant est nécessaire pour les moteurs de moyennes puissances.

Dans notre cas nous avons utilisé une alimentation tension du fait que nos moteurs sont des moteurs à aimants permanent.

II.6.3 Alimentation bidirectionnelle :

Pour pouvoir inverser le courant à partir d'une alimentation fixe unique (12v dans notre cas), on utilise un pont en H fig. II.12. Un premier sens est obtenu en saturant T1 et T4 et en bloquant T2 et T3. Le second sens correspond à T2 et T3 saturé et T1 et T4 bloqués. Dans chaque cas, le fonctionnement est analogue à celui de l'alimentation. Cette structure est disponible sous forme intégrée pour les domaines de courant employé.

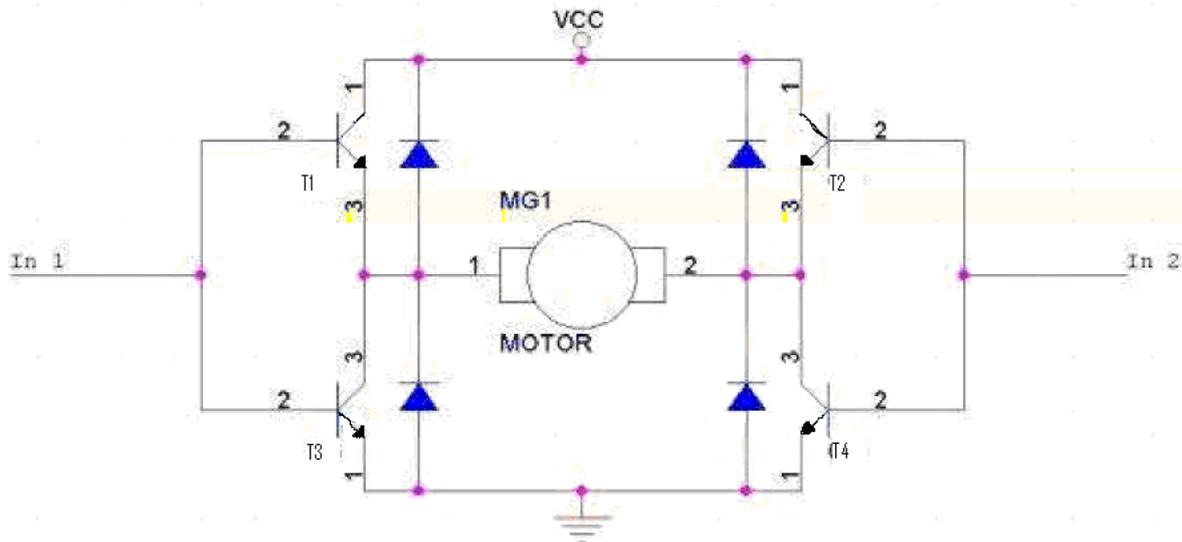


fig.II.12 : schéma d'un pont en H

L'avantage de ces circuits intégrés est d'être simple à réaliser tout en restant très efficace : Les circuits intégrés ponts en H sont des circuits de commande de puissance utilisés entre autre pour les moteurs pas à pas. Ils intègrent presque tous les composants nécessaires pour la réalisation d'un montage en pont. Les diodes de roue libre sont ajoutées pour protéger les transistors, de surtensions produites par le moteur.

Afin de pouvoir alimenter les phases du moteur de telle façon qu'il tourne dans un sens donné à une vitesse réglable, il faut un commutateur électronique permettant d'être commandé par un microcontrôleur. Nous nous sommes amenés à utiliser des ponts en H intégrés dans un driver L298. Le circuit de puissance L298 est formé de deux ponts en H de quatre transistors bipolaires chacun (sans les diodes de roue libre) avec une tension d'alimentation maximale de 50 volts capable de transmettre un courant atteignant les 3A. Chaque moteur a besoin d'un driver L298.

II.6.4 Constitution du L298 :

La fig.II.13 donne le diagramme bloc détaillant la structure interne du circuit intégré. Le L298 contient 6 entrées et 4 sorties en plus des trois broches d'alimentation. En effet, une tension V_{SS} sensée alimenté la partie logique du composant doit être reliée à une alimentation de 5V. La tension V_S est la tension avec laquelle sera alimenté le moteur. Le point GND constitue la masse du circuit intégré. Les trois sorties doivent être reliées aux deux bobines du moteur. En ce qui concerne les deux entrées Enable, chacune est destinée à valider le fonctionnement d'un des ponts en H qui lui est associé. Elles ont toutes les deux été reliées à la broche V_{SS} , ce qui implique que les deux ponts sont en permanence validés.

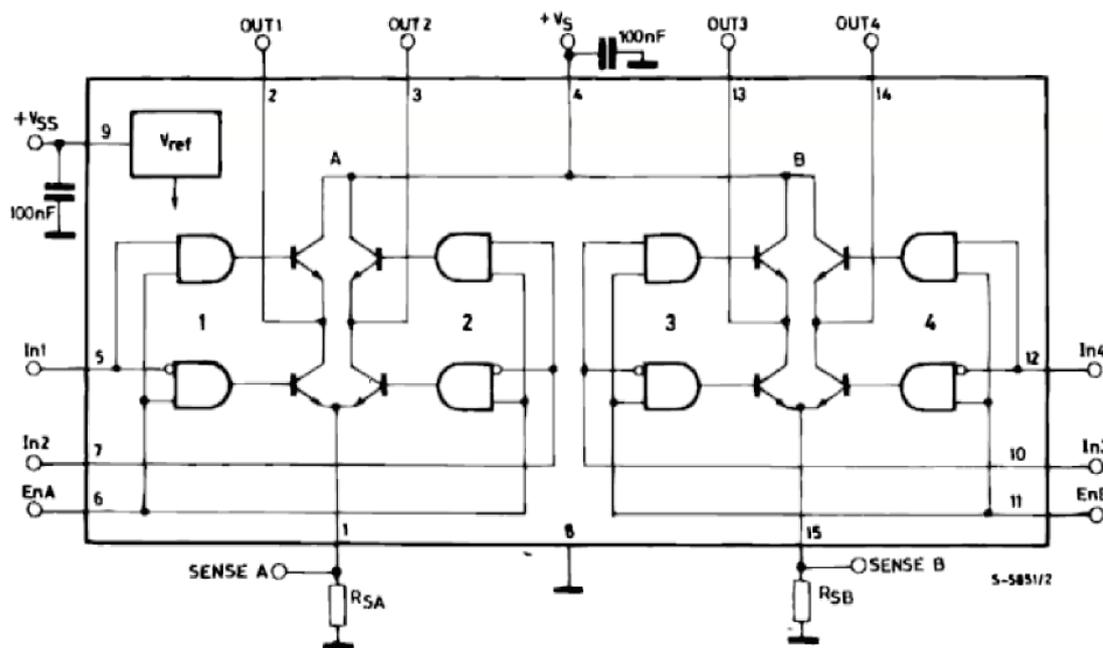


Fig.II.13: Diagramme bloc du L298

II.7 Conclusion

Dans ce chapitre, on a présenté les trois grandes familles de moteur pas à pas leurs caractéristiques, leurs principes de fonctionnement, ainsi que les différents modes de commande.

Cette étude a montré que pour générer un mouvement incrémental de translation, le moteur à aimant permanent représente une solution intéressante. En effet, cette solution se distingue de celle utilisant un moteur pas à pas à réluctance variable, par le couple important.

L'étage de puissance utilisé dans cette commande se présente sous la forme d'un double pont en H intégré qui est le L298.

Malgré toutes ces qualités en faveur des moteurs pas à pas. Il y a des inconvénients qui limitent leur utilisation dans certains domaines :

- La faible puissance.
- Le phénomène de résonance pour une fréquence donnée.

Après avoir présenté le système d'entraînement du robot le chapitre suivant sera consacré à l'étude de la partie de pilotage électronique.

CHAPITRE III : ELECTRONIQUE DE COMMANDE

Chapitre III

Electronique de Commande

III.1 Introduction

L'avènement des microcontrôleurs, qui associent au microprocesseur de base un programme intégré au circuit, ainsi que des périphériques et de la RAM, a permis de faire évoluer les montages vers plus de simplicité et de rapidité (les périphériques étant intégrés). Les microcontrôleurs sont aujourd'hui implantés dans la plupart des applications grand public ou professionnelles et il en existe plusieurs familles. La société Américaine MICROCHIP Technologie a mis au point dans les années 90 un microcontrôleur CMOS : le PIC, ce composant encore très utilisé à l'heure actuelle, est un compromis entre simplicité d'emploi, rapidité et prix de revient.

L'évolution technologique dans le domaine de semi-conducteur et microprocesseur permet de réaliser un système de commande à encombrement et prix réduits, avec des performances plus intéressantes que celles à logique câblée. En même temps, la programmabilité de ces circuits donne une grande souplesse des performances.

La commande en mouvement des robots manipulateurs actionnés par des moteurs pas à pas est assurée par ces circuits numériques où les lois de commande sont implémentées sous forme de programmes écrits selon un algorithme bien adapté.

Les principaux avantages de ces circuits numériques sont: la rapidité de développement, leur flexibilité et leur faible espace d'occupation.

III.2 Les composants électronique constituant la carte de commande :

Le schéma bloc général de la fig.III.1 traduit l'étude du système de commande du bras manipulateur et il est constitué :

- d'une unité de traitement des données qui est essentiellement le PIC 16F877 et le PC.
- des circuits d'interfaces pour les moteurs qui sont les ponts en H.
- d'une alimentation stabilisée de 5v et 12v.
- un adaptateur de tension MAX232.
- d'une unité de visualisation des données qui est l'afficheur LCD.

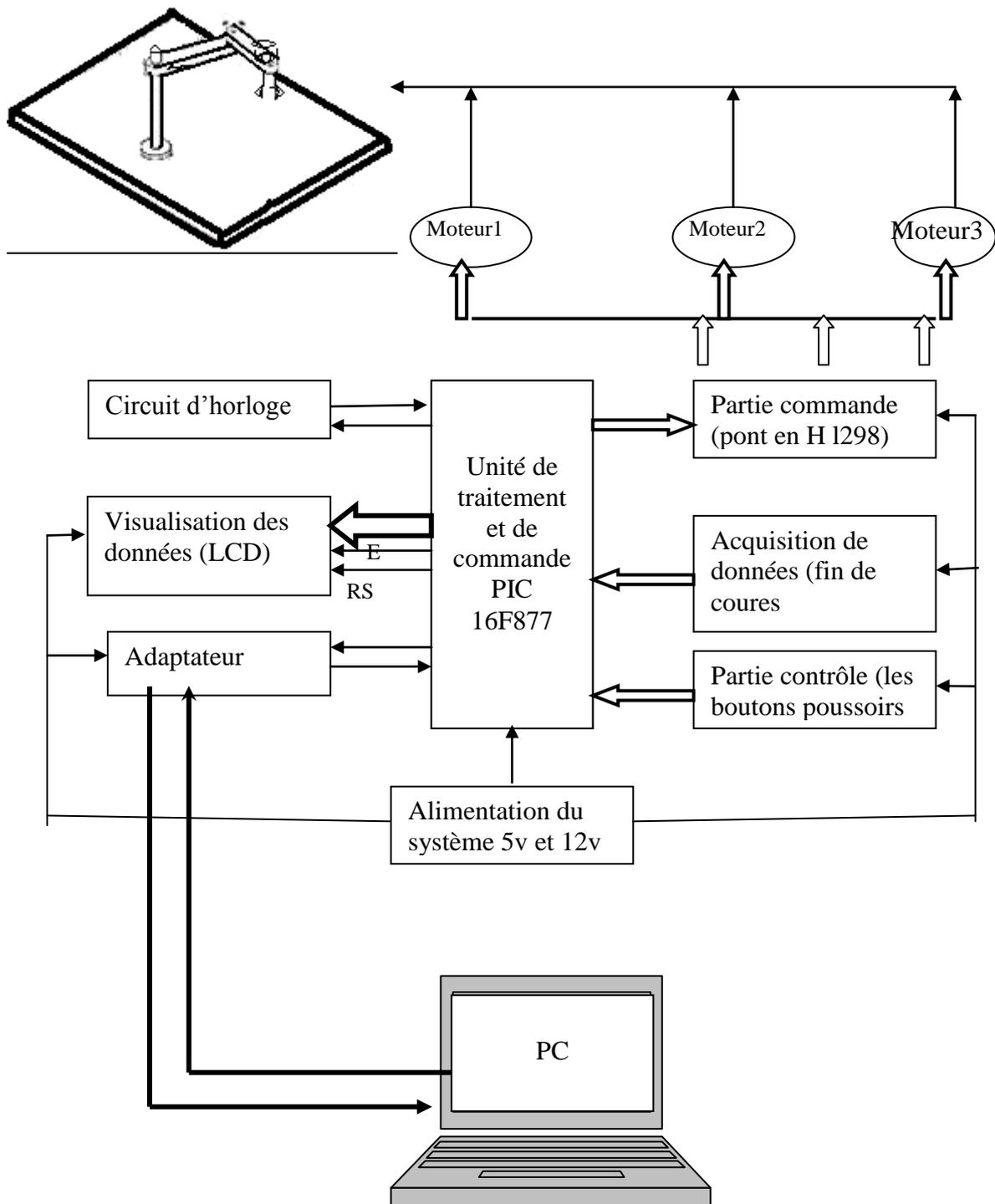


Fig.III.1 : schéma bloc général

III.3 Structure globale

Le rôle de l'électronique associée à un moteur pas à pas est d'assurer les commutations nécessaires pour obtenir un déplacement d'un nombre de pas donné ou une rotation à une vitesse fixée [3]. Deux dispositifs sont nécessaires (fig.III.2).

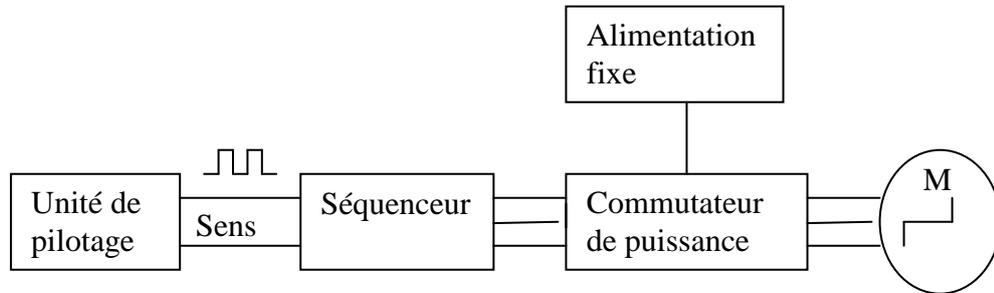


Fig.III.2 : organisation de l'alimentation d'un moteur pas à pas

- **L'unité de pilotage** : elle élabore deux signaux numériques. Une première sortie donne un bit qui permet de définir le sens de rotation. Une seconde sortie délivre des impulsions. Un nombre N d'impulsions correspond à un déplacement de N pas. Une répétition périodique des impulsions avec une fréquence f_p correspond à une rotation continue à une vitesse angulaire $\omega = f_p \alpha_p$. Cette fonction est souvent réalisée par un circuit programmable.
- **Le séquenceur** : il dirige les impulsions vers les différentes phases du moteur. Suivant le moteur utilisé (nombre de phases) et le mode de fonctionnement choisi (pas entier, demi pas...), Ce circuit délivre sur ces sorties des signaux logiques définissant les commutations à réaliser pour chaque phase.

III.3.1 Le microcontrôleur :

Etant donné la vitesse de calcul, les traitements arithmétiques et logique, le nombre important des entrées sorties, nous avons choisi parmi les microcontrôleurs de différents fabricants, le microcontrôleur PIC qui est mis au point par la firme MICROCHIP Technologie [microchip.com]. Le mot PIC est l'acronyme de Programmable Interrupt Controller en anglais, soit un Contrôleur d'Interruption Programmable.

Actuellement la famille des PIC est subdivisée en 3 grandes familles: La famille Base-Line, qui utilise des mots d'instructions de 12 bits, la famille Mid-Range, qui utilise des mots de 14 bits (dont fait partie le 16F877), et la famille High-End, qui utilise des mots de 16 bits. Tous les PIC Mid-Range ont un jeu de 35 instructions, stockent chaque instruction dans un seul mot de programme, et exécutent chaque instruction (sauf les sauts) en 1 cycle machine. On atteint donc de très grandes vitesses, et les instructions sont en plus, très rapidement assimilées. L'exécution en un seul cycle est typique des composants RISC.

III.3.1.1 Identification des PIC :

On peut identifier un PIC à travers sa référence, les deux premiers chiffres indiquent la famille du PIC par exemple: 16 représente la famille Mid-Range, 18 la famille High-End. Juste après, vient parfois la lettre L qui indique que le PIC peut fonctionner avec une plage de tension beaucoup plus tolérante. Le type de mémoire programme peut être indiqué grâce à la lettre C si elle est de type EPROM, CR si elle est de type ROM, ou bien F si la mémoire est de type FLASH. En fin, les derniers chiffres représentent le PIC lui-même.

III.3.1.2 Etude du microcontrôleur 16F877 :

Pour notre cas, le microcontrôleur choisi est le PIC 16F877A-20 de microchip, donc il appartient à la famille Mid-Range (16) : sa mémoire est de type Flash (F), sa référence (877A), sa fréquence d'horloge est de (20MHz). L'horloge fournie au PIC est pré-divisée par 4 au niveau de celle-ci. C'est cette base de temps qui donne le temps d'un cycle. Nous avons utilisé un quartz de 20MHz, donc 5000000 de cycles/seconde, or, comme le PIC exécute pratiquement une instruction par cycle, hormis les sauts, cela donne une puissance de l'ordre de 5MIPS (5 Million d'Instructions Par Seconde). C'est une vitesse de traitement très satisfaisante.

Les broches du PIC et leurs différentes fonctions sont montrées par la fig.III.2.

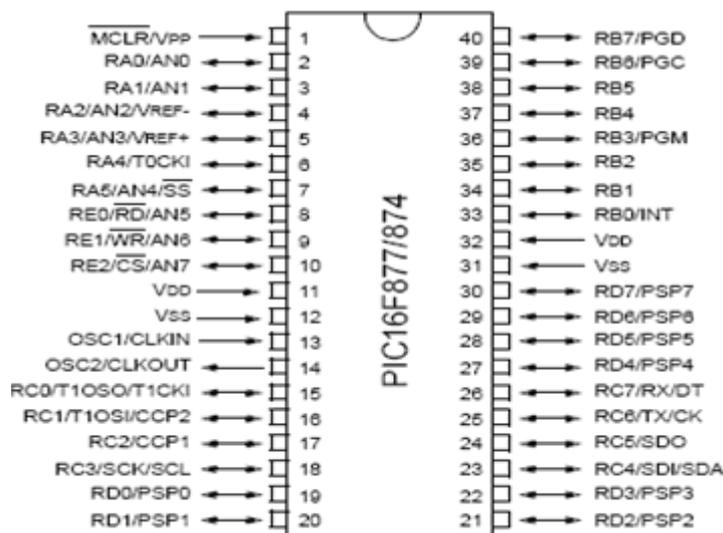


Fig.III.3 : Les broches du PIC et leurs différentes fonctions

III.3.1.3 Caractéristiques principales du 16F877 :

Les principales caractéristiques du 16F877 sont regroupées dans le (tab.III.1).

Key Features PIC micro™ Mid-Range Reference Manual (DS33023)	PIC16F877
Fréquence System	DC - 20 MHz
RESETS (et Délais)	POR, BOR (PWRT, OST)
Mémoire Programme de type FLASH (mots de 14 bits)	8K
Mémoire de données	368 Octets
Mémoire de données EEPROM	256 Octets
Sources d'interruptions	14
Les Ports d'Entrées/Sorties	Ports A (6 bits), B (8 bits), C (8 bits), D (8 bits), E (3 bits)
Les timers	3
Les modules de captures et de comparaison avec PWM (modulation de largeur d'impulsions)	2
Communications séries	MSSP, USART
Communications parallèles	PSP
Convertisseur A/N de 10 bits	8 entrées multiplexées
Jeu d'instruction	35 instructions

Tab.III.1: les principales caractéristiques du PIC16F877

En plus de ces caractéristiques, le 16F877 est programmable et débbugable sur site ICSP (In Circuit Serial Programming) bootloader et ICD (In Circuit Debbuging). Il possède un chien de garde ainsi qu'une surveillance d'horloge OST et de tension d'alimentation BOR.

Le PIC 16F877 possède trois Timers : le Timer0, Timer1, Timer2.

Le Timer0 ainsi que le Timer2 sont sur 8 bits, tandis que le Timer1 est sur 16 bits. Ils peuvent jouer le rôle de compteur, de temporisateur ou les deux en même temps selon la configuration du Timer.

Pour les interfaces de communication, le 16F877 est doté d'interfaces série synchrone et asynchrone USART. Ainsi que les interfaces série synchrone SSP, SPI et I2C.

La tension de fonctionnement du PIC varie entre 2 à 5 volts et sa consommation est très faible :

- <2mA à 5V pour une fréquence de Quartz=4Mhz.
- 20µA à 3V pour une fréquence de Quartz à 32Khz.
- <1µA en mode sommeil.

Dans la figure qui suit, nous allons exposer d'une manière simple et explicite les éléments constituant le PIC 16F877A :

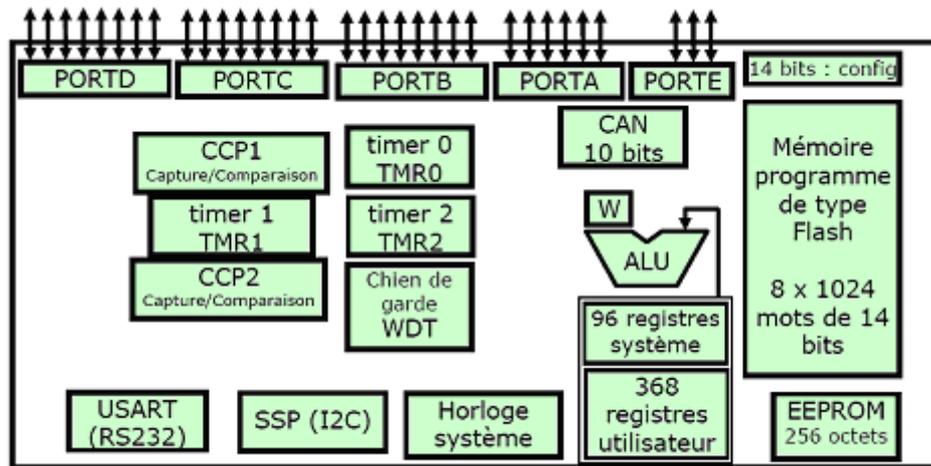


Fig.III.4 organisation interne du PIC 16F877

Les ressources utilisées dans notre travail sont les suivantes :

- Le Timer0.
- Les ports d'Entrée/Sortie.
- Les sources d'interruption du timer0 et du port série RS232.
- L'interface de communication série USART.
- L'écriture et la lecture en EEPROM du PIC.

III.3.1.4 Le port série du PIC 16F877 USART:

USART : Universal Synchronous Asynchronous Receiver Transmitter. C'est un module qui permet d'envoyer et de recevoir des données en mode série, soit de façon synchrone, soit asynchrone. Le module USART de notre microcontrôleur fonctionne sur 2 pins, à savoir RC6/TX/CK et RC7/RX/DT.

III.3.2 Le MAX232

Le composant qui permet l'adaptation entre les tensions TTL du côté du PIC et les tensions du port série du côté du PC est le circuit MAX232 appelé adaptateur RS232. C'est un double convertisseur de tension continue (DC/DC). Il donne à partir d'une tension d'alimentation +5V et quatre condensateurs 10 μ F, une tension \pm 12V. En effet, les niveaux logiques de la norme RS232 ne sont pas TTL (0, 5V). Elle affecte pour le niveau logique 1 une tension entre -3V et -25V, généralement -12V et pour le niveau logique 0 une tension entre +3V et +25V, généralement +12V, ceci assurera une transmission par câble à une distance d'au moins 2 mètres. Et c'est à ce circuit MAX232 de convertir les tensions -12V à +5V et +12V à 0V.

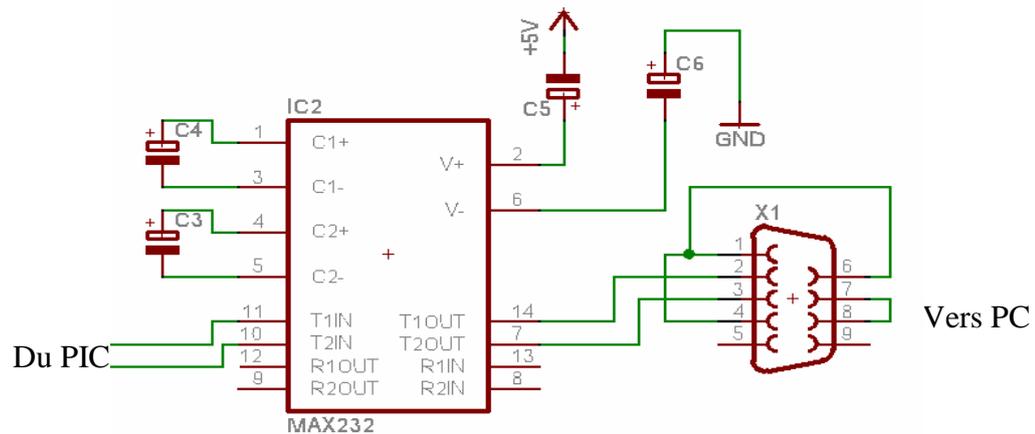


Fig.III.5 : L'interface de communication entre le PIC et le PC.

III.3.3 Communication série RS232 :

Pour améliorer les performances de notre système et faciliter son utilisation, nous avons chargé tous ses paramètres de fonctionnement via le port série RS232 en créant un logiciel contenant une interface graphique visuelle adéquate.

Nous écrivons un programme en Delphi qui peut envoyer des données du RS232 vers le PIC, puis ce dernier les affichera sur l'afficheur LCD. Le choix du port série n'est pas un simple hasard. Ce choix est dû au fait que la majorité des logiciels de développement le prennent en charge, d'autre part la majorité des microcontrôleurs possèdent un port série intégré. Les détails de ce système de liaison sont donnés comme suit :

A l'origine les ports série permettaient uniquement d'envoyer des données, mais pas d'en recevoir, mais les ordinateurs actuels sont équipés par des ports séries bidirectionnels donc on a besoin de deux fils pour communiquer.



Fig.III.6: Connecteur série male DB9

Pour notre application, la communication série se fait de façon asynchrone. Cela signifie qu'aucun signal de synchronisation (appelé horloge) n'est nécessaire: les données peuvent être envoyées à intervalles de temps arbitraires. En contrepartie, le périphérique doit être capable de distinguer les caractères (un caractère possède une longueur de 8 bits) parmi la suite de bits qui lui est envoyée. C'est la raison pour laquelle dans ce type de transmission, chaque caractère est précédé d'un bit de début (appelé bit START) et d'un bit de fin (bit STOP). Un ordinateur personnel possède généralement deux ports séries (certains de ces ports possèdent des connecteurs DB9, d'autres des connecteurs DB25).

III.3.3.1 Brochage du port série:

Signal	In/Out	PIN
Carrier Detect	In	1
Receive Data	In	2
Transmit Data	Out	3
Data Terminal Ready	Out	4
Signal Ground	-	5
Data Set Ready	In	6
Request to Send	Out	7
Clear to Send	In	8
Ring Indicator	In	9

Tab.III.2: brochage du port série

Les deux broches principales sont les broches de transmission des données 2 et 3, la broche d'émission d'un équipement est reliée à la broche de réception de l'autre (câble croisé). Les autres broches sont utilisées pour les protocoles de transmission.

III.3.3.2 Brochage du PIC 16F877 :

Comme le montre le schéma (fig.III.6), c'est un boîtier PDIP de type DIL (Dual In Line) :

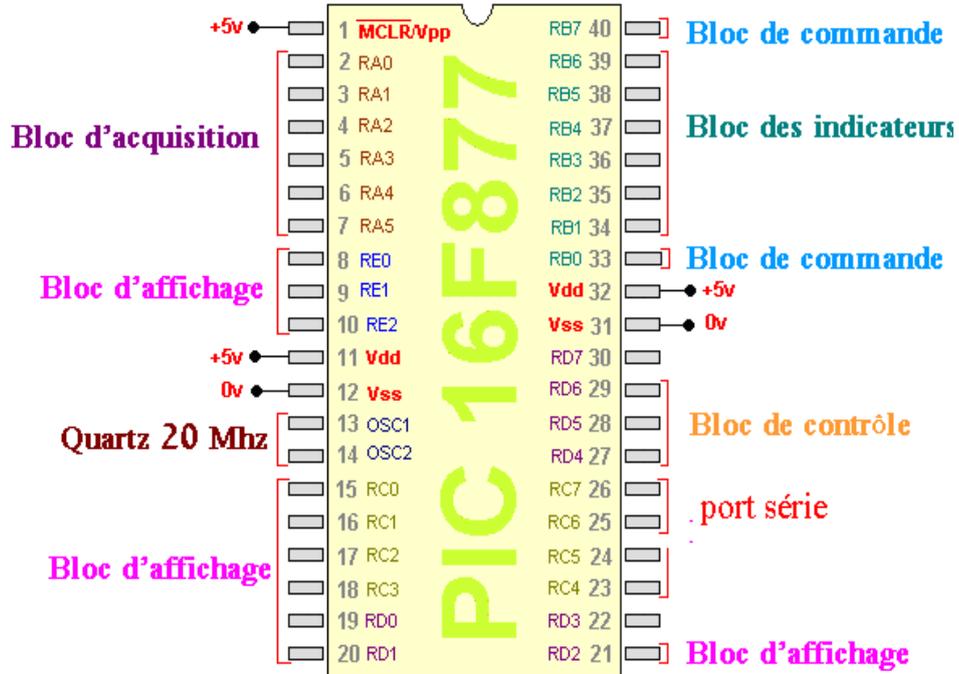


Fig.III.7 : Brochage du PIC

Les pins utilisées du PIC

Les pins	Fonction	E/S
RA0-RA3	Lié aux pin B4-B7 de l'afficheur	S
RA4, RA5, RE2	Navigation des boutons poussoirs	E
RB0-RB7	Liés aux capteurs de fin de course	E
RC0-RC3	Commande du moteur 1	S
RC6	TX : transmission	S
RC7	RX : réception	E
RD0-RD3	Commande du moteur 2	S
RE0-RE1	Commande de l'afficheur LCD	S

Tab.III.3: Les pins utilisées du PIC16F877.

III.3.4 Bloc de visualisation (l'afficheur LCD)

Afin d'élargir la fonctionnalité de notre projet nous avons rajouté un bloc de visualisation qui nous permet de communiquer facilement avec le microcontrôleur. Le bloc d'affichage permet de visualiser les données et à afficher l'ensemble des résultats et des commandes transmises au microcontrôleur. Ces résultats seront par la suite gérés et traités par l'unité de traitement. Ce bloc d'affichage est composé essentiellement d'un afficheur LCD à 2 lignes 16 caractères, son brochage est donné par la fig.III.7:

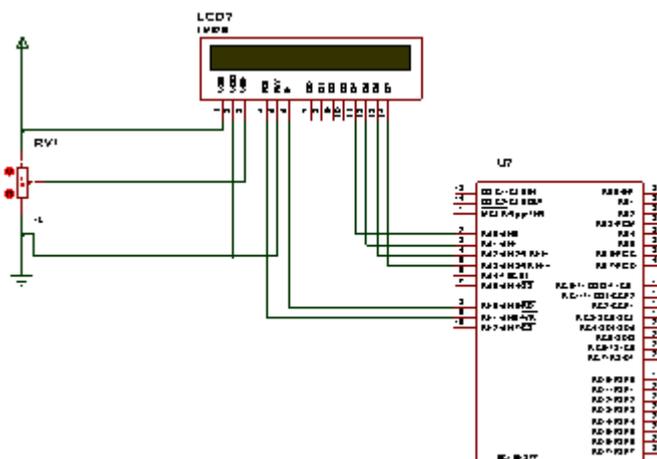


Fig.III.8 : Brochage de l'afficheur

Le type d'afficheur que nous avons utilisé est l'afficheur LCD (Liquid Crystal Display) deux lignes 16 caractères. Ces types d'afficheurs sont dotés d'une certaine intelligence grâce au microcontrôleur placé sur l'afficheur de la firme GDM1602A. On assure la réception des données, leur décodage et le pilotage des matrices de cristaux liquides. Les broches de l'afficheur sont au nombre de 14: 2 pour l'alimentation, 4 pour le contrôle et 8 constituant le bus de données. L'utilisation de l'interface d'affichage nous aide à visualiser les données et faciliter leur programmation.

III.3.4.1 Les pins de contrôle

Trois pins permettent le contrôle et la gestion de l'afficheur LCD, leurs fonctions sont données comme suit :

- Pin 3 : Vee pour l'ajustement du contraste relié à +5V par une résistance variable de valeur 330 Ω . Pour améliorer le degré d'affichage en matière d'obscurité nous avons fait appel à une option existante dans l'afficheur LCD qui est le rétro éclairage. Pour intervenir sur la qualité d'affichage ou le contraste une résistance variable d'une valeur de 680 Ω fait l'affaire.
- pin 4 : RS (Register Select) : permet de sélectionner le mode de fonctionnement de l'afficheur (réception de caractère ou commande) RS =1 pour la réception d'un caractère

RS=0 pour la réception d'une commande.

-pin 5: R/W (read / write) R/W=0 pour écrire, R/W +=1 pour la lecture.

-pin 6 : E (Enable), pour valider la donnée par un front descendant.

Les données peuvent être envoyées de deux manières :

Soit 8 bits à la fois, ce qui nécessite le câblage de tout un bus de 8 lignes, ou bien par paquet de 4 bits en utilisant les bits b4 à b7 de l'afficheur. C'est ce mode que nous avons choisi vu le gain de 4 pins du microcontrôleur.

La sélection du mode se fait en envoyant le mot de commande approprié lors de l'initialisation

III.3.5 La partie contrôle :

Le circuit de contrôle sert à gérer la partie commande à travers des boutons poussoirs et trois résistances de pull up qui le constituent.

Il permet de régler la vitesse des moteurs ainsi leurs sens de rotation et les coordonnées x et y

III.3.6 Bloc d'acquisition

C'est un circuit qui joue le rôle de capteurs. Il est constitué de quatre fins de course. Ces capteurs sont connectés au PIC à travers une barrette.

III.3.7 L'alimentation du circuit :

L'alimentation des circuits logiques a été réalisée sur le circuit imprimé, car elle est toujours constante et égale à 5V. Contrairement à celle des moteurs qui peut être variable.

- **Alimentation de la carte :**

Pour éviter les modifications de l'alimentation des moteurs pas à pas, nous avons opté pour l'utilisation de l'alimentation variable qui est disponible au niveau du laboratoire d'essai. Et Pour la partie logique nous avons réalisé une alimentation stabilisée de 5V.

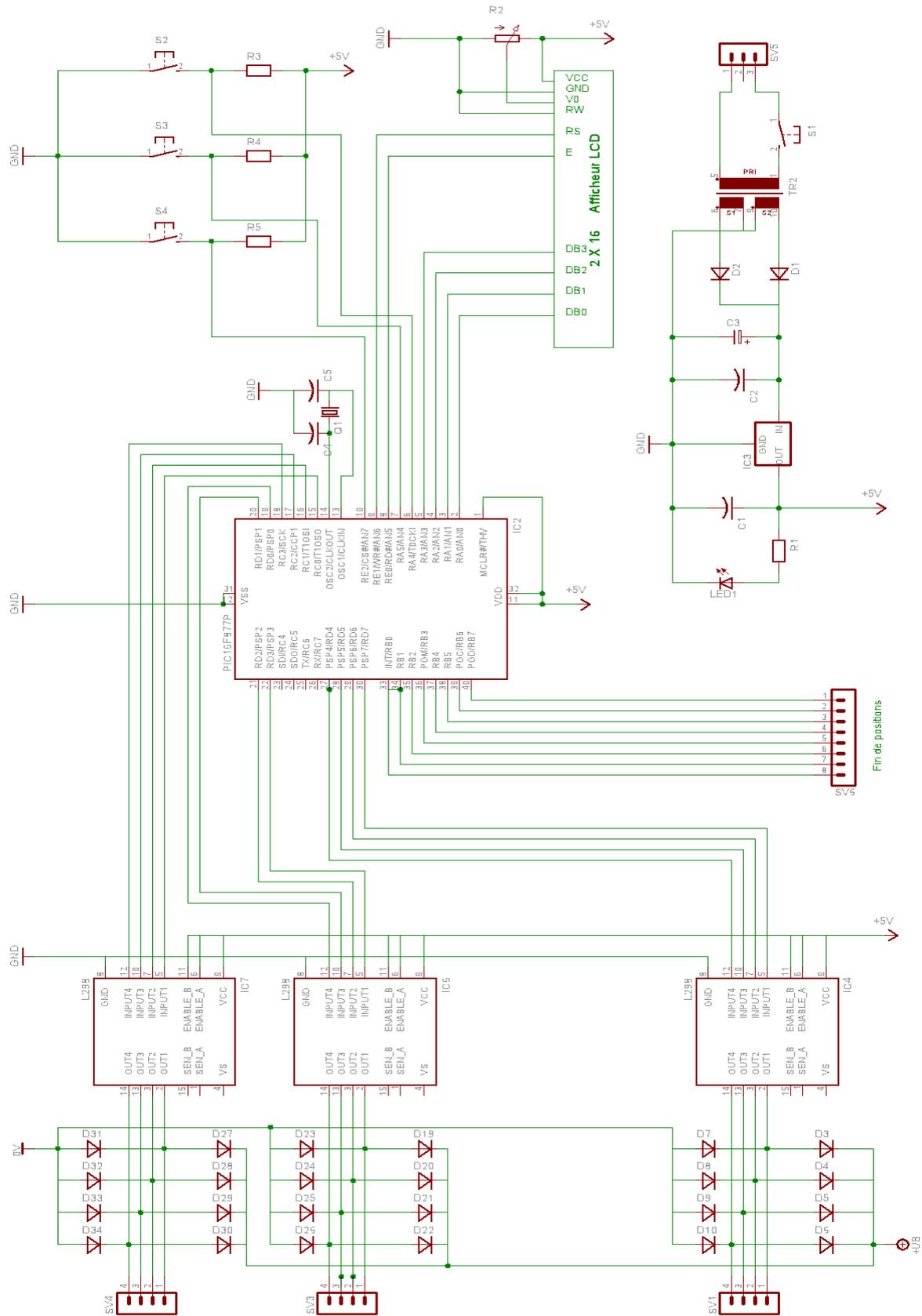


Fig.III.9: schéma électrique de la commande

III.4 Conclusion :

Ce chapitre nous a permis de présenter le microcontrôleur utilisé. Ce circuit intégré qui comprend à la fois l'unité centrale, le programme et les organes d'entrées /sorties. aussi nous avons détaillé les différents blocs constituant la carte à microcontrôleur que nous avons conçu et réalisé pour la commande des trois moteurs pas à pas qui sont destinés à la commande du bras manipulateur SCARA à deux articulations. Il s'agit à présent de présenter le programme informatique qui permet de piloter l'ensemble des circuits électroniques. Ce travail fera l'objet du chapitre suivant.

CHAPITRE IV : CONCEPTION LOGICIELLE DE LA COMMANDE

Chapitre IV

Conception Logicielle de la Commande

IV.1 Introduction

Après avoir vu les caractéristiques techniques et électroniques des composants qui constituent la carte de commande, nous nous dirigeons vers l'étude de la commande.

A toute conception matérielle à base d'un système à microcontrôleur, on doit systématiquement lui associer un logiciel qui aura à gérer l'ensemble de la structure. On peut classer le type de programme de l'interface en deux catégories:

- Le programme écrit en langage assembleur qui après sa compilation, sera chargé dans la mémoire du microcontrôleur.
- Le programme de communication avec le PC est écrit en langage évolué qui est le langage Delphi.

IV.2 Les outils de développements

Ils permettent la simulation, la compilation et le chargement des programmes dans le microcontrôleur.

IV.2.1 Le simulateur

Le logiciel MPLAB constitue un environnement presque complet. C'est dans l'interface de ce logiciel que l'on écrit les programmes du PIC. Ce dernier converti le programme assembleur (Fichier. asm) en un (fichier. hex), près à être chargé dans le microcontrôleur.

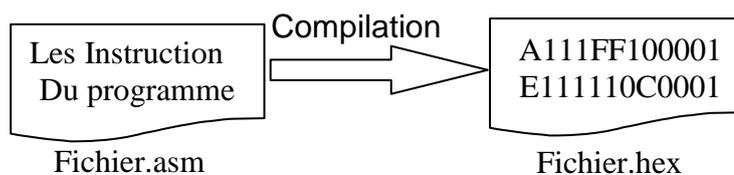


Fig.IV.1: compilation du programme source

L'aspect de l'interface de la dernière version du logiciel MPLAB est donné par la fig.IV.2. On présentera une méthode pratiquée pour la création d'un projet.

Une fois le projet établi, l'étape suivante consiste à créer le programme qui permet de répondre au cahier des charges. Le programme source que nous avons défini, est une suite d'ordres dit "Mnémoniques" qui font partie du jeu d'instructions du PIC utilisé.

Pour créer le programme source, il faut uniquement un éditeur de texte. En effet le programme source sera sauvegardé sous format ASCII. On sauvegarde le fichier en général avec l'extension .ASM (assembleur).

Une fois que le programme source est figé, l'étape suivante consistera à compiler le programme, c'est à dire à transformer le programme source en un programme dit "binaire". L'extension du fichier sera alors .HEX (hexadécimal).

Un logiciel spécifique permet de compiler le programme source. Le plus connu "MPLAB" de chez MICROCHIP est téléchargeable à l'adresse suivante: www.microchip.com.

Une fois compilé, le fichier binaire sera transféré via un logiciel approprié (Le plus connu MPLAB) vers la mémoire de programme du PIC.

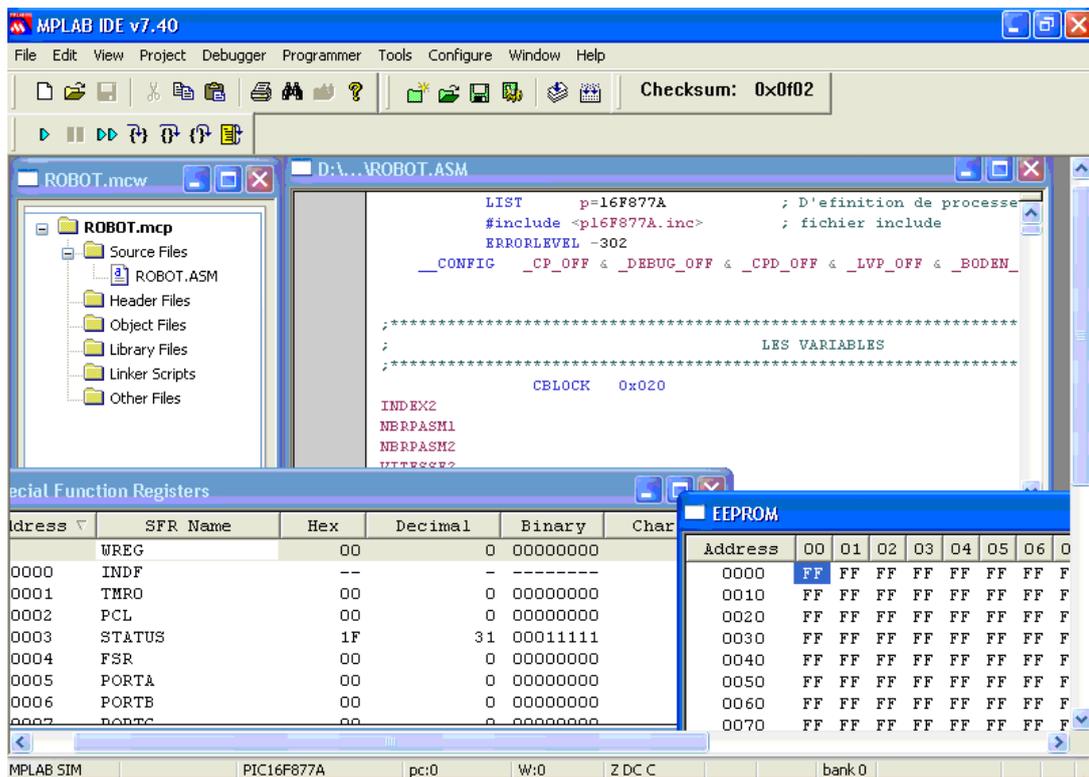


Fig.IV.2: Le simulateur MPLAB7.4

IV.2.2 Le programme à implémenter

Le programme implémenté dans le PIC doit réaliser les fonctions électroniques de calculs d'interfaçage et de communication avec l'ordinateur afin de transférer les données et les traiter ultérieurement. Les différentes fonctions implémentées dans ce programme sont illustrées par des organigrammes.

En plus de la réception des ordres venant des boutons poussoirs, le programme contient les fonctions d'affichage qui sont :

- Gestion d'un afficheur LCD.
- Gestion du port série, Envoi des caractères et des commandes au PC.
- Gestion des interruptions.

IV.3 Fonctionnement général du système de commande du bras manipulateur :

Le programme gérant le fonctionnement du bras est composé de :

1) Le programme principal dont le rôle est l'attente d'un appui sur le BP et la surveillance d'un arrivage de données à partir du port série.

2) Le programme d'interruption du Timer0, dont le rôle est la gestion de rotation des moteurs (vitesse, MARCHE /ARRET, AVANT/ARRIERE). La fig.4.5 montre l'organigramme du fonctionnement général du programme.

IV.3.1 Déroulement de l'organigramme :

Une fois la carte électronique de commande allumée le microcontrôleur démarre le programme interne résidant dans sa mémoire programme, dont le rôle est l'attente des ordres venant soit du port série soit des boutons poussoirs.

Au démarrage, après initialisation des ports d'entrées sorties, du port série ainsi que de l'afficheur LCD le programme du PIC entre dans une boucle d'attente, deux événements peuvent le faire sortir de cette attente, qui sont :

Si le bouton START est appuyé, les moteurs se mettent en marche en prenant comme repère de position les dernières positions atteintes au démarrage les valeurs par défaut. Ces moteurs continueront à tourner tant qu'ils n'ont pas reçu un ordre de s'arrêter venant du bouton STOP ou du port série.

Si le bouton MODE PROG est appuyé, on passe au test des boutons UP et DOWN et le bouton de validation des paramètres (vitesse du moteur 1, vitesse du moteur 2 coordonnées x, coordonnées y).

- Si le bouton UP est appuyé, on incrémente le paramètre vitesse Sinon :
- Si le bouton DOWN est appuyé, on décrémente la vitesse des moteurs.
- Si le bouton VALID est appuyé, on valide la valeur de la vitesse, et on passe au prochain paramètre qui est la COORDONNEES X. Pour le troisième paramètre, c'est la même procédure d'incrémementation et de décrémentation qui sera suivie. Vers la fin, on retourne au point de départ (A).

Sinon, si le bouton PROG n'est pas appuyé on retourne au point de départ (A).

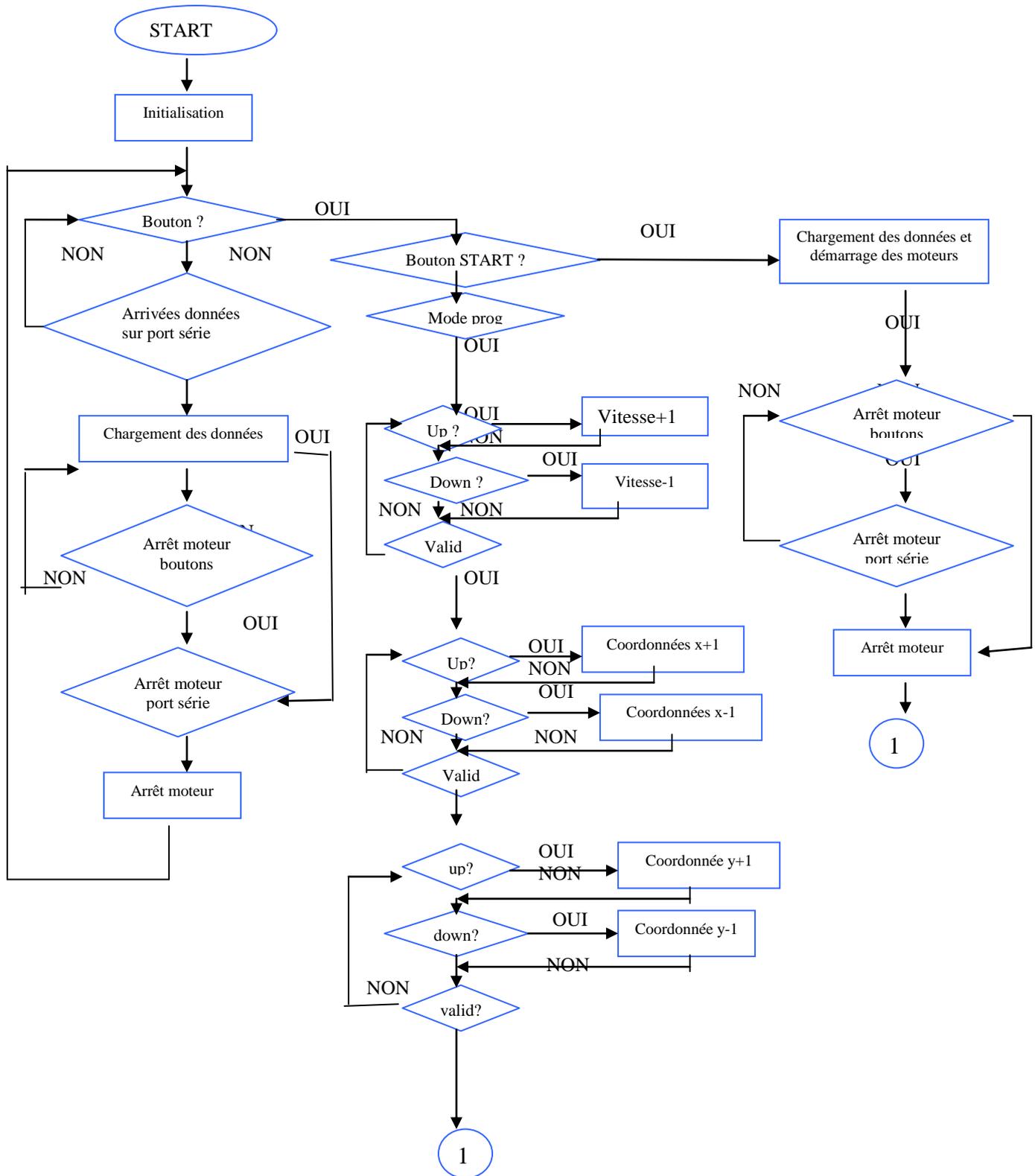


Fig.IV.4 : organigramme de fonctionnement général

IV.3.1.1 Modification du sens de rotation :

Le sens de rotation est défini par l'ordre dans lequel les différents enroulements sont alimentés.

IV.3.1.2 Gestion de la vitesse :

La vitesse est contrôlée par la fréquence F d'envoi des mots binaires de commande. On transforme ainsi une donnée numérique en vitesse. La vitesse des deux moteurs est soit modifiable par les boutons poussoirs comme cité dans l'organigramme, sinon par PC.

❖ Procédure de gestion de la vitesse par microcontrôleur :

Le réglage de la valeur de la vitesse se fait par le pré-diviseur du registre OPTION_REG

Le temps du débordement du Timer0 est réglé dans notre cas à 3276.8micro secondes correspondant approximativement à 3.2milli secondes.

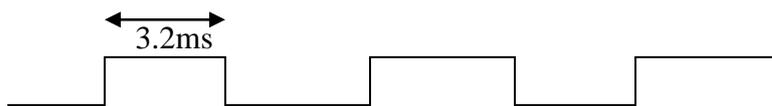


Fig.IV.5 : gestions de la vitesse des moteurs

Chaque 3.2 ms, le moteur effectue un pas. Ceci pour la plus grande vitesse. Le réglage de la vitesse se fait aussi par PC, nous l'avons faite varier de 1 à 10(3.2 à 32ms) pour effectuer un pas. Et comme nos moteurs ont une résolution de 24 pas donc pour faire une rotation complète, le moteur à besoin de $24 * 3.2 = 76.8ms$: $76.8ms/tours$ =donc la fréquence est $76.8 * .001 / 2.3.14 = 0.0122$ Hz. Pour les autres vitesses il faut multiplier ce chiffre par une valeur allant de 1 à 10.

❖ Gestion des interruptions du Timer0 :

La fig.IV.6 illustre la gestion des interruptions du Timer0. La vitesse des moteurs est contrôlée par cette ressource.

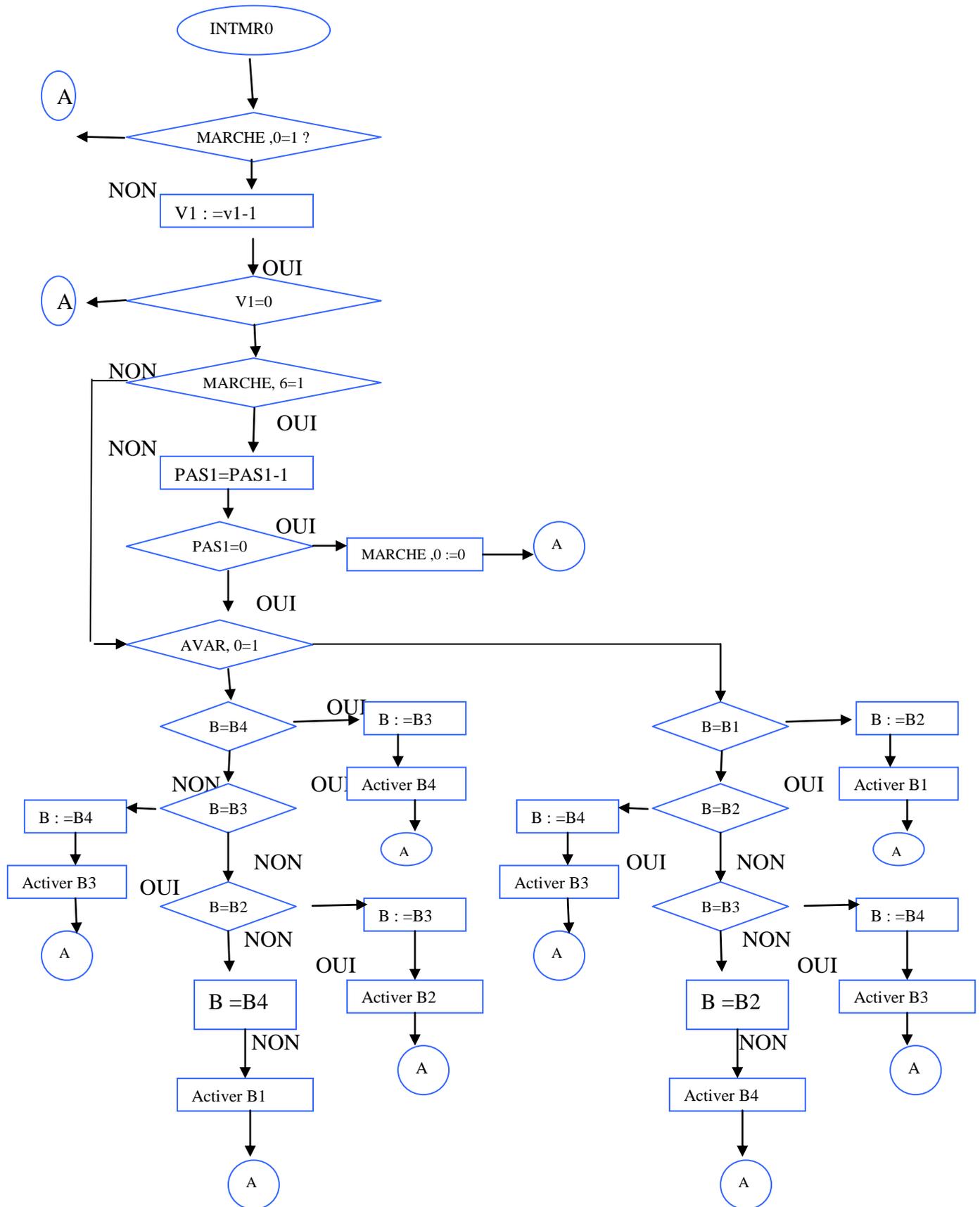


Fig.IV.6 : organigramme de gestion des interruptions du Timer0

Plusieurs registres interviennent dans la gestion de la vitesse des moteurs. Ils sont détaillés comme suit :

❖ **Les registres spéciaux du PIC 16F877A:**

a) **Le registre STATUS (Registre d'état):**

C'est un registre très important car il suit les instructions de notre programme et positionne ses bits suivant le résultat des opérations effectuées. Les cinq premiers bits de ce registre (bits 0 à 4) correspondent à des flags en d'autres termes des drapeaux que le programmeur peut interroger pour obtenir des informations lui permettant d'écrire correctement la suite des instructions de son programme ; tandis que les bits 5, 6 et 7 (RP0, RP1, RP2), pourraient sélectionner les 4 pages de registres.

7	6	5	4	3	2	1	0
RP2	RP1	RP0	TO	PD	Z	DC	C

Bit 0: C (Carry) Flag indiquant si une retenue a eu lieu dans un octet lors d'une addition ou d'une soustraction. Si une retenue a été générée, ce bit passe à 1.

Bit1 : DC (Digit Carry) Flag fonctionnant comme le bit de Carry, sauf qu'ici la surveillance de la retenue s'exerce non pas sur l'octet entier, mais sur le premier demi octet. Ce flag se positionne à 1 si une retenue est générée du bit 3 (bit de poids fort du quartet inférieur) vers le **bit 0** du quartet supérieur. Il est utile pour corriger le résultat des opérations effectuées en code BCD.

Bit 2 : Z (Zero) Ce flag passe à 1 si le résultat d'une opération (arithmétique ou logique) est 0.

Bit 3 : PD (Power Down) Mise en veille de l'alimentation, effectuée par l'instruction SLEEP. Il passe à 1 lorsqu'on utilise l'instruction CLWDT, ou à la mise sous tension.

Bit 4 : TO (Time Out) dépassement de délai, passe à 0 si le Timer du Watch-Dog (chien de garde) déborde, est mis à 1 par les instructions CLWDT et SLEEP, ainsi qu'à la mise sous tension.

Bit 5 : RP0 (Register Page zero) Sert à sélectionner l'une des deux pages de registres (Page 0 ou Page1).

Mais dans le 16F877A le Bit 5 : RP0 et le Bit 6 : RP1 serviront à sélectionner l'un des quatre pages de registre (Bank 0, 1, 2 et3), comme illustré par le tab.IV.1

RP1 : RP0	Page (Bank)
0 0	0
0 1	1
1 0	2
1 1	3

Tab.IV.1 : bits de sélection de bank

b. **Le registre OPTION :**

C'est le registre qui contrôle le fonctionnement de l'horloge interne du microcontrôleur et la manière dont fonctionne le PORTB.

Bit 7: RBPU (Register B Pull Up) : Valide les résistances de pull-up présentes, à l'intérieur du boîtier, sur les lignes du port B.

Bit 6 : INTEDG (INTerrupt EDGe) : Détermine le front du signal d'horloge sur lequel on veut que soit prise en compte une demande d'interruption provenant de l'extérieur (commande appliquée sur la pin 21 : RB0/INT). Une telle demande est réalisée soit à l'apparition d'un front montant (passage de zéro à 1), soit à l'apparition d'un front descendant (passage de 1 à zéro).

1 = interruption programmée pour se déclencher

Sur un front montant ()

0 = interruption programmée pour se déclencher

Sur un front descendant ()

Bit 5 : T0CS (Timer zero Clock Source)

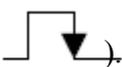
Sert à choisir la provenance du signal qu'on souhaite utiliser comme clock pour piloter l'horloge interne. Il existe deux choix possibles : soit utiliser l'horloge interne utilisant le quartz pilote du microcontrôleur et fournissant un signal dont la fréquence est celle du quartz divisée par 4, soit utiliser un signal externe prélevé sur la pin RA4 (bit 4 du port A).

0 = le timer est piloté par l'horloge interne

1 = le timer est piloté par un signal externe.

Bit 4 : T0SE (Timer zero Signal Edge) : Sert à déterminer si l'horloge doit avancer sur front montant ou sur front descendant.

0 = l'horloge avance sur front montant ()

1 = l'horloge avance sur front descendant ()

Bit 3 : PSA (Pre-Scaler Assignment)

Sert à affecter le pré diviseur soit au Timer TMR0 soit au Watch-Dog.

0 = le pré diviseur est affecté au Timer TMR0.

1 = le pré diviseur est affecté au Watch-Dog.

Bits 2 – 1 – 0 : PS2 – PS1 – PS0 (Pre-Scaler rate).

Ces trois bits servent à programmer le facteur de division qu'on veut assigner au pré diviseur dans le but d'avoir des signaux plus lents. A remarquer que le facteur de division n'est pas le même selon que le pré diviseur soit affecté au TMR0 (Timer) ou au Watch-Dog (chien de garde) : ces facteurs sont représentés par le tab.IV.2

PS2	PS1	PS0	Facteur de division	
			Pour le Timer 0	Pour le Watch-dog
0	0	0	2	1
0	0	1	4	2
0	1	0	8	4
0	1	1	16	8
1	0	0	32	16
1	0	1	64	32
1	1	0	128	64
1	1	1	256	128

Tab.IV.2représentation des facteurs du pré diviseur

c. Le système d'interruption:

Le microcontrôleur 16F877A est doté d'un système d'interruption très performant permettant au programmeur de prendre en charge les événements extérieurs et intérieurs au totale 13 sources en temps réel et d'écrire des programmes très optimisé.

Le Timer0 (TiMeR zéro):

Le Timer0 est un compteur 8 bits, un compteur qui compte les cycles d'instructions du microcontrôleur son registre principal est le registre TMR0 c'est là que le comptage est fait. Son rôle principal est la création de temporisation. Ce Timer peut fonctionner seul, ou bien précédé par un pré diviseur programmable à 8 bits dont la programmation se fait par l'intermédiaire du registre OPTION. Ce registre peut être lu et modifié à tout moment, soit pour connaître sa position courante, soit pour le déclencher à partir d'une valeur déterminée.

Le comptage est sur 8 bits, une fois que le comptage est arrivé à FF, celui-ci revient à 00 (ce qui provoque le passage à 1 du bit 2 du registre INTCON appelé TOIF). En programmation on peut écrire :

- 1) pour le lire :
 - MOVF TMR0, W (porte la valeur de TMR0 dans W)
- 2) pour lui donner une valeur de départ :
 - MOVLW valeur
 - MOVWF TMR0
- 3) pour le mettre à zéro :
 - CLRF TMR0

Ses différents registres sont : TMR0, OPTION_REG, INTCON.

d. La mémoire EEPROM DATA :

C'est une mémoire de sauvegarde semi permanente de données elle peut être lue est écrite à tous moments elle sert a conserver les données même après coupure d'électricité.

Ses différents registres sont :

EEPGD	—	—	—	WRERR	WREN	WR	RD
-------	---	---	---	-------	------	----	----

1- EEADR (EEprom ADRes):

Registre dans lequel on écrit l'adresse de la mémoire de données EEPROM

2- EEDATA (EEprom DATA):

Pendant une opération de lecture : registre dans lequel est disponible la donnée située à une certaine adresse de la mémoire EEPROM.

- Pendant une opération d'écriture : registre dans lequel on place la donnée à l'emplacement qu'on veut écrire.

3- EECON1 (EEprom CONTro1):

Registre de contrôle permettant de définir le mode de fonctionnement de la mémoire de données EEPROM). Registre à 8 bits, mais dont 5 seulement sont utilisés:

4 3 2 1 0

7	6	5	4	3	2	1	0
			EEIF	WRERR	WREN	WR	RD

Bit 0: RD (ReaD)

Normalement à 0. Il se met dans cet état de lui-même. Le programmeur ne peut y écrire que un 1. (N'accepte pas d'être programmé à zéro).

Bit 1: WR (WRite)

Normalement à 0. Il se met dans cet état de lui-même. Le programmeur ne peut écrire que un 1. (N'accepte pas d'être programmé à zéro).

Bit 2: WREN (WRite ENable)

Mis à zéro, interdit toute écriture en mémoire.

Mis à 1, autorise une écriture en mémoire.

Bit 3: WRERR (WRite ERRor)

Flag d'erreur. Normalement à zéro. Passe à 1 pour signaler qu'une erreur s'est produite Juste au moment où une écriture était en cours (Celle-ci n'a pu aboutir parce qu'un événement inopiné s'est produit ; par exemple un Reset).

Bit 4 : EEIF (EEprom Interrupt Flag)

Flag d'interruption. Il est automatiquement mis à 1 lorsque la programmation de l'EEPROM de données est terminée, il doit être mis à zéro par programmation.

4- EECON2 (EEprom CONTro2)

Registre n'ayant aucune consistance physique, et dont le seul rôle consiste à obliger le programmeur à vérifier les données qu'il envoie dans l'EEPROM. Par exemple pour lire une donnée dans la mémoire eeprom, les étapes à suivre sont les suivantes :

- 1) On écrit l'adresse dans le registre EEADR;
- 2) On met à 1 le bit 0 (RD : Read Data) du registre EECON1 (ce qui provoque le transfert de la donnée dans le registre EEDATA);
- 3) On lit la donnée dans le registre EEDATA où elle est devenue disponible.

Pour écrire dans l'EEPROM, les étapes à suivre sont un peu plus complexes, car on est obligé de passer d'abord par EECON2 avant de confirmer la donnée par EECON1 :

- 1) on interdit les interruptions.
- 2) on écrit la donnée dans le registre EEDATA.
- 3) on écrit l'adresse dans le registre EEADR.
- 4) on configure le registre EECON1.
- 5) on envoie la séquence définie par MICROCHIP (55 et AA) à EEcon2.
- 6) on reconfigure les registres EECON1, INTCON et STATUS.

e. Les entrées sortie :

PORT A, PORT B et PORT C -TRIS A, TRIS B et TRIS C :

Le PORT A, PORTB et PORTC permettent au microcontrôleur de communiquer avec l'extérieur.

Alors que TRISA, TRISB et TRISC se limitent à définir le sens de chaque ligne des ports entrée ou sortie).

f. Le Timer2 :

Le Timer2 est un compteur sur 8 bits, il possède un pré diviseur, qui peut être paramétré avec une des 3 valeurs suivantes : 1,4, ou 16. Cependant, il dispose également d'un post diviseur, qui effectue une seconde division après l'unité de comparaison. Ce post diviseur peut prendre n'importe quelle valeur comprise entre 1 et 16, ce qui donne un grand choix possible. La valeur du diviseur total, vue par l'utilisateur, est bien entendu obtenue en multipliant la valeur du pré diviseur par celle du post diviseur.

Les valeurs de diviseur sont :

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,20,24,28,32,36,40,44,48,52,56,60,64,80,96,112,128,144,160,176,192,208,224,240,256.

IV.3.2 Les modes de fonctionnement du robot SCARA :

Trois modes de fonctionnement sont programmés pour être exécutés par le robot :

Le mode automatique, le mode pas à pas, et le mode tâche automatisée.

- **le mode auto** : dans ce mode les moteurs pas à pas fonctionneront comme s'ils étaient des moteurs synchrone, c'est-à-dire ils tournent en permanence sans arrêt jusqu'à atteinte de la position finale désirée décidée par l'opérateur à travers les deux coordonnées x et y .donc si les coordonnées x et y sont introduite par clavier, et si la commande send robot est activée, la pince sera positionnée sur le point indiqué.
- **le mode step angle** : dans ce mode, la position désirée sera atteinte dès Que nous indiquons le nombre de pas que doivent effectuer les moteurs 1 et 2.

Les modes 1 et 2 (auto et step angle) sont des modes de réglage manuel, ils serviront de test.

- **le mode tâche automatisée** : dans ce mode, nous désignons un nombre de positions à atteindre par le bras. Ce dernier se charge de les exécuter une par une jusqu'à la dernière position.

Exemple d'application :

Comme application, nous avons choisi les positions de coordonnées suivantes indiquées sur la figIV.7 A (0.2, 0.2) B (0.35, 0.2) C (0.40, 0.1) D (0.15, 0.30).

En tenant compte des conditions initiales : effecteur à la position (0.4, 0.0).

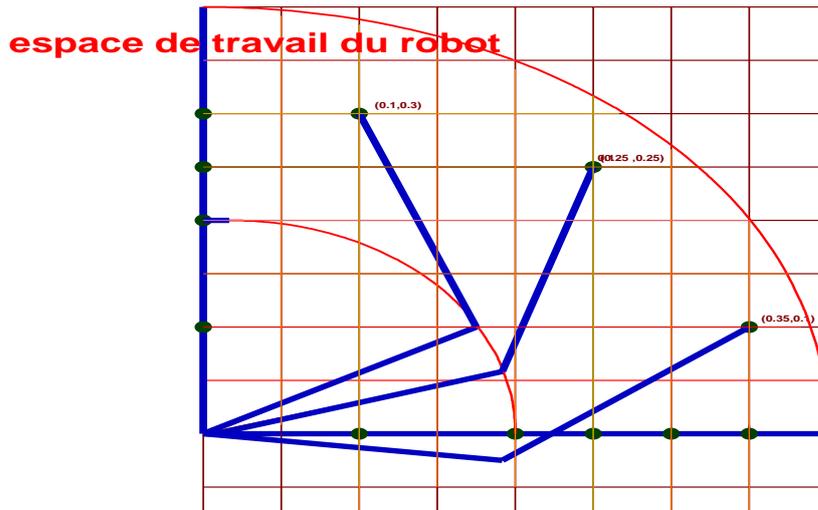


Fig.IV.7 quelques positions du bras

IV.4 Interface logicielle sur PC :

Afin d'établir une communication entre le PIC et le PC à travers la liaison série RS232, nous avons réalisé une interface graphique sous l'environnement Delphi 7.

IV.4.1 Aperçu sur Delphi :

Delphi est un logiciel de développement rapide (RAD) (Rapid Application Development) conçu par Borland pour écrire des applications Windows. Il permet de produire des programmes Windows plus vite et plus facilement que tous les autres outils qui existaient auparavant.

Lorsqu'on développe des applications sous Delphi, on dispose de toute la puissance d'un véritable langage complet qui est le pascal intégré dans cet environnement. Cela signifie qu'on peut rapidement construire l'interface: les menus, les boîtes de dialogue, la fenêtre principale, à l'aide du mécanisme glisser, déplacer de Windows.

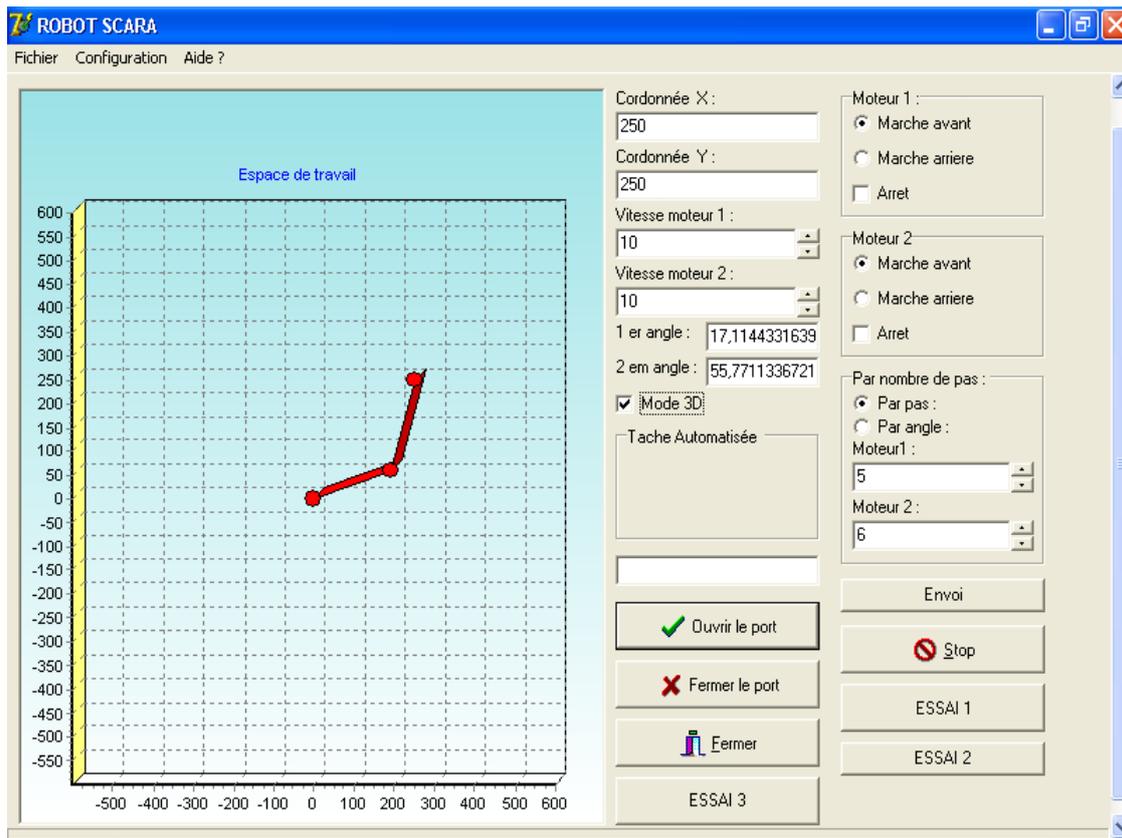


Fig. IV.8 : Création de l'interface robot SCARA

IV.4.2 Le Composant C port :

Dans la palette d'objet, on peut voir tous les composants standards qui sont installés d'office dans Delphi (exemple : les boutons, les étiquettes, les Timers,). Mais il existe d'autres composants gratuits ou bien prépayés qui ne sont pas intégrés dans Delphi. Comme notre projet a besoin d'un composant qui peut communiquer avec le port série (RS232) pour envoyer et recevoir des données de l'application vers le microcontrôleur.

Le Cport est un composant gratuit qui peut être téléchargé sur Internet à l'adresse suivante: <ftp://ftp.serfop-reims.org/Delphi/Serie32/Serie32.zip>.

Ce dernier sert à communiquer entre l'application Delphi et le microcontrôleur à travers le port série.

IV.4.2.1 Installation d'un nouveau Composant : (Cport 3.0)

Après avoir lancé Delphi, cliquer sur 'outils' ensuite sur 'options d'environnements' dans la fenêtre qui apparaît, et sur l'onglet Bibliothèque cliquer sur chemin bibliothèques et parcourir le composant Cport 3.0.

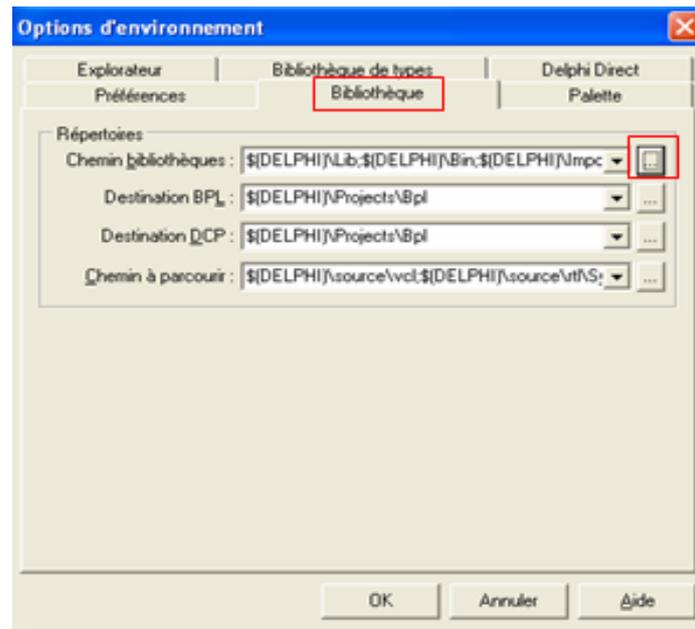


Fig IV.9 installation du composant Comport

Dans le menu 'composant' cliquer sur ' installer un composant ' : une fenêtre apparaît, cliquer sur parcourir ' Nom d'unité ' et après parcourir le composant Cport 3.0, ensuite une autre fenêtre apparaît, cliquer sur le bouton compiler. Enfin un message de confirmation demande l'autorisation de l'installation du composant. Cliquer Oui pour l'installer.

IV.4.3 La configuration du port série

Lorsque le port est ouvert, plusieurs paramètres peuvent être configurés à savoir : la vitesse de transmission, le nombre de bits stop, le contrôle de parité...etc. Dans notre cas nous avons utilisé le protocole de communication par défaut (9600 bauds, 8 bits de données, 1bit stop, pas de bit de parité). Le type de câblage utilisé est la connexion NULL MODEM.

IV.4.3.1 Protocole de communication :

a) Format des trames :

Au repos, la ligne est à 1. Un passage à 0 indique que la transmission va commencer, c'est le bit de START.

-Les bits de données sont ensuite transmis sur un format de 7 ou 8 bits.

-Un bit de parité peut être ajouté pour vérifier la validité des données.

-Un ou deux bits de Stop terminent la transmission pour le retour au repos de la ligne.

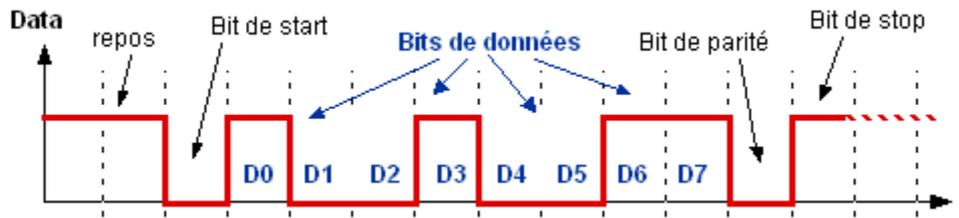


Fig IV.10 Les différents bits Constituant un octet dans le protocole du port série

Le bit de parité est tel que la somme des bits à l'état 1 est paire, dans le cas contraire, il y a eu une erreur de transmission. Il est aussi possible de travailler avec un nombre Impair de bits.

b) Débit des données :

La vitesse de transmission caractérise le nombre de bits transmis par secondes. Elle s'exprime en bauds. Les valeurs courantes sont entre 300 et 9600 bauds

IV.4.3.2 Exemple de configurations :

Exemple 1 :

4800, 8, n, 1 : 4800 bauds, 8 bits de données, pas de parité (no), 1 bit de stop

Exemple 2 :

600, 7, p, 2 : 600 bauds, 7 bits de données, 1 bit de parité, 2 bit de stop.

Notre configuration :

9600, 8, n, 1 9600 bauds, 8 bits de données, pas de parité (no), 1 bit de stop.

Ces informations sont nécessaires pour configurer les deux équipements de transmission

IV.4.3.3 Transmission des données: le programme d'interface envoie les six commandes suivantes à la carte qui les interprète :

- La vitesse des moteurs 1 et 2.
- Le nombre de pas du moteur 1 et 2.
- Le sens de rotation avance ou arrière.

IV.4.4 L'interface graphique :

Les différentes commandes envoyées sont illustrées par la fig.IV.11.

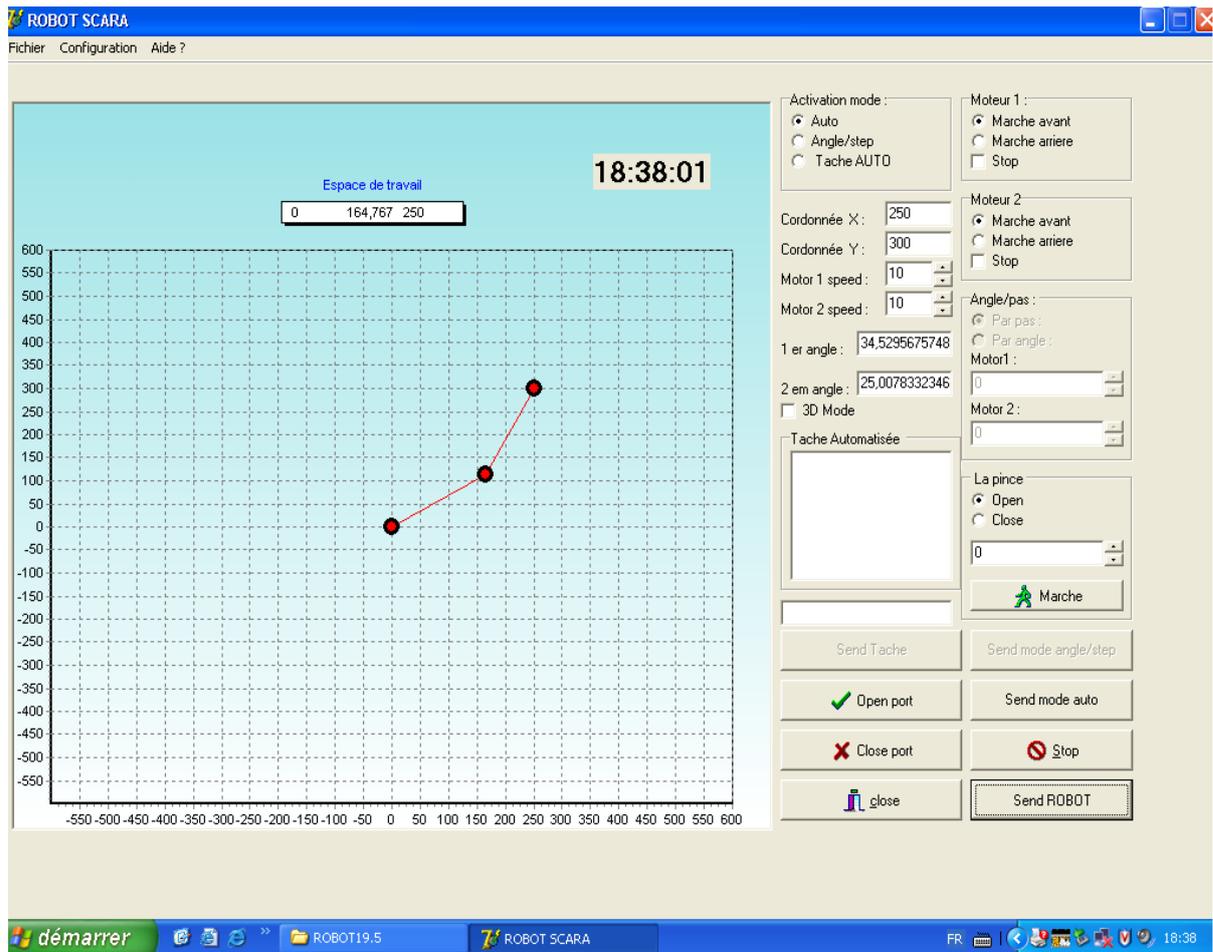


Fig.IV.11 : Interface du projet

La programmation de l'interface pour la commande des moteurs pas à pas permet le calcul des positions des moteurs elle fournit une interruption de commande pour créer et contrôler la génération d'une trajectoire donnée.

Le programme principal prend la forme d'une boucle qui va durer pendant un temps déterminé par la base de temps.

IV.4.4.1 Initialisation de la position des moteurs :

Comme nous avons utilisé une boucle ouverte pour asservir les moteurs pas à pas (ce qui les caractérise des moteurs à courant continu) donc nous avons initialisé ces moteurs logicielle- ment à des valeurs de (0, 0, 0) référence 0 (zéro) dans un milieu bien sécurisé c'est-à-dire qu'il n' y'aurait pas des perturbations ou bien une intervention d'un opérateur sur les moteurs pour modifier leurs positions manuellement. Dans le cas échéant le recours à des capteurs de positions serait une nécessité.

IV.4.4.2 Le déroulement du programme :

Après initialisation des PORTS d'entrés sorties selon la configuration voulue tel que :
 PORTA : 001111 (4 entrées venant de l'afficheur LCD, 2 sorties vers les deux Switch
 S2 :RA4 S3 :RA5.

PORT B : 11111111 (les 8 entrées des fins de course).

PORTC 10000000 sorties vers le moteur 3 une sorti vers le PC et une entrée du PC.

PORTD 00000000 sorties vers les deux moteurs 1 .2.

PORTE : 001 deux sorties vers l'afficheur LCD RE0=E Enable de l'afficheur LCD.
RE1=RS (register select de l'afficheur) RE2.

Une entrée provenant du switch S4. A la première mise sous tension à travers le switch (SW), le PIC s'initialise automatiquement à des valeurs par défaut. Ces valeurs sont coordonnées $x=0$, coordonnées $y=0$, la vitesse $=0$ (x, y, v) = (0, 0, 0). C'est à l'opérateur de régler ces valeurs à sa guise.

Après réglage par le clavier formé des trois boutons poussoirs, ces valeurs seront stockées en mémoire EEPROM via des registres spécifiés. Ainsi après une coupure d'alimentation, ces valeurs resteront inchangées. Pour une éventuelle modification de ces valeurs de fonctionnement du bras de robot les BP1 BP2 et BP3 sont utilisés comme un clavier de navigation et de mémorisation.

IV.4.4.3 Temporisation

Comme l'horloge du PIC 16F877A est d'une valeur de 20 Mhz et avec le pré diviseur par 4 interne donc il tourne à une fréquence de $20/4=5\text{Mhz}$ et $1/5*10^6=200$ nano secondes une très grande vitesse qui nécessite d'introduire des temporisations.

La fréquence de commutation des bobines des moteurs pas à pas :

La fréquence de commutation des moteurs pas à pas est concrétisée dans le programme par une temporisation de 39 milli secondes ce qui donne $1/39*10^{-3}=25.64\text{hz}$ (modifiable par logiciel).

Cette fréquence est déterminée expérimentalement par rapport au fonctionnement auquel le bras est dévoué, au domaine de travail, et à la tâche à laquelle le bras est destiné.

IV.5 Sélection du mode de fonctionnement :

Le mode de fonctionnement peut être de deux types soit manuel, soit automatique:

IV.5.1 Fonctionnement manuel :

Le mode manuel est réalisé par la navigation entre les trois boutons poussoirs S2, S3 et S4. La position désirée du robot manipulateur sera atteinte grâce aux réglages manuels.

IV.5.2 Fonctionnement automatique :

Les ordres dans cette situation seront donnés par un logiciel en liant la carte de commande à un PC à travers la connexion du port série RS232.

Dans ce cas les ordres seront donnés par un logiciel réalisé sous Delphi.

Le robot est destiné essentiellement à fonctionner comme un bras de perçage de cartes électroniques guidés par logiciel de dessin de circuits électronique et circuits imprimés. Ce guidage se fait via un PC avec un port série RS232.

Le logiciel se charge d'orienter le bras vers différents points de perçage désirés et ce pour chaque composant électronique ou chaque trou à percer.

Pour visualiser les données issues du PIC vers l'afficheur et vers les moteurs, une temporisation est imposée. Elle est de différentes valeurs selon le besoin.

IV.5.3 Le programme :

Le programme est destiné à gérer un nombre important de robot, on peut citer les robots à chaîne ouverte simple, les robots parallèles et les robots à structure fermée. Pour cela il suffit d'intervenir sur les longueurs des tiges, l'espace de travail et le nombre de degrés de liberté.

L'utilisation d'un afficheur LCD nous permet de voir, de scruter et de maintenir le bras à n'importe quelle situation imprévue.

IV.6 Conclusion :

Dans ce dernier chapitre, nous avons exposé la partie logicielle de notre travail, les routines et les sous routines que doit exécuter le microcontrôleur pour le bon fonctionnement de la carte ainsi réalisée. Ainsi, nous avons présenté la commande des trois moteurs pas à pas par le microcontrôleur PIC 16F877A. Par programme, nous avons modifié la vitesse des moteurs.

CONCLUSION GENERALE

Conclusion Générale

Dans ce travail, nous avons présenté la commande en position/vitesse des moteurs pas à pas. L'objectif de cette étude est l'utilisation de ces actionneurs dans la motorisation des robots manipulateurs pour des applications de très grande précision et de faible puissance.

La commande en boucle ouverte est un atout majeur de ce moteur. De plus, la sortie de ce dernier, contrairement aux autres moteurs, est une position et non pas une vitesse, donc la commande de position est typique à ce genre d'actionneurs. La commande par des microcontrôleurs est utilisée dans ce travail car le moteur pas à pas s'adapte bien à la commande numérique [20].

Nous avons conçu et réalisé une carte de commande à base d'un microcontrôleur interfacée à un PC. Cette carte se charge de l'asservissement en position de trois moteurs pas à pas destinés à la commande d'un robot SCARA à deux articulations rotoides doté d'une pince à deux doigts.

Toutefois, le système de commande réalisé dans ce travail est applicable dans la majorité des bras manipulateurs actionnés par des moteurs pas à pas en boucle ouverte. Il est économique et permet au concepteur d'utiliser les multiples possibilités inexploitées du microcontrôleur inclus dans le montage.

Perspectives :

A la lumière des résultats obtenus plusieurs perspectives s'ouvrent à nous, en collaboration avec des mécaniciens pour réaliser des robots SCARA plus sophistiqués, de même la réalisation de plusieurs cartes de commande interfacées à un seul PC.

Pour performer notre carte de commande, nous proposons des capteurs optiques qui vont détecter si la position désirée est atteinte ou non et ce dans le cas où nous travaillons dans un environnement encombrant.

REFERENCES
BIBLIOGRAPHIQUES

Références Bibliographiques

- [1] :E. Dombre, "Programmation des robots" Techniques de l'ingénieur, Génie électrique
- [2] :N.SEGUY, "Système de commande d'un manipulateur". Techniques de l'ingénieur, Génie électrique.
- [3] :H.MIYAMOTO : "Modélisation et commande optimale d'un moteur pas à pas par microprocesseur" Thèse de docteur ingénieur, Institut National Polytechnique de Lorraine, 31, mai, 1979.
- [4] :P.OGUIC, "Moteur pas à pas et PC", DUNOD, Paris 2004.
- [5] : M.KANT : "Les actionneurs électriques pas à pas", Edition Hermès, Paris ,1989.
- [6] :J.M. Kauffman : "Les petits moteurs évolution et perspectives", 7^{ème} colloque international sur les Moteurs Pas à Pas, pp1-12, Nancy ,1992.
- [7] :M.ABIGNOLI et C.GOELDEL "Moteur pas à pas ", Techniques de l'ingénieur, Génie électrique, DIII ,1991 pp.D550-1 D550-21.
- [8] :B. Multon : " Conception et alimentation électronique des machines à réluctance variable à double saillance", Thèse de Docteur de Troisième Cycle, Ecole Normale Supérieur de Cachan, 1994.
- [9] :M.JUFER : "Electromécaniques", Presse Polytechniques et Universitaires Romandes, Lausanne 1995.
- [10]: J.F.GIERAS and M.WING:" Permanent magnet motor technology", design and application. Edition Marcel Dekker, United States of America 2002.
- [11] :C.PAROZ " Commande incrémentale à fréquence fixes et à avance variable" 7^{ème} colloque international sur les Moteurs Pas à Pas, pp95-102, Nancy ,1992.
- [12]: J.LUCIDARME, A.AMOURI and M.POLOUJADOFF "Optimum design of longitudinal field variable reluctance motor-application to a high performance actuator", IEEE Transaction on Energy Conversion, vol .8N°3 pp361-375, 1993.
- [13]: T.HANS, J.FILIPPINI et P.GUYENOT "Asservissements numériques, éléments de cours –applications " Edition Eyrols –France 1991.
- [14] :G.LACROUX, "Actionneurs électriques pour la robotique et les asservissements", Technique et documentation LAVOISIER ,1994.
- [15] :P.MAYE, "Moteurs électriques pour la robotique", DUNOD, Paris 2000.
- [16] :W. Khalil et E. Dombre, "Modélisation, identification et commande des robots", 2^{ème}

Édition revue et augmentée, Hermès Sciences, 1999.

[17]:P.PAUL, ACARNLEY, A.HUGHES, IEEE 1988; "Machine/Drive Circuit Interactions in Small Variable-Reluctance Stepping and Brushless DC Stepping and Systems", IEEE Transaction on Industrial Electronics, VOL 35, NO 1, pp.67-73, February 1988.

[18] :A.MAILFERT, "Machines à réluctance variable", Techniques de l'ingénieur, Génie électrique, D3II, pp.D550-1 D550-20, 1994.

[19] :L.EL AMRAOUI : "Conception électromécanique d'une gamme d'actionneurs linéaires à réluctance variable"Thèse de Docteur, Ecole Centrale de Lille, Université des Sciences et Technologies de Lille, Ecole Nationale d'Ingénieur de Tunis 18, décembre, 2002.

[20] :H.SAHRAOUI, M.S.BOUCHERIT et L ZEBROWSKY. "Etude de l'influence du type de l'alimentation sur le comportement dynamique d'un moteur pas à pas à réluctance variable", proceedings of the Maghrebin Conference on Automatics and Industrial Electronics, pp258-267, Algeria 1993.

[21] :P.BARRET, "Régime transitoire des machines électrique tournantes", Techniques de l'ingénieur, Génie électrique, D3II, pp.D554-1 D554-23,1994

[22] : W.KHALIL : "Commande des robots manipulateurs"LAVOISIER, HERMS 2002.

[23]: S.KRUT, O.COMPANY and F.PIERROT, "Velocity Performance Indexes for Parallel Mechanisms with Actuation Redundancy", Int. J. of Robotica, Cambridge University Press, 2003.

[24]: S.KRUT, O.COMPANY, F.PIERROT, "Velocity Performance Indexes for Parallel Mechanisms with Actuation Redundancy", in Proc. of Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators, pp. 46-56, Quebec City, Quebec, Canada, October 3-4, 2002.

[25]: S.KRUT, O. Company O, F.MARQUET, F.PIERROT, "Twice: A Tilting Angle Amplification System for Parallel Robots", in Proc. of IEEE ICRA: Int. Conf. on Robotics and Automation, pp. 4108-4113, Washington, DC, USA, May 11-15, 2002.

[26]:S.KRUT, O.COMPANY, M.BENOIT, H.OTA and F.PIERROT, "I4: A new parallel mechanism for SCARA motions", in proc. of IEEE ICRA: Int. Conf. on Robotics and Automation, Taipei, Taiwan, September 14-19, 2003.

[27]: C.ATKESON and J.MCINTYRE ("Robot trajectory learning through practice", in Proceedings of IEEE Conference on Robotics and Automation, (San Francisco, Californie), 1986.

[28]:J.ANGELES, A.MOROZOV and O.NAVARRO, novel manipulator architecture for the production of SCARA motions " in proceeding of IEEE ICRA: International Conference on Robotic and Automation, pp, 2370-2375, San Francisco, April 24-28, 2000.

[29]:J.LOVINE, "PIC Robotics", McGraw-Hill, 2004

[30]:G.J.LIPOVSKY,"introduction to microcontrollers", Academic Press Series in Engineering, 1999.

[31]:J.ILETT,"How to use intelligent L.C.D.s", everyday Practical Electronics, February 1997.

[32] Apprendre la programmation des PIC

Webliographie :

[33] :delphi J-C Armici janvier 2003 www.unvrai.com.

[34] : [http:// www.microchip.com](http://www.microchip.com)

[35] : [http:// www.abcelectronique.com](http://www.abcelectronique.com)

[36] :[http:// www.developpez.com](http://www.developpez.com)

[37] : [http:// www.efrance.fr/labibledupic16f84](http://www.efrance.fr/labibledupic16f84)

[38] Cours de Bigonoff : www.von-info.ch/pic/bigonoff_01.Html

[39] Data sheet: www.all_datasheet.com

[40] MPLAB 7.40 : www.microchipe.com

[41] Le programmeur de PIC : www.SUPERPRO.com

[42] IC PROG 1.05D: www.Icprog.com

[43] Les modes de Transmission : www.comment_ça_marche.com.

[44] ScienceDirect Home <http://www.sciencedirect.Com>.
Techniques de l'Ingénieur <http://www.Techniques-ingenieur.fr/>

[45] Control Theory and engineering Links <http://www.Theorem.Net/control.html>

[46] Center for Mechanics of Micro-Systems, NUT, Nanyang <http://cmms001.cmms.mpe.ntu.edu.Sg/groups/cmmstry/>

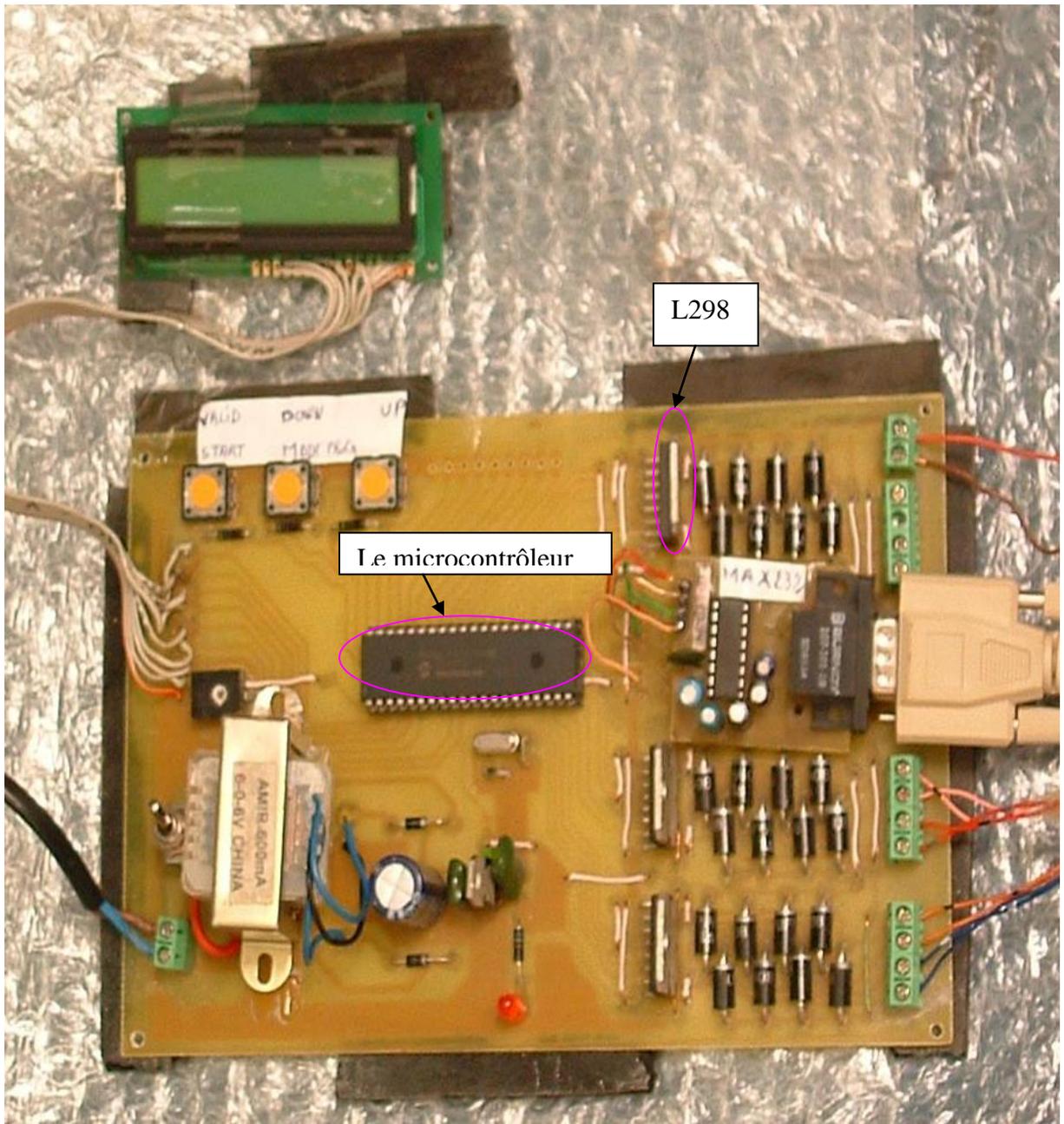
[47] USU center for self-organizing intelligent systems (CSOIS) autonomous control robotics, <http://www.csois.usu.edu/>

[48] Control & Communication, Linköpings universitet, Sweden, <http://www.control.isy.liu.se/>

ANNEXE

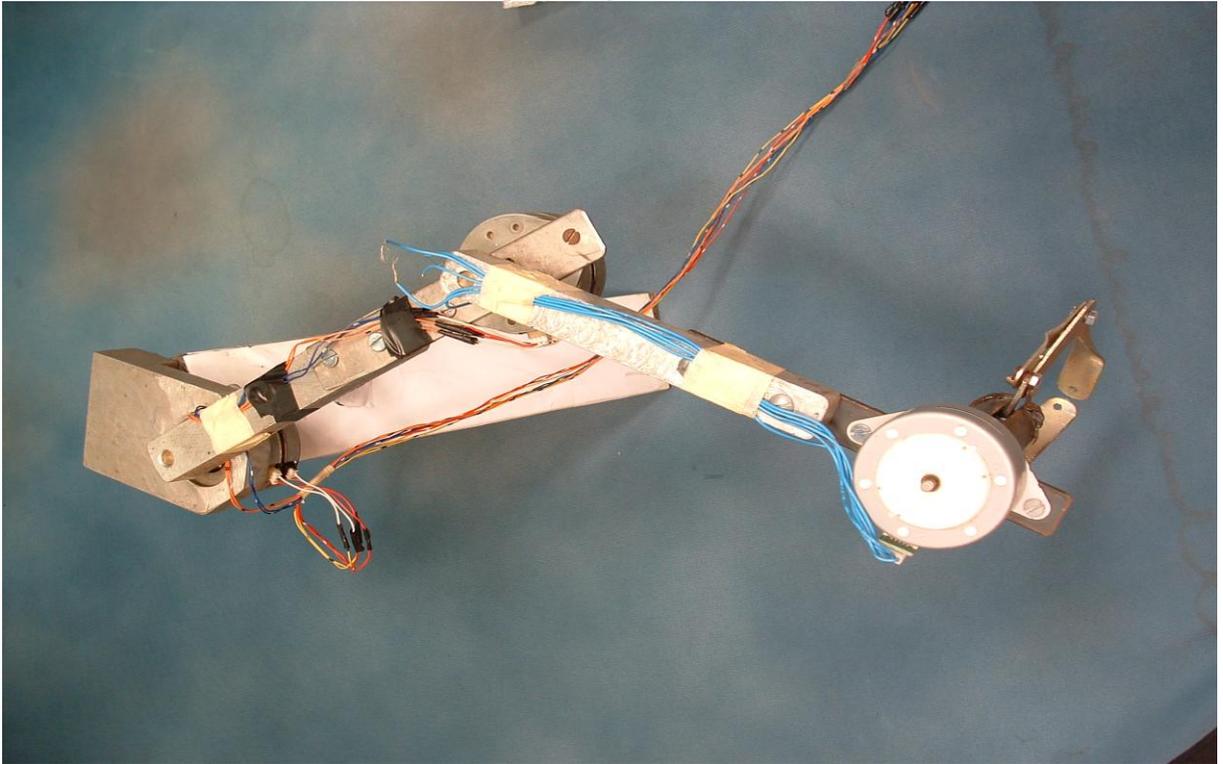
Annexe

La carte de commande réalisée :



Annexe

La structure du bras réalisé est donnée par la figure suivante :



La structure complète de commande

