

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Electronique
Option : micro informatique et instrumentation

Réf:.....

Mémoire de Fin d'Etudes
En vue d'obtention de diplôme:
MASTER

Thème :

**Réalisation de carte à
microcontrôleur pour le contrôle de
bras manipulateur via un pc**

Présenté par :

✚ Mr. SAADI RAMZY
✚ Mr. SALHI NASSEREDDINE

Proposé et Dirigé Par:

✚ Mr. GUESBAYA TAHAR

Promotion :2009/2010

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra
Faculté des Sciences et de la Technologie
Département de Génie Electrique
Filière : Electronique
Option : micro informatique et instrumentation

Mémoire de Fin d'Etudes
En vue d'obtention de diplôme:
MASTER

Thème :

Réalisation de carte à microcontrôleur pour le contrôle de bras manipulateur via un pc

Présenté par :

SAADI RAMZY

SALHI NASSEREDDINE

Avis favorable du président du Jury

Nom et prénom

.....

.....

Avis favorable de l'encadreur

Signature

.....

Cachet et signature



Résumé du mémoire de fin d'étude

En vue d'obtention de diplôme:

MASTER

Option : **micro informatique et instrumentation**

Thème :

Réalisation de carte à microcontrôleur pour le contrôle de bras manipulateur via un pc

Proposé par : **GUESBAYA Tahar.**

Dirigé par : **GUESBAYA Tahar.**

RESUME

L'objectif de ce travail est de réaliser une carte d'interface à base de microcontrôleur dans le but de commander un bras manipulateur via un PC en utilisant le port série RS232.

Ainsi, cette thèse se compose à deux parties ; la première partie concerne une étude théorique sur le bras manipulateur ROB03 et le PIC16F877A.

La deuxième partie est consacrée pour l'étude conceptuelle des schémas de commande du bras via un PC et le microcontrôleur.

Cette étude est faite dans l'optique de réaliser un support comportant :

- Sur le plan *Hardware*: une carte de commande comportant le microcontrôleur *16F877* est réalisée, ensuite une carte de puissance pour assurer l'alimentation des six moteurs du *ROB3* est construite.
- Sur le plan *Software*: une interface graphique est développée en employant le *Builder-C*

Dedicace

Je dédie ce travail à mes très chers parents qui ont tant prié à mon réussite,

De soutien moral. Je les remercie de m'avoir encouragé

Et aidés à devenir ce que Je suis.

Et A mes frère, mes sœurs,

et à toute la famille.

Et mon collègue SALHI Nassereddine, et à toute la promotion

Master De l'année 2010

A mes très chers amis.

RAMZY

Dédicace

Je dédie ce travail à mes très chers parents qui ont tant prié à mon réussite,

De soutien moral. Je les remercie de m'avoir encouragé

Et aidés à devenir ce que Je suis.

Et mon frère Youcef, mes sœurs,

et à toute la famille.

Et mon collègue SAADI Ramzy, et à toute la promotion

Master De l'année 2010

A mes très chers amis.

Nassereddine

Remerciement

Tout d'abord, Nous tiens à remercier DIEU le miséricordieux de nos avoir donné la possibilité de réaliser notre projet, d'arriver à notre souhaits et d'atteindre notre objectifs.

Nous aimerons dans ces quelques lignes remercier toutes les personnes qui d'une manière ou d'une autre, ont contribué au bon déroulement de notre travail, tout au niveau humain qu'au niveau scientifique.

Nous tenons tout d'abord à remercier notre encadreur MONSIEUR, GUESBAYA TAHAR, on a pu bénéficier à la fois de ses compétences scientifiques, et de sa grande disponibilité, tant pour résoudre les difficultés rencontrées lors de notre réalisation, de répondre à nos questions.

Nous ajoutons en particulière sa patience et ses encouragements, nous a permis de travailler dans bonnes conditions.

Nous exprimons notre gratitude à l'ensemble des professeurs du département d'automatique qui ont contribué à notre formation, on les prie de bien vouloir croire à notre gratitude en espérant que cet humble travail de trois années fera crédibilité de leurs efforts.

Grand remerciement à tous les ingénieurs des laboratoires de génie électrique : Mr. HAMZA, Mr. ABDELATIF, Mr AHMED.

Nos remerciements s'adressent également à tous membres de Jury, qui ont accepté de nous honorer de leur présence et de juger notre travail Merci.

Et à toute personne ayant contribué de près ou de loin à notre soutien moral.

Sommaire

INTRODUCTION GENERALE	1
-----------------------------	---

Chapitre I : Généralité sur les robots

I.1 Introduction	4
I.2 Définitions	5
I.3 Constituants d'un robot	6
I.4 Classification des robots	8
I.4.1 Classification fonctionnelle	8
I.4.1.a manipulateurs à commande manuelle ou télécommande	8
I.4.1.b manipulateurs automatiques à cycles pré-réglés	8
I.4.1.c robots programmables	9
I.4.1.d robots intelligents	10
I.4.2 Classification géométrique	10
I.5. Performances des robots	11
I.5.1 Les performances-tâches	12
I.5.1.a Le volume atteignable (volume de travail)	12
I.5.1.b Charge utile	14
I.5.1.c Précision / Répétabilité	14
I.5.1.d La vitesse	14
I.5.1.e L'orientation possible de l'organe terminal	14
I.5.1.f La fiabilité	15
I.5.2 Les performances-homme	15
I.5.3 Les performances économiques	15
I.6. Les différents types des robots	16
I.6.1 Les robots SCARA	16
I.6.2 Les robots cylindriques	17
I.6.3 Les robots sphériques	18
I.6.4 Les robots Cartésiens	18
I.6.5 Les robots parallèles	19
I.6.6 Les robots anthropomorphe	19
I.7. Utilisation des robots	20
I.7.1. Tâches simples	20
I.7.2. Tâches complexes	20
I.8. Avenir de la robotique	21
I.9. Conclusion	21

Chapitre II : Description du bras manipulateur ROB03

II.1 Introduction	23
II.2 Description cinématique du bras manipulateur	23
II.3. Espace du travail du manipulateur <i>ROB3</i>	24
II.4 Propriétés du bras robot <i>ROB3</i>	24

II.4.1. Propriétés électrique	24
II.4.2. L'unité <i>TEACHBOX</i>	25
II.4.3. Limite en commande	26
II.4.4. Propriétés mécaniques	26
II.5 Les actionneurs	28
II.5 .1 Caractéristiques du moteur à courant continu	29
II.5 .2 Modélisation du moteur à courant continu	29
II.6 Transmissions et systèmes réducteurs de vitesse	30
II.7 Les capteurs	30
II.8 Conclusion	31

Chapitre III : Le PIC 16F877A

III.1. INTRODUCTION	33
III.2. LES CARACTERISTIQUES DU PIC 16F877A	33
III.3. DESCRIPTION GENERALE DU PIC 16F877A	34
III.3.1. Architecture externe	34
III.3.2. Architecture interne	36
III.3.3. Organisation de la mémoire	37
III.3.4. Les registres du PIC 16F877A	38
III.3.5. Les timers	39
III.3.6. Les interruptions	41
III.3.7. Les ports d'entré sorties	41
III.3.8. Configuration des portx, les registres PORTX et TRISX	42
III.3.9. Le convertisseur analogique numérique	42
III.3.10. Les modules CCP1 et CCP2	44
III.4. CONCLUSION	45

Chapitre IV : Réalisation hardware et software

IV.1 Introduction	47
IV.2. Schéma simplifié du système de commande	47
IV.3. La carte de commande	48
IV.3.1- Schéma électronique de la carte de commande	48
IV.3.2 Eléments constituant la carte de commande	49
IV.3.2.1 Le microcontrôleur PIC	49
IV.3.2.2 Le circuit MAX232	50
IV.3.2.3 Le circuit <i>74HC138</i>	54

III.3.2.4 Le circuit 74HC04	54
IV.4 La carte de puissance	55
IV.4.1 Schéma électronique de la carte de puissance	56
IV.4.2 Eléments constituant la carte de puissance	56
IV.4.2.1 Le circuit L298N	56
IV.4.2.2 Les diodes 1N4007.....	58
IV.4.2.3 Le circuit d'alimentation	58
IV.5 La programmation du PIC	59
IV.5.1 Environnement de programmation	59
IV.5.2 Le circuit programmeur	59
IV.5.2 .1 Schéma de principe	60
IV.6 Type de commande	61
IV.6.1 Le signal de commande PWM(MLI)	61
IV.7 L'organigramme général	62
IV.8 La carte d'interface	63
IV.9 Réalisation software	64
IV.9.1 Interface principale	64
IV.9.1.1 Interface de commande manuelle	65
IV.9.1.2 Interface de commande par angle	65
IV.10 Conclusion	66

Chapitre V : Software

V.1 Introduction	68
V.2 Présentation générale du logiciel 'PROTEUS '	68
V.2.1 L'éditeur de schéma ISIS	68
V.2.1.1. Interface utilisateur	69
V.2.1.1.a. Fenêtre d'ensemble	69
V.2.1.1.b. Fenêtre d'édition	69
V.2.1.1.c. La boîte à outils	69
V.2.1.1.d. Coordonnées du curseur	70
V.2.1.1.e. Organisation de la boîte à outils	70
V.2.1.2. Etapes de la saisie de schéma	71
V.2.2 L'outil de conception de circuit imprimé ARES	72
V.2.2.1 Présentation de ARES	72
V.2.2.2 Création d'une carte	73

V.2.2.3 Le placement des composants	73
V.2.2.3.a Placement automatique	73
V.2.2.3.b Placement Manuel	73
V.2.2.4 Le routage	74
V.2.2.4.a Routage automatique	74
V.2.2.4.b Routage manuel	74
V.3 Le pic c compiler	74
V.4 L'ICPROG	78
V.4.1 Configuration icprog	78
V.5 LE C++ Builder	80
V.5.1. Environnement RAD	80
V.5.2. L'interface de C++ Builder	80
V.5.3. Création d'une application simple C++ Builder	81
V.5.4 Conclusion	81
Conclusion générale	82
Perspective	82
Bibliographie	83
Annexe	84

Liste de figure

Chapitre I : Généralité sur les robots

Figure I.1: vocabulaire du robot	6
Figure I.2: parties principales dans un robot	6
Figure I.3: manipulateurs à commande manuelle	8
Figure I.4: Manipulateur à cycle pré-réglé	9
Figure I.5: Robot programmable	9
Figure I.6: Robot intelligent	10
Figure I.7: Coordonnées cartésiennes et cylindriques	11
Figure I.8: Coordonnées polaires et universelles	11
Figure I.9: Robot SCARA	11
Figure I.10: Représentation du volume de travail	12
Figure I.11: Structure PPP	13
Figure I.12: Structure RPP (ou PRP)	13
Figure I.13: Structure RPR	13
Figure I.14: Structure RRP	13
Figure I.15: Structure RRR	14
Figure I.16: schéma du robot SCARA	16
Figure I.17: Sankyo	17
Figure I.18: robot Adept	17
Figure I.19: Le robot cylindrique	17
Figure I.20: Robot Seiko	17
Figure I.21: robot sphérique	18
Figure I.22 : ROBOT FANUC	18
Figure I.23 : robot Cartésien	18
Figure I.24 : robot Toshiba	18
Figure I.25 : robot COMAU	19
Figure I.26 : robot anthropomorphe	19
Figure I.27 : robot Kawasaki	19
Figure I.28 : robot ABB	20
Figure I.29: Robot soudeurs Par points	20
Figure I.30: Robot soudeurs a l'arc	20
Figure I.31: Robot pompiste	21
Figure I.32: Robot de construction	21
Figure I.33: Robot 'Computer motion'	21
Figure I.34: Robot pour Assistance aux personnes handicapées	21

Chapitre II : Description du bras manipulateur ROB3

Figure II.1 : Représentation schématique d'une chaîne cinématique	23
Figure II.2 : L'espace du travail du bras robotique ROB3	24
Figure II.3 : la pince du robot ROB3	25
Figure II.4 : unité de commande TEACHBOX	26
Figure II.5: Les axes du robot ROB3.....	27

Figure II.6 : Les dimensions du robot <i>ROB3</i> (vue de haut)	27
Figure II.7 : Les dimensions du robot <i>ROB3</i> (vue de face)	27
Figure II.8 Architecture d'un MCC à aimant permanent	28
Figure II.9: Moteur DC à excitation séparée	29
Figure II.10 : les pignons	30
Figure II.11 : Le capteur du robot <i>ROB3</i>	30
Figure II.12 Photo de <i>ROB3</i> avec son alimentation	31

Chapitre III : Le PIC 16F877A

Figure III.1: Boitier du PIC 16F877A	34
Figure III.2 : Architecture interne du pic 16F877A	37
Figure III.3 : Schéma du Timer 0	40
Figure III.4: Schéma descriptive du timer 1	40
Figure III.5 : Schéma du Timer 2	40
Figure III.6 : Les différents PORT de PIC16F877A	42
Figure III.7 : Schéma fonctionnel analogique numérique	43

Chapitre VI : Réalisation hardware et software

Figure IV.1 : Schéma simplifié du système de commande	47
Figure IV.2 : L'alimentation du ROB3 (9V 3A)	48
Figure IV.3 : Schéma de la carte de commande	48
Figure IV.4 circuit oscillant du PIC	50
Figure IV.5 Le 'driver / receiver ' MAX232	50
Figure IV.6 : Trame de communication série	52
Figure IV.7: Le connecteur <i>DB9</i> mâle	52
Figure IV.8 : Le circuit et diagramme logique du <i>74HC138</i>	54
Figure IV.9 : Schéma d'utilisation du circuit <i>74HC138</i>	54
Figure IV.10 : Circuit et diagramme du <i>74HC04</i>	54
Figure IV.11 : Schéma d'utilisation du circuit <i>74HC04</i>	55
Figure IV.12 : Schéma de la carte de commande	56
Figure IV.13 Le circuit <i>L298N</i>	56
Figure IV.14: <i>Diagramme bloqué du L298</i>	56
Figure IV.15: <i>Brochage du circuit intégré L298</i>	57
Figure IV.16 : Diode <i>1N4007</i>	58
Figure IV.17 : Fusible et son support.....	58
Figure IV.18. Circuit d'alimentation.....	59
Figure IV.19: Schéma de principe du programmeur.....	60
Figure IV.20 Schéma du programmeur.....	60
Figure IV.21 : schéma représentant un signal PWM	61
Figure IV.22: Organigramme de fonctionnement.....	62
Figure IV.23: carte de commande	63
Figure IV.24: carte de puissance	63
Figure IV.25: Interface principale.....	64
Figure IV.26: Interface de commande manuelle.....	65
Figure IV.27: Interface de commande angulaire.....	65

Figure IV.28: Organigramme de commande.....	66
---	----

Chapitre V : Software

Figure V.1: Interface utilisateur ISIS	69
Figure V.2: Organisation de la boîte à outils isis	70
Figure V.3: organigramme des Etapes de la saisie de schéma.....	71
Figure V.4: fenêtre principale ARES	72
Figure V.5: fenêtre de placement automatique	73
Figure V.6: fenêtre de routage automatique.....	74
Figure V.6: fenêtre principale de pic c compiler.....	75
Figure V.7: Lancement de PIC C Compiler	75
Figure V.8: Création d'un nouveau projet	75
Figure V.9: Choix du dossier de travail et du nom du projet.....	76
Figure V.10: entête de programme source	76
Figure V.11: paramètres du projet	76
Figure V.12: exemple de programme	77
Figure V.13: Compilation de programme	77
Figure V.14: fenêtre de compilation.....	77
Figure V.15:fenêtre principale de icprog	78
Figure V.16:fenêtre de configuration hardware	78
Figure V.17:fenêtre sélection de composant	78
Figure V.18:programmer un composant.	79
Figure V.19 : L'interface de C++ Builder.....	80

Introduction générale

Depuis le milieu des années 70, la robotique est devenue une science extrêmement populaire dans les milieux universitaire. Alliant un grand intérêt pédagogique et industriel, Cette nouvelle science demande beaucoup de créativité et des connaissances pluridisciplinaires (Mécanique, Electronique numérique et analogique, électrotechnique, Programmation, Intelligence artificielle, Temps réel, Automatique.....)

Nous avons choisi la réalisation de la carte de commande d'un bras manipulateur (ROB03) qui est disponible au niveau du laboratoire du département de génie électrique à l'université de Mohamed Khider Biskra et qui n'a pas fonctionner depuis l'année 1991 à cause de la panne sur la de commande originale qu'elle était à base de microprocesseur.

Les sujets traités dans ce travail concernent principalement des études théoriques (mécaniques, capteurs) sur bras ROB03, études théorique et pratique sur le microcontrôleur PIC16F877A et les circuits intégrés utilisés (hacheurs, max232,...). Après la réalisation de les cartes (commande et puissance), et on a programmé finalement le microcontrôleur. Enfin on a créé un interface graphique sur le micro-ordinateur.

Notons que la commande sera par PC ce qui nous donne la possibilité de modifier et développer l'interface suivant la méthode d'asservissement et de commande qu'on la veut utiliser , et le signal pwm généré par le microcontrôleur permet de commander la tensien des moteurs sans de changer l'amplitude et la fréquence ce qui nous permet aussi d'appliquer les différents lois de commande (PID, réseau de neurones, logique floue,...).

Organisation du mémoire :

Ce travail est divisé en cinq chapitres :

Dans le premier chapitre, nous commençant par une généralité sur les robots (ses constituants et classifications).

Le deuxième chapitre est consacré à la description du bras ROB03 (mécanique, électrique)

Dans le troisième chapitre, nous présentons le cœur de la carte (le microcontrôleur).

Le quatrième chapitre est consacré à la réalisation hardware et software et les autres composants constituant la carte.

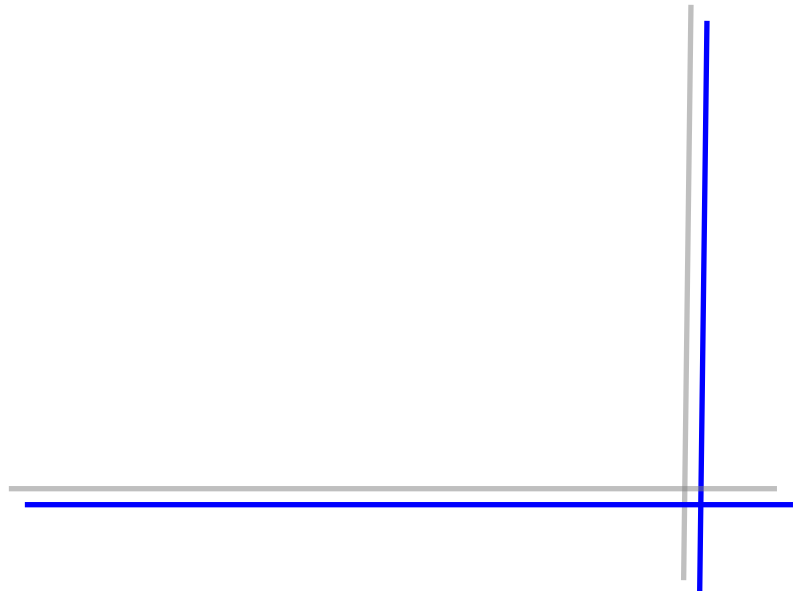
Enfin le contenu du cinquième chapitre présente les outils software que nous avons mis en œuvre pendant la réalisation des circuits et la programmation du PIC16F877A.

Nous terminons ce mémoire par une conclusion et quelques perspectives.



Chapitre 1

Généralité sur les robots



I.1 Introduction

La robotique est un ensemble de disciplines techniques (mécanique, électronique, automatique, informatique) articulées autour d'un objectif et d'un objet communs. Cet objectif est l'automatisation flexible de nombreux secteurs de l'activité humaine réputés jusqu'à très récemment comme ne pouvant se passer de la présence de l'homme, et l'objet est le robot, sorte de machine universelle dont l'homme rêve depuis toujours (mais qui n'a pas encore la forme de ce rêve !).

Historiquement, le terme «robot» a été introduit en 1920 par l'écrivain tchèque *Karel Čapek* dans sa pièce de théâtre *RUR (Rossum's Universal Robots)*. Ce terme, provenant du tchèque robot signifie «travail forcé », désigne à l'origine une machine androïde capable de remplacer l'être humain dans toutes ses tâches. Ensuite, dans les années quarante, les progrès de l'électronique permettent de miniaturiser les circuits électriques (inventions des transistors et circuits intégrés), ouvrant ainsi de nouvelles horizons à la fabrication de robots. Dans les premiers temps de la robotique, le robot est considéré comme une imitation de l'homme, aussi bien fonctionnelle que physique. Aujourd'hui, les constructeurs ne tentent plus de reproduire l'aspect humain sur un robot, privilégiant avant tout sa fonctionnalité.

Actuellement, les robots sont très répandus dans le secteur industriel, notamment en construction automobile et chez la plupart des fabricants d'ordinateurs. Leurs capacités d'effectuer rapidement des travaux répétitifs ne cessent de croître. Ils sont notamment utilisés dans les chaînes de montage et de fabrication. On les emploie également dans des environnements difficilement supportables par l'homme caractérisés par des conditions extrêmes de température ou de pression, radioactivité élevée, etc... L'industrie du nucléaire a ainsi largement contribué au développement de la robotique, notamment dans la conception de bras télémanipulateurs.

I.2 Définitions [5]

La définition que l'on donne actuellement du robot industriel diffère quelque peu selon les pays.

Le Petit Larousse définit un robot comme étant un appareil automatique capable de manipuler des objets, ou d'exécuter des opérations selon un programme fixe ou modifiable.

C'est au Japon que la définition est la plus vague. Où le rôle du robot y est essentiellement de servir d'intermédiaire entre l'homme et la machine. On le qualifie de :

« Tout mécanisme permettant d'effectuer, en tout ou en partie, une tâche normalement réalisée par l'homme ».

Le rôle du robot y est essentiellement de servir d'intermédiaire entre l'homme et la machine. Il permet de changer le système de production d'un système à interaction directe entre la machine et l'homme en un système où l'homme gère la machine par l'intermédiaire du robot. Le peu de précision de la définition

japonaise du robot a une incidence immédiate sur les statistiques correspondantes :

Le *JIRA (Japan Industrial Robot Association)* a recensé en 1981 77000 robots industriels dans l'industrie japonaise. Toutefois, plus de 50% de ces robots sont des manipulateurs à séquence fixe qui ne seraient pas recensés comme tels aux USA ou en France, où la définition d'un robot est beaucoup plus restrictive.

La définition américaine du robot (*Robot Institute of America*) est beaucoup plus spécifique :

« Un robot est un manipulateur reprogrammable à fonctions multiples. Il est conçu pour déplacer des matériaux, des pièces, des outils ou des instruments spécialisés suivant des trajectoires variables programmées, en vue d'accomplir des tâches très diverses ».

C'est la définition de l'Association Française de Robotique Industrielle (*A.F.R.I.*) qui est la plus explicite, et aussi la plus proche de la technologie actuelle des robots :

« Un robot industriel est une machine formée de divers mécanismes comportant divers degrés de liberté, ayant souvent l'apparence d'un ou de plusieurs bras se terminant par un poignet capable de maintenir un outil, une pièce ou un instrument de contrôle. En particulier, son unité de contrôle doit contenir un système de mémorisation, et il peut parfois utiliser des accessoires sensitifs et adaptables qui tiennent compte de l'environnement et des circonstances. Ces machines, ayant un rôle pluridisciplinaire, sont généralement conçues pour effectuer des fonctions répétitives, mais sont adaptables à d'autres fonctions ».

En plus de la définition américaine, la définition française, du fait qu'elle envisage la perception de l'environnement par le robot, implique une certaine prise de décision. Elle annonce la génération des robots dits « intelligents ».

I.3 Constituants d'un robot [6]

la Figure I.1 représente les vocabulaires du robot :

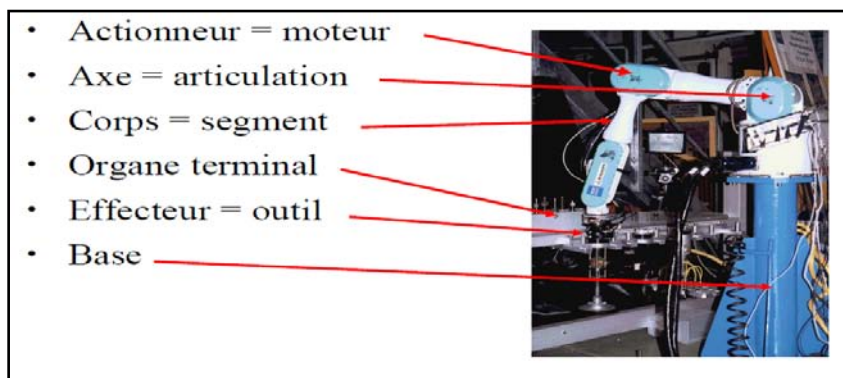


Figure I.1: vocabulaire du robot

On distingue classiquement 4 parties principales dans un robot manipulateur :

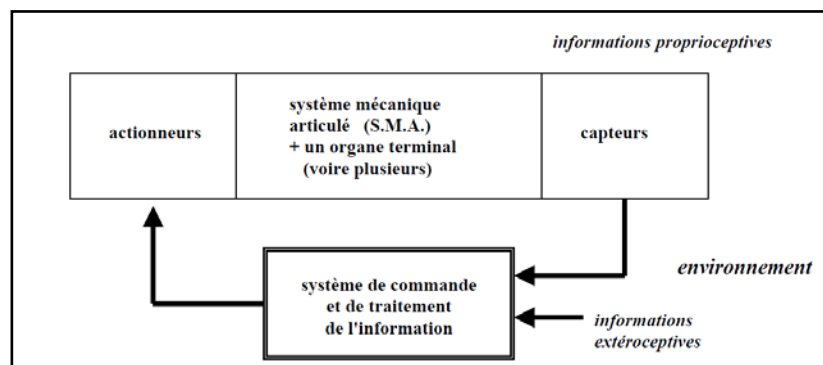


Figure I.2: parties principales dans un robot

-Organe terminal : tout dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, à dépression, ...), ou à les transformer (outils, torche de soudage, pistolet de peinture, ...). En d'autres termes, il s'agit d'une interface permettant au robot d'interagir avec son environnement. Un organe terminal peut être multifonctionnel, au sens où il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel, mais interchangeable. Un robot, enfin, peut-être multi-bras, chacun des bras portant un organe terminal différent. On utilisera indifféremment le terme organe terminal, préhenseur, outil ou effecteur pour nommer le dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique.

-Le système mécanique articulé (S.M.A.) : est un mécanisme ayant une structure plus ou moins proche de celle du bras humain. Il permet de remplacer, ou de prolonger, son action (le terme "manipulateur" exclut implicitement les robots mobiles autonomes). Son rôle est d'amener l'organe terminal dans une situation (position et orientation) donnée, selon des caractéristiques de vitesse et d'accélération données. Son architecture est une chaîne cinématique de corps, généralement rigides (ou supposés comme tels), assemblés par des liaisons appelées articulations. Sa motorisation est réalisée par des actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations par des systèmes appropriés.

- Les actionneurs : le S.M.A. comporte des moteurs le plus souvent avec des transmissions (courroies crantées), l'ensemble constitue les actionneurs. Les actionneurs utilisent fréquemment des moteurs électriques à aimant permanent, à courant continu, à commande par l'induit (la tension n'est continue qu'en moyenne car en général l'alimentation est un hacheur de tension à fréquence élevée ; bien souvent la vitesse de régime élevée du moteur fait qu'il est suivi d'un réducteur, ce qui permet d'amplifier le couple moteur). On trouve de plus en plus de moteurs à commutation électronique (sans balais), ou, pour de petits robots, des moteurs pas à pas. Pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique), les actionneurs sont le plus souvent hydrauliques, agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique).

Les actionneurs *pneumatiques* sont d'un usage général pour les *manipulateurs à cycles (robots tout ou rien)*. Un manipulateur à cycles est un S.M.A. avec un nombre limité de degrés de liberté permettant une succession de mouvements contrôlés uniquement par des capteurs de fin de course réglables manuellement à la course désirée (asservissement en position difficile dû à la compressibilité de l'air).

-Les capteurs : Les organes de perception permettent de gérer les relations entre le robot et son environnement. Les capteurs Dits *proprioceptifs* lorsqu'ils mesurent l'état interne du robot (positions et vitesses des articulations) et *extéroceptifs* lorsqu'ils recueillent des informations sur l'environnement (détection de présence, de contact, mesure de distance, vision artificielle).

-La partie commande : synthétise les consignes des asservissements pilotant les actionneurs, à partir de la fonction de perception et des ordres de l'utilisateur.

S'ajoutent à cela :

- L'interface homme-machine à travers laquelle l'utilisateur programme les tâches que le robot doit exécuter,
- Le poste de travail, ou l'environnement dans lequel évolue le robot

I.4 Classification des robots [5]

On peut classer les robots d'un point de vue fonctionnel ou d'après leur structure géométrique.

I.4.1 Classification fonctionnelle

Le nombre de classe et les distinctions entre celles-ci varient de pays à pays (6 classes au Japon, 4 en France). L'A.F.R.I. distingue 4 classes illustrées ci-dessous :

I.4.1.a manipulateurs à commande manuelle ou télécommande.

La Figure I.3 représente les manipulateurs à commande manuelle :

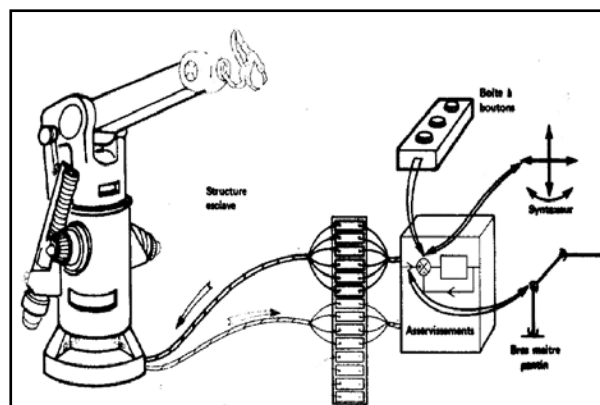


Figure I.3: manipulateurs à commande manuelle

I.4.1.b manipulateurs automatiques à cycles pré-réglés :

Le réglage se fait mécaniquement par cames, butées comme le montre la Figure I.4, la commande peut se faire par automate programmable on peut distinguer entre manipulateurs à cycle fixe et manipulateurs à cycle programmable.

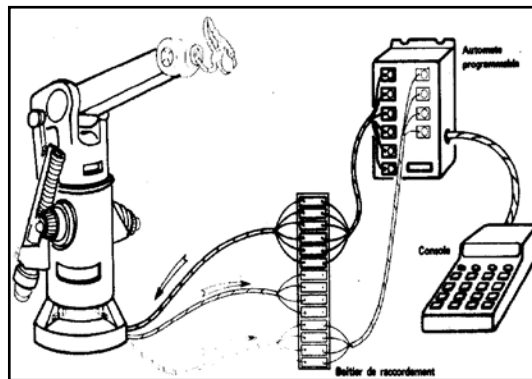


Figure I.4: Manipulateur à cycle pré-réglé

I.4.1.c robots programmables :

C'est la première génération de robots industriels ; ils répètent les mouvements qu'on leur a appris ou programmés, sans informations sur l'environnement ou la tâche effectuée. On peut aussi faire la distinction entre robots « play-back » qui reproduisent la tâche apprise et robots à commande numérique qui peuvent être programmés hors-ligne. Pour de nombreux robots, l'apprentissage de la tâche se fait à l'aide d'un « syntaxeur » (« boîte à boutons », « teach pendant ») qui permet à un opérateur d'amener le robot en un certain nombre de points, qui sont ensuite mémorisés, lors de l'exécution de la tâche, le robot suivra une trajectoire passant successivement par tous les points programmés, le passage d'un point au suivant se faisant suivant un profil de vitesse en fonction du temps qui est prédéfini (triangulaire ou trapézoïdal), l'opérateur n'ayant qu'à choisir la fraction de la vitesse maximum à laquelle il souhaite que le robot effectue la tâche. Pour certains robots, par exemple les robots de peinture, qui doivent suivre une trajectoire complexe qu'il est difficile d'exprimer mathématiquement, un opérateur humain spécialiste de la tâche effectue la trajectoire en guidant le bras du robot à l'aide d'un « pantin », et l'entièreté de la trajectoire est mémorisée par le robot et la figure I.5 représente les robots programmable :

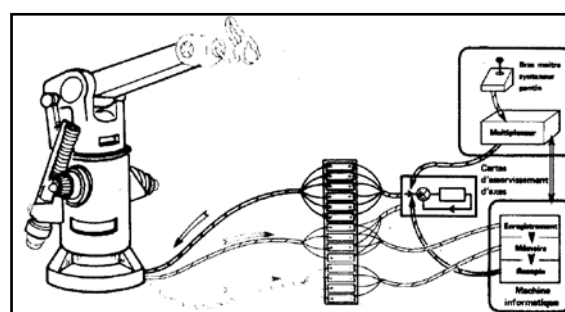


Figure I.5: Robot programmable

I.4.1.d robots intelligents :

On trouve actuellement des robots de seconde génération qui sont capables d'acquérir et d'utiliser certaines informations sur leur environnement (systèmes de vision, détecteurs de proximité, capteurs d'efforts,...) comme le montre la Figure I.6. On étudie des robots de troisième génération, capables de comprendre un langage oral proche du langage naturel et de se débrouiller de façon autonome dans un environnement complexe, grâce à l'utilisation de l'intelligence artificielle.

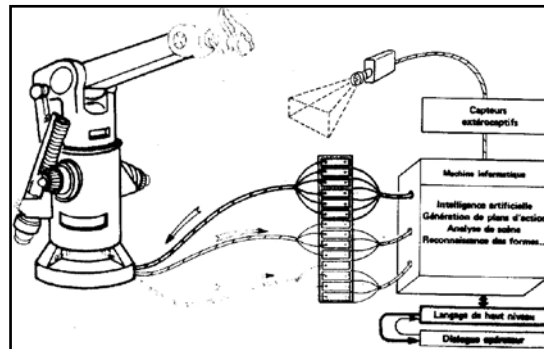


Figure I.6: Robot intelligent

I.4.2 Classification géométrique

On peut aussi classer les robots suivant leur configuration géométrique, autrement dit l'architecture de leur porteur. Les 3 premiers ddm d'un robot peuvent être réalisés avec un grand nombre de combinaisons de translations (max. 3T) et de rotations (max. 3R), autrement dit par des articulations **prismatiques** (P) ou **rotoïdes** (R) ; en pratique, on n'utilise que 4 ou 5 d'entre elles :

- porteur cartésien (TTT ou PPP) : les 3 axes sont animés d'un mouvement de translation.
- porteur en coordonnées cylindriques (RTT ou RPP) : un mouvement de rotation et une translation axiale, complétées par une translation radiale³(voir figure I.7).
- porteur en coordonnées polaires ou sphériques (RRT ou RRP) : deux rotations (longitude et latitude) autour d'axes orthogonaux, complétées par une translation radiale comme le montre la figure I.8.
- porteur en coordonnées universelles, appelé aussi configuration poly articulée ou anthropomorphe (RRR), trois rotations dont les deux dernières se font autour d'axes parallèles orthogonaux au premier, les trois articulations correspondant respectivement au tronc (base), à l'épaule et au coude d'un être humain.

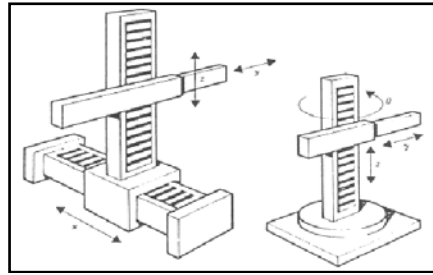


Figure I.7: Coordonnées cartésiennes et cylindriques

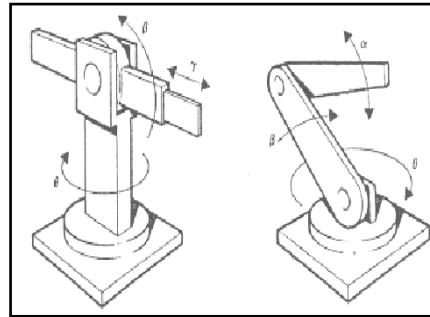


Figure I.8: Coordonnées polaires et universelles

Une cinquième architecture comprend deux rotations autour de deux axes parallèles, précédées ou suivies d'une translation dans la même direction (éventuellement celle-ci peut être reportée au niveau du poignet, qui peut aussi tourner autour du même axe, soit au total 4 ddm). Cette architecture est celle des robots SCARA (Selective Compliance Arm for Robotic Assembly) que l'on utilise dans des opérations d'assemblage et la figure I.9 représente le robot scara.



Figure I.9: Robot SCARA

I.5. Performances des robots [5]

Lorsqu'on désire fabriquer un appareil, une machine ou un équipement quelconque, l'utilisateur ou celui qui passe commande présente un cahier des charges. Ce cahier des charges définit les contraintes d'exploitation et il est souvent aisé de mesurer les performances de l'appareillage réalisé car elles s'expriment vis-à-vis de ces contraintes. Celles-ci peuvent être le poids et l'indépendance énergétique pour une machine transportable, le respect de la précision, la vitesse d'exécution, la fiabilité, etc... Pour d'autres machines.

On a vu qu'une des caractéristiques d'un robot est la versatilité, c'est-à-dire cette possibilité de faire des tâches diverses mais dont on ne sait pas dire lesquelles à l'avance. Comme on ignore à priori les utilisations du robot, on peut difficilement connaître les paramètres de référence permettant d'établir les performances. Certains se sont lancés dans des énumérations de qualité impliquant plusieurs centaines de paramètres si bien qu'il est impossible de s'y retrouver.

I.5.1 Les performances-tâches

Un robot appliquant des efforts et des déplacements à son organe terminal, on peut en cerner les caractéristiques qui sont importantes en pratique.

I.5.1.a Le volume atteignable (volume de travail)

Le volume de travail en position d'un robot manipulateur est défini comme l'espace physique engendré par un point de l'organe terminal lorsque la configuration du robot évolue. Il s'exprime en unités volumiques, mais la forme de son enveloppe (qui peut-être compliquée puisque formée par la combinaison des mouvements de plusieurs articulations) est aussi importante (voir figure I.10). Il est plus simple et plus parlant de la décrire par une surface inscrite au vrai volume mais de forme simple (par exemple intersection de sphères ou des parallélépipèdes ou des ellipsoïdes). Sinon il faut une représentation graphique. Le volume de travail est alors habituellement représenté par deux sections perpendiculaires choisies en fonction du type de robot manipulateur, cette représentation étant préférable à une seule vue en perspective.

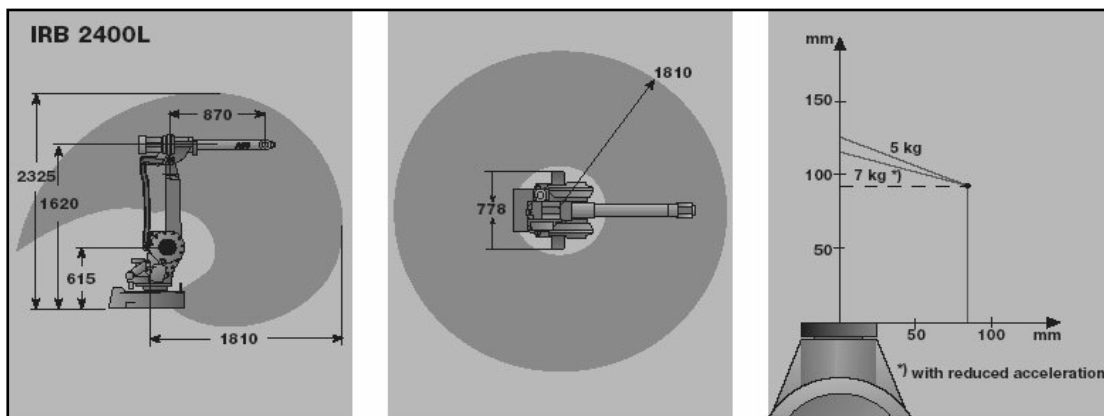


Figure I.10: Représentation du volume de travail

Quelques exemples sur les volumes de travail du robot :

- **robot PPP** : Le volume de travail est un parallélépipède dont les dimensions sont les translations permises par les trois liaisons prismatiques, d'où :

$$V = L1L2L3$$

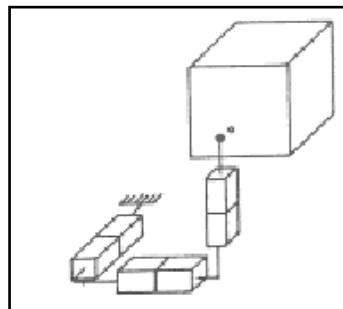


Figure I.11: Structure PPP

➤ robots RPP et RPR ou RRP

Le volume de travail est un cylindre plein ou creux, autrement dit un tore à section rectangulaire, dont la hauteur L est la translation permise par une liaison prismatique et dont les rayons intérieur et extérieur, Ri et Re, sont fixés soit par la disposition d'une liaison prismatique et la translation qu'elle permet, soit par les longueurs des deux parties du bras :

$$V = \pi L (R_e^2 - R_i^2)$$

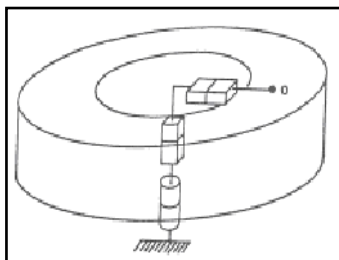


Figure I.12: Structure RPP (ou PRP)

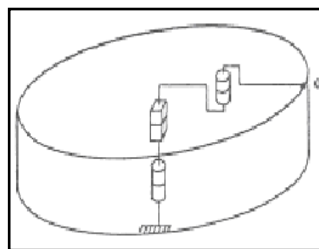


Figure I.13: Structure RPR

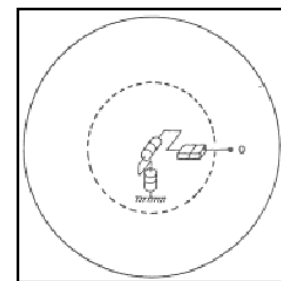


Figure I.14: Structure RRP

➤ robots RRP et RRR

Le volume de travail est une sphère creuse, dont les rayons intérieur et extérieur, Ri et Re, sont fixés soit par la disposition de la liaison prismatique et la translation qu'elle permet, soit par les longueurs des deux parties du bras :

$$V = \frac{4}{3} \pi (R_e^3 - R_i^3)$$

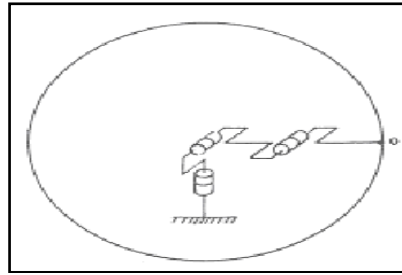


Figure I.15: Structure RRR

I.5.1.b Charge utile

C'est la charge maximale que peut porter le robot sans dégrader la répétabilité et les performances dynamiques.

La charge utile est nettement inférieure à la charge maximale que peut porter le robot qui est directement dépendante des actionneurs.

I.5.1.c Précision / Répétabilité

- La précision du robot manipulateur est définie par l'écart entre la situation programmée et la situation moyenne. Elle caractérise l'aptitude du robot à situer l'organe terminal en conformité avec la situation programmée.
- Répétabilité : la répétabilité d'un robot est l'erreur maximale de positionnement répété de l'outil en tout point de son espace de travail.
- En général, la répétabilité < 0.1 mm.

I.5.1.d La vitesse

C'est évidemment une caractéristique fondamentale pour les robots industriels, puisqu'elle déterminera les temps nécessaires à l'exécution d'une tâche. Or, ce sont ceux—ci qui intéressent essentiellement l'utilisateur. Dans l'évaluation des performances à priori, on ignore naturellement ces tâches. Il faut donc s'en tenir à des vitesses de déplacement et de rotation de l'organe terminal.

Les constructeurs sont assez évasifs sur les performances en vitesse de leurs robots. On parle d'ordre de grandeur ou de vitesse maximum, mais toujours pour des translations de l'organe terminal (1 à 2 m/s). On donne aussi souvent les vitesses de rotations articulaires maximales (1 rad/s à 1 tour/s), mais il est difficile d'en déduire la vitesse instantanée dans l'espace de la tâche, qui est la grandeur que l'utilisateur attend.

I.5.1.e L'orientation possible de l'organe terminal

De par la présence de butées sur chaque articulation, et aussi parfois de par l'existence de couplages mécaniques entre articulations, il n'est pas toujours possible d'atteindre le

débattement maximum sur chacun des degrés de mobilité de l'organe terminal dans tout le volume atteignable.

Il y aurait donc lieu, soit de donner les débattements angulaires minimaux de l'organe terminal par rapport au repère fixe dans tout le volume atteignable, soit de fournir une cartographie de ces débattements. (Ceci serait plus utile que la présentation des constructeurs, qui donnent le débattement de chaque articulation à partir d'une position de référence, en négligeant les problèmes de butées et de couplages qui modifient quelquefois de façon non négligeable les possibilités du mécanisme).

I.5.1.f La fiabilité

Bien qu'elle mette en jeu tous les éléments présents dans le robot, c'est une donnée qui intéresse au plus haut point l'utilisateur, même s'il ignore tout du système. C'est pourquoi on peut la classer dans les performances-tâche.

Comme pour tout autre système, elle sera définie par un taux de pannes, celui-ci étant exprimé soit par la fraction du temps durant laquelle le robot ne remplit pas la fonction qui lui est assignée, soit par un MTBF (Mean Time Between Failures). On exprime aussi souvent cette fiabilité à l'aide d'un diagramme fréquentiel (fréquence de pannes) tout au long de la vie du système.

Le robot peut être sujet à deux grandes catégories de pannes

1. arrêt total du fonctionnement
2. dégradation des performances. Par exemple sur la précision spatiale : une dérive plus ou moins lente peut se produire et, après un certain temps, le robot n'exécute plus la tâche. Ou bien un degré de mobilité ne fonctionne plus (il serait intéressant alors d'avoir une redondance des degrés de mobilité qui permette par une procédure automatique de compenser la défaillance en attendant de pouvoir réparer le robot).

I.5 .2 Les performances-homme

On entend par là la facilité d'utilisation du robot, ce dernier étant sur le site d'utilisation et entre les mains d'un opérateur non spécialiste. Les compétences exigées pour faire fonctionner un robot ont trait :

- à la connaissance du matériel réglage, etc.
- à la connaissance du logiciel programmation du robot.

I.5.3 Les performances économiques

Il s'agit essentiellement de la rentabilité d'utilisation de robots dans l'entreprise. Elle dépend bien sûr d'un grand nombre de facteurs :

1. coûts d'investissements :

- prix d'achat
- coût d'installation du robot et des équipements de péri-robotique
- coût de modification de l'atelier ou de la chaîne de production.

2. coûts d'exploitation

- énergies, maintenance, espérance de vie, fiabilité (doit-on laisser une possibilité de travail manuel en cas de panne ?) et pourcentage temporel d'utilisation.

3. nature des tâches que les robots vont exécuter

- tâches qui revenaient cher en main-d'œuvre ou non ?
- la cadence de la fabrication augmentera-t-elle ?
- la qualité de la production sera-t-elle améliorée ?
- y aura-t-il réduction de la main-d'œuvre ? (supprime-t-on un poste d'O.S. pour le remplacer par un poste de technicien ?)

4. le degré d'automatisation déjà présent dans l'atelier

- Insérer un robot entre deux hommes ou réciproquement ne se justifie que dans des circonstances particulières, par exemple si la tâche est dangereuse

5. la taille de l'entreprise

- Un robot seul peut ne pas être "rentable" mais une chaîne de plusieurs robots en ligne peut l'être.

Le constructeur de robots ne peut que proposer des robots "les plus performants possibles" et au "meilleur prix", en espérant atteindre le bon créneau après une étude de marché. C'est à l'utilisateur d'établir les performances économiques qu'il peut espérer, après avoir défini les tâches qu'il veut automatiser et les performances tâches et homme que les constructeurs peuvent lui proposer.

I.6. Les différents types des robots [7]**I.6.1 Les robots SCARA**

SCARA = Selective Compliance Articulated Robot for Assembly.

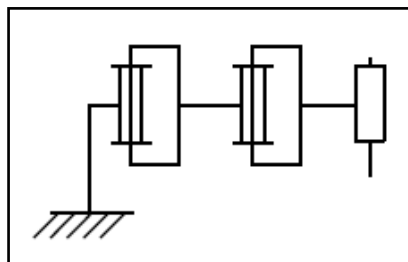


Figure I.16:schéma de Les robots SCARA

Caractéristiques :

- 3 axes, série, RRP, 3 DDL.
- Espace de travail cylindrique.
- Précis.
- Très rapide.

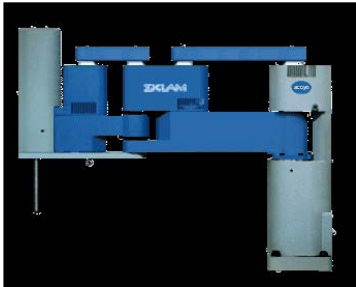
Exemples :

Figure I.17: Robot Sankyo



Figure I.18: robot Adept

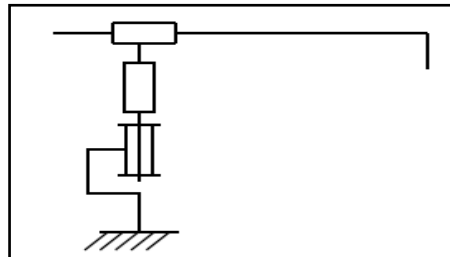
I.6.2 Les robots cylindriques

Figure I.19: Le robot cylindrique

Caractéristiques :

- 3 axes, série, RPP, 3 DDL.
- Espace de travail cylindrique.
- Très rapide.

Exemples :

Figure I.20: Robot Seiko

I.6.3 Les robots sphériques

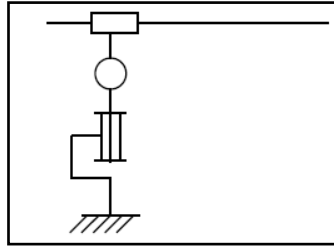


Figure I.21 : robot sphérique

Caractéristiques :

- 3 axes, série, RRT, 3 DDL.
- Espace de travail sphérique.
- Grande charge utile.

Exemples :



Figure I.22 :ROBOT FANUC

I.6.4 Les robots Cartésiens

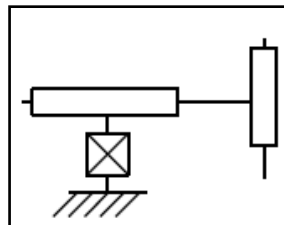


Figure I.23 : robot Cartésien

Caractéristiques :

- 3 axes \perp 2 à 2, série, PPP, 3 DDL.
- Très bonne précision.
- Lent.

Exemples :



Figure I.24 : robot Toshiba

I.6.5 Les robots parallèles

Caractéristiques :

- Plusieurs chaînes cinématiques en parallèle.
- Espace de travail réduit.
- Précis (grande rigidité de la structure).
- Rapide.

Exemples :



Figure I.25 : robot COMAU

I.6.6 Les robots anthropomorphe

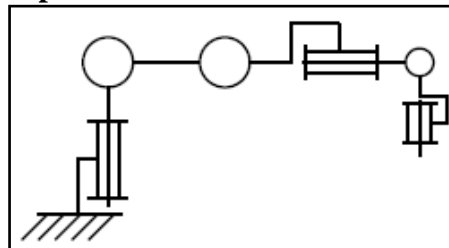


Figure I.26 : robot anthropomorphe

Caractéristiques :

- Reproduisent la structure d'un bras humain.
- 6 axes, série, 6R, 6 DDL.

Exemples :

- Architecture standard :



Figure I.27 : robot Kawasaki

- Architecture à parallélogramme:

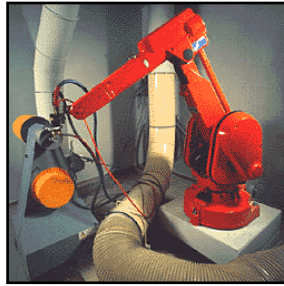


Figure I.28 : robot ABB

I.7. Utilisation des robots [11]

I.7.1. Tâches simples

- La grande majorité des robots est utilisée pour des tâches simples et répétitives.
- Les robots sont programmés une fois pour toute au cours de la procédure *d'apprentissage*.

-Critères de choix de la solution robotique:

- La tâche est assez simple pour être robotisée.
- Les critères de qualité sur la tâche sont importants.
- Pénibilité de la tâche (peinture, charge lourde, environnement hostile, ...).

Exemples :

- **Robot soudeurs :**

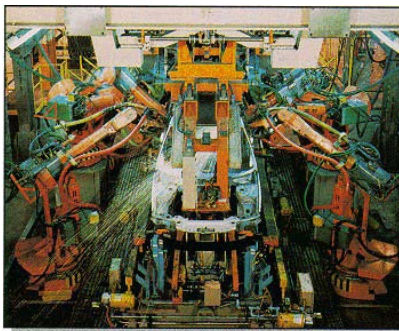


Figure I.29: Robot soudeurs Par points



Figure I.30: Robot soudeurs a l'arc

I.7.2. Tâches complexes

- Robotique de service :



Figure I.31: Robot pompiste



Figure I.32: Robot de construction



Figure I.33: robot Computer motion



Figure I.34: robot Assistance aux personnes Handicapées

I.8. Avenir de la robotique [7]

- Stagnation du nombre de robots utilisés pour des tâches simples.
- Forte croissance du nombre des robots utilisés pour des tâches complexes :
 - robotique de service
 - robotique d'assistance aux manipulations dans la recherche biologique et génétique.
 - robotique médicale.

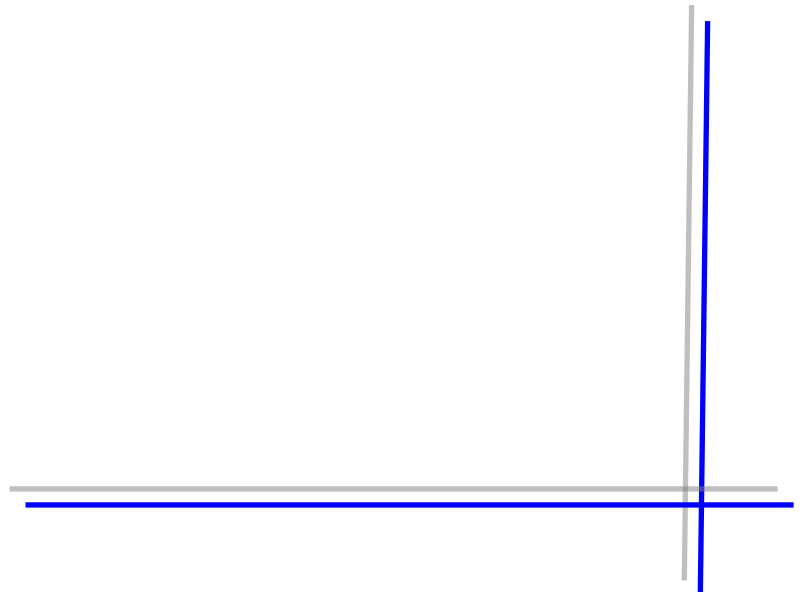
I.9. Conclusion

Le présent chapitre fait l'objet d'une description sur le domaine de la robotique en mettant l'accent sur les domaines applicatifs des robots industriels. Ensuite la classification des robots est abordée. Une description détaillée du bras manipulateur didacticiel *ROB3* fait l'objet du chapitre suivant.



Chapitre 2

Description du bras manipulateur ROB03



II.1 Introduction

Le ROB3 est un robot de haute qualité de formation, particulièrement conçu pour l'usage dans toutes les sphères d'éducation et de formation de robotique. Le bras articulé de ROB3 a cinq axes et une pince, qui sont actionnées par les servomoteurs à courant continue. La position absolue de tous les axes est déterminée par les capteurs de position rotatoires potentiométriques.

II.2 Description cinématique du bras manipulateur

La figure II.1 présente un exemple d'une représentation schématique d'une chaîne cinématique d'un robot à 2 axes. Le tableau II.1 présente la signification des différents éléments de la figure II.1 Pour une classification des types des robots, la cinétique des axes principaux est déterminante.

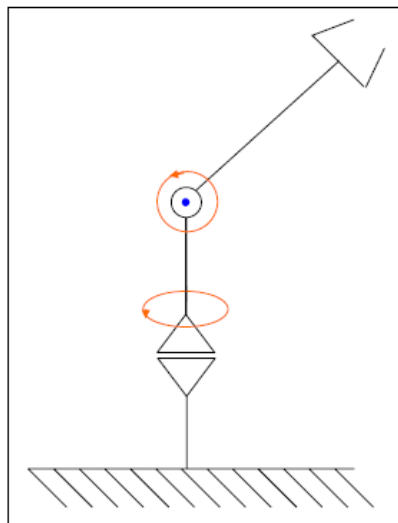


Figure II.1 : Représentation schématique d'une chaîne cinématique.

Nom	Symbole
Axe prismatique	
Axe rotoïde	
Outil actionneur	
Joint	

Tableau II.1 : Table des symboles.

II.3. Espace du travail du manipulateur ROB3 [10]

Figure II.2 de la zone de travail (l'anglais est « plage » de travail) du bras robot décrit. Si un objet extérieur à bout de ce Région, il ne peut pas être atteint. Lieu où le robot avec ses pince peut atteindre, sont à la pointe centre de l'outil (TCP) est décrite.

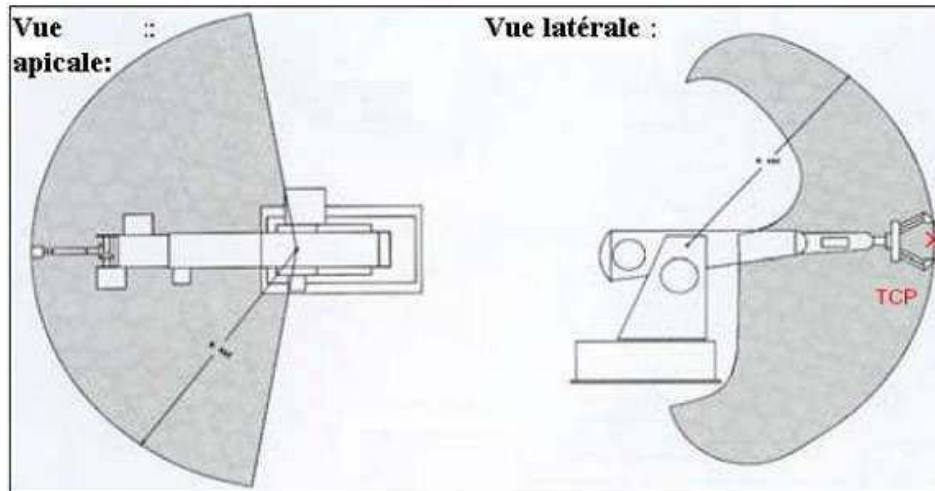


Figure II.2 : L'espace du travail du bras robotique ROB3.

II.4 Propriétés du bras robot ROB3

II.4.1. Propriétés électrique

Le bras du robot ROB3 possède un système de contrôle intégré qui peut être programmé directement à travers un clavier «l'unité TEACHBOX». Le bras est configuré en introduisant une séquence d'instructions pour faire bouger le bras vers des positions désirées, ces instructions sont mémorisées dans une mémoire qui représente un programme de commande. Notons que ROB3 peut aussi être commandé par un PC via une interface série RS232.

Le contrôleur est intégré dans la base du ROB3. Un processeur 8031 constitue le cœur du contrôleur.

Les programmes utilisés peuvent être stockés dans une mémoire auxiliaire. La capacité de mémoire est de 8 ko, ce qui correspond à environ 1000 instructions. En outre, le bras peut être connecté avec un PC via l'interface série RS 232. Par contre, 8 entrées/sorties digitales sont à la disposition de l'utilisateur par l'intermédiaire d'un connecteur port parallèle.[1]

La pince est alimentée par un servomoteur à courant continu par le biais d'un câble comme le montre la Figure II.3. De très minces pièces peuvent être manipulées par l'outil. Le manipulateur ROB3 fait partie des robots de hautes qualités qui a démontré sa fiabilité durant plusieurs années, dans tous les domaines de l'éducation et de la formation. Bien que l'objet

d'un processus continu d'amélioration de la fiabilité du modèle, 5 axes rotatifs et une pince parallèle, plusieurs possibilités d'applications peuvent être réalisées.

Comme le contrôleur est intégré dans la base du robot, un accès facile et rapide d'installation et la mise en œuvre peut être assurée avec une facilité d'utilisation

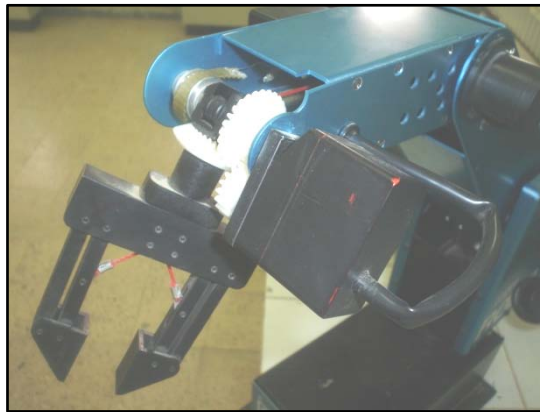


Figure II.3 : la pince du robot ROB3.

II.4.2. L'unité *TEACHBOX*

Le *TEACHBOX* (voir Figure II.4) permet de contrôler et programmer le *ROB3* sans utiliser un ordinateur, en utilisant un clavier digital avec 25 boutons, répartis en trois groupes :

- Boutons numériques qui sont utilisés pour introduire le numéro de l'axe désigné et la valeur de la position sous forme de valeur décimale,
- Boutons de commandes : pour indiquer la commande à exécuter directement (*RUN*, *STOP*, *INS*, *DEL*, *POS*, *OUT*),
- Boutons d'instructions : pour implanter le programme d'instructions de commande, (*POS*, *TIM*, *OUT*, *MARK*, *IF*, *GOTO*, *NOP*). Une fois l'instruction est introduite, elle est sauvegardée dans la mémoire du robot par ordre d'entrée (*FIFO*) et elles sont exécutées en mode *RUN* ou *STEP*. [10]

Pour programmer le bras il suffit de connecter le *ROB3* avec le *TEACHBOX* via le câble série *RS232*.

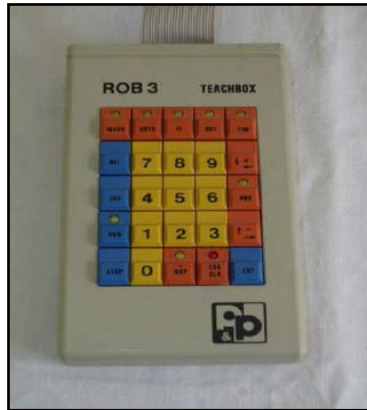


Figure II.4 : unité de commande *TEACHBOX*.

II.4.3. Limite en commande

Le bras *ROB3* a un système de contrôle intégré à base d'un microprocesseur Intel 8031. Ce dernier permet de commander le manipulateur uniquement en boucle ouverte. Dans le cadre de notre travail, nous sommes rendu compte que la carte de commande originale était défectueuse notamment le programme de commande implanté dans l'*EPROM* est effacé suite à l'enlèvement du papier de protection, outre l'absence des références des différents circuits électroniques (composants) constituant notre carte, cette situation a rendu la compréhension du fonctionnement de la carte originale une tâche difficile.

Ces raisons nous ont obligé à étudier, développer et à concevoir une nouvelle carte de commande qui permet de contrôler le bras *ROB3* en boucle fermée via une loi de commande conventionnelle.

II.4.4. Propriétés mécaniques

La structure *ROB3* est composée de pièces en aluminium vissées entre elles. Chacun des cinq axes indépendants est lié par un servomoteur à courant continu. La puissance de transmission est assurée par des engrenages. Pour la transmission des axes 3 et 4, des chaînes dentées sont utilisées en plus. Les positions absolues de tous les axes sont déterminées par des capteurs potentiométriques. Cette caractéristique la plus marquante du *ROB3* garantit que, même après le passage à une autre position, le robot repère la position exacte de chaque axe.

Comme il est représenté dans la Figure II.5, le bras manipulateur *ROB3* possède en total 5 axes de rotations. Le tableau II.2 permet de représenter les différentes limites des angles du bras manipulateur.

Les Figures II.6 et II.7 montrent les différentes longueurs des tiges selon 2 angles différents.

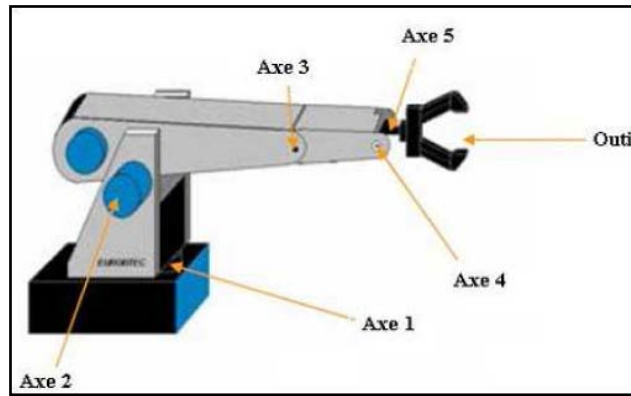


Figure II.5: Les axes du robot ROB3.

Les Articulations	Axe	Limite articulaire	Résolution
Axe 1 : Base	Q0	+80° ... 0° ... -80°	0 ... 255
Axe 2 : Epaule	Q1	+50° ... 0° ... -50°	0 ... 255
Axe 3 : Coude	Q2	0 ... +100°	0 ... 255
Axe 4 : Poignet	Q3	+100° ... 0° ... -100°	0 ... 255
Axe 5 : Orientation	Q4	+100° ... 0° ... -100°	0 ... 255
Outil (Pince)		0 ... 60 mm	0 ... 255

Tableau II.2 : Les angles de rotation du robot ROB3.

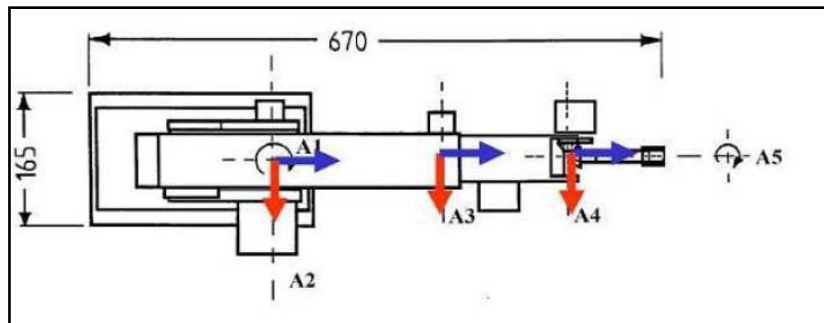


Figure II.6 : Les dimensions du robot ROB3 (vue de haut).

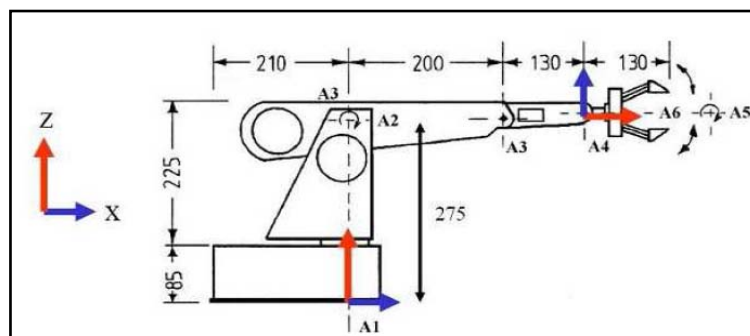


Figure II.7 : Les dimensions du robot ROB3 (vue de face).

II.5 Les actionneurs

Le bras est muni de six petits moteurs à courant continu, trois moteurs de marque «Bühler» et les trois autres sont des petits servomoteurs «Multiplex». Cinq d'entre d'eux sont destinés chacun à faire mouvoir une articulation. En revanche, le dernier est adapté pour ouvrir et fermer la pince.[10]

Les moteurs utilisés sont des moteurs à aimants permanents. Ils fonctionnent avec une tension de 9V générée par une alimentation continue stabilisée et un courant max de 3A.

Le moteur électrique à courant continu est le muscle du bras de robot, qui est un dispositif électromécanique servant à convertir de l'énergie électrique d'entrée en énergie mécanique.

En effet, il existe plusieurs types de moteurs électriques, on a étudié un seul cas, il s'agit d'un moteur à courant continu à aimants permanents. Ces moteurs qui présentent une très faibles inertie mécanique, sont très bien adaptés aux applications nécessitant des variations rapides de vitesse de rotation ou commande en position (périphériques d'ordinateurs, robotique, etc...). Ils sont toujours commandés par un ensemble électronique comportant une alimentation de puissance avec une électronique de commande réalisant un asservissement.

Classiquement, un moteur à courant continu est constitué d'un stator fixe et d'un rotor mobile. Le courant continu arrive par les cosses à l'extérieur de la carcasse sur des balais souples. Le collecteur est la partie du rotor qui touche les balais. Le rotor est constitué, quand à lui, de plusieurs bobines, toujours en nombres impairs, reliées au collecteur. Lorsque l'une des bobines reçoit le courant continu, elle crée un champ magnétique, mais la carcasse du stator abrite aussi deux aimants de pôles opposés. Le champ magnétique créé attire les pôles des aimants, créant par cela une rotation de l'axe du moteur.

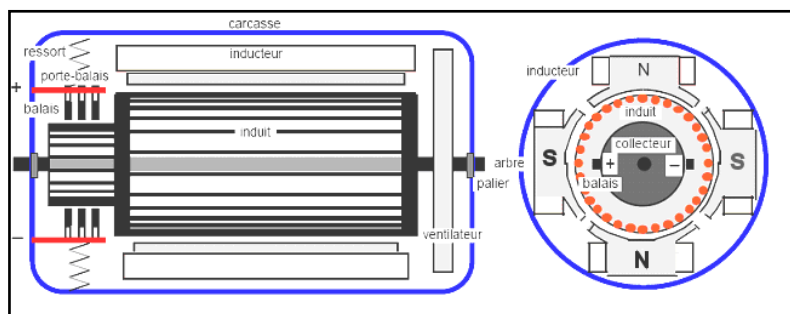


Figure II.8 Architecture d'un MCC à aimant permanent.

II.5 .1 Caractéristiques du moteur à courant continu

- Plage de variation de vitesse très grande (> 1000 en boucle d'asservissement)
- Couple de démarrage important, idéal pour l'entraînement de charges à forte inertie.
- Rapport volume/puissance très supérieur à toutes les autres technologies
- Rendement élevé
- Linéarité tension/vitesse, couple/courant

Mais :

- Prix élevé
- Maintenance coûteuse (remplacement des balais en graphite, usure du collecteur)
- Source importante de parasites.

II.5 .2 Modélisation du moteur à courant continu

Vu de l'extérieur, la machine peut être représentée par la mise en série d'une résistance R, d'une inductance L(par la suite, on ajoutera la valeur de l'inductance de lissage mise en série avec la machine) et d'une f.e.m à vide E_v donnée par la relation $E_v = K \Omega$, si Ω est la vitesse de rotation. Le schéma technologique d'un moteur DC à excitation séparée constante est donné sur la Figure II.9.

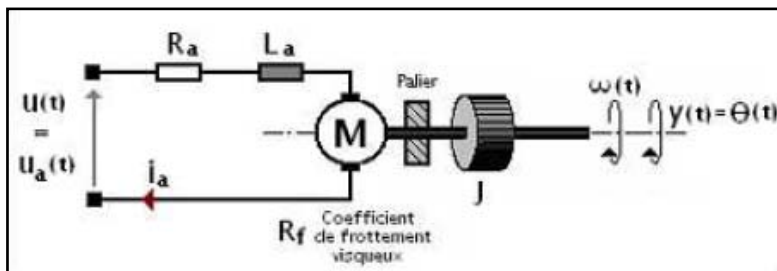


Figure II.9: Moteur DC à excitation séparée.

Equation électrique:

$$V_s(t) = R \cdot i(t) + L \cdot \frac{di(t)}{dt} + K \cdot \Omega(t) \tag{II.01}$$

Soit en variable de Laplace

$$V_s(p) = R \cdot i(p) + L \cdot p \cdot I(p) + K \cdot \Omega(p) \tag{II.02}$$

Equation mécanique:

$$j \cdot \frac{d\Omega(t)}{dt} = K \cdot i(t) - f \cdot \Omega(t) - C_{ch}(t) \tag{II.03}$$

Soit en variable de Laplace

$$j \cdot p \cdot \Omega(p) = K \cdot I(p) - f \cdot \Omega(p) - C_{ch}(p) \tag{II.04}$$

Remarque : Cch(t) est le moment du couple de charge.

II.6 Transmissions et systèmes réducteurs de vitesse :

La transmission de mouvement dans notre bras de robot est assurée à l'aide des courroies et des engrenages qui assurent une liaison entre les actionneurs et les articulations surtout au niveau du poigné. Suite aux vitesses élevées des moteurs à courant continu, le système réducteur de vitesse sert tout simplement à réduire la vitesse de rotation de l'axe de sortie.

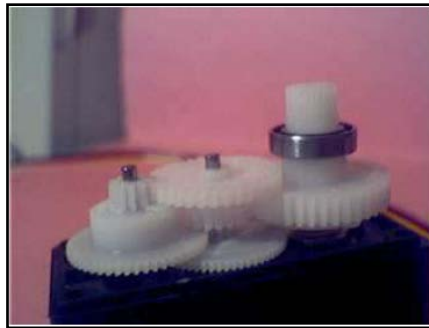


Figure II.10 : les pignons

II.7 Les capteurs

Plusieurs technologies de construction des capteurs existent. On rencontre dans ce contexte des capteurs incrémentaux et potentiométrique. Pour notre plateforme, des capteurs de position potentiométriques sont utilisés. (Figure II.11).

Le potentiomètre (Figure II.11) sert à l'asservissement de position. L'axe du chaque potentiomètre est solidaire de chaque l'axe de robot *ROB3*. Donc le potentiomètre tourne en même temps que l'axe de sortie. La résistance à la borne du potentiomètre varie donc en fonction de la position de l'axe. La résistance du potentiomètre est en rapport avec la durée du signal.



Figure II.11 : Le capteur du robot *ROB3*

Le tableau II.3 donne la relation entre les angles et les tensions correspondante délivrées par le potentiomètre.

Les Articulations	Axe	Limite articulaire	Résolution (Volts)
Axe 1 : Base	Q0	+80° ... 0° ... -80°	0...1.37
Axe 2 : Epaule	Q1	+50° ... 0° ... -50°	0.39...5
Axe 3 : Coude	Q2	0 ... +100°	2.35...4.31
Axe 4 : Poignet	Q3	+100° ... 0° ... -100°	0.98...4.90
Axe 5 : Orientation	Q4	+100° ... 0° ... -100°	0...5
Outil (Pince)		0 ... 60 mm	3.62...1.76

Tableau II.3 : Caractéristiques des capteurs du robot *ROB3*.Figure II.12 Photo de *ROB3* avec son alimentation.

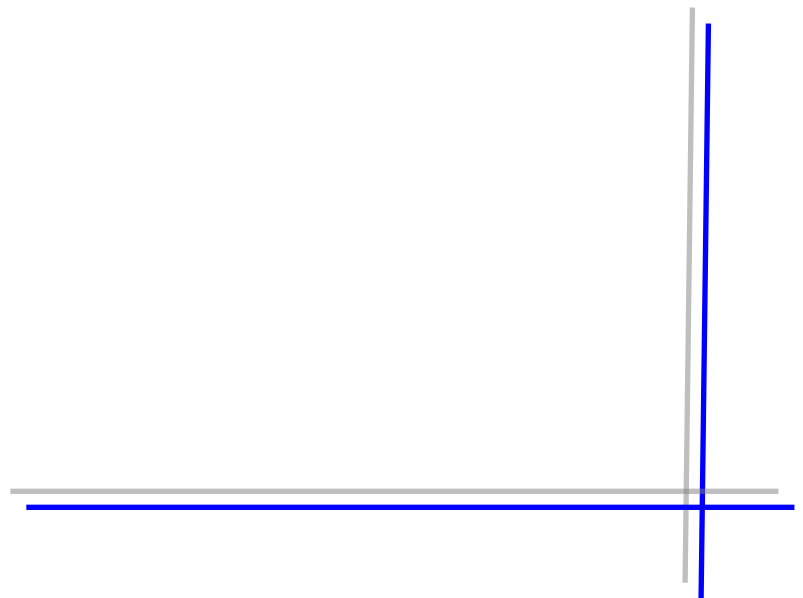
II.8 Conclusion

Dans le cadre de ce chapitre, on a pu décrire les différents éléments constitutifs de notre bras manipulateur *ROB3*. Une modélisation des *DC* moteurs qui animent la structure mécanique du *ROB3* est établie. Un essai expérimental nous a permis de délimiter l'espace du travail qui constitue une étape importante servant par la suite d'imposer les trajectoires de mouvement des différentes articulations.



Chapitre 3

Le PIC16F877A



III.1. INTRODUCTION

Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires « mémoire morte pour le programme, mémoire vive pour les données », unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique « quelques milliwatts en fonctionnement, quelques nano watts en veille », une vitesse de fonctionnement plus faible « quelques mégahertz » et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

Par rapport à des systèmes électroniques à base de microprocesseurs et autres composants séparés, les microcontrôleurs permettent de diminuer la taille, la consommation électrique et le coût des produits. Ils ont ainsi permis de démocratiser l'utilisation de l'informatique dans le domaine d'industrie pour la commande et contrôle des systèmes.

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile, etc.

Des nombreux fabricants se sont implantés sur le marché des microcontrôleurs, la société américaine Micro-chip à mis au point un microcontrôleur CMO « **Complementary Metal Oxide Semi-conductor** » appelé PIC ; cette dénomination est sous copyright de Micro-chip, donc les autres fabricants ont été dans l'impossibilité d'utiliser ce terme pour leur propre microcontrôleur. Actuellement ce microcontrôleur est très utilisé.

III.2. LES CARACTERISTIQUES DU PIC 16F877A

Nous avons utilisé le microcontrôleur **PIC 16F877A** pour contrôler la vitesse du moteur, ce μ c est un circuit intégré à 40 broches dont les caractéristiques générales peuvent être résumé comme suit :

- Vitesse d'exécution allant jusqu'à 20MHz permettant l'exécution rapide du programme, une instruction en cycle d'horloge de 200ns.
- Une mémoire vive « **RAM** » de 368 octets, répartie sur 4 banque 0, 1, 2, 3 « mémoire de données ».
- Une mémoire **EEPROM** pour sauver des paramètres de 256 octets « mémoire de données ».
- Une mémoire morte « **ROM** » mémoire programme de type FLASH de 8ko « 1mot=14bit ».
- **33** Entrées/Sorties programmables multiplexés.

- 8 entrées « canaux » du module de conversion analogique numérique 10 bits.
- 2 Comparateurs analogiques programmables avec une référence de tension, programmable également.
- Conservation des informations en mémoire vive jusqu'à 1.5v.
- Une plage d'alimentation de 2.0V à 5V.
- Module de capture **PWM** programmable permettant de capturer un signal « logique » en fonction du temps, ou d'offrir un signal **PWM** : **Module de Largeur d'Impulsion**, ce paramètre représente une grande importance dans notre réalisation.
- De nouvelles fonctionnalités, comme les gestions de ports «Série».

Le PIC 16F877A parmi les plus puissants microcontrôleurs de la gamme « Midrange» chez Micro-chip. Dans notre travail, on s'intéresse au convertisseur analogique/numérique au Timer2 et aux modules CCP1 et CCP2 pour la génération du signal PWM nécessaire à la commande du moteur à courant continu.

III.3. DESCRIPTION GENERALE DU PIC 16F877A

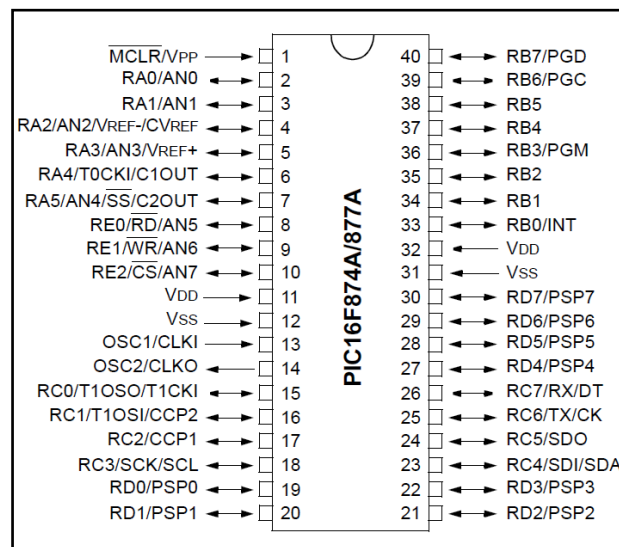


Figure III.1: Boitier du PIC 16F877A

III.3.1. Architecture externe

Le microcontrôleur **16F877A** fait partie de la sous famille des **16F87X**. Cette branche fait partie intégrant de la famille des **Pics** Mid-Range, Le **PIC 16F877A** se présente sous la forme d'un boitier **PDIP 40** broches, Le numéro peut être suivi d'un «**A**», et d'un «**-XX**» qui Donne la fréquence d'horloge maximum du composant. A l'heur actuel, la version la plus courante est la version **-20**. Donc la fréquence maximale sera de **20MHz**.

PIN	UTILISATION
RA0 à RA5	Lignes d'entrées /sorties du port A
RB0 à RB7	Lignes d'entrées /sorties du port B
RC0 à RC7	Lignes d'entrées /sorties du port C
RD0 à RD7	Lignes d'entrées /sorties du port D
RE0 à RE3	Lignes d'entrées /sorties du port E
VDD	Broche d'alimentation +5V
VSS	Broche du 0V
MCLR	Reset du circuit
OSC1 et OSC2	Broche recevant quartz externe.

Tableau III.1: Broches du PIC

a. Alimentation « VDD ET VSS »

Ce sont les broches d'alimentation du circuit, les tensions qui peuvent être appliqués sont :

- De **4V** à **6V** pour la gamme standard.
- De **2V** à **6V** pour la gamme étendue.

L'intensité du courant consommé peut aller de **1µA** à **10mA**.

La consommation sera en fonction de :

- La tension d'alimentation.
- La fréquence interne.
- Le mode de fonctionnement.

b. L'horloge

Deux pates **11** et **12** sont présentées sur le boîtier du **PIC 16F877A**. La première nommée **OSC1/CLKIN** et la deuxième **OSC/CLKOUT**. Ces deux broches peuvent être utilisées de plusieurs manières; cela dépend de l'oscillateur ou de l'horloge employés: **XT** : version à oscillateur à quartz «standard» fonctionnant jusqu'à une fréquence maximale de **4MHz**.

HS : High Speed, version à oscillateur à quartz capable de fonctionner jusqu'à une fréquence maximale de **20MHz**.

RC : capable de fonctionner jusqu'à **4MHz** mais avec une stabilité de fréquence plus faible que les versions à quartz.

LP « Low Power » :, version à quartz également, mais prévue spécialement pour les applications à très faible consommation. C'est donc une version dont la fréquence maximale de fonctionnement est limitée à **20KHz**.

c. MCLR

Cette broche sert à initialiser le **PIC**, ce dernier dispose de plusieurs sources de **RESET**.

POR « Power on reset » : mise sous tension. Un front descendant sur **MCLR** déclenche l'initialisation du μc . Cette broche peut être simplement reliée à **VDD** si on n'a pas besoin de **RESET** externe.

EXTERNAL RESET « mis à l'état de MCLR ». Remise à zéro extérieur. Il faut appliquer un niveau bas sur l'entrée.

RESET au moins **2 μs** pour que l'initialisation soit prise en compte.

WDT : chien de garde. Si le **WDT** arrive à la fin du temps de garde sans être rafraîchi il aura une initialisation du μc .

BOR : baisse de l'alimentation. Si l'alimentation **VDD** chute en dessous de **4V** pendant **100 μs** au moins, le microcontrôleur peut générer un **RESET**.

III.3.2. Architecture interne

Cette architecture nous présente les principaux organes qui composent le μc , la figure (2.2) illustre cette architecture :

- La mémoire **FLASH** de **8K** mots à **14** bits.
- La mémoire de donnée **EEPROM** à **256** octets.
- La mémoire de donnée **RAM** à **368** octets.
- Les registres particuliers **W** « registre de travail »; **FSR** et **STATUS** « d'état ».
- L'unité arithmétique et logique « **UAL** ».
- Cinq ports bidirectionnels **E/S A, B, C, D, E**.
- Huit niveaux de pile.

Cependant, même si les op-codes sont codés sur **14** bits, les données et les résultats sont toujours sur **8** Bits.

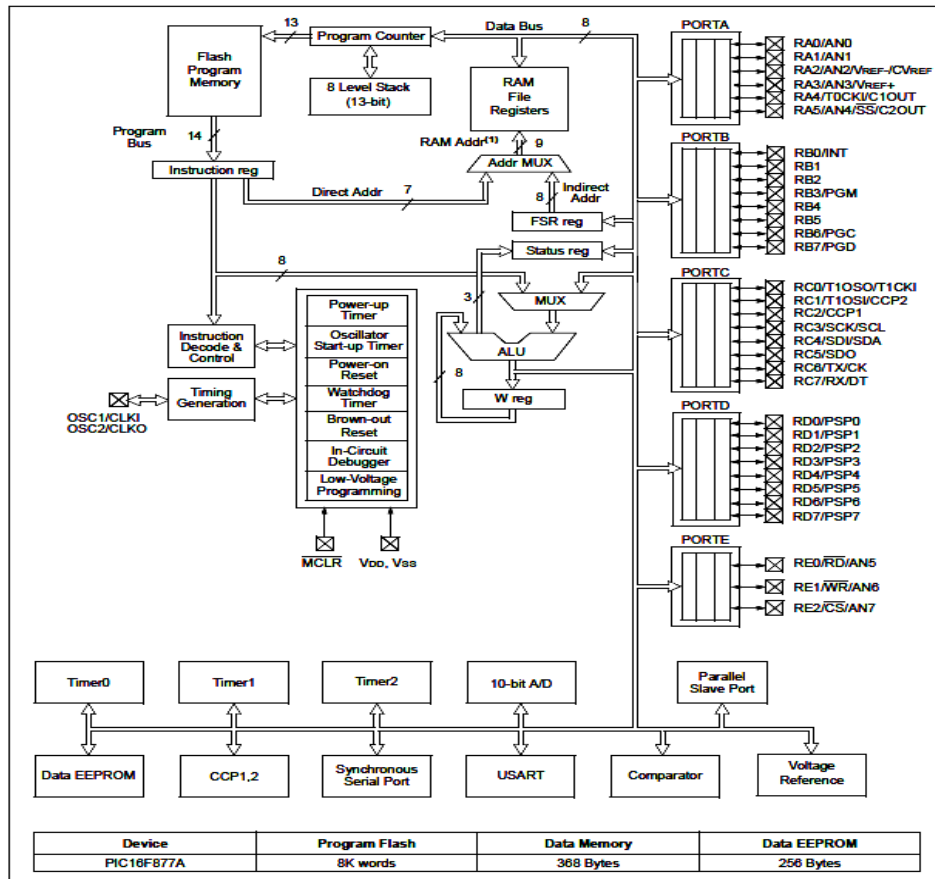


Figure III.2 : Architecture interne du pic 16F877A

III.3.3. Organisation de la mémoire

La mémoire du **PIC 16F877A** est divisée en 3 parties :

1. La mémoire programme

La mémoire programme est constituée de 8k mots de 14 bits. C’est dans cette zone que nous allons écrire notre programme. Le PIC exécute une à une les instructions logées dans la mémoire de programme.

2. La mémoire EEPROM

La mémoire EEPROM « Electrical Erasable Programmable Read Only Memory », est constituée de 256 octets, ces octets sont conservés après une coupure de courant et sont très utiles pour conserver des paramètres semi-permanents.

3. La mémoire RAM et organisation

La mémoire RAM est celle qui est sauvant utilisée. Toutes les données qui y sont stockées sont perdues lors d’une coupure de courant. La mémoire RAM disponible du 16F877 est de 368 octets. Elle est répartie de la manière suivante :

- 80 octets en banque 0, adresses 0*20 à 0*6F
- 80 octets en banque 1, adresses 0*A0 à 0*EF

- 96 octets en banque 2, adresses 0*110 à 0*16F
- 96 octets en banque 3, adresses 0*190 à 0*1EF

16 octets commun aux 4 banques, soit 0*70 à 0*7F = 0*F0 à 0*FF.

a. Plan mémoire pour les instructions « code programme »

Le plan mémoire est linéaire les adresses vont de **0000h** à **1FFFh** « **8k mots de 14 bits** », Par page de **2k mots**.

Les **Pics** n'ont qu'un seul vecteur d'interruption en **0004h**. Lors d'une interruption, le sous programme associé devra déterminer quel périphérique a demandé en mémoire de donnée comme avec les microcontrôleurs classiques, mais dans la mémoire programme. Elles sont utilisées lors d'appels de sous programmes, on ne peut pas imbriquer plus de **8** sous programmes.

b. Plan mémoire pour les données et registres interne

« *SFR : Spécial function register* »

Le plan mémoire des données et des registres internes est découpé en **4** zones ou Bank de **128** octets, pour accéder à une zone il faut positionner les bits **RB0** « **bit 5** » et

RP1 « **bit 6** » du registre **STATUS**.

Les registres appelés **General purpose Register** ne sont ni plus ni moins que des cases mémoires pour stocker les données.

III.3.4. Les registres du PIC 16F877A

La mémoire **RAM** est organisée en **4** banques pour le **16F8777A**. La **RAM** est subdivisée de plus en quatre parties. Dans chacune des banques on trouve des «cases mémoires spéciales» appelées **REGISTRES SPECIAUX** et des cases mémoires «libres» dont nous pouvons nous en servir à notre guise.

a. Le registre status d'adresse 0X03 ou registre d'états

Comme son nom l'indique ce registre contient un certain nombre de bits d'états de l'unité centrale mais aussi des bits de sélection de la page qui s'appellent ici **RP1** et **RP0**.

De ce fait, il peut être lu comme n'importe quel autre registre étant entendu que certains bits d'états restant non modifiables.

b. Le registre FSR d'adresse 0X04

Le **PIC 16F877A** possède une unité arithmétique et logique « **UAL** » et un registre de travail **W**. L'**UAL** effectue des fonctions arithmétiques booléennes entre les données se trouvant dans le registre de travail et n'importe quel registre. Pour les instructions à deux opérandes, typiquement une donnée est dans le registre de travail « **W** » et l'autre opérande dans un autre

registre ou se trouve une valeur immédiate constante. Pour les instructions à un seul opérande, la donnée se trouve soit dans le registre de travail « **W** » soit dans un autre registre.

Le registre de travail « **W** » possède **14** bits, utilisé pour les opérations avec l'**UAL**, ce n'est pas un registre adressable. Elle « **UAL** » est composée :

- D'un accumulateur **8** bits **W : WORKING** « travail », c'est lui qui effectue toutes les opérations arithmétiques et logiques.
- Un registre d'état **8** bits **STATUS**.

IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C	
Bit7								Bit0

- **Bit 0** ou bit **C Carry** c'est le bit de retenue en addition ou soustraction sur un octet. Il est positionné à 1 si une retenue est générée depuis le bit de poids fort
- **Bit 1** ou bit **DC** comme **Digit Carry0** il est positionné 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 d'opérations effectuées en code BCD
- **Bit 2** ou **Z** comme **Zéro**, ce bit est mis à 1 si le résultat de l'opération arithmétique ou logique est nul.
- **Bit 3** ou **PD** comme **Power Down**, ce bit est à 1 lors de la mise sous tension du circuit ou lors de l'exécution d'une instruction **CLWDT** relative au timer chien de garde il est mis à 0 par l'instruction **SLEEP**.
- **Bit 4** ou bit **TO** comme **Time Out**, ce bit est à 1 lors de la mise sous tension du circuit ou lors de l'exécution d'une instruction **CLWDT** ou **SLEEP**, il passe à 0 si le timer chien de garde déborde.
- **Bit 5** ou bit **RP0** « **Register Bank Select0** », ce bit sert à sélectionner l'une des deux pages de la mémoire « Page 0 ou Page 1 » sachant que chaque page mesure **128** octet.
RP0 = 0 sélectionne la page mémoire **0** « adresse de **00** à **7F** »
RP0 = 1 sélectionne la page mémoire **1** « adresse de **80** à **FF** »
- **Bit 6** ou bit **RP1** « **Register Bank Select1** », permet la sélection des pages **2** et **3**.
- **Bit 7** ou bit **IRP** ce bit est prévu pour un futur adressage paginé indirect.

III.3.5. Les timers

Les **Pics 16F877A** disposent **3** timers :

- a. **Le timer 0 « 8 bits »**: il peut être incrémenté par des impulsions extérieures via la broche « **TOCKI/RA4** » ou par l'horloge interne « **Fosc/4** ».

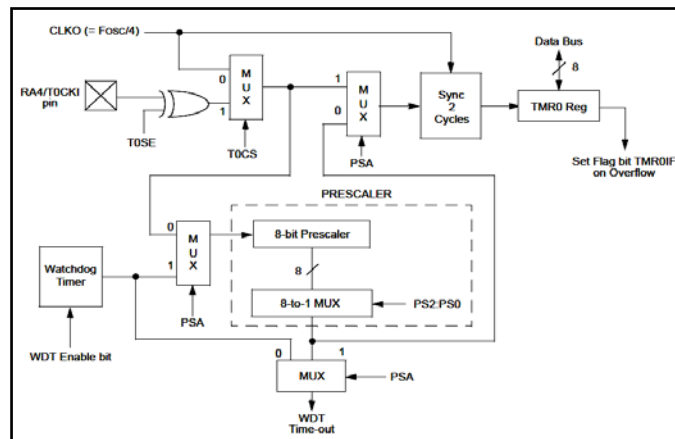


Figure III.3 : Schéma du Timer 0

b. **Le timer 1 « 16 bits »**: il peut être incrémenté soit par l'horloge interne, par des impulsions sur les broches **TICKI/RC0** ou par un oscillateur « **RC ou quartz** » connecté sur les broches **TOSO/RC0** et **TIOS1/RC1**.

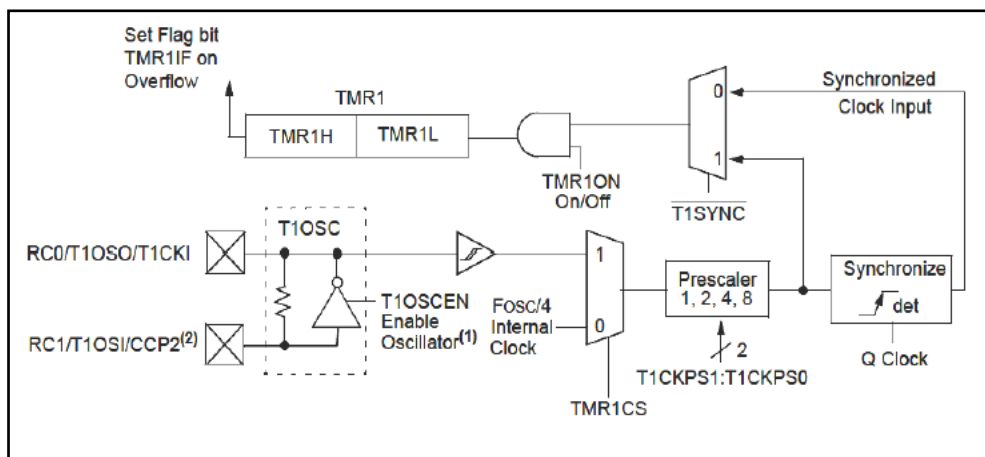


Figure III.4: Schéma descriptive du timer 1

c. **Le timer 2 « 8 bits »**: il est incrémenté par l'horloge interne, celle peut être pré divisé.

Tous ces **timers** peuvent déclencher une interruption interne, s'ils ont été autorisés.

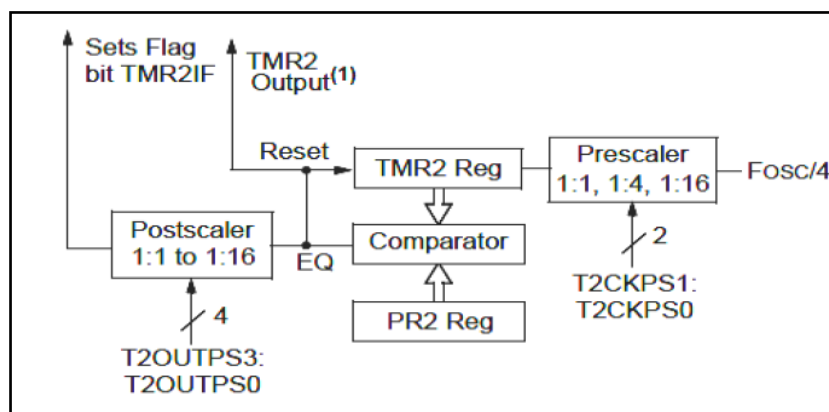


Figure III.5 : Schéma du Timer 2

III.3.6. Les interruptions

Le PIC 16F877A possède 15 sources d'interruptions. Les plus importantes à connaître, car ce sont les plus utilisées en pratique, sont :

- **TMR0I** : Dépassement du TIMER0 « débordement le passage de **255** à **0** »
- **TMR1I** : Dépassement du TIMER1 « débordement le passage de **65535** à **0** »
- **TMR2I** : Contenu du PR2 du TIMER2 « débordement le passage de la valeur qui contient dans le registre **PR2** à **0** »
- **INT** : Interruption extérieure RB0
- **RBI** : Changement d'état du port B « RB7 : RB4 »
- **PSPI** : Lecture / Ecriture sur le port parallèle
- **ADI** : Convertisseur Analogique / Numérique
- **TXI** : Emission sur le port RS232
- **RCI** : Réception sur le port RS232
- **SSPI** : Bus SPI / I²C
- **CCP1I** : Capture / Compare / PWM
- **CCP2I** : Capture / Compare / PWM
- **EPI** : Ecriture dans EEPROM
- **CMI** : Comparateur
- **BCLIF** : Collision sur Bus.

III.3.7. Les ports d'entré sorties

Le microcontrôleur **16F877A** dispose **5 PORT** « **A, B, C, D, E** ». Tous ces ports sont bidirectionnels.

La plupart des lignes de **Ports** ont une double fonction.

- Le **PORT A** « **5 bits** » **I/O** pure et/ou convertisseur analogique et/ou **TIMER0**.
La broche **RA4** du **PORT A** « Entrée du **TIMER T0CKI** » est du type **DRAIN OUVERT**.
- Le **PORT B** « **8 bits** » **I/O** pure et/ou programmation **ICSP/ICD** « broche **RB3/PGM, RB6/PGC** et **RB7/PGD** » et l'entrée d'interruption externe **RB0/INT**.
- Le **PORT C** « **8 bits** » **I/O** : C'est un port tout ce qu'il a de plus classique, or qu'il a Deux pins qu'on utilisera plus tard dans la communication série avec le PC à travers « **TX** et **RX** ».

- Le **PORT D « 8 bits » I/O** Le registre **TRISD** comportera donc les 8bits de direction, pendant que le registre port D correspond aux pins I/O concernés.
D'où les 8 pins I/O, en mode entrée, sont du type « trigger de Schmitt ».
- Le **PORT E « 3 bits » I/O** Ce port n'est présent que sur les PIC 16F877. Il ne comporte que 3 pins **RE0** à **RE2**, mais, contrairement aux ports, les bits non concernés de **TRISE** sont, cette fois, implantés pour d'autres fonctions. Les pins **REX** peuvent également être utilisés comme pins d'entrées analogiques. D'où le registre **ADCON1** qui détermine si ce port est utilisé comme port I/O ou comme port analogique.

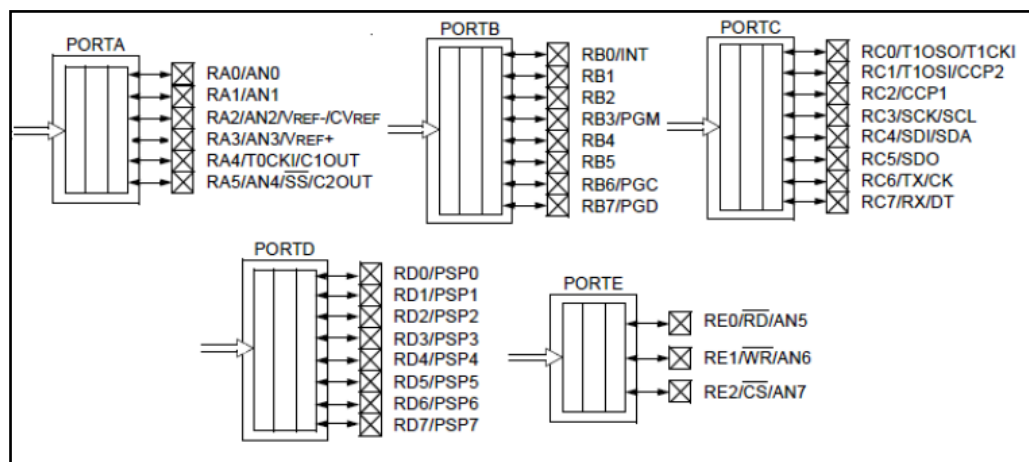


Figure III.6 : Les différents PORT de PIC16F877A

III.3.8. Configuration des portx, les registres PORTX et TRISX

Tous les ports sont pilotés par deux registres :

- Le registre **PORTx**, si le **PORTx** ou certaines lignes de **PORTx** sont configurés en sortie, ce registre détermine l'état logique des sorties.
- Le registre **TRISx**, c'est le registre de direction. Il détermine si le **PORTx** ou certaines lignes de port sont en entrée ou en sortie. L'écriture d'un 1 logique correspond à une entrée « 1 comme Input » et l'écriture d'un 0 logique correspond à une sortie « 0 comme Output ».

Au **RESET** toutes les lignes de ports sont configurés en sorties.

III.3.9. Le convertisseur analogique numérique

Notre **16F877A** travaille avec un convertisseur analogique/numérique qui permet un échantillonnage sur 10 bits. Il est composé de :

- Un multiplexeur analogique 5 voies « **PIC 16F877A** ».
- Un échantillonneur bloqueur.
- Un convertisseur analogique numérique de **10** bits.

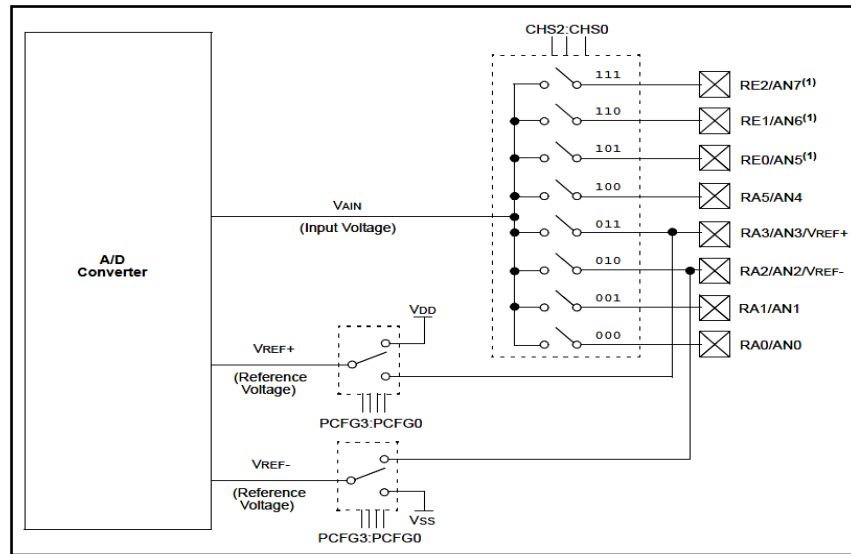


Figure III.7 : Schéma fonctionnel analogique numérique

a. Les registres ADRESL et ADRESH

Il est important de savoir que le convertisseur donne un résultat sur 10 bits, et donc que ce résultat devra donc obligatoirement être sauvegardé dans 2 registres. Ces registres sont tout simplement les registres ADRESL et ADRESH. Comme 2 registres contiennent 16 bits, et que nous n'en utilisons que 10, Micro chip nous a laissé le choix sur la façon dont est sauvegardé le résultat, nous pouvons soit justifier le résultat à gauche, soit à droite.

La justification à droite complète la partie gauche du résultat par des « 0 ». Le résultat sera donc de la forme :

ADRESH						ADRESL									
0	0	0	0	0	0	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

La justification à gauche procède bien évidemment de la méthode inverse :

ADRESH										ADRESL					
B9	B8	B7	B6	B5	B4	B3	B2	B1	B0	0	0	0	0	0	0

b. Le registre ADCON1

Ce registre permet de déterminer le rôle de chacune des pins AN0 à AN7. Il permet donc de choisir si un pin sera utilisé comme entrée analogique, comme entrée/sortie standard, ou comme tension de référence. Il permet également de décider de la justification du résultat. Pour pouvoir utiliser un pin en mode analogique, il faudra que ce pin soit configuré également en entrée par TRISA. Le registre ADCON1 dispose, comme tout registre accessible de notre PIC, de 8 bits, dont seulement 5 sont utilisés

B7	ADFM	A/D result format select
B6	INUTILISE	lu comme « 0 »
B5	INUTILISE	lu comme « 0 »
B4	INUTILISE	lu comme « 0 »
B3	PCFG3	Port Configuration control bit 3
B2	PCFG2	Port Configuration control bit 2
B1	PCFG1	Port Configuration control bit 1
B0	PCFG0	Port Configuration control bit 0

Tableau. III.2: bits de registre ADCON1

c. Le registre ADCON0

Ce registre est le dernier utilisé par le convertisseur analogique/numérique. Il contient les bits que nous allons manipuler lors de notre conversion. Sur les 8 bits de notre registre, 7 seront utilisés.

B7	ADFM	A/D result format select
B6	INUTILISE	lu comme « 0 »
B5	INUTILISE	lu comme « 0 »
B4	INUTILISE	lu comme « 0 »
B3	PCFG3	Port Configuration control bit 3
B2	PCFG2	Port Configuration control bit 2
B1	PCFG1	Port Configuration control bit 1
B0	PCFG0	Port Configuration control bit 0

Tableau III.3.: bits de registre ADCON0

III.3.10. Les modules CCP1 et CCP2

Ces deux modules peuvent fonctionner dans l'un des trois modes ci-dessous :

- Mode capture « **CAPTURE** » : ce mode permet en outre d'effectuer des mesures de temps.
- Mode comparaison « **COMPARE** » : ce mode permet en outre de générer des événements périodiques.
- Mode **PWM** signifie « **Pulse Width Modulation** », ce qu'on pourrait traduire par modulation de largeur d'impulsion, il s'agit d'un signal binaire de fréquence fixe dont le rapport cyclique peut être modulé par logiciel. Le rapport cyclique d'un signal

binaire à fréquence fixe peut être défini comme étant le rapport entre le temps où il se trouve à l'état «1» par rapport au temps total d'un cycle.

Ces modules sont associés aux broches **RC2/CCP1** et **RC1/T1OSI/CCP2** suivant le mode choisi, les **timer 1** ou **2** vont être utilisés. Les modes capture et comparaison utilisent le **timer 1**, tandis que le mode **PWM** utilise le **timer 2**.

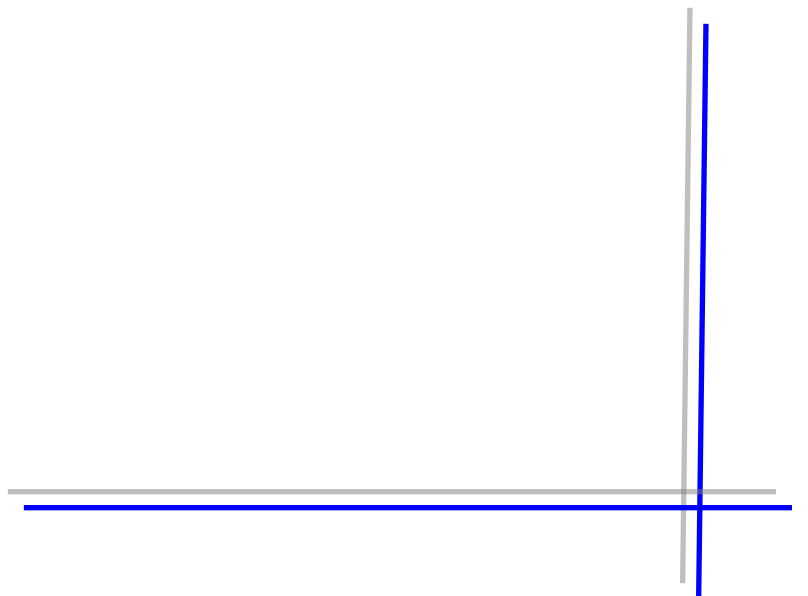
III.4. CONCLUSION

En conclusion, nous pouvons dire que le microcontrôleur 16F877A peut bien jouer le rôle d'une unité de commande pour notre système, il contient tous les modules nécessaires pour générer le signal PWM utilisé dans la commande de la carte de puissance qui alimente notre moteur à courant continu.



Chapitre 4

Réalisation hardware et software



IV.1 Introduction

La commande en mouvement des robots manipulateurs est essentiellement réalisée par des circuits numériques programmables tels que les microprocesseurs, les microcontrôleurs et les *DSP* où les lois de commande sont implémentées sous forme d'algorithmes. A noter que les principaux avantages de la commande numérique sont la rapidité de développement, flexibilité et faible espace d'occupation.

Dans le présent chapitre, nous présentons la réalisation d'une carte de commande à base d'un microcontrôleur et une carte de puissance à base d'hacheurs et un interface graphique dans le but de créer la communication entre le PC et le bras robot *ROB3*.

IV.2. Schéma simplifié du système de commande [9]

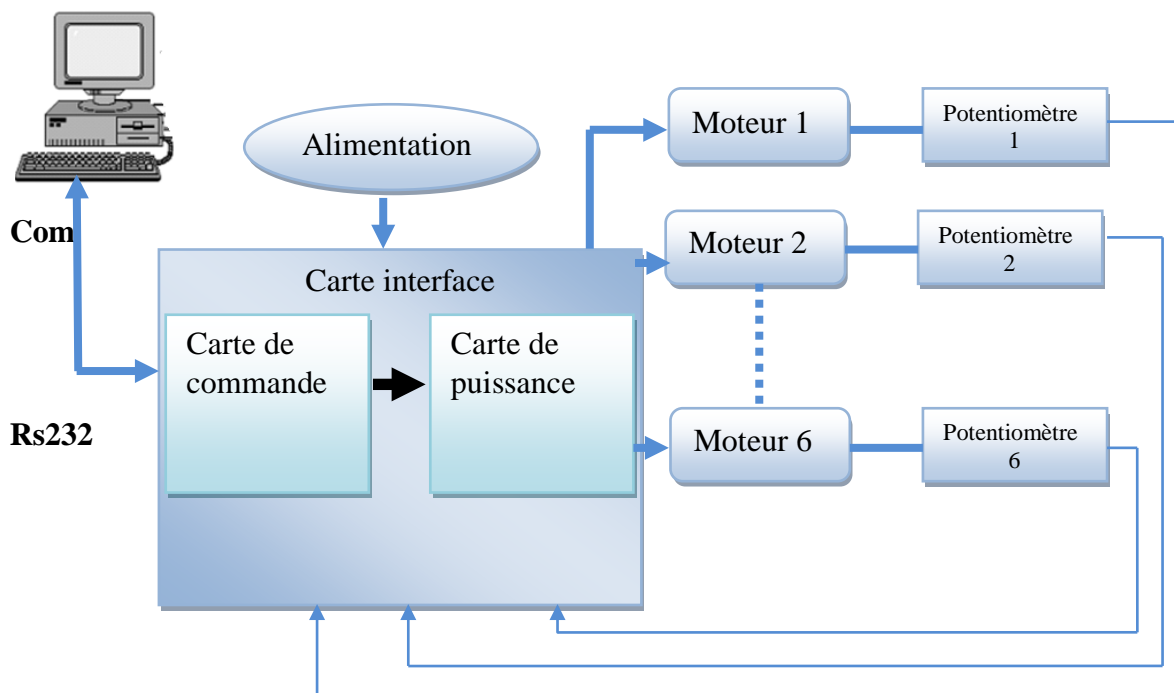


Figure IV.1 : Schéma simplifié du système de commande

Notons que la carte d'interface est composée de deux parties:

1. La carte de commande,
2. La carte de puissance.

L'alimentation du *ROB3* fournit une tension de 9V sous un courant de 3A (voir Figure IV.2). Les six moteurs du bras *ROB3* sont de types moteurs à courant continu associés à des potentiomètres rotatifs qui constituent les capteurs de positions.



Figure IV.2 : L'alimentation du ROB3 (9V 3A)

IV.3. La carte de commande

IV.3.1- Schéma électronique de la carte de commande

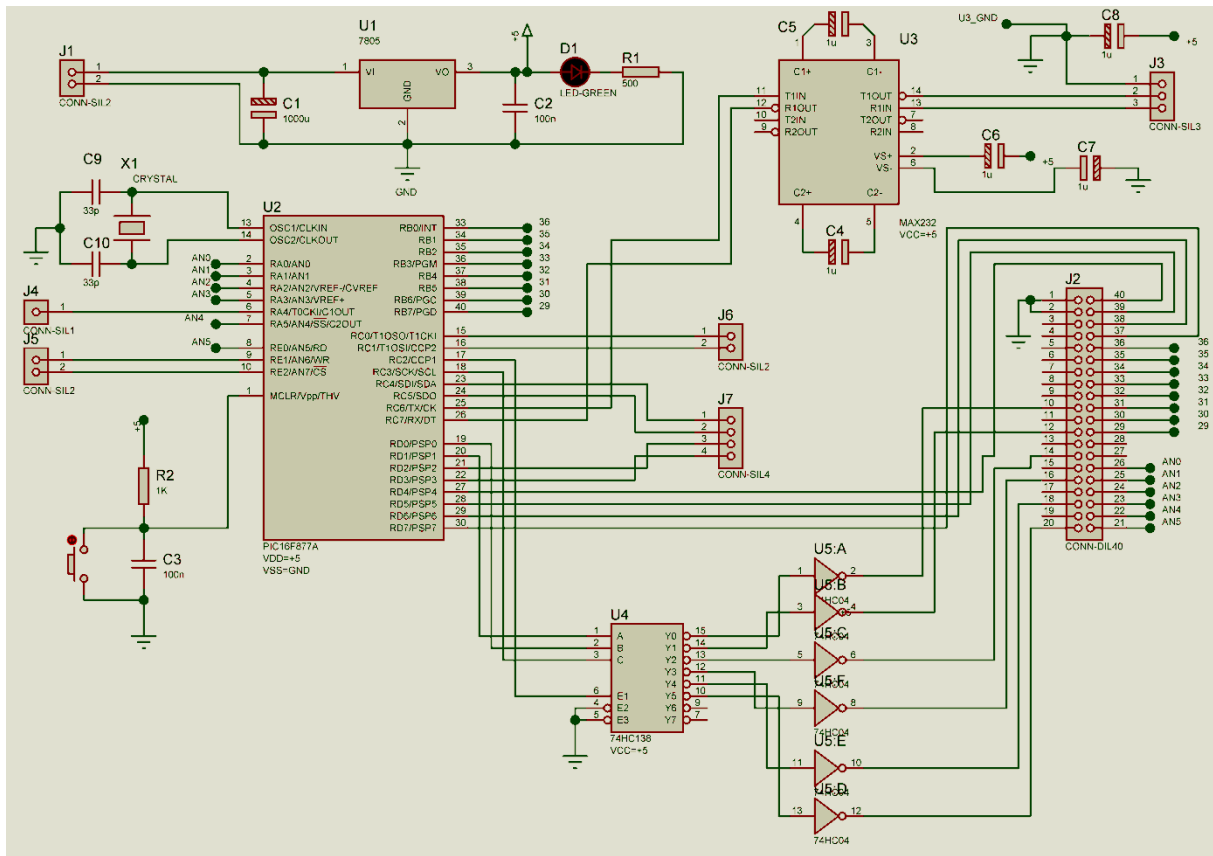


Figure IV.3 : Schéma de la carte de commande

IV.3.2 Eléments constituant la carte de commande

IV.3.2.1 Le microcontrôleur PIC

Il est intéressant de souligner que notre choix s'est porté sur un microcontrôleur de la firme *MICROCHIP*. Ce microcontrôleur est le plus répandu dédié pour des applications de la robotique. Son prix compétitif, la disponibilité de la documentation, ses performances et la distribution gratuite des logiciels pour le programmer et le déboguer, font de lui un bon candidat pour différentes applications industrielles.

Les 40 broches du *PIC* dont les 33 E/S et les différents modules, sont réparties sur cinq ports. Leur configuration est la suivante :

Le port A:

Nous avons configurés les pins *A0*, *A1*, *A2*, *A3* et *A5* du port en entrées analogiques pour pouvoir mesurer les différentes positions des segments du bras de robot *ROB3*.

Le port B: (8 pins entrées/sorties.)

- Les pins *RB7* et *RB6* → sorties destinées à commander le sens de rotation du moteur N°=6.
- Les pins *RB5* et *RB4* → sorties destinées à commander le sens de rotation du moteur N°=5.
- Les pins *RB3* et *RB2* → sorties destinées à commander le sens de rotation du moteur N°=4.
- Les pins *RB1* et *RB0* → sorties destinées à commander le sens de rotation du moteur N°=3.

Le port C: (8 pins entrées / sorties.)

- Le pin *RC2* → sortie utilisée pour la génération du signal *PWM*,
- Le pin *RC3* → sortie utilisée pour la sélection du moteur à commander par le signal *PWM*.

Le port D: (8 pins entrées / sorties.)

- Le pin *RD0* → sortie employée pour la sélection du moteur à commander par le signal *PWM*.
- Le pin *RD1* → sortie employée pour la sélection du moteur à commander par le signal *PWM*.
- Les pins *RD7* et *RD6* → sorties employées pour la commande du sens de rotation du moteur N°=2.

- Les pins *RD5* et *RD4* → sorties employées pour la commande du sens de rotation du moteur N°=1.

Le port E: (3 pins entrées / sorties.)

- Le pin *RE0* → Entrée servant à mesurer la position du moteur de la base.

La réinitialisation du pic est assurée en générant un *RESET MATERIEL* sur la broche */MCLR*. Le signal d'horloge fourni au *PIC* est généré par un petit circuit oscillant composé de 2 condensateurs d'une capacité de *33 pF* chacun et d'un oscillateur quartz. La fréquence fournie est de *20MHz* et la figure IV.4 montre le brochage du circuit d'oscillation.

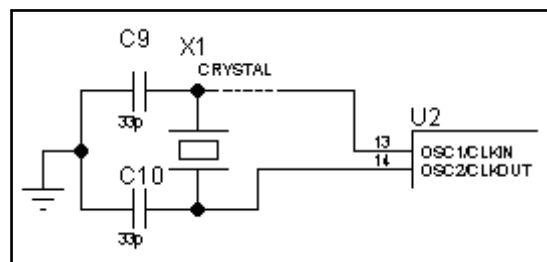


Figure IV.4 circuit oscillant du PIC

IV.3.2.2 Le circuit MAX232

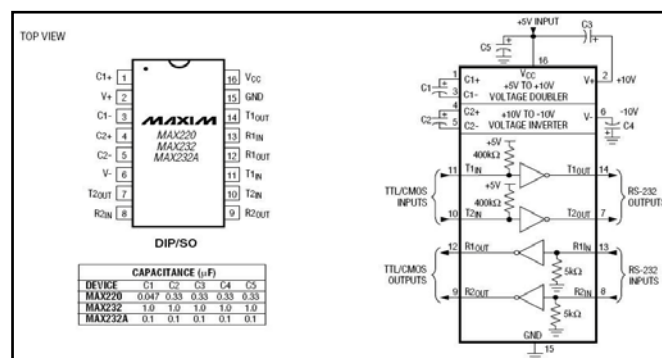


Figure IV.5 Le driver / receiver MAX232

Les cartes électroniques à base de microcontrôleurs fonctionnent très souvent avec des niveaux TTL soit 0-5Volt, 0V pour le niveau 0 et 5Volt pour le niveau 1. Brancher donc directement une ligne RS232 sur un microcontrôleur n'aurait donc aucun sens et pourrait aussi endommager le système en imposant des tensions de 25volt.

Pour rendre compatible une ligne RS232 avec une carte de ce type il existe un composant très simple d'utilisation que nous allons étudier : le max232. En regardant Figure IV.5, nous constatons directement qu'il est premièrement doté d'un convertisseur de tension, à travers des capacités C1 et C3 il génère une tension de 10Volt depuis les 5Volt (doubleur de tension), et au moyen des capacités C2 et C4 il génère une tension de -10Volt a partir de la tension de 10Volt. Il est bien sur évident que la puce est munie de tout un système, avec un oscillateur, des diodes et ... afin d'intégrer ce convertisseur DC-DC. Il existe une version de cette puce, le

max233, où les capacités sont intégrées directement dedans, mais nous ne rentrerons pas dans ce détail.

A. La liaison RS232

La liaison série à la norme RS 232 est utilisée dans tous les domaines de l'informatique. Elle permet la réalisation d'une liaison simple à mettre en œuvre entre 2 équipements. Elle est de type asynchrone, c'est à dire qu'elle ne transmet pas de signal horloge, les deux équipements doivent être configurés avec une même vitesse de transmission identique. Ils doivent par ailleurs utiliser le même protocole.

Le *PIC*, configuré en mode *USART*(Universal Synchronous Asynchronous Receiver Transmitter), permet une communication série de type asynchrone sous la norme *RS232*, avec un ordinateur via ses pattes *Rx* (réception) et *Tx* (transmission) branchées sur un port série.[10]

Les niveaux logiques de l'utilisation de tension pour ce protocole sont donnés comme suit:

Etat	Niveau logique	<i>Tia / eia - 232 - f</i> (<i>RS-232C</i> → <i>MAX232</i>)	<i>TTL/CMOS</i> (<i>PIC16F877A</i> → <i>MAX232</i>)
<i>Tx</i>	1	-12V	5V
<i>Rx</i>	0	+12V	0V

Tableau IV.1: Adaptation des signaux *TTL* et *Tia/eia-232*

B. Paramètres de la liaison RS232

L'utilisation du port série, impose la détermination des paramètres de communications sur le port. Ces paramètres sont décrits ci-dessous:

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.

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.

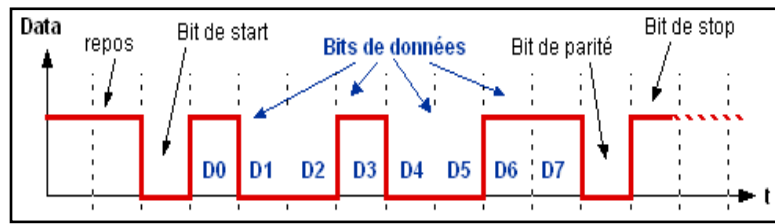


Figure IV.6 : Trame de communication série.

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

Remarque : 2 erreurs de transmission s'annulent.

Exemple de configurations :

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

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

C. Le brochage de la liaison série (DB9) Les prises RS232 sont aux formats DB9 ou DB25, nous nous intéresserons uniquement au format DB9.

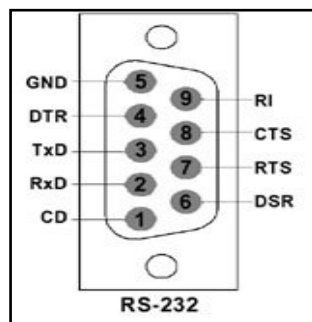


Figure IV.7: Le connecteur DB9 mâle

N°	Signal	E/S	Utilisation
1	CD	E	Carrier Detect: Annonce que l'autre équipement reçoit une réponse.
2	RD	E	Received Data : Entrée de réception des données
3	TD	S	Transmitted Data : Sortie d'émission des données
4	DTR	S	Data Terminal Ready : Indique à l'autre équipement que l'on souhaite communiquer.
5	SG		Signal Ground : Masse de référence des signaux (0V)
6	DSR	E	Data Set Ready : Indique que l'équipement opposé est prêt.
7	RTS	S	Request To Send : Demande à l'équipement opposé de se tenir prêt à recevoir.
8	CTS	E	Clear To Send : Indique que l'équipement opposé est prêt à recevoir.
9	RI	E	Ring Indicator : Annonce que le modem reçoit un appel.

Tableau IV.2 : trame de communication série

D. Types de liaisons séries

Si on se penche sur le nombre de fils nécessaires pour réaliser une liaison série on peut trouver quatre types:

- Liaison simplex.
- Liaison half duplex.
- Liaison full duplex.
- Liaison full duplex complète.

D.1) Liaison simplex

Un système est émetteur, l'autre récepteur. C'est par exemple le cas des émissions radio ou télévision.

D.2) Liaison half duplex

Les deux systèmes peuvent être à la fois émetteur et récepteur, mais les émissions ne peuvent avoir lieu en même temps.

D.3) Liaison full duplex

Elle dispose du même câblage que la liaison half duplex mais les deux interlocuteurs peuvent émettre et recevoir en même temps.

D.4) Liaison full duplex complète

- C'est la plus perfectionné.
- Permet de dialoguer dans les deux sens en même temps.
- Elle possède des fils supplémentaires permettant de contrôler le flux de données.
- A l'origine, elle a été décrite uniquement pour faire communiquer un ordinateur avec un périphérique en mode **RS232**.

Remarque

1. La vitesse de transmission choisie est de *9600 Bps*.
2. La trame est composée de: 1bit de START, 8bits pour la donnée et 1bit de stop. Par contre il n'y a de stop pas de parité.
3. Dans notre montage, nous avons utilisé une liaison half duplex en prenant les deux broches principales de transmission (2 et 3) correspondantes respectivement à *Tx* et *Rx*. La broche 5 est leur masse de référence.

IV.3.2.3 Le circuit 74HC138

Le 74HC138 est un circuit intégré qui fournit 8 sorties actives mutuellement en basses tensions (Y0 à Y7). Ce dispositif est à grande vitesse compatible avec les signaux de basses puissances (TTL). En effet, il s'agit d'un décodeur qui accepte trois entrées d'adresse binaires A0, A1 et A3 (ces adresses sont codées selon le nombre de moteurs à courant continu du bras ROB3. C'est-à-dire de 0 à 5 dont une seule sortie est active.

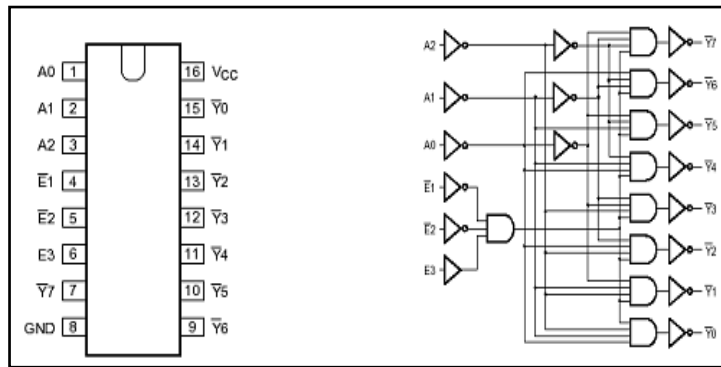


Figure IV.8 : Le circuit et diagramme logique du 74HC138.

Notre signal PWM provenant du microcontrôleur (PIN C2) est injecté dans le 74HC138 par l'entrée (E3), et est démultiplexé, par la suite, sur six sorties correspondantes à nos six (06) signaux PWM pouvant activer les circuits hacheurs (L298N) de la partie puissance. Mais à ce stade, nous avons des signaux de sorties à états logiques bas (Y0 à Y5).

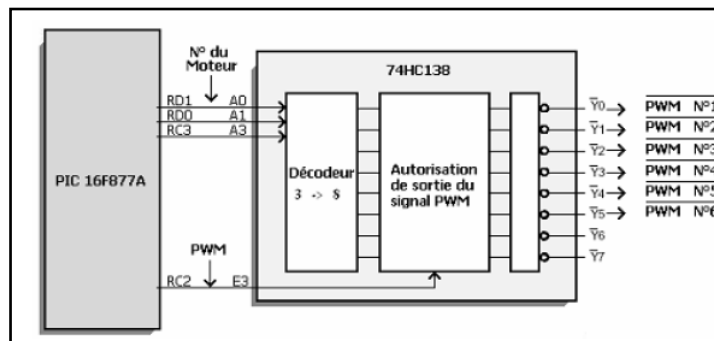


Figure IV.9 : Schéma d'utilisation du circuit 74HC138.

III.3.2.4 Le circuit 74HC04

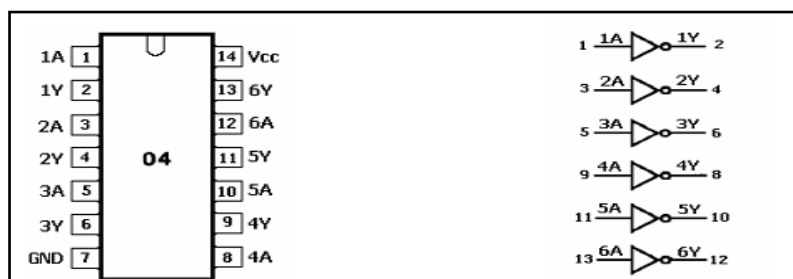


Figure IV.10 : Circuit et diagramme du 74HC04.

Dans notre montage, Le rôle du *74HC04* est d'inverser les six signaux de sorties de commandes (*PWM*) provenant du décodeur (*74HC138*) avec un niveau de tension (bas).

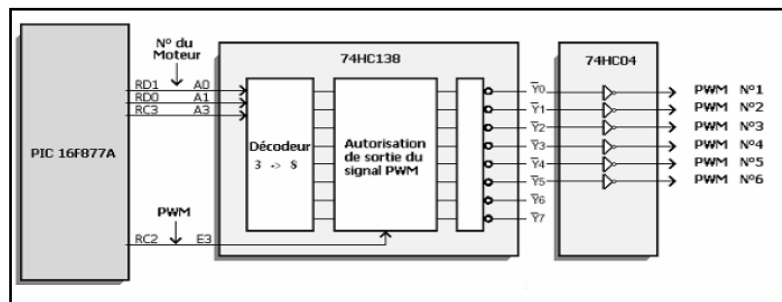


Figure IV.11 : Schéma d'utilisation du circuit *74HC04*.

IV.4 La carte de puissance

Cette partie du montage est alimentée avec une tension de *9V/3A* fournit par l'alimentation du bras de robot *ROB3*. Ce montage se concentre sur l'élément central, le *L298N*, qui est le cœur du montage. En effet, le circuit amplificateur de puissance *L298N* de *SGS Thomson* est une référence en matière d'interface pour les moteurs *DC* et les moteurs pas à pas. Ces caractéristiques à basses tensions, lui confèrent une place incontestée dans les circuits de puissance intégrée. Il est le plus utilisé pour les petits robots alimentés par quelques accus.

1. Les signaux de commande entrants du circuit considéré sont :

- Six (06) signaux *PWM* (Un signal *MLI* démultiplexé sur six voies).
- Six (06) paires de signaux de sens correspondant au sens du moteur sélectionné.
- Une masse commune entre les deux parties puissance et commande.

2. Les signaux de mesures sortant sont :

- Six (06) signaux analogiques fournies par les capteurs à potentiomètres rotatifs situés sur les axes de rotation du robot.

IV.4.1 Schéma électronique de la carte de puissance

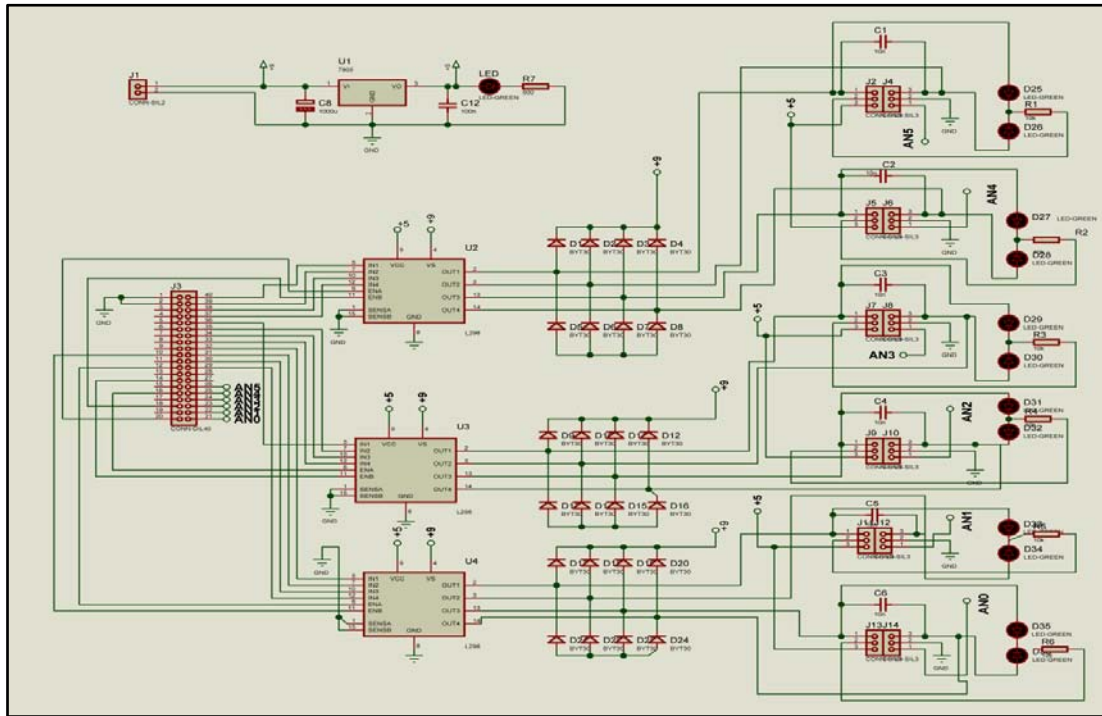


Figure IV.12 : Schéma de la carte de commande

IV.4.2 Eléments constituant la carte de puissance

IV.4.2.1 Le circuit L298N

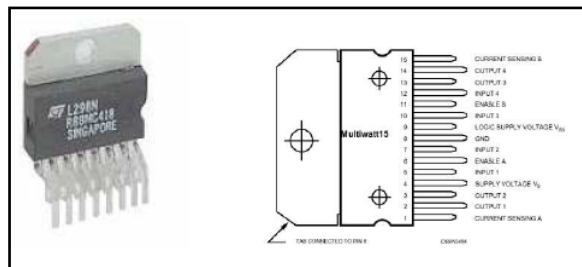


Figure IV.13 Le circuit L298N.

Dans notre montage, nous avons besoin de six ponts en H, c'est pour cette raison que nous avons utilisées trois circuits intégrés L298, car chaque circuit offre un double pont en H.

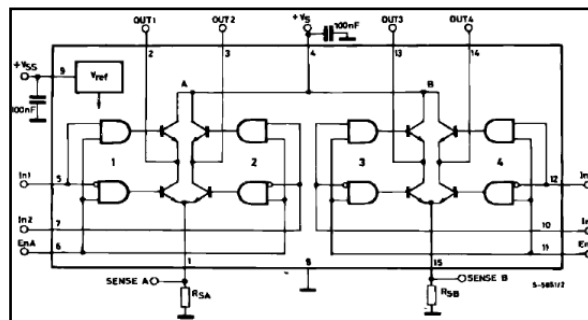


Figure IV.14: Diagramme bloqué du L298.

Les caractéristiques du L298 sont les suivantes :

- Intensité maximale : 2A par pont.
- Alimentation de puissance de 5.5V à 50V.
- Type de boîtier : Multiwatt15.
- Dissipation puissance total : 25w.
- Trois entrées par pont : In1, In2 et ENABLE.

Comme on le constatera, par la suite, sur le schéma de la figure(IV.14), le *L298N* possède deux broches d'alimentation. L'une pour le moteur (V_s) et l'autre pour la logique interne (V_{ss} , +5V), ceci lui permet de limiter la dissipation du boîtier.

Par ailleurs, il possède sa propre logique de commande (pour un moteur) :

Entrées		Sorties	Fonction
ENA = 1	In1=1 ; In2=0	Out1=1 ; Out2=0	Sens 1
	In1=0 ; In2=1	Out1=0 ; Out2=1	Sens 2
	In1=In2	Out1=1 ; Out2=1 Ou Out1=0 ; Out2=0	Le frein
ENA = 0	In1=X ; In2=X	Rien	roue libre

Tableau IV.3: Logique de commande du *LN298* (un pont).

Donc, le pilotage des moteurs se fait par génération d'une *MLI* en s'appliquant sur les entrées «Enable A» et «Enable B» et le *L298N* se charge de contrôler la vitesse et le sens de rotation des moteurs indépendamment.

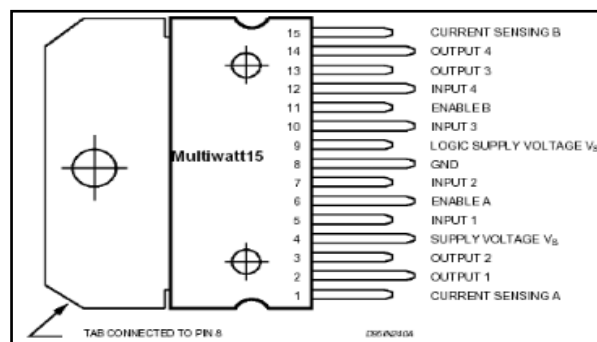


Figure IV.15: Brochage du circuit intégré *L298*.

III.4.2.2 Les diodes 1N4007:

Les diodes (*1N4007*) sont des diodes de puissance présentent un temps de commutation faibles et capables de laisser passer un courant important. Elles servent de protection « roue libre », afin de compenser le courant inverse qui se fera pour chaque déclenchement ON/OFF du moteur. Donc nous avons câblé huit (08) diodes en antiparallèle sur chaque L298N pour la protection de ce circuit parce que contrairement à la majorité des circuits, le *L298N* n'intègre pas les diodes de protections des transistors internes.



Figure IV.16 : Diode *1N4007*

IV.4.2.3 Le circuit d'alimentation

Cette partie est très simple. Nous avons besoin de produire une tension stabilisée. Il existe pour cela des composants dédiés, les 78xx. Ces composants sont des composants à trois pattes. Leur fonctionnement est très simple. Ils ont entre les deux premières pattes une tension d'alimentation quelconque (supérieure à la tension de sortie désirée). Ils fournissent alors en sortie une tension stabilisée égale à xx (05, 09, 12,...). On utilise donc ces composants en parallèle avec des condensateurs de découplage comme conseillé dans le datasheet.

Pour notre montage on a besoin de tension stabilisée de 5v donc en utilise le régulateur *lm7805* et pour la protection contre les courts-circuits ou les surintensités de courant en ajoute un fusible est disposé directement après l'alimentation. Suivie d'un groupe *RESISTANCE-LED* de contrôle qui confirme la mise sous tension du circuit et le sens de rotation du chaque moteur.



Figure IV.17 . fusible et son support.

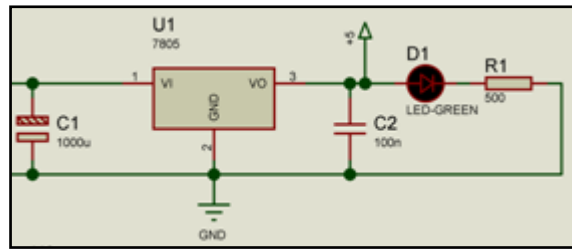


Figure IV.18. Circuit d'alimentation.

Remarque

1- Nous notons que la carte de puissance contient aussi des condensateurs afin d'assurer le filtrage de la tension issue du *LM7805* ainsi que les tensions appliquées aux moteurs du bras *ROB3* via des connecteurs (2*3).

2- Pour la liaison entre les deux cartes on a utilisé une nappe *IDE* de type (2*20).

IV.5 La programmation du PIC**IV.5.1 Environnement de programmation**

Nous avons utilisés pour programmer le microcontrôleur, le langage *C* sous le logiciel *PCM pic c* compiler de la compagnie *CCS* (Custum Computer Services), est distribué gratuitement sur le site web: www.ccsinfo.com/download.

IV.5.2 Le circuit programmeur

Pour la programmation de pic nous avons utilisés un simple circuit programmeur qui s'appelle *JDM* Ce type de programmeur est dépourvu d'alimentation. Ca à l'avantage d'être simple, puisque l'alimentation est tirée du port série et il est composé de très peu de composants.

Les logiciels adaptés au circuit hardware (programmeur) sont *ICPROG* ou *WinPic800* qui sont distribué gratuitement sur le net. Bien entendu, pour chaque *PIC*, il faut une configuration comme par exemple, le choix du *COM*, l'oscillateur, activer ou désactiver le code de protection (*CP*) et le chien de garde (*WDT*).

Avant de passer à la programmation du *PIC*, le fichier en hexadécimal *.HEX* doit être établi.

IV.5.2 .1 Schéma de principe

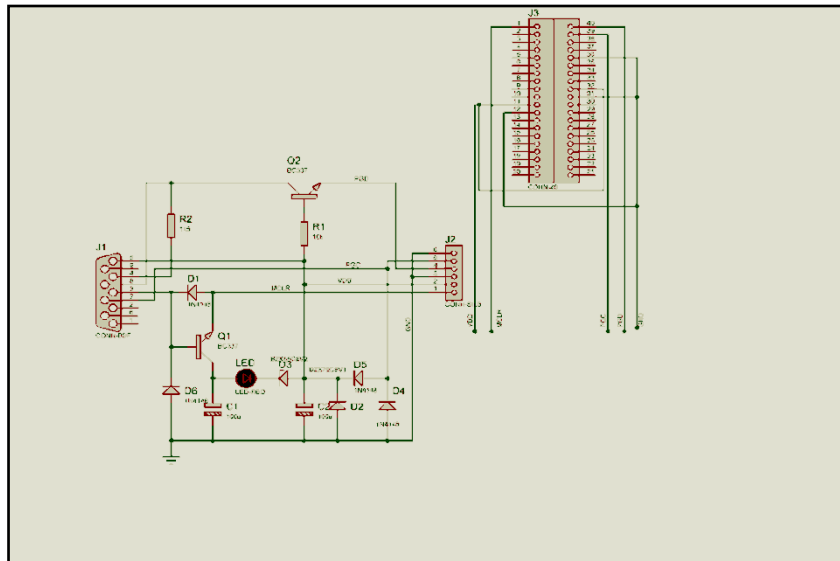


Figure IV.19: Schéma de principe du programmeur

L'aspect technique (courant, tension, état des composants...) a été volontairement oublié. Le sujet est plutôt centré sur la réalisation et l'utilisation du programmeur. C'est pour cela que le schéma de principe n'est pas accompagné de commentaires.

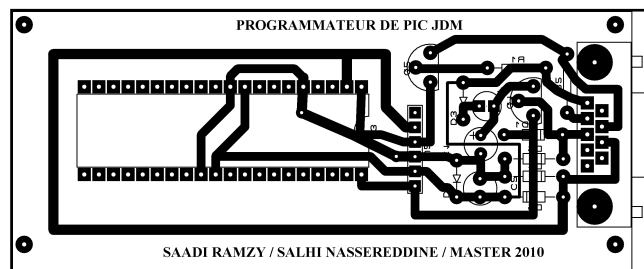


Figure IV.20 Schéma du programmeur

Composant	Valeur
Resistance 10k	1
Resistance 1k5	1
Transistor BC 337	2
Diode 1N4148	4
Diode zener 8.1V	1
Diode zener 5.2V	1
Connecteur db9	1
Capacité 100u	2
Connecteur DIL-40	1
Connecteur SIL-6	1

Tableau IV.4 Liste des composants

IV.6 Type de commande

Pour commander les six (06) moteurs du bras manipulateur ROB03 on a réalisé deux types de commande, commande en boucle ouverte (commande manuelle) et en boucle fermée (commande par les angles des organes du robot) .La commande en boucle fermée repose sur la mesure de la position angulaire de chaque articulation. Le dispositif assurant cette tâche est le potentiomètre rotatif ou à partir d'un mouvement relatif, il renseigne sur la position de l'arbre. L'information retournée pourra être traitée dans le module « convertisseur Analogique / Numérique » du *PIC 16F877A*.

IV.6.1 Le signal de commande PWM(MLI)

PWM veut dire **Pulse Width Modulation** ou en français **MLI Modulation à Largeur d'Impulsion** est un signal auquel on fait varier la tension sans modifier ni l'amplitude ni la fréquence mais la largeur de l'impulsion. En fait le signal PWM n'est pas continu, c'est un signal à impulsions (ressemble au signal carré). Il possède un état haut (un 1 logique) et un état bas (un 0 logique).

Voici un schéma représentant un signal PWM :

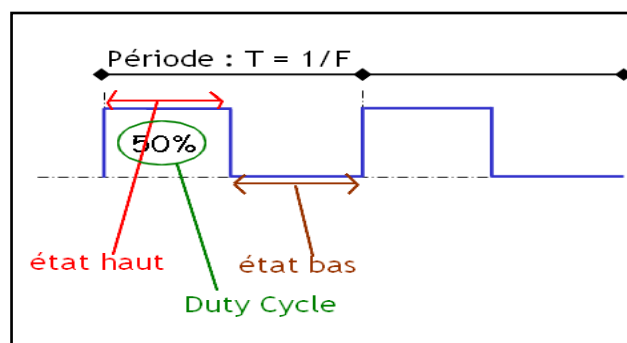


Figure IV.21 : schéma représentant un signal PWM

Dans notre montage Le module Capture/Compare/PWM (*CCP*) en mode *PWM de PIC16F877A* permet de générer un signal numérique pouvant être utilisé afin de reproduire un signal analogique situé entre 0 et 5V en faisant varier la durée du "Duty Cycle" et en se basant sur le *Timer2*. Cette tension active et désactive à son tour le pont *H* faisant varier analogiquement la tension d'alimentation des moteurs de 0 à 9V.

IV.7 L'organigramme général

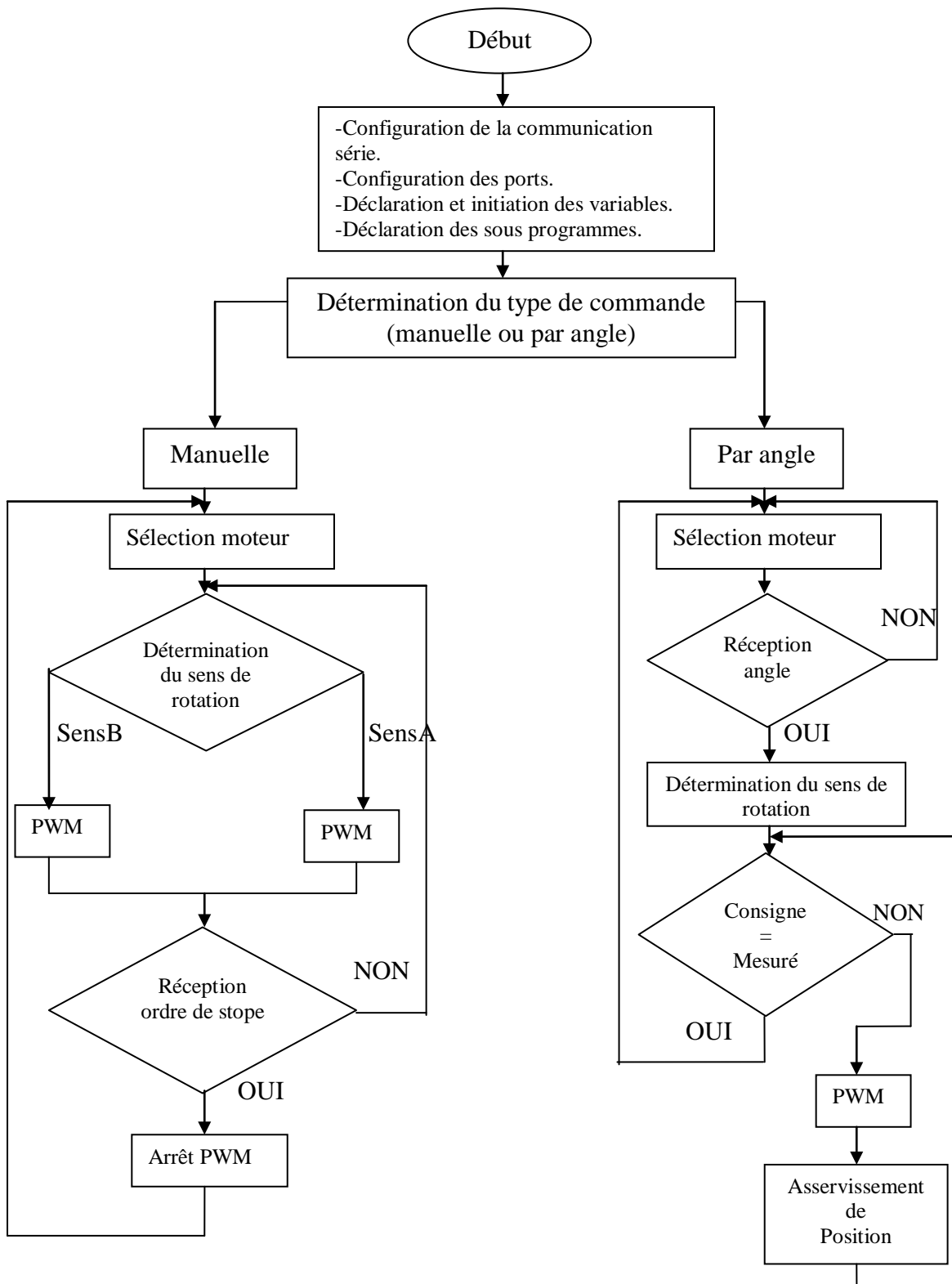


Figure IV.22 : Organigramme de fonctionnement.

IV.8 La carte d'interface

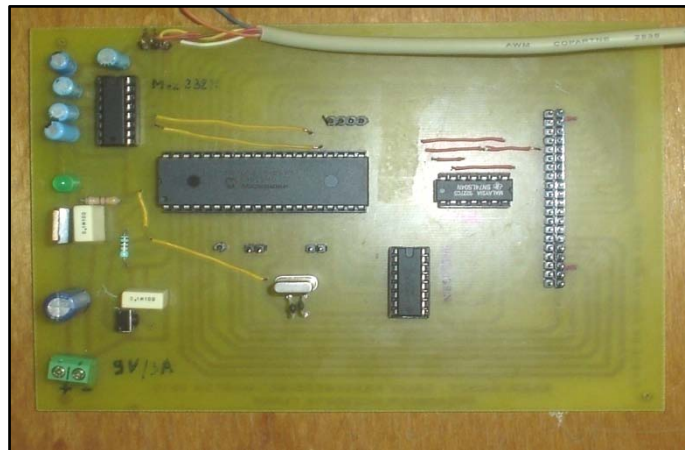


Figure IV.23: carte de commande

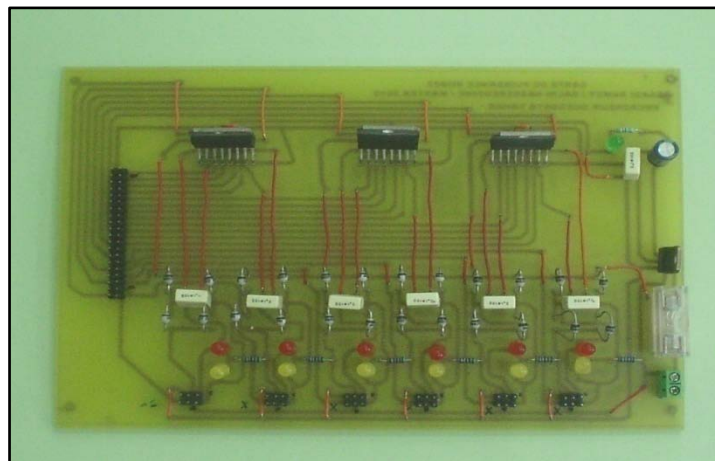


Figure IV.24: carte de puissance.

IV.9 Réalisation software

L'interface graphique constitue la troisième partie de notre travail après la réalisation de la carte ainsi que la programmation du microcontrôleur PIC 16f877A. Le chapitre présent donne une présentation générale des éléments du logiciel.

Pour la commande du bras, nous avons développé un logiciel avec un langage de programmation orienté objet le **C++ Builder v6**.

On a utilisé le produit **C++ Builder v6** pour le développement de notre logiciel car c'est un langage qui oriente l'objet, avec faciliter et la possibilité de réaliser un programme multitâches capable d'utiliser les périphérique du PC ainsi ces port série et port parallèle aussi de gérer une application autonome qui peut être installer sur n'importe quel système d'exploitation (Windows 98, xp...).

IV.9 Interface principale



Figure IV.25: Interface principale

Dans cette fenêtre, on trouve les différentes informations de notre projet et on choisit le type de commande, si on appuie sur le bouton « manuelle » on appelle l'interface de commande manuelle et lorsque on appuie sur le bouton « par angle » on appelle l'interface de commande par angle.

IV.9.1 Interface de commande manuelle

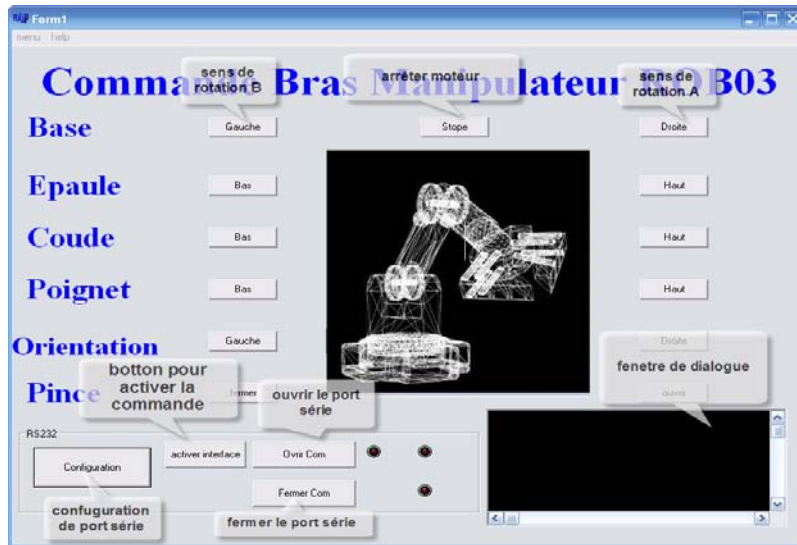


Figure IV.26 : Interface de commande manuelle

Dans l'interface de commande manuelle On peut commander les moteurs du bras en simple click sur le bouton du moteur désiré.

IV.9.1 Interface de commande par angle

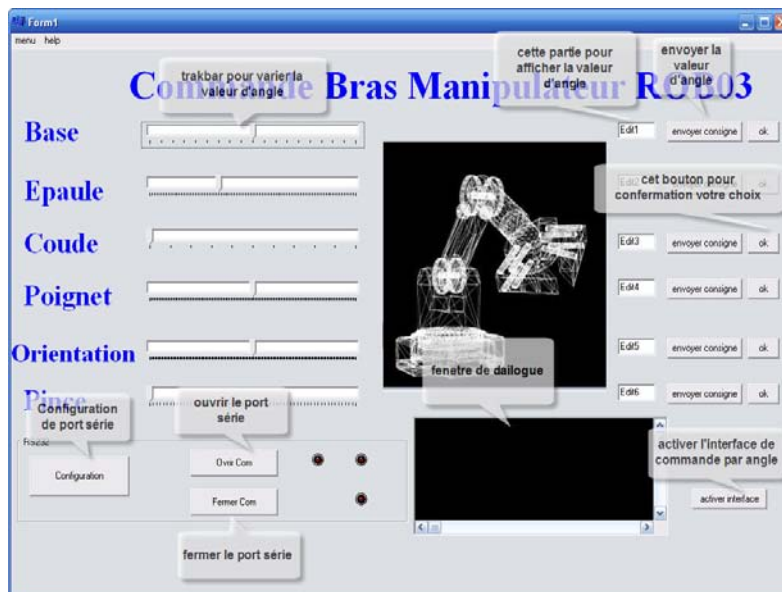


Figure IV.27 : Interface de commande par angle

La Figure (IV.28) montre un organigramme qui résume les principales étapes à respecter lors du développement de l'algorithme de l'interface de commande par angle

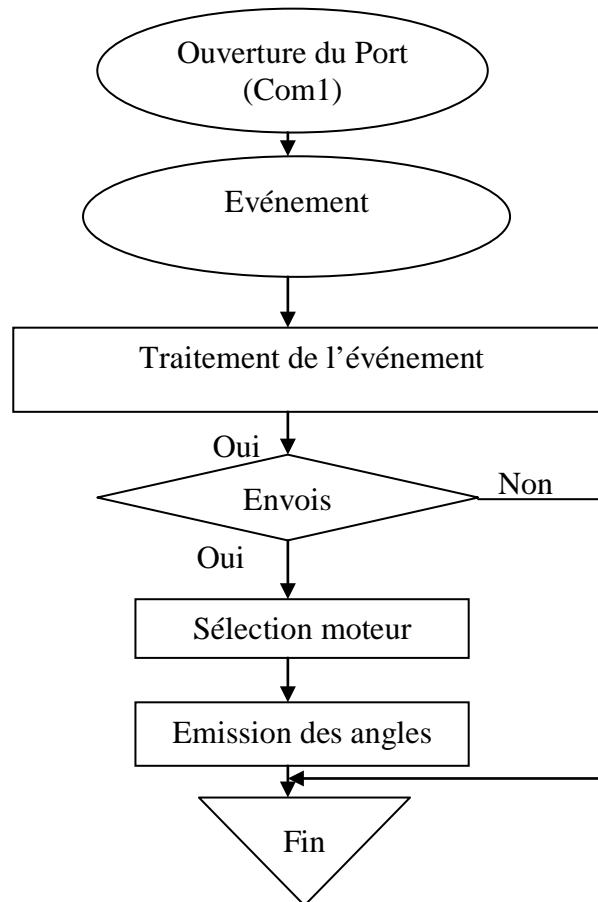


Figure IV.28: Organigramme de commande.

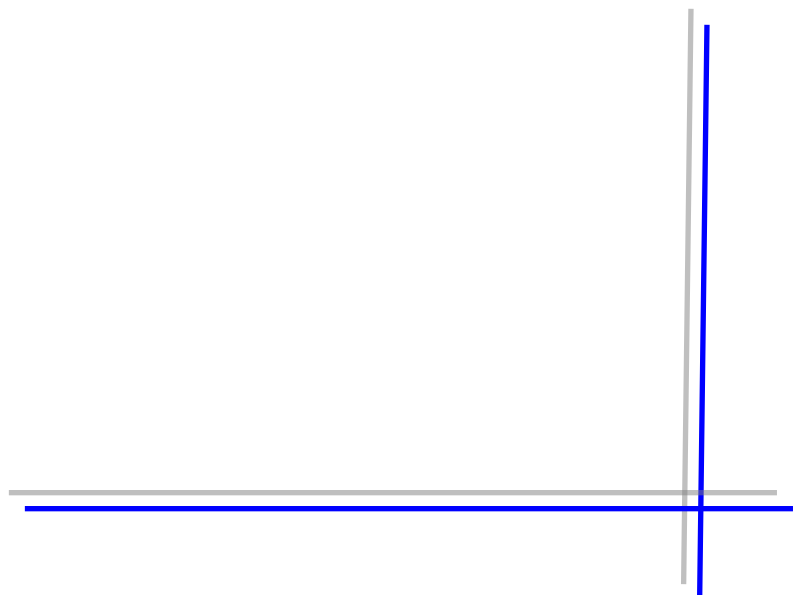
IV.10 Conclusion

Le présent chapitre constitue le cœur de notre projet. En effet, nous avons développé la carte d'interface *PC-ROB3* et une interface graphique permettant la communication *Homme/Machine*. Elle consiste à gérer le positionnement de notre robot en appliquant la commande par angle et manuelle qui utilise le port série (*COM*) de l'ordinateur.



Chapitre 5

Software



V.1 Introduction

Ce chapitre est consacré à la présentation de la différent logiciel qui nous utilisons dans notre projet, pour la simulation et le traçage de circuit imprimé en utilise le logiciel proteus v7 et pour la programmation du pic on utilise le pic c compiler, et pour flasher le programme dans le pic le logiciel utilisé est le icprog, et pour créer l'interface graphique on a utilisé le c++ Builder v6.

V.2 Présentation générale de proteus [12]

Le logiciel PROTEUS est composé de trois modules :

- L'éditeur de schéma ISIS
- Le simulateur LISA
- L'outil de conception de circuit imprimé ARES

- ISIS produit d'une part, une liste d'équipotentiels qui peut être utilisée par le simulateur LISA et l'outil de conception de circuit imprimé ARES, d'autre part, une liste de matériel et des rapports de contrôle des règles électriques.

- LISA est un ensemble de modules de simulation lié à ISIS. Le noyau de simulation PROSPICE est basé sur la version 3F5 du moteur SPICE publié par l'université de Berkeley.

- ARES est un module de conception de circuit imprimé compatible Windows, 98 2000 et XP. Il permet le placement des composants en mode automatique, manuel ou semi-automatique et le routage des liaisons sur plusieurs couches en mode automatique manuel ou semi-automatique.

V.2.1 L'éditeur de schéma ISIS

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

V.2.1.1. Interface utilisateur

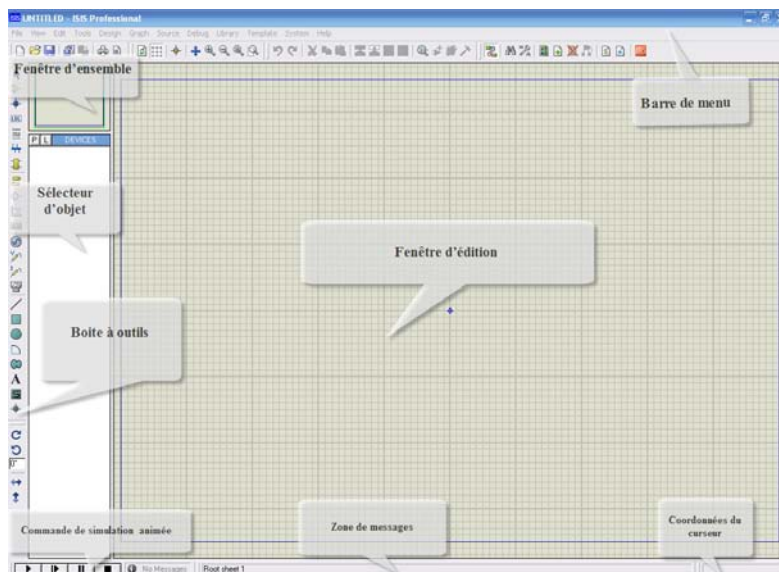


Figure V.1: Interface utilisateur ISIS

V.2.1.1.a. Fenêtre d'ensemble

Le cadre en bleu délimite l'espace de travail tel qu'il a été défini par la commande '*Définir taille des feuilles*' du menu '*système*'.

Le cadre en vert délimite la zone de travail, c'est à dire la partie du schéma visible dans la fenêtre principale.

- Vous pouvez déplacer cette zone de travail en pointant la souris sur la zone désirée de la fenêtre d'ensemble et en effectuant un clic gauche.
- Vous pouvez redéfinir la zone de travail dans la fenêtre d'ensemble en appuyant sur la touche majuscule '**shift**' du clavier, associée au déplacement de la souris en maintenant appuyé le bouton gauche.

V.2.1.1.b. Fenêtre d'édition

C'est dans cette fenêtre que vous éditez votre circuit. Elle ne représente que la partie du circuit qui est définie par le cadre vert dans la fenêtre d'ensemble.

- Vous pouvez redéfinir la zone travail à l'aide de la fenêtre d'ensemble ou en utilisant la commande '*Zoom*' du menu '*Affichage*'.

V.2.1.1.c. La boîte à outils

Elle est composée d'un ensemble **d'icônes** dont les fonctions seront détaillées ultérieurement et d'un sélecteur d'objet utilisé pour choisir les boîtiers, le style des pastilles, des traces, des traversées, etc....

V.2.1.1.d. Coordonnées du curseur

Les coordonnées déterminent la position du curseur par rapport à l'origine qui par défaut se trouve au centre de la fenêtre d'édition. [12]

Les coordonnées sont affichées en 1/1000 de pouce (th).

V.2.1.1.e. Organisation de la boîte à outils

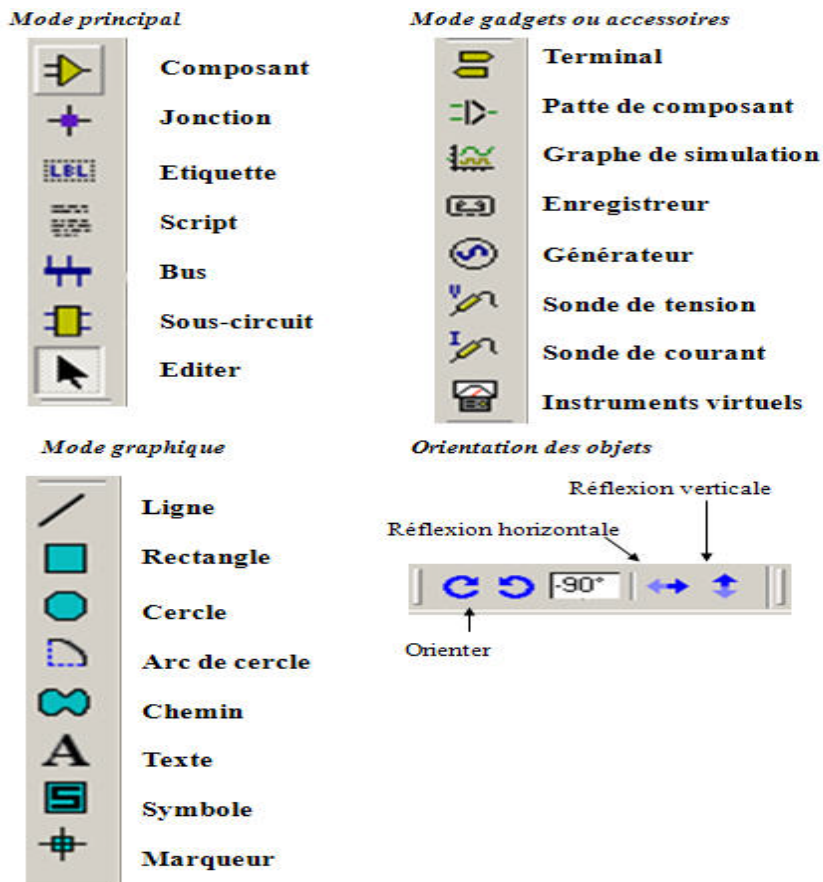


Figure V.2: Organisation de la boîte à outils isis

V.2.1.1. Etapes de la saisie de schéma

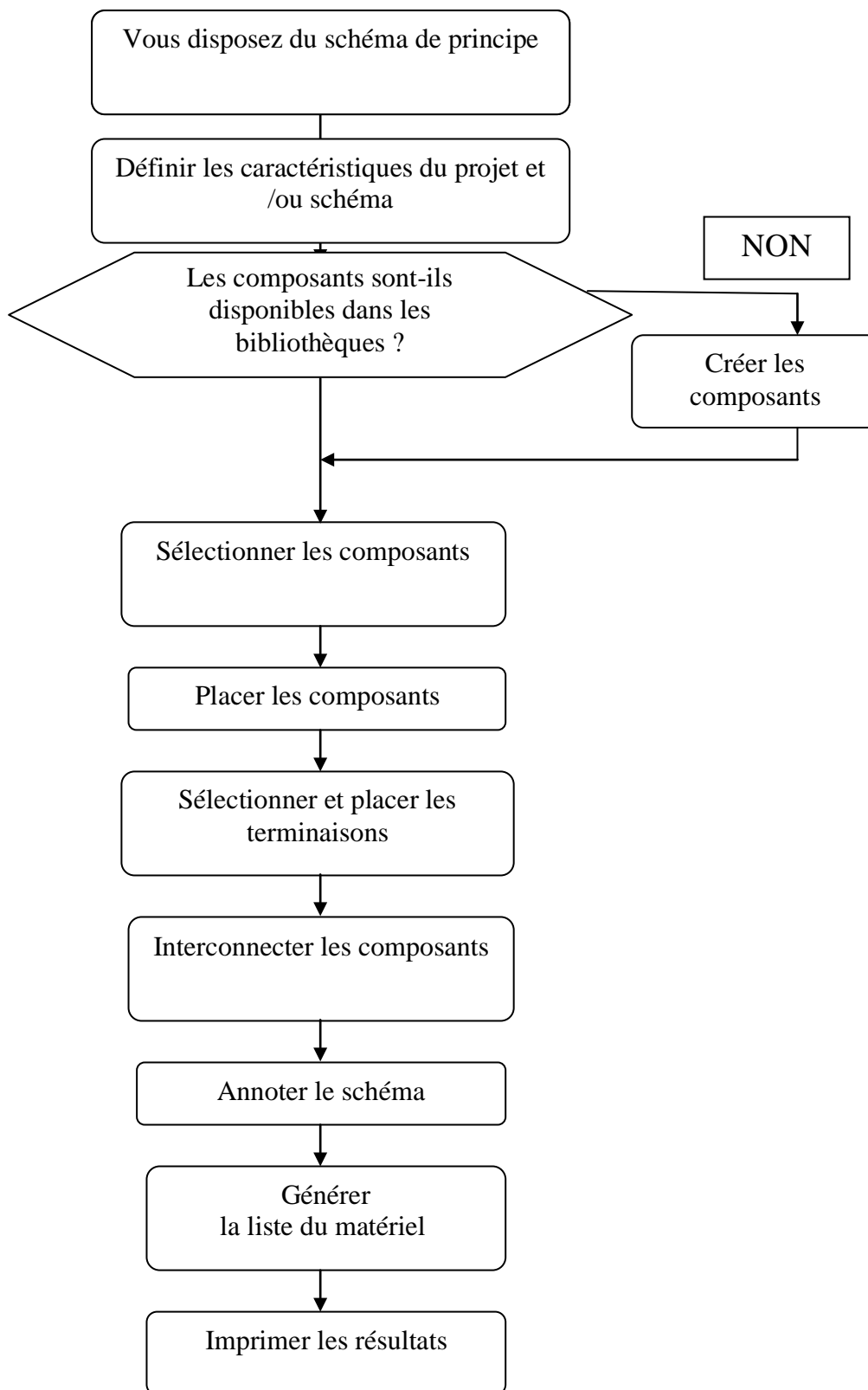


Figure V.3: organigramme des Etapes de la saisie de schéma

V.2.2 L'outil de conception de circuit imprimé ARES

ARES est un logiciel permettant le routage de cartes électronique en mode automatique ou manuel.

Il est possible d'utiliser ARES sans avoir au préalable créé de schéma dans ISIS. Cette fonctionnalité permet de réaliser très rapidement des circuits de faible complexité en plaçant les composants et traçant les pistes directement dans ARES.

Une fois les connections établies il est possible d'effectuer un routage automatique des pistes.

Dans ce logiciel vous pouvez également créer de nouveaux boîtiers et les placer dans une bibliothèque. Couplé avec ISIS vous avez un système complet qui vous permet d'effectuer avec un seul schéma toutes les étapes de la conception de cartes.

V.2.2.1 Présentation de ARES

L'aspect Général de ARES est très approchant de celui de ISIS.

1. Une barre de menu.
2. Une fenêtre principale dans laquelle vous allez créer votre routage mais aussi vos nouveaux composants.
3. Une fenêtre d'aperçu en haut à droite
4. Une palette en dessous et un sélecteur de boîtiers..
5. Un sélecteur de surface active, en bas à gauche.

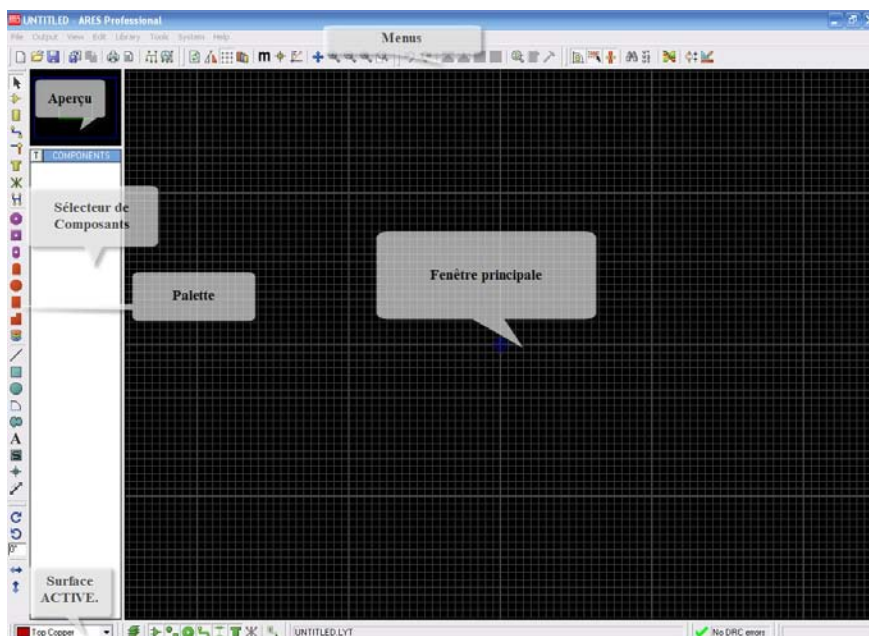


Figure V.4: fenêtre principale ARES

Les sélections, les zooms, les copies se font à l'identique de ISIS.

V.2.2.2 Création d'une carte

La première chose à faire avant de commencer un routage est de définir une carte aux dimensions de projet. Il est fortement conseillé de réaliser, si possible, le projet sur des formats standards (Demi Europe 100x80, Europe 100x160).

Pour cela placer en mode graphique et sélectionner "Board Edge" dans le sélecteur de surfaces. Sélectionner l'icône rectangle et dessiner un rectangle correspondant au contour de la carte.

V.2.2.3 Le placement des composants

V.2.2.3.a Placement automatique

Après avoir défini une carte vous pouvez placer les composants en mode automatique. Outils « Placement automatique » La fenêtre ci-contre apparaît. Vous pouvez choisir les composants à placer. Après validation les composants sont positionnés sur la carte.

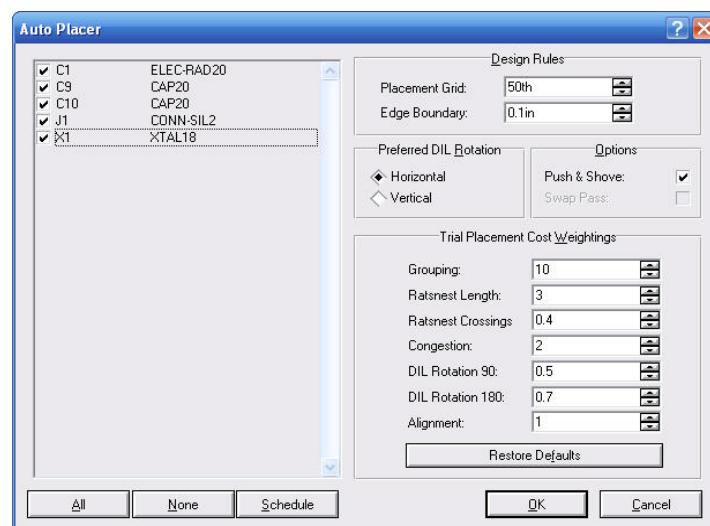


Figure V.5: fenêtre de placement automatique

V.2.2.3.b Placement Manuel

On peut rarement se contenter d'un placement automatique. En raison d'exigences dues au projet il est très souvent nécessaire de placer des composants à des endroits précis. Dans ce cas vous sélectionnez le composant dans la fenêtre *de sélecteur*. Vous orientez votre sélection au moyen des icônes appropriés. Vous cliquez dans la fenêtre de placement et sans relâcher le bouton de la souris vous déplacez votre fantôme de composant.

V.2.2.4 Le routage

V.2.2.4.a Routage automatique

Le routage automatique se lance dans le menu *Outils "Routeur automatique"*. Dans la fenêtre de configuration vous pouvez choisir de router tout le chevelu, ou une partie. Vous pouvez également déterminer les isolations à respecter.

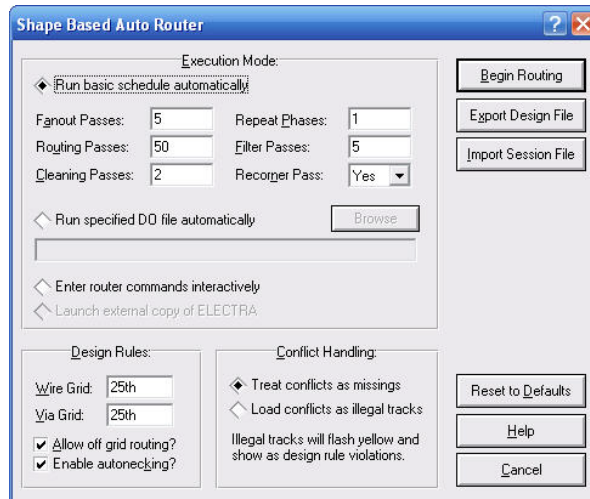


Figure V.6: fenêtre de routage automatique

V.2.2.4.b Routage manuel

Comme pour le placement automatique il est très rare qu'un routage automatique convienne sans modifications. Il est donc très utile de pouvoir modifier ou créer des pistes manuellement.

Choisissez l'icône *pistes* puis sélectionnez un guide (en vert). Par des clics successifs dessinez votre piste en partant d'une extrémité du guide. Le tracé se termine lorsque vous atteignez l'autre extrémité.

V.3 Le pic c compiler

CCS est un compilateur C pour les processeurs de la famille Micro Chip. Des fonctions intégrées permettent de développer le code de manière très aisée.

L'environnement intégré de développement C donne à l'utilisateur une méthode rapide de produire un code efficient par le biais du langage évolué C. Le compilateur comprend des fonctions intégrées comme `READ_ADC` qui lit une valeur provenant du convertisseur A/D du hardware PIC. Des fonctions telles que `INPUT` et `OUTPUT_HIGH` maintiennent proprement les registres trois-états. Les variables y compris des structures peuvent être directement mappées en mémoire tels que les ports I/O pour mieux représenter la structure du hardware en C. Des fonctions d'I/O Série permettent aux fonctions standard telles que `GETC` et `PRINTF` d'être utilisés comme I/O pour le RS-232. Les opérateurs du standard C et les

fonctions intégrées spéciales sont optimisés pour produire des codes efficaces pour les bits et les fonctions d'I/O.

Le fichier de sortie Hex et ceux du débogage sont sélectionnables et compatibles avec les émulateurs et programmeurs populaires y compris MPLAB pour le débogage au niveau source.

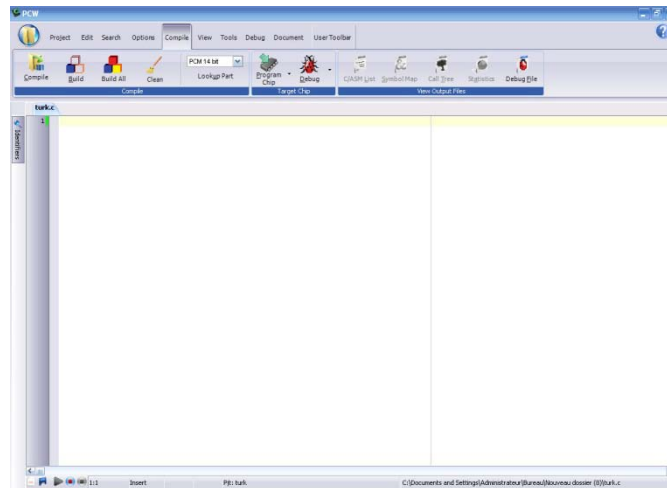


Figure V.6: fenêtre principale de pic c compiler

La démarche à suivre pour créer un projet avec le compilateur CCS :

1. Lancement de PIC C Compiler :

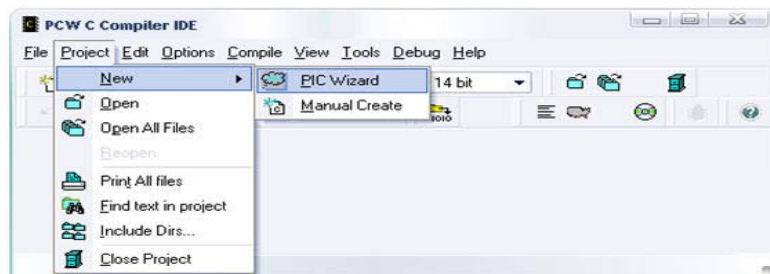


Figure V.7: Lancement de PIC C Compiler

2. Création d'un nouveau projet : project /new /PIC Wizard

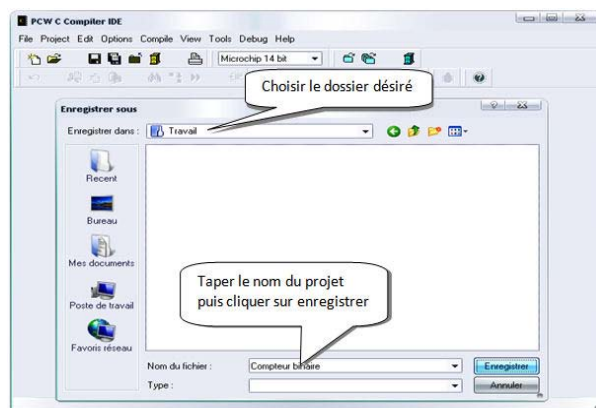


Figure V.8: Création d'un nouveau projet

3. Choix du dossier de travail et du nom du projet :

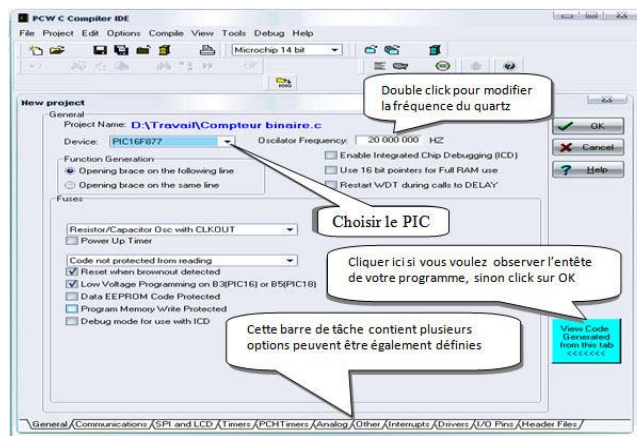


Figure V.9: Choix du dossier de travail et du nom du projet

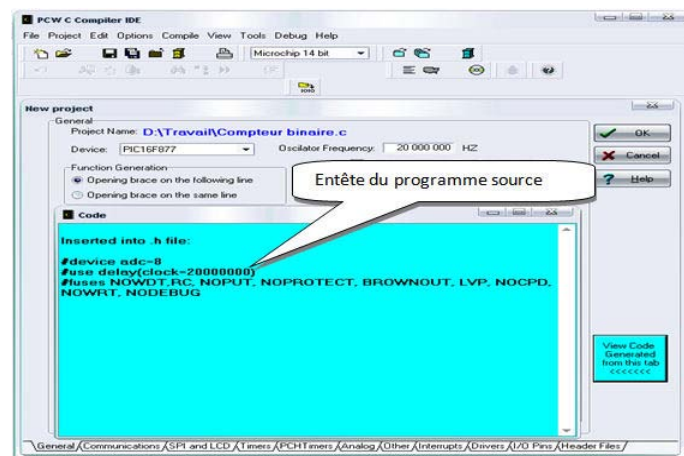


Figure V.10: entête de programme source

4. Choix des paramètres du projet :

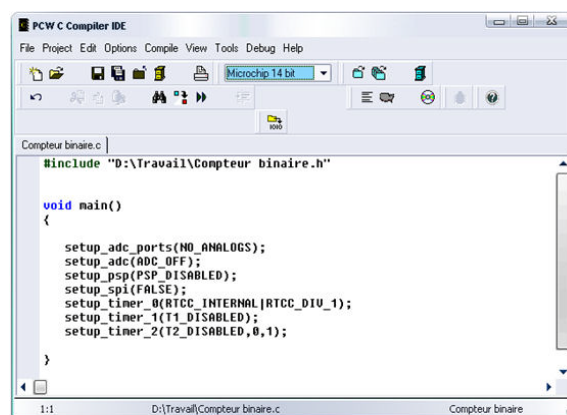


Figure V.11: paramètres du projet

5. Saisie du programme :

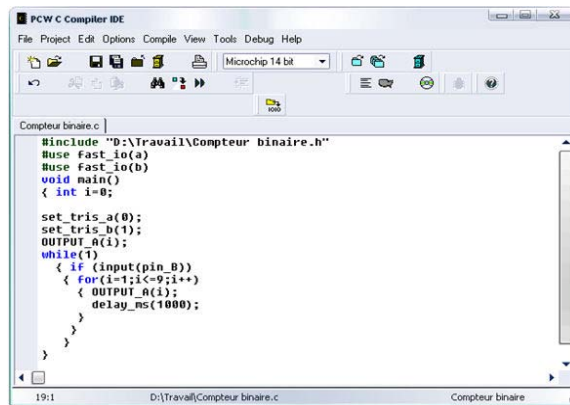


Figure V.12: exemple de programme

6. Compilation : compile F9

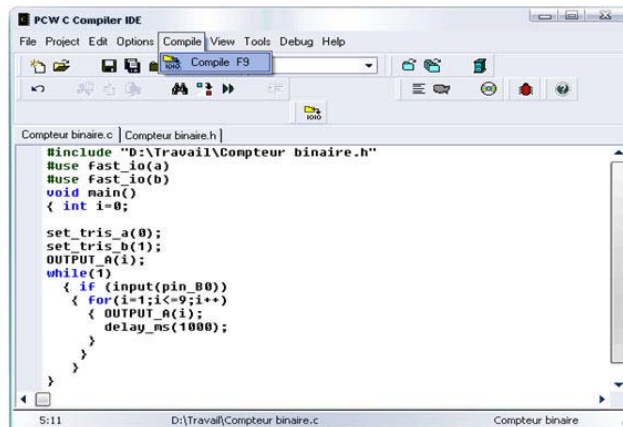


Figure V.13: Compilation de programme

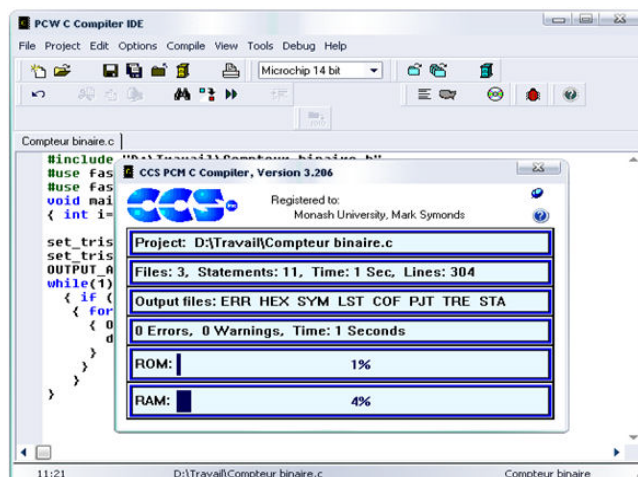


Figure V.14: fenêtre de compilation

V.3 L'ICPROG

Icprog est un programme permet de transférer un fichier compilé (hex) vers un pic ou un mémoire de type EEPROM.

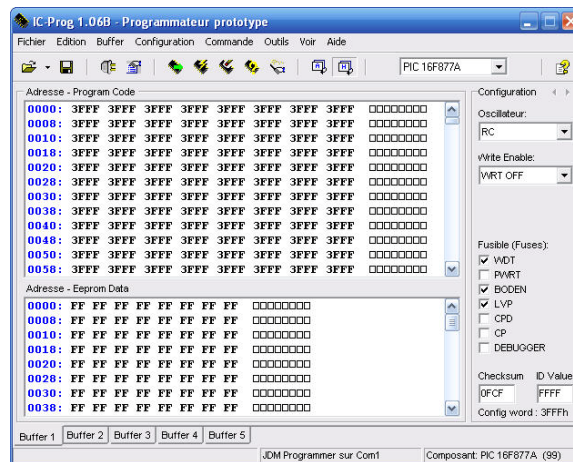


Figure V.15:fenêtre principale de icprog

V.3.1 Configuration icprog

Depuis le menu « configuration » puis « hardware » sélectionner selon la programmeur une des option dans la figure ci-dessous.

A titre d'information pour un programmeur de type série (par le port com de pc) la configuration nécessaire est dans l'écran figure V.14.

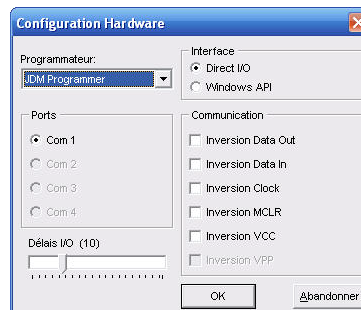


Figure V.16:fenêtre de configuration hardware

- Pour sélectionner un composant aller dans « configuration » puis « composant »

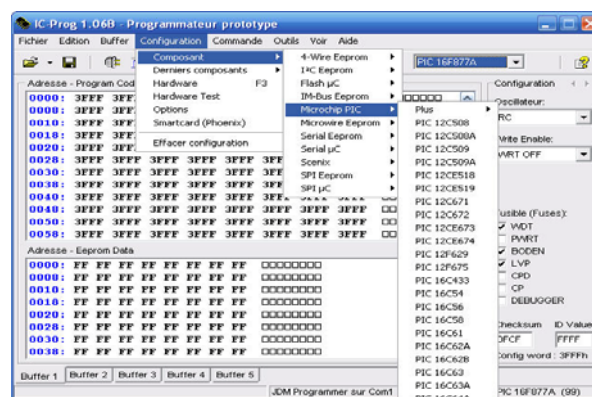


Figure V.17:fenêtre sélection de composant

Une fois le composant sélectionné il reste plus qu'à ouvrir le fichier et transférer.

- Il y'a encore la possibilité de modifier certaine option qui sont proposées avec les cases à cocher :

-CP (code de protect) : si cette option est cochée alors il ne sera plus possible de relire le programme de pic.

-WDT (watchdog timer) : le watchdog vérifier le programme ne s'est perdu dans une boucle sans fin.

-PWRT (Power up timer) : si cette option est cochée un timer interne provoquera un arrêt typique de pic pendant 72ms après la mise sous tension.

-Oscillateur : il ya quatre option cette liste déroulante :

- **Mode LP:** Ce mode est réservé si le quartz utiliser est compris entre 32Khz et 200Khz.
- **Mode XT:** si la fréquence de quartz utiliser comprises entre 455Khz et 4Mhz.
- **Mode HS:** si la fréquence de quartz utiliser comprises entre 8Mhz et 20Mhz.
- **Mode RC:** ce mode est réservé si l'oscillateur est un circuit RC.

- Et maintenant il ne reste plus qu'à transférer le fichier : Menu « commande » puis « tous programmer »

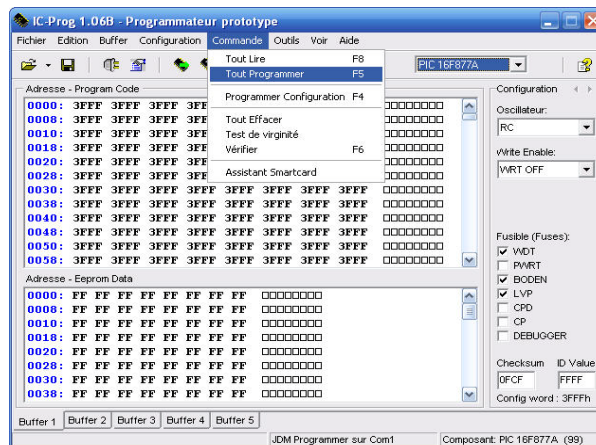


Figure V.18: programmer un composant.

V.5 LE C++ Builder

C++ Builder est le nouvel environnement de développement basé sur C++ proposé par Borland. Fort du succès de Delphi, Borland a repris la philosophie, l'interface et la bibliothèque de composants visuels de ce dernier pour l'adapter depuis le langage Pascal Orienté Objet vers C++ répondant ainsi à une large fraction de programmeurs peu enclins à l'utilisation du Pascal qu'ils jugent quelque peu dépassé. [8]

V.5.1. Environnement RAD

Tout d'abord C++ est un outil RAD, c'est à dire tourné vers le développement rapide d'applications (Rapid Application Development) sous Windows. En un mot, C++ Builder permet de réaliser de façon très simple l'interface des applications et de relier aisément le code utilisateur aux événements Windows, quelle que soit leur origine (souris, clavier, événement système, etc.)

Pour ce faire, C++ Builder repose sur un ensemble très complet de *composants visuels* prêts à l'emploi. La quasi totalité des contrôles de Windows (boutons, boîtes de saisies, listes déroulantes, menus et autres barres d'outils) y sont représentés, regroupés par famille. Leurs caractéristiques sont éditables directement dans une fenêtre spéciale intitulée *éditeur d'objets*. L'autre volet de cette même fenêtre permet d'associer du code au contrôle sélectionné.

V.5.2. L'interface de C++ Builder

La Figure V.17 représente un exemple typique de l'interface de C++ Builder au cours d'une session de travail.

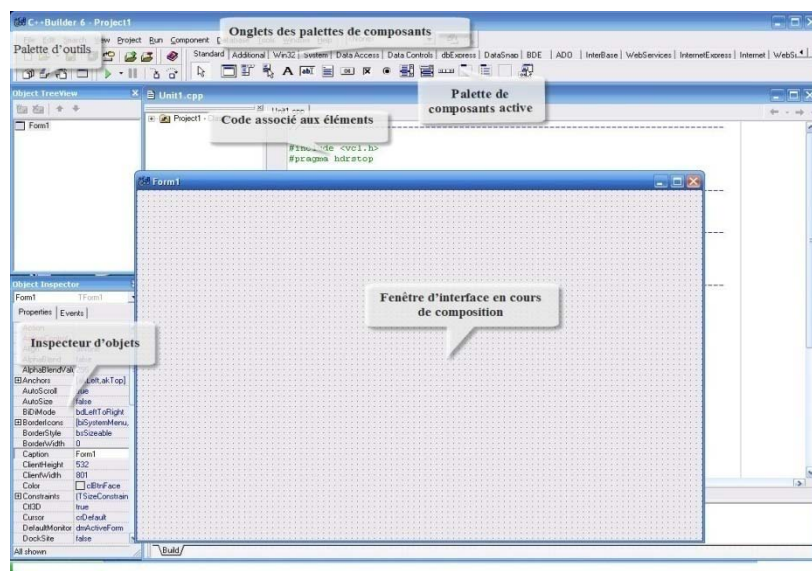


Figure V.19 : L'interface de C++ Builder

On peut toutefois distinguer plusieurs grandes parties :

- □ La classique barre de menu
- □ La barre d'outils qui se décompose en 2 grandes parties :
 - La palette d'outils permettant d'effectuer les opérations les plus courantes (sauvegarde, ouverture de fenêtres, etc.).
 - Les palettes de composants disposées accessibles par des onglets.
- L'inspecteur d'objets qui permet de manipuler les propriétés des composants et d'associer du code à leurs événements
- Les fenêtres d'interface créées par l'utilisateur. Ce sont les fenêtres de l'application en cours de création, elles portent ici le nom de *fiche* (form en anglais). Certaines peuvent être cachées, d'autres présentes à l'écran. On verra que dans la plupart des cas, leur position (ainsi que leurs autres caractéristiques géométriques) à l'exécution sont le reflet exact de ce qu'elle étaient lors de la conception.
- L'éditeur de code. A chaque fiche ouverte correspond deux fichiers source (un fichier .h et un fichier .cpp) qui sont éditables dans cette fenêtre avec mise en évidence de la syntaxe.

V.5.3. Création d'une application simple C++ Builder

C++ Builder permet de créer différents types de module très simplement en se laissant guider par des experts. Toutefois, il est possible de demander à créer une application simple en activant l'option **Nouvelle application** du menu **Fichier**. Les éléments automatiquement créés sont les suivants : une fiche nommée **Form1** ainsi que les fichiers associés **Unit1.cpp** et **Unit1.h**. Notons au passage que la terminologie unit est directement calquée sur celle chère à Delphi et que les fonctionnalités ainsi créées sont toujours renommables après coup.[8]

V.5.4 Conclusion :

Ce chapitre prend une partie très important dans le projet car l'utilité offerte par les programmes utilisés qui nous aident à dessiner les schémas électroniques et les voir, vérifier, et simuler. Comme nous avons la main pour consulter l'effet du programme avant de l'envoyer au microcontrôleur. N'oublions pas que le C++ builder important ; plus qu'il nous donne l'environnement pour construire l'interface graphique il facilite la communication entre PC, carte de commande et l'utilisateur.

Conclusion générale

Finalement, le *ROB3* est en état de marche. Ceci est réalisé au prix d'immense sacrifice, tant sur le plan intellectuel que matériels.

Ainsi, Pratiquement nous avons consacré notre temps aux tâches suivantes :

➤ **Aspect mécanique :**

- Etude de la robotique du point de vue réalisations
- Etude les différents types des bras manipulateur.

➤ **Aspect électronique :**

- La réalisation des cartes électroniques pour la commande de notre robot et la communication avec le PC.

➤ **Aspect informatique :**

- Programmation du robot pour la commande des moteurs à courant continu utilisés et pour l'exécution des commandes transmises par le PC.
- Réalisation d'une interface graphique en C++ Builder permettant de commander le robot soit en mode de commande manuelle soit en mode de commande par angle.

En effet, ce projet de réalisation nous a été une expérience très fructueuse, bénéfique et enrichissante. Il nous a permis d'exploiter les différentes notions théoriques vues durant notre période de formation. Il nous permet également de mettre le premier pas dans le monde pratique de la robotique. Ce projet nous a effectivement permis de proposer des schémas de commande électrique du *ROB3* ainsi que la réalisation des cartes électroniques associées. Sur le plan software, ce projet nous a mis directement en contact avec des langages de programmation puissants tel que le langage *C*, ainsi que différents logiciels spécialisés dans ce domaine, à savoir, *PROTEUS*, *ICPROG* et le *BUILDER*.

En fin nous espérons que notre travail va être amélioré en futur par d'autres étudiants.

Comme perspectives nous proposons :

- Emploi de la géométrie inverse pour piloter le *ROB3*
- Commander le robot par automate programmable
- La commande du bras à distance via *Internet*.
- La commande en 3D par des caméras.
- Utilisation les deux cartes réalisées pour commander d'autres systèmes.

Bibliographie

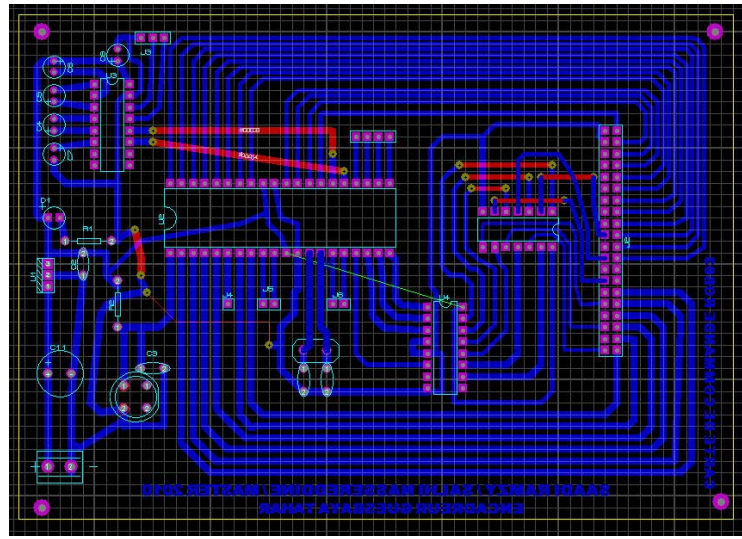
- [1] : Catalogue du bras de robot *ROB3* livré par *EUROBTEC*
- [2] : Wisama Khalil, Etienne Dombre « Modélisation identification et commande des robots »
-2 édition revue et augmenté, Edition : Lermes Science publication
- [3] : B.Kainka « J’exploite Les Interfaces de Mon PC » ,Edition :Publitronic.
- [4] :Kent Reisdorph « Le programmeur apprenez C++ builder en 14 jours » ,
Edition :S&sM.
- [5] : P. Fissette, H.Buyse, J.C.Samin « Introduction à la robotique », 29 juin 2009
>><http://meca2732.pdf>
- [6] : Jean-Louis Boimond « ROBOTIQUE », ISTIA, Université Angers
>> www.istia.univ-angers.fr/~boimond/Cours_robotique.pdf
- [7] : Jacques Gangloff « Cours de Robotique »
>> http://eavr.u-strasbg.fr/wiki/upload/a/a4/Cours_rob_intro.pdf
- [8] : « Introduction à C++ Builder », Ourse Blanc des Carpates.Isima 1988-1999
>> http://eric.univ-lyon2.fr/~ricco/cours/cours/intro_ccp_builder.pdf
- [9] : Mansouri Mohamed Sofiane, Maziz Mohamed « Réalisation d’une carte de commande d’un bras de robot 5 axes » , *USTO,2008/2009*
- [10] : Marzine Abd ElKarim « Commande du bras de robot *ROB3* par microcontrôleur »,
Université Ibn Khaldoun – Tiaret-.
- [11] : O.Ahcene, NEMMICHE Belkacem, BELHAMITI hadj Afif, BERRAHO Ismail,
« Commande en 3D d’un bras robot à 5 DL par positionnement en temps réels par deux caméras », Université ABDELHAMID IBN BADIS De Mostaganem.
- [12] : Carlos Valente « L'EDITEUR ISIS » .

Sites internet

- [13] : <http://www.codes-sources.com/>
- [14] : <http://www.datasheetcatalog.net/>
- [15] : <http://www.developpez.com/>

Annexe

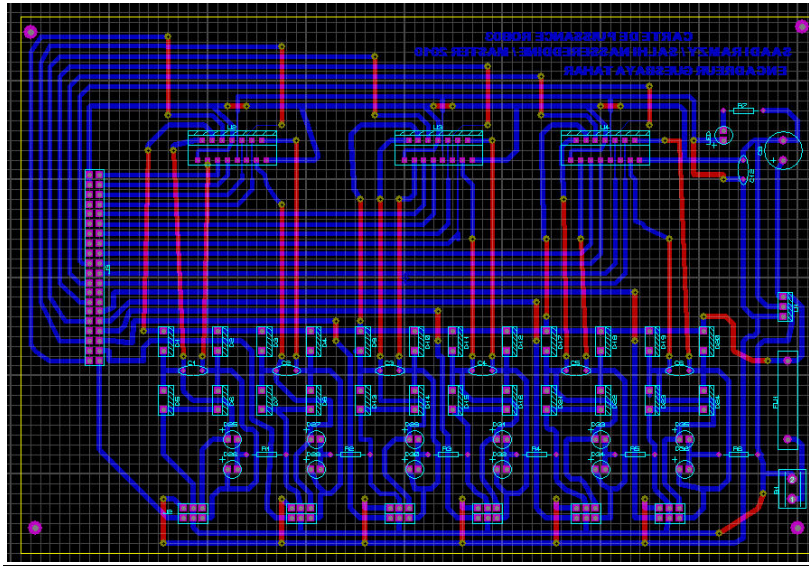
1-Circuit imprimé de la carte de commande



Liste des composants :

Composant	Nombre
PIC 16F877A	1
Quartz 20 MHZ	1
MAX232	1
74HC138	1
74HC04	1
Régulateur LM7805	1
Résistance 500 Ohm	1
Résistance 1 K Ohm	1
Capacité de 100nf	2
Capacité BIAS 1 μ F 50V	4
Capacité BIAS 10 μ F 50V	1
Capacité BIAS 1000 μ F 16V	1
Led verte	1
Connecteur 2*20	1
Bouton poussoir	1
DB9 avec son cable	1

1-Circuit imprimé de la carte de puissance



Liste des composants :

Composant	Nombre
L298	3
BYT 28-50	24
Fusible	1
Régulateur <i>LM7805</i>	1
Résistance 50 Ohm	1
Capacité de 100nf	7
Capacité BIAS 1000 μ F 16V	1
Led verte	1
Led rouge	6
Led jaune	6
Connecteur 2*20	1
Connecteur 2*3	6

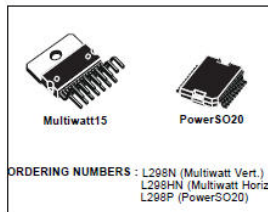
1- Data Scheet PIC 16F877A



L298

DUAL FULL-BRIDGE DRIVER

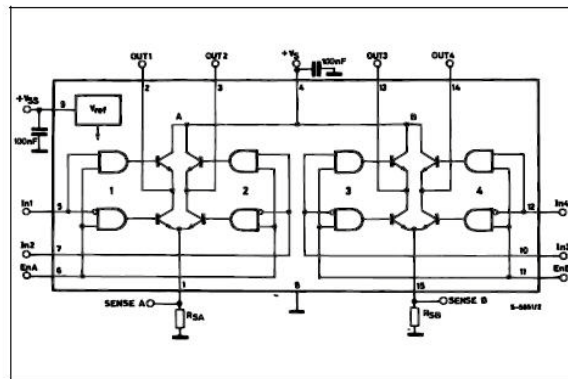
- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)



DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM



4- Data ROB03

Die Anatomie des ROB3 erlaubt, ausgehend von der Grundposition (Abb. 3.3) folgende Winkelbereiche:

Gelenk	Winkel	erreichbare Bereiche	Auflösung
Achse 1	q1	+80°..0°...-80°	0...255
Achse 2	q2	+70°..0°...-30°	0...255
Achse 3	q3	0°...-100°	0...255
Achse 4	q4	+100°..0°...-100°	0...255
Achse 5	q5	+100°..0°...-100°	0...255
Greifer	-	0...60mm	0...255

Tabelle 3.1 : Winkelbereiche des ROB3

Der Winkelbereich der Achse 2 wurde vom Verfasser im Visual Basic – Programm +70°...-15° begrenzt. Der Grund war die mögliche Kollisionsgefahr des Roboterarmes (siehe dazu Abbildung 3.4).

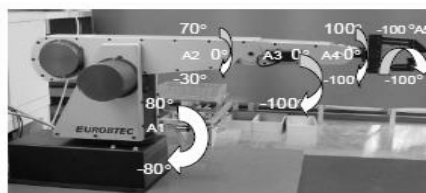


Abbildung 3.3 : Grundstellung des Trainingsroboters mit ROB 3

5- Sous programme de réception

```
void get_string(char* s, unsigned int8 max) {
```

```
unsigned int8 len;
char c;
--max;
len=0;
do {
    c=getc();
    if(c==8) {
        if(len>0) {
            len--;
            putc(c);
            putc(' ');
            putc(c);
        }
    } else if ((c>=' ')&&(c<=~'))
        if(len<=max) {
            s[len++]=c;
            putc(c);
        }
    } while(c!=13);
s[len]=0;
}
```