

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



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

## Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Spécialité : **Commande des systèmes**  
Filière : **Génie électrique**

*Présenté par*  
**Rafik MANSOURI**  
**Ali KHELID**

Mémoire dirigé par **Ourida HEDJEM.**

### Thème

## **Conception et réalisation d'un bras manipulateur à trois degrés de liberté commandé par la carte Arduino UNO**

*Mémoire soutenu publiquement le 14/07/2015 devant le jury composé de :*

**M ACHOUR. H**  
MCB, UMMTO, Président

**Mme HEDJEM. O**  
MAA, UMMTO, Encadreur

**M BOUDJEMAA. F**  
MAA, UMMTO, Examineur

**M CHELLI. T**  
MAA, UMMTO, Examineur

## Remerciements

---

Nous tenons à remercier profondément Dieu, le tout puissant de nous avoir donné le courage, la volonté, ainsi que la santé pour réaliser ce modeste travail.

Nous tenons à exprimer nos vifs remerciements et nos sincères reconnaissances à **Mme HEDJEM Ourida**, pour sa disponibilité, ses conseils judicieux, ses directives et ses orientations concernant notre projet de fin d'étude, tout en nous accordant sa confiance et en nous faisant profiter de sa large expérience tout au long de la réalisation de ce projet de fin d'étude.

Nous tenons à exprimer nos sincères remerciements à Mr le chef de département **Mr BENSIDHOUM Med Tahar** pour ses conseils et ses orientations,  
**Mr ACHOUR Hakim** pour ses conseils et ses orientations ainsi son aide précieuse.  
**Mr SAIDI Kheir eddine** pour ses conseils et ses orientations ainsi son aide précieuse.

Nous remercions les membres de jury qui ont fait l'honneur de participer au jugement de ce travail.

Nous tenons à remercier toutes les personnes qui ont contribué au bon déroulement de ce projet de fin d'étude :

- ❖ A tous les professeurs de la faculté génie électrique pour leurs conseils et leurs orientations.
- ❖ Aux personnels du département.
- ❖ A nos parents et à toute la famille **MANSOURI** et **KHELID**
- ❖ A nos frères et sœurs.
- ❖ A tous nos amis.

Et enfin nos remerciements à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce projet de fin d'étude.

## Dédicaces

---

### **Je dédie ce modeste travail à :**

- Mes très chers parents qui ont fait de moi ce que je suis aujourd'hui et qui ont veillé de guider mes pas durant toute ma vie par leurs aides, leur grands émotions, leur sacrifices, et leur soutien et encouragement pendant la réalisation du projet de fin d'étude.
- Ma très chère sœur : Chahira
- Mes très chers frères : Aziz, Arezki, Merzouk, Mourad, Youcef
- Mon binôme : Rafik
- Mes amis : Djamel, Mdjid, Fateh, Nourdine, Sofiane, Nabil, Ahmed, Mokrane, Younes  
Et Mohand
- Toute la promotion Automatique 2015

**ALI**

## Dédicaces

---

### **Je dédie ce modeste travail à :**

- Mes très chers parents qui ont fait de moi ce que je suis aujourd'hui et qui ont veillé de guider mes pas durant toute ma vie par leurs aides, leur grands émotions, leur sacrifices, et leur soutien et encouragement pendant la réalisation du projet de fin d'étude.
- Ma très chère sœur : Feroudja ainsi son époux youcef et leur petites filles : Chaima, Kenza, Ouacila
- Mes très chers frères : Mohand, Malik, Anis
- Ma très chère compagne Karima
- Mon binôme : Ali
- Mes amis de la Cité Abane Ramdane : Mokrane, Nacer, Djafar, Hacene, Lyes, Aziz, Larvi, M'hamed, Yacine, sofienne, et tous les gens de la Cité Abane Ramdane
- Mes amis de la chambre M10 : Abedelah (chikh), Brahime (putchi), said (achandiw), Messaoud(l'américain), Hakim, Ghiles(ihema), Mourad(si lhouass), Nassim(agoubli),ferhat(nessema), Si dali, Rachid, Nabil, Yacine, Kamel, Amrane, Nacer et la chambre M15(Azedine, Smail,Mensour, Adel, Idir) et la chambre C35 (Anis, Embarek(damba), Belaid (Bily),Sofienne,Lyes, Smail(ami said), Hakim)
- Mes amis de la promotion : Djamel, Mdjid, Fateh, Nourdine, Sofiane et Nabil
- Mes amis agents de Hasnaoua : Aziz, Kamel et Ahcene
- Toute la promotion Automatique 2015

**RAFIK**

# Sommaire

---

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

## Chapitre 1 : Généralités sur la robotique et les servomoteurs

Partie 1 : Généralités sur la robotique .....	Erreur ! Signet non défini.
1.1. Introduction.....	Erreur ! Signet non défini.
1.2. Définition et historique .....	Erreur ! Signet non défini.
1.3. Les différentes catégories des robots .....	Erreur ! Signet non défini.
1.4. Vocabulaire utilisé dans le domaine de la robotique .....	Erreur ! Signet non défini.
1.5. Caractéristiques des robots.....	Erreur ! Signet non défini.
1.5.1. Description de sa géométrie .....	Erreur ! Signet non défini.
1.5.2. Volume de travail .....	Erreur ! Signet non défini.
1.5.3. Précision / Répétabilité .....	Erreur ! Signet non défini.
1.5.4. Performances dynamiques.....	Erreur ! Signet non défini.
1.5.5. Charge utile .....	Erreur ! Signet non défini.
1.6. Les différents types de robots.....	Erreur ! Signet non défini.
1.6.1. Les robots SCARA (Sélective Compliance Articulated) .....	Erreur ! Signet non défini.
1.6.2. Les robots cylindriques.....	Erreur ! Signet non défini.
1.6.3. Les robots sphériques.....	Erreur ! Signet non défini.
1.6.4. Les robots Cartésiens .....	Erreur ! Signet non défini.
1.6.5. Les robots parallèles.....	Erreur ! Signet non défini.
1.6.6. Les robots anthropomorphes.....	Erreur ! Signet non défini.

# Sommaire

---

Partie 2 : Généralités sur les servomoteurs .....	Erreur ! Signet non défini.
1.7. Présentation .....	Erreur ! Signet non défini.
1.8. Utilisation .....	Erreur ! Signet non défini.
1.9. Variété .....	Erreur ! Signet non défini.
1.10. Fonctionnement .....	Erreur ! Signet non défini.
1.11. Rôle du potentiomètre .....	Erreur ! Signet non défini.
1.12. Les avantages et les inconvénients .....	Erreur ! Signet non défini.
1.13. Conclusion .....	Erreur ! Signet non défini.

## Chapitre 2 : Modélisation géométrique des robots manipulateurs

2.1. Introduction.....	Erreur ! Signet non défini.
2.2. Description géométrique d'un bras manipulateur.....	Erreur ! Signet non défini.
2.3. Espace des coordonnées .....	Erreur ! Signet non défini.
2.4. Notations .....	Erreur ! Signet non défini.
2.5. Orientation d'un solide.....	Erreur ! Signet non défini.
2.5.1 Orientation d'un point appartenant au solide S .....	Erreur ! Signet non défini.
2.6. Transformation des repères .....	Erreur ! Signet non défini.
2.7. Matrices homogènes .....	Erreur ! Signet non défini.
2.8. Application de la matrice de transformation homogène.....	Erreur ! Signet non défini.
2.8.1. Matrice de transformation de translation .....	Erreur ! Signet non défini.
2.8.2. Matrice de transformation de rotation.....	Erreur ! Signet non défini.

# Sommaire

---

2.9. Modélisation géométrique .....	Erreur ! Signet non défini.
2.10. Convention de Denavit-Hartenberg(DH) .....	Erreur ! Signet non défini.
2.11. Modèle géométrique direct .....	Erreur ! Signet non défini.
2.12. Modèle géométrique inverse .....	Erreur ! Signet non défini.
2.13. Modélisation du robot à commander .....	Erreur ! Signet non défini.
2.13.1. Description du robot à commander .....	Erreur ! Signet non défini.
2.13.2. Modèle géométrique direct du robot à commander .....	Erreur ! Signet non défini.
2.13.3. Modèle géométrique inverse du robot à commander .....	Erreur ! Signet non défini.
2.14. Conclusion .....	Erreur ! Signet non défini.

## Chapitre 3 : Conception et réalisation

Partie 1 : Conception matérielle et logicielle .....	Erreur ! Signet non défini.
3.1. Introduction.....	Erreur ! Signet non défini.
3.2. Arduino UNO .....	Erreur ! Signet non défini.
3.2.1 Vue d'ensemble.....	Erreur ! Signet non défini.
3.2.2. Alimentation.....	Erreur ! Signet non défini.
3.2.3. La mémoire.....	Erreur ! Signet non défini.
3.2.4. Les entrées et sorties .....	Erreur ! Signet non défini.
3.2.5. Communication .....	Erreur ! Signet non défini.
3.2.6. Programmation .....	Erreur ! Signet non défini.
3.3. Commande PWM (Pulse Width Modulation).....	Erreur ! Signet non défini.

# Sommaire

---

3.4. Les Capteurs .....	Erreur ! Signet non défini.
3.4.1. Le capteur de présence de pièce .....	Erreur ! Signet non défini.
3.4.2. Le capteur de présence de pièce sur la pince .....	Erreur ! Signet non défini.
3.5. Afficheur à cristaux liquides (LCD).....	Erreur ! Signet non défini.
3.5.1. Introduction.....	Erreur ! Signet non défini.
3.5.2. Schéma fonctionnel.....	Erreur ! Signet non défini.
3.5.3. Brochage.....	Erreur ! Signet non défini.
3.5.4. La mémoire.....	Erreur ! Signet non défini.
3.5.5. Commande d'un afficheur LCD.....	Erreur ! Signet non défini.
3.5.6. Branchement de l'afficheur LCD avec Arduino UNO :.....	Erreur ! Signet non défini.
3.6. Simulation.....	Erreur ! Signet non défini.
3.7. Choix des actionneurs .....	Erreur ! Signet non défini.
3.8. Tache du bras manipulateur.....	Erreur ! Signet non défini.
Partie 2 : Réalisation pratique .....	Erreur ! Signet non défini.
3.9. Introduction.....	Erreur ! Signet non défini.
3.10. Structure mécanique du bras manipulateur .....	Erreur ! Signet non défini.
3.11. Différentes parties du robot.....	Erreur ! Signet non défini.
3.11.1. La base.....	Erreur ! Signet non défini.
3.11.2. Premier corps .....	Erreur ! Signet non défini.
3.11.3 Deuxième corps.....	Erreur ! Signet non défini.
3.11.4. Troisième corps .....	Erreur ! Signet non défini.
3.11.5. L'organe terminal .....	Erreur ! Signet non défini.



# Sommaire

---

3.11.6. Assemblage complet du bras manipulateur.....	Erreur ! Signet non défini.
3.12. Carte de commande pour le bras manipulateur .....	Erreur ! Signet non défini.
3.13. Conclusion .....	Erreur ! Signet non défini.
<b>Conclusion générale.....</b>	<b>Erreur ! Signet non défini.</b>

# Introduction générale

---

Regroupant l'électronique, l'électrotechnique, la mécanique, l'automatique et l'informatique, la robotique est devenue depuis les années soixante-dix, une discipline à part entière. Avec les grands progrès qu'elle a connus, la robotique s'impose de plus en plus dans l'industrie moderne. Actuellement, les chercheurs parviennent à concevoir des robots légers et rapides...

Les robots manipulateurs sont les plus utilisés dans les systèmes industriels, car ils sont flexibles permettent d'augmenter la productivité, réduire les coûts de production et améliorer la qualité de la production. Nous retrouvons également ce type de robots dans les hôpitaux où ils assistent l'être humain dans des procédures chirurgicales difficiles ainsi que dans des milieux qui lui sont hostiles, tel que dans des pièces à températures très élevées ou radioactive. Ils ont également fait leurs preuves dans des domaines tels que l'agriculture, la construction, l'exploitation des fonds marins ou de l'espace.

Commander un robot manipulateur consiste à lui permettre d'atteindre un état désiré ou de réaliser une tâche bien spécifique avec une grande précision dans un environnement réel.

Le but de notre travail consiste en la conception et la réalisation d'un robot de type bras manipulateur à trois degrés de liberté en utilisant des servomoteurs permettant ainsi au bras manipulateur de se déplacer avec une grande précision.

Dans le premier chapitre nous avons présenté de manière générale la robotique ainsi que les actionneurs utilisés dans notre projet (les servomoteurs) et leur principe de fonctionnement.

Le deuxième chapitre est consacré à la modélisation des robots manipulateurs. Nous avons donné dans ce chapitre les notions de base de la modélisation géométrique qui ont été retenues pour modéliser le bras manipulateur.

Le troisième chapitre est consacré à présenter la démarche que nous avons suivie

## Introduction générale

---

pour la conception et la réalisation du bras manipulateurs, dont nous avons parler sur la carte Arduino UNO que nous avons utilisé pour le commander ainsi les capteurs utilisés pour le rendre plus autonome ainsi une étude mécanique pour déterminer le choix des servomoteurs a utilisés pour qu'il soit capable d'effectuer sa tache librement.

### Partie 1 : Généralités sur la robotique

#### 1.1. Introduction

La commande d'un robot a pour but de contrôler le mouvement des actionneurs selon la trajectoire programmée. Les actionneurs peuvent être de nature électrique, hydraulique ou pneumatique. Les actionneurs électriques les plus utilisés en robotique sont : les moteurs pas à pas, les moteurs synchrones, les moteurs asynchrones et les servomoteurs. Dans notre projet nous allons utiliser les servomoteurs et cela pour leur facilité au niveau de la commande.

#### 1.2. Définition et historique

##### Définition d'un robot

Un robot est un dispositif mécanique poly-articulé mus par des actionneurs et commandé par un contrôleur (calculateur) accomplissant automatiquement une grande variété des tâches qui sont généralement considérées comme dangereuses, pénibles, répétitives et impossibles pour les humains ou dans un but d'une plus grande efficacité. Le terme robot vient du Tchèque robota, il veut dire le travail forcé.

##### Boucle de décision

Un robot est capable d'extraire de l'information à partir de son environnement et d'utiliser ses connaissances pour décider comment agir. Un robot est équipé de capteurs et d'effecteurs.

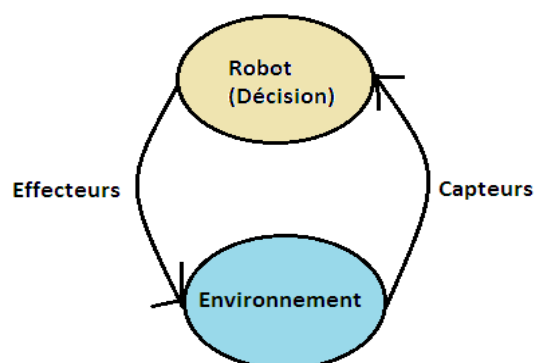


Figure 1.1 : Boucle de décision

### Historique :

En 1947 : L'apparition de premier manipulateur électrique télé-opéré.

En 1945 : Invention du premier robot qui peut être programmé.

En 1961 : Montage d'un robot sur une chaîne de fabrication de General Motors.

En 1961 : Premier robot avec contrôle en effort.

En 1963 : Le premier contrôle du robot avec vidéo.

### 1.3. Les différentes catégories des robots

- Robots mobiles figure (1.2).
- Robots sous marins figure (1.3).
- Robots volants figure (1.4).
- Robots humanoïdes figure (1.5).
- Robots manipulateurs figure (1.6)

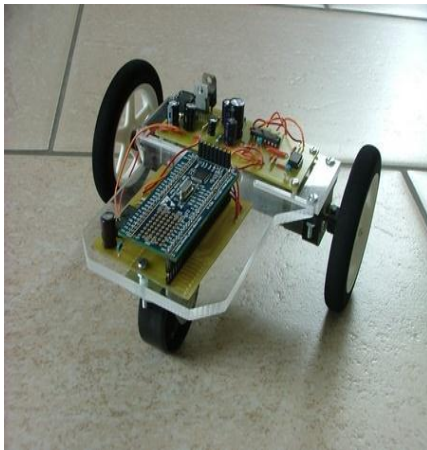


Figure 1.2 : Robot mobile

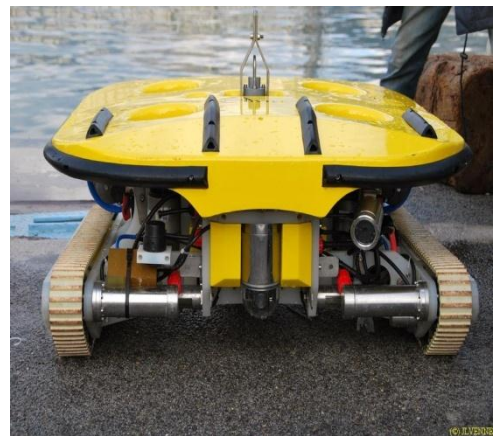


Figure 1.3 : Robot sous marin



Figure 1.4 : Robot volant



Figure 1.5 : Robot humanoïde



Figure 1.6 : Robot manipulateur

### 1.4. Vocabulaire utilisé dans le domaine de la robotique

1. Actionneur = moteur, vérin...
2. Axe = articulation.
3. Corps = segment.
4. Organe terminal.
5. Effecteur = outil.
6. Base

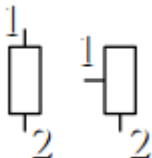
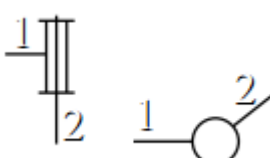


Figure 1.7 : Constituants du bras manipulateur

### 1.5. Caractéristiques des robots

#### 1.5.1. Description de sa géométrie

Robot = système mécanique poly-articulé [1]:

- Articulation prismatique :  (symbole)
- Articulation rotoïde :  (symbole)

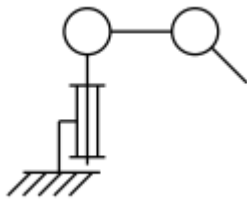
### Caractéristiques géométriques

- Nombre d'axes (mus par un actionneur).
- Architecture (série ou parallèle).
- Chaînage des articulations.
- Nombre de degrés de liberté.

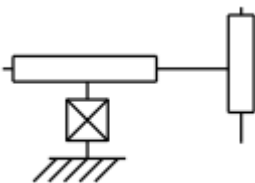
P : Prismatique(Translation)

R : Rotation

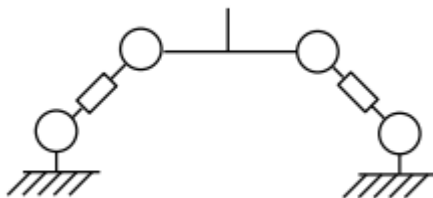
DL : Degrés de liberté



3 axes, série, RRR, 3DL.



3 axes, série, PPP, 3DL.



4 axes, parallèle, RP+RP, 3DL.

### 1.5.2. Volume de travail

Le volume de travail accessible par l'outil du robot est le volume qui peut balayer l'outil. Ce volume dépend de la géométrie du robot, de la longueur des segments et du débattement des articulations [1].

### 1.5.3. Précision / Répétabilité

On dit que le positionnement absolu est imprécis lorsqu'il dépasse 1mm, alors on a l'une des erreurs suivantes:

- Erreurs de modèle géométrique.
- Erreurs de quantification de la mesure de position.
- Flexibilité.

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 ne doit pas dépasser 0.1 mm [1].

### 1.5.4. Performances dynamiques

- Vitesse maximale :

C'est la vitesse de translation ou de rotation de chaque axe du robot. Souvent les constructeurs donnent une vitesse maximale de l'outil ou de l'organe terminal [1].

- Accélération maximale :

Elle dépend de l'inertie, donc de la position du robot. Pour chaque axe, cette accélération est donnée dans la configuration la plus défavorable (inertie maximale, charge maximale) [1].

### 1.5.5. Charge utile

C'est la charge qui garantit une durée de vie la plus longue possible. Donc, c'est la charge maximale que peut porter le robot sans perturber ni la répétabilité ni les performances dynamiques du robot. Cette charge utile est nettement inférieure à la charge maximale que peut porter le robot et est directement dépendante des actionneurs [1].

## 1.6. Les différents types de robots

### 1.6.1. Les robots SCARA (Selective Compliance Articulated)

Ce type de robot à la fois précis et rapide, est utilisé dans l'assemblage. C'est un robot à 3 axes série (RRT), 3 degrés de liberté, son espace de travail est cylindrique [1].

Exemple :



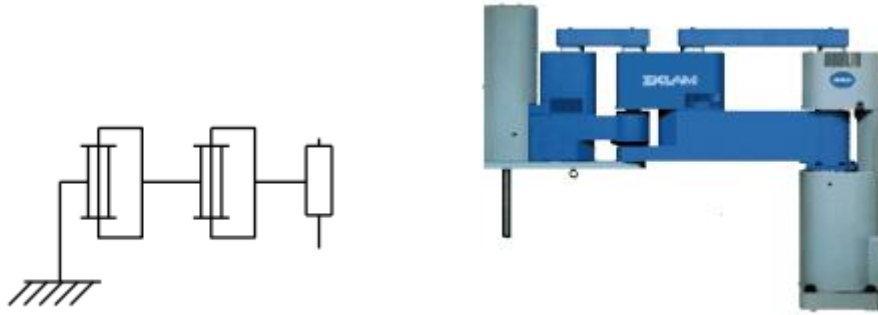


Figure 1.8 : Bras manipulateur (robot SCARA(Sankyo))

### 1.6.2. Les robots cylindriques

Ce type de robots est caractérisé par une rapidité et un espace de travail cylindrique.

C'est un robot à 3 axes, série (RTT) et 3 degrés de liberté, son espace de travail est cylindrique [1].

Exemple :

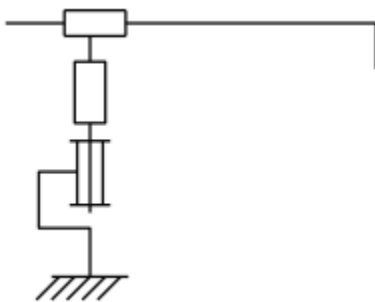


Figure 1.9 : Bras manipulateur (robot cylindrique(Seiko))

### 1.6.3. Les robots sphériques

Ce sont des robots à 3 axes, série RRT, et 3 degrés de libertés, leurs espace de travail est sphérique. Ils sont caractérisés par leur grande charge utile [1].

Exemple :

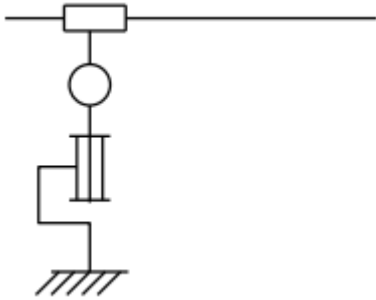


Figure 1.10 : Bras manipulateur (robot sphérique(Fanuc))

### 1.6.4. Les robots Cartésiens

Ce sont des robots à 3 axes perpendiculaires 2 à 2, série, TTT, et 3 degrés de liberté et caractérisés par leur Très bonne précision, mais ils sont lents [1].

Exemple :

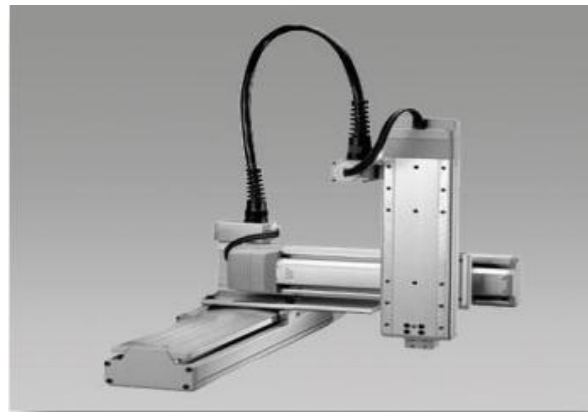
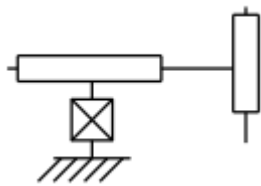


Figure 1.11 : Bras manipulateur (robots cartésien(Toshiba))

### 1.6.5. Les robots parallèles

Ce sont des robots à Plusieurs chaînes cinématiques en parallèles et un espace de travail réduit. Ils sont caractérisés par la rapidité et la précision (grande rigidité de la structure) [1].

Exemple :



Figure 1.12 : Bras manipulateur (robots parallèle(Comau))

### 1.6.6. Les robots anthropomorphes

Ce sont des robots à 6 axes, série, 6 rotations, et 6 degrés de liberté, ils Reproduisent la structure d'un bras humain [1].

Exemple :

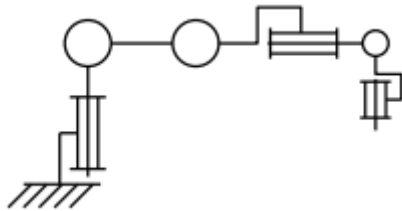


Figure 1.13 : Bras manipulateur (robot anthropomorphe (Kawasaki))

### Partie 2 : Généralités sur les servomoteurs

#### 1.7. Présentation

Les servomoteurs permettent de déplacer un bras, sur lequel est fixé un objet, jusqu'à une certaine position (ou angle de rotation), puis à maintenir solidement cette position. Le terme même de servomoteur signifie qu'il s'agit d'un moteur asservi, obéissant à une commande externe [5].

Un servomoteur est un ensemble complet de mécanique et d'électronique, qui contient [5]:

- Un moteur à courant continu (CC) souvent de petite taille
- Une carte électronique d'asservissement
- Un réducteur de vitesse
- Un potentiomètre, pour contrôler la position de l'axe du moteur
- Un axe dépassant hors du boîtier avec différents bras ou roues de fixation

Un servomoteur est principalement caractérisé par :

- Sa vitesse de rotation pour aller à la position désirée en seconde/degré
- Son couple (torque en anglais) qui signifie la force que le servomoteur peut dégager pour garder la position souhaitée en kg/cm.

L'unité kg/cm signifie le poids que peut supporter le servomoteur en considérant qu'on y a fixé une tige de 1 cm.



Figure 1.14 : Un servomoteur  
et ses accessoires

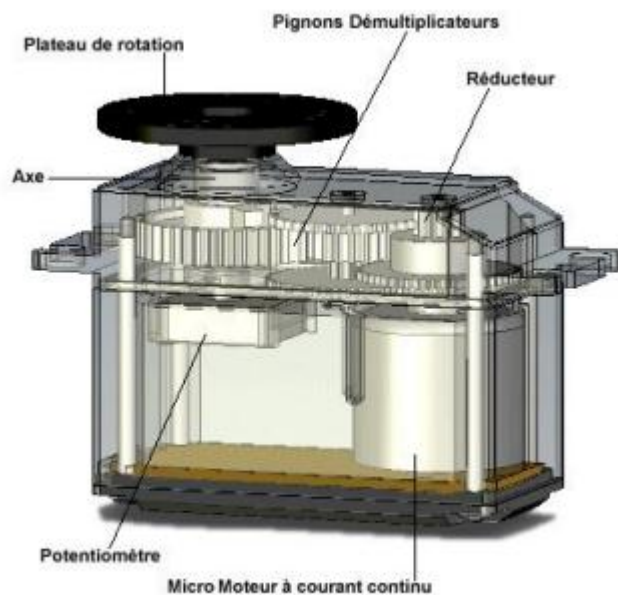


Figure 1.15 : Vue de l'intérieur et  
l'extérieur d'un servomoteur

### CONNECTIQUE

Un servomoteur a trois fils de connexion pour fonctionner. Deux fils servent à son alimentation, le dernier étant celui qui reçoit le signal de commande :

**Rouge** : pour l'alimentation positive (4.5V à 6V en général).

**Noir** ou **marron** : pour la masse (0V).

**Orange, jaune, blanc, ...** : entrée du signal de commande.

### 1.8. Utilisation

Les servomoteurs se trouvent fréquemment dans les magasins de modélisme ou dans les catalogues d'électronique. Sur des maquettes comme en robotique, ils sont utilisés pour positionner, faire pivoter, rentrer/sortir des éléments d'aiguillage, onglets, signalisations, train d'atterrissage d'un avion, ils permettent de déplacer un bras jusqu'à une certaine position (ou angle) [5].

### 1.9. Variété

Il existe une vaste gamme de servomoteurs : des gros modèles industriels de plusieurs kilogrammes à courant alternatif, avec des boîtiers et carters d'engrenages en métal, jusqu'aux minuscules servomoteurs en plastique de quelques grammes à courant continu que l'on retrouve en modélisme dans les avions, les bateaux, les voitures et les hélicoptères modèles réduits.

En modélisme Les plus courants et les moins chers ont une course limitée à 90° ou 120°. Un modèle de base par exemple a pour dimensions 40 x 20 x 36 mm, il peut lever 3kg/cm, Il existe aussi des servomoteurs qui font 180° ou même à rotation continue.

Exemple :



Figure 1.16 : Servomoteur Futaba S3003

**Caractéristiques :**

- Alimentation : 4.8 à 6 Volt
- Vitesse : 0.23 sec/60°
- Couple : 3.2 kg/cm
- Poids : 37.2 g



Figure1.17 : Servomoteur HS-5745MG

**Caractéristiques :**

- Alimentation : 4.8 à 6 Volt
- Vitesse : 0.15 sec/60°
- Couple : 18 kg/cm
- Poids : 161 g

### 1.10. Fonctionnement

Les servomoteurs sont commandés par l'intermédiaire d'un câble électrique à trois fils qui permet d'alimenter le moteur et de lui transmettre des consignes de position sous forme d'un signal codé en largeur d'impulsion plus communément appelé PWM ( Pulse width modulation, Modulation en largeur d'impulsion). Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur. Le signal est répété périodiquement, en général toutes les 20 millisecondes, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre. Lorsque le moteur tourne, l'axe du servomoteur change de position, ce qui modifie la résistance du potentiomètre. Le rôle de l'électronique est de commander le moteur pour que la position de l'axe de sortie soit conforme à la consigne reçue, c'est un asservissement [5].

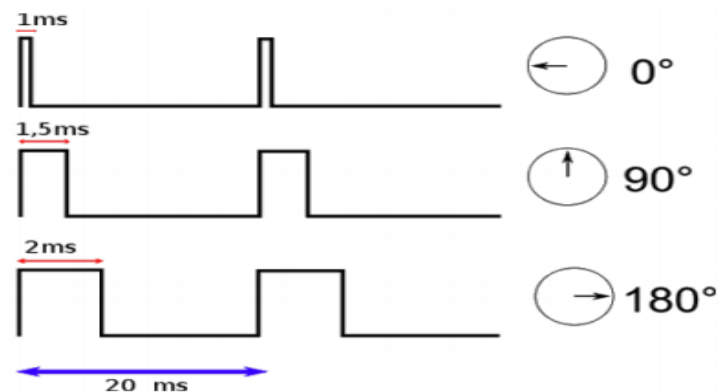


Figure 1.18 : Relation PWM / Position de l'arbre

### 1.11. Rôle du potentiomètre

Le potentiomètre sert à l'asservissement de position. L'axe du potentiomètre est solidaire avec l'axe de sortie. Donc le potentiomètre tourne en même temps que l'axe de sortie. La résistance au 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. C'est le rôle de l'électronique de faire la relation entre durée de l'impulsion et la résistance aux bornes du potentiomètre [5].

### 1.12. Les avantages et les inconvénients

- **Les avantages sont les suivants**

Le fil signal à faible courant peut être raccordé directement à une sortie du microcontrôleur. Pas besoin de circuit d'interface.

On peut commander l'arrêt, la marche, le sens de rotation et la vitesse du servomoteur à l'aide d'un seul fil. Economie d'E/S.

Le servomoteur offre un couple important sous un volume réduit.

- **Les inconvénients**

Il faut modifier le servomoteur pour une rotation complète.

Le prix est légèrement plus élevé qu'un bloc motoréducteur à 2 moteurs CC.

### 1.13. Conclusion

Dans ce chapitre nous avons cité les différents types de servomoteurs ainsi que leur fonctionnement. Cela nous a permis de mieux comprendre leurs principes de fonctionnement et de bien aborder l'objectif principal de notre travail qui est la conception d'une commande pour un bras manipulateur en utilisant comme articulateurs : les servomoteurs.



### 2.1. Introduction

Pour commander ou simuler le comportement d'un système mécanique articulé (robot), on doit disposer d'un modèle. Plusieurs niveaux de modélisation sont possibles selon les objectifs, les contraintes de la tâche et les performances recherchées.

Les modèles mathématiques nécessaires sont :

- les modèles géométriques directs et inverses qui expriment la situation de l'organe terminal en fonction des variables articulaire et inversement.
- les modèles cinématiques direct et inverse qui expriment les vitesses de l'organe terminal en fonction des variables articulaires et inversement.
- les modèles dynamiques définissant les équations du mouvement du robot qui permettent d'établir les relations entre les couples ou forces exercées par les actionneurs et les positions, vitesses et accélérations des articulations.

### 2.2. Description géométrique d'un bras manipulateur

Un robot manipulateur à structure ouverte simple est composé de  **$n+1$**  corps rigides notés :  $C_0 \dots C_j \dots C_n$  et de  **$n$**  articulations ou liaisons. Le corps  $C_0$  constitue la base du robot tandis que le corps  $C_n$  porte l'organe terminal. L'articulation  $J$  relie le corps  $C_j$  au corps  $C_{j-1}$ , le mouvement de chaque articulation produit le mouvement relatif du corps.

- ✓ Articulation prismatique notée (P) : lorsque le mouvement effectué est une translation.
- ✓ Articulation rotoïde notée (R) : lorsque le mouvement effectué est une rotation.

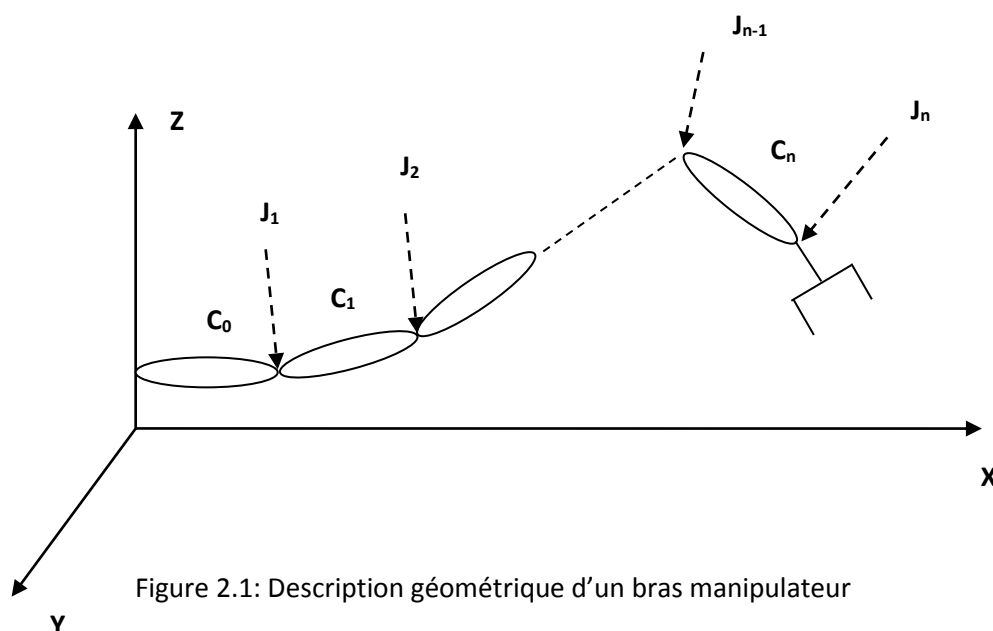


Figure 2.1: Description géométrique d'un bras manipulateur

### 2.3. Espace des coordonnées

On distingue deux types de coordonnées :

#### a- Coordonnées opérationnelles

elles permettent de caractériser et de définir la position de l'organe terminal, pour cela on définit un espace appelé espace opérationnel qui a pour référence un repère ( $R_{op}$ ) orthonormé ( $O_0, X_0, Y_0, Z_0$ ) lié généralement à la base du robot manipulateur, de même il est nécessaire d'introduire un autre repère ( $R_n$ ) orthonormé ( $O_n, X_n, Y_n, Z_n$ ) lié à l'organe terminal afin de mieux définir la situation (position et orientation de l'organe terminal). Les coordonnées associées sont donc appelées coordonnées opérationnelles.

#### b- Coordonnées généralisées

on appelle coordonnées généralisées la quantité définissant le mouvement relatif du corps (segment)  $C_i$  par rapport au corps  $C_{i-1}$ , on la note  $q_i$ . Elle peut être un angle autour d'un axe (liaison de rotation), distance (longueur) le long d'un axe (liaison de translation).

### 2.4. Notations

Afin de faciliter la compréhension des différentes écritures mathématiques utilisées dans ce chapitre, nous avons défini les notations suivantes :

$R_i$  : Repère  $i$

$P$  : Point

${}^iP$  : Coordonnées du point dans le repère  $n^\circ i$

$\overrightarrow{OP}$  : Vecteur

$I_{op}$  : Coordonnées de  $\overrightarrow{OP}$  dans le repère  $i$

$R_{01}$  : Rotation du repère  $R_0$  vers le repère  $R_1$

$M_{01}$  : Matrice de passage homogène du repère  $R_0$  vers  $R_1$

${}^{i-1}M_i$  : Matrice de passage homogène du repère  $R_i$  vers  $R_{i-1}$

## 2.5. Orientation d'un solide

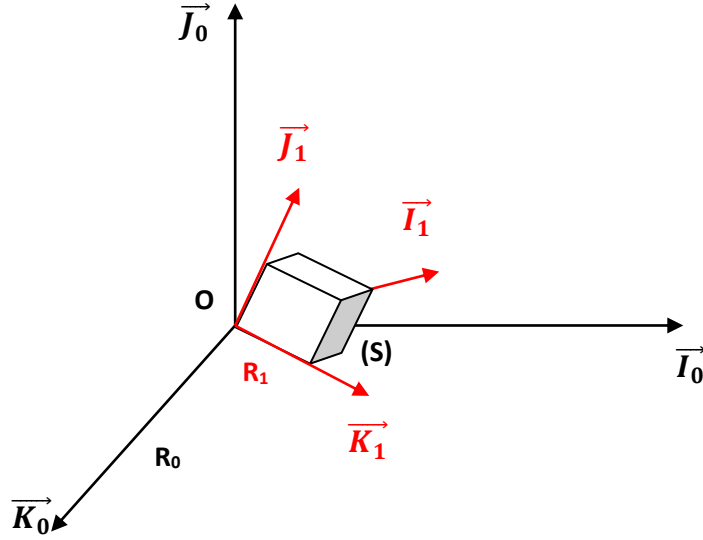


Figure 2.2: Orientation d'un solide

$$\vec{i}_1 = (\vec{i}_1 \cdot \vec{i}_0) \vec{i}_0 + (\vec{i}_1 \cdot \vec{j}_0) \vec{j}_0 + (\vec{i}_1 \cdot \vec{k}_0) \vec{k}_0$$

$$\vec{j}_1 = (\vec{j}_1 \cdot \vec{i}_0) \vec{i}_0 + (\vec{j}_1 \cdot \vec{j}_0) \vec{j}_0 + (\vec{j}_1 \cdot \vec{k}_0) \vec{k}_0$$

$$\vec{k}_1 = (\vec{k}_1 \cdot \vec{i}_0) \vec{i}_0 + (\vec{k}_1 \cdot \vec{j}_0) \vec{j}_0 + (\vec{k}_1 \cdot \vec{k}_0) \vec{k}_0$$

D'où :

$$\begin{pmatrix} \vec{i}_1 \\ \vec{j}_1 \\ \vec{k}_1 \end{pmatrix} = \begin{bmatrix} \vec{i}_1 \cdot \vec{i}_0 & \vec{i}_1 \cdot \vec{j}_0 & \vec{i}_1 \cdot \vec{k}_0 \\ \vec{j}_1 \cdot \vec{i}_0 & \vec{j}_1 \cdot \vec{j}_0 & \vec{j}_1 \cdot \vec{k}_0 \\ \vec{k}_1 \cdot \vec{i}_0 & \vec{k}_1 \cdot \vec{j}_0 & \vec{k}_1 \cdot \vec{k}_0 \end{bmatrix} \begin{pmatrix} \vec{i}_0 \\ \vec{j}_0 \\ \vec{k}_0 \end{pmatrix}$$

$$\begin{pmatrix} \vec{i}_1 \\ \vec{j}_1 \\ \vec{k}_1 \end{pmatrix} = R_{10} \begin{pmatrix} \vec{i}_0 \\ \vec{j}_0 \\ \vec{k}_0 \end{pmatrix}$$

Avec  $R_{10}$  la matrice de rotation de  $R_1$  vers  $R_0$

De la même manière :

$$\begin{pmatrix} \vec{l}_0 \\ \vec{j}_0 \\ \vec{k}_0 \end{pmatrix} = \begin{bmatrix} \vec{l}_1 \cdot \vec{l}_0 & \vec{j}_1 \cdot \vec{l}_0 & \vec{k}_1 \cdot \vec{l}_0 \\ \vec{l}_1 \cdot \vec{j}_0 & \vec{j}_1 \cdot \vec{j}_0 & \vec{k}_1 \cdot \vec{j}_0 \\ \vec{l}_1 \cdot \vec{k}_0 & \vec{j}_1 \cdot \vec{k}_0 & \vec{k}_1 \cdot \vec{k}_0 \end{bmatrix} \begin{pmatrix} \vec{l}_1 \\ \vec{j}_1 \\ \vec{k}_1 \end{pmatrix}$$

Avec  $\mathbf{R}_{01}$  : matrice de rotation de  $\mathbf{R}_0$  vers  $\mathbf{R}_1$

Comme  $\mathbf{R}_{01}$  et  $\mathbf{R}_{10}$  sont des transformations inverses, il vient:  $\mathbf{R}_{01} = \mathbf{R}_{10}^{-1}$

$\mathbf{R}_{01} = \mathbf{R}_{10}^T$  (matrice transposée).

### 2.5.1 Orientation d'un point appartenant au solide S

Soit  $\mathbf{P}$  un point du solide ( $\mathbf{s}$ ) de coordonnées  $(x_{p0} \ y_{p0} \ z_{p0})$  et  $(x_{p1} \ y_{p1} \ z_{p1})$  dans les repères  $\mathbf{R}_0$  et  $\mathbf{R}_1$  respectivement [3].

On déduit que :

D'où :

$$\begin{pmatrix} \vec{X}_{p0} \\ \vec{Y}_{p0} \\ \vec{Z}_{p0} \end{pmatrix} = R_{01} \begin{pmatrix} \vec{X}_{p1} \\ \vec{Y}_{p1} \\ \vec{Z}_{p1} \end{pmatrix} \Leftrightarrow {}^0\mathbf{P} = \mathbf{R}_{01} {}^1\mathbf{P}$$

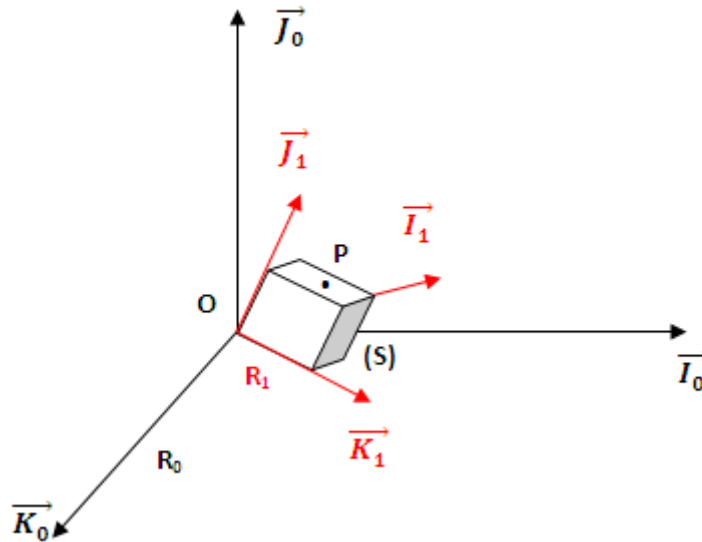


Figure 2.3: Orientation d'un point appartenant au solide (S)

Remarque :

Par ailleurs si on considère plusieurs rotations :

- $R_{01}$  rotation de  $R_1$  vers  $R_0$
- $R_{12}$  rotation de  $R_2$  vers  $R_1$

On aura donc comme précédemment :

$${}^1P = R_{12} {}^2P$$

$${}^0P = R_{01} {}^1P$$

D'où :

$${}^0P = R_{01} R_{12} {}^2P = R_{02} {}^2P$$

### 2.6. Transformation des repères

Pour transformer (translation ou rotation) un repère  $R_i$  et l'amener dans un autre repère  $R_j$  on fait appel à une matrice dite matrice de transformation homogène ou bien matrice de passage homogène notée  ${}^i T_j$ .

Soit le repère  $R_j$  défini par trois vecteur unitaires  $a_j, b_j, c_j$  et son origine  $P_j$ .

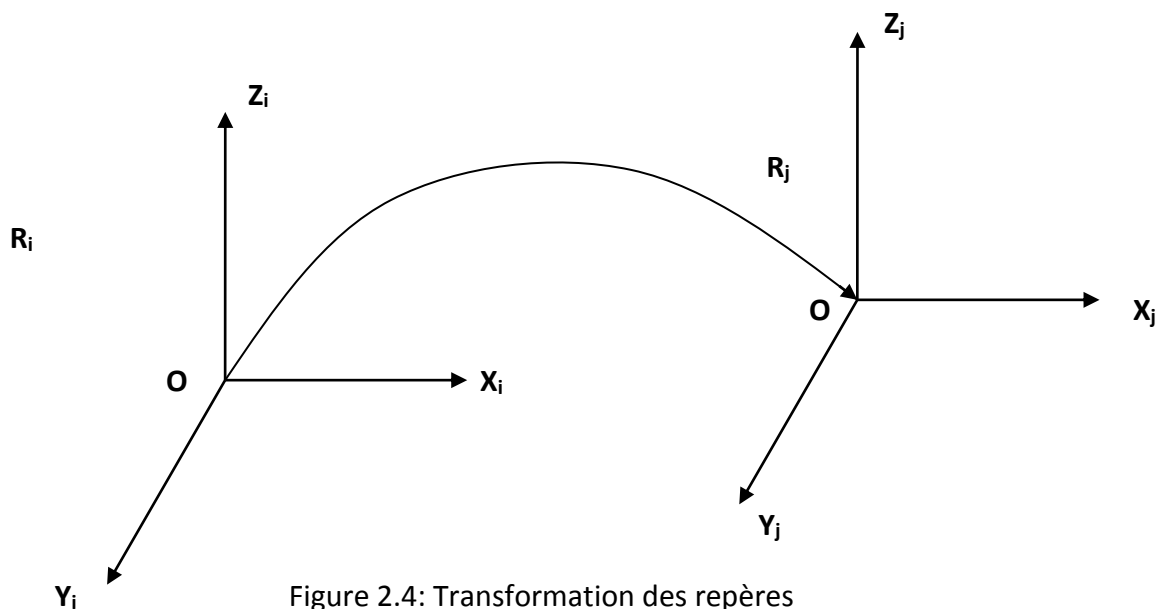


Figure 2.4: Transformation des repères

Les coordonnées des vecteurs unitaires et de l'origine du repère  $R_j$  dans le repère de base  $R_i$  sont respectivement données comme suit :

$${}^i\mathbf{a}_j = \begin{bmatrix} \mathbf{a}_x \\ \mathbf{a}_y \\ \mathbf{a}_z \end{bmatrix}, \quad {}^i\mathbf{b}_j = \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \\ \mathbf{b}_z \end{bmatrix}, \quad {}^i\mathbf{c}_j = \begin{bmatrix} \mathbf{c}_x \\ \mathbf{c}_y \\ \mathbf{c}_z \end{bmatrix}, \quad {}^i\mathbf{p}_j = \begin{bmatrix} \mathbf{p}_x \\ \mathbf{p}_y \\ \mathbf{p}_z \end{bmatrix}$$

la matrice de transformation homogène est donnée comme suit :

$${}^i\mathbf{T}_j = [{}^i\mathbf{a}_j, {}^i\mathbf{b}_j, {}^i\mathbf{c}_j, {}^i\mathbf{p}_j] = \begin{bmatrix} \mathbf{a}_x & \mathbf{b}_x & \mathbf{c}_x & \mathbf{p}_x \\ \mathbf{a}_y & \mathbf{b}_y & \mathbf{c}_y & \mathbf{p}_y \\ \mathbf{a}_z & \mathbf{b}_z & \mathbf{c}_z & \mathbf{p}_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## 2.7. Matrices homogènes

C'est un outil mathématique permettant de calculer un changement de base en une seule opération matricielle. Pour cela on rajoute une 4<sup>ème</sup> coordonnée.

$$\mathbf{M}_{01} = \begin{bmatrix} & \mathbf{R}_{01} & \begin{matrix} \mathbf{T}_x \\ \mathbf{T}_y \\ \mathbf{T}_z \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^1\mathbf{P} = \begin{bmatrix} \mathbf{P}_x \\ \mathbf{P}_y \\ \mathbf{P}_z \\ 1 \end{bmatrix} \quad \leftarrow \begin{matrix} \text{Ajout d'une} \\ \text{coordonnée} \\ \text{Supplémentaire} \end{matrix}$$

En utilisant les matrices homogènes le calcul des coordonnées d'un point  $\mathbf{P}$  par rapport à un autre repère sera donné par la relation :

$${}^0\mathbf{P} = \mathbf{M}_{01} {}^1\mathbf{P}$$

**Remarque :**

${}^0\mathbf{P} = \mathbf{M}_{01} {}^1\mathbf{P}$  : est une relation de changement de coordonnées pour un point.

$\mathbf{M}_{02} = \mathbf{M}_{01} \mathbf{M}_{12}$  : est une relation de composition entre plusieurs matrices homogènes.

$\mathbf{M}_{01} \mathbf{M}_{12} \neq \mathbf{M}_{12} \mathbf{M}_{01}$  : la matrice homogène n'est pas commutative.

L'inverse d'une matrice de passage  $\mathbf{M}^{-1}$  est donnée par :

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{T} \\ 0 & 1 \end{bmatrix} \quad \mathbf{M}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{T} \\ 0 & 1 \end{bmatrix}$$

## 2.8. Application de la matrice de transformation homogène

### 2.8.1. Matrice de transformation de translation

Faisons subir au repère  $R_i$  une translation donnée par  $\mathbf{d} = [a \ b \ c]^t$

Avec  $\mathbf{d}$  coordonnées de l'origine du nouveau repère dans le repère  $R_i$  [3].

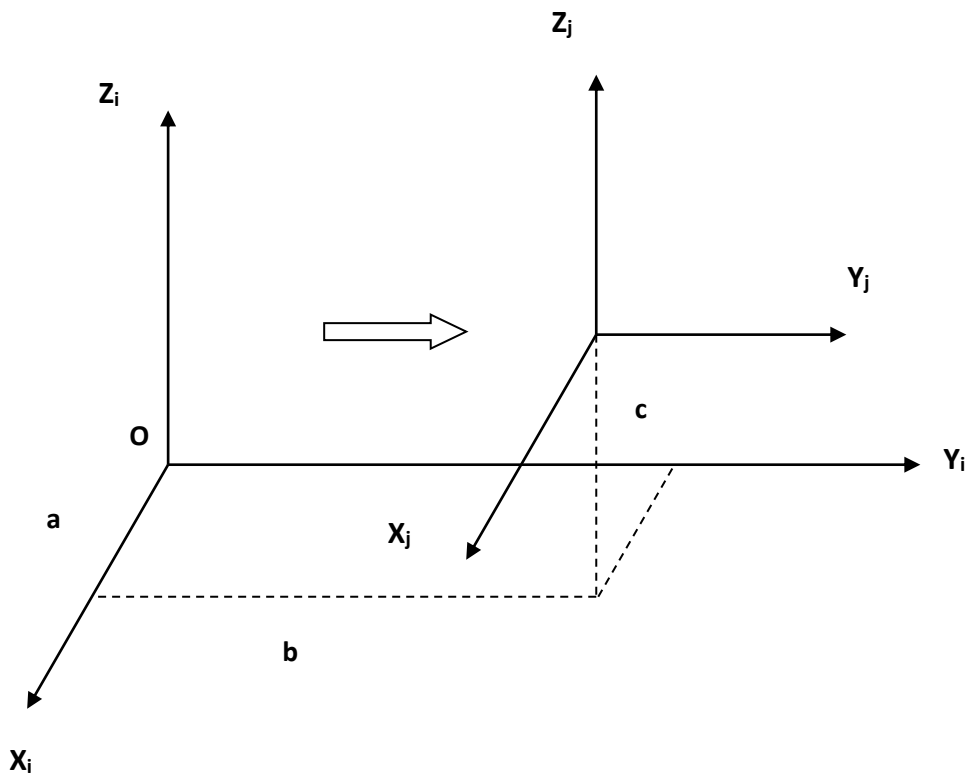


Figure 2.5: Translation d'un repère

La translation notée **Trans (a,b,c)** est donnée par :

$$\text{Trans (a,b,c)} = {}^i T_j = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 2.8.2. Matrice de transformation de rotation

#### a- Rotation autour de l'axe X

Faisons subir au repère  $R_i$  une rotation  $\Theta$  autour de l'axe  $X$  [3].

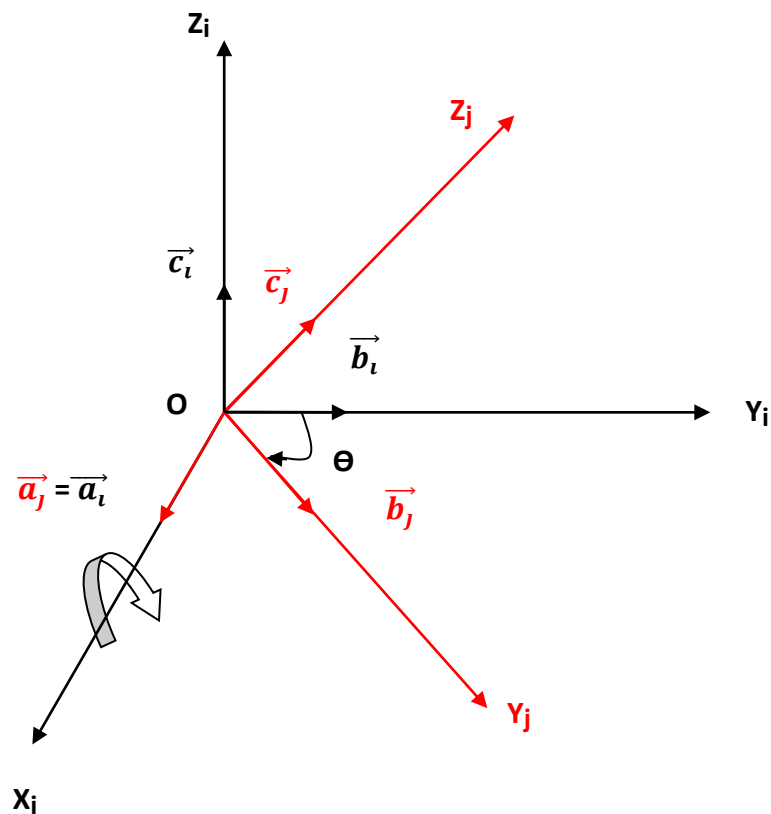


Figure 2.6: rotation autour de  $X_i$

On trouve par projection :

$$\vec{a}_j = \vec{a}_i$$

$$\vec{b}_j = \cos \Theta \cdot \vec{b}_i - \sin \Theta \cdot \vec{c}_i$$

$$\vec{c}_j = \sin \Theta \cdot \vec{b}_i + \cos \Theta \cdot \vec{c}_i$$

Pour simplifier l'écriture matricielle, posons :

$$\cos \Theta = C(\Theta)$$



**Sin  $\Theta = S(\Theta)$**

La matrice de transformation de rotation autour de l'axe **X** notée **Rot(x ,  $\Theta$ )** est donnée par :

$$\text{Rot}(\mathbf{x} , \Theta) = {}^i T_j = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C(\Theta) & -S(\Theta) & 0 \\ 0 & S(\Theta) & C(\Theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

**b- Rotation autour de l'axe Y**

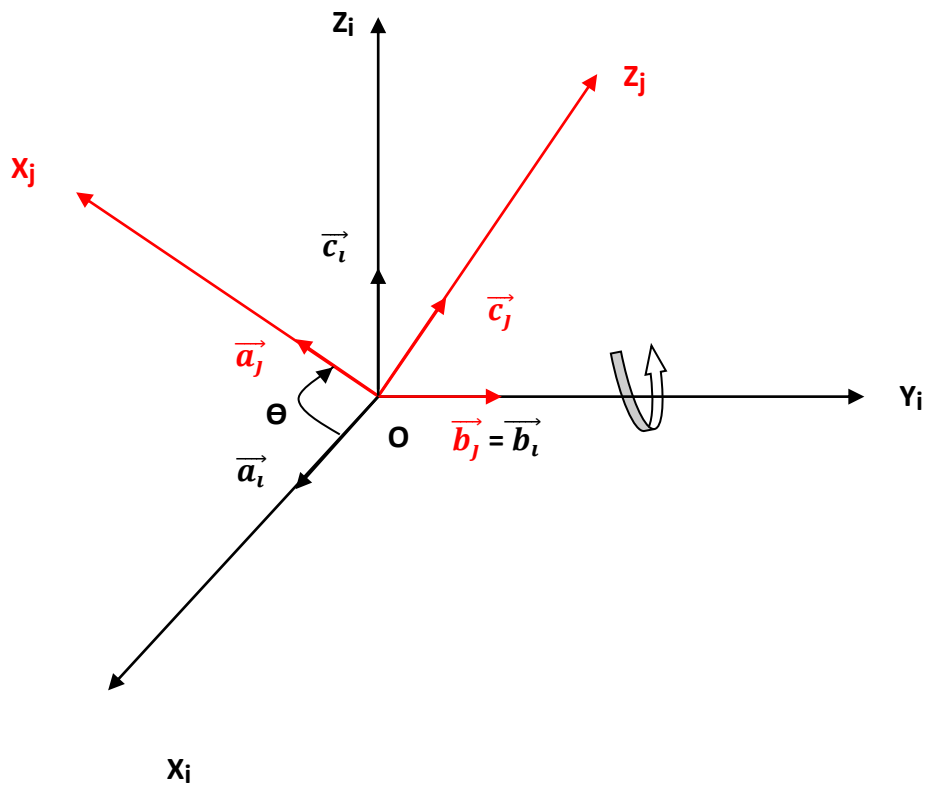


Figure 2.7: rotation autour de  $Y_i$

On trouve par projection [3] :

$$\vec{a}_j = \cos \Theta \cdot \vec{a}_i + \sin \Theta \cdot \vec{c}_i$$

$$\vec{b}_j = \vec{b}_i$$

$$\vec{c}_j = -\sin \Theta \cdot \vec{a}_i + \cos \Theta \cdot \vec{c}_i$$

La matrice de transformation de rotation autour de l'axe **Y** notée **Rot (y,  $\Theta$ )** est donnée par :

$$\text{Rot (y , } \Theta) = {}^i T_j = \begin{bmatrix} C(\Theta) & 0 & S(\Theta) & 0 \\ 0 & 1 & 0 & 0 \\ -S(\Theta) & 0 & C(\Theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### C- Rotation autour de l'axe Z

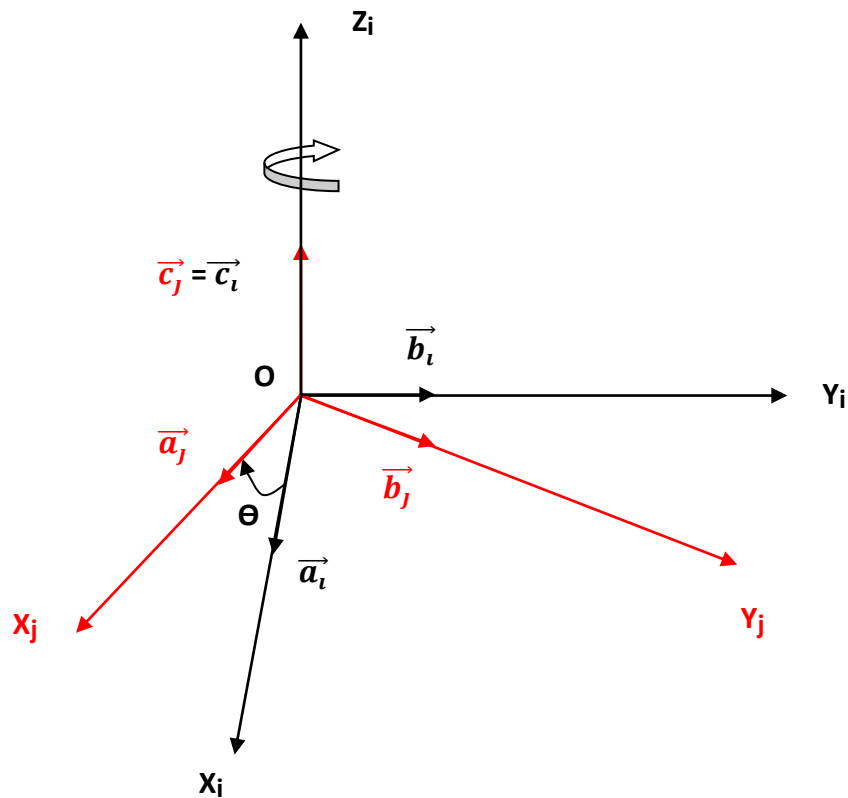


Figure 2.8: rotation autour de  $Z_i$

On trouve par projection [3] :

$$\vec{a}_j = \cos \Theta \cdot \vec{a}_i - \sin \Theta \cdot \vec{b}_i$$

$$\vec{b}_j = \sin \Theta \cdot \vec{a}_i + \cos \Theta \cdot \vec{b}_i$$

$$\vec{c}_j = \vec{c}_i$$

La matrice de transformation de rotation autour de l'axe **Z** notée **Rot (Z ,  $\Theta$ )** est donnée par :

$$\text{Rot (Z ,  $\Theta$ )} = {}^i T_j = \begin{bmatrix} C(\Theta) & -S(\Theta) & 0 & 0 \\ S(\Theta) & C(\Theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 2.9. Modélisation géométrique

Quelle que soit la tâche que doit réaliser un robot manipulateur, il est important de maîtriser la position et le déplacement de son outil de travail (organe terminal). La modélisation géométrique qui est une configuration définie par un ensemble de variables articulaires traduisant les déplacements relatifs d'un corps par rapport au précédent, pour faciliter les calculs du modèle géométrique, on doit suivre ces étapes essentielles dans l'ordre suivant [3] :

1-fixer des repères à chaque corps du robot.

2-calculer la matrice homogène entre chaque corps.

2-calculer la matrice homogène entre la base et l'organe terminal.

### 2.10. Convention de Denavit-Hertenberg(DH)

C'est une méthode destinée à systématiser la modélisation de n'importe quel types de robot série. Ses principaux avantages sont [3] :

- Simplification maximale du modèle géométrique.
- Etablissement d'une norme reconnue par tous.

On peut représenter l'attitude d'un repère **R<sub>i</sub>** par rapport à un repère **R<sub>i-1</sub>** à l'aide de 4 paramètres unique à condition de fixer les deux conditions de Denavit-Hertenberg(DH) suivante :

- DH1-L'axe **X<sub>i</sub>** de **R<sub>i</sub>** est perpendiculaire à l'axe **Z<sub>i-1</sub>** de **R<sub>i-1</sub>**.
- DH2-L'axe **X<sub>i</sub>** coupe l'axe **Z<sub>i-1</sub>**.

Afin de satisfaire ces deux conditions fondamentales, on doit suivre ces quatre décompositions élémentaires dans l'ordre suivant :

- Rotation autour de **Z** d'un angle  **$\Theta_i$**
- Translation le long de **Z** d'une longueur  **$d_i$**

- Translation le long de X d'une longueur  $a_i$
- Rotation autour de X d'un angle  $\alpha_i$

Les paramètres  $\theta_i$ ,  $d_i$ ,  $a_i$ ,  $\alpha_i$  sont appelés : les paramètres de Denavit-Hartenberg(DH) .

$\theta_i$  : angle entre les axes  $X_{i-1}$  et  $X_i$  correspondant à une rotation autour de  $Z_{i-1}$

$\alpha_i$  : angle entre les axes  $Z_{i-1}$  et  $Z_i$  correspondant à une rotation autour de  $X_i$

$a_i$  : distance entre  $Z_{i-1}$  et  $Z_i$  le long de  $X_i$

$d_i$  : distance entre  $X_{i-1}$  et  $X_i$  le long de  $Z_{i-1}$

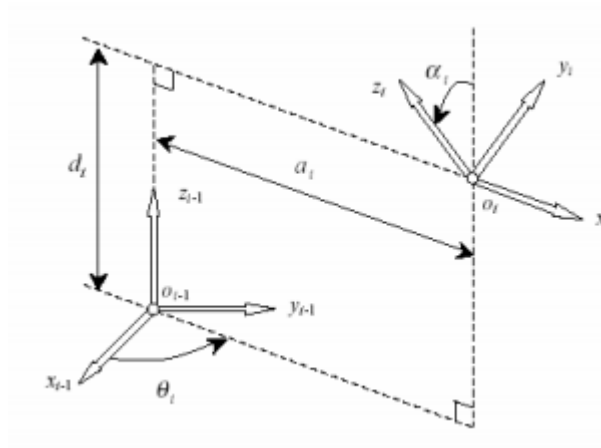


Figure 2.9: paramètres de Denavit-Hartenberg(DH)

On constate que DH1 et DH2 sont respectées.

La matrice de transformation définissant le repère  $R_i$  dans le repère  $R_{i-1}$  est donnée par :

$${}^{i-1}T_i = R_{(z_{i-1}, \theta_i)} T_{(z_{i-1}, d_i)} T_{(x_i, a_i)} R_{(x_i, \alpha_i)}$$

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 2.11. Modèle géométrique direct

Le modèle géométrique direct (MGD) est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, c'est-à-dire les coordonnées opérationnelles du robot, en fonction de ses coordonnées articulaire. Dans le cas d'une chaîne ouverte simple, il peut être représenté par la matrice de passage  ${}^0T_n$  [3].

Le modèle géométrique direct du robot peut être représenté par la relation  $\mathbf{X}=\mathbf{f}(\mathbf{q})$

Tel que :

$\mathbf{X}$  : représente les coordonnées opérationnelles.

$\mathbf{q}$  : représente les coordonnées généralisées.

### 2.12. Modèle géométrique inverse

On appelle modèle géométrique inverse la relation réciproque du modèle géométrique direct, c'est-à-dire il permet la synthèse d'un système d'équations exprimant explicitement les coordonnées généralisées en fonction des coordonnées opérationnelles [3].

Le modèle géométrique direct du robot peut être représenté par la relation  $\mathbf{q}=\mathbf{f}^{-1}(\mathbf{X})$ .

On remarque que ce modèle donne les valeurs des variables articulaires en fonction de la position et la situation de l'organe terminal.

Pour un même robot, il ya plusieurs modèles géométrique inverse alors qu'il n'y a qu'un seul modèle géométrique direct [3].

### 2.13. Modélisation du robot à commander

#### 2.13.1. Description du robot à commander

Le robot à commander est un bras manipulateur constitué principalement de trois segments Qui peuvent tourner respectivement autour des axes  $Z_1$ ,  $Z_2$  et  $Z_3$ , le troisième segment comporte l'organe terminal qui est une pince à deux doigts.

Les rotations  $\theta_1$ ,  $\theta_2$  et  $\theta_3$  ainsi que l'ouverture et la fermeture de la pince sont assurées par des actionneurs électriques (servomoteur)

### 2.13.2. Modèle géométrique direct du robot à commander

La figure ci-dessous donne une description géométrique du robot à commander.

Le placement des repères et la détermination des paramètres géométriques sont effectués selon la méthode décrite précédemment.

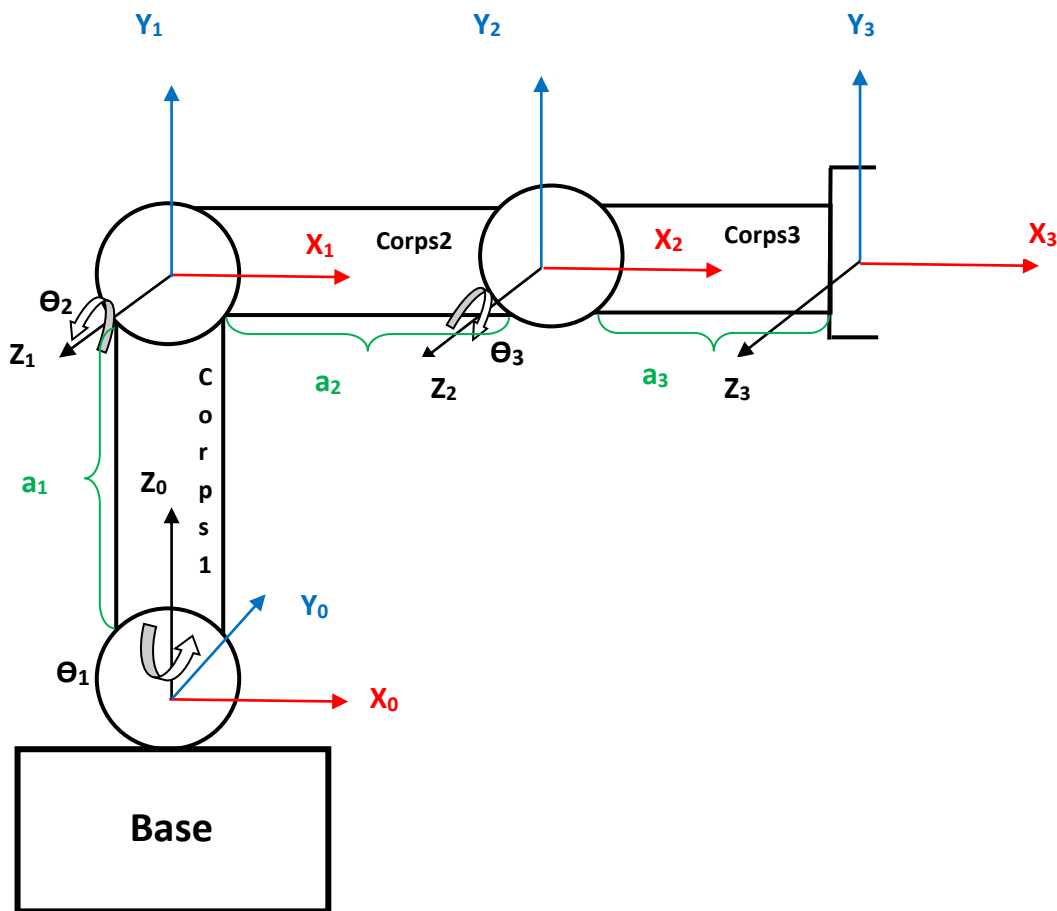


Figure 2.10: Modélisation du robot à commander

Les paramètres géométriques de Denavit-Hartenberg(DH) du robot à commander sont donnés dans le tableau suivant :

corps	$\Theta_i$	$d_i$	$a_i$	$\alpha_i$
<b>1</b>	<b><math>\Theta_1</math></b>	<b><math>a_1</math></b>	<b>0</b>	<b>90</b>
<b>2</b>	<b><math>\Theta_2</math></b>	<b>0</b>	<b><math>a_2</math></b>	<b>0</b>
<b>3</b>	<b><math>\Theta_3</math></b>	<b>0</b>	<b><math>a_3</math></b>	<b>0</b>

Tableau 2.1 : tableau des paramètres de Denavit-Hertenberg(DH)

Les matrices de transformations homogènes sont données comme suit :

Le calcul des matrices a été fait sous Matlab

```
syms t1 t2 t3 a1 a2 a3

M01=[cos(t1) 0 sin(t1) 0;sin(t1) 0 -cos(t1) 0;0 1 0 a1;0 0 0 1]

M12=[cos(t2) -sin(t2) 0 a2*cos(t2);sin(t2) cos(t2) 0 a2*sin(t2);0 0 1 0 ;0 0 0 1]

M23=[cos(t3) -sin(t3) 0 a3*cos(t3);sin(t3) cos(t3) 0 a3*sin(t3);0 0 1 0 ;0 0 0 1]

M03=simple(M01*M12*M23)
```

$t1=\theta_1$  ,  $t2=\theta_2$  ,  $t3=\theta_3$

Après exécution du programme nous trouvons :

$$M_{01} = \begin{bmatrix} c\theta_1 & 0 & s\theta_1 & 0 \\ s\theta_1 & 0 & -c\theta_1 & 0 \\ 0 & 1 & 0 & a_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{12} = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & a_2c\theta_2 \\ s\theta_2 & c\theta_2 & 0 & a_2s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_{23} = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & a_3c\theta_3 \\ s\theta_3 & c\theta_3 & 0 & a_3s\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La matrice de transformation homogène de l'organe terminal par rapport à la base du robot est donnée par :

$$M_{03} = M_{01} M_{12} M_{23} , \text{ ce qui donne :}$$

$$M_{03} = \begin{bmatrix} c(\theta_2 + \theta_3)c\theta_1 & -s(\theta_2 + \theta_3)c\theta_1 & -s\theta_1 & c\theta_1(a_3c(\theta_2 + \theta_3) + a_2c\theta_2) \\ c(\theta_2 + \theta_3)s\theta_1 & -s(\theta_2 + \theta_3)s\theta_1 & -c\theta_1 & s\theta_1(a_3c(\theta_2 + \theta_3) + a_2c\theta_2) \\ s(\theta_2 + \theta_3) & c(\theta_2 + \theta_3) & 0 & a_1 + a_3s(\theta_2 + \theta_3) + a_2s\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

### 2.13.3. Modèle géométrique inverse du robot à commander

Il existe plusieurs méthodes pour calculer le modèle géométrique inverse, Puisque le bras manipulateur est à trois degrés de libertés, on a choisi la méthode de la projection pour faciliter les calculs du modèle géométrique inverse.



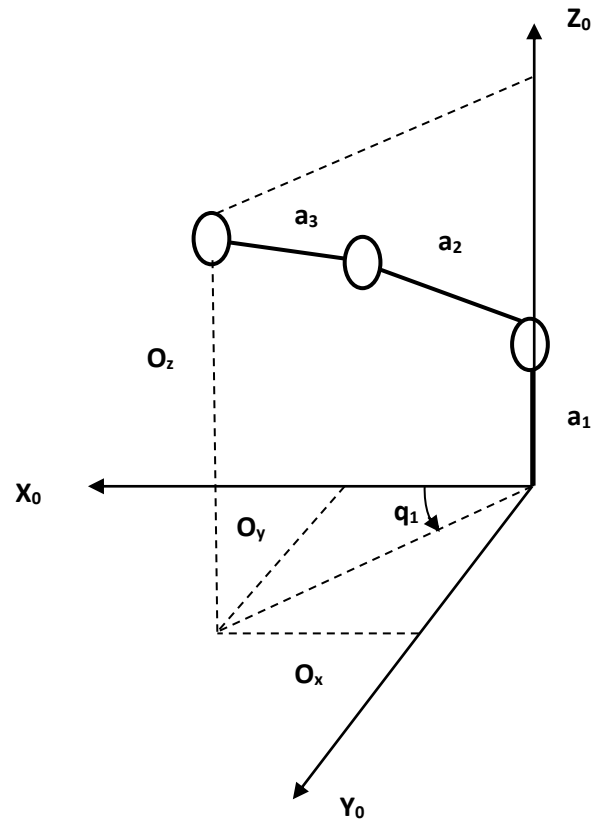


Figure 2.11: projection du robot sur un plan à trois dimensions

nous avons:  $q_1 = \arctan2(O_y, O_x)$

Le problème peut être traité dans un plan

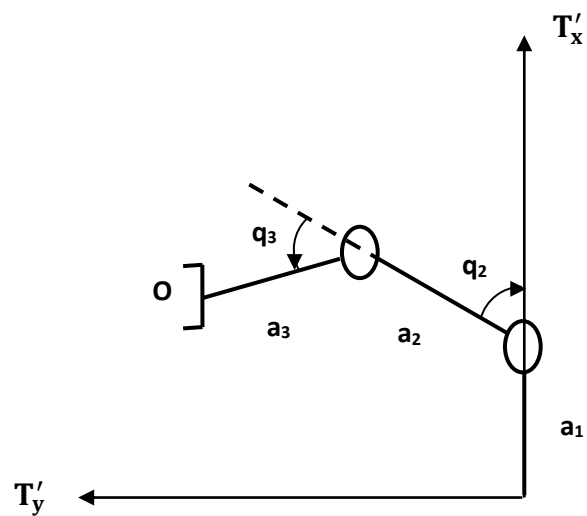


Figure 2.12: projection du robot sur un plan à deux dimensions

Nous avons:

$$\mathbf{T}'_y = \sqrt{\mathbf{O}_x^2 + \mathbf{O}_y^2}$$

$$\mathbf{T}'_x = \mathbf{O}_z$$

Après la projection de  $\mathbf{T}'_x$  et  $\mathbf{T}'_y$  cela donne:

$$\mathbf{T}'_x = a_1 + a_2 \cos q_2 + a_3 \cos(q_2 + q_3)$$

$$\mathbf{T}'_y = a_2 \sin q_2 + a_3 \sin(q_2 + q_3)$$

Nous mettons  $(\mathbf{T}'_x - a_1)^2$  et  $\mathbf{T}'_y{}^2$

$$(\mathbf{T}'_x - a_1)^2 = a_2^2 \cos^2 q_2 + a_3^2 \cos^2(q_2 + q_3) + 2a_2 a_3 \cos q_2 \cos(q_2 + q_3) \dots \dots \dots (1)$$

$$\mathbf{T}'_y{}^2 = a_2^2 \sin^2 q_2 + a_3^2 \sin^2(q_2 + q_3) + 2a_2^2 a_3^2 \sin q_2 \sin(q_2 + q_3) \dots \dots \dots (2)$$

la somme (1) +(2) donne :

$$(1) + (2) = a_2^2 + a_3^2 + 2a_2 a_3 \cos q_3$$

En utilisant  $\cos(a-b) = \cos a \cos b + \sin a \sin b$

$$\frac{\mathbf{T}'_y{}^2 + (\mathbf{T}'_x - a_1)^2 - a_2^2 - a_3^2}{2a_2 a_3} = D \Rightarrow q_3 = \mp \arccos D$$

$$\mathbf{T}'_y = a_2 \sin q_2 + a_3 \sin q_2 \cos q_3 + a_3 \cos q_2 \sin q_3$$

$$= \underbrace{(a_2 + a_3 \cos q_3)}_{K_1} \sin q_2 + \underbrace{a_3 \sin q_3}_{K_2} \cos q_2$$

$$\mathbf{T}'_x - a_1 = a_2 \cos q_2 + a_3 \cos q_2 \cos q_3 - a_3 \sin q_3 \sin q_2$$

$$= \underbrace{(a_2 + a_3 \cos q_3)}_{K_1} \cos q_2 - \underbrace{a_3 \sin q_3}_{K_2} \sin q_2$$

$$\Rightarrow \left. \begin{aligned} T'_x - a_1 &= K_1 \cos q_2 - K_2 \sin q_2 \\ T'_y &= K_1 \sin q_2 + K_2 \cos q_2 \end{aligned} \right\} \Rightarrow \begin{aligned} \sin q_2 &= \frac{K_1 T_y - K_2 (T_x - a_1)}{K_1^2 + K_2^2} \\ \cos q_2 &= \frac{K_2 T_y + K_1 (T_x - a_1)}{K_1^2 + K_2^2} \end{aligned}$$

$$\Rightarrow q_2 = \arctan2 ( K_1 T_y - K_2 (T_x - a_1) , K_2 T_y + K_1 (T_x - a_1) )$$

### 2.14. Conclusion

Au cours de ce chapitre nous avons exposé les concepts de base utilisés en robotique, qui comporte les outils mathématiques nécessaires pour la modélisation géométrique qui a été retenue pour modéliser le bras manipulateur.

## Partie 1 : Conception matérielle et logicielle

### 3.1. Introduction

Dans cette partie de notre travail nous allons utiliser la carte Arduino UNO, pour générer la commande PWM pour les servomoteurs, et voir comment les choisir avec les calculs des différents couples pour pouvoir manipuler le bras manipulateur librement. Ainsi réaliser les différents parties du bras manipulateur.

### 3.2. Arduino UNO

#### 3.2.1 Vue d'ensemble

Le modèle UNO de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATmega328. Le microcontrôleur ATmega328 est un microcontrôleur 8bits avec une vitesse d'horloge de 16Mhz de la famille AVR dont la programmation peut être réalisée en langage C. L'intérêt principal des cartes ARDUINO (d'autres modèles existent) est leur facilité de mise en œuvre. ARDUINO fournit un environnement de développement s'appuyant sur des outils open source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties courantes : gestion des E/S TOR, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus TWI/I2C, exploitation de servomoteurs [4]...

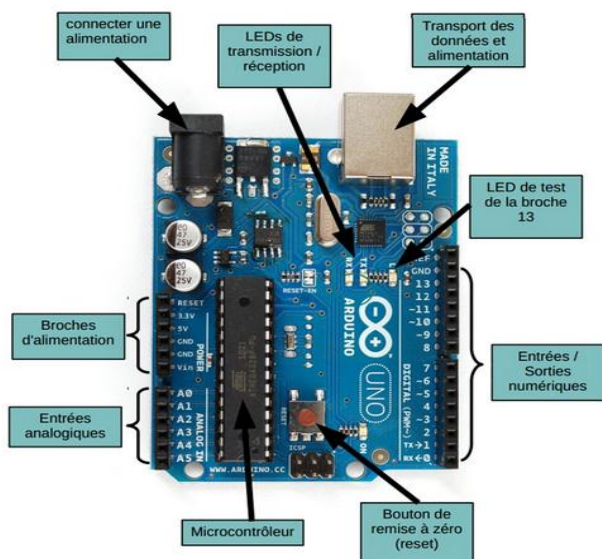


Figure 3.1 : Carte Arduino UNO

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

Figure 3.2 : Brochage de l'ATmega328

### 3.2.2. Alimentation

Arduino UNO peut être alimentée via la connexion USB ou avec une alimentation externe. La source d'alimentation est automatiquement sélectionnée [4].

La plage de tension recommandée est de 7 à 12 volts, les broches d'alimentation sont les suivantes :

- VIN : la tension d'entrée de la carte arduino, quand c'est une source d'alimentation externe (par opposition à 5 volts de la connexion USB ou tout autre source d'énergie réglementés).
- 5V : l'alimentation régulée utilisée pour alimenter le microcontrôleur et les autres composants sur la carte. Cela peut provenir soit de VIN via un régulateur de bord, ou par l'intermédiaire d'un port USB ou d'une autre alimentation 5V régulée.
- 3.3V : une alimentation de 3.3 Volts générée par un circuit intégrée FTDI, avec un courant maximal de 50 mA.
- GND : broche de masse.

### 3.2.3. La mémoire

Le ATmega328 a 32Ko de mémoire flash pour stocker le code dont 0.5 Ko utilisé par le bootloader, 2 Ko de SRAM et 1 Ko de mémoire EEPROM qui peuvent être lues et écrites avec la bibliothèque EEPROM [4].

### 3.2.4. Les entrées et sorties

#### Broche numérique :

Chacune des 14 broches numériques de la carte UNO (numérotées des 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite(broche, HIGH)` [4].

### Broche analogique :

La carte Uno dispose de 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (c-à-d sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction `analogRead()` du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023) [4].

De plus, certaines broches ont des fonctions spécialisées [4] :

- **Communication Série:** Port série (USART) Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données série
- **Interruptions Externes:** Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction `attachInterrupt()` pour plus de détails
- **Timers/Counters:** Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits), on peut les utiliser comme temporisateur ou compteur.
- **Watchdog Timer** : programmable.
- **Impulsion PWM (largeur d'impulsion modulée):** Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.
- **SPI (Interface Série Périphérique):** Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la bibliothèque pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
- **I2C:** Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la bibliothèque `Wire/I2C` (ou TWI - Two-Wire interface - interface "2 fils") .
- **LED:** Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.
- **Reset** : bouton de réinitialisation du microcontrôleur.

### 3.2.5. Communication

Arduino UNO peut communiquer avec un ordinateur, ou un autre arduino, ou d'autres microcontrôleurs, L'ATmega 328 dispose d'une UART (Universal Asynchronous Receiver Transmitter) pour communication série [4].

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignotent lorsque les données sont transmises via la connexion USB vers l'ordinateur

Le logiciel Arduino inclut la librairie Wire, qui simplifie l'utilisation du bus I2C, et la librairie pour communication SPI, pour utiliser la communication SPI (Interface Série Périphérique).

### 3.2.6. Programmation

Pour programmer la carte Aduino UNO, nous avons utilisé le logiciel Arduino prévu à cet effet. C'est un logiciel libre et gratuit développé pour Windows, Mac et Linux. Ce logiciel présente une interface épurée et permet de développer facilement des programmes qui peuvent être transférés sur la carte en branchant celle-ci sur un port USB. Le langage utilisé par le logiciel est basé sur le C, auquel s'ajoutent des instructions spécifiques au langage Arduino. Le logiciel dispose d'une banque d'exemples très riche qui permet d'utiliser des morceaux de codes pré-écrit [4].

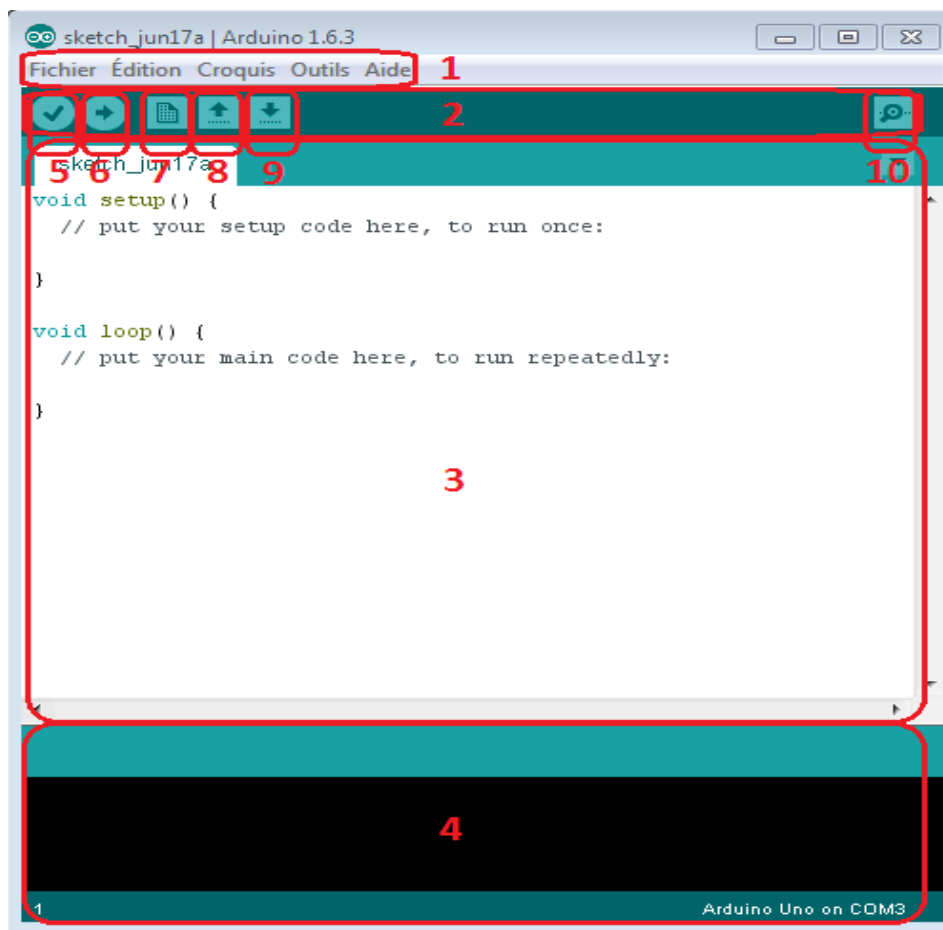


Figure 3.3 : Présentation du logiciel Arduino 1.6.3

- Le cadre numéro 1 : ce sont les options de configuration du logiciel

- Le cadre numéro 2 : il contient les boutons qui vont nous servir lorsque l'on va programmer la carte Arduino
- Le cadre numéro 3 : ce bloc va contenir le programme que nous allons écrire
- Le cadre numéro 4 : celui-ci est important, car il va nous aider à corriger les fautes dans notre programme.
- Bouton 5 : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans le programme
- Bouton 6 : Charge le programme dans la carte Arduino
- Bouton 7 : Crée un nouveau fichier
- Bouton 8 : Ouvre un fichier
- Bouton 9 : Enregistre le fichier
- Bouton 10 : Ouvre le moniteur série.

### 3.3. Commande PWM (Pulse Width Modulation)

PWM signifie Pulse Width Modulation, modulation par largeur d'impulsion.

La commande par PWM permet de produire simplement une variation de la valeur moyenne d'une tension.

En effet, si on considère une période  $T$  (donc  $1/F$ ) et si on appelle  $T_{on}$  le temps où le signal est à 1, et  $T_{off}$  le temps où le signal est au repos alors on a  $T = T_{on} + T_{off}$ . En agissant sur  $T_{on}$  on modifie la valeur moyenne de  $V$  sur cette période. En choisissant une fréquence adéquate et en agissant sur le rapport cyclique ( $T_{on}/T_{off}$ ) on fait varier la valeur moyenne de  $V$  aux bornes du moteur.

#### ➤ commander un servomoteur avec la PWM

Le principe de base est assez simple. Il suffit d'envoyer une impulsion et c'est le temps que durera cette impulsion qui déterminera l'angle du servomoteur [4]. Ce temps d'impulsion est de quelques millisecondes et doit être répété à intervalle régulier (toutes les 20 ms). Si le temps d'impulsion varie d'un fabricant à l'autre,

les valeurs suivantes sont assez standard:

- 1 ms = 0 degré
- 1.50 ms = 90 degrés
- 2 ms = 180 degrés



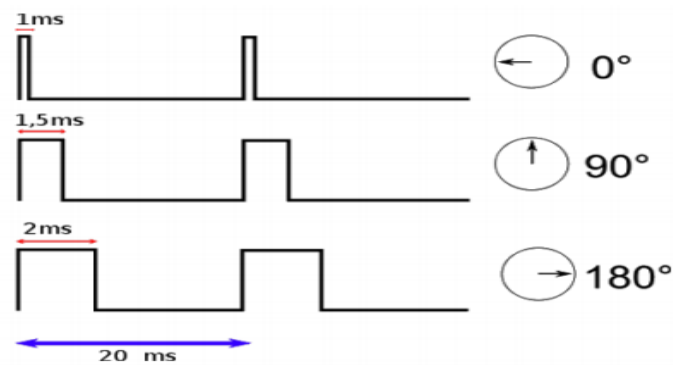


Figure 3.4 : Relation PWM / Position de l'arbre

Pour commander un servomoteur avec la carte ARUIDO UNO il existe une

librairie (LIBRAIRIE SERVO.H) qui est incluse dans l'IDE, pour faciliter son pilotage.

`servo.attach(pin)` : permet d'attacher une pin à un servomoteur.

`servo.write(angle)` : demande au servomoteur de se positionner à l'angle transmis en paramètre [4].

### ➤ L'électronique d'asservissement

La consigne donnée par l'utilisateur (dans notre cas il va s'agir du signal envoyé par la carte Arduino), est comparée par rapport à la position réelle de l'axe du moteur. Ainsi, s'il y a une différence d'angle entre la consigne et l'angle mesuré par le capteur (le potentiomètre qui est fixé sur l'axe du servomoteur) le comparateur va commander le moteur et le faire tourner jusqu'à ce que cette différence s'annule.

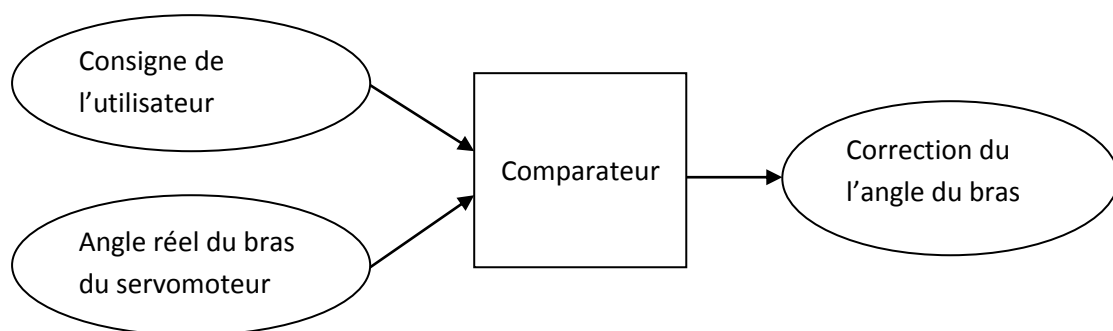


Figure 3.5 : Principe de fonctionnement de l'électronique de commande

### Exemple : programmation d'un servomoteur

```
#include <Servo.h>
Servo monservo; // crée l'objet pour contrôler le servomoteur
void setup()
{
  pinMode(9, OUTPUT); //mettre la broche 9 en sortie
  monservo.attach(9); // utilise la broche 9 pour le contrôle du servomoteur
  monservo.write(0); // initialisation du servomoteur à 0°
}

void loop()
{
  monservo.write(45); // commander le servomoteur à 45°
  delay(1000);        // attendre une seconde
  monservo.write(90); // commander le servomoteur à 90°
  delay(1000);        // attendre une seconde
  monservo.write(180); // commander le servomoteur à 180°
  delay(1000);        // attendre une seconde et répéter la tâche
}
```

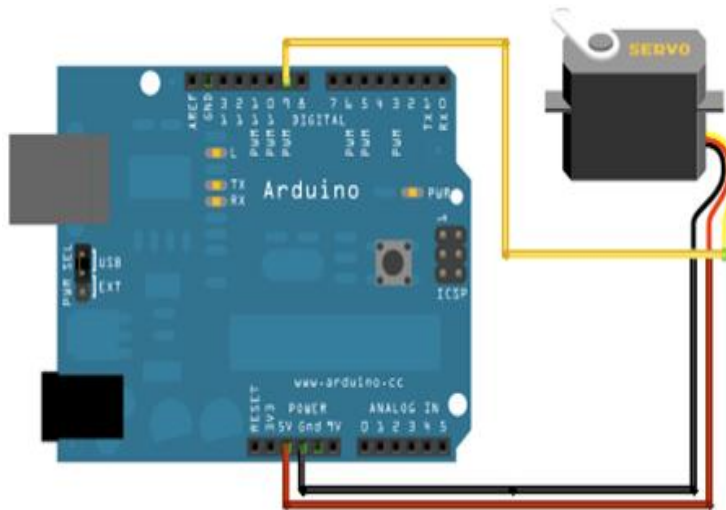


Figure 3.6 : Schéma de branchement d'un servomoteur avec Arduino UNO

### 3.4. Les Capteurs

Afin de donner au robot un certain aspect d'autonomie et d'intelligence, nous avons optés pour l'utilisation de deux capteurs [2].

Dans notre projet nous utilisons la LED infrarouge et Le Phototransistor comme capteur de présence de pièces, la LED infrarouge émet la lumière vers Le Phototransistor, et si le fuseau lumineux est coupé ça veut dire que le capteur capte la présence d'une pièce, et le rôle du capteur de force c'est de signaler la prise de pièce par la pince.

### 3.4.1. Le capteur de présence de pièce

Le capteur de présence de pièce est constitué d'une Led infrarouge pour l'émission de la lumière, et un phototransistor pour la réception de la lumière.

- **La LED infrarouge:**

Une diode émettrice infrarouge est construite à partir de semi-conducteurs. Il s'agit d'une diode un peu particulière, qui a la propriété d'émettre de la lumière infrarouge quand un courant la parcourt (de l'Anode vers la Cathode) [2].



Figure 3.7 : LED infrarouge

- **Le Phototransistor :**

Transistor dont la base est sensible aux rayonnements lumineux. Il crée un courant lorsque qu'il est éclairée sa sensibilité est entre 100 et 400 fois supérieure à celle d'une photodiode [2].

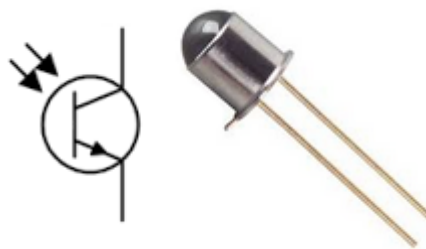


Figure 3.8 : Phototransistor

### 3.4.2. Le capteur de présence de pièce sur la pince

Nous avons utilisé un capteur de force résistif rond d'un demi pouce de diamètre (12,5mm) est sensible à la pression, il est très utilisé dans la robotique, Sa résistance varie selon l'effort qui est appliquée sur sa zone utile. Plus la force est importante, plus la résistance diminue. Lorsqu'aucune pression n'est appliquée sur le capteur, sa résistance est très grande ( $1M\Omega$ ). Le capteur dispose d'un connecteur à deux pôles permettant de l'utiliser sur une plaque d'essai (breadboard). Le capteur est pourvu d'une fixation adhésive au dos pour son montage. Ce type de capteur est très simple à mettre en œuvre et même si sa sensibilité ne permet pas de l'utiliser pour mesurer très précisément un poids, il est idéal pour la détection de présence d'objets. Sa dimension : 45\*7mm (épaisseur<1mm) [2].

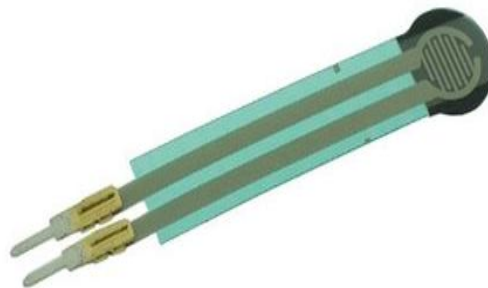


Figure 3.9 : Capteur de force FSR01

## 3.5. Afficheur à cristaux liquides (LCD)

### 3.5.1. Introduction

Les afficheurs à cristaux liquides, appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA). Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), et aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétro-éclairage. Cette fonction fait appel à des LED montées derrière l'écran du module.

## 3.5.2. Schéma fonctionnel

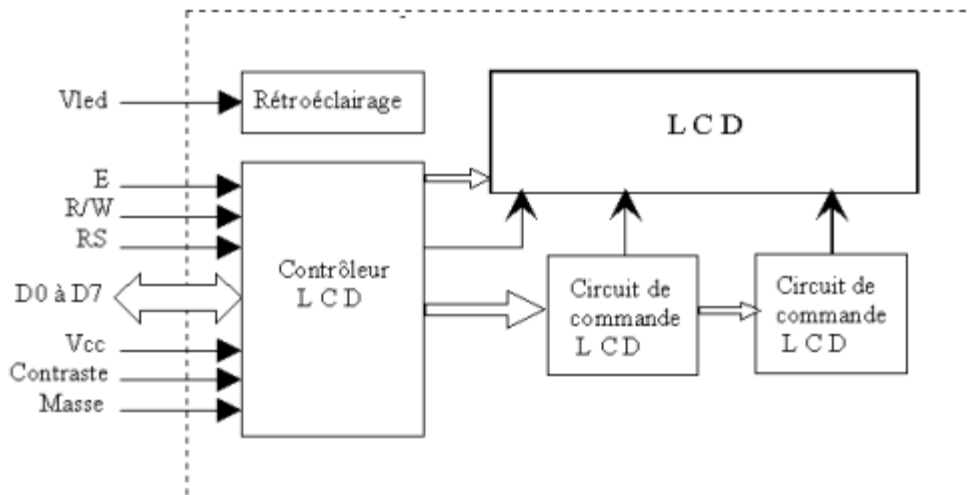


Figure 3.10 : Schéma fonctionnel d'un afficheur LCD

Comme le montre le schéma fonctionnel, l'affichage comporte d'autres composants que l'afficheur à cristaux liquides (LCD) seul. Un circuit intégré de commande spécialisé, le LCDController, est chargé de la gestion du module. Le "contrôleur" remplit une double fonction: d'une part il commande l'affichage et de l'autre se charge de la communication avec l'extérieur.

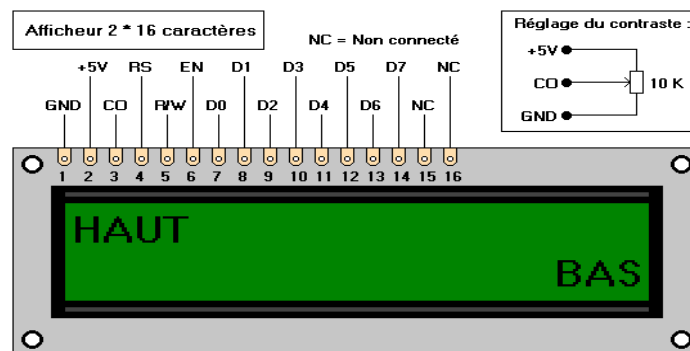


Figure 3.11 : Afficheur LCD 2\*16

## 3.5.3. Brochage

Broche	Nom	Fonction
1	Vss	Masse
2	Vdd	Alimentation positive 5V
3	Vo	Cette tension permet en la faisant varier entre 0 et 5V, le réglage du contraste de l'afficheur
4	RS	Sélection du registre (Register Select) grâce à cette broche, l'afficheur est capable de faire la différence entre une commande et

		une donnée. Un niveau bas indique une commande et un niveau haut indique une donnée
5	R/W	Lecture ou écriture (Read/Write) L : Ecriture/H : Lecture
6	E	Entrée de validation(Enable) active sur front descendant. Le niveau haut doit être maintenue pendant 450ns à l'état haut.
7	D0	
8	D1	
9	D2	
10	D3	D0->D7 Bus de données bidirectionnel 3 états (haute impédance lorsque E=0)
11	D4	
12	D5	
13	D6	
14	D7	
15	A	Anode rétro éclairage (+5V)
16	K	Cathode rétro éclairage (masse)

Tableau 3.1: Brochage d'un Afficheur LCD

Les broches 15 et 16 ne sont présentes que sur les afficheurs LCD avec retro-éclairage.

### 3.5.4. La mémoire

L'afficheur possède deux types de mémoire, la DD RAM et la CG RAM. La DD RAM est la mémoire d'affichage et la CG RAM est la mémoire du générateur de caractères.

### 3.5.5. Commande d'un afficheur LCD

Deux modes de fonctionnement de l'afficheur sont disponibles, le mode 4 bits et le mode 8 bits, modes que l'on choisira à l'initialisation de l'afficheur.

#### Mode 8 bits :

Dans ce mode, les données sont envoyées à l'afficheur sur les broches D0 à D7. On place la ligne RS à 0 ou à 1 selon que l'on désire transmettre une commande ou une donnée. Il faut aussi placer la ligne R/W à 0 pour indiquer à l'afficheur que l'on désire effectuer une écriture. Il reste à envoyer une impulsion d'au moins 450 ns sur l'entrée E, pour indiquer que des données valides sont présentes sur les broches D0 à D7. L'afficheur lira la donnée sur le front descendant de cette entrée. Si on désire au contraire effectuer une lecture, la procédure est identique, mais on place cette fois la ligne R/W à 1 pour demander une lecture. Les données seront valides sur les lignes D0 à D7 lors de l'état haut de la ligne E.

### Mode 4 bits :

Il peut, dans certains cas, être nécessaire de diminuer le nombre de fils utilisés pour commander l'afficheur, comme, par exemple lorsqu'on dispose de très peu de broches d'entrées sorties disponibles sur un microcontrôleur. Dans ce cas, on peut utiliser le mode quatre bits de l'afficheur LCD. Dans ce mode, seuls les 4 bits de poids fort (D4 à D7) de l'afficheur sont utilisées pour transmettre les données et les lire. Les 4 bits de poids faible (D0 à D3) sont alors connectés à la masse. On a donc besoin, hors alimentation de sept fils pour commander l'afficheur. Les données sont alors écrites ou lues en envoyant séquentiellement les quatre bits de poids fort suivi des quatre bits de poids faible.

#### 3.5.6. Branchement de l'afficheur LCD avec Arduino UNO :

L'afficheur LCD possède deux mode de fonctionnement le 1er c'est le mode 8 bits qui est le branchement normale des 8 lignes de données; le 2eme est le mode 4 bits qui utilise seulement 4 lignes et les autres lignes sont mises a la masse donc il va envoyer 4 bit puis 4 autre bits à des fréquences élevée, c'est le mode que nous allons utiliser dans notre cas parce qu'il nous permettra d'économiser des lignes et nous évitera l'encombrement. La figure ci-dessous montre le schéma de branchement de l'afficheur LCD avec l'Arduino UNO en mode 4bits.

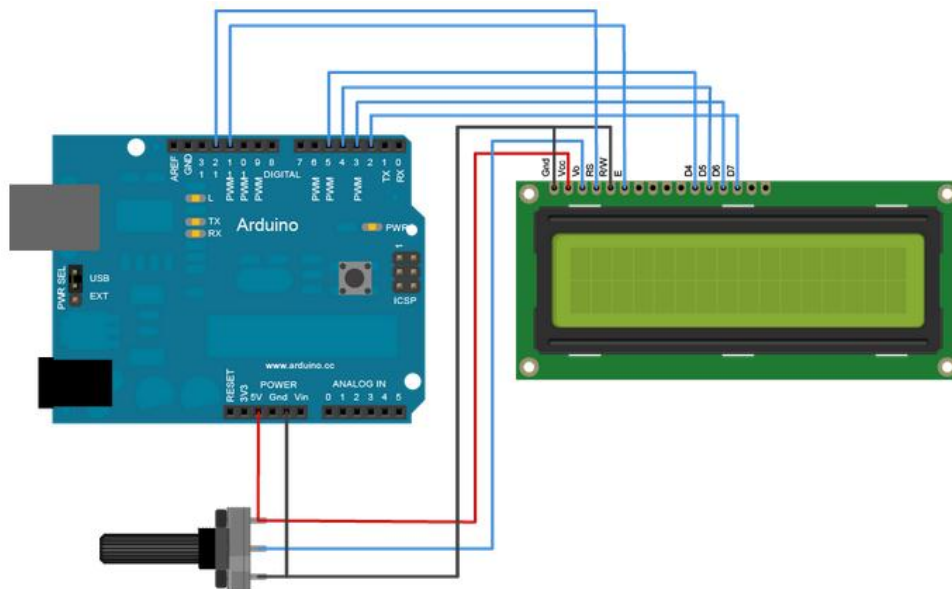


Figure 3.12 : Schéma de branchement de l'afficheur LCD avec Arduino UNO

### 3.6. Simulation

#### Représentation d'ISIS :

ISIS (Intelligent Schematic Input Système) est le module de saisie de schéma électronique et de simulation de la suite logicielle Proteus. Et ARES est le module de routage et de conception de circuit imprimé de la suite logicielle Proteus.

Avant de passer à la réalisation pratique de notre système nous avons eu recours à la simulation des différentes parties du système. Pour cela nous avons utilisé le logiciel ISIS qui est un très bon logiciel de simulation en électronique. La simulation permet d'ajuster et de modifier le circuit comme si on manipulait un montage réel. Ceci permet d'accélérer le prototypage et de réduire son coût.

Il faut toujours prendre en considération que les résultats obtenus de la simulation sont un peu différents de celles du monde réel.

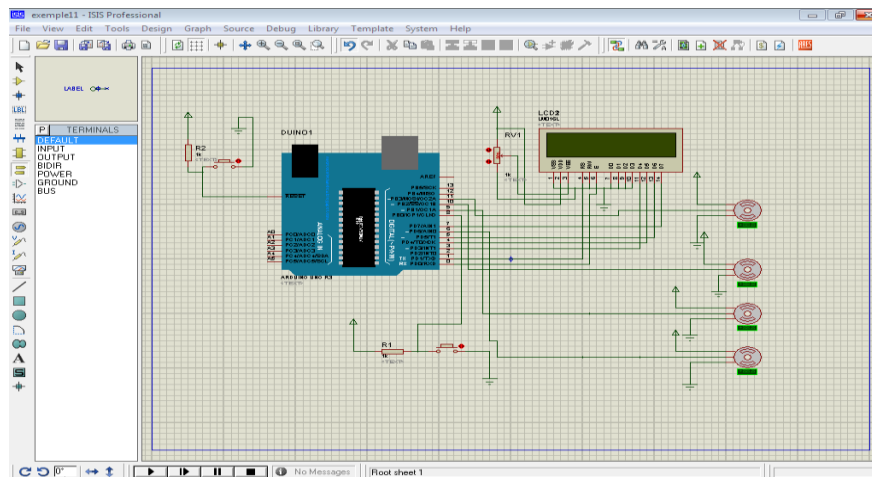


Figure 3.13: Fenêtre du logiciel de simulation Proteus ISIS

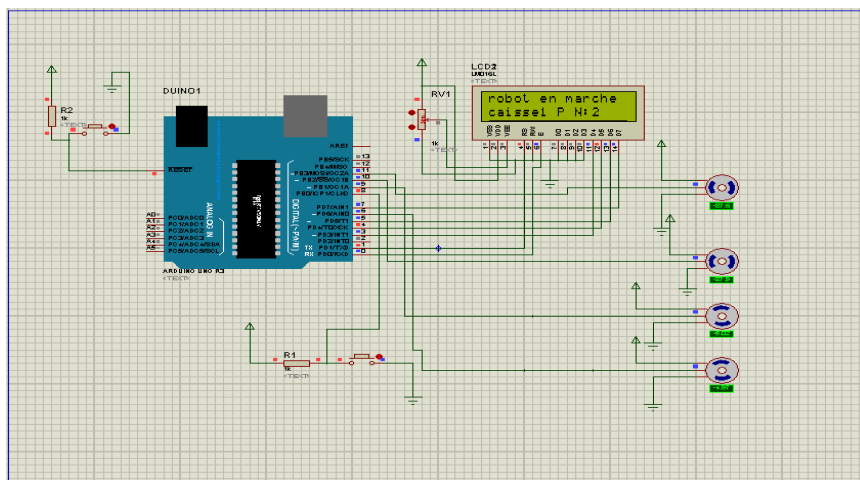


Figure 3.14: Le logiciel ISIS en mode simulation du projet



### 3.7. Choix des actionneurs

Les mouvements du bras manipulateur seront exécutés par des servomoteurs pour des raisons de précision, et de simplification de commande.

➤ **Problèmes :**

- quelle est la valeur du couple nécessaire à notre bras manipulateur ?
- Comment calcule-t-on ce couple ?
- quelles longueurs utiliser pour chaque porteur du bras manipulateur ?

➤ **Approche :**

Par soucis de simplicité et d'efficacité nous avons adopté le raisonnement suivant :

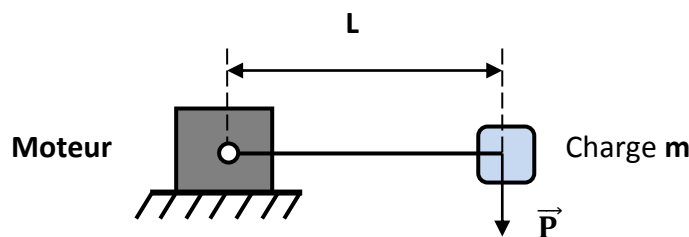


Figure 3.15 : Tige maintenue par un moteur

Considérons une tige de masse négligeable de longueur «  $L$  », portant sur l'une de ses extrémités une charge de masse «  $m$  », l'ensemble peut être entraîné dans un mouvement de rotation par l'arbre d'un moteur fixé à l'autre extrémité.

Pour maintenir la tige parallèle au sol (en équilibre), le moteur doit appliquer un moment de couple  $C_{eq}$  égale au moment de la force «  $P$  », due à la charge [6].

$$C_{eq} = |\vec{P}| * L$$

Pour faire tourner l'ensemble (tige + charge), le moteur doit être capable de fournir un couple supérieur à  $C_{eq}$ . Les performances dynamiques seront d'autant meilleures que le couple moteur choisi est plus grand que  $C_{eq}$  calculé auparavant [6].

➤ **Calculs des couples nécessaires pour le bras manipulateur**

Pour les calculs, une masse de **50g**, pour l'organe terminal porté par l'effecteur, **50g** pour les trois servomoteurs, **10g** pour le mini-servomoteur, **50g** pour le troisième segment (L1), **70g** pour le deuxième segment(L2) et **100g** pour le premier segment(L3). Pour la longueur des porteurs du bras manipulateur nous avons :

$L1 = 10 \text{ cm}$ ,  $L2 = 10 \text{ cm}$ ,  $L3 = 15 \text{ cm}$

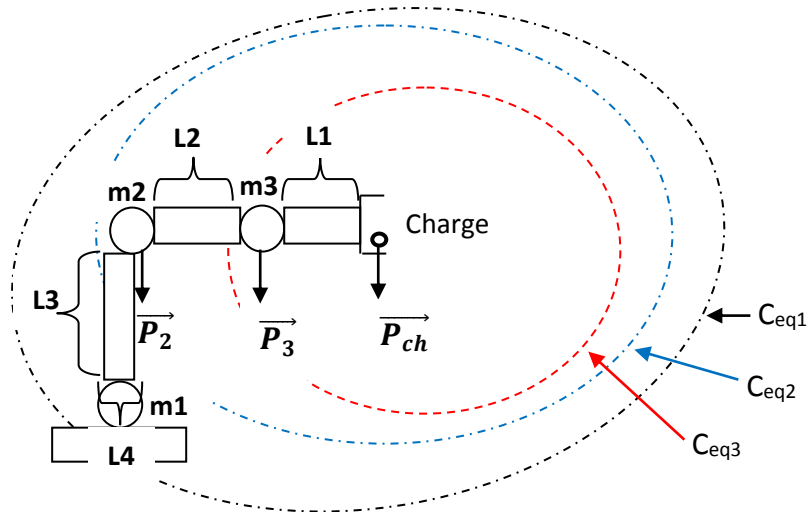


Figure 3.16 : Couples nécessaires pour le bras manipulateur

Puisque l'unité du couple des servomoteurs est le Kg.cm alors on prend  $P = m$

$m$  : la masse (kg)

- **Calcul du moment du couple  $C_{eq3}$  nécessaire pour le moteur 3 portant l'effecteur**

$$C_{eq3} = P_{ch} * L1$$

$$C_{eq3} = 0.11 * 10 = 1.1 \text{ kg.cm}$$

- **Calcul du moment du couple  $C_{eq2}$  nécessaire pour le moteur 2 portant le système (moteur 3 + charge)**

$$C_{eq2} = P_{ch} * (L1+L2) + P_3 * L2$$

$$C_{eq2} = 0.11*20 + 0.12*10 = 3.4 \text{ kg.cm}$$

- **Calcul du moment du couple  $C_{eq1}$  nécessaire pour le moteur 1 portant le système (moteur2 + moteur 3 + charge)**

$$C_{eq1} = P_{ch} * (L1+ L2 + L4) + P_3 *(L2 + L4) + P_2 * L4$$

On a  $L4$  t'en vers zéros alors :

$$C_{eq1} = P_{ch} * (L1+ L2 ) + P_3 * L2$$

$$C_{eq1} = 0.15*20 + 0.11*10 = 4.1 \text{ kg.cm}$$

Après les calculs des différents couples nécessaires pour le bon fonctionnement du bras manipulateur nous avons opté à utiliser des servomoteurs Futaba S3010 de la marque Futaba, vue leur couples qui concordes avec le calcul des couples nécessaires pour le bras Manipulateur, et le servomoteur TowerPro SG90 pour la pince.



Figure 3.17 : Servomoteur Futaba S3010

### Caractéristiques du servomoteur Futaba S3010

Dimensions : 40x20x38.1mm

Poids : 41g

Couple : 5.2kg.cm (4.8 Volts) / 6.5kg.cm (6 Volts)

Vitesse : 0.20s/60°(4.8 Volts) / 0.16s/60°(6 Volts)

Alimentation : 4.8 Volts , 6 Volts

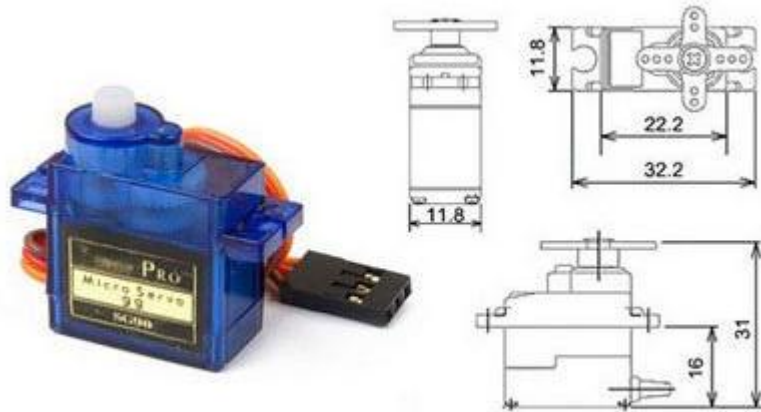


Figure 3.18 : Servomoteur TowerPro SG90

### Caractéristiques TowerPro SG90

Dimensions : 22mm x 11.5mm x 27mm

Poids : 9 g

Vitesse : 0.12 sec/60° sous 4.8v

Couple : 1.2kg/cm sous 4.8v

Tension : 4.8v à 6v

### 3.8. Tache du bras manipulateur

Dans notre projet le bras manipulateur a comme tache de déplacer les pièces présentes sur le tapis roulant0 vers la caisse1 (tapis roulant1) et la caisse2 (tapis roulant2).

Quand le capteur de pièce capte la présence d'une pièce, les actionneurs seront actionnés, et le bras manipulateur commence sa course, vers le tapis roulant0 pour charger la pièce à l'aide de la pince, puis il la décharge sur le tapis roulant1, jusqu'à arrivé à 5 pièces sur la caisse1, la 6ème pièce sera déchargée sur la caisse2, jusqu'à arrivé à 5 pièces sur la caisse2, (5 pièces sur la caisse1, et 5 pièces sur la caisse2)

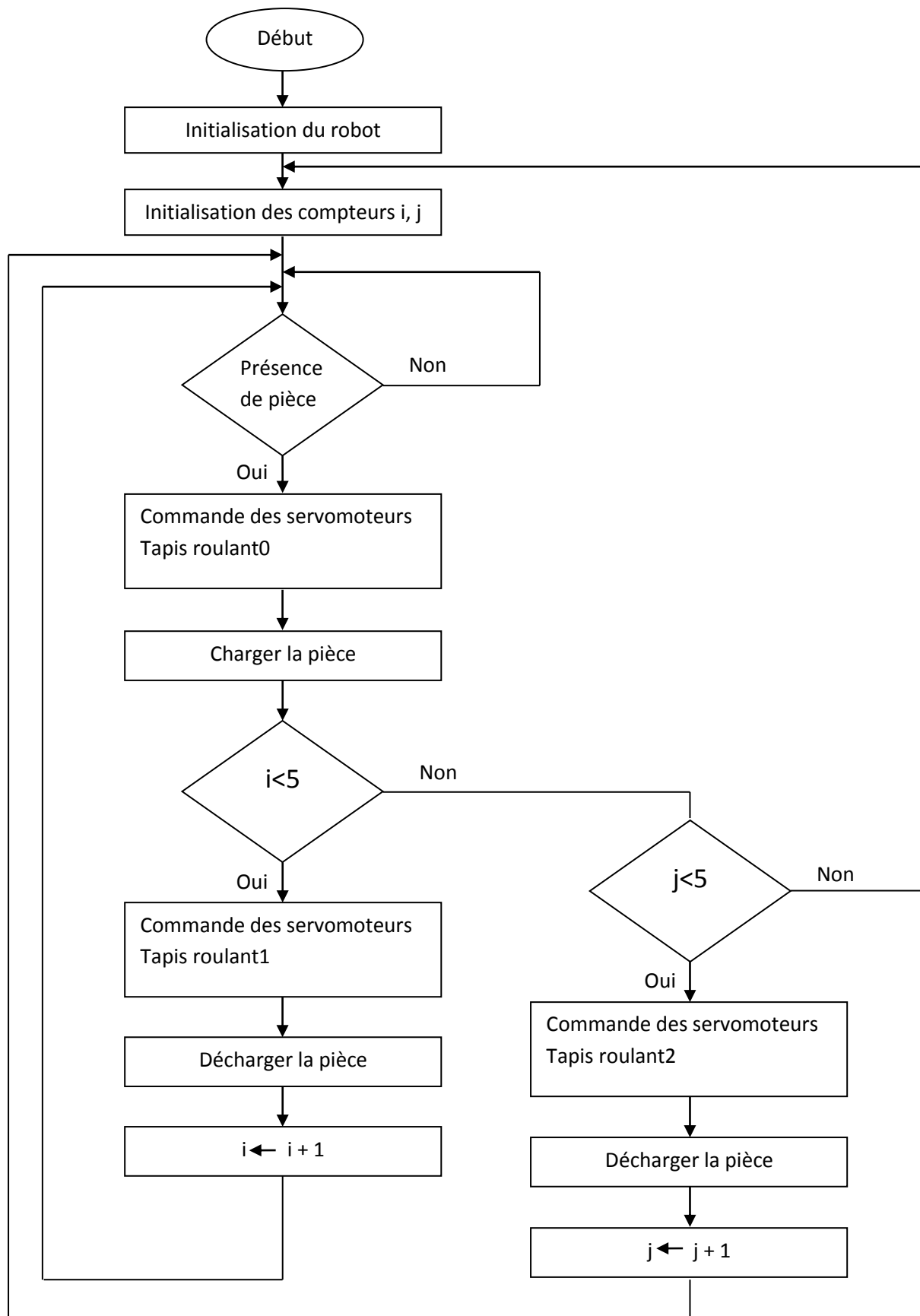


Figure 3.19 : Organigramme montrant la tâche du bras manipulateur

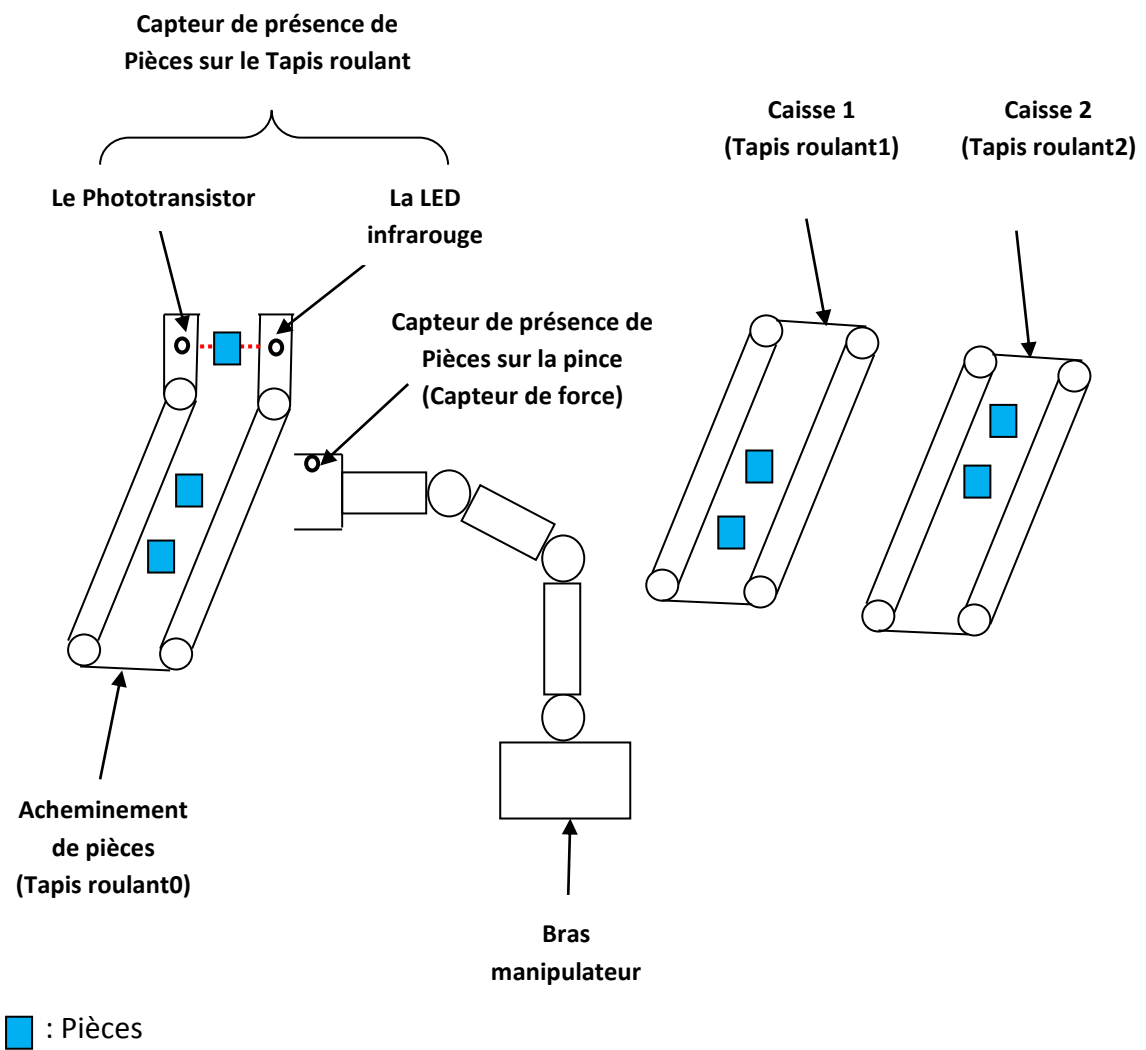


Figure 3.20 : Tache du bras manipulateur

### Partie 2 : Réalisation pratique

#### 3.9. Introduction

Dans cette partie de notre travail nous allons voir les différents éléments constitutifs du bras manipulateur et comment les assembler. Nous allons également faire des tests afin et voir son fonctionnement.

#### 3.10. Structure mécanique du bras manipulateur

Nous avons opté pour le bois en raison de sa disponibilité et sa faible densité. Nous avons conçu une structure mécanique conformément à un modèle à trois degrés de liberté.

#### 3.11. Différentes parties du robot

Le bras est constitué de plusieurs parties dont toutes les articulations sont rotoïdes.

##### 3.11.1. La base

Elle est composée de deux pièces identiques carré 15\*15cm, nous les avons relié à l'aide de quatre pièces d'une hauteur de 5cm, une cavité a été découpée de telle manière à avoir une ouverture pour le passage du servomoteur Futaba S3010, à l'aide des visse.



Figure 3.21 : Vue de dessus et de face de la base



Figure 3.22 : Vue de dessus de la base et le servomoteur implanté dedans

### 3.11.2. Premier corps

C'est un segment fait en bois en forme de **U** de 15\*7\*3cm relié à au servomoteur implanté sur la base, une cavité a été découpée de telle manière à avoir une ouverture pour le passage du servomoteur Futuba S3010 sur ce segment de manière à ce qu'il soit stable.



Figure 3.23 : Vue de dessus et de face du premier corps





Figure 3.24 : Vue de dessus du premier corps et le servomoteur implanté dedans

### 3.11.3 Deuxième corps

C'est un segment fait en bois, composé de deux pièces identiques de 10\*5.5\*cm relié au servomoteur implanté sur le premier corps, une cavité a été découpée sur l'une des pièces de telle manière à avoir une ouverture pour le passage du servomoteur Futaba S3010 sur ce segment de manière à ce qu'il soit stable.



Figure 3.25 : Vue de dessus du deuxième corps



Figure 3.26 : Vue de face du deuxième corps et le servomoteur implanté dedans

### 3.11.4. Troisième corps

C'est un segment fait en bois, composé de deux pièces identiques de 7\*3\*cm reliées entre elles avec une pièce de 3cm, une extrémité est reliée au servomoteur fixé sur le deuxième corps, l'autre extrémité est reliée à l'organe terminal.



Figure 3.27 : Vue de dessus du troisième corps

### 3.11.5. L'organe terminal

C'est une pince qui est fabriquée avec du bois, elle est composée de deux doigts, un doigt fixe et l'autre mobile doté d'un servomoteur TowerPro SG90, son rôle est de saisir l'objet désiré.



Figure 3.28 : Vue de dessus de la pince



Figure 3.29 : Vue de dessous de la pince

### 3.11.6. Assemblage complet du bras manipulateur

Après avoir cité les différentes parties du bras manipulateur, nous allons les assembler pour avoir un bras manipulateur complet, la figure ci-dessous le montre.



Figure 3.30 : Assemblage complet du bras manipulateur

### 3.12. Carte de commande pour le bras manipulateur

La carte de commande est constituée d'une carte Arduino UNO, alimentée par l'ordinateur, qui délivre les signaux PWM pour commander les différents servomoteurs qui contrôlent le bras manipulateur, qui sont de leurs cotés alimentés avec une tension de 4volts délivré par un générateur de tension, et un capteur de présence de pièces que nous avons utilisé comme un bouton poussoir connecté sur la carte Arduino UNO, la figure 3.14 montre le schéma de branchement sur le simulateur ISIS Proteus. La figure ci-dessous montre le schéma de branchement sur la platine d'expérimentation.



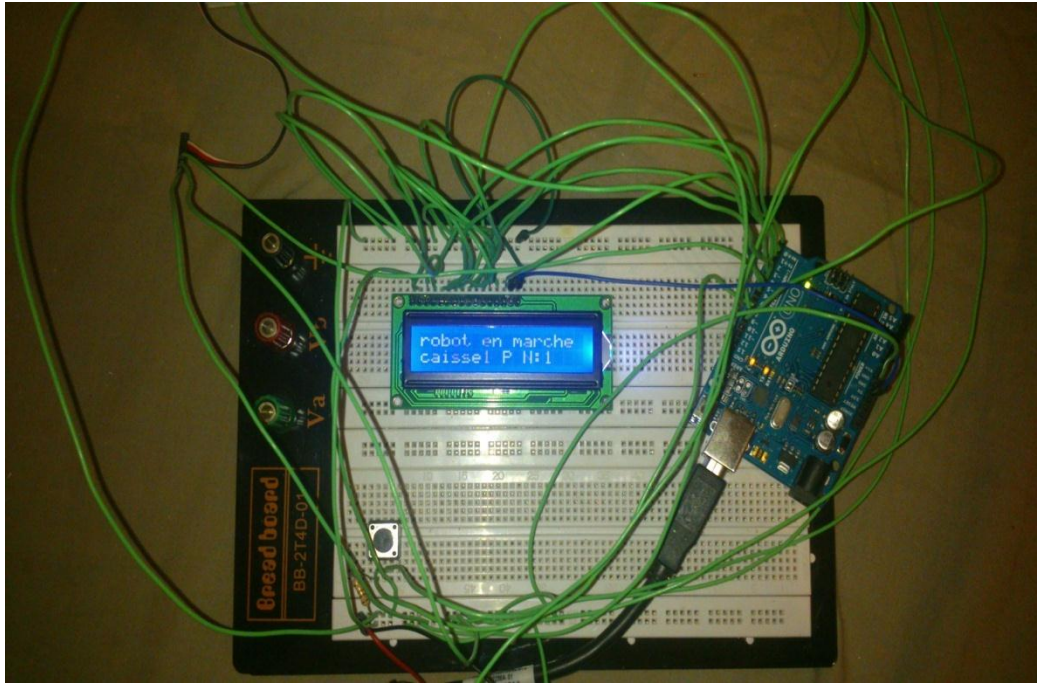


Figure 3.31 : Schéma électrique de commande pour le bras manipulateur

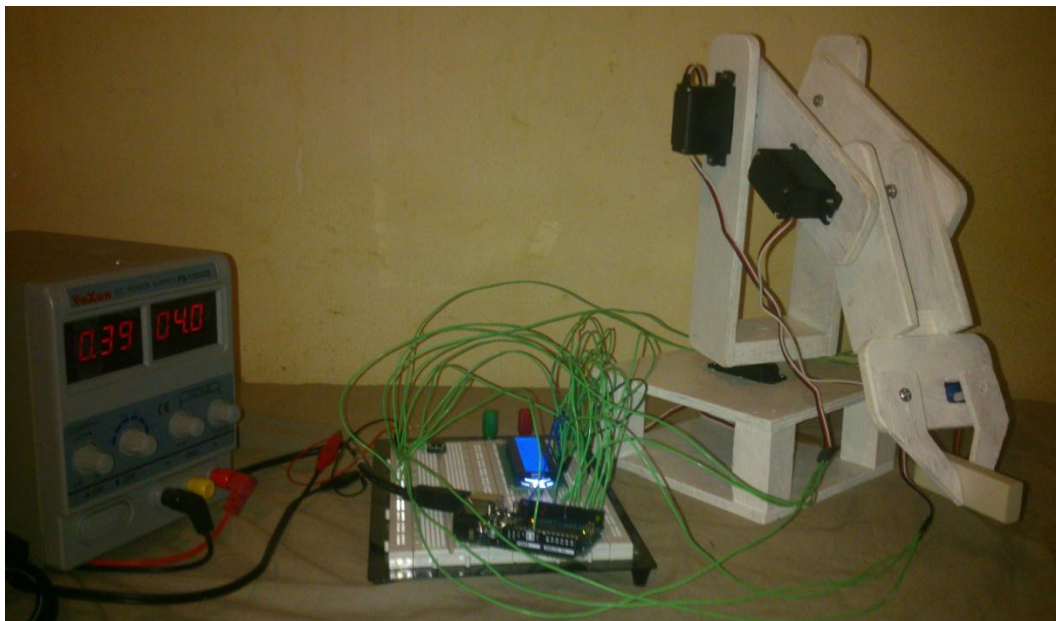


Figure 3.32 : Bras manipulateur et sa carte de commande

### 3.13. Conclusion

Dans ce chapitre, nous avons pu décrire les différents éléments constitutifs du bras manipulateur, et injecter le programme dans la carte Arduino UNO, pour commander le bras manipulateur et ainsi voir son fonctionnement lorsqu'il effectue sa tâche.

## Conclusion générale

---

Au terme de ce travail, nous avons pu concevoir et réaliser un bras manipulateur à trois degrés de liberté qu'on a commandé avec une carte Arduino UNO.

Ce travail nous a permis de mettre en valeur nos connaissances théoriques et pratiques, que nous avons cumulées pendant toute la durée de nos études, il nous a permis aussi d'acquérir une expérience nouvelle dans la programmation des cartes Arduino.

Nous nous sommes intéressés dans notre travail à l'aspect théorique de la robotique mais surtout à l'aspect pratique, ce qui a présenté la difficulté majeure de notre démarche, et cela parce que dans notre formation la théorie est prédominante. Et en s'intéressant à cet aspect nous avons pu mettre en relief les différentes notions théoriques que nous avons acquises. Il est évident que notre travail est loin d'être parfait, mais il peut constituer une base très intéressante pour les promotions à venir qui voudront travailler sur ce type de sujet.

Dans le prolongement direct de ce travail plusieurs études sont à envisager :

- Ajout d'une vision artificielle pour le bras manipulateur pour qu'il soit plus autonome
- Intégrer une commande en vitesse pour le bras manipulateur sous Labview à l'aide de la liaison Arduino-Labview
- Intégrer une commande par neuro-flou pour le bras manipulateur sous Matlab à l'aide de la liaison Arduino-Matlab

## Références bibliographiques

---

[1] **Jacques Gangloff** « Cours de Robotique » Université de strasbourg 2010.

[2] **Alain Pruski** « Robotique générale » 1988, 207p.

[3] **M. Cazin, J. Metge** « Mécanique de la robotique » 1989 259P.

[4] **Massimo BANZI** « Démarrez avec Arduino » 2011 102P.

[5] Oukaci S, Ouazar Y : Mémoire UMMTO « Conception et réalisation d'un bras manipulateur à six degrés de liberté autonome assisté par la vision artificielle »  
ING.Auto/N°15/2011.

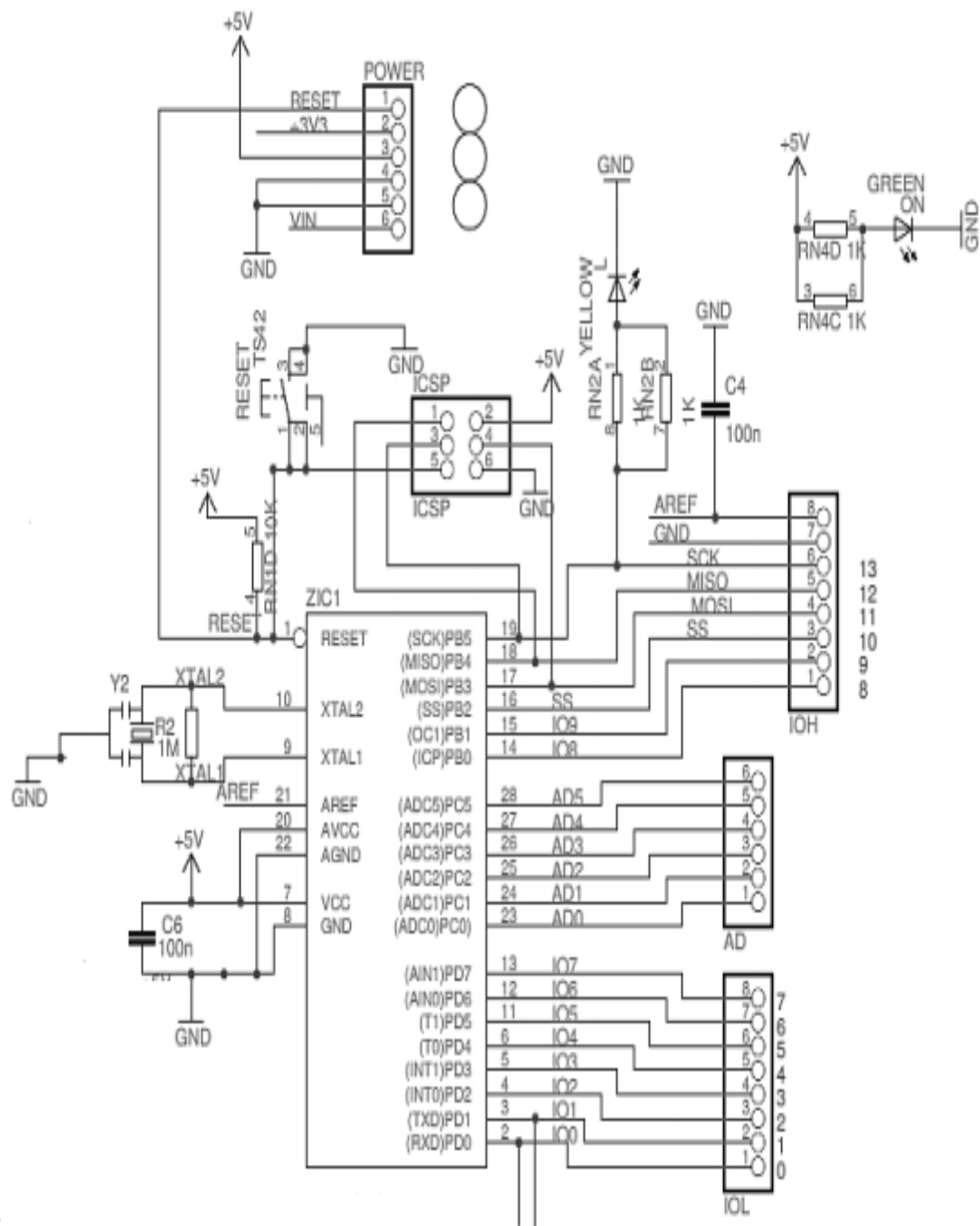
[6] Chaib A, Mahfouf S : Mémoire UMMTO« Etude et réalisation d'un bras articulé commandé par un micro-ordinateur » ELN.DEUA. 32/2008.

[7] <http://www.arduino.cc>

[8] <http://www.mon-club-elec.fr>

## Annexe

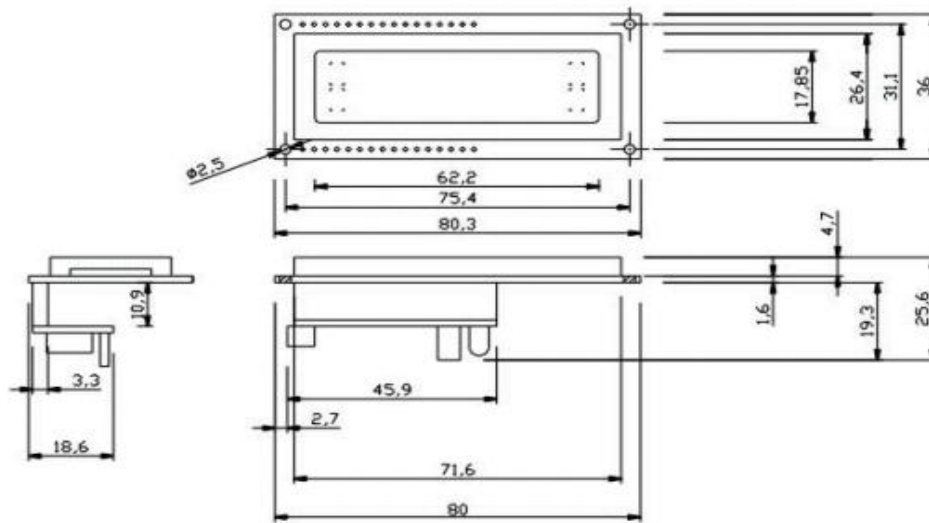
### Schéma simplifié de la carte Arduino UNO





## Annexe

### Caractéristiques de l'afficheur LCD 2\*16



### L’Affichage d’un caractère

Le tableau résume l’affichage des caractères sous LCD en fonction de code ASCII.

Upper & Lower ASCII	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000				0@P`P								-タミαp				
xxxx0001	(2)			!1AQa9								。アチ4äq				
xxxx0010	(3)			"2BRbr								「イツ/ρθ				
xxxx0011	(4)			#3CScs								」ウテモε∞				
xxxx0100	(5)			\$4DTdt								、イトμΩ				
xxxx0101	(6)			%5EUeu								・オナ1εÜ				
xxxx0110	(7)			&6FVfv								ヲカニヨρΣ				
xxxx0111	(8)			'7GW9w								アキヌラgπ				
xxxx1000	(1)			<8HXhx								ィクネリJX				
xxxx1001	(2)			>9IYiy								ウケノル`y				
xxxx1010	(3)			*:JZjz								エコンレjチ				
xxxx1011	(4)			+;K[k<								オサヒロ*π				
xxxx1100	(5)			,<L¥ll								カシフワφπ				
xxxx1101	(6)			-=M]m>								ユスヘンも÷				
xxxx1110	(7)			.>N^n→								ヨセホ`ñ				
xxxx1111	(8)			/?O_o←								ッソマ°ö				

**Conception et réalisation d'un bras manipulateur à trois degrés de liberté**  
**Commandé par une carte Arduino UNO, Servomoteur, Robot manipulateur,**  
**Afficheur LCD, Capteur, Bouton poussoir, Modèle géométrique direct, Modèle**  
**Géométrique inverse, Convention de Denavit-Hartenberg(DH).**