

Réf :

**Mémoire de Fin d'Etudes
En vue de l'obtention du diplôme :**

MASTER

Filière : Electromécanique
Option : Automatique & Informatique Industrielle

Thème

**Réalisation & Commande d'un prototype de
Robot Palettiseur ABB-IRB460**

Présenté par : - Ghebouli Yassine
-Benyahia Samy

Devant le jury :

Président	: Bekouche Toufik	Université de Bordj Bou Arreridj
Examineur 1	: Ayadi Madani Toufik	Université de Bordj Bou Arreridj
Examineur 2	: Maghlaoui Issam	Université de Bordj Bou Arreridj
Encadreur	: Zaoui Fares	Université de Bordj Bou Arreridj

Remerciements

Avant toute chose, je remercie « Allah », qui nous aide et nous donne la volonté,
le courage et la patience durant ces années d'étude.

Je souhaite adresser mes remerciements les plus chères aux personnes qui ont
Contribués à l'élaboration et l'avancement de mon sujet d'étude.

Je désire aussi remercier les enseignants de l'université Mohamed Elbachir El
Ibrahimi, qui nous ont fourni les outils nécessaires à la réussite de nos études
Universitaires pendant mon cursus.

Je remercie nos encadreur Zaoui Fares ,

Qui est toujours montré leurs intérêts et leur suivi de ce travail et qui
Ont été disponibles tout au long de nos travail, sans oublier leurs remarques
Pertinentes et leurs conseils et patiences durant l'étude de ce sujet pour aboutir à
un travail de qualité.

Enfin je tiens à exprimer toute nos gratitudes à l'ensemble de nos familles et
Proches pour leur soutien tout au long de ces années

Dédicace

*à mes parents,
à mon frère et
ma sœur,
à toute ma famille,
à mes amis,
je dédie ce mémoire.*

Ghebouli Yassine

*à mes parents,
à mon frère,
à mes Sœurs
à toute ma famille,
à mes amis,
je dédie ce mémoire.*

Benyahia Samy

Table des matières

Table des matières	i
Table des Figures	iii
Symboles & Abréviations	v
Introduction Générale	1
1 Généralités sur les Robots & les Bras Manipulateur	
1.1 Introduction.....	3
1.2 Historique.....	4
1.3 Les systèmes Mécaniques Robotisé.....	4
1.4 Les Types de Robot.....	6
1.4.1 Les Robots Mobiles	6
1.4.2 Les Robots Manipulateurs	7
1.4.2.1 Compositions	8
1.4.2.2 Catégorisation des Bras Manipulateur	10
1.5 Conclusion	13
2 Matériel	
2.1 Introduction	14
2.2 Partie Mécanique	15
2.2.1 Alucobonde	15
2.2.2 Roulement	16
2.2.3 Goujons & Bague en acier.....	16
2.2.4 Pince	17
2.2.5 Pignon	18
2.3 Partie Electrique	19
2.3.1 Carte de Commande ARDUINO	19
2.3.2 La Carte PCA 9685	27
2.3.3 Carte alimentation stups up	29
2.3.4 Servomoteur	30
2.3.5 Alimentation 5V-10A	32
2.4 Partie Software	33
2.4.1 Logiciel IDE ARDUINO	33
2.5 Conclusion	35
3 Modélisation& Réalisation	
3.1 Introduction.....	36
3.2 Cotémécanique.....	36
3.2.1 Conception mécanique	36

3.2.2 Fabrication de la structure mécanique du bras	37
3.3 Modélisation du Bras	
Manipulateur.....	41
3.3.1 Convention de Dénavit Hardenberg (DH 1955)	41
3.3.2 Géométrie.....	44
3.3.3 Model Géométrique Direct	45
3.3.4 Model Géométrique Inverse	46
3.4 Conclusion	52
Conclusion Générale	53
Bibliographie	55
Annexe	57

Table des Figures

1.1	Chaine de production robotise	4
1.2	Robot humanoïde	5
1.3	Fauteuil équipé d'un bras robot	5
1.4	Main articulée robot	5
1.5	Les robots mobiles	6
1.6	Bras manipulateur en série vs en parallèle	7
1.7	Structure d'un bras manipulateur	8
1.8	Représentation d'une articulation rotoïde	9
1.9	Représentation d'une articulation prismatique	9
1.10	Un bras manipulateur manuelle	10
1.11	Un robot automatique	10
1.12	Manipulation cartésienne	11
1.13	Manipulation cylindrique	11
1.14	Manipulation sphérique	12
2.1	alucobonde	15
2.2	roulement	16
2.3	gougeons.....	17
2.4	bague en acier	17
2.5	pince en aluminium	17
2.6	pignon imprimer en 3D	18
2.7	synthèse de caractéristique	23
2.8	Composition de carte Arduino UNO	24
2.9	microcontrôleur Atmega328P	24
2.10	Alimentation de la carte arduino	26
2.11	carte pca 9685.....	28
2.12	détailler sur les port de pca	28

2.13	connexions entre pca et module Arduino uno	29
2.14	carte alimentation stups up	29
2.15	servomoteur détailler	30
2.16	impulsion et signal de contrôle d'un servomoteur	32
2.17	Alimentation 5V 10A	33
2.18	interface du logiciel Arduino	33
3.1	emplacement des deux pignons	38
3.2	représentent la base du robot	39
3.3	support de bras manipulateur	39
3.4	premier segment	40
3.5	2eme segment	40
3.6	système de coordonnée et paramètre de DH	41
3.7	représentation des paramètre DH	42
3.8	un point d'intersection	43
3.9	exemple de ce genre de situation	43
3.10	simulation avec matlab	46
3.11	Prototype réalisé ABB IRB-460	47
3.12	vue haut de bras manipulateur	49
3.13	vue en Face de bras manipulateur	49
3.14	vue en Face de bras manipulateur par tone	50
3.15	Tringle explicateur	50

Symboles & Abréviations

d_i	Représente la distance entre les deux origines O_{i-1} et O_i le long d'axe Z_{i-1} .
θ_i	Représente l'angle entre les deux axes X_{i-1} et X_i autour de l'axe Z_{i-1} .
a_i	Représente la distance entre les deux origines O_{i-1} et O_i le long de l'axe X_i .
α_i	Représente l'angle entre les deux axes Z_{i-1} et Z_i autour de l'axe X_i .
θ_1	Angles entre l'axes des X et l'axes des Y
θ_2	Angles entre L_2 et r_1
θ_3	Angles extérieurs entre L_2 et L_3
ϕ_1	Angles entre r_3 et r_1
ϕ_2	Angles entre L_2 et r_3
ϕ_3	Angles entre L_2 et L_3
r_1	Représente la distance entre les origines de θ_2 et la projection de point de Outil terminale sur l'axe des X
r_2	Représente la distance entre les points de outils terminale et la projection sur r_1
r_3	Représente l'hypoténuse de triangle r_1, r_2, r_3
L_1	Représente la distance entre la base et le départ du segment 2
L_2	Représente longueur du segment 1
L_3	Représente longueur du segment 2
DDL	degré de liberté
IDE	intégrates développement environnement
DH	paramètre Denavit-Hartenberg
RRRR	Bra manipulateur à 4 rotations (rotoïde rotoïde rotoïde rotoïde)
MGD	Model Géométrique direct
MGI	Model Géométrique Inverse

Introduction Générale

Le robot manipulateur qu'on connaît tous est devenu un objet de travail indispensable dans différents secteurs qui touchent essentiellement le côté activité et industrie de l'homme, et qui ne cesse d'augmenter année après année, il passe par l'industrie manufacturière à l'agriculture, le nucléaire, médecine, industrie mécanique et montage allant jusqu'à la conquête de l'espace, cette présence souvent très remarquable du robot manipulateur dans tous ces domaines peut être justifiée par sa remarquable précision et rapidité d'exécution, son agilité et le plus important sa légèreté à faire une tâche considérée comme tâche fatigante et dangereuse pour un homme.

Dernièrement l'utilisation des machines dans le côté recherche et industrie en tout genre est devenue de plus en plus fréquente allant jusqu'à dire indispensable. En effet, l'automatisation des systèmes permet un nouveau point de vue et une conception plus sûre et plus rapide de ce qui est des résultats espérés.

On remarque que dans le domaine de la recherche l'utilisation de la robotique est en évolution remarquable, ce dernier permet de manipuler avec beaucoup de clarté et donc de manière plus sûre dans une variation d'objet ou produit, neutre ou dangereux. Comme elle permet aussi d'exécuter des tâches répétitives ou pénibles pour l'homme, les chercheurs en robotique sont toujours passionnés par un sujet qui est la manipulation d'objet.

On nomme les robots qui assure ce genre de tâches les robots manipulateurs, on les trouve dans les chaînes de production, construction urbanisme, l'industrie automobile, industrie électromécanique et aussi dans les milieux hostiles ; hautes et basses températures, fortes et faible radiation, bases et hautes performances...etc.

Actuellement, il existe une large variété de robots chacun conçu pour une tâche bien définie dans un domaine bien défini.

Dans ce travail de mémoire, nous nous sommes intéressés au bras manipulateur ABB à 4 degrés de liberté (4ddl), Notre principal objectif dans notre travail est de faire une conception, étude et une réalisation pour commander et contrôler un prototype bras manipulateur de quatre degrés de liberté commandé par ARDUINO.

Notre mémoire est divisée en trois chapitres fondamentaux que nous avons résumés comme suit :

Dans le chapitre 1, nous aborderons les robots et les bras Manipulateurs de façon globale dans de multiples domaines, plus précisément robots et bras robotisés Industriel ; leur collaboration dans l'industrie et sa répercussion sur la production.

Le chapitre 2 couvre la modélisation de façon globale (Partie 1) et la simulation avec l'Utilisation du logiciel Matlab sur le bras manipulateur ABB (Partie 2) d'une manière spécifique et on termine par le cote matérielle de ce dernier.

Le chapitre 3 est l'application de l'angle de 3 degrés de liberté du prototype du bras manipulateur ABB IRB 460 contrôlés par le module Arduino.

Généralités sur les Robots & les Bras Manipulateur

1.1 Introduction

A la base les robots ont été élaboré pour remplacer le travail de l'homme, qui souffre des difficultés industrielles au paravent, on croyait que les robots seraient des machines universelles capables de la reprogrammation rapide pour une grande variété de tâches, et ces idées qui ont motivé leur développement.

La robotique est un regroupement de matière spécialement techniques (mécanique, électronique, automatique, informatique) rassembler autour d'une bute et d'un objet commun. Tout se si pour simplifier facilite et sur tout automatise l'ensemble des taches sur de nombreux domaine des activités de l'homme qui ont pour réputation de ne jamais pouvoir être sépare de la présence humaine.

Allons jusqu'à dire ce qui était un rêve se rapproche à très grand pas de la réalité dans tous les secteurs possibles car actuellement, les robots sont très courants dans les domaines industriels, particulièrement dans le secteur automobile de façon générale et aussi dans la plupart des fabricants d'ordinateurs, par rapport à leur vitesse d'exécution qui ne cessent d'évoluer aussi souvent utiliser dans les chaines de montage pour des raison répétitifs [Zeg94].

1.2 Historique

Apparition du mot robot en 1920 un mot originel Tchèque dans le quelle son prédécesseur (robota) est la signification du travail force, apporter en 1920 par l'écrivain tchèque Karel Capek dans la pièce de théâtre Rossum's Universal Robots.

Le premier robot industriel en 1961 déployer pour la première fois dans le secteur nucléaire qui est aussi un descendant direct des télémanipulateurs, apparus pour la 1ere fois par une entreprise américaine unimation (DSU), fonder par George Devol et Joseph Engelberg. Exploiter pour la première fois sur les lignes d'assemblage de général Motors. Un robot qui pèse 1.5tonnes était apte de manipuler des pièces de force allant jusqu'à 150kilogramme.

La 1ere chaine de production robotisé des recherches on trouver que 2142 robots ont été conçu entre 1960 et 2010, la maison Nissan ouvre la première chaine de production complètement robotisé.

Aujourd'hui on estime avoir entre 1 et 1.3 million de robots travaillant pour Lhomme dans plusieurs industries dans le monde. [Djo12]



FIGURE 1.1 – chaine de production robotise [Djo12]

1.3 Les systèmes Mécaniques Robotisé

Les systèmes robotiques intéressent de très nombreux domaines civils et militaires. Les Grands champs d'application de la robotique sont :

- . Industrie (usinage, assemblage, soudure, polissage, formage, etc.)
- . Milieu alarmant (sous-marin, nucléaire, exploration, planétaire, etc.)
- . Domaines des transport (automobile tesla , etc.)

- . Aide et appui pour personne en difficulté (robots personnels, etc.)
- . Domaine médicale (chirurgie, rééducation, etc.).
- . Le ludique (robots jouets) Ils prennent des formes très diverses du point de vue de leur structure mécanique et de leur Commande. Plusieurs types de robots ‘génériques’ sont illustrés sur les figures qui suivent. [Djo12]



FIGURE 1.2– robot humanoïde [01]



FIGURE 1.3– fauteuil équipé d’un bras robot [01]



FIGURE 1.4– main articulée robot [01]

1.4 Les Types de Robots

On peut concevoir deux types de robots (famille) :

1.4.1 Les Robots Mobiles

Généralement un robot mobile est équipé de moyens de transport qui lui facilite le déplacement et le mouvement. Aussi équipée de moyens de connaissance des lieux et avec un raisonnement automatique. [Zeg94]

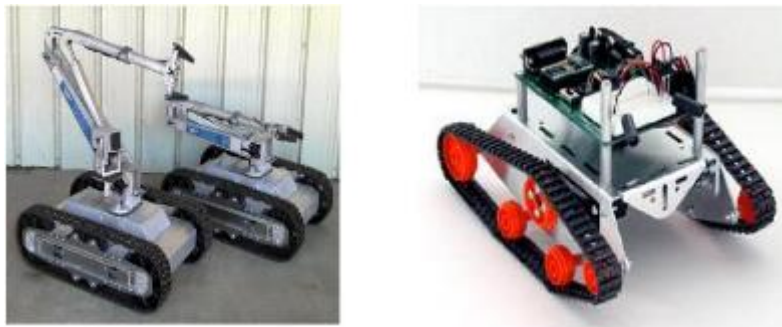


FIGURE 1.5 – les robots mobiles [01]

➤ Classification d'un robot mobile

Elle s'établit en suivant plusieurs critères :

- Degré de liberté
- Système locomotion
- Energie utilisée
- Moyens d'acquisition
- Vitesse de traitement de l'information
- Véhicule télécommander par un opérateur qui lui impose chaque tâche à réaliser
- Véhicule télécommander au sens de la tâche à réaliser.
- Véhicule semi-autonome qui réalise des opérations prédéfinies

On perçoit aussi d'énormes difficultés à la construction de ces robots comme on peut citer :

1. la mobilité mécanique
2. l'orientation et le positionnement
3. détermination des étapes liées au chemin

➤ Domaine d'application

Un large domaine où les robots peuvent être utilisés, nous présentons quelques

domaines :

- Industrie nucléaire : manipulation des matériaux radioactifs

Sécurité civile :

- ✓ Détection de mine explosive
- ✓ Neutralisation d'activité terroriste
- ✓ détection d'infraction

- Chimique : manipulation de matériaux toxiques
- Mine : assistance d'urgence
- Agricole : cueillage des fruits et légumes.
- Nettoyage : nettoyage industriel.
- Industrie : surveillance
- Militaire : pose d'explosifs



• FIGURE 1.6 – bras manipulateur en série vs en parallèle [01]

1.4.2 Les Robots Manipulateurs

Il est composé de plusieurs degrés de liberté automatiquement reprogrammable polyvalent qui peut être aussi fixe ou mobile selon le besoin humain (soudage, assemblage, transfert d'objet) il permet de déplacer un objet avec son outil terminal du point a au point b dans l'espace de travail appelé (volume de travail).

Le bras effectue ces tâches grâce aux ordres donnés par la personne qui sont transmis comme informations délivrées par des capteurs qu'on peut citer comme extéroceptifs.

Le travail se fait avec l'organe terminal de ce dernier, c'est une pratique qui devient de plus en plus courante dans la vie quotidienne. [Eda01]



FIGURE 1.7 – structure d'un bras manipulateur [Eda01]

1.4.2.1 Compositions

- **la base** : la base est fixée sur le dessous du bras manipulateur (lieu du travail), C'est un assemblage de matière souples ou rigide lies pas des articulations, son travail consiste à Supporter le poids de notre robot et établir un équilibre commun et enfin a facilité le déplacement de l'outil terminal d'un point A eu un point B.

On peut le considérer comme l'éléments le plus important et la partie la plus difficile et qui contiens le qui contient le plus de précision possible pour ce qui est de la partie mécanique de notre robot.

- **Le porteur** : Le porteur représente la partie primordiale du système mécanique articulé (segment, articulation, actionneur, organe terminal), il a pour rôle de conduire l'organe terminal dans une position imposée, Il est constitué de :

a. **Segment** : c'est un genre de relais entre la bases et l'outil terminal qui comporte aussi des propriétés spéciales comme la forme, la taille, la matière, la rigidité

b. **Articulation** : Une articulation liés deux corps successifs en limitant le nombre de degrés de liberté, de l'un par rapport à l'autre

-**articulation rotoïde** : c'est un genre de pivot qui permet de baisser le mouvement entre deux corps est donnée par l'angle autour. [Eda01]

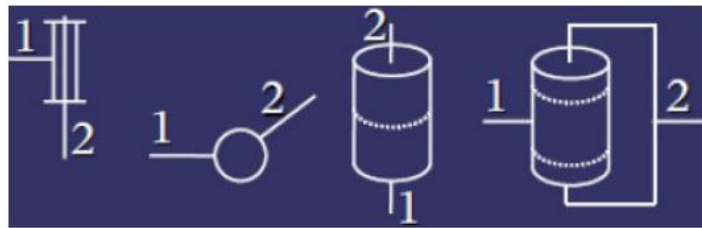


FIGURE 1.8 – représentation d'une articulation rotoïde [Ach07]

-**articulation prismatique** : c'est en quelque sorte une articulation de type glissière, notée P, diminuant le mouvement entre deux corps à une translation le long d'un axe commun, la situation corrélative entre les deux corps est mesurée par la distance le long de cet axe.

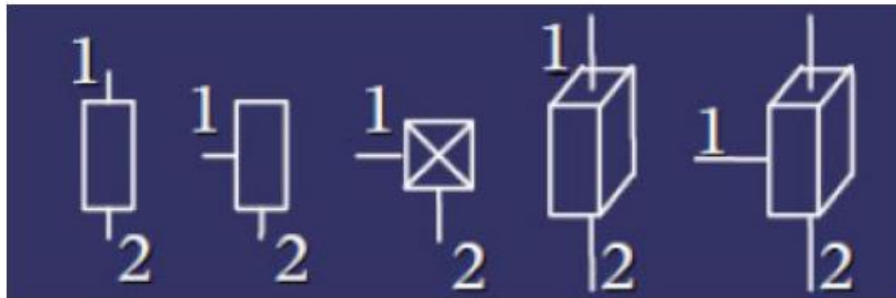


FIGURE 1.9 – représentation d'une articulation prismatique [Ach07]

-**organe terminal (outil terminal)** : c'est un dispositif mécanique utilise pour le serrage (pinces a deux doit), dispositifs magnétique (attraction et répulsion), dispositif avec pression de Lair (aspirateur, ventouse), outil de transformation (découpeuse, soudage peinture et autre).

On d'autre termes c'est la partie opératif finale du robot qui agit directement avec l'environnement extérieur.

1.4.2.2 Catégorisation des Bras Manipulateur

Tout dépend du mode de fonctionnement du robot et tâches à accomplir

-**Classification fonctionnelle** : ils varient d'un pays a un autre (4en France,6au japon)



FIGURE 1.10–un bras manipulateur manuelle [02]

-**manipulateur automatique** :



FIGURE 1.11–un robot automatique [02]

- classification géométrique** : on a aussi un classement sur structure géométrique
- robot cartésiens comporte 3 articulations prismatiques
- structure cartésienne qui nous permet d'avoir une bonne consistance mécanique avec une forte précision. [Ach07]
- utilisation typique** : manutention et assemblage.
- Actionneurs** : majoritairement électrique, quelque fois pneumatique

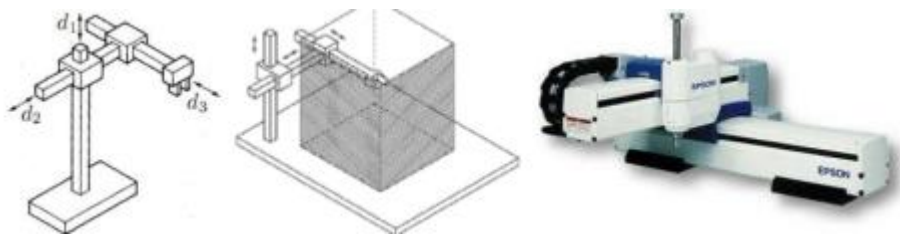


FIGURE 1.12– manipulation cartésienne[Tay18]

-robot cylindrique :

- il existe une différence être la géométrie cylindrique et cartésienne ou la partie prismatique est remplacé par une articulation rotoïde ;3DDL
- la structure cylindrique assure une solidité mécanique.
- mode d'utilisation : déplacement de produit de forte taille.

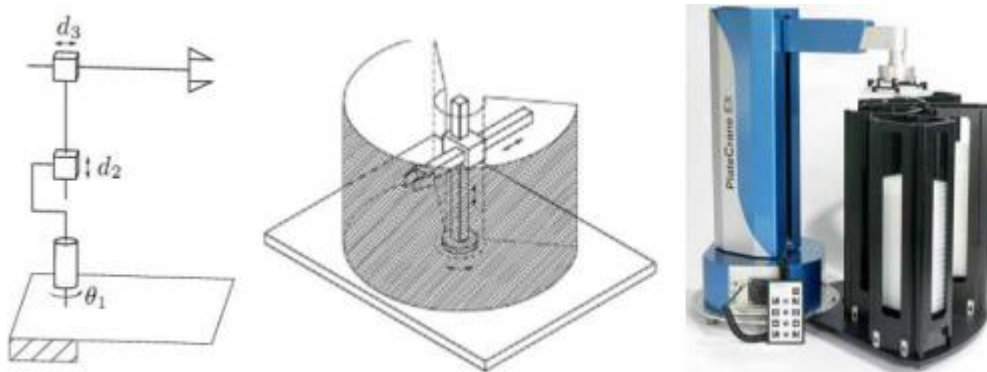


FIGURE 1.13– manipulation cylindrique[Tay18]

-robot sphérique :

- une autre différence pour ce qui est du sphérique et cylindrique dans la 2eme articulation prismatique qui est remplacé par une rotoïde ,3DDL
- moins de solidité mécanique aux robots précédent.
- environnement d'utilisation : usinage

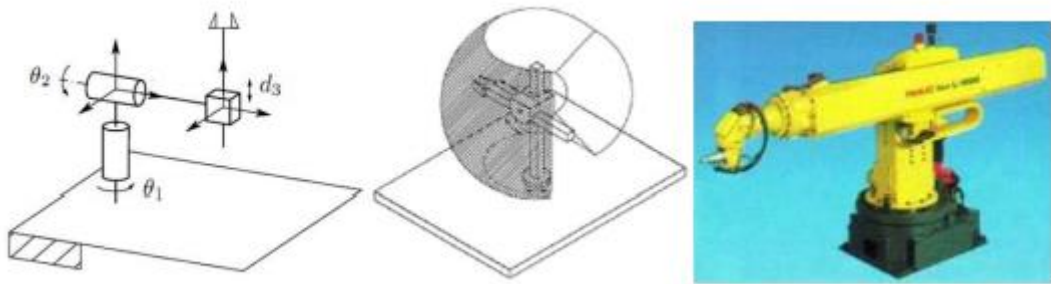


FIGURE 1.14– manipulation sphérique [Jag 06]

Parmi d'autre caractéristique on a :

- La charge utile transportable par le robot.
- L'espace de travail : c'est l'ensemble des situations de l'espace que l'organe terminal du robot peut atteindre.
- Les vitesses et les accélérations maximales, qui conditionnent le temps du cycle.
- Les performances (l'écart entre une situation commandée et la moyenne des Situations atteintes. [Jag 06]

1.5 Conclusion

Dans ce chapitre, nous avons présenté vague idée sur la robotique et une idée générale sur les robots et ainsi les bras manipulateur, l'historique des robots, leurs structures, leurs utilisations et les différents types de robots ainsi que leurs classifications, leurs domaines d'applications et Caractéristiques d'un robot manipulateur et ce qui va nous appuyer pour la construction de notre bras manipulateur qui va respecter les classifications et les définitions donner dans ce chapitre.

Puis dans le chapitre suivant on s'intéressera à on va aborder le matérielle utiliser dans notre projet.

Matériel

2.1 Introduction

Avant d'entreprendre le troisième chapitre qui est la base de notre projet, nous allons montrer au cours de ce 2eme chapitre un ensablement d'éléments primordiale à la fabrication de notre bras manipulateur.

Tout d'abord on fait la conception de ce prototype en 2D / 3D à l'aide d'un programme (Autocade, Onshape)

Cet assemblage est constitué de 3 partie essentielle :

Partie mécanique : (matière utilise pour notre prototype (La lecobonde), roulement (plusieurs dimensions), goujon et bague en acier usiné chez un tourneur, pince (outil terminal), écrous et boulon, rondelle, Visse, pignons (à base de plastique), socle on boit)

Parties électriques : (CARTE DE COMMANDE ARDUINO UNO, PCA9685, ALIMENTATION 5V -10A, STEP UP (5V jusqu'à 24V), SERVOMOTEUR)

Partie software : (Programmation de la carte Arduino uno, avec le logiciel (IDE), on Shape / Autocad, Matlab).

2.2 Partie Mécanique

2.2.1 Alucobonde

Un panneau composite constitué de deux tôles d'aluminium avec un noyau minéral. Les caractéristiques excellentes du matériau soutiennent l'inspiration et permettent des solutions avancer dans tous les domaines de l'architecture et notamment la fabrication de tout genre, que ce soit la construction artistique traditionnelle de l'habitat privé.

La cause de ce choix : malgré la rigidité et la solidité du produit il reste souple et léger, simple à découper, modelé et trouser et pour finir il donne un joli aspect esthétique. [03]

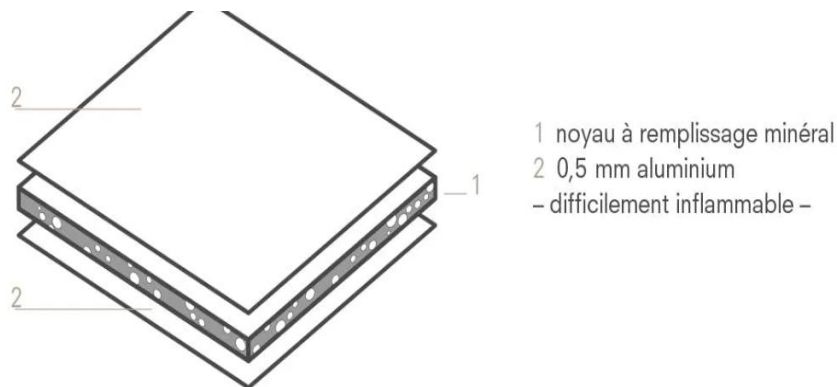


FIGURE 2.1 – alucobonde [03]

2.2.2 Roulement

Le roulement est purement mécanique c'est un mécanisme destiné à guider un assemblage en rotation, c'est-à-dire à permettre à une pièce de tourner par rapport à une autre selon un axe de rotation bien défini.

Il permet aussi de supporter des efforts importante résiste aux efforts radiaux et à l'effort axiaux, résiste à l'environnement, accepter un débattement angulaire.

Type de roulement utilise : 2 roulements 32mm réf :6002

4 roulements 20mm réf : 5433

8 roulements 15mm réf :6102

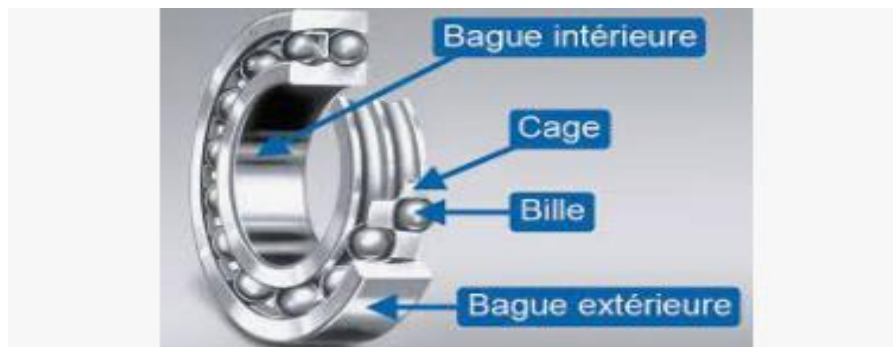


FIGURE 2.2 – roulement

2.2.3 Goujons & Bague en acier

Les 2 Principaux éléments qui assurent la fixation, la stabilisation et le parallélisme des 2 partie (partie fixe et parti mobile)

Goujons : usiner sur mesure chez le tourneur avec une longueur de 7,5cm et un diamètre de 20mm Avec 1cm de chaque cote pour les croups et 1cm pour la superposition de notre goujon sur les roulements axiaux.

Bague en acier : usiner aussi chez le tourneur sur mesure le diamètre est de 35mm de tel manière quelle sois superpose sur l'extrémité des deux roulements axiles sans pour autant toucher notre goujon.

Distance entre l'extrémité du goujon et la paroi de notre bague et de 10mm

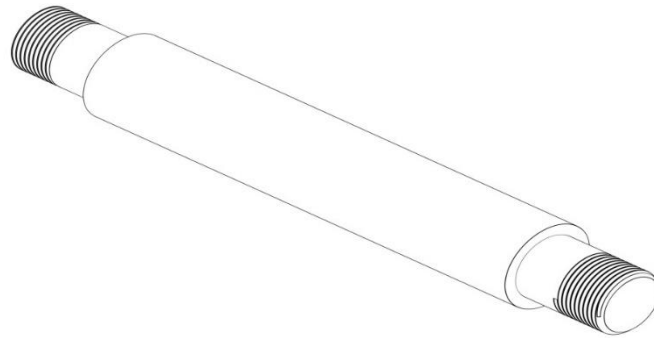


FIGURE 2.3 – goujon



FIGURE 2.4 – bague en acier

2.2.4 Pince

Appeler outil terminale ou organe terminale, c'est le dernier composant d notre bras manipulateur qui a comme bute d'attraper l'objet et de l'accrocher, et sur lequel le dernier servomoteur se base.



FIGURE 2.5– pince en alluminium

2.2.5 Pignon

C'est un système mécanique composé d'au moins deux roues d'entrée engrenée, pour une facile transmission de mouvement de rotation ou bien pour une propulsion.

Dans notre cas on a besoin d'une transmission de mouvement et pour assurer cela on a besoin de deux roues d'entrée engrenée qui doivent être en contact physique lune avec l'autre.

Engrenée veut dire des roues qui possèdent des dents pour que chaque dent, pour que chaque roue puisse s'accrocher à l'autre roue.

Pour notre cas on a deux pignons un grand de 120mm et un petit de 60mm faites sur mesure avec une imprimante 3D Matière utilisée plastique.

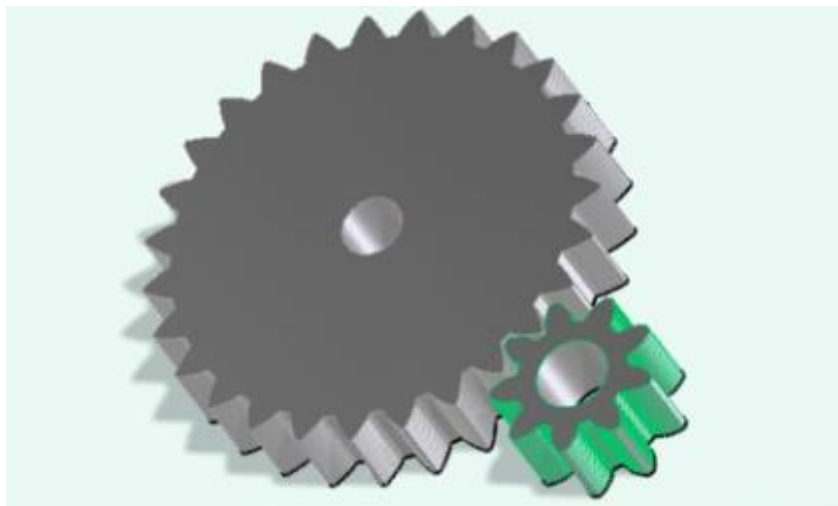


FIGURE 2.6– pignon imprimé en 3D

2.3 Partie Electrique

2.3.1 Carte de Commande ARDUINO

Arduino est un projet fonder par une équipe de développeurs, qui se compose de six personne. Cette équipe a créé le « système Arduino ». L'Arduino est une infime carte électronique, équipé d'un microcontrôleur non pas d'un microprocesseur comme on habitude de voir.

Ce n'est donc pas un ordinateur, et elle ne possède donc pas de système d'exploitation mais elle permet de connecter un bon nombre de composants (LEDs, moteurs, servomoteurs, capteurs, écran LCD, Wifi, Bluetooth...) et de les faire fonctionner de manière autonome en la programmant via son logiciel Arduino IDE (ou d'autre logiciel comme Scratch, Arduiblock, ou encore Processing...) avec un langage de programmation assez simple (dérivé du C++) idéal donc pour réaliser des projets de Domotiques, de Robotiques ou d'Informatique Embarqué.

On parle de carte de prototypage, car ce petit bijou permet de créer des prototypes très simple. En effet, elle comporte des broches appelées PIN qui accorde un facile branchement très diversifier par rapport aux composants afin de tester les branchements et les programmes, une fois qu'on a acquiert ce que on veut, il ne reste plus qu'à joindre les composants pour obtenir un prototype. [Erk 14]

Principe de fonctionnement : on a des étapes très précise à suivre :

- On conçoit ou on ouvre un programme existant avec le logiciel Arduino.
- on insert les bibliothèque don on a besoin
- On vérifie ce programme avec le logiciel Arduino (compilation).
- Si des erreurs sont signalées, on modifie le programme et on ajuste les erreurs.
- On charge le programme sur la carte.
- On câble le montage électronique.
- L'exécution de programme est automatique après quelques secondes.
- On alimente la carte soit par le port USB, soit par une source d'alimentation Autonome (5V -10A).

Les familles de carte Arduino :

On trouve généralement 3 cartes Arduino selon le mode d'emploi et mode de fabrication :

- les appeler « officielles » qui sont fabriquées en Italie par le fabricant officiel : Smart Project
- Les appeler « compatibles » qui n'est pas fabriqués par Smart Project, mais qui sont totalement.
- compatible Les « autres » fabriquées par multiple entreprise et commercialisées sous un nom différent (Freeduino, Seeduino, Femtoduino, ...). Es avec les Arduino officielles.

Les types de carte Arduino : en ce moment, il existe plus de 20 versions de module Arduino, nous allons citer quelques modelés afin d'élucide l'évaluation de ce produit scientifique et académique :

- Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un Atmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur Atmega168.
- L'Arduino Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur Atmega168 (Atmega328 pour une plus nouvelle version).
- Le Lily Pad Arduino, une conception de minimaliste pour l'application wear able en utilisant un microcontrôleur Atmega168.
- Le NG d'Arduino plus, avec une interface d'USB pour programmer et usage d'un Atmega168.
- L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur Atmega168.
- L'Arduino Diecimila, avec une interface d'USB et utilise un microcontrôleur Atmega168.
- L'Arduino Duemilanove (« 2009 »), en utilisant un microcontrôleur l'Atmega168 (Atmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- L'Arduino Méga, en utilisant un microcontrôleur Atmega1280 pour I/O additionnel et mémoire. L'Arduino UNO, utilisations microcontrôleur Atmega328.
- L'Arduino Mega2560, utilisations un microcontrôleur Atmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel Atmega8U2 (Atmega16U2 dans le jeu de puces d'USB de révision 3).

-L'Arduino Leonardo, avec un morceau Atmega3U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.

-L'Arduino Esplora : ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération.

Parmi tous ces types qui diffère, nous avons opté une carte Arduino UNO (carte Basique), L'avantage principal de cette carte est aidé et faciliter l'exécution d'une telle commande qui sera détaillée par la suite. [Erk 14]

Applications :

Plate-forme logicielle et matérielle de création d'objets numériques, Arduino permission et facilite de commander des circuits électroniques qui ont un contacte directe avec le milieu qui les entoure.

Connectés particulièrement à des capteurs sonores, thermiques, de mouvement, ces circuits électroniques peu coûteux, dénommés microcontrôleurs, peuvent en retour générer des images, actionner un bras articulé, envoyer des messages sur Internet, etc. Des dizaines de milliers d'artistes, de designers, d'ingénieurs, de chercheurs, d'enseignants et même d'entreprises l'utilisent pour réaliser des projets dans de multiples domaines :

- prototypage rapide de projets innovants utilisant l'électronique, Arduino facilitant l'expérimentation en amont de la phase d'industrialisation.
- production artisanale d'objets numériques et de machines-outils à faible coût dans la perspective d'une culture d'appropriation technologique favorisant le bricolage et la débrouille.
- captation et analyse de données scientifiques (environnement, énergie, etc.) à des fins éducatives, de recherche ou d'appropriation citoyenne.
- spectacle vivant, grâce aux nombreuses fonctions d'interaction offertes par Arduino, il est possible de créer des performances de Vjing, utiliser le mouvement des danseurs pour générer en temps réel des effets sonores et visuels dans un spectacle.
- installations d'arts numériques, Arduino permettant de réaliser des œuvres d'art interagissant de manière autonome avec le public.

- Mode et design textile, plusieurs stylistes et designers investissant ce domaine créatif en exploitant les possibilités offertes par l'intégration de l'électronique notamment dans des vêtements (e-textile).
- projets pédagogiques à destination d'étudiants, de professionnels ou du grand public selon les porteurs de ces initiatives : écoles supérieures, centres de formation spécialisée ou des Media Labs. [Jen06]

Bonnes raisons de choisir Arduino :

En fait, il existe de plusieurs microcontrôleurs utilisables sur le marché tels que BX-24 Phidgets, Basic Stamp et Parallax de Netmedia Raspberry Pi et tous ont de puissantes adaptations et ont la capacité de contrôler multiples composants électroniques. Bien sûr, chacun a sa particularité avec une référence différente mais aussi hautement distingué l'Arduino est un regroupement d'objets qui font la différence entre lui et les autres le plus influent :

- Simplicité : la pièce Arduino est conçue pour répondre aux besoins de tous les pros, professeurs, étudiants et amateurs d'électronique interactive.
- Prix : L'Arduino est moins cher que d'autres panneaux du même type.
- Auto assemblage : on peut télécharger la fiche technique privée Arduino gratuitement à partir du site officiel et acheter les morceaux et l'installer nous-mêmes.
- Multiplateforme : Le programme Arduino a la capacité de fonctionner sous Windows, Mac OS, Linux et la plupart des autres commandes électroniques seulement sous Windows.
- Un environnement logiciel simple et convivial : l'environnement de programmation conçu pour être facile pour les débutants, fixes et puissants pros.
- Open Source :
 - Le logiciel Open Source est écrit en C ++ et il est de plus téléchargeable. Les programmeurs peuvent le modifier en fonction de leurs besoins.
 - le matériel Open Source est principalement constitué de microcontrôleurs Atmega8 et Atmega168 et les schémas sont publiés sous une licence Creative Commons, permettant aux concepteurs de circuits électroniques de concevoir leurs propres circuits.

La carte Arduino UNO :

Version de base du système basé sur un microcontrôleur Atmega328 cette carte délivre un maximum de 14 points d'entrée / sortie digitaux et de 6 entrées analogiques. Existe avec un processeur en version DIL sur support ou SMD soudée. Outre le processeur, la principale évolution consiste en un remplacement du chipset de l'interface de programmation USB de la puce d'interface USB / Série FTDI en une puce Atmel8U2 ou sur les dernières versions 16U2. Voici les principales caractéristiques de la carte Arduino Uno :

Arduino Uno	
	
Price Points	\$19.99-\$23.00
Dimension	2.7 in x 2.1 in
Processor	Atmega328P
Clock Speed	16MHz
Flash Memory (kB)	32
EEPROM (kB)	1
SRAM (kB)	2
Voltage Level	5V
Digital I/O Pins	14
Digital I/O with PWM Pins	6
Analog Pins	6
USB Connectivity	Standard A/B USB
Shield Compatibility	Yes
Ethernet/Wi-Fi/Bluetooth	No (a Shield/module can enable it)

FIGURE 2.7– synthèse de caractéristique [Erk 14]

Constitution de la carte Arduino UNO :

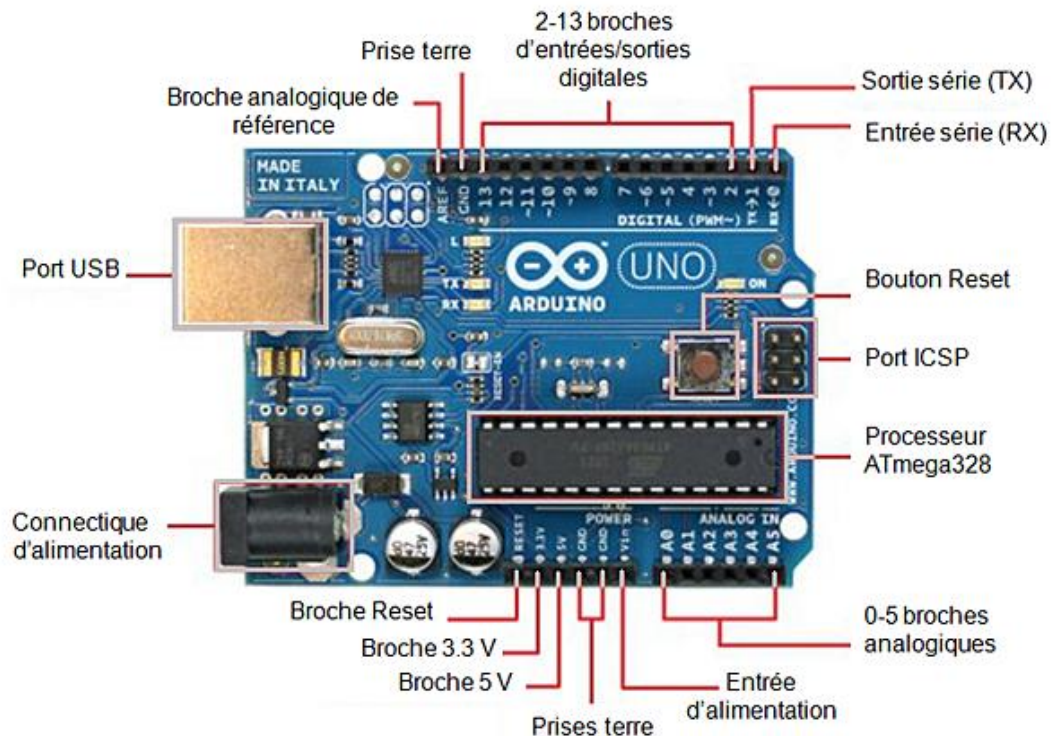


FIGURE 2.8– Composition de carte Arduino UNO [Erk 14]

Le Microcontrôleur ATmega328 :

Est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants, tels que les transistors, les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C.



FIGURE 2.9– microcontrôleur Atmega328P

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur.

Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- La mémoire Flash : C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont bootloader de 0.5 ko).
- RAM : c'est la mémoire dite « vive », elle va contenir les variables du programme. Elle est dite « volatile » car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- EEPROM : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme.

L'alimentation de la carte Arduino :

La carte Arduino Uno doit alimenter soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La carte Arduino sélectionne automatiquement le mode d'alimentation.

L'alimentation externe (non-USB) peut être ou bien attacher à un secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (batterie externe). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées GND (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être déstabilisé.

Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte (destruction de la carte). Aussi, la fourchette idéale recommandée pour alimenter la carte Uno est entre 5V et 7V.

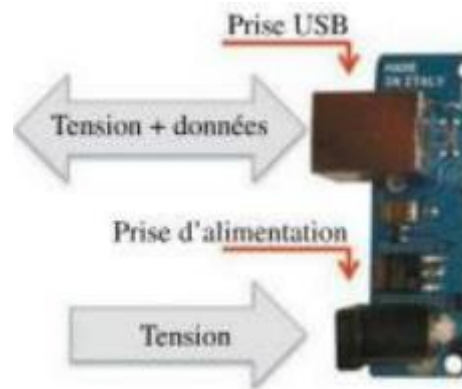


FIGURE 2.10– Alimentation de la carte arduino [Erk 14]

Entrées et sorties numériques :

Toute carte arduino uno contient 14 broches (numérotées des 0 à 13) qui sont utilisées comme entrée numérique ou sortie numérique comme on veut selon le besoin, en utilisant les instructions « `pinMode`, `digitalWrite` et `digitalRead` » du langage Arduino. Ces broches fonctionnent en 5V (4.8V).

Toute la broche quelque soit donne où 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)` ». De plus, on doit souligner que parmi ces broches on a celle qui procède des fonctions spécialisées :

-Communication Série : Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL.

Ces broches sont connectées aux broches correspondantes du circuit intégré Atmega8U programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

-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 détail

- 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 librairie 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 librairie «`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.

Les ports de communications :

La carte Arduino UNO possède plusieurs moyen et possibilités de communications avec l'externe :

- Communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX). On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré Atmega328.
- Programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur. Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC.

2.3.2 La Carte PCA 9685

Le module PCA9685 est un contrôleur 16 canaux qui permet de guider 16 sorties PWM par une communication I2C. Il autorise en d'autres termes la libération des entrées sorties de votre microcontrôleur et administre jusqu'à 16 LED ou servomoteurs (ou tout autre module prenant en entrée un signal PWM) à l'aide de deux broches (SCL et SDA) tout en conservant les broches de votre microcontrôleur pour d'autres modules comme des capteurs ou une alimentation. [07]

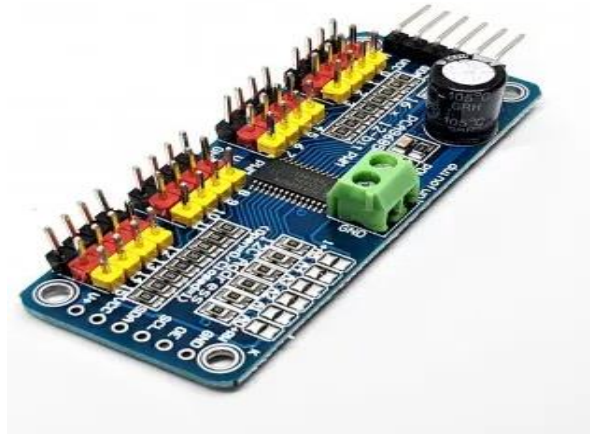


FIGURE 2.11– carte pca 9685 [04]

Principe de fonctionnement :

Le module est basé sur le contrôleur PCA9685 qui permet de contrôler des sorties PWM à l'aide de la communication I2C et d'une horloge intégrée. Ce module comporte 6 ponts permettant de sélectionner l'adresse de la carte et de se fait d'installer sur le même bus jusqu'à 62 contrôleurs pour un total de 992 servomoteurs (Adresses disponibles 0x40 à 0x7F), qui est aussi muni de bus I2C et d'un entrée de puissance, Il permet de contrôler des sorties PWM avec une fréquence équitable et ajustable et avec une résolution de 12 bits. Le module est compatible avec les microcontrôleurs 5V et 3.3V.

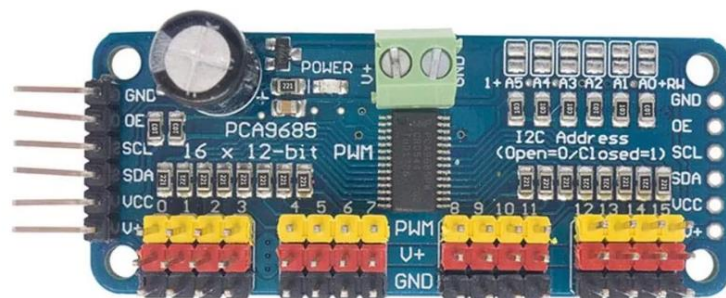


FIGURE 2.12– détailler sur les port de pca

Schéma :

Le module est muni d'un bus I2C et d'une entrée de puissance. Le bus I2C est branché comme suit :

- Broche A5 ou SCL à la broche SCL du module.
- Broche A4 ou SDA à la broche SDA du module.
- Broche 5V à la broche Vcc du module.
- Broche GND à la broche GND du module.

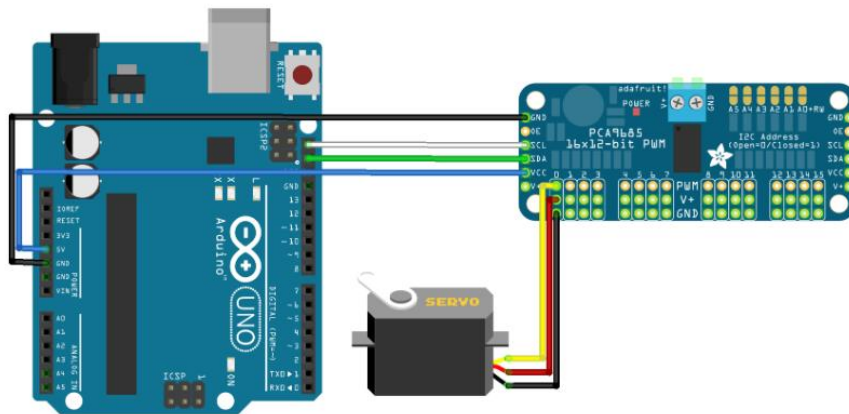


FIGURE 2.13– connexion entre pca et module Arduino uno [Lab18]

2.3.3 Carte alimentation steps up

C'est une carte électronique qui permet d'augmenter le voltage (tension) dans un intervalle de 5V jusqu'à 24V à l'aide d'une liaison directe (câble) entre l'alimentation et le steps up avec un petit inconvénients chaque augmentation de variation diminue le courant i output.



FIGURE 2.14– carte alimentation steps up

2.3.4 Servomoteur

Un servomoteur est un système peu compliquée qui a pour objectif de produire un mouvement précis en quelque sorte une rotation précise en réponse à une commande externe, C'est un actionneur (système produisant une action) qui intègre l'électronique, la mécanique et l'automatique.

Un servomoteur est qualifié pour attendre des positions prédéterminées avec les adresses qui lui on était donné, puis de les maintenir. Un servomoteur est composé de :

- Un micro moteur à courant continu.
- Un réducteur en sortie de ce moteur diminuant la vitesse et augmentant le couple.
- Un potentiomètre (faisant fonction de diviseur résistif) qui génère une tension variable, proportionnelle à l'angle de l'axe de sortie.
- Un axe dépassant hors du boîtier avec différents bras ou roues de fixation.
- Un dispositif électronique d'asservissement.

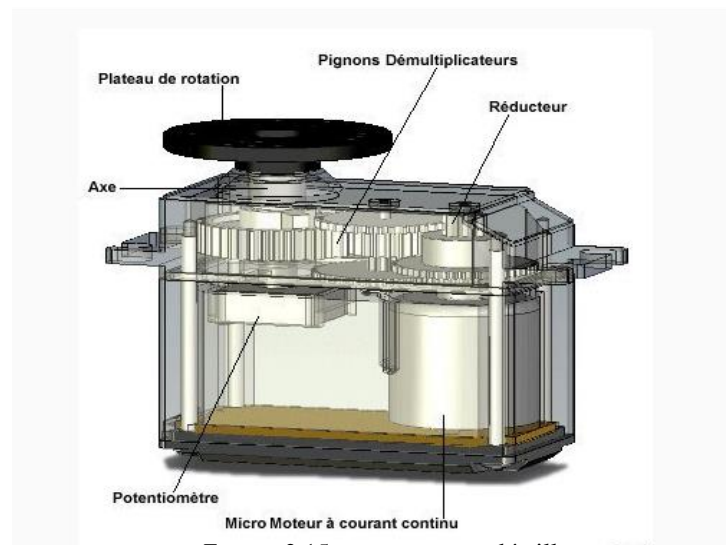


FIGURE 2.15– servomoteur détailler

Dans l'univers de Arduino, le servomoteur est utilisé que pour ce qui est des application et programmation robotique. Le servomoteur comporte trois fils de connexion pour pouvoir marcher, 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 [Mew18]

Fonctionnements des servomoteurs :

Le mode de fonctionnement d'un servomoteur marche comme ce si :

Un petit circuit électronique permet de contrôler un moteur à courant continu en fonction de la position d'un potentiomètre intégré au servomoteur. La sortie du moteur à courant continu est attaché mécaniquement à une série d'engrenage qui augmente la force (le couple) du servomoteur en minimisant la vitesse de rotation de celui-ci. Quand le moteur tourne, les engrenages s'animent, le bras bouge et entraîne avec lui le potentiomètre. Le circuit électronique ajuste constamment la vitesse du moteur pour que le potentiomètre (et par extension le bras) reste toujours au même endroit.

Le servomoteur à l'attribut d'être asservi en position angulaire, cela signifie que l'axe de sortie du servomoteur conservera la consigne d'instruction envoyée à son entrée, même si une difficulté ou un empêchement extérieur ou intérieur intervient pour faire changer l'orientation de sa trajectoire, le servomoteur cherchera de garder la position. Pour un ajustement précis de la position voulu, le moteur et son ajustement sont équipés d'un système de mesures qui détermine la position courante.

Il suffit de donner un signal a notre servomoteur ("reste à 90°" par exemple) et le servomoteur fera son MAXIMAL limite pour garder et préserver cette consigne (angle).

Cette consigne est transmise au moyen d'un signal numérique, d'une impulsion pour plus clairement. Pour que le servomoteur reste à une position donnée, il faut transmettre toutes les 20 millisecondes (soit à une fréquence de 50Hz) une impulsion d'une longueur comprise entre 1 et 2 millisecondes. [Lab18]

- Une impulsion de 1 ms correspond à un angle de 0°.
- Une impulsion de 2 ms correspond à un angle de 180°.
- Une impulsion de 1,5 ms correspond à un angle de 90.
- le fonctionnement d'un servomoteur dépend de son paramètre par rapport au poids et chaque poids dépend de la tension donner VOLTAGE, (exemple 20kg marche entre 4,8v-6v / 40kg marche entre 6v-7.4).

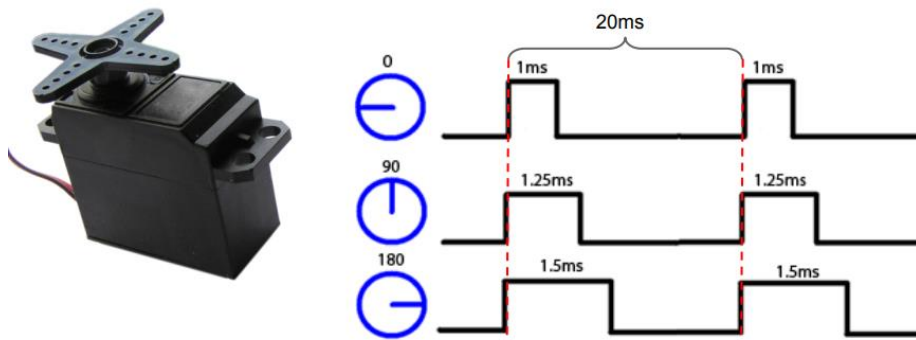


FIGURE 2.16– impulsion et signal de contrôle d'un servomoteur [Lab18]

Les avantages

- Aucune connaissance en programmation n'est nécessaire dans la plupart des cas.
- un projet avec des servomoteurs n'est probablement pas coûteux.
- il propose un simple projet dans lequel la précision peut ne pas être importante.
- il propose un projet qui n'oblige pas de rétroaction. - Robot de petite taille (les servomoteurs intelligents ne sont pas encore disponibles en micro format).
- quel que soit le modèle et la marque du servomoteur il reprend automatiquement au signal MLI.

Les inconvénients

- il assure une communication seulement dans 1 seul sens.
- il est infaisable d'avoir une rétroaction sur une position réel. Il n'est pas possible d'obtenir de rétroaction sur la position réelle.
- pour connecter un servomoteur avec un contrôleur sa demande beaucoup de câble.

2.3.5 Alimentation 5V-10A

Un ensemble de systèmes capables de fournir de l'électricité aux appareils (servomoteur) fonctionnant avec cette énergie.

Une alimentation électrique est, en particulier, un appareil ou appareillage plus ou moins complexe, destiné à fournir de l'électricité, à partir d'un réseau électrique. Cet appareil assurant une ou plusieurs fonctions grâce à l'électricité, mais celle-ci doit avoir des caractéristiques en courant, tension et, fréquence spécifique comme la nôtre (5V avec 10 A). Elle possède aussi un variateur intégré qui permet de diminuer ou augmenter la tension entre 4V ET 6V.



FIGURE 2.17– Alimentation 5V 10A

2.4 Partie Software

2.4.1 Logiciel IDE ARDUINO

L'IDE est un logiciel de programmation qui permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte. Il programme par code, contenant une cinquantaine de commandes différentes. Le langage de l'IDE Arduino est un mélange entre le C et le C++, il est très aisé en ce qui concerne les instructions. A l'ouverture, L'interface du logiciel Arduino se présente de la façon suivante :

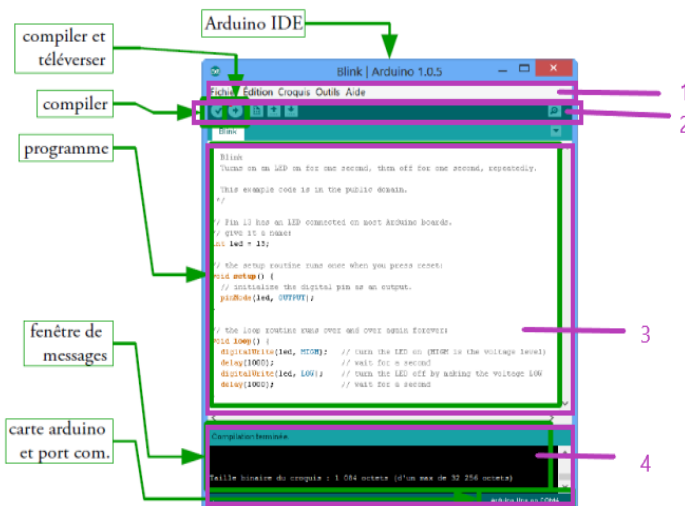


FIGURE 2.18– interface du logiciel Arduino [Djm16]

Dans cette figure on trouve 4 barre numérotée :

Barre 1 : **barre d'outils**

- créer un nouveau programme
- ouvrir programme existant
- enregistrer
- insertion bibliothèque

Barre 2 : **icone pour**

- vérifier
- téléviser
- sauvegarder / Ouvrir

Barre 3 : **écrire et modifier un programme**

Barre 4 : vérification de la compilation

- vérification de la compilation
- Affichage des erreurs et notifications. [Djm16]

2.5 Conclusion

Nous avons abordé dans ce chapitre les trois parties indispensable et les éléments fondamentaux qui font l'objet de notre projet, on a parlé en premier lieu de la partie mécanique qui résume le matériel nécessaire pour l'assemblage de notre robot.

On a entamé le côté électrique en deuxième lieu en présentant les servomoteurs et en définissant ses propriétés et ses différents modèles comme on a fourni les raisons de notre choix par rapport à leur propriété et au coût, et en dernier lieu on a évoqué le côté logiciel les différents programmes utilisés et principalement l'Arduino IDE, on a détaillé ce dernier en tenant compte de ses propriétés, son interface et ses options.

Le chapitre suivant sera consacré à la réalisation de notre bras manipulateur, à sa mise en fonctionnement et à son évaluation ainsi que sa programmation.

Modélisation & Réalisation

3.1 Introduction

Enfin c'est le dernier chapitre, nous allons parler des multiples étapes de réalisation de notre bras. L'assemblage débute par une structure mécanique possédant trois degrés de libertés, ainsi que le montage mécanique et électrique de la carte Arduino uno avec les servomoteurs. Sans oublier le côté modélisation qui contient le programme utilise fonctionnel, le modèle géométrique directe à partir des paramètres DH (Denavit et Hardenberg) à l'aide du programme Matlab en suite on calcule le modèle géométrique inverse (Matrice Jacobéenne).

Pour permettre le déplacement du bras manipulateur il faut assembler tous ces points et tous ces calculs par ordre précis, nous nous sommes basés sur le modèle géométrique inverse afin de donner des coordonnées (X Y Z) à notre programme ce qui nous permettra de le faire pivoter jusqu'au point déterminé. [Rg10]

3.2 Coté mécanique

3.2.1 Conception mécanique

On a fait 3 essais avec 3 prototypes les 2 premiers sont des échecs qu'on peut considérer comme un plus pour notre expérience dans ce projet ; on commence par le premier prototype qui est fait avec une matière qu'on appelle forex (à base de plastique) l'inconvénient avec cette matière le robot devient très fragile et se brise très rapidement, ce qui nous a amenés à choisir une autre matière qui soit plus rigide mais légère en contrepartie

Pour le 2eme prototype on a choisis une matière qui est la alucobonde intérieur plastique et extérieure en aluminium qu'on a décidé que c'est un échec par la suite par rapport à sa taille, pour finir on a décidé de faire un dernier prototype comme le 2eme mais on plus grands par rapport à la taille et par rapport à sa texture et sa rigidité.

En premier lieu ont découpé la feuille d'alucobonde avec une machine CNC bien évidemment avec le logiciel AUTOCAD /ONCHAPE avec des mesures bien précise, par rapport à la taille des roulements utilise et aussi par rapport au modèle de servomoteur et à son emplacement dans le prototype.

Impression 3D à base de plastique pour une conception de deux pignons obligatoires pour assurer le bon pivotement de notre robot et pour garantir et gagnée en couple par conséquence optimisé le couple et la puissance du servomoteur.

Conception du goujon et de la bague qui forme la partie essentielle et qui permet de tenir la base du moteur avec le coté inferieur en acier, pour éviter les débordements et le balancement sur les côtes de la base et cette du coté inferieur du robot car l'un et suspendu par rapport à l'autre.

3.2.2 Fabrication de la structure mécanique du bras

La structure mécanique du bras manipulateur a 3DDL est conformer de 3 éléments : la base qui est divisé en deux partie, bras qui est aussi former de deux axes (AX1 /AX2) avec un trois supports de même longueur pour fixer les 2 axe.

Premièrement on a la base qui est diviser on deux partie (deux bases) :

-La 1ere partie qui est la partie inferieure : elle est faite en bois d'un diamètre de 2cm qui porte un cercle d'alucobonde fixer avec des vices au centre de ce cercle se trouve un trou de 32mm, dans ce trou on a inséré un roulement de même diamètre à l'aide d'une presse (roulement réf : 6002) pour assurer la fixation.

Au même temps sur même partie on a placé un servomoteur qui est responsable de la rotation d'un quart de sphère du robot et qui détermine le plant du travaille avec une précision qui nous permet d'assembler et alignée les deux pignons sachons que le premier pignon de 6cm et attacher à ce dernier en tenant compte de la hauteur qui sépare la partie inferieur a la partie supérieure.

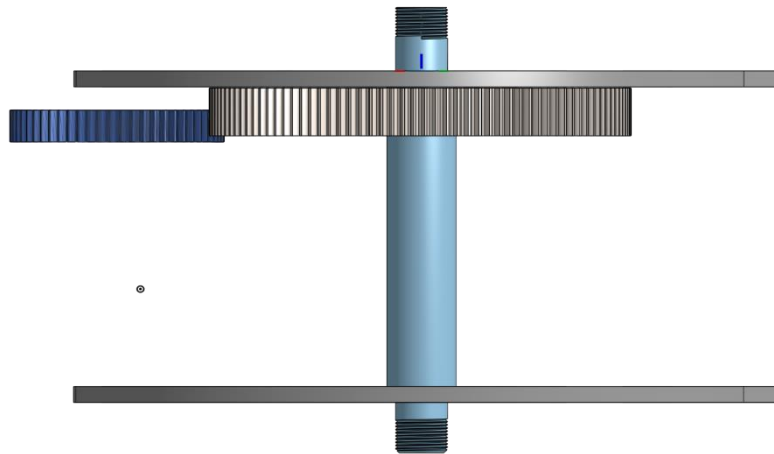


Figure 3.1– emplacement des deux pignons

-La 2ème partie de la base (partie supérieur) : de même façon, de même matière et de même diamètre que le cercle d'alucobonde de la partie inferieur il porte un 2eme pignon de 12cm de diamètre de telle façon et de telle précision à joindre le 1ere pignon pour assurer le contacte physique des deux.

Aussi nous trouvons dans le centre de cette partie un autre roulement de même diamètre (32mm) qui permet avec un goujon de 7,5cm le relai et l'assemblage les deux parties.

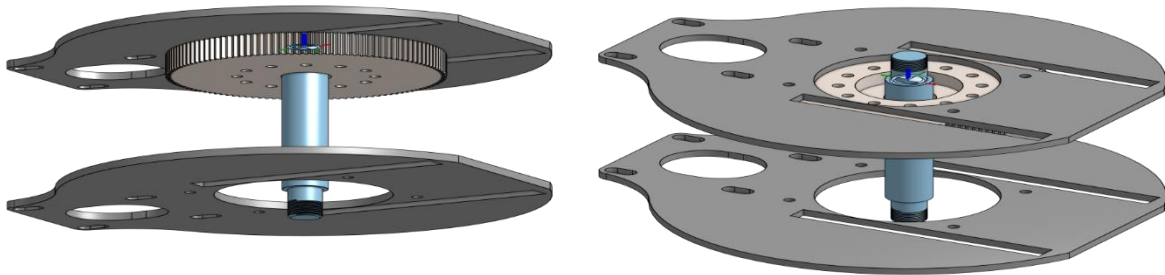


Figure 3.2– représentent la base du robot

Sans oublier Qu'il y'a 3 support qui repose sur cette partie, fixé avec des vices.

Deuxième Partie :

On trouve dans cette partie les supports et les AX, les 3 supports : 1 à gauche et 1 à droite qui permettent de porter les deux servomoteurs en parallèle.

Et ainsi pour porter les segments (AX), le dernier support qui se trouve au milieu permet de soutenir la structure générale du robot et qui comprends aussi 2 roulements de diamètre 12 mm (côté droit /milieu) pour faciliter les mouvements avec les segments.



Figure 3.3– support de bras manipulateur

Les segments : ont possédé deux segments

-le premier segment (30cm) attacher directement au servomoteur et qui est responsable du pivotement et de la trajectoire du bras manipulateur.



Figure 3.4– premier segment

-le deuxième segment (35cm) assure beaucoup plus l'altitude (axe z) du bras manipulateur et liée directement avec la pince (l'organe terminale).



Figure 3.5– 2eme segment

Pour finir on rajoute la pince estimer de 175grame et de 9cm de longueur qui est aussi l'outil terminale attache sur l'extrémité du 2eme segment sur une partie d'aluminium qui est aussi sur mesure pour assurer une bonne fixation et un bon accrochage.

3.3 Modélisation du Bras Manipulateur

3.3.1 Convention de Dénavit Hardenberg (DH 1955)

La convention DH consiste en un ensemble de règles qui permettent de simplifier l'analyse de la cinématique directe. L'avantage de cette méthode dévient apparent lorsque le nombre de liaison (prismatique et rotoïde) devient important. Ainsi l'application de cette méthode permet d'exprimer toutes les transformations homogènes de la chaine cinématique en termes d'un produit de 4 transformations de bras comme suit [Dh95] :

$$H_{i-1}^i = \text{Trans}_z(i-1)(d_i) * \text{Rot}_z(i-1)(\theta_i) * \text{Trans}_x(i)(a_i) * \text{Rot}_x(i)(\alpha_i) \dots \dots \dots (3.1)$$

$$H_i = \begin{array}{ccc|c} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \sin \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \alpha_i \sin \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \theta_i & d_i \\ \hline 0 & 0 & 0 & 1 \end{array}$$

$$\begin{array}{c|c} R & T \\ \hline 0 & 0 & 0 & 1 \end{array}$$

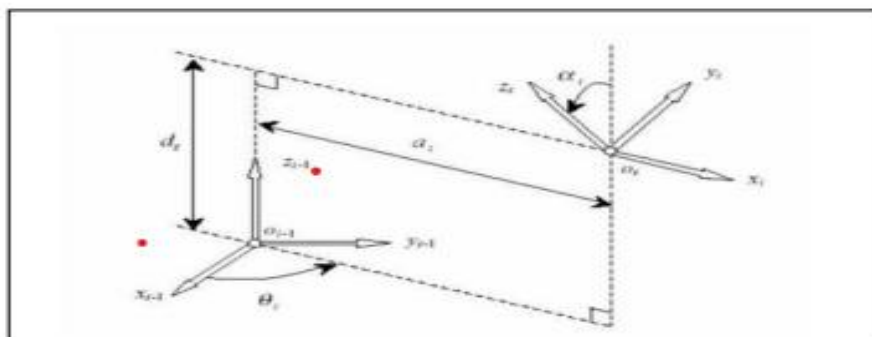


Figure 3.6– système de coordonnée et paramètre de DH [Dh95]

Les variables Θ_i , a_i , d_i , α_i , sont des paramètres associés au lien i et l'articulation $i-1$, Cette Matrice nous montre donc qu'avec cette méthode de 4 paramètres sont suffisants pour Caractériser une transformation homogène, alors qu'il en fallait 6(3angles et 3 coordonnées) Avec la représentation précédente.

Les conditions essentielles qu'il faudra respecter sont :

- 1- Les mouvements de l'articulation (Rotoïde ou prismatique) sont toujours sur axes Z_i
- 2- X_i doit être perpendiculaire à Z_{i-1}
- 3- X_i doit couper à l'axe Z_{i-1}

La figure suivante montre le cas général où les axes de mouvement (Z_{i-1} et Z_i) n'ont pas de point d'intersection, dans ce cas il n'y a qu'une seule et unique droite perpendiculaire aux Deux axes (normal commune).

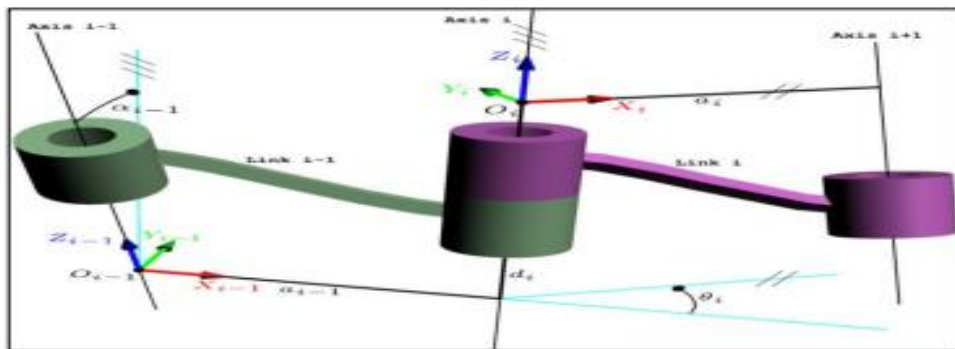


Figure 3.7– représentation des paramètres DH[Dh95]

Pour ce cas les paramètres de la convention DH représentent l'opération suivante :

- d_i représente la distance entre les deux origines O_{i-1} et O_i le long de l'axe Z_{i-1} .
- Θ_i représente l'angle entre les deux axes X_{i-1} et X_i autour de l'axe Z_{i-1} .
- a_i représente la distance entre les deux origines O_{i-1} et O_i le long de l'axe X_i .
- α_i représente l'angle entre les deux axes Z_{i-1} et Z_i autour de l'axe X_i .

Remarque :

1) Dans le cas où les axes (Z_{i-1} et Z_i) ont un point d'intersection, l'origine du repère(i) sera placé sur ce même point et l'axe X_i choisi comme étant le produit vectoriel ($Z_{i-1} * Z_i$), il sera aussi perpendiculaire à ces deux axes :

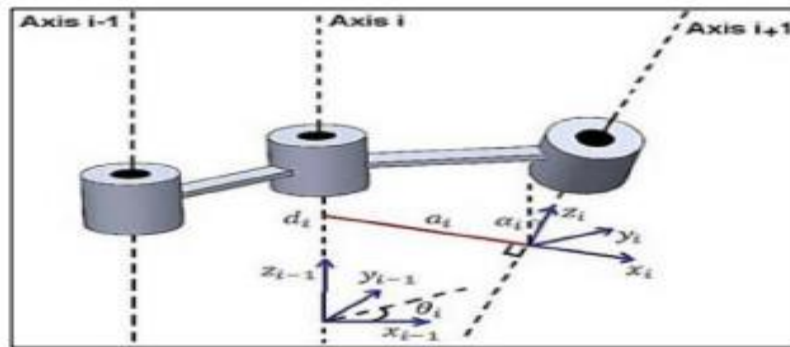


Figure 3.8– un point d'intersection [Dh95]

2) Concernant le cas où les axes (Z_{i-1} et Z_i) sont parallèles nous avons en nombre infini de droites perpendiculaires aux deux axes, dans cette situation il est préférable de Choisir l'axe X_i le long de la perpendiculaire passant par l'origine du repère O_{i-1} .

La figure 3.9 ci- dessous illustre un exemple de ce genre de situation

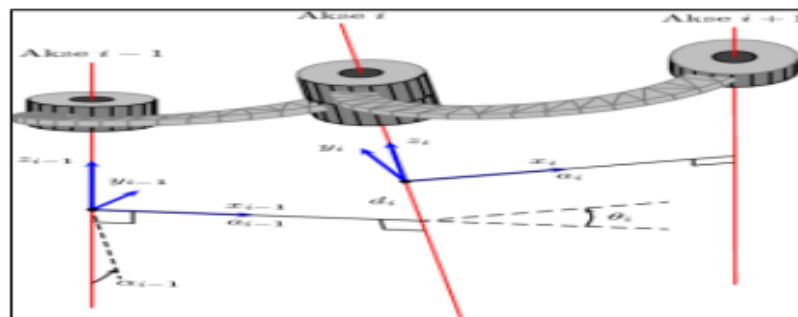


Figure 3.9– exemple de ce genre de situation[Dh95]

Tableau des paramètres **DH** : Après avoir placé les repères relatifs à toutes les articulations il faut dresser un tableau contenant les variables de la chaîne cinématique comme suit :

Articulation	Θ	d	a	α
1	Θ_1	d	a 1	a1
2	Θ_2	d	a2	a2
i	Θ_i	d	a _i	a _I
I+1	Θ_{i+1}	d	a _{i+1}	a _{I+1}
n	Θ_{n+1}	D	a _{n+1}	a _{n+1}

Tableau 3.1- les 4 paramètres D_ H

3.3.2 Géométrie

La non-conformité entre les valeurs nominales des paramètres géométriques et les valeurs réelles, c'est ce qui rend le contrôle de l'effecteur final imprécis. Dans ce travail, nous cherchons à améliorer la précision de notre bras manipulateur utilisés pour des opérations basique (Pick and place) à partir de l'identification des paramètres géométriques du robot.

Nous choisissons le bon choix d'un modèle géométrique propice pour atteindre la précision attendue. En effet, le choix d'un modèle géométrique adéquat nous permet de réduire le modèle aux paramètres géométriques strictement nécessaires au respect de la précision exigée par la tâche à réaliser.

Pour cela, nous utilisons une analyse de sensibilité des paramètres géométriques introduits par rapport à chacun des axes de repère du robot avant de réaliser l'identification des paramètres géométriques influents. Différents modèles géométriques de robots sont proposés et plusieurs techniques de calibration ont été développées dans le domaine scientifique. Le fait de réaliser l'identification des paramètres dans un espace de travail restreint nous permettra d'évaluer la pertinence de l'identification des paramètres géométriques dans un espace local vis-à-vis d'une identification globale dans tout l'espace de travail du robot. En effet, la littérature présente des approches sur l'identification globale des paramètres géométriques.

Dans ces travaux nous proposons une méthodologie pour l'amélioration de la précision du bras manipulateur.

L'objectif de ces travaux de thèse est d'améliorer la performance d'un robot sériel en termes de précision de pose de l'outil à partir de l'application d'une identification des paramètres géométriques dans le but de le rendre capable de réaliser des opérations d'usinage conformes au niveau de précision requis. [Rg10]

Les modèles géométriques directs et inverses : qui donnent l'endroit de l'effecteur en fonction des variables articulaires du mécanisme et vice versa.

3.3.3 Model Géométrique Direct

Pour analyser le comportement de la structure mécanique d'un robot ou d'un bras manipulateur, il est nécessaire de lier un repère orthonormé à chacun de ces corps et un référentiel $\{R0\}$ attaché à la base du robot ou du bras manipulateur. L'étude des mouvements des corps revient alors à l'étude des mouvements des repères. Etant donnés $1+n$ repères $\{R0\}$, $\{R1\}$, $\{Rn\}$

$$\overline{\theta_n P} = \begin{bmatrix} X_n \\ Y_n \\ Z_n \end{bmatrix} \dots \quad (3.2)$$

et

$${}^0P_P = {}^0T_n \cdot {}^nP_P \quad (3.3)$$

un point P défini relativement au repère P_n par :

De la relation

$${}^0T_n = {}^0T_1 \cdot {}^1T_2 \dots \dots \cdot {}^{n-2}T_{n-1} \cdot {}^{n-1}T_n \quad (3.4)$$

la connaissance de la position de P relativement à $\{R0\}$ revient à connaître la matrice de transformation :,,,,,,,,,vide3,,,,

Qui renvoie elle-même sur la connaissance de la position et de l'orientation de chaque repère $\{Ri\}$ $i=1\dots n$ par rapport au repère $\{Ri-1\}$. Plusieurs méthodes et notations ont été proposées : la notation de Denavit- Hartenberg, la notation de Paul, la notation de Khalil-Kleifinger, La méthode C-B proposée par Livitin et la méthode de Megahed. La plus répandue est celle de Denavit- Hartenberg.

3.3.4 Model Géométrique Inverse

Le modèle géométrique inverse (MGI) est l'ensemble des relations inverses à celles du modèle direct. Ce modèle permet d'exprimer les variables articulaires q du bras manipulateur en fonction des coordonnées opérationnelles X exigées pour l'exécution d'une tâche donnée.

$$q = F^{-1}(X)$$

À l'aide du programme matlab on a obtenu c'est résultat :

les paramètres DH :

tableau :

abb:: 4 axis, RRRR, stdDH, slowRNE

j	theta	d	a	alpha	offset
1	q1	0	9	0	0
2	q2	15	5	1.5708	0
3	q3	0	24	0	0
4	q4	0	27	0	0

Après les calculs des Paramètres DH et à l'aide du programme matlab toujours on a pu simuler notre bras manipulateur pour faciliter les calculs du modèle géométrique inverse :

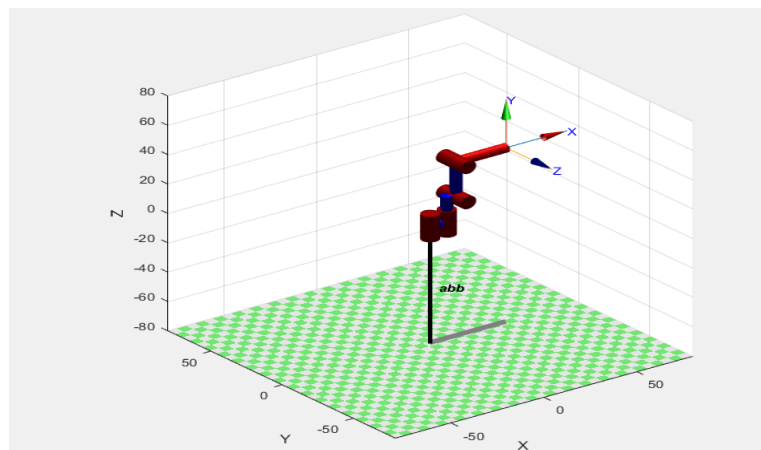


Figure 3.10– simulation avec matlab



Figure 3.11– Prototype réalisé ABB IRB-460

Modèle géométrique directe

D'après les paramètres DH on peut déduire les Matrice de transformation :

(Après la correspondance)

$$A_{01} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 9\cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 9\sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$A_{12} = \begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) & 5\cos(\theta_2) \\ \sin(\theta_2) & 0 & -\cos(\theta_2) & 5\sin(\theta_2) \\ 0 & 1 & 0 & 16 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$A_{23} = \begin{bmatrix} \cos(\theta_3) & 0 & \sin(\theta_3) & 24\cos(\theta_3) \\ \sin(\theta_3) & 0 & -\cos(\theta_3) & 24\sin(\theta_3) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$A_{34} = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 27\cos(\theta_4) \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 27\sin(\theta_4) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$A_{04} =$$

$$[\sin(\theta_1 + \theta_2)\sin(\theta_4) + \cos(\theta_1 + \theta_2)\cos(\theta_3)\cos(\theta_4), \cos(\theta_1 + \theta_2)\sin(\theta_3), \cos(\theta_1 + \theta_2)\cos(\theta_3)\sin(\theta_4) - \sin(\theta_1 + \theta_2)\cos(\theta_4), 5\cos(\theta_1 + \theta_2) + 9\cos(\theta_1) + 27\sin(\theta_1 + \theta_2)\sin(\theta_4) + (27\cos(\theta_3 - \theta_4)\cos(\theta_1 + \theta_2))/2 + (27\cos(\theta_1 + \theta_2)\cos(\theta_3 + \theta_4))/2 + 24\cos(\theta_1 + \theta_2)\cos(\theta_3)]$$

$$[\sin(\theta_1 + \theta_2)\cos(\theta_3)\cos(\theta_4) - \cos(\theta_1 + \theta_2)\sin(\theta_4), \sin(\theta_1 + \theta_2)\sin(\theta_3), \cos(\theta_1 + \theta_2)\cos(\theta_4) + \sin(\theta_1 + \theta_2)\cos(\theta_3)\sin(\theta_4), 5\sin(\theta_1 + \theta_2) + 9\sin(\theta_1) + (27\cos(\theta_3 - \theta_4)\sin(\theta_1 + \theta_2))/2 + (27\cos(\theta_3 + \theta_4)\sin(\theta_1 + \theta_2))/2 + 24\sin(\theta_1 + \theta_2)\cos(\theta_3) - 27\cos(\theta_1 + \theta_2)\sin(\theta_4)]$$

$$[\cos(\theta_4)\sin(\theta_3), -\cos(\theta_3), \sin(\theta_3)\sin(\theta_4), 24\sin(\theta_3) + 27\cos(\theta_4)\sin(\theta_3) + 16]$$

$$[0, 0, 0, 1]$$

Modèle géométrique Inverse calculer :

D'après la théorie de de modèle géométrique on peut résoudre le modèle géométrique inverse, avec une méthode de deux parties :

Vue haut :

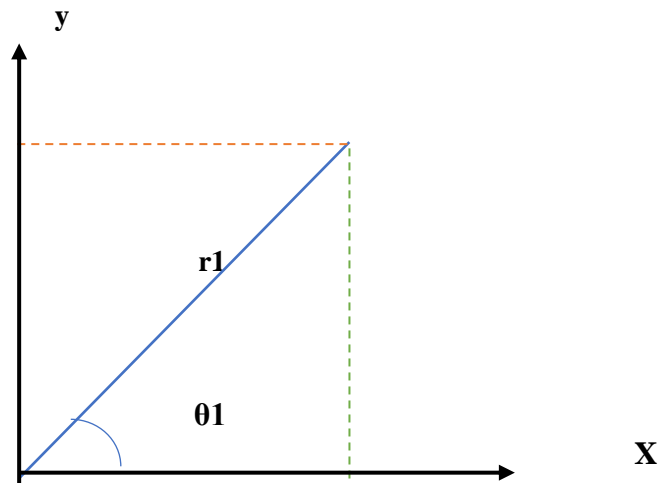


Figure 3.12– vue haut de bras manipulateur

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) \tag{3.5}$$

$$r_1 = \sqrt{x^2 + y^2} \tag{3.6}$$

Vue en Face :

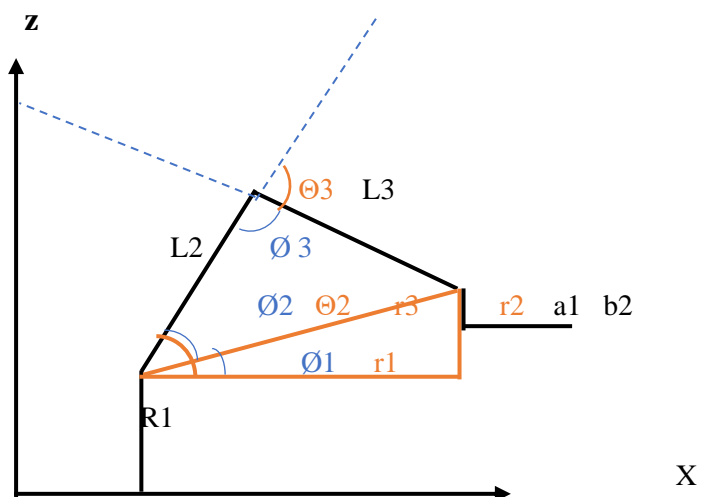


Figure 3.13– vue en Face de bras manipulateur

$$\theta_2 = \phi_1 + \phi_2 \quad (3.7)$$

$$\phi_1 = \tan^{-1}\left(\frac{r_2}{r_1}\right) \quad (3.8)$$

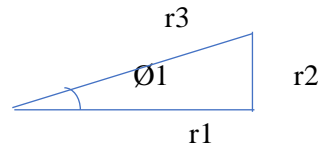


Figure 3.14– vue en Face de bras manipulateur par tone

Calculer ϕ_2

Pour obtenir ϕ_2 on a :

$$l_3^2 = l_2^2 + r_3^2 - 2 \cdot l_2 \cdot r_3 \cdot \cos \phi_2 \quad (3.9)$$

Donc

$$\phi_2 = \cos^{-1}\left(\frac{l_3^2 - l_2^2 - r_3^2}{-2 \cdot l_2 \cdot r_3}\right) \quad (3.10)$$

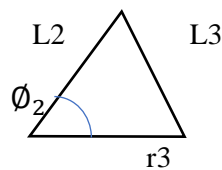


Figure 3.15– Triangle explicateur

On va Calculer (r_1, r_2, r_3)

On a déjà r_1 dans l'équation (3.6)

$$r_2 = Z - l_1 \quad (3.11)$$

$$r_3 = \sqrt{r_1^2 + r_2^2} \quad (3.12)$$

$$\theta_3 = \pi - \phi_3 \quad (3.13)$$

$$r_3^2 = l_2^2 + l_3^2 - 2l_2 \cdot l_3 \cos \phi \quad (3.14)$$

$$\phi_3 = \cos^{-1} \left(\frac{r_3^2 - l_2^2 - l_3^2}{-2 \cdot l_2 \cdot l_3} \right) \quad (3.15)$$

Donc notre équation de modèle géométrique inverse est :

$$\theta_1 = \tan^{-1} \left(\frac{y}{x} \right)$$

$$\theta_2 = \phi_1 + \phi_2$$

$$\theta_2 = \tan^{-1} \left(\frac{Z - l_1}{\sqrt{x^2 + y^2}} \right) + \cos^{-1} \left(\frac{l_3^2 - l_2^2 - (x^2 + y^2 + (Z - l_1)^2)}{-2 \cdot l_2 \cdot \sqrt{x^2 + y^2 + (Z - l_1)^2}} \right) \quad (3.16)$$

$$\phi_3 = \cos^{-1} \left(\frac{x^2 + y^2 + (Z - l_1)^2 - l_2^2 - l_3^2}{-2 \cdot l_2 \cdot l_3} \right) \quad (3.17)$$

3.4 Conclusion

dans ce chapitre on résumé l'essentiel de notre travail on a tenu a expliqué les différentes étapes de façon détailler et stratégique pour le montage de notre prototype (bras manipulateur) avec une définition de chaque partie et propriété comme on a parlé de la partie modélisation qui inclus bien évidemment les paramètres de denavit Hardenberg, le modèle géométrique directe et pour finir le modèle géométrique inverse de notre prototype et pour finir on a donné un exemple d'application consistant à faire déplacer un objet d'un endroit à un autre.

Conclusion Générale

La réalisation de ce projet nous a permis de découvrir et surtout d'affronter les nombreux problèmes rencontrés lorsque nous ne disposons pas des outils nécessaires pour travailler dans le domaine de la robotique. Le premier défi était donc de construire une structure mécanique suffisamment robuste et fiable capable supporter les différentes contraintes liées aux rotations des différentes articulations de ce robot. Cette partie nous a pris beaucoup de temps mais nous a porté une grande satisfaction aux vues de son comportement durant les différents tests que nous avons effectués. L'implémentation de la cinématique inverse du bras proposé a été présentée au chapitre 2, l'algorithme proposé, qui dans un premier temps a été testé sous Matlab, a été concrétisé dans le chapitre 3. La solution proposée qui consiste à utiliser l'environnement Matlab pour programmer les mouvements du robot s'est avérée très utile dans le sens où cela permet d'exploiter ce bras par un opérateur qui n'a pas nécessairement de connaissances dans le domaine de la robotique.

Ce mémoire présente notre travail qui concerne la réalisation d'un bras manipulateur à base d'une carte Arduino.

L'aboutissement de ce projet nous a permis de voir et de découvrir une multitude de problème, notamment sur la partie mécanique et outillage qui est indispensable pour un travail dans le milieu de la robotique.

Le tout premier challenge était de trouver une matière à la fois rigide et légère suffisamment solide et au même temps fiable pour pouvoir supporter les difficulté liée au poids, rotation et articulation (rotoïde) de ce prototype, sans oublier bien évidemment les accessoire mécanique utilisé dans on a démontrer dans le chapitre deux.

On a consacré une grande partie de notre temps dans cette partie mais à dire vrai elle nous a apporter une certaine satisfaction aux vues de son amélioration après chaque essaye qu'on a pu effectuer durant tout ce temps.

On a parlé de la partie modélisation qui et principalement le modèle géométrique inverse de notre prototype expliqué dans le chapitre 3 avec une simulation Matlab pour voir les mouvements du robot et une concrétisation avec le prototype qui nous permet le déplacement d'un objet d'un point à un autre dans un espace de travail réduit en quelque sorte un quart de sphère.

Dans ce projet on a essayé de mettre en place toute la documentation possible par

conséquence beaucoup de relation théorique on était mise en place.

Le domaine de la robotique est très vaste ce qui est le cas de notre travail et il reste une part d'amélioration, de perfectionnement à faire plus précisément sur la partie mécanique.

Par ailleurs, le secteur des bras robotiques, ouvert au développement, conservera son importance à l'avenir.

Bibliographie

- [Ach07] A. Chrifa, Analyse Cinématique des Robots Parallèles. Thèse Docteur D'état. Université Mentouri Constantine 2007/2008.
- [Dh95] Denavit, J., R.S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," ASME Journal of Applied Mechanics, June 1955, 215–221.
- [Djm16] DJAFRI Menad et CHELOUCHE Djalal, « Etude et Réalisation d'une Carte Arduino », mémoire master, Université Bejaia (2016).
- [Djo12] DJOUGGANE Fadhila ETUDE ET CONCEPTION D'UN ROBOT MANIPULATEUR POUR L'ENMTP, Mémoire Majestère, Université Hadj-Lakhdar Faculté des Technologies Département Génie Industriel, 2012.
- [Eda01] E. Dombre, Analyse et modélisation des robots manipulateurs, 2ème éd. France : Hermès Sciences, 2001.
- [Erk 14] Erik Bartmann, « LE GRAND LIVRE D'ARDUINO », (02-01-2014).
- [Jag 06] Jacques Gangloff « Cours de Robotique » +figure + Figure manipulation cartésienne
- [Jen06] Jean-Noël « Livret Arduino en français », Centre de Ressources Art Sensitif, novembre 2006.
- [Lab18] LAIB Bachir et SAADANI Fathi, « Commande d'une Barrière Automatique à Base d'Arduino », mémoire master, Université El-Oued (2018).
- [Mew18] MEKKI Walid et BELHADJ Mohammed, « Conception et réalisation d'un panneau solaire à base d'Arduino », mémoire master, Universitaire d'Ain-Temouchent (2018)
- [Rg10] R. Gourdeau, ELE4203 — Robotique, Modélisation des Robots Manipulateurs, Département de génie électrique, Ecole Polytechnique de Montréal, 4 novembre 2010.
- [Tay18] Tazerout Yacine, « Commande d'un bras robotisé avec une application androïde », mémoire master, Université LORRAINE (2018).
- [Zeg94] S. Zeghloul. Lallemand, Robotique. Aspects fondamentaux, page 312, Masson 1994.
- [Zeg94] S. Zeghloul. Lallemand, Robotique. Aspects fondamentaux, page 314, Masson 1994.

- [01] <http://www.usinenouvelle.com/article/bios-de-robots-unimate-lepremierouvriermecanique>. N276769
- [02] <https://www.istockphoto.com/be/vectoriel/bras-robotis%C3%A9-bras-m%C3%A9caniquemanipulateur>. (05-06-2019)
- [03] <https://alucobond.com/alucobond-idea?locale=fr>
- [04] PCA en ligne <https://www.digikey.com/en/supplier-centers/pca-electronics>

Résumé

De nos jours, la robotique s'est imposée comme un phénomène de l'industrie moderne.

Nous suggérons dans ce manuscrit fin d'études que nous l'avons fait en étudiant et en réalisant un modèle de bras manipulateur robotique à 3 degrés de liberté ABB IRB 460 dont la tâche principale est de disposer les marchandises et en exploitant l'ingénierie géométrique qui nous permet de contrôler avec une grande précision et basé sur la carte Arduino qui nous a permis de programmer l'alarme du manipulateur.

ملخص

في وقتنا الحاضر، استطاع علم الروبوتات أن يثبت نفسه كظاهرة للصناعة العصرية. نقترح في هذه النسخة من مذكرة التخرج التي انجزناها بدراسة و انجاز نموذج روبوت ذراع مناور ذو 3 درجات للحرية ABB IRB 460 مهمته الأساسية هي ترتيب السلع و باستغلال الهندسة الجيومترية تخول لنا التحكم بدقة عالية و استنادا الى بطاقة اردوينو التي مكنتنا من برمجة الذراع المناور

Abstract

Nowadays, robotics has established itself as a phenomenon in modern industry.

We suggest in this end-of-study manuscript that we have done so by studying and realizing a model of an ABB IRB 460 3-degree-of-freedom robotic manipulator arm whose main task is to arrange the goods and by exploiting the geometric engineering that allows us to control with great precision and based on the Arduino board that allowed us to program the manipulator alarm.

Annexe

```
////////////////////////////////////
////////////////////////////////////
#include <math.h>
#include <VarSpeedServo.h>
////////////////////////////////////
////////////////////////////////////

VarSpeedServo motor1;
VarSpeedServo motor2;
VarSpeedServo motor3;
VarSpeedServo motor4;

////////////////////////////////////
////////////////////////////////////

float L1 =16;
float L2 =24;
float L3 =27;
float x =29.93;
float y =27.93;
float z =28.90;
float theta1;
float theta2;
float theta3;
float x1 =10;
float y1 =50;
float z1 =20;
float pi =3.14159265359;
```

```
float ph1;
float ph2;
float ph3;
float ph21;
float ph22;
float ph31;
float ph32;
float ph11;
float ph202;
float ph33;
float ph212;
float ph222;
float ph312;
float ph322;
void setup() {
  motor1.attach(3);
  motor2.attach(5);
  motor3.attach(9);
  motor3.attach(11);
  motor1.write(0,30);
  motor1.wait();
  motor2.write(0,30);
  motor2.wait();
  motor3.write(0,30);
  motor3.wait();
}
```

```

Serial.begin(9600);
}
void loop(){

float angle1;
float angle2;
float angle3;
float radangle1;
float radangle2;
float radangle3;
float ANGLE1;
float ANGLE2;
float ANGLE3;
float angle11;
float angle22;
float angle33;
float PH3;
float radangle11;
float radangle22;
float radangle33;
////////////////////////////////
////////////////////////////////
////////////////////////////////
    ph1 = atan((z-L1)/(sqrt(sq(x)+sq(y))));
////////////////////////////////
////////////////////////////////
    ph21= sq(L3)-sq(L2) - sq(z-L1)-sq(x)-sq(y);
    ph22 = - 2 * L2 * sqrt(sq(z-L1)+sq(x)+sq(y));

```

```

////////////////////
    ph2= acos(ph21/ph22);
////////////////////
    ph31= sq(z-L1) + (sq(x)+sq(y)) - sq(L2)-sq(L3);
    ph32= (-2*L2*L3 );
    ph3= acos(ph31/ph32);
////////////////////
PH3= (ph3*180)/pi;
////////////////////
    radangle1= atan( y/x );
    theta1=radangle1;

    Serial.print("th1 is ");
    //Serial.println(radangle1);
    Serial.println(theta1);
////////////////////
    radangle2 = ph2+ph1;
    Serial.print("th2 is ");
    theta2=radangle2;
    // Serial.println(radangle2);
    Serial.println(theta2);
////////////////////
    radangle3 = pi-ph3 ;
    Serial.print("th3 is ");
    theta3 = radangle3;

    // Serial.println(radangle3);
    Serial.println( theta3);

```

```

////////////////////////////////////
////////////////////////////////////

    ANGLE1= (theta1*180)/pi;
    angle1= ANGLE1*2;

    ANGLE2=(theta2*180)/pi;
    angle2= ANGLE2;

    ANGLE3= (theta3*180)/pi;
    angle3= ANGLE3;
    Serial.print("PH3 is "); Serial.println(PH3);
    Serial.print(" ANGLE3 is "); Serial.println( ANGLE3);

////////////////////////////////////
////////////////////////////////////

    motor1.write(0,25);
    motor1.wait();delay(1000);
    motor2.write(angle2,25);
    motor2.wait();delay(1000);
    motor3.write(angle3,25);
    motor3.wait();delay(1000);
    motor4.write(90,25);
    motor4.wait();delay(1000);

////////////////////////////////////
////////////////////////////////////

    Serial.print("angle1 is "); Serial.println(angle1);
    Serial.print("angle2 is "); Serial.println(angle2);
    Serial.print("angle3 is "); Serial.println(angle3);

```

```

ph11 = atan((z1-L1)/(sqrt(sq(x1)-sq(y1))));
ph212= sq(L3)+sq(L2) - sq(z1-L1)-sq(x1)-sq(y1) ;
ph222 = - 2 * L1 * sqrt(sq(z1-L1)+sq(x1)+sq(y1));
ph202= acos(ph21/ph22);
ph312= sq(z1-L1) + sq(x1)+sq(y1) - sq(L2)-sq(L3);
ph322= (-2*L2*L3 );
ph33= acos( ph31/ph32 );

```

```

radangle11= atan2( y1,x1 );
Serial.print("th11 is ");
Serial.println(radangle11);

```

```

radangle22 = ph2+ph1 ;
Serial.print("th22 is ");
Serial.println(radangle22);

```

```

radangle33 = pi - ph33;
Serial.print("th33 is ");
Serial.println(radangle33);

```

```

angle11= ((radangle11*180)/pi)*2;
angle22= (radangle22*180)/pi;
angle33= (radangle33*180)/pi;

```

```

////////////////////

```

```

////////////////////

```

```

Serial.print("angle11 is "); Serial.println(angle11);
Serial.print("angle22 is "); Serial.println(angle22);
Serial.print("angle33 is "); Serial.println(angle33);

```

```
////////////////////////////////////
```

```
////////////////////////////////////
```

```
motor1.write(angle11,50);
```

```
motor1.wait();delay(1000);
```

```
motor2.write(angle22,50);
```

```
motor2.wait();delay(1000);
```

```
motor3.write(angle33,50);
```

```
motor3.wait();delay(1000);
```

```
motor4.write(0,50);
```

```
motor4.wait();delay(1000);
```

```
delay(2000);
```

```
}
```