

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



Université Ibn Khaldoun de Tiaret
Faculté des Sciences Appliquées
Département de Génie Mécanique



MÉMOIRE DE FIN D'ÉTUDES

Pour l'obtention du Diplôme de Master

Domaine : Sciences et Technologie

Filière : Génie Mécanique

Parcours : Master

Spécialité : Construction Mécanique

Thème

Étude et réalisation d'un bras
manipulateur

Préparé par :

DJAAFRI Lakhdar Nouredinne et KHENFAR Racha

Soutenu publiquement le : / 07 / 2021, devant le jury composé de :

M. HAMMOU Mahmoud	Maître de Assistant "A"(Univ. Ibn Khaldoun)	Président
M. BEY Mohamed	Maître de Conférences "B"(Univ. Ibn Khaldoun)	Examineur
M. GUEMMOUR Mohamed Boutkhil	Maître de Conférences "B" (Univ. Ibn Khaldoun)	Examineur
M. MECHEKOUR El-Hadi	Maître de Conférences "B" (Univ. Ibn Khaldoun)	Encadreur

Année universitaire : 2020 - 2021



REMERCIEMENTS

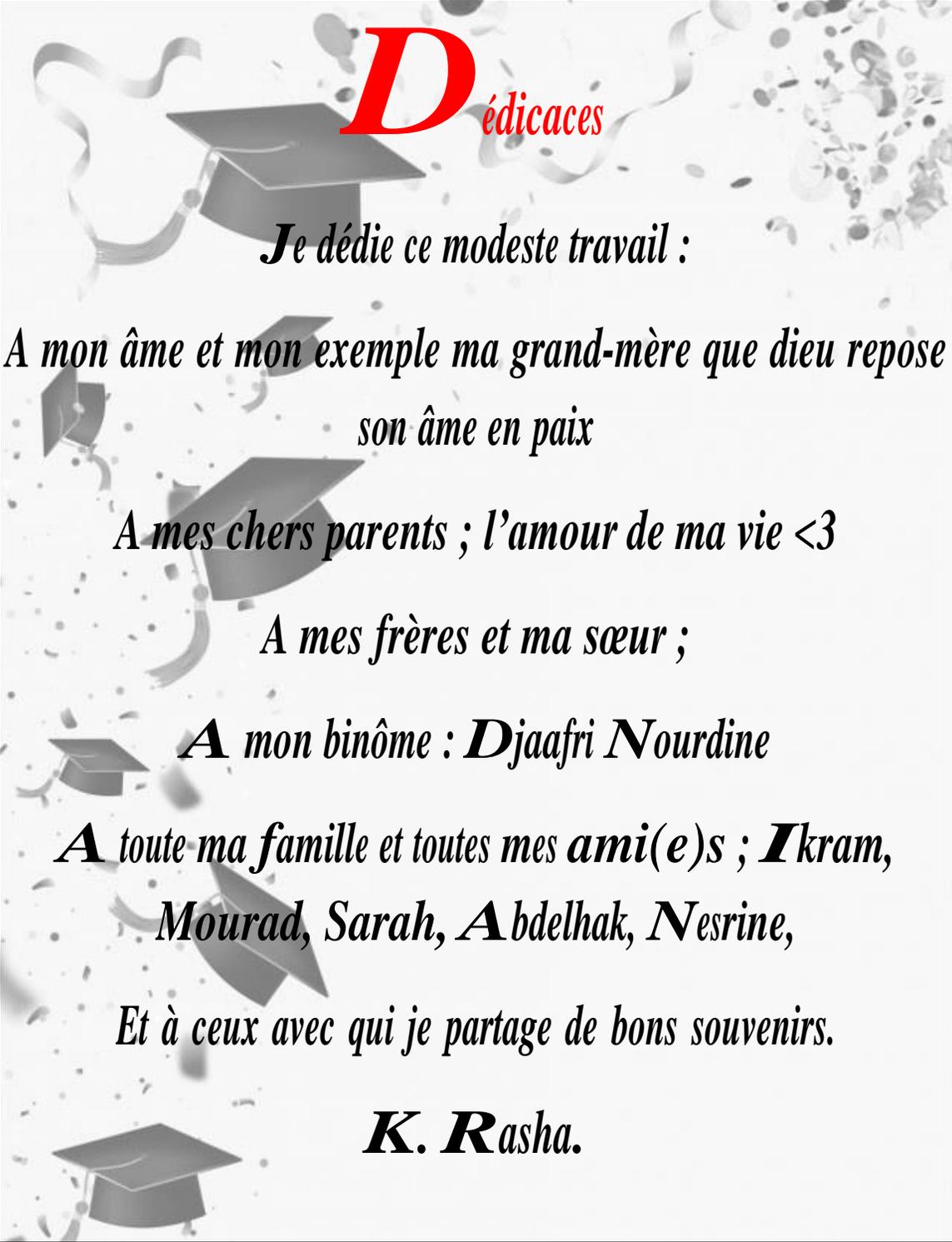
Nous tenons à remercier le Dieu tout puissant de nous avoir donné le courage, la volonté, ainsi que la santé pour réaliser ce modeste travail.

Nous tenons à exprimer nos sincères remerciements et notre reconnaissance à MR MECHEKOUR EL HADI pour son encadrement et sa précieuse disponibilité qu'il a bien voulu nous accorder. Son expérience académique, sa rigueur intellectuelle et ses conseils dans le but de réaliser ce travail, nous ont permis de nous développer et de réussir ce projet.

Nous remercions très respectueusement Mr HAMMOU Mahmoud et Mr GUEMMOUR Mohammed Boutkhil, et Mr BEY Mohammed, de nous avoir fait l'honneur de juger notre modeste travail.

On ne saura clôturer ces remerciements sans exprimer toute notre affection pour les enseignants de département ST et département génie mécanique de Tiaret, qui ont bien voulu partager leurs savoirs durant nos études.

Enfin, nous remercions tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet.



Dédicaces

Je dédie ce modeste travail :

*A mon âme et mon exemple ma grand-mère que dieu repose
son âme en paix*

A mes chers parents ; l'amour de ma vie <3

A mes frères et ma sœur ;

*A mon binôme : **D**jaafri **N**ourdine*

*A toute ma famille et toutes mes **a**mi(e)s ; **I**kram,
Mourad, **S**arah, **A**bdelhak, **N**esrine,*

Et à ceux avec qui je partage de bons souvenirs.

K. Rasha.

DEDICACES

Je dédie ce travail à ceux à qui quels que soient les termes embrassés, je n'arriverais jamais à exprimer mon amour sincère:

- ✓ *A mes adorables parents qui m'ont toujours encouragé durant toute ma vie.*
- ✓ *A ma grand mère*
- ✓ *A mes chers frères et sœurs à qui je souhaite beaucoup le bonheur*
- ✓ *A tous les cousin(e)s, mes ami(e)s que j'ai connu jusqu'à maintenant. Merci pour leurs encouragements.*
- ✓ *A mon frère et mon ami Kociela Sahir qui ma aidé pour engager mon projet de fin des études*
- ✓ *A mon binôme Khenfar Racha*
- ✓ *A toute la promotion construction mécanique 2021*

NOUREDINE

Liste des abréviations

A.F.N.O.R. L'Association Française de Normalisation

RUR (Rossum's Universal Robots).

SRI Stanford Research Institute

MTBF (Mean Time Between Failures)

3D trois dimensions

CAO conception assisté par ordinateur

MCC moteurs à courant continu

STL (STereo-Lithography)

DH Denavit-Hartenberg

DHM (Denavit-Hartenberg Modifié)

D.D.L degrés de liberté

MGD modèle géométrique direct

RTL Roulis-Tangage-Lacet

RRR (ROTATION, ROTATION, ROTATION)

MGI le modèle géométrique inverse

MIT Massachusetts Institute of Technology

SCARA selective compliance assembly robot arm

La liste des figures

Figure I.1 : Premiers automates ancêtres des robots actuels.....	5
Figure I.2 : Premiers robots mobiles.....	6
Figure I.3 : Premiers robots industriels.....	6
Figure I.4 : Premiers robots mobiles intelligents	7
Figure I.5 : Développement des robots intelligents.....	7
Figure I.6 : Robots mobiles d’exploration extraplanétaire	8
Figure I.7 : Démocratisations des robots.....	8
Figure I.8 : Bras manipulateur de dernière génération.....	10
Figure I.9 : Manipulateur pneumatique.....	10
Figure I.10 : Architecture de robot fanuc.....	13
Figure I.11 : Représentation du volume de travail.....	14
Figure I.12 : Structure cylindrique RPP	16
Figure I.13 : Structure cartésienne (ppp).....	17
Figure I.14 : Structure RRP	17
Figure I.15 : Structure RRR	18
Figure I.16 : Utilisation de la robotique pour l’usinage de matériaux différents	19
Figure I.17 : Tête d’usinage sur un bras robotisé	19
Figure I.18 : Ligne de packaging avec convoyeurs.....	20
Figure I.19 : Utilisation des bras manipulateurs en industrie automobile.....	21
Figure I.20 : Pistolet de peinture sur un bras robotisé.....	21
Figure I.21 : Robotique en science de la vie	22
Figure I.22 : Robotisation d’une chaîne de production du chocolat	23
Figure II.1 : Pièces constituant le bras manipulateur SNAM1900.....	25
Figure II.2 : Bras manipulateur	26
Figure II.3 : la base d’un Bras manipulateur.....	27
Figure II.4 : le porteur d’un Bras manipulateur	28
Figure II.5 : le pince d’un bras manipulateur	28

Figure II.6 : un servomoteur	29
FigureII.7 : Branchement électrique de servomoteur.....	30
FigureII.8 : brochage de la carte UNO.....	34
FigureII.9 : Arduino UNO... ..	35
Figure II.10 : le servomoteur avec l'Arduino... ..	36
Figure II.11 : Bluetooth HC5.....	36
Figure II.12 : l'Arduino avec Bluetooth HC5.....	37
FigureII.13 : Microcontrôleur	38
FigureII.14 : les Shields Arduino.....	38
Figure II.15 . Utilisation de la fonction Print3D du logiciel SolidWorks avec notre pièce	39
Figure II.16 . Choix des paramètres de la fonction Print3D	40
Figure II.17 . Exportation de la préparation à l'impression de la pièce en format STL... ..	40
Figure II.18 . Interface du logiciel Prusa Slicer de l'imprimante 3D Prusa.....	41
Figure II.19 . Importation du fichier STL de la pièce dans le logiciel de l'imprimante 3D Prusa	41
Figure II.20 . Génération des couches d'impression de la pièce ainsi que le G-Code associé	42
Figure II.21 . Vue d'une coupe de la pièce avec un remplissage de 70%	43
Figure II.22 . Simulation de l'impression 3D de la pièce	43
Figure III.1 : Modèle géométrique d'une chaîne simple.....	47
Figure III.2 : Type de chaînes cinématiques.....	48
Figure III.3 : Chaîne cinématique simple ouverte	48
FigureIII.4 : Chaîne cinématique arborescente.....	49
FigureIII.5 : Chaîne cinématique fermée	50
Figure III.6 : des modèles cinématique.....	51
Figure III.7 : Robot plan à 2ddl	52
Figure III.8 : Exemples de robots plans à 3 ddl le 3-RPR et le 3-RRR	52
FigureIII.9 : modèle RRR	54
Figure III.10 : Robot a structure ouverte	55
FigureIII.11 : méthode de transformation homogène	56

Figure III.12 : Modèle Khalil.....	58
FigureIII.13 : exemple de méthode de Khalil.....	60
FigureIII.14 : diagramme 3D D'un Bras.....	63
FigureIII.15 : L'espace de travail 3D (oxyz) et 2D (oxy,oyz,oxz).....	68
Figure VI.1 : Assemblage du bras manipulateur (Avant – Après).....	71
Figure VI.2 : Assemblage des composants du Shield Arduino (Avant – Après).....	72
FigureVI.3 : Environnement de travail vu de dessous.....	73
Figure VI.4 : Environnement de travail vu de droite.....	73
Figure VI.5 : Organigramme de l'algorithme de commande manuelle du robot.....	74
Figure VI.6 : Interface qui permet le design de notre application sous App Inventor.....	77
Figure VI.7 : Interface finale de l'application.....	78
Figure VI.8 : Les blocs de programmation application via Bluetooth.....	79
Figure VI.9 : Organigramme de l'algorithme de commande à distance du robot.....	80
Figure VI.10 : Organigramme de l'algorithme de commande « Pick-and-Place » du robot.	82
Figure VI.11 : les blocs de deux boutons sauvegarder et Lancer le cycle automatique	83

SOMMAIRE :

Remerciement	I
Dédicace	II
Liste des abréviations	IV
Liste figures	V
INTRODUCION GENERALE	1
CHAPITRE I : Généralités sur la Robotique	
I.1 Introduction	4
I.2 Définitions	4
I.2.1 Définition d'un robot.....	4
I.2.2 Définition de la robotique.....	4
I.3 Historique de la robotique	5
I.4 Type de robots	7
1.4.1 Robots mobiles.....	9
1.4.2 Robots domestiques	9
1.4.3 Robots collaboratifs	9
1.4.4 Robots humanoïdes.....	9
1.4.5 Robots industriels.....	9
1.4.6 Robots manipulateurs.....	9
I.5 Architecture d'un robot	11
I.5.1 La Base	11
I.5.2 Le segment.....	11
I.5.3 Articulation.....	11
I.5.3.1 Articulation rotoïde.....	11
I.5.3.2 Articulation prismatique	11
I.5.4 L'actionneur.....	11
I.5.5 Organe terminal	12

I.5.6 Les capteurs	12
I.5.7 La commande	12
I.6 Les caractéristiques techniques du robot	13
I.6.1 Le volume atteignable (volume de travail).....	13
1.6.2 Charge utile.....	14
1.6.3 Précision / Répétabilité	14
1.6.4 La vitesse	14
1.6.5 L'orientation possible de l'organe terminal.....	15
1.6.6 La fiabilité.....	15
I.7 Classification géométrique des robots	16
I.7.1 Structure cylindrique RPP ou PRP	16
I.7.2 Structure cartésienne (PPP)	16
I.7.3 La structure SCARA ou polaire à axe de rotation orthogonale (RRP).....	17
I.7.4 La structure anthropomorphe (RRR).....	18
I.8 Domaine d'application de la robotique.....	18
I.8.1 L'usinage	18
I.8.2 Conditionnement/packaging.....	20
I.8.3 L'assemblage	20
I.8.4 Peinture.....	21
I.8.5 Sciences de la vie.....	22
I.8.6 L'agroalimentaire	22
I.9 Conclusion.....	23

CHAPITE II : Étude du Bras Manipulateur

II.1. Introduction.....	25
II.2. Choix du bras manipulateur étudié.....	25
II.3. Description de l'architecture du bras manipulateur	26
II.3.1. Présentation du bras manipulateur	26

II.3.2. Définition de la base.....	27
II.3.3. Définition le porteur	27
II.3.4. Définition de la pince	28
II.3.5. Définition de servomoteur.....	28
II.4. Description des autres parties constitutives du bras manipulateur.....	29
II.4.1. Les actionneurs du bras manipulateur	29
II.4.2. Les servomoteurs.....	29
II.4.3. Les types de servomoteur	30
II.5. Les capteurs.....	31
II.5.1 Les types de capteurs.....	31
II.5.1.1. Capteurs.....	31
II.5.1.2. Cameras.....	31
II.5.1.3. Capteurs infrarouges.....	32
II.5.1.4. Capteurs magnétiques	32
II.5.1.4. Radar	32
II.6. La carte programmable ARDUINO UNO.....	32
II.6.1 Brochage de la carte UNO.....	34
II.6.2 L'utilisation de la carte Arduino UNO.....	34
II.6.3 Utilisation d'un servomoteur avec l'Arduino	35
II.7. Définition de Bluetooth HC5	36
II.7.1 Câblage de Arduino avec Bluetooth HC5	36
II.7.2 Matériel utilise pour câblage	37
II.7.3. Présentation de Bluetooth HC 5	37
II.7.4. Le microcontrôleur	38
II.7.5 Shield Arduino	38
II.8. Réalisation des pièces par impression 3D.....	39
II.9. Conclusion.....	44

CHAPITRE III : Cinématique du Bras Manipulateur

III.1 Introduction.....	46
III.2. Modèle géométrique d'un robot en chaine simple.....	46
III.2.1. Nécessité d'un modèle.....	46
III.2.2. Mécanismes.....	47
III.2.2.1. Cinématiques série.....	48
III.2.2.2. Chaîne cinématique arborescente.....	49
III.2.2.3. Chaîne cinématique fermée.....	49
III.2.3. Morphologie des robots manipulateurs.....	50
III.3. Modèle plan.....	52
III.3.1. Robots plans à 2ddl.....	52
III.3.2. Robots plans à 3 ddl.....	52
III.4. Modèle géométrique direct.....	53
III.4.1. Modèle de Denavit Hartenberg (DH).....	54
III.4.2. Modèle de Khalil.....	58
III.5. Modèle géométrique indirect.....	60
III.5.1. Modélisation cinématique.....	61
III.5.1.1 Modèle cinématique direct.....	61
III.5.1.2 Modèle cinématique inverse.....	61
III.5.1.3. Forme générale du modèle cinématique.....	62
III.5.1.4. Modélisation dynamique.....	62
III.6. Application : bras manipulateur RRR.....	63
III.7. Conclusion.....	69

CHAPITRE VI: Application

VI.1. Introduction.....	71
VI.2. Assemblage du bras manipulateur.....	71
VI.3. Environnement de travail.....	72

VI.4. Partie I : Commande manuelle du bras manipulateur par potentiomètres.....	74
VI.4. Commande à distance du bras manipulateur par Bluetooth et application Android	76
VI.4.1. MIT Inventor.....	77
VI.4.1.1. Design de l'interface de l'application.....	77
VI.4.1.2. Programmation par block de notre application Android	78
VI.4.1.3. Programme Arduino.....	80
VI.5. Application « Pick-and-Place »	80
VI.5.1. Algorithme de la commande « Pick-and-Place ».....	82
VI.5.2. Programmation par block de l'application Android.....	83
VI.6. Conclusion.....	83
CONCLUSION GENERALE.....	84
Références Bibliographiques.....	85
ANNEXE A : Mises en plan des pièces du bras manipulateur	
ANNEXE B : Programmes Arduino	

INTRODUCTION GÉNÉRALE

Depuis la révolution industrielle, une discipline a marqué l'évolution du monde technologique : la Robotique. L'avènement des robots dans l'industrie a permis de soulager l'homme des travaux répétitifs et difficiles tels que : le déplacement d'objets lourds, les tâches d'assemblages, les microsoudures... etc

La **Robotique industrielle** est officiellement définie par l'Organisation Internationale de Normalisation ISO comme étant un système commandé automatiquement, multi-applicatif, reprogrammable, polyvalent, manipulateur et programmable sur trois axes ou plus.

L'objectif principal de notre travail consiste à l'étude théorique, ainsi qu'à la réalisation d'un bras manipulateur à quatre degrés de liberté, en utilisant une structure d'un squelette préexistant mu par des servomoteurs lui permettant d'avoir des déplacements d'une bonne précision. L'autre aspect de notre réalisation consistera à programmer la carte Arduino pour lui permettre d'exécuter des tâches d'une manière manuel ou à distance. Dans ce dernier cas une application Android sera développée pour manipuler le bras manipulateur par Bluetooth.

Pour décrire notre travail, nous avons organisé notre mémoire de la manière suivante :

Le premier chapitre donne un bref historique sur la robotique et les différents types de robots, analyse aussi les différents constituants d'un bras manipulateur qui constitue l'objet de notre étude. L'objet du premier chapitre est d'apporter quelques définitions de base et de décrire les constituants technologiques d'un robot et enfin définir les principaux termes du domaine.

Le deuxième chapitre traite de la structure du bras manipulateur et de la conception de son modèle CAO à l'aide du logiciel SolidWorks. Dans un deuxième temps nous allons présenter la carte programmable Arduino Uno et les composants électriques qui permettent de commander le bras manipulateur. Et pour finir, nous exposons la manière d'imprimer en 3D les pièces constituants le robot pour faciliter la reproductibilité de notre travail.

Dans le troisième chapitre, nous allons présenter les types des modèles géométrique direct et inverse ainsi que le modèle de Khalil et modèle de Denavit-Hartenberg et le modèle cinématique et la modélisation dynamique d'un bras manipulateur. Le modèle géométrique

direct de notre robot à quatre degrés de liberté est présenté. Pour finir, nous utilisons ce modèle pour calculer l'espace de travail en utilisant le logiciel Matlab.

Dans le quatrième et dernier chapitre, nous allons développer les programmes de commande du bras manipulateur. La création de l'application Android est présentée ainsi que la programmation par block.

CHAPITRE I :
Généralités sur la
Robotique

I.1 Introduction :

Dans ce chapitre, nous allons définir la robotique ainsi que certains concepts de la robotique dont nous aurons besoin dans les chapitres suivants. Nous parlerons des domaines de son application qui ne cessent de se développer chaque jour. Par la suite, nous allons nous intéresser plus particulièrement aux bras manipulateurs qui sont des robots très répandus dans le domaine de l'industrie qui nous intéresse et qui correspond à notre spécialité de formation à savoir la construction mécanique.

I.2 Définitions :

I.2.1 Définition d'un robot :

L'Association Française de Normalisation (A.F.N.O.R.) définit un robot comme étant un système mécanique de type manipulateur commandé en position, reprogrammable, polyvalent (i.e., à usages multiples), à plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces, des outils et des dispositifs spécialisés, au cours de mouvements variables et programmés pour l'exécution d'une variété de tâches. Il a souvent l'apparence d'un, ou plusieurs, bras se terminant par un poignet. Son unité de commande utilise, notamment, un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. Ces machines polyvalentes sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel [1][2]

I.2.2 Définition de la robotique :

La robotique est l'ensemble des techniques permettant la conception, la réalisation de machines automatiques ou de robots. L'ATILF définit le robot de la manière suivante : "Appareil effectuant, grâce à un système de commande automatique à base de micro-processeur, une tâche précise pour laquelle il a été conçu dans le domaine industriel, scientifique ou domestique".

De cette définition découlent deux interprétations. La première affirme qu'un robot est une machine, qui possède des capteurs, un système logique et des actionneurs. Il est matériel. La deuxième considère qu'un robot est un travailleur artificiel. Selon cette dernière, un robot peut être également virtuel. La robotique actuelle trouve des applications dans différents domaines : La

robotique industrielle, la robotique domestique, la robotique médicale ou encore robotique militaire [2] ,[3]

I.3 Historique de la robotique :

L'histoire de la robotique s'inscrit dans celle, plus vaste, du développement et de la prolifération des machines, qui débute au XVIII^e siècle avec la Révolution industrielle. Elle commence donc avec l'automate. La différence fondamentale entre l'automate et ce que deviendra le robot au XX^e siècle tient à ce que le premier obéit à un programme strictement préétabli, que ce soit de manière mécanique ou électrique, alors que le second dispose de capteurs et d'appareillages électroniques, de sorte que ses actions découlent de ses contacts avec son environnement, ce qui - à la différence de l'automate - le rend autonome, "intelligent".

❖ Avant 1950 : (voir figure I.1)

* XVIII^e éme siècle : création d'automates. En 1738, Jacques Vaucanson crée un canard articulé en cuivre capable de boire, manger, cancaner digérer comme un véritable animal.

* 1921 : Karel Capek , RUR (Rossum's Universal Robots).

* 1940 : Isaac Asimov écrit un ensemble de nouvelles sur les robots. Trois lois de la robotique.

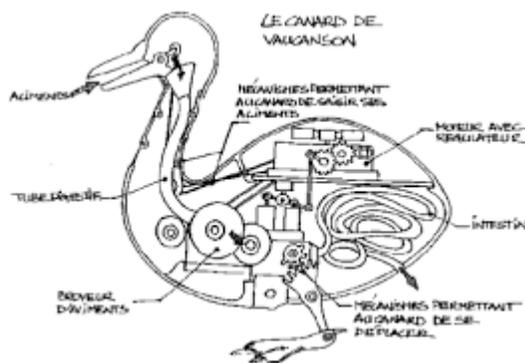


Figure I.1 : Premiers automates ancêtres des robots actuels

❖ 1950-1960 : (voir figure I.2)

* 1948 : Grey Walter invente le premier robot mobile autonome, une tortue se dirigeant vers les sources de lumière qu'elle perçoit. Cependant, ce robot n'est pas programmable.

* La mise en place de robots n'a été possible que dans les années 50 avec la création des transistors et circuits intégrés.

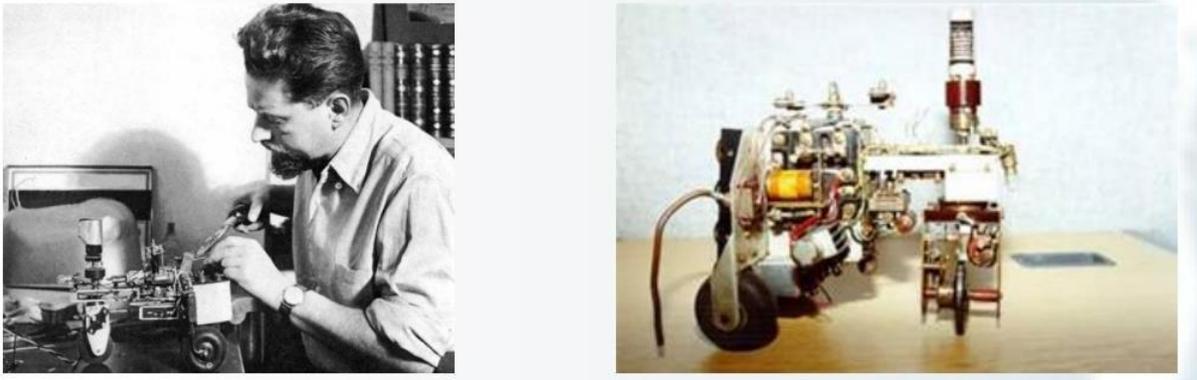


Figure I.2 : Premiers robots mobiles

❖ **1960-1970 : Robotique industrielle** (voir figure I.3)

* 1961 : Premier robot industriel mis en place dans une usine de General Motors : UNIMATE (tubes cathodiques de télévision).

* 1972 : Nissan ouvre la première chaîne de production complètement robotisée.

* 1978 : PUMA (Programmable Universal Machine for Assembly) développé par General Motors (toujours utilisé).

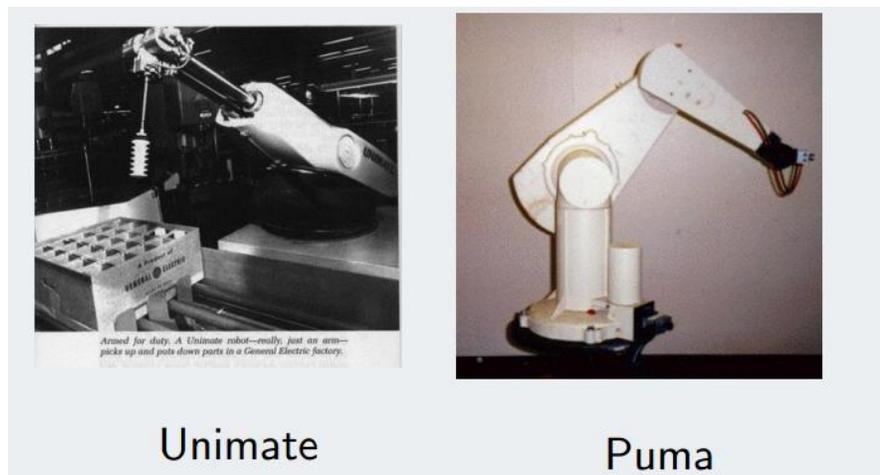


Figure I.3 : Premiers robots industriels

❖ **1960-80 : Premiers robots mobiles** (voir figure I.4)

1960-64 : Ouverture des laboratoires d'Intelligence Artificielle au M.I.T., Stanford Research Institute (SRI), Stanford University, University of Edinburgh.

- * Fin des années 60 : Mise en place de "Shakey" premier robot mobile intégrant perception, planification et exécution.
- * 1970 : Stanford Cart
- * 1977 : premier robot mobile français HILARE au LAAS (CNRS Toulouse)

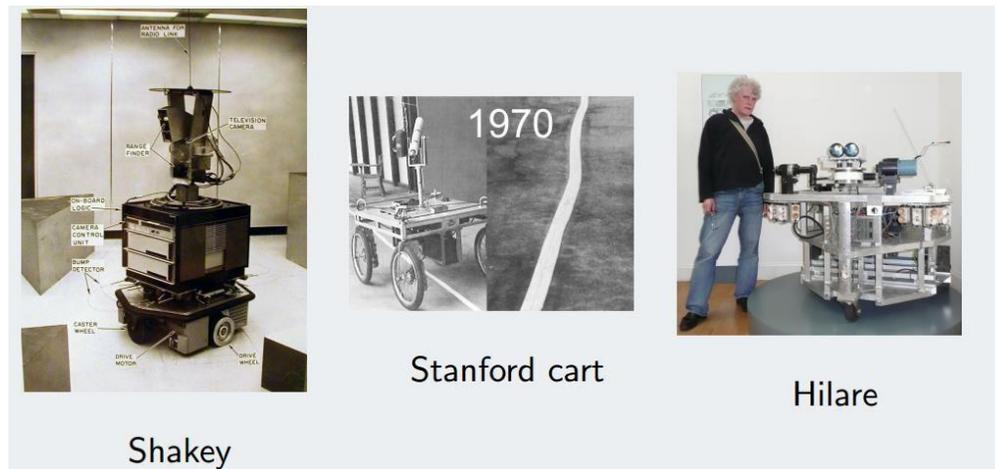


Figure I.4 : Premiers robots mobiles intelligents

❖ **1980-1990 :**

Développement important de l'intelligence artificielle et de la robotique : de nouveaux robots apparaissent constamment.

❖ **1990-2000 : Essor de la robotique mobile** (voir figure I.5)

- * 1992 : Mise en place de la compétition annuelle AAAI sur la robotique mobile.
- * 1995 : Mise en place de la RoboCup (lien vidéo).
- * 1997 : premier robot mobile extraplanétaire sur Mars.
- * 1999 : Lancement de Aibo.



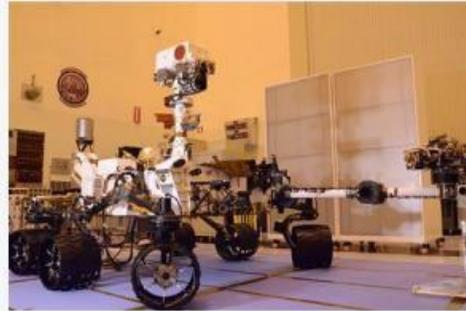
Figure I.5 : Développement des robots intelligents

❖ **Depuis 2000 : Exploration** (voir figure I.6)

- * 2003 : Projet "Mars Exploration Rover" (Spirit & Opportunity).
- * 2009 : projet "Mars Science Laboratory" succédant au projet Rover, envoi prévu de Curiosity fin 2011.



Spirit



Curiosity

Figure I.6 : Robots mobiles d'exploration extraplanétaire

❖ **Depuis 2000 : Démocratisation des robots** (voir figure I.7)

- * 2000 : Lancement d'Asimo
- * Diversification des compétitions de robotique.
- * Utilisation de drones en situation réelle (Irak...).
- * 2006 : le projet Aibo n'est plus assez rentable.
- * 2009 : robot Nao utilisé à la Robocup Soccer



Figure I.7 : Démocratisations des robots

I.4 Type de robots :

Ils existent plusieurs types de robots. On peut citer les catégories les plus répandues, à savoir les :

1.4.1 Robots mobiles

Robots capables de se déplacer dans un environnement (air, terre, mer...). Ils sont équipés ou non de manipulateurs suivant leur utilisation.

1.4.2 Robots domestiques

Robots utilisés pour des tâches ménagères, par exemple en vaisselle, en repassage, en nettoyage.

1.4.3 Robots collaboratifs

Hommes et robots travaillent ensemble, les robots permettant de diminuer la pénibilité des manipulations manuelles, des efforts ou des mouvements réalisés par l'opérateur.

1.4.4 Robots humanoïdes :

Le terme humanoïde signifie « ressemblant à l'humain ». Il évoque la bipédie, la présence de deux bras et d'une tête

1.4.5 Robots industriels :

Le robot industriel est officiellement défini comme un contrôle automatique, reprogrammable, polyvalent, manipulateur programmable dans trois ou plusieurs axes.

1.4.6 Robots manipulateurs : (objet de notre mémoire)

Un manipulateur est un dispositif utilisé pour manipuler des matériaux sans contact physique direct de l'opérateur. Les applications étaient à l'origine pour traiter des matières radioactives ou bio - dangereuses, en utilisant des bras robotiques, ou elles étaient utilisées dans des endroits inaccessibles. Dans des développements plus récents, ils ont été utilisés dans diverses applications (voir la Figure I.8 et I.9)



FigureI.8 : Bras manipulateur de dernière génération [4]



figure.I.9 : Manipulateur pneumatique [5]

I.5 Architecture d'un robot :

I.5.1 La Base :

Précision du travail La basé ou bâti : le bâti supporte l'ensemble des organes mécaniques de machine, robot. la qualité et la sur une machine est entre autres fonction de rigidité de bâti et la précision des points d'appui et de fixation disposés sur le bâti.

Articulé (segment, articulation, actionneur, organe terminal), il a pour rôle d'amener l'organe terminal dans une situation imposée. Il est constitué de :

I.5.2 Le segment :

Corp rigide solide susceptibles d'être en mouvement par rapport a la base du porter et les uns par rapport aux autres

I.5.3 Articulation :

Assemblage de plusieurs pièces mobiles les unes sur les autres (cardan, charnière, rotule...).

I.5.3.1 Articulation rotoïde :

. Articulation rotoïde Il s'agit d'une articulation de type pivot, notée R, réduisant le mouvement entre deux corps à une rotation autour d'un axe commun. La situation relative entre les deux corps est donnée par l'angle autour

I.5.3.2 Articulation prismatique :

Il s'agit d'une articulation de type glissière, notée P, réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe Fig.1.11. Représentation d'une articulation rotoïde sous ses différentes formes Chapitre 1 Généralités [6].

I.5.4 L'actionneur :

Une Pour être animé, la structure mécanique articulée comporte des moteurs le plus souvent associés à des transmissions (courroies crantées), l'ensemble constitue les actionneurs. Ils utilisent fréquemment des moteurs. On trouve de plus en plus de moteurs à commutation électronique (sans

balais), ou pour de petits robots, des moteurs pas à pas. Pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique), les actionneurs sont le plus souvent hydrauliques, agissant en translation (vérin, hydraulique) ou en rotation (moteur hydraulique). (Les actionneurs pneumatiques sont d'un usage général pour les manipulateurs à cycles (robots tout ou rien)). Un manipulateur à cycles est une structure mécanique articulée avec un nombre limité de degrés de liberté permettant succession de mouvements contrôlés uniquement par des capteurs de fin de course réglables manuellement à la course désirée (asservissement en position difficile dû à la compressibilité de l'air) [7]

I.5.5 Organe terminal :

Désigne tout dispositif destiné soit à manipuler des objets comme les dispositifs de serrage (Pincers à deux ou trois doigts), les dispositifs magnétiques ou à dépression (ventouse), soit à Transformer (outils de découpe, torche de peinture, torche de soudage). Il s'agit d'une Interface permettant au robot d'interagir avec son environnement [8]

I.5.6 Les capteurs :

La perception permet de gérer les relations entre le robot et son environnement. Les Organes de perception sont des capteurs dits proprioceptifs lorsqu'ils mesurent l'état Interne du robot (positions et vitesses des articulations) et extéroceptifs lorsqu'ils Recueillent des informations sur l'environnement (détection de présence, de contact, mesure de distance, (vision artificielle) [9]

I.5.7 La commande

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

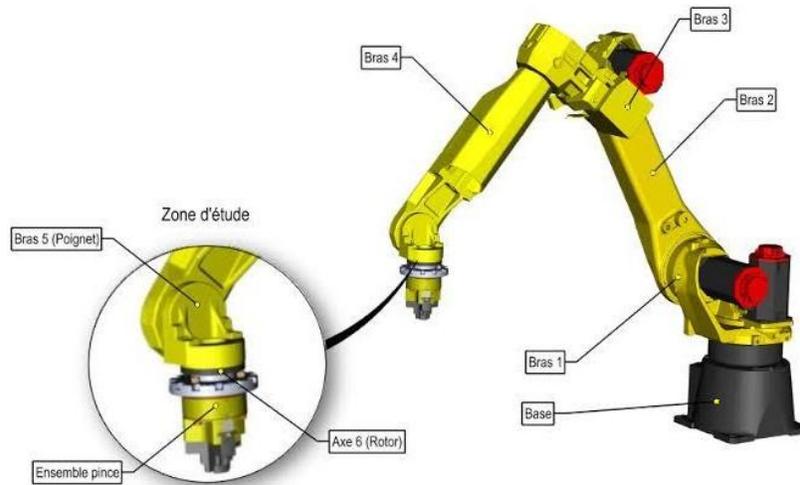


Figure I.10 : Architecture de robot fanuc [10]

I.6 Les caractéristiques techniques du robot :

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

I.6.1 Le volume atteignable (volume de travail)

Le volume de travail en position d'un robot manipulateur est défini comme l'espace physique engendré par un point de l'organe terminal lorsque la configuration du robot évolue. Il s'exprime en unités volumiques, mais la forme de son enveloppe (qui peut être compliquée puisque formée par la combinaison des mouvements de plusieurs articulations) est aussi importante (voir figure I.11). Il est plus simple et plus parlant de la décrire par une surface inscrite au vrai volume mais de forme simple (par exemple intersection de sphères ou des parallélépipèdes ou des ellipsoïdes). Sinon il faut une représentation graphique. Le volume de travail est alors habituellement représenté par deux sections perpendiculaires choisies en fonction du type de robot manipulateur, cette représentation étant préférable à une seule vue en perspective.

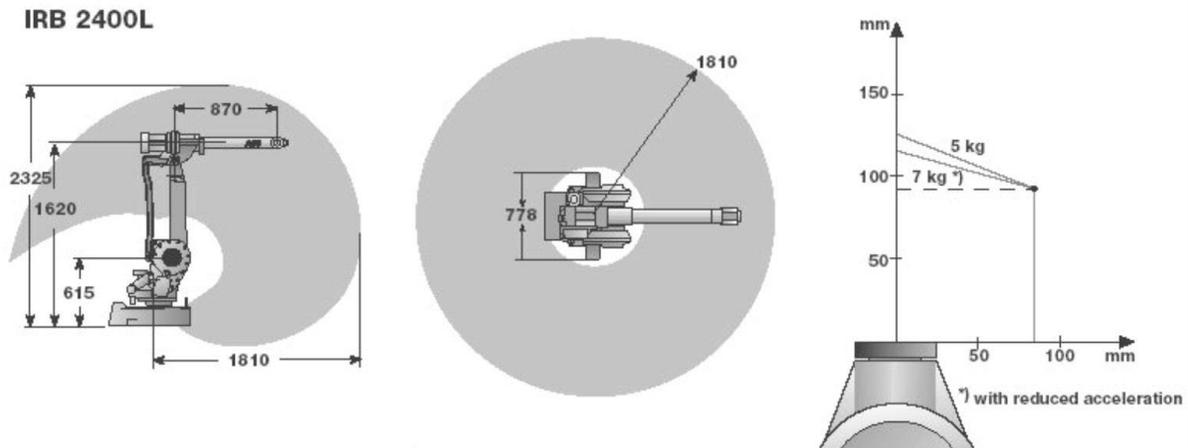


Figure I.11 : Représentation du volume de travail [11]

I.6.2 Charge utile

C'est la charge maximale que peut porter le robot sans dégrader la répétabilité et les performances dynamiques. La charge utile est nettement inférieure à la charge maximale que peut porter le robot qui est directement dépendante des actionneurs.

1.6.3 Précision / Répétabilité

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

1.6.4 La vitesse

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

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

1.6.5 L'orientation possible de l'organe terminal

De par la présence de butées sur chaque articulation, et aussi parfois de par l'existence de couplages mécaniques entre articulations, il n'est pas toujours possible d'atteindre le débattement maximum sur chacun des degrés de mobilité de l'organe terminal dans tout le volume atteignable. Il y aurait donc lieu, soit de donner les débattements angulaires minimaux de l'organe terminal par rapport au repère fixe dans tout le volume atteignable, soit de fournir une cartographie de ces débattements. (Ceci serait plus utile que la présentation des constructeurs, qui donnent le débattement de chaque articulation à partir d'une position de référence, en négligeant les problèmes de butées et de couplages qui modifient quelquefois de façon non négligeable les possibilités du mécanisme).

1.6.6 La fiabilité

Bien qu'elle mette en jeu tous les éléments présents dans le robot, c'est une donnée qui intéresse au plus haut point l'utilisateur, même s'il ignore tout du système. C'est pourquoi on peut la classer dans les performances-tâche. Comme pour tout autre système, elle sera définie par un taux de pannes, celui-ci étant exprimé soit par la fraction du temps durant laquelle le robot ne remplit pas la fonction qui lui est assignée, soit par un MTBF (Mean Time Between Failures). On exprime aussi souvent cette fiabilité à l'aide d'un diagramme fréquentiel (fréquence de pannes) tout au long de la vie du système.

Le robot peut être sujet à deux grandes catégories de pannes :

1. arrêt total du fonctionnement.
2. dégradation des performances.

Par exemple sur la précision spatiale : une dérive plus ou moins lente peut se produire et, après un certain temps, le robot n'exécute plus la tâche. Ou bien un degré de mobilité ne fonctionne plus (il

serait intéressant alors d'avoir une redondance des degrés de mobilité qui permette par une procédure automatique de compenser la défaillance en attendant de pouvoir réparer le robot).

I.7 Classification géométrique des robots :

Il existe différentes architectures du porteur : la structure cartésienne (PPP), la structure cylindrique (RPP ou PRP), la structure sphérique ou polaire (RRP), la structure dite SCARA (RRP) et enfin la structure anthropomorphe (RRR).

I.7.1 Structure cylindrique RPP ou PRP :

Les caractéristiques d'une structure cylindrique (RPP ou PRP) sont : trois axes non perpendiculaires en série avec trois degrés de liberté et une grande rapidité. Le volume de travail est un cylindre plein ou creux, autrement dit un tore à section rectangulaire, dont la hauteur L est la translation permise par une liaison prismatique. (Voir la Figure I.12)

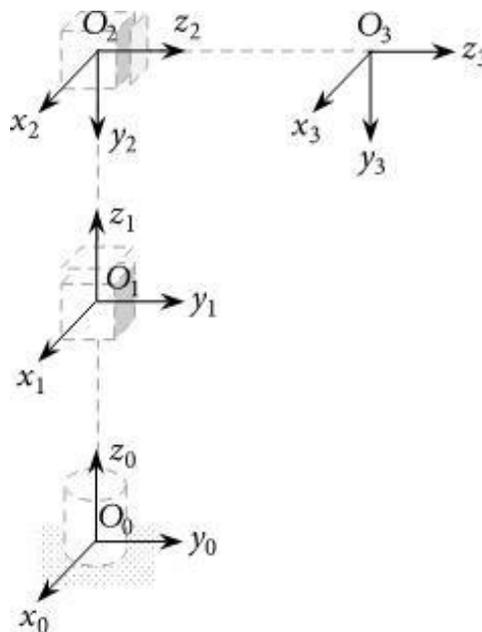


Figure I.12 : Structure cylindrique RPP

I.7.2 Structure cartésienne (PPP) :

Les caractéristiques d'une structure cartésienne (PPP) sont : trois axes deux à deux en série avec trois degrés de liberté, une très bonne précision et une grande lenteur. Le volume de travail est un

parallélépipède dont les dimensions sont les translations permises par les trois liaisons prismatiques (voir **Figure I.13**) [12]

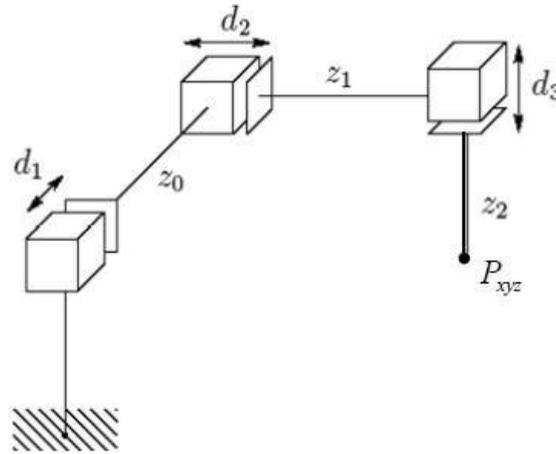
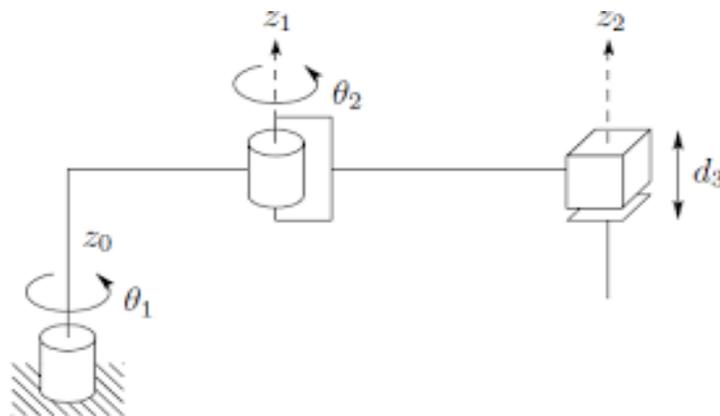


Figure I.13 : Structure cartésienne ppp)

I.7.3 La structure SCARA ou polaire à axe de rotation orthogonale (RRP) :

Le volume de travail est une sphère creuse, dont les rayons intérieur et extérieur sont fixés soit par la disposition de la liaison prismatique et la translation qu'elle permet, soit par les longueurs des deux parties du bras. (voir FigureI.14)



FigureI.14 : Structure RRP

I.7.4 La structure anthropomorphe (RRR) :

Cette structure à une architecture plus généraliste reproduisant le bras humain. Son enveloppe de travail à une cinématique et dynamique complexes, mais sa configuration est plus flexible. (voir la Figure.I.15)

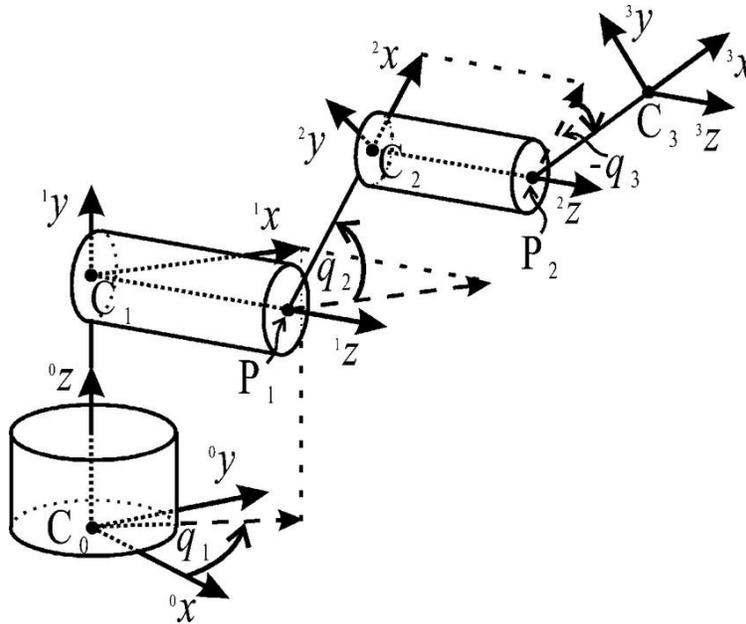


Figure.I.15 : Structure RRR

I.8 Domaine d'application de la robotique :

I.8.1 L'usinage

Le principe de l'usinage est d'enlever de la matière de façon à donner à la pièce brute la forme et les dimensions voulues, à l'aide d'une machine-outil. Par cette technique, on obtient des pièces d'une grande précision.

L'usinage est très utilisé dans l'aéronautique, l'automobile, le ferroviaire, le nucléaire ou encore le prototypage

Le robot est capable d'usiner de nombreux matériaux (vor figure I.16)

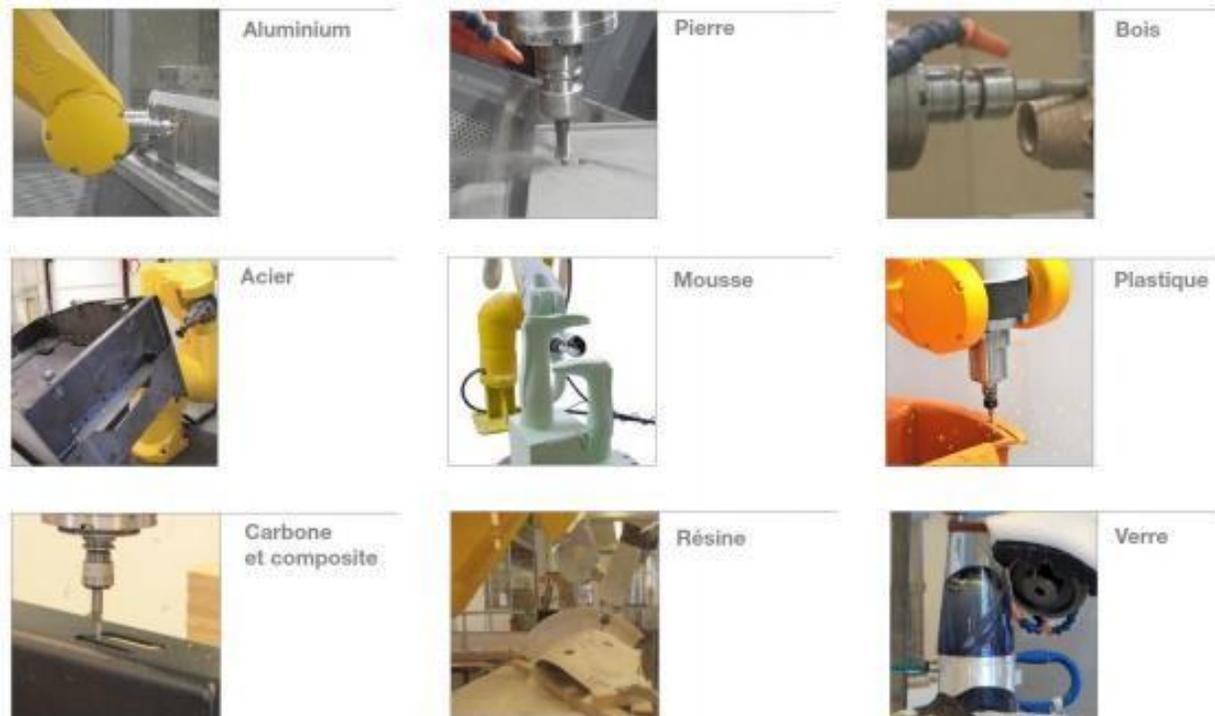


Figure.I.16 : Utilisation de la robotique pour l'usinage de matériaux différents [13]

Lorsque l'on utilise un robot pour usiner, on met en place une broche en bout du sixième axe (voir figure I.17)



Figure.I.17 : Tête d'usinage sur un bras robotisé [13]

I.8.2 Conditionnement/packaging

Conditionner un produit signifie l'emballer. Cela permet de protéger le produit, de faciliter son acheminement et son rangement dans les rayons.

C'est une opération très souvent robotisée (notamment grâce à la flexibilité des robots)

Le packaging/conditionnement regroupe des opérations variées comme la dé-palettisation, l'encaissage, la palettisation, l'emboxage... (voir figure I.18)

Les robots de packaging sont souvent utilisés autour d'une ligne équipée de convoyeurs et de systèmes de vision.

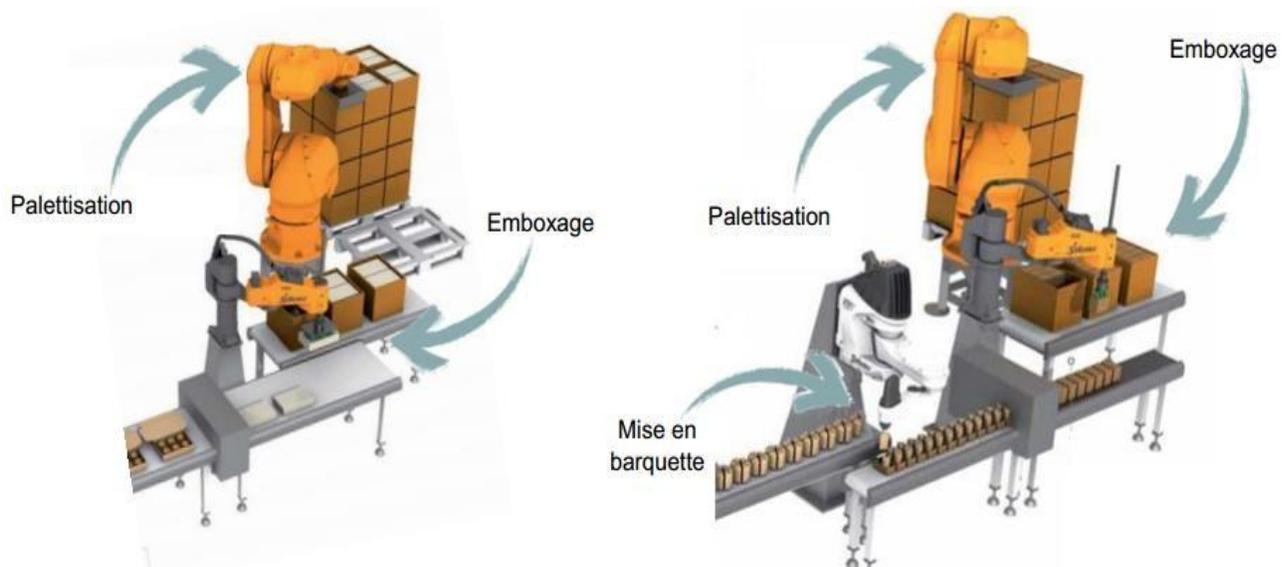


Figure.I.18 : Ligne de packaging avec convoyeurs [13]

1.8.3 L'assemblage

Les opérations d'assemblage nécessitent souvent des mouvements complexes et répétitifs. Les qualités intrinsèques des robots (dextérité, travailler avec une très grande précision de l'ordre de 0,1mm et fiabilité) permettent de réduire les risques de TMS et accroît la qualité des opérations.

Les équipementiers et constructeurs automobiles utilisent largement la robotique pour les applications d'assemblage. (voir figure I.19)



a) Equipementiers automobiles



b) Constructeurs automobiles

Figure.I.19 : Utilisation des bras manipulateurs en industrie automobile [13]

I.8.4 Peinture

Les robots de peinture sont des robots spécifiques leur permettant de fonctionner dans des ambiances explosives et correspondantes aux normes en vigueur. Par exemple, en Europe, les robots répondent à la norme CE ATEX.

D'autre part, utiliser un robot pour des opérations de peinture permet de réduire les quantités de solvant et d'optimiser les épaisseurs projetées, limitant ainsi l'impact sur l'environnement et les couts.

Les robots sont utilisés pour peindre tous types de pièces

Lorsque l'on utilise un robot de peinture, on met en place un pistolet de peinture pneumatique au bout du sixième axe. (voir figure I.20)

**Figure.I.20** : Pistolet de peinture sur un bras robotisé [13]

I.8.5 Sciences de la vie

Les robots utilisés en science de la vie sont des robots aux caractéristiques spécifiques leur permettant de fonctionner dans des environnements gris, blancs ou stériles et correspondant aux normes en vigueur : ISO 14644-1.

Utiliser des robots dans ce domaine permet plus de sécurité envers le personnel, de sécuriser les préparations ainsi que de diminuer les coûts de maintenance.

Les robots sont utilisés pour diverses applications. (voir figure I.21)



Figure.I.21 : Robotique en science de la vie

I.8.6 L'agroalimentaire

Les robots utilisés dans l'industrie agroalimentaire sont des robots aux caractéristiques spécifiques leur permettant d'être nettoyés tous les jours avec des produits agressifs et correspondant aux normes en vigueur : la directive machine 1935.2004 (révisé en 2006) et son volet hygiène

Utiliser des robots dans ce domaine permet de faciliter et réduire les coûts de processus de nettoyage (amortissement en moins d'un an)

La figure I.22 montre un exemple de la robotisation de l'usine de chocolat.

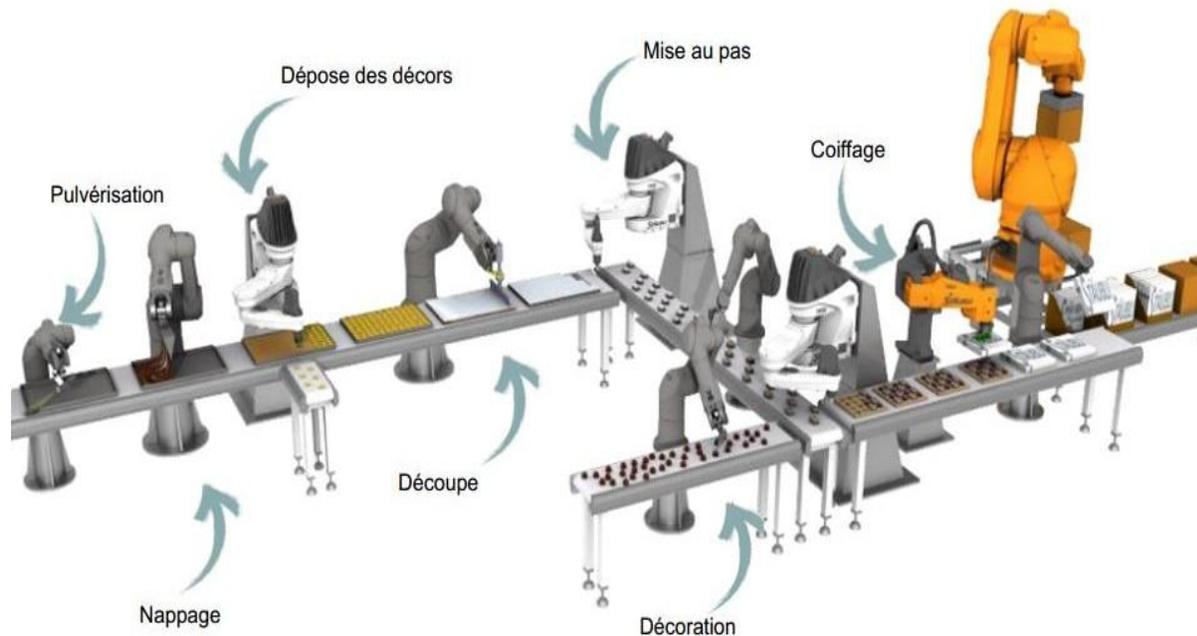


Figure.I.22 : Robotisation d'une chaîne de production du chocolat [13]

I.9 Conclusion :

Dans ce chapitre, nous avons donné un aperçu général sur la robotique : l'historique des robots, leurs structures, leurs utilisations, leurs différents types, leurs classifications ainsi que leurs domaines d'application ce qui va faciliter la compréhension de la suite de notre mémoire.

Dans notre mémoire nous allons nous focaliser sur les bras manipulateurs de type RRR qui fera l'objet du reste de notre mémoire.

CHAPITRE II :

Étude du Bras

manipulateur

II.1. Introduction

Dans ce chapitre, nous allons présenter nos choix sur la structure du bras manipulateur ainsi que sur la réalisation de celui-ci. La partie commande du bras manipulateur est présentée à savoir la carte programmable Arduino ainsi que les parties capteurs et actionneurs.

Pour finir, nous allons détailler la fabrication des pièces du bras manipulateur par impression 3D pour donner une démarche précise de réalisation du robot étudié dans notre mémoire.

II.2. Choix du bras manipulateur étudié :

De nos jours, la robotique c'est généralisé dans les pays développés comme nous l'avons vu dans le chapitre précédent. Cette généralisation a rendu l'acquisition d'un robot moins onéreuse que par le passé et ainsi éviter de partir de zéro comme dans notre cas. Plus précisément, ils existent des bras manipulateurs sous forme de squelette (en pièces détachées) et une carte de programmation vide que nous pouvons programmer à notre guise pour effectuer les tâches qui nous intéressent. (voir figure II.1)

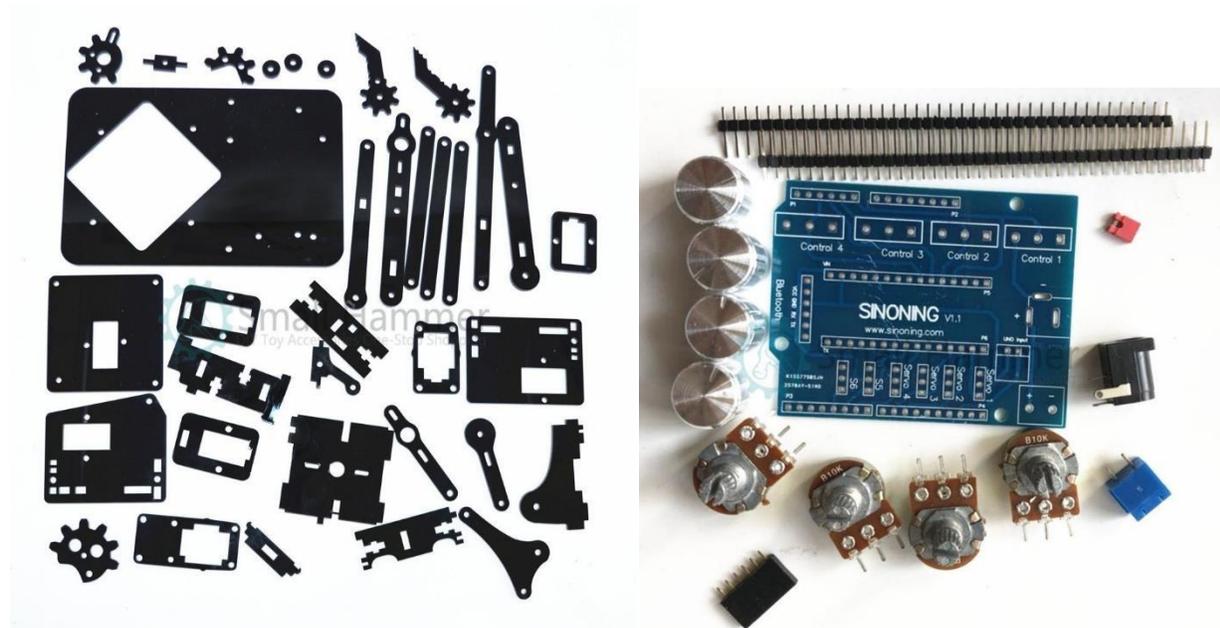


Figure II.1 : Pièces constituant le bras manipulateur SNAM1900 [ref]

Afin d'atteindre les objectifs de notre mémoire, nous allons étudier la structure du bras manipulateur en deux étapes :

1. La première étape concerne la modélisation du bras manipulateur avec le logiciel de CAO « Solidworks » (voir Annexe A), décrire les différentes parties de sa structure mécanique, décrire la partie commande avec la carte programmable Arduino, les servomoteurs et la carte « Shield ». Et pour finir, nous allons détailler l'impression 3D des pièces de ceux bras manipulateurs dans le cas où d'autres personnes veulent reproduire notre travail.
2. La deuxième étape c'est l'étude du mouvement ou l'étude cinématique du bras manipulateur que nous allons voir dans le chapitre suivant, qui décrit le modèle mathématique qui permet de faire le lien entre les paramètres articulaires du robot avec la position et l'orientation de son organe terminal dans le repère de base. Cela permet aussi de déterminer l'espace de travail du robot afin de déterminer certaines caractéristiques du robot.

II.3. Description de l'architecture du bras manipulateur

II.3.1. Présentation du bras manipulateur :

Un **bras manipulateur** est le bras d'un robot généralement programmable, avec des fonctions similaires à un bras humain. Les liens de ce manipulateur sont reliés par des axes permettant, soit de mouvement de rotation (comme dans un robot articulé) et/ou de translation (linéaire) de déplacement (figure II.2). [14]

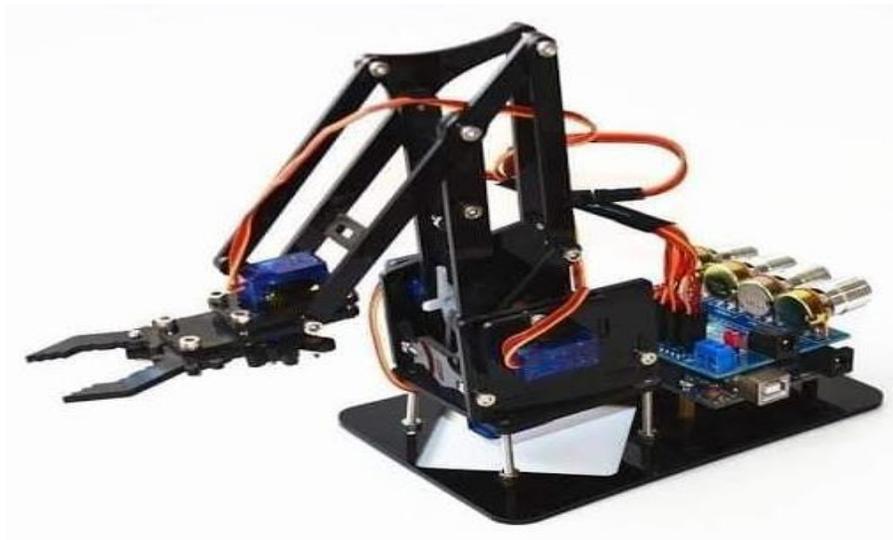


Figure II.2: Bras manipulateur

II.3.2. Définition de la base :

C'est l'organe fixe du bras manipulateur qui porte tous les organes et la charge sans tomber. C'est un assemblage vis-écrou de deux pièces en plastique par quatre boulons et un servomoteur de 180° de rotation (voir figure II.3).

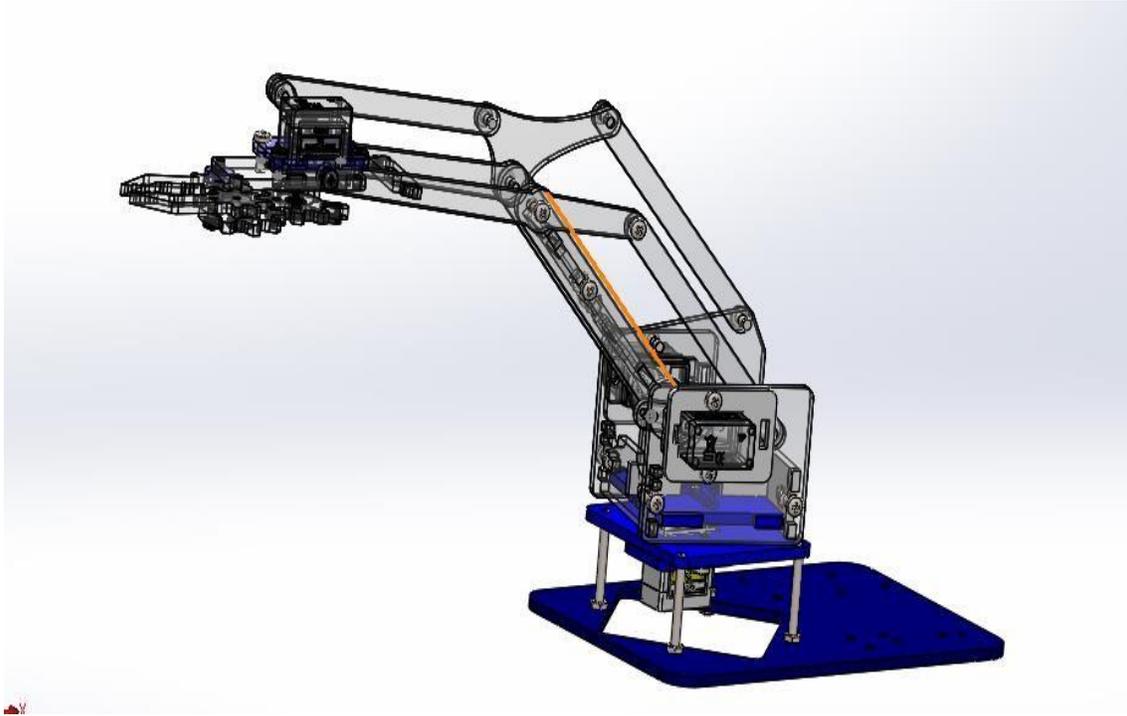
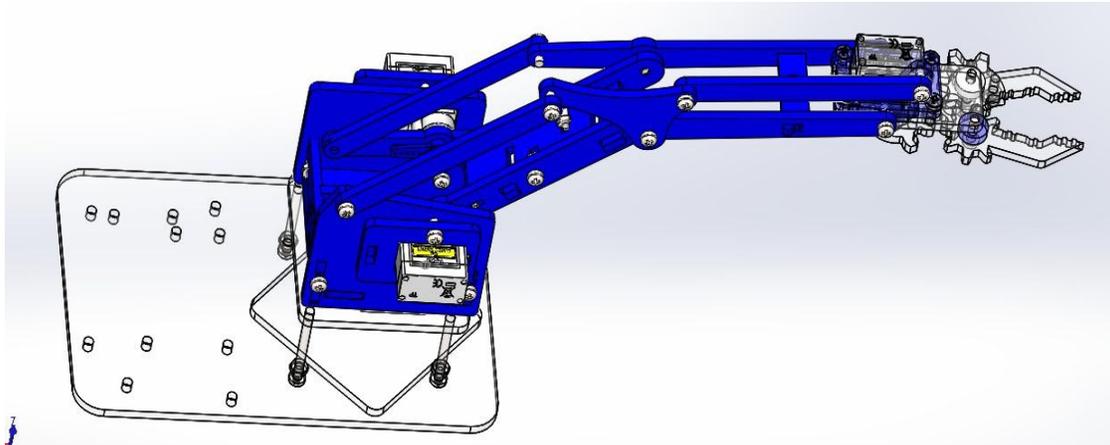


Figure II.3 : la base d'un Bra manipulateur

II.3.3. Définition le porteur :

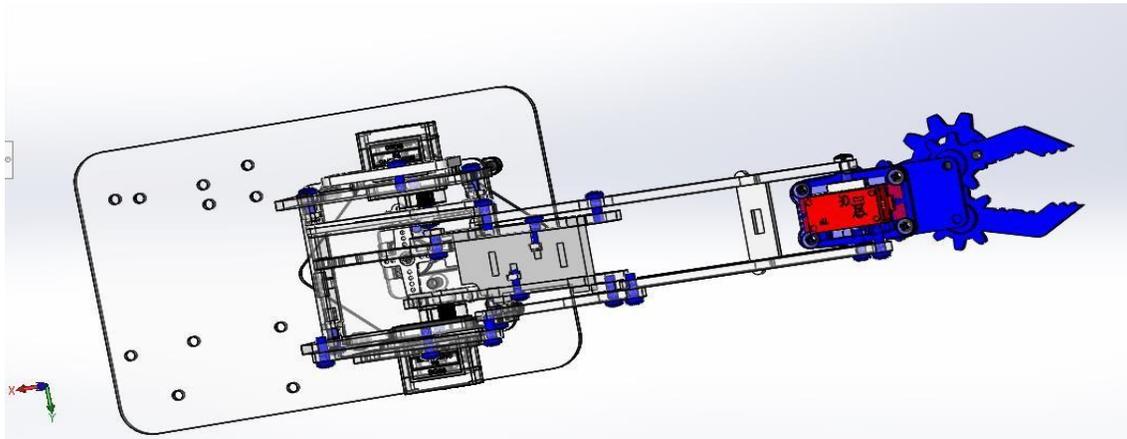
Le porteur représente l'essentiel du système mécanique articulé (segment, articulation, actionneur, organe terminal), il a pour rôle d'amener l'organe terminal dans une situation imposée. Il est constitué de : segment et articulation (figure II.4) [15]



FigureII.4 : le porteur d'un Bras manipulateur

II.3.4. Définition de la pince :

La pince est l'organe terminal de notre robot. En effet, elle permet de saisir les différents objets qu'on lui présente. La pince est constituée d'un servomoteur qui contrôle l'ouverture ou la fermeture de celle-ci grâce à un mécanisme d'engrenage (figure II.5). [16]



FigureII.5 : le pince d'un bras manipulateur

II.3.5. Définition de servomoteur :

Un **servomoteur** (souvent abrégé en « servo », provenant du latin servus qui signifie « esclave ») est un moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure. C'est donc un système asservi. Le servomoteur intègre dans un même boîtier, la mécanique (moteur et engrenage), et l'électronique, pour la commande et l'asservissement du moteur. La position est définie avec une limite de

débattement d'angle de 180 degrés, mais également disponible en rotation continu (figure II.6).[17]

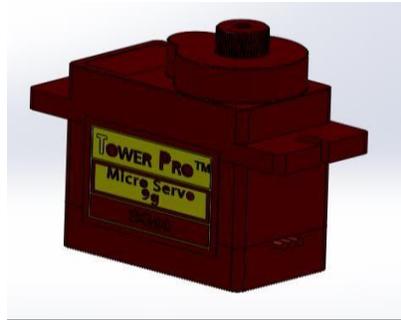


Figure II.6 : un servomoteur

II.4. Description des autres parties constitutives du bras manipulateur :

Dans cette partie, on va parler des types d'actionneurs et capteurs que nous avons utilisés dans notre bras manipulateur

II.4.1. Les actionneurs du bras manipulateur :

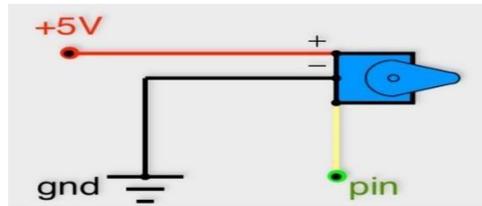
Le rôle des actionneurs est d'exécuter une tâche. Ils permettent de fournir assez de force pour pouvoir provoquer un mouvement. On a choisi d'utiliser comme actionneurs quatre servomoteurs qui sont des moteurs à courant continu (MCC), asservis en position à l'aide d'un capteur de position et d'un circuit électronique interne au moteur. [18]

II.4.2. Les servomoteurs :

Est un moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure. C'est donc un système asservi.

Le terme servomoteur désigne également de manière abusive le pré-actionneur, la plupart du temps pneumatique (système membrane/ressort), plus rarement hydraulique où électrique d'une vanne de régulation. Dans ce cas l'asservissement de position est toujours effectué par un organe distinct nommé positionneur.

Les servomoteurs sont pilotés par un fil de commande (jaune) et alimentés par deux autres fils : le premier est relié à l'alimentation positive +5V ou +6V (rouge), le deuxième est relié à la masse (noir) (figure II.7). [19]



FigureII.7 : Branchement électrique de servomoteur

II.4.3. Les types de servomoteur :

Il y a 2 types 2 servomoteur

- Servomoteur du type MG90S
- Servomoteur du Type SG90

	Mg90s	Sg90
Les caractéristiques		
Poids	14g	9 g
Vitesse	<ul style="list-style-type: none"> • 4.8V: 0.11 sec/60° • 6.0V: 0.10 sec/60° 	0,1s/60°
Dimensions	23,2 × 12,5 × 22mm	32 x 12 x 32 mm
Alimentation	4,8 à 6 v	4,8 à 6 v
Angle de rotation	180°	180°
Couple d'arrêt	2,5 kgf.cm	1,8 kgf.cm (4,8V), 2,2 kgf.cm (6V)

Tableau 1 : Les paramètres de chaque servomoteur :

II.5. Les capteurs :

Les capteurs actifs introduisent souvent un son ou une lumière et recherchent comment l'énergie introduite réagit avec l'environnement. Certains exemples de ce type de détection sont les détecteurs réfléchissants sonar, laser et infrarouge. Les détecteurs de sonar introduisent un son, généralement plus élevé en fréquence que ce que les humains peuvent entendre, et écoutent l'écho. L'écho renvoyé est utilisé par les télémètres sonar pour envoyer une impulsion sonore, puis calculer le temps nécessaire au retour du son. Ce temps est directement proportionnel à la distance que le son doit parcourir pour rebondir sur l'objet le plus proche et revenir. La vitesse du son est d'environ 770 mi / h, mais elle peut facilement être mesurée à l'aide d'une puce de microcontrôleur moyennement rapide comme celles que l'on trouve dans de nombreux robots. Les systèmes de capteurs réfléchissants infrarouges (IR) émettent une longueur d'onde spécifique de la lumière et recherchent une réflexion de la lumière. Étant donné que la lumière voyage beaucoup plus vite que le son, il est difficile de mesurer le temps nécessaire pour recevoir la lumière réfléchi. Les détecteurs infrarouges sont généralement utilisés pour détecter si un objet est présent dans la portée du détecteur plutôt que pour savoir à quelle distance l'objet est du détecteur. Certains détecteurs infrarouges intelligents utilisent une géométrie simple présente dans un triangle formé par l'émetteur qui génère la lumière, l'objet réfléchi et le détecteur qui détecte la lumière émise. [20]

II.5.1 Les types de capteurs :

II.5.1.1. Capteurs :

Le contrôleur doit savoir où se trouve le chemin par rapport au véhicule. Ces informations de localisation sont fournies par des capteurs sur le véhicule. Différents capteurs sont disponibles pour effectuer cette tâche et leur précision et leur facilité de mise en œuvre varient. En outre, certains types de capteurs nécessitent des modifications des routes elles-mêmes tandis que d'autres peuvent être utilisés sur les routes existantes. [21]

II.5.1.2. Caméras :

De nombreuses recherches ont été consacrées à l'utilisation des caméras dans les véhicules autonomes. La caméra est utilisée pour prendre des images de la chaussée devant le véhicule. Un traitement d'image est ensuite effectué pour extraire des informations de l'image concernant l'emplacement de la voiture sur la route. Ce type de détection ressemble le plus à celui utilisé par

les conducteurs humains. La caméra voit devant et le contrôleur peut ajuster la direction en fonction de la courbe de la route. Patricia Mel lodge [21]

II.5.1.3. Capteurs infrarouges :

Des capteurs infrarouges ont été utilisés pour détecter les lignes blanches sur les chaussées sombres. La lumière infrarouge est émise par les LED sous la voiture. La lumière est réfléchiée par la ligne blanche et absorbée par le pavé sombre. Les capteurs détectent la lumière réfléchiée et ainsi l'emplacement de la ligne blanche est connu. Cette méthode suppose que la voiture doit suivre la ligne blanche. [21]

II.5.1.4. Capteurs magnétiques :

Les capteurs magnétiques fonctionnent en détectant la présence d'un champ magnétique. Des capteurs sous la voiture détectent les aimants intégrés dans la chaussée. Cette méthode est très similaire à l'utilisation de capteurs infrarouges mais nécessite des modifications majeures de l'infrastructure car la plupart des routes ne comportent pas d'aimants intégrés. [21]

II.5.1.4. Radar :

L'utilisation du radar suit le même principe que la technologie infrarouge, sauf qu'un type d'énergie différent est utilisé. Un signal RF est émis vers la route et il peut être redirigé par une bande réfléchissante. Ainsi, des informations de position sont fournies au contrôleur. Chacun des types de capteurs présente des avantages et des inconvénients. [21]

II.6. La carte programmable ARDUINO UNO :

Le système Arduino est une carte électronique basée autour d'un microcontrôleur et de composants minimum pour réaliser des fonctions plus ou moins évoluées à bas coût. Elle possède une interface USB pour la programmer. C'est une plateforme open-source qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

Arduino peut être utilisé pour développer des applications matérielles industrielles légères ou des objets interactifs (créations artistiques par exemple), et peut recevoir en entrées une très grande variété de capteurs. Arduino peut aussi contrôler une grande variété d'actionneurs (lumières, moteurs ou toutes autres sorties matériels). Les projets Arduino peuvent être autonomes, ou

communiquer avec des logiciels sur un ordinateur (Flash, Processing ou MaxMSP). Les cartes électroniques peuvent être fabriquées manuellement ou bien être achetées préassemblées [22]

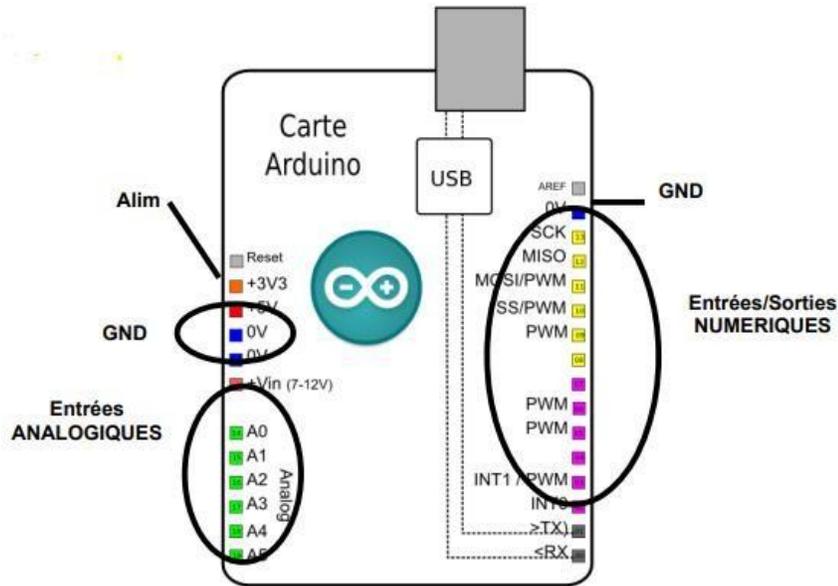
Alimentation	-Via port USB -7 à 12 V sur connecteur alim 5.5 x 2.1 mm
Microprocesseur	-ATMega 328 -Mémoire flash : 32 KB -Mémoire SRAM : 2 KB -mémoire EEPROM : 1 KB
Interfaces	-14 broche d'E/S dont 6 PWM -6 entrées analogiques 10 bits -Bus série. I2C et SPI
Intensité par E/S	-40 MA
Cadencement	--16 MHz Gestion des interruptions fiche USB B
Version	-Rev . 3
Dimensions	-74 x 53 x 15 mm
Référence	-Arduino uno Rev3 A000066 module prêt à l'emploi -Livré avec support en plastique Version d'origine conçue et assemblée en Italie

Tableau 2 : Les caractéristiques principales de l'Arduino :

II.6.1 Brochage de la carte UNO :

Nous montrons les pins de la carte arduino uno comme suis :

- Les entré et sortie numérique : (sck, miso, PWM, SS/PWM) , GND
- Alimentation +3V3, GND(OV)
- Les entrées analogique (A0,A1,A2,A3,A4,A5)



FigureII.8 : brochage de la carte UNO [23]

II.6.2 L'utilisation de la carte Arduino UNO :

La carte Arduino se programme avec le langage Arduino créé à partir des langages C et C++. On ne programme pas au collège sur l'IDE Arduino avec des lignes de code, mais nous le ferons en EPI ...

En mode découverte, on les fait fonctionner à partir du logiciel SCRATCH que vous connaissez déjà : Principe :

La carte Arduino est déjà programmée par le professeur. Une application appelée S2A (Scratch to Arduino) utilise ce programme appelé « firmware » pour établir une communication entre l'ordinateur et la carte via SCRATCH. La carte devient une simple interface entre l'ordinateur, et les composants branchés sur la carte. Ce système est agréable, car on teste « en direct » le

programme sur la carte. Mais ce système est limité, car il ne permet pas la programmation réelle de la carte : on faut toujours rester connecté à l'ordinateur (voir figure II.19). [24]



FigureII.9 : Arduino UNO

II.6.3 Utilisation d'un servomoteur avec l'Arduino :

Pour commander un servomoteur, il faut lui envoyer un train d'impulsions dont la période (intervalle de temps entre chaque impulsion) est toujours de 20 ms (millisecondes). Ce qui va varier et qui finalement déterminera la position du bras n'est pas la période (figureII.11), mais bien la durée de l'impulsion :

- Pour une impulsion de 1 ms, le servomoteur se met en position 0° ;
- Pour une impulsion de 1,5 ms, le servomoteur se met en position 90° ;
- Pour une impulsion de 2 ms, le servomoteur se met en position 180° . [25]

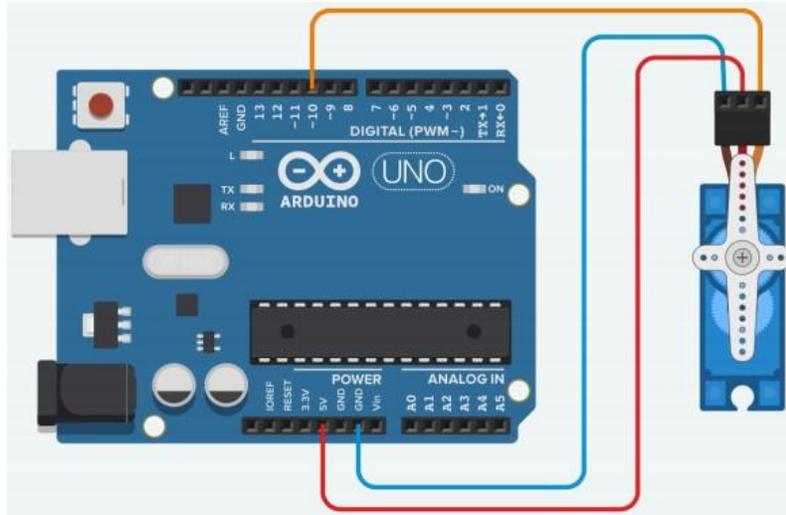


Figure II.10: le servomoteur avec l'Arduino

II.7. Definition de Bluetooth HC5:

Le module Bluetooth HC-05 est un module Bluetooth SPP (protocole de port série) facile à utiliser, conçu pour la configuration d'une connexion série sans fil transparente. Sa communication s'effectue via une communication série, ce qui facilite l'interface avec le contrôleur ou le PC. Le module Bluetooth HC-05 fournit un mode de commutation entre les modes maître et esclave, ce qui signifie qu'il ne peut utiliser ni la réception ni la transmission de données (figure II.12). [26]

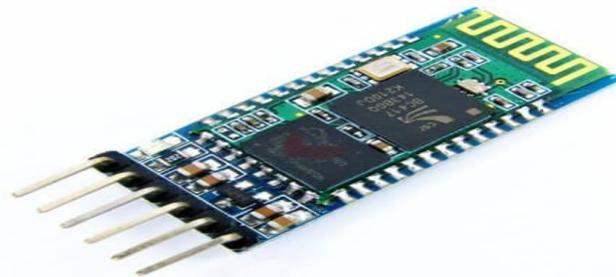


Figure II.11 : Bluetooth HC5

II.7.1 Câblage de Arduino avec Bluetooth HC5 :

Votre carte Arduino peut communiquer avec d'autres appareils (Smartphone, ordinateur ou un autre microcontrôleur) par liaison Bluetooth en utilisant un module HC-05 (maître/esclave). Cela vous permet de connecter la carte Arduino à un autre système afin d'envoyer et de recevoir des données. La communication par Bluetooth permet, par exemple, de piloter votre robot via une

application Smartphone, récupérer les mesures d'un capteur sur votre ordinateur ou encore permettre à deux robots de communiquer entre eux.[27]

II.7.2 Matériel utilise pour câblage :

Matériel Ordinateur Arduino UNO x2 câble USB pour connecter l'Arduino à l'ordinateur x2
Module Bluetooth HC-05 ZS-040 x2 (ou 1x HC-06 et 1x HC-05) Câbles Dupont M/F [28]

II.7.3. Présentation de Bluetooth HC 5 :

Le module Bluetooth HC-05 présente 6 broches pour permettre d'établir la connexion. VCC broche d'alimentation. Typiquement connectée à la broche 5V de l'Arduino. GND masse. Typiquement connectée à la broche GND de l'Arduino RX broche de réception. Typiquement connecté à la broche de transmission (TX) de l'Arduino TX broche de transmission. Typiquement connecté à la broche de réception (RX) de l'Arduino State retourne 1 lorsque le module est connecté Key ou EN doit être alimentée pour entrer dans le mode de configuration et ne doit pas être connecté pour être en mode communication. La particularité du module Bluetooth HC-05 est qu'il peut être utilisé en mode esclave (dans ce cas il est équivalent à un module HC-06 et s'utilise de la même manière) ou en mode maître, ce qui veut dire qu'il peut de manière autonome se connecter à un autre module Bluetooth (HC-06, HC-05 en mode esclave ou autre) sans action extérieure. Schéma de câblage pour la configuration (figure II.12) [29]

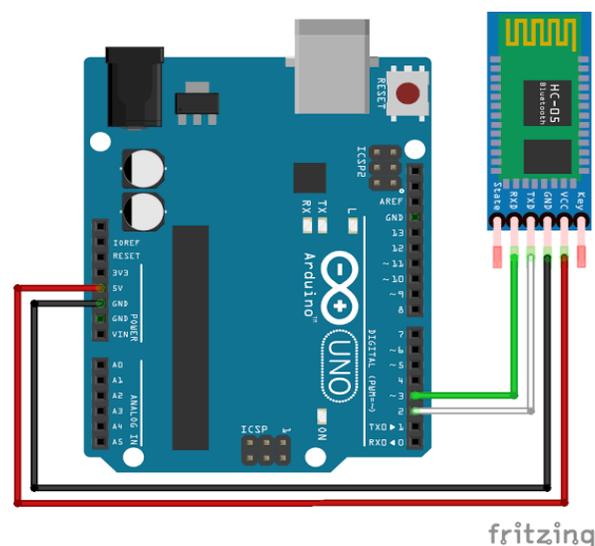


Figure II.12 : l'Arduino avec Bluetooth HC5

II.7.4. Le microcontrôleur :

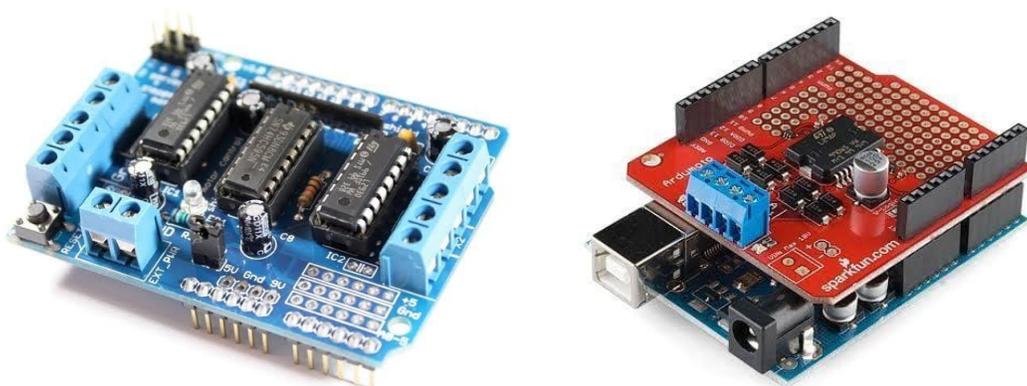
Un microcontrôleur est un circuit intégré rassemblant dans un même boîtier un microprocesseur, plusieurs types de mémoires et des périphériques de communication (Entrées-Sorties) .Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible (de quelques mégahertz jusqu'à plus d'un giga hertz et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels(figureII.13). [30]



FigureII.13 : Microcontrôleur

II.7.5 Shield Arduino :

Les Shields Arduino sont des cartes qui se branchent sans soudeure aux cartes Arduino ou à d'autres Shields Arduino pour augmenter leurs capacités (Wi-Fi, 4G, écran, bluetooth...). Les Shields Arduino conservent l'esprit original de Arduino, facile à produire et à utiliser(figureII.14).[31]



FigureII.14 : les Shields Arduino

II.8. Réalisation des pièces par impression 3D :

Afin de fabriquer les différentes pièces en plastique de notre bras manipulateur plusieurs choix s'offrent à nous. Parmi ces choix on peut citer la découpe laser, l'usinage par commande numérique ou l'impression 3D qui commence à se généraliser.

Notre choix c'est porté sur cette dernière technologie qui permet facilement de passer d'une pièce 3D à une pièce fabriquée en impression 3D. Nous allons prendre une pièce de notre assemblage du bras manipulateur pour montrer comment nous procédons à l'impression 3D d'une pièce réalisée avec le logiciel SolidWorks (voir figure II.15).

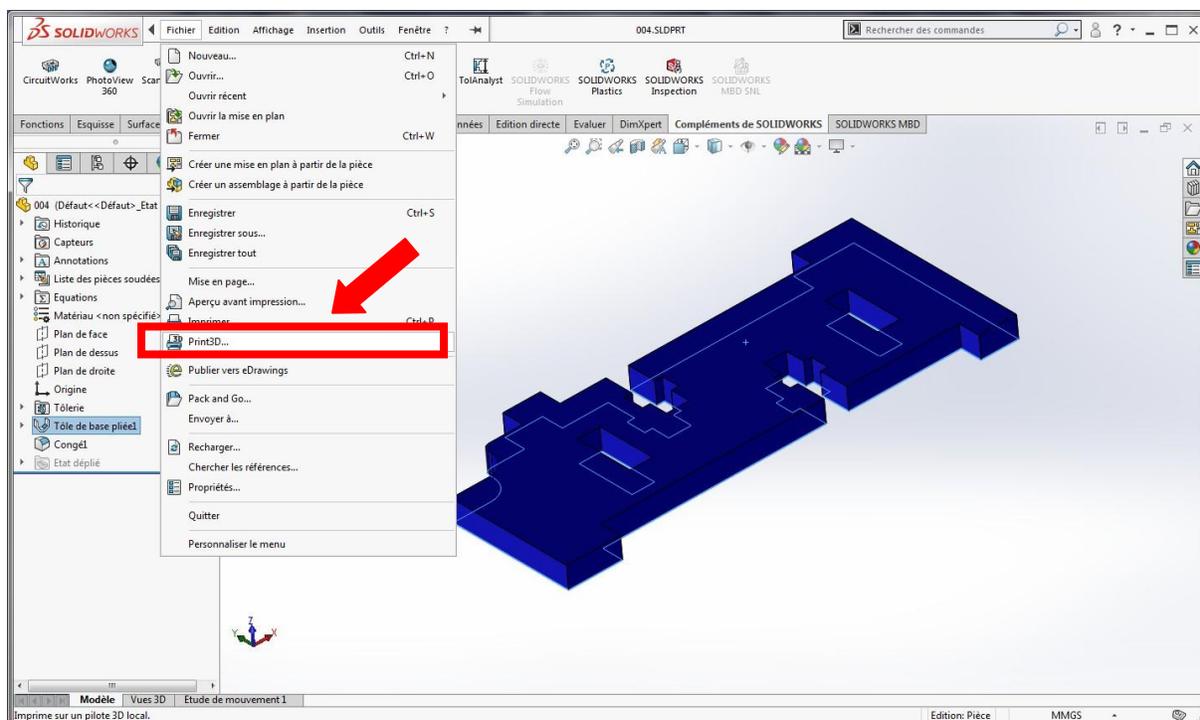


Figure II.15. Utilisation de la fonction Print3D du logiciel SolidWorks avec notre pièce

La figure II.16 nous montre l'interface du panneau de configuration de la fonction Print3D qui permet de choisir le type d'imprimante ainsi que l'espace d'impression, de positionner la pièce dans cet espace, de faire des réglages liés à la précision d'impression et pour finir il permet de calculer les surfaces qui nécessitent un support lors de l'impression (pour les surfaces en surplomb).

Une fois les choix des paramètres effectués, nous sauvegardons la pièce sous le format adéquat pour l'impression 3D à savoir : STL (STereo-Lithography), 3D manufacturing format (3mf), Additive manufacturing file (amf). Dans notre cas c'est le format STL (voir figure II.17).

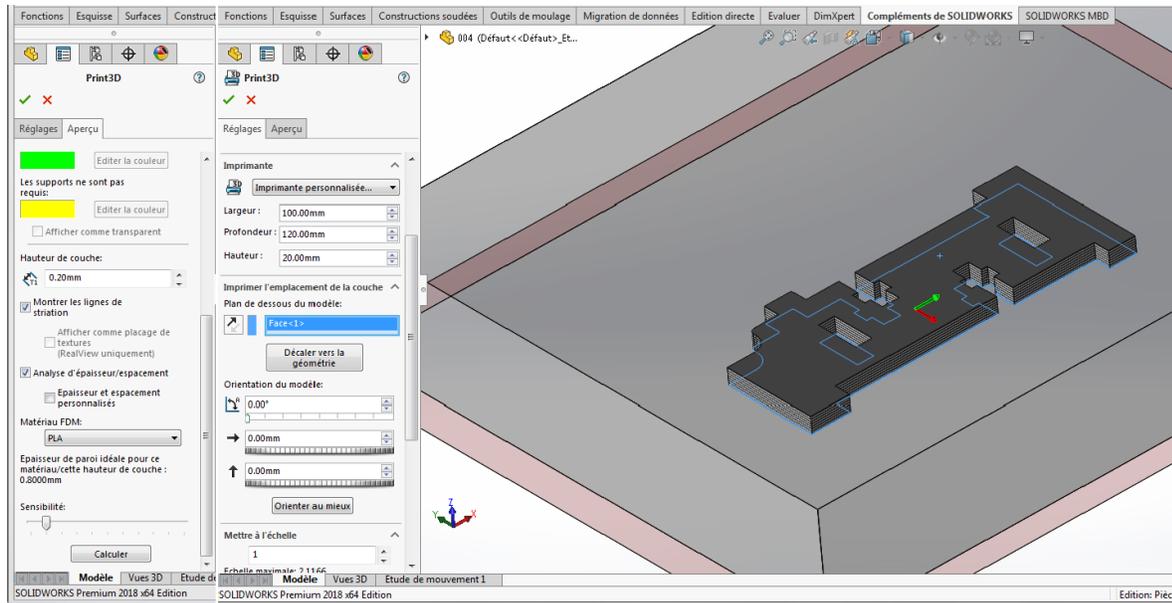


Figure II.16. Choix des paramètres de la fonction Print3D

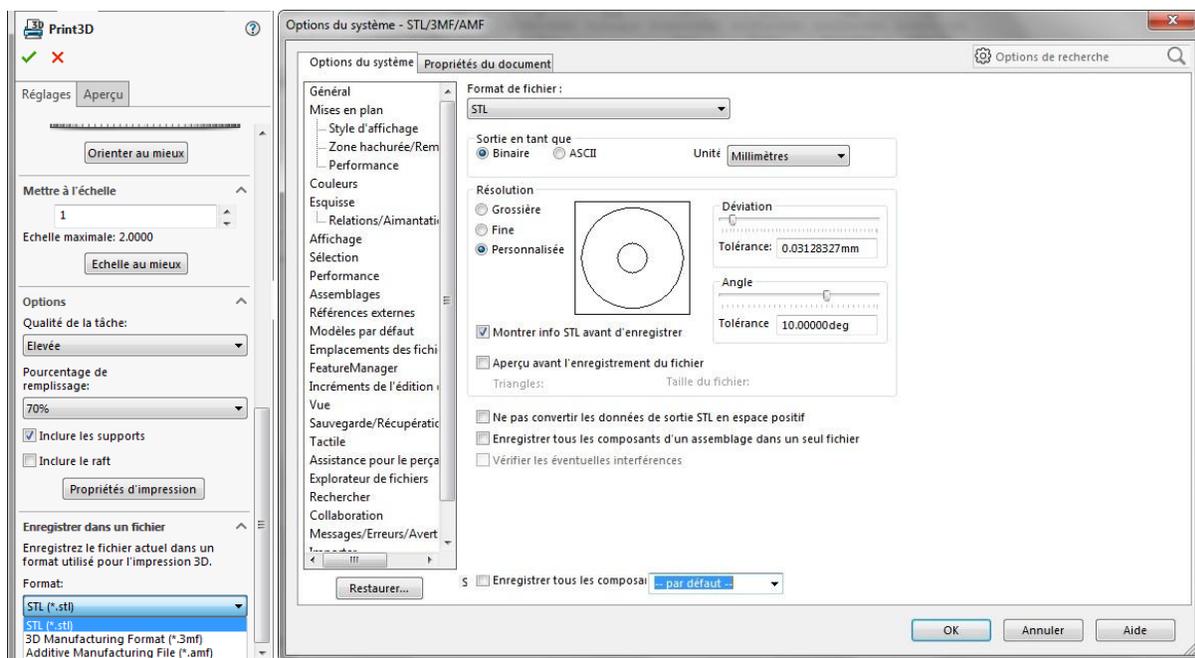


Figure II.17. Exportation de la préparation à l'impression de la pièce en format STL

Une fois le fichier STL de la pièce générée, il faudra choisir le logiciel de l'imprimante 3D qui va exécuter la tâche d'impression. Ils existent une multitude d'imprimantes 3D et de logiciels associés. Pour montrer la faisabilité de notre travail, nous avons choisis d'utiliser le logiciel des imprimantes PRUSA (voir figure II.18).

La figure II.19 nous montre notre pièce au format STL une fois importée avec le logiciel Prusa Slicer. La pièce est positionnée dans le plan d'impression comme elle le sera après impression.

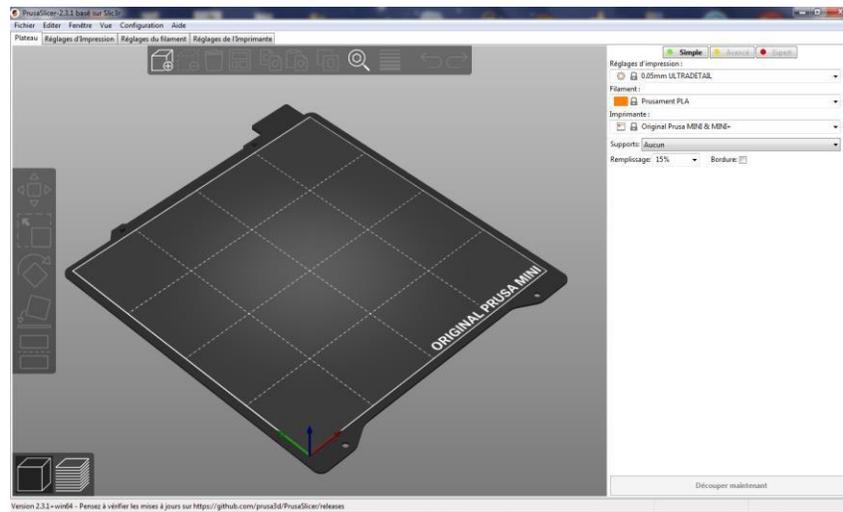


Figure II.18. Interface du logiciel Prusa Slicer de l'imprimante 3D Prusa

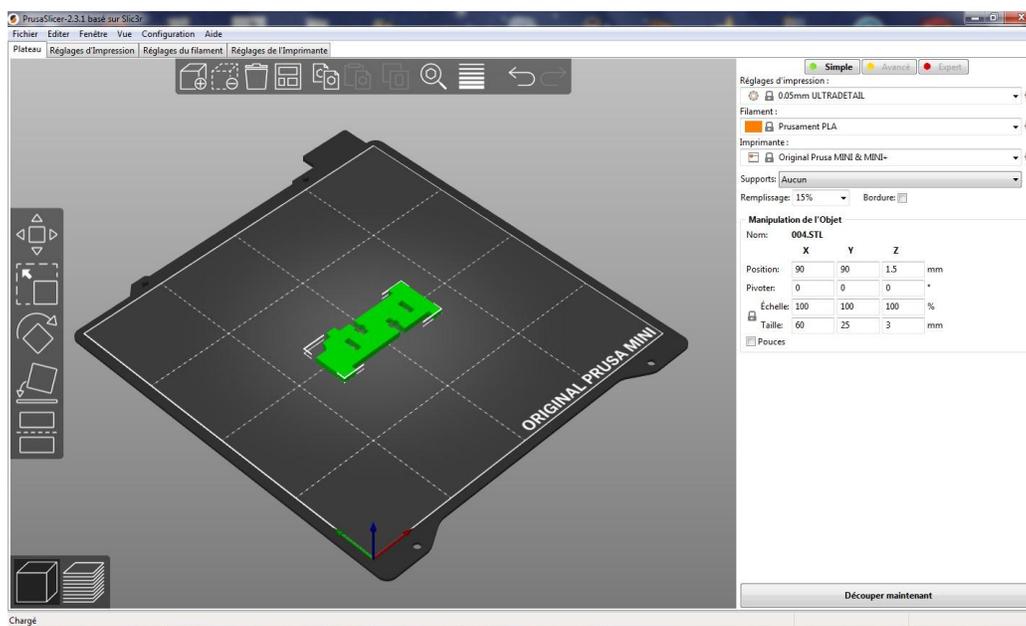


Figure II.19. Importation du fichier STL de la pièce dans le logiciel de l'imprimante 3D Prusa

Une fois les paramètres d'impression choisis, à savoir le type de matériau avec lequel la pièce sera imprimée (PLA pour acide poly lactique un bioplastique), le pourcentage de remplissage, l'existence de support, etc. le logiciel nous permet de découper (Slicer) la pièce suivant les paramètres adaptés à l'imprimante Prusa et de générer le fichier G-code associé pour l'exécuter (voir figure II.20).

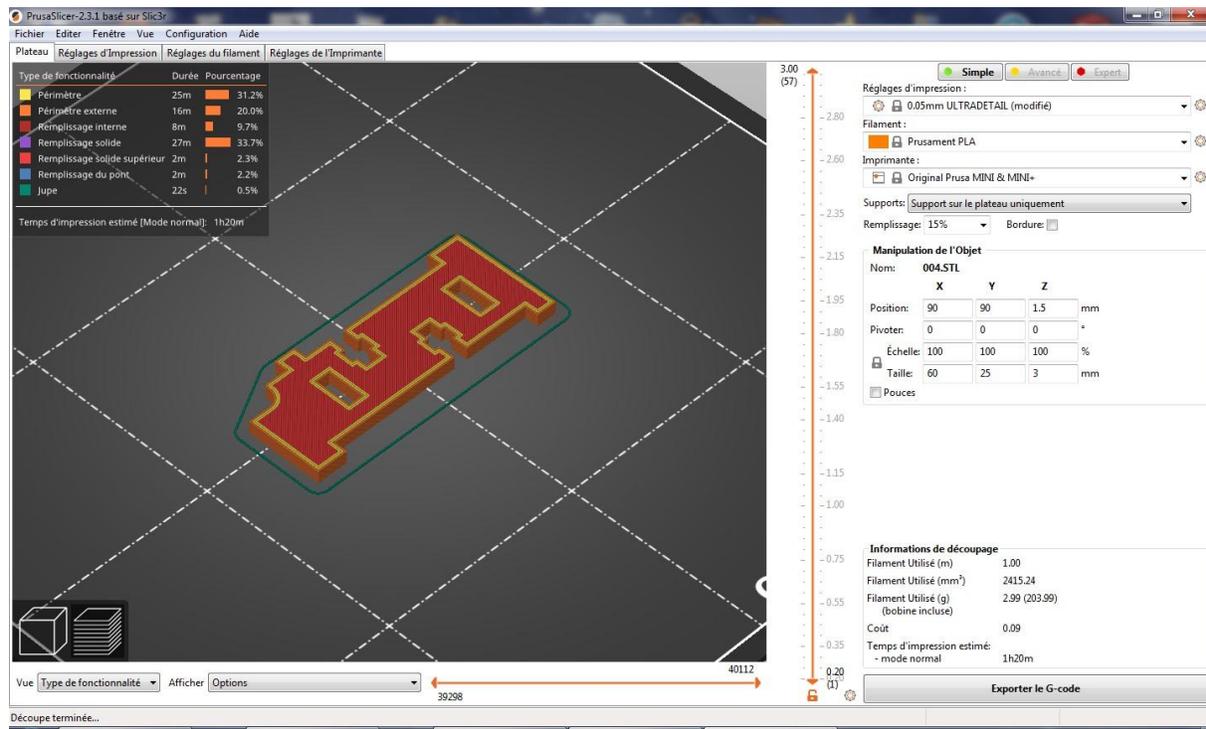


Figure II.20. Génération des couches d'impression de la pièce ainsi que le G-Code associé

Afin de vérifier le résultat obtenu avant impression, le logiciel Prusa Slicer nous permet de voir comme sur la figure II.21 des coupes de la pièce pour voir le remplissage et d'effectuer une simulation d'impression de la pièce comme nous pouvons le voir dans la figure II.21.

Avant de finir cette partie, nous voulons porter l'attention sur le temps nécessaire pour l'impression qui est donné après simulation et qui est de l'ordre de 1h20 pour cette pièce. Ce temps est très important vu le nombre de pièces nécessaires dans notre bras manipulateur. Pour le réduire, il va falloir exploiter l'espace d'impression d'une manière optimale pour imprimer le maximum de pièces à la fois et ainsi réduire le temps global d'impression.

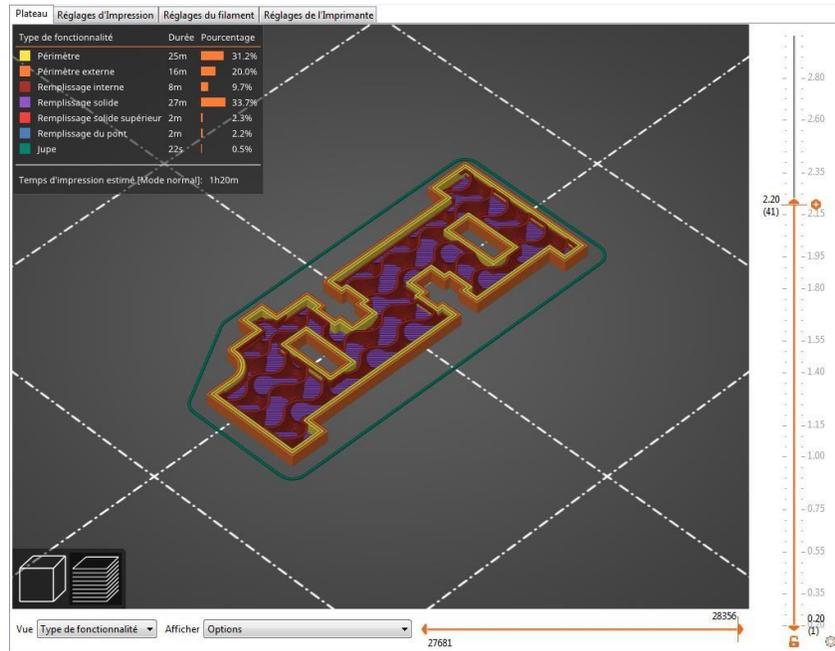


Figure II.21. Vue d'une coupe de la pièce avec un remplissage de 70%

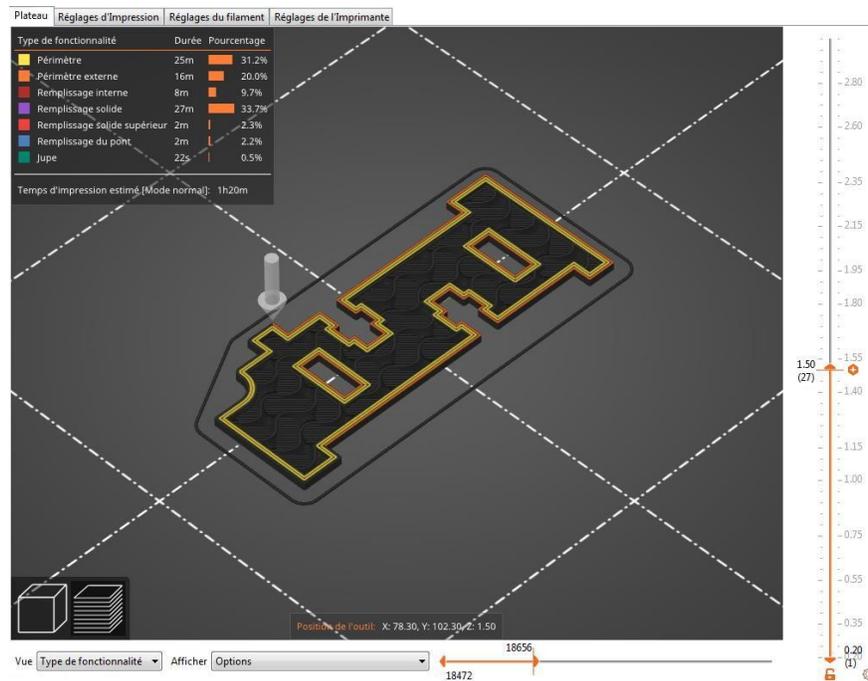


Figure II.22. Simulation de l'impression 3D de la pièce

II.8. Conclusion

Dans ce second chapitre, nous avons présenté la première partie de l'étude qui concerne l'architecture du bras manipulateur et ses composants. L'ensemble des pièces du bras manipulateurs ont été modélisé avec le logiciel Solidworks et les dessins de définition sont dans l'annexe.

L'aspect commande avec la carte programmable Arduino (Arduino Uno dans notre cas) et les servomoteurs sont décrit d'une manière succincte, notre objectif n'est pas de faire un mémoire de génie électrique mais juste de donner les notions nécessaires à la compréhension de la partie programmation du chapitre application.

Chapitre III :
Cinématique du
Bras Manipulateur

III.1 Introduction

La conception et la commande des robots nécessitent le calcul de modèles mathématiques décrivent le comportement de la structure du robot en fonction des paramètres articulaires de celui-ci. Selon la structure du robot ces modèles peuvent être de difficultés très variables.

Dans ce chapitre, on va présenter le modèle géométrique direct basé sur le modèle de Khalil modifiant le modèle de Denavit-Hartenberg (DH) sachant que ce modèle est très utilisé pour sa simplicité et sa capacité à décrire un grand nombre de structures de robots. À la fin de ce chapitre, le modèle de DHM (Denavit-Hartenberg Modifié) nous permettra de déterminer l'espace de travail du bras manipulateur étudié pour mettre en place notre application dans le chapitre suivant.

III.2. Modèle géométrique d'un robot en chaine simple :

III.2.1. Nécessité d'un modèle :

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques, tel que : Les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal) et l'espace articulaire (dans lequel est définie la configuration du robot). Parmi ces modèles, on distingue :

- Les modèles géométriques direct et inverse qui expriment la situation de l'organe terminal en fonction de la configuration du mécanisme et inversement,
- Les modèles cinématiques direct et inverse qui expriment la vitesse de l'organe terminal en fonction de la vitesse articulaire et inversement,
- Les modèles dynamiques définissant les équations du mouvement du robot, qui permettent d'établir les relations entre les couples ou forces exercés par les actionneurs et les positions, vitesses et accélérations des articulations. Définir les différentes tâches d'un robot réclame de pouvoir positionner l'organe terminal par rapport à un repère de référence. (voir figureIII.1) [32]

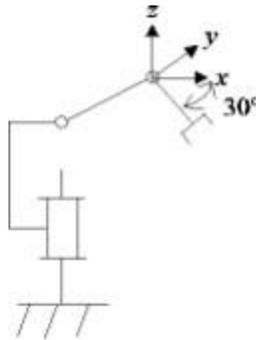


Figure III.1 : Modèle géométrique d'une chaîne simple [32]

III.2.2. Mécanismes :

On appelle mécanisme ou Système Mécanique Articulé un ensemble de solides reliés 2 à 2 par des liaisons. On distingue 2 types de mécanismes :

- Les mécanismes en chaîne simple ouverte (ou en série). Lorsque l'on parcourt le mécanisme, on ne repasse jamais 2 fois sur la même liaison, ou sur le même solide. Ce type de système est le plus répandu.
- Les mécanismes en chaîne complexe, i.e., tout ce qui n'est pas en série (au moins un solide avec plus de 2 liaisons). De tels systèmes se subdivisent en 2 groupes : les chaînes structurées en arbre, et les chaînes fermées (dont l'avantage est d'être a priori plus rigide, plus précis, capable de manipuler de lourdes charges).

Pour représenter un mécanisme, on dispose de 2 méthodes :

- Le schéma cinématique : On utilise la représentation normalisée des liaisons pour représenter le mécanisme, soit en perspective, soit en projection.
- Le graphe, non normalisé. A titre d'exemples, considérons quelques mécanismes :

(voir figure III.2) [33]

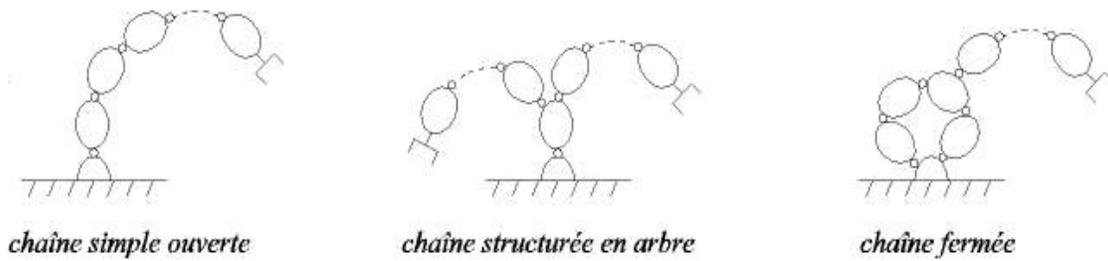


Figure III.2 : Type de chaînes cinématiques [33]

III.2.2.1. cinématiques série :

Une chaîne cinématique simple ouverte est une succession de solides placés en série et liés entre eux un à un par des liaisons simples. Autrement dit lorsqu'une pièce (n) est liée par l'intermédiaire de n-1 pièces placées en série et si n-1 pièces sont liées entre elles par des liaisons simples alors ces n pièces et n liaisons constituent une chaîne cinématique simple ouverte ou des liaisons en série. (Voir figure III.6)

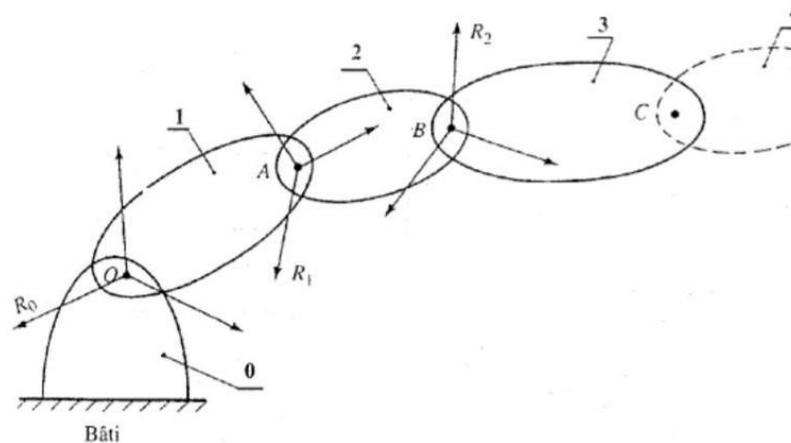


Figure III.3 : Chaîne cinématique simple ouverte

La plupart des robots industriels construits à ce jour est de type sériel, c'est-à-dire que leur structure mobile est une chaîne ouverte formée d'une succession de segments reliés entre eux par des liaisons à un degré de liberté. Chaque articulation est commandée par un actionneur situé à l'endroit de l'articulation ou sur un des segments précédents. [34]

III.2.2.2. Chaîne cinématique arborescente

Un robot à chaîne cinématique arborescente est constitué de plusieurs chaînes cinématiques ouvertes reliées à un corps central.

En fonctionnement, un robot à chaîne cinématique arborescente peut devenir un robot à chaîne cinématique fermée par la saisie d'une pièce avec deux bras ou le contact avec le sol de deux ou plusieurs pattes (voir figure III.7). [35]

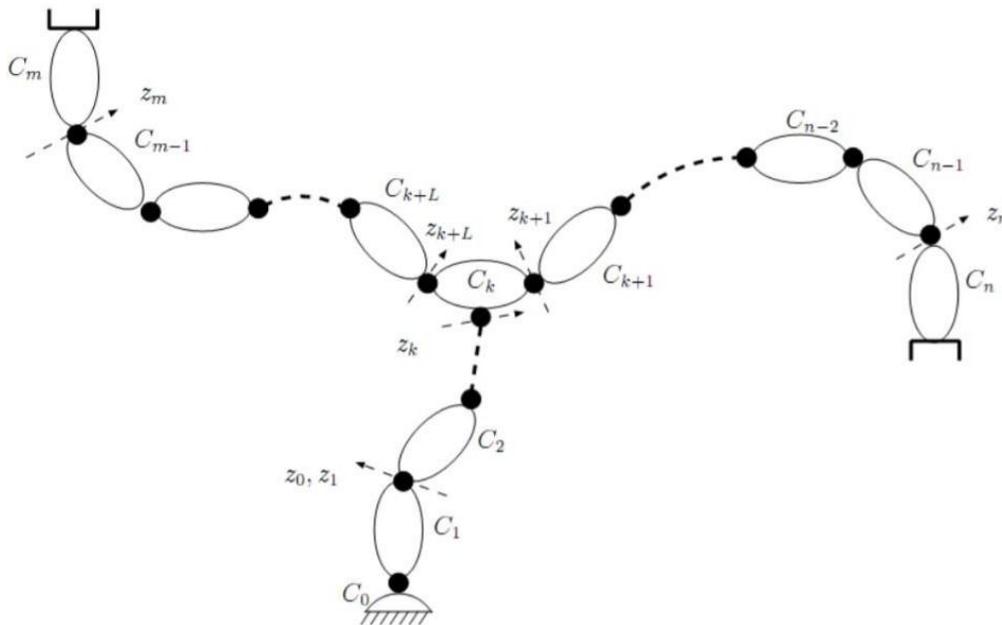


Figure III.4: Chaîne cinématique arborescente

III.2.2.3. Chaîne cinématique fermée

On appelle chaîne fermée une chaîne ouverte dont les deux solides extrêmes ont une liaison entre eux. Les $n+1$ solides sont reliés par $n+1$ liaisons. Un Mécanisme en chaîne cinématique fermée dont l'organe terminal est relié à la base par plusieurs chaînes cinématiques indépendantes (voir figure III.8). [36]

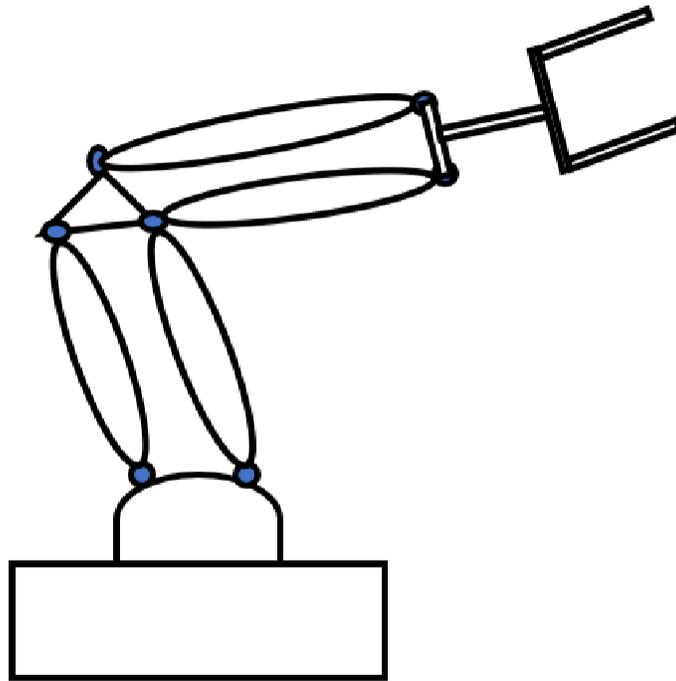


Figure III.5 : Chaîne cinématique fermée

III.2.3. Morphologie des robots manipulateurs :

Ce paragraphe est relatif aux chaînes ouvertes simples. Afin de dénombrer les différentes architectures possibles, on ne considère que 2 paramètres : le type d'articulation (rotoïde (R) ou prismatique (P)) et l'angle que font deux axes articulaires successifs (0° ou 90° ; sauf cas très particulier, les axes consécutifs d'un robot sont soit parallèles, soit perpendiculaires).

On convient d'appeler les 3 premiers degrés de liberté (D.D.L) le porteur du robot. Les (D.D.L) résiduels forment le poignet, caractérisé par des dimensions beaucoup plus petites et une plus faible masse.

Sont schématisées dans la figure III.3 qui suit les 12 morphologies possibles de porteur. Ces morphologies sont non redondantes (on élimine a priori les structures limitant les mouvements du porteur à des déplacements linéaires ou planaires : 3 liaisons prismatiques d'axes parallèles, par exemple, ou 3 liaisons rotoïdes d'axes parallèles).[37]

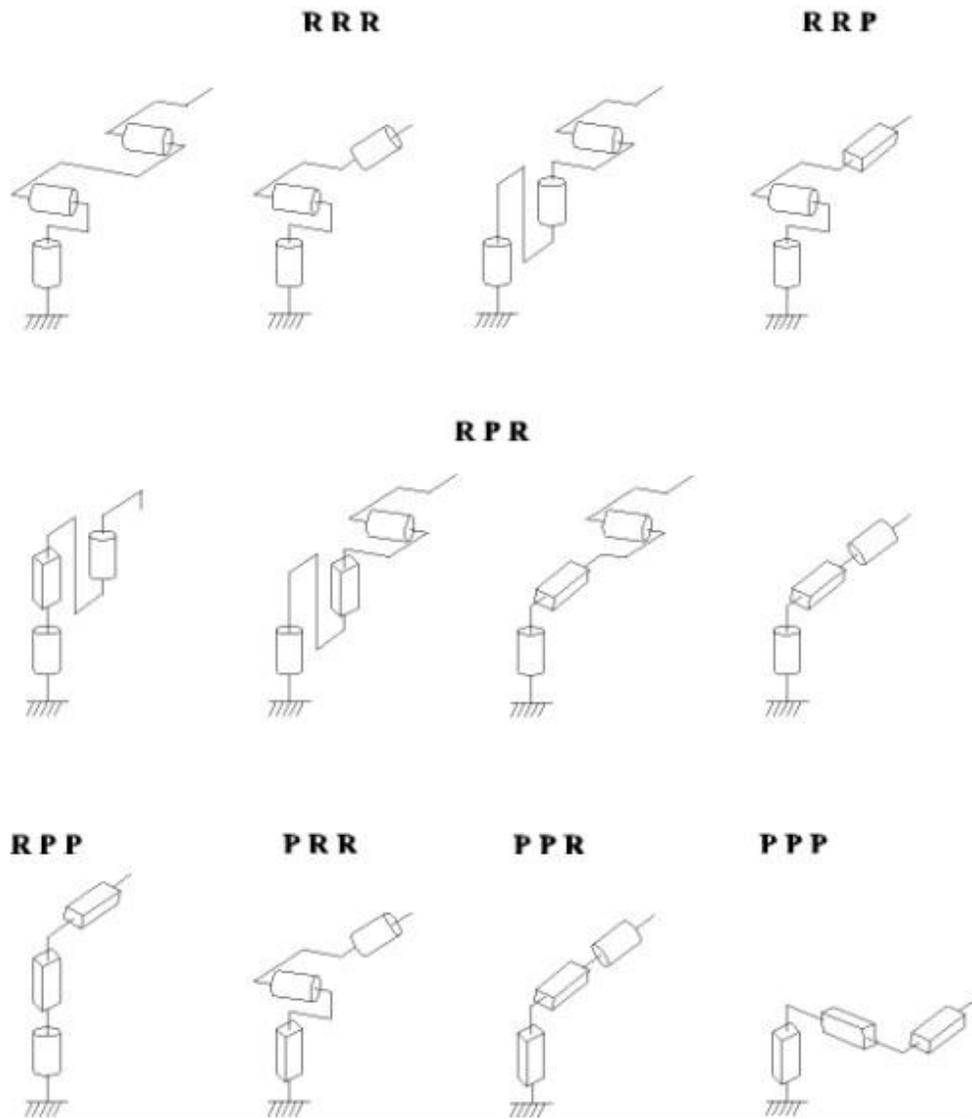


Figure III.6 : des modèles cinématique [37]

III.3. Modèle plan :

Les robots pleinement parallèles plans sont soit à deux ddl soit à trois ddl.

III.3.1. Robots plans à 2ddl :

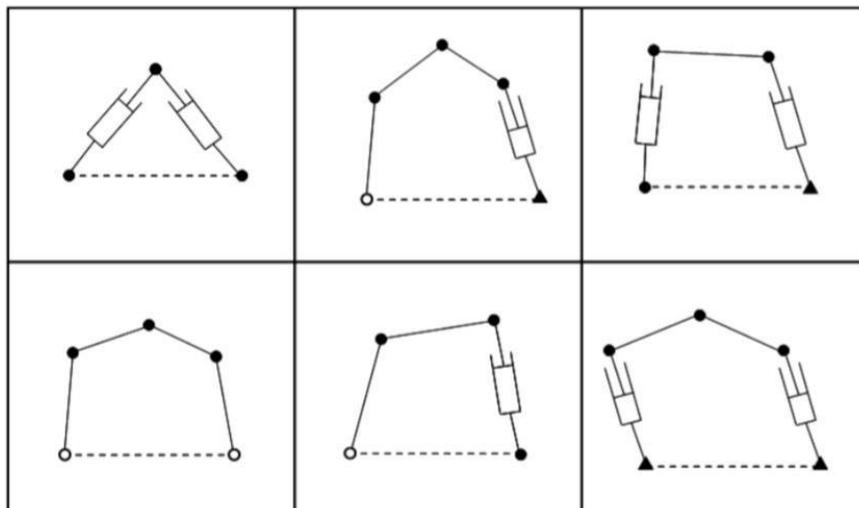


Figure III.7 : Robot plan à 2ddl

III.3.2. Robots plans à 3ddl :

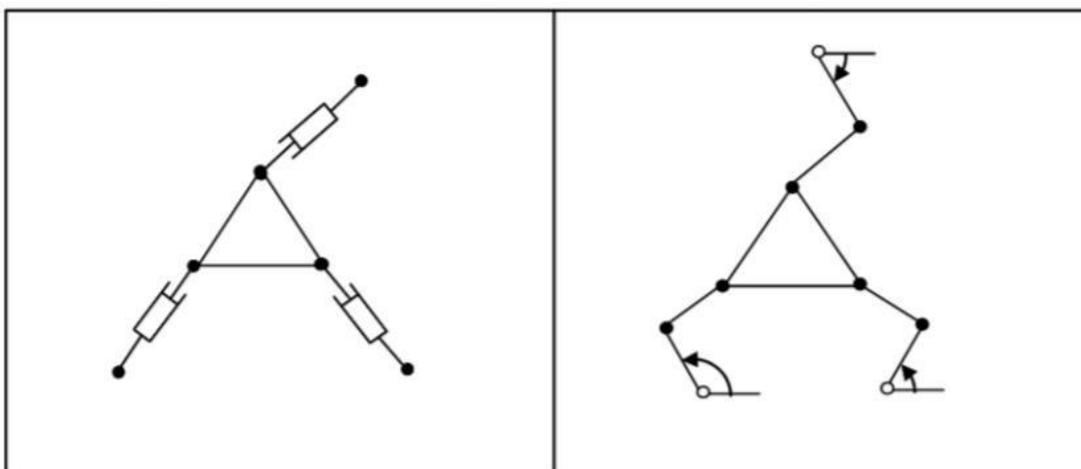


Figure III.8 : Exemples de robots plans à 3ddl le 3-RPR et le 3-RRR

Ces robots sont composés d'une plateforme mobile reliée par trois chaînes cinématiques. Chaque chaîne contient trois articulations qui peuvent être, soit de type rotoïde (R), soit de type prismatique (P)

III.4. 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 articulaires.

Dans le cas d'une chaîne ouverte simple, il peut être représenté par la matrice de transformation ${}^0\mathbf{T}_n$

$${}^0\mathbf{T}_n = {}^0\mathbf{T}_1(q_1) {}^1\mathbf{T}_2(q_2) \dots {}^{n-1}\mathbf{T}_n(q_n) \quad (1)$$

Le modèle géométrique direct du robot peut aussi être représenté par la relation :

$$\mathbf{X} = \mathbf{f}(\mathbf{q}) \quad (2)$$

\mathbf{q} : étant le vecteur des variables articulaires tel que :

$$\mathbf{q} = [q_1 q_2 \dots q_n]^T \quad q = [q_1 q_2 \dots q_n]^T \quad (3)$$

Les coordonnées opérationnelles sont définies par :

$$\mathbf{X} = [x_1 x_2 \dots x_m]^T \quad (4)$$

Plusieurs possibilités existent pour la définition du vecteur \mathbf{X} . Par exemple, avec les éléments de la matrice ${}^0\mathbf{T}_n$.

$$\mathbf{X} = [P_x P_y P_z s_x s_y s_z n_x n_y n_z a_x a_y a_z]^T \quad (5)$$

Ou bien, sachant que $\mathbf{s} = \mathbf{n} \times \mathbf{a}$:

$$\mathbf{X} = [P_x P_y P_z n_x n_y n_z a_x a_y a_z]^T \quad (6)$$

Pour les rotations, d'autres représentations sont couramment utilisées comme les angles d'Euler, les angles de Roulis-Tangage-Lacet (RTL) ou les paramètres d'Euler (quaternions). On peut facilement passer des cosinus directeurs \mathbf{s} , \mathbf{n} , \mathbf{a} à l'une quelconque de ces représentations et inversement.

- Pour notre étude d'un bras manipulateur de trois degré de liberté (ROTATION,

ROTATION , ROTATION) (RRR) (Figure III.9), on utilise la méthode de Denavit-Hartenberg pour calculer la matrice de transformation globale.

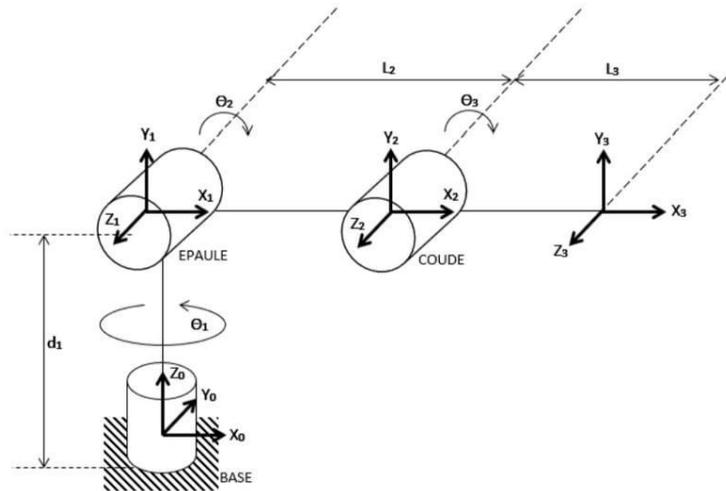


Figure III.9 : modèle RRR

Pour effectuer la description cinématique du manipulateur, nous associerons un référentiel à chaque bras du robot, et nous décrirons leur position relative à l'aide des paramètres de Denavit-Hartenberg. La description cinématique du manipulateur complet est alors obtenue très simplement, par récurrence. [38]

III.4.1. Modèle de Denavit Hartenberg (DH) :

Les paramètres de Denavit-Hartenberg modifié permettent de disposer d'un paramétrage des liaisons tel que les matrices de passage aient toutes la même forme littérale, ce qui facilite les calculs.

La méthode qui suit s'applique lorsque le robot correspond à une chaîne simple ouverte et que ses articulations sont rotoïdes, ou prismatiques (ce qui est le cas en général). Les corps constituant le robot sont supposés parfaitement rigides et connectés par des articulations idéales (pas de jeu mécanique, pas d'élasticité).

Notations :

On numérote les solides par ordre croissant en partant du socle. Ainsi le robot est composé de $n + 1$ corps, notés C_0, \dots, C_n , et de n articulations ($n - 1$). Le corps C_0 désigne le socle (la base) du robot, le corps C_n le corps portant l'organe terminal.

Le repère R_j est lié au corps C_j du robot.

La variable de l'articulation j , qui lie le corps C_j au corps C_{j-1} , est notée q_j .

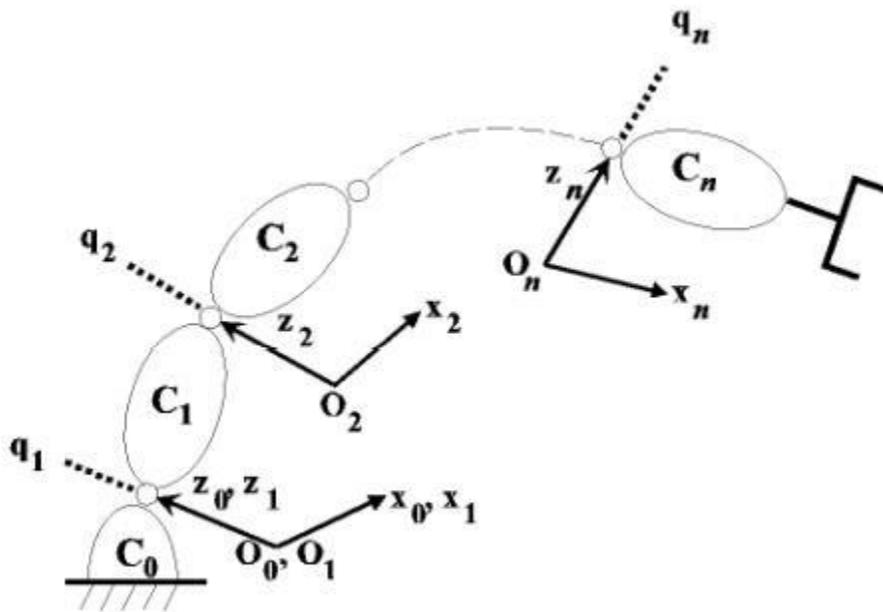


Figure III.10 : Robot a structure ouverte

Denavit et Hartenberg ont proposés une méthode pour décrire la transformation homogène entre deux solides adjacents avec une position et orientation particulière des repères.

Pour la construction des repères il faut suivre la procédure suivante : Nommer les corps du robot de $i=0$ jusqu'à $i=n$ en commençant par la base du robot avec $i=0$;

Nommer les articulations de $i=1$ à n (1 pour le premier degré de liberté et n pour le dernier) ; Pour $i=0$ jusqu'à $i=n-1$ fixer l'axe Z_i sur l'articulation $i+1$;

L'origine du repère R_0 sera un point quelconque de l'axe Z_0 , de sorte que, X_0 et Y_0 forment un repère orthonormé direct ;

Pour $i=1,2,\dots,n-1$, l'origine du repère est fixé à l'intersection de l'axe Z_i avec la droite perpendiculaire commune à Z_{i-1} et Z_i . Si les deux axes se coupent, l'origine est le point d'intersection. Si par contre, les axes sont parallèles l'origine est l'origine du repère de l'articulation $i+1$.

L'axe X_i est la droite perpendiculaire commune à Z_{i-1} et Z_i , et l'axe Y_i est choisi de sorte que le repère soit orthonormé direct. (La Figure III.11) illustre cette méthode.

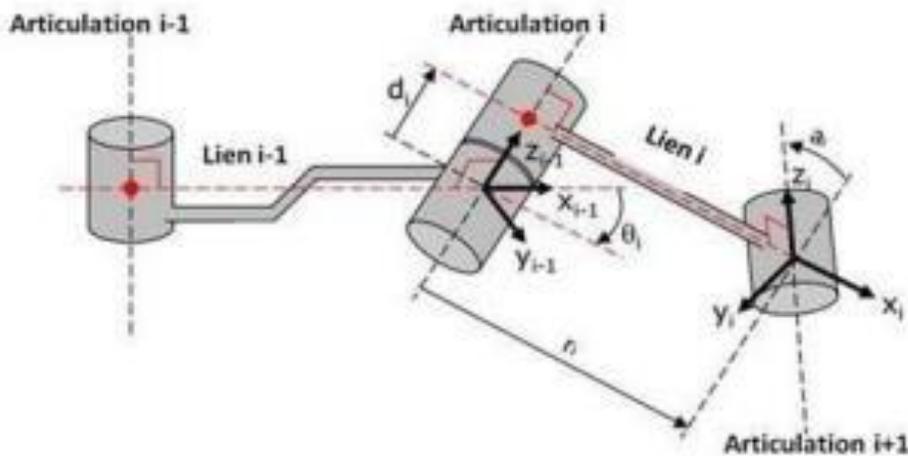


Figure III.11 : méthode de transformation homogène

Avec cette convention l'équation de la matrice de transformation du repère $i-1$ au repère i est :

$${}^{i-1}T_i = \text{Rot}(Z_{i-1}, \theta_i) \text{Trasl}(Z_{i-1}, d_i) \text{Trasl}(X_i, a) \text{Rot}(X_i, \alpha_i) \quad (7)$$

Ou:

$\text{Rot}(Z_{i-1}, \theta_i)$ représente la rotation un angle θ_i autour de l'axe Z_{i-1}

Trasl($Z_{i=1}, d_i$) représente la translation une distance d_i le long de l'axe

Trasl(X_i, r_i) représente la translation une distance r_i le long de l'axe X_i .

Rot(X_i, α_i) représente la rotation un angle α_i autour de l'axe

θ_i est l'angle entre et autour l'axe .

d_i est la distance entre et le long de .

r_i est la distance entre et le long , et

α_i est l'angle entre et autour.

De cette façon les matrices de transformation s'écrivent :

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & r & 1 & 0 & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & d_i & 0 & 0 & 1 & 0 & 0 & 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Finalement, nous obtenons la matrice de transformation du repère $i-1$ au repère i :

$${}^{i-1}T_i = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & r_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & r_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

Le modèle de DH est le plus répandu dans la communauté scientifique pour la modélisation des robots.

Par contre ce modèle a la particularité que l'axe Z_i du repère R_i est confondu avec l'axe de rotation $i+1$ du robot et pas avec l'axe de rotation i . Ainsi, les repères sont mis en place de façon différente comme dans les travaux de Paul [Paul81] et de Craig [Craig86]. Dans les conventions de Paul et de Craig, des changements de placement de repères par rapport à la convention originale de DH sont faits comme le montre le tableau comparatif Tableau 1.

L'objectif de leur convention est d'avoir une cohérence entre les valeurs des paramètres associés à chaque lien et d'éviter les confusions car la convention de DH a été développée pour des robots avec des mécanismes en série simples. [39]

Critère	DH	PAUL
repère i en liaison i	1	1
θ_i et d_i autour de l'axe Z_i	1	0
α_i et r_i autour de l'axe X_i	0	1

Tableau II.1

III.4.2. Modèle de Khalil :

En 1986, Khalil et Klein ont fait des améliorations du modèle de DH en raison des ambiguïtés vis à vis des robots avec des structures fermées ou arborescentes [Khalil99]. Ils ont aussi proposé une méthode pour des robots avec des architectures ouvertes simples. Pour des robots à structure ouverte, la convention suivante est proposée : le repère R_j est fixe avec la liaison j , l'axe de l'articulation j est l'axe z_j et l'axe x_j est aligné avec la perpendiculaire commune aux axes d'articulation actuels j et suivants $j+$ (figure III.12).

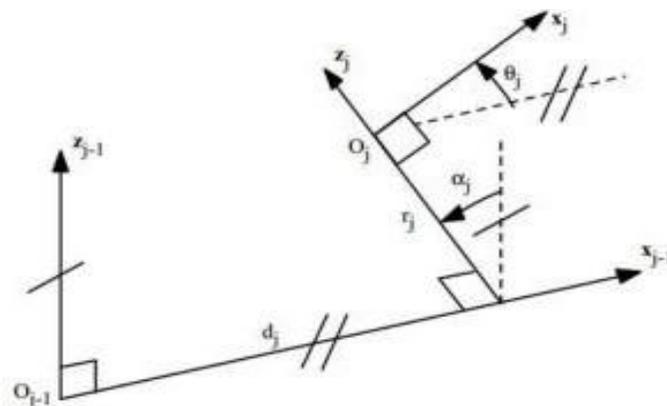


Figure III.12 : Modèle Khalil

Avec cette convention l'équation de la matrice de transformation du repère j-1 au repère j est :

$${}^{j-1}T_j = \text{Rot}(Z_{j-1}, \theta_j) \text{Trasl}(Z_{j-1}, d_j) \text{Trasl}(X_j, r_j) \text{Rot}(X_j, \alpha_j) \quad (10)$$

ou:

θ_j est l'angle entre x_{j-1} et x_j autour l'axe z_j .

r_j est la distance entre x_{j-1} et x_j le long de z_j .

d_j est la distance entre z_{j-1} et z_j le long x_{j-1} , et

α_j est l'angle entre z_{j-1} et z_j autour x_{j-1} .

La matrice de transformation de repère j relative au repère j-1 est :

$$T_j = \begin{bmatrix} c\theta_j & -s\theta_j & 0 & d_j \\ c\alpha_j s\theta_j & c\alpha_j c\theta_j & -s\alpha_j & -r_j s\alpha_j \\ s\alpha_j s\theta_j & s\alpha_j c\theta_j & c\alpha_j & r_j c\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Pour des robots à structure fermée, la matrice de transformation du repère de l'effecteur final au repère de base du robot est la composition des matrices de transformation en prenant le chemin le plus court entre le repère de l'effecteur final (Ref) et le repère de la base du robot (R0). Pour le cas des structures fermées de type parallélogramme Khalil et Klein finger appliquent une méthode simple en utilisant le fait que les droites C1, C2, C3 et C4 forment un parallélogramme, (voir Figure III.13) dans ce cas les matrices d'orientation entre les repères R1 et R3 du parallélogramme son constantes.

Le modèle de Khalil-Klein finger et les modèles précédents sont dérivés de la notation de DH et, en général, sont les plus répandus pour la modélisation des robots. D'autres travaux s'intéressent à l'ajout de paramètres géométriques au modèle de DH [40]

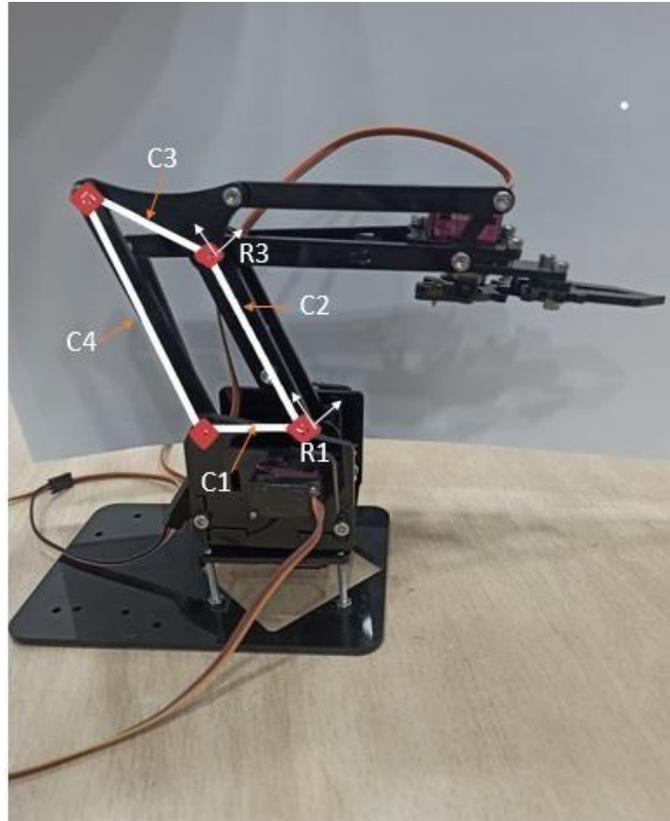


Figure III.13 : exemple de méthode de Khalil

III.5. Modèle géométrique indirect :

On a vu que le modèle géométrique direct d'un robot permettait de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires. Le problème inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles (il y a rarement unicité de solution) constitue ce que l'on appelle le modèle géométrique inverse (MGI). On peut distinguer trois méthodes de calcul du MGI :

- la méthode de Paul [Paul81] qui traite séparément chaque cas particulier et convient pour la plupart des robots industriels
- la méthode de Peiper [Pieper68] qui permet de résoudre le problème pour les robots à six degrés de liberté possédant trois articulations rotoïdes d'axes concourants ou trois articulations prismatiques

– la méthode générale de Raghavan et Roth [Raghavan90], donnant la solution générale des robots à six articulations à partir d'un polynôme de degré au plus égal à 16.

Lorsqu'il n'est pas possible de trouver une forme explicite du modèle géométrique inverse, on peut calculer une solution particulière par des procédures numériques [Pieper68], [Whitney69], [Fournier80], [Featherstone83], [Wolovich84], [Goldenberg85], [Sciavicco86] [41].

III.5.1. Modélisation cinématique :

III.5.1.1 Modèle cinématique direct :

Le modèle cinématique directe d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires. Il est noté :

$$\dot{\mathbf{X}}=\mathbf{J}(\mathbf{q}) \dot{\mathbf{q}} \quad (17)$$

Où $\mathbf{J}(\mathbf{q})$ désigne la matrice jacobienne de dimension (mxn) du mécanisme, égale à $\frac{\partial \mathbf{X}}{\partial \mathbf{q}}$ et fonction de la configuration articulaire \mathbf{q} . La même matrice jacobienne intervient dans le calcul du modèle différentiel direct qui donne les variations élémentaires $d\mathbf{X}$ des coordonnées opérationnelles en fonction des variations élémentaires des Coordonnées articulaires $d\mathbf{q}$, soit :

$$d\mathbf{X}=\mathbf{J}(\mathbf{q}) d\mathbf{q} \quad (18)$$

– elle est à la base du modèle différentiel inverse, permettant de calculer une solution locale des variables articulaires \mathbf{q} connaissant les coordonnées opérationnelles \mathbf{X} ;

– en statique, on utilise le jacobienne pour établir la relation liant les efforts exercés par l'organe terminal sur l'environnement aux forces et couples des actionneurs;

– elle facilite le calcul des singularités et de la dimension de l'espace opérationnel accessible du robot [42]

III.5.1.2 Modèle cinématique inverse :

L'objectif du modèle cinématique inverse est de calculer, à partir d'une configuration \mathbf{q} donnée, les vitesses articulaires $\dot{\mathbf{q}}$ qui assurent au repère terminal une vitesse opérationnelle $\dot{\mathbf{X}}$ imposée. Cette définition est analogue à celle du modèle différentiel inverse : ce dernier permet de déterminer la différentielle articulaire $d\mathbf{q}$ correspondant à

une différentielle des coordonnées opérationnelles $d\mathbf{X}$ spécifiée. Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique directe en résolvant un système d'équations linéaires. La mise en œuvre peut être faite de façon analytique ou numérique :

- la solution analytique a pour avantage de diminuer considérablement le nombre d'opérations, mais on doit traiter séparément tous les cas singuliers
- les méthodes numériques sont plus générales, la plus répandue étant fondée sur la notion de pseudo-inverse : les algorithmes traitent de façon unifiée les cas réguliers, singuliers et redondants. Elles nécessitent un temps de calcul relativement important.

Nous présentons dans ce paragraphe les techniques à mettre en œuvre pour établir un modèle cinématique inverse dans les cas réguliers, singuliers et redondants. [42]

III.5.1.3. Forme générale du modèle cinématique :

Soit $\mathbf{X}=[\mathbf{X}_p^T \mathbf{X}_r^T]^T$ une représentation quelconque dans le repère R_0 de la situation du repère R_n fixé à un solide, les éléments \mathbf{X}_p et \mathbf{X}_r désignant respectivement la position et l'orientation opérationnelles du solide. Les relations entre les vitesses $\dot{\mathbf{X}}_p$ et $\dot{\mathbf{X}}_r$ et les vecteurs vitesses ${}^0\mathbf{V}_n$ et ${}^0\boldsymbol{\omega}_n$ du repère R_n sont telles que :

$$\begin{bmatrix} \dot{\mathbf{X}}_p \\ \dot{\mathbf{X}}_r \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Omega}_p & \mathbf{0}_3 \\ \mathbf{0}_3 & \boldsymbol{\Omega}_r \end{bmatrix} \begin{bmatrix} {}^0\mathbf{V}_n \\ {}^0\boldsymbol{\omega}_n \end{bmatrix} = \boldsymbol{\Omega} \begin{bmatrix} {}^0\mathbf{V}_n \\ {}^0\boldsymbol{\omega}_n \end{bmatrix} \quad (19)$$

Les matrices $\boldsymbol{\Omega}_p$ et $\boldsymbol{\Omega}_r$ dépendant de la représentation choisie respectivement pour la position et pour l'orientation. [42]

III.5.1.4. Modélisation dynamique :

Le modèle dynamique est la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires. On représente le modèle dynamique par une relation de la forme (19) :

$$\boldsymbol{\Gamma} = \mathbf{f}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{f}_e) \quad (20)$$

Avec :

- $\boldsymbol{\Gamma}$: Vecteur des couples/forces des actionneurs, selon que l'articulation est rotoïde ou prismatique. Dans la suite, on écrira tout simplement *couples*

- \mathbf{q} : Vecteur des positions articulaires ;
- $\dot{\mathbf{q}}$: Vecteur des vitesses articulaires ;
- $\ddot{\mathbf{q}}$: Vecteur des accélérations articulaires ;
- \mathbf{f}_e : Vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement.

On convient d'appeler modèle dynamique inverse, ou tout simplement modèle dynamique, la relation de la forme. [43]

III.6. Application : bras manipulateur RRR :

Dans notre cas d'étude on termine l'approche analytique A partir de la méthode de Denavit-Hartenberg

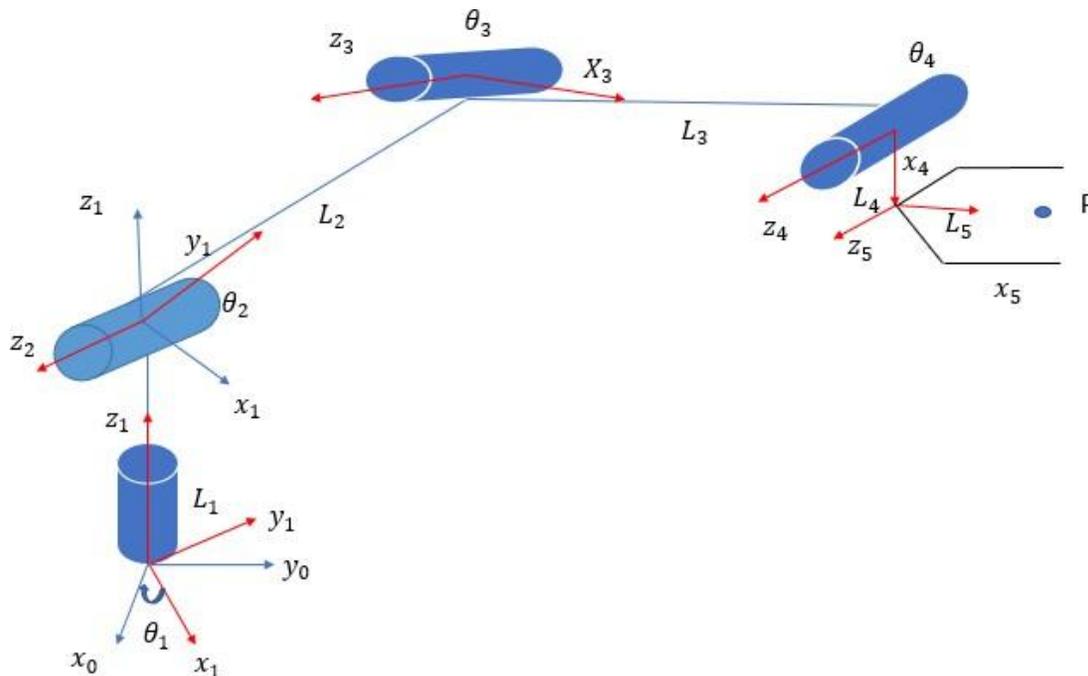


Figure III.14 : diagramme 3D d'un Bra

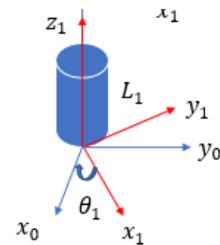
	α_i	d_i	θ_i	r_i
1	0	0	θ_1	L_1
2	90°	0	θ_2	0
3	0	L_2	θ_3	0
4	0	L_3	θ_4	0
5	0	L_4	90°	0

TableauII.1 : Paramètres de Denavit-Hartenberg.

Les matrices de transféré :

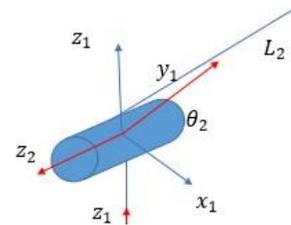
1- Pour le repère de l'origine (R_0) la liaison de rotation(R)

$$H_0^1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$



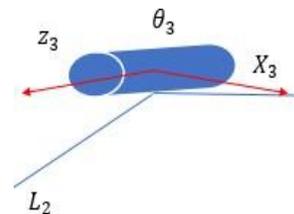
2- Pour le repère de l'origine (R_1) la liaison de rotation(R)

$$H_1^2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (22)$$



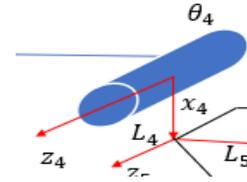
3- Pour le repère de l'origine (R_2) la liaison de rotation(R)

$$H_2^3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & L_2 \\ \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$



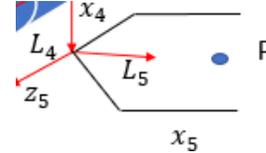
4-pour le repère de l'origine (R₃) la liaison de rotation(R)

$$H_4^3 = \begin{bmatrix} \cos\theta_4 & -\sin\theta_4 & 0 & L_3 \\ \sin\theta_4 & \cos\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (24)$$



4- Pour le repère (R₃) la liaison de Translation(P)

$$H_5^4 = \begin{bmatrix} 1 & 0 & 0 & L_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (25)$$



$$H_5^3 = \begin{bmatrix} -\sin\theta_4 & -\cos\theta_4 & 0 & L_4\cos\theta_4 + L_3 \\ \cos\theta_4 & -\sin\theta_4 & 0 & L_4\sin\theta_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

$$I_5^2 = \begin{bmatrix} -\sin(\theta_3 + \theta_4) & -\cos(\theta_3 + \theta_4) & 0 & L_4\cos(\theta_3 + \theta_4) + \cos\theta_3L_3 + L_2 \\ \cos(\theta_3 + \theta_4) & -\sin(\theta_3 + \theta_4) & 0 & L_4\sin(\theta_3 + \theta_4) + \sin\theta_3L_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

$$H_5^1 = \begin{bmatrix} -\sin(\theta_2 + \theta_3 + \theta_4) & -\cos(\theta_2 + \theta_3 + \theta_4) & 0 & L_4\cos(\theta_2 + \theta_3 + \theta_4) + L_3\cos(\theta_2 + \theta_3) + L_2\cos\theta_2 \\ 0 & 0 & -1 & 0 \\ \cos(\theta_2 + \theta_3 + \theta_4) & -\sin(\theta_2 + \theta_3 + \theta_4) & 0 & L_4\sin(\theta_2 + \theta_3 + \theta_4) + L_3\sin(\theta_2 + \theta_3) + L_2\sin\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (28)$$

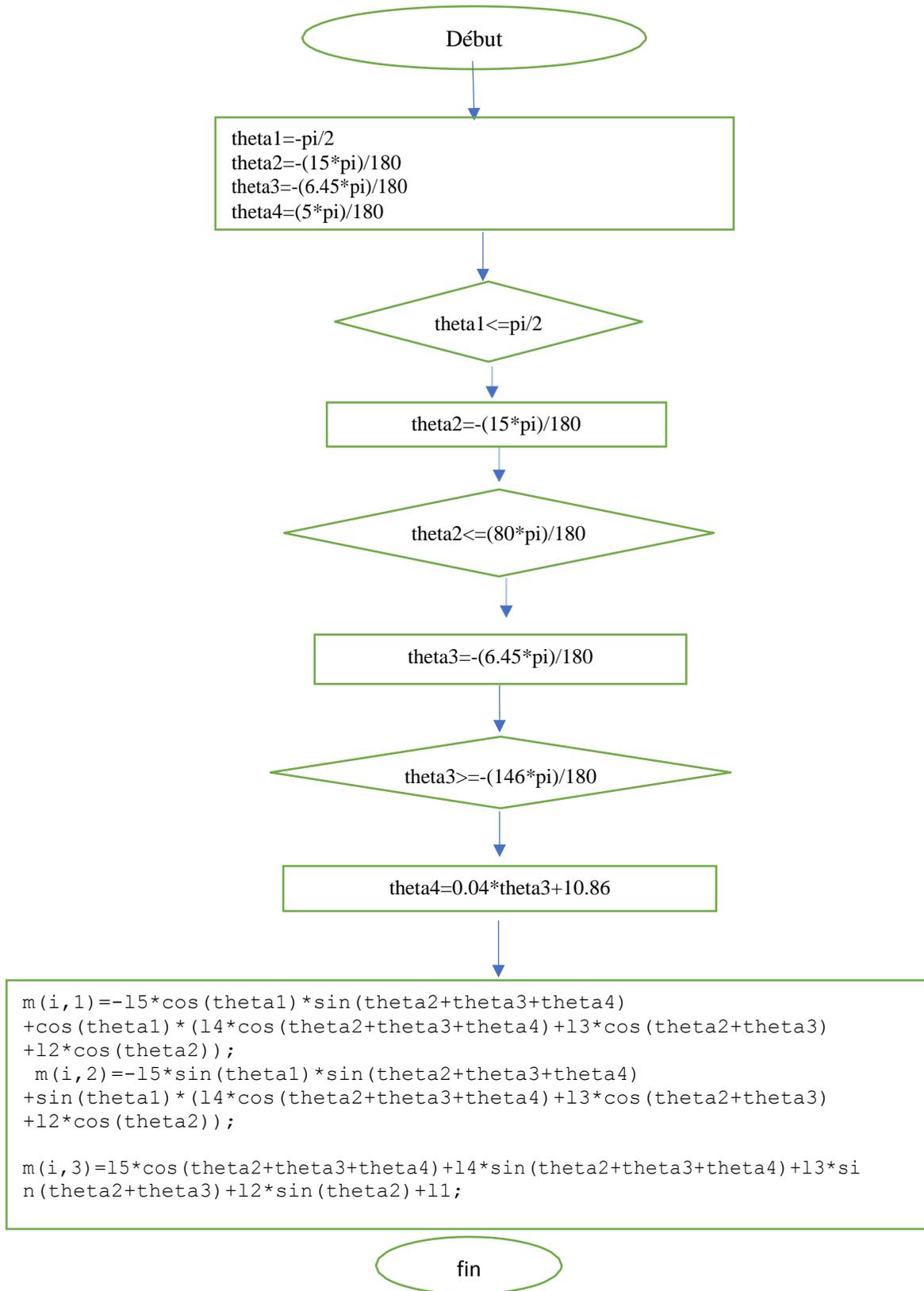
$$H_5^0 = \begin{bmatrix} -\cos\theta_1\sin(\theta_2 + \theta_3 + \theta_4) & -\cos\theta_1\cos(\theta_2 + \theta_3 + \theta_4) & \sin\theta_1 & \cos\theta_1(L_4\cos(\theta_2 + \theta_3 + \theta_4) + L_3\cos(\theta_2 + \theta_3) + L_2\cos\theta_2) + L_1\sin\theta_1 \\ -\sin\theta_1\sin(\theta_2 + \theta_3 + \theta_4) & -\sin\theta_1\cos(\theta_2 + \theta_3 + \theta_4) & -\cos\theta_1 & \sin\theta_1(L_4\cos(\theta_2 + \theta_3 + \theta_4) + L_3\cos(\theta_2 + \theta_3) + L_2\cos\theta_2) + L_1\cos\theta_1 \\ \cos(\theta_2 + \theta_3 + \theta_4) & -\sin(\theta_2 + \theta_3 + \theta_4) & 0 & L_4\sin(\theta_2 + \theta_3 + \theta_4) + L_3\sin(\theta_2 + \theta_3) + L_2\sin\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (29)$$

Pour effectuer l'analyse géométrique, réécrivons la matrice (29) sous la forme suivante :

$$P/R_0 = H_5^0 x P/R_5 \quad P_{(R5)} = \begin{pmatrix} L_5 \\ 0 \\ 0 \\ 1 \end{pmatrix} \quad (31)$$

$$P_{(R0)} = \begin{bmatrix} -L_5\cos\theta_1\sin(\theta_2 + \theta_3 + \theta_4) + \cos\theta_1(L_4\cos(\theta_2 + \theta_3 + \theta_4) + L_3\cos(\theta_2 + \theta_3) + L_2\cos\theta_2) + L_1\sin\theta_1 \\ -L_5\sin\theta_1\sin(\theta_2 + \theta_3 + \theta_4) + \sin\theta_1(L_4\cos(\theta_2 + \theta_3 + \theta_4) + L_3\cos(\theta_2 + \theta_3) + L_2\cos\theta_2) - L_1\cos\theta_1 \\ L_5\cos(\theta_2 + \theta_3 + \theta_4) + L_4\sin(\theta_2 + \theta_3 + \theta_4) + L_3\sin(\theta_2 + \theta_3) + L_2\sin\theta_2 + L_1 \\ 1 \end{bmatrix} \quad (32)$$

Organigramme de l'algorithme pour générer l'espace de travail du robot RRR :



Algorithme transcrit en programme sur Matlab :

```

theta1=-pi/2;
theta2=-(15*pi)/180;
theta3=-(6.45*pi)/180;
%theta4=(5*pi)/180;
l1=35;
l2=100;
l3=80;
l4=10;
l5=53;
i=1;
%m=zeros(1000,3);
while theta1<=pi/2
    theta2=-(15*pi)/180;
    while theta2<=(80*pi)/180
        theta3=-(6.45*pi)/180;
        while theta3>=-(146*pi)/180
            theta4=0.04*theta3+10.86;

            m(i,1)=-
15*cos(theta1)*sin(theta2+theta3+theta4)+cos(theta1)*(14*
cos(theta2+theta3+theta4)+13*cos(theta2+theta3)+12*cos(th
eta2));
            m(i,2)=-
15*sin(theta1)*sin(theta2+theta3+theta4)+sin(theta1)*(14*
cos(theta2+theta3+theta4)+13*cos(theta2+theta3)+12*cos(th
eta2));

m(i,3)=15*cos(theta2+theta3+theta4)+14*sin(theta2+theta3+
theta4)+13*sin(theta2+theta3)+12*sin(theta2)+l1;
                i=i+1;
                theta3=theta3-(5*pi)/180;
            end;
            theta2=theta2+(5*pi)/180;
        end;
        theta1=theta1+(5*pi)/180;
    end;
end;
save('E:/results6.txt','m','-ascii')

```

Dans notre étude du modèle géométrique direct de notre bras manipulateur RRR, nous avons utilisé le logiciel de MATLAB pour déterminer l'espace de travail de ce bras.

Nous avons réalisé le programme MATLAB et nous avons obtenu les graphes suivants (voir figure III.18) :

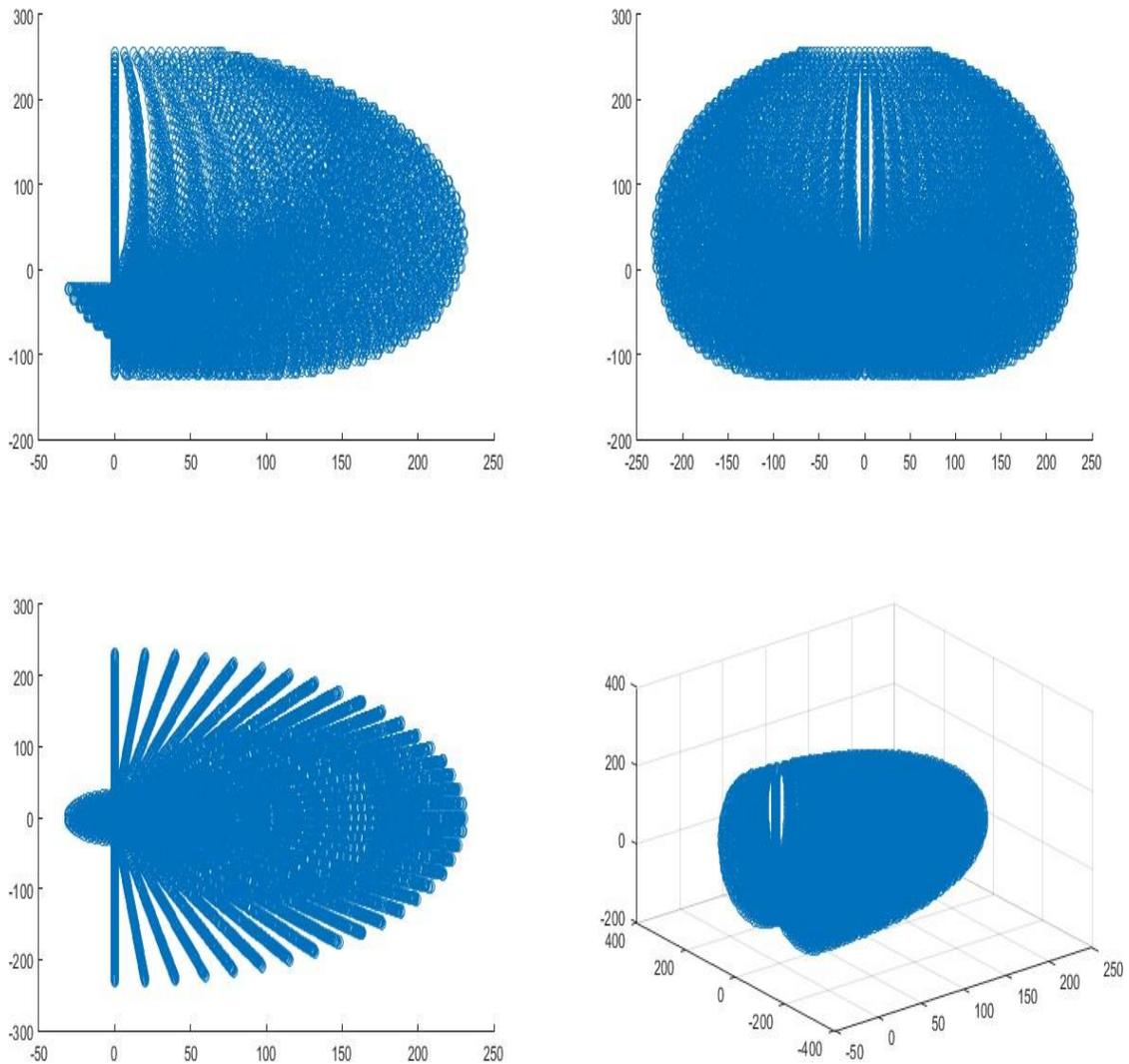


Figure III.15 : L'espace de travail 3D (xyz) et 2D (xy,oyz,oxz).

Nous observons d'après les graphes de la figure III.18 que l'espace de travail obtenu par notre algorithme et en tenant compte des paramètres de la fiche technique du Robot ROBV (voir organigramme) que sont espace représente un demi-sphère creuse. Ce résultat correspond parfaitement à l'espace de travail des robots de type RRR.

III.7. Conclusion

Dans ce chapitre, nous avons présenté le modèle cinématique qui permet de décrire le mouvement d'un robot et en particulier pour notre cas un Bras Manipulateur. Nous avons appliqué le modèle de Denavit-Hartenberg modifié (Modèle de Khalil) à notre robot pour établir le modèle géométrique direct.

Pour finir, nous avons mis en place un programme Matlab pour générer l'espace de travail du robot à partir du Modèle Géométrique Direct.

CHAPITRE VI :

Application

VI.1. Introduction :

Dans ce dernier chapitre, nous allons voir l'application qui consiste à l'assemblage du bras manipulateur et surtout la programmation de la carte Arduino pour effectuer le contrôle du robot.

Nous allons réaliser dans ce chapitre plusieurs programmes qui permettent de manipuler le bras manipulateur en mode manuel avec des potentiomètres et un mode à distance en utilisant une interface sur Android sur Smartphone et le Bluetooth pour communiquer avec notre robot.

Pour finir nous allons parler du mode de programmation automatique du robot pour effectuer une tâche en apprentissage passive pour réaliser une opération de « Pick-and-Place » pour une application future.

VI.2. Assemblage du bras manipulateur :

Dans les chapitres précédents nous avons vu la structure du bras manipulateur ainsi que le modèle mathématique qui décrit son mouvement. Maintenant, nous passons à la réalisation en commençant par assembler les pièces du bras manipulateur et souder tous les composants du Shield Arduino (voir figure VI.1 et 2).

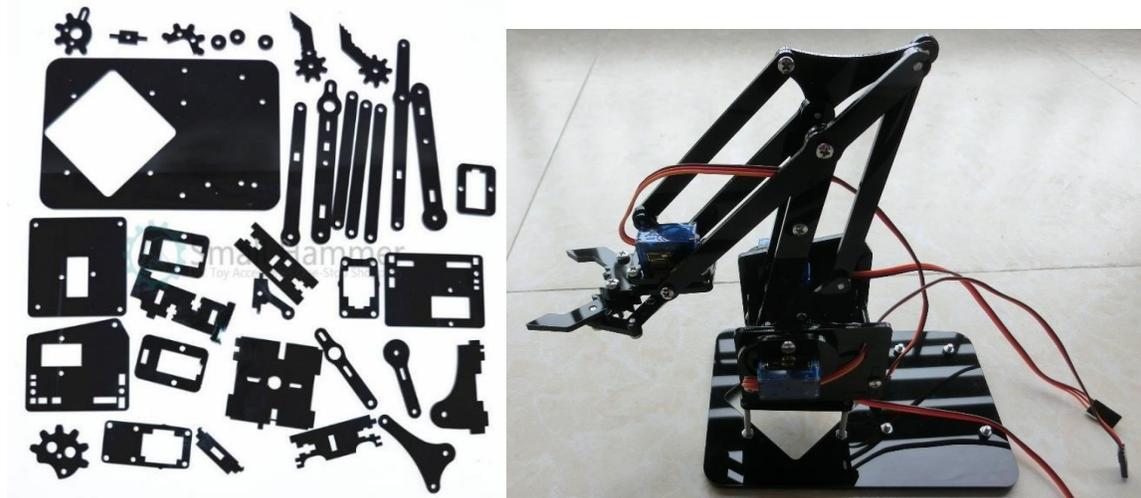


Figure VI.1 : Assemblage du bras manipulateur (Avant – Après)

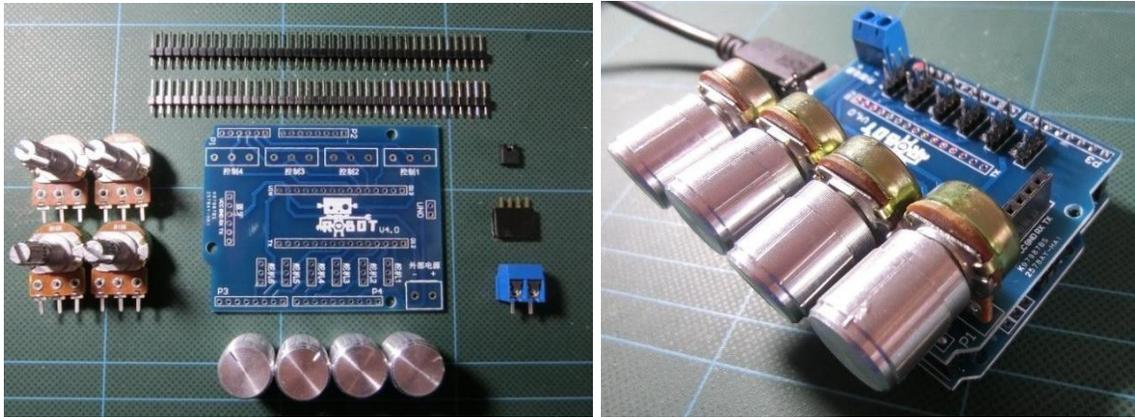


Figure VI.2 : Assemblage des composants du Shield Arduino (Avant – Après)

Pour faciliter l'assemblage des pièces mécaniques, le modèle CAO sur Solidworks a été utilisé comme référence et nous a permis de vérifier les degrés de liberté du bras manipulateur.

Après l'assemblage des deux parties présentées dans les figures VI.1 et 2, des tests de bon fonctionnement ont été réalisés pour éviter tout problème lors de la programmation sur Arduino Uno.

VI.3. Environnement de travail :

Dans le milieu de travail de robot, dans notre cas un bras manipulateur, il y a un environnement calculer précisément pour un travail forcé et répétitive sans commettre d'erreurs et mettre en danger la vie des travailleurs.

Le système robotique assiste le geste du professionnel en environnement de travail (atelier, Hangar de stockage, usine ...) le principe de leur fonctionnement est de travailler rapidement d'une manière répétitive et précise, sans mettre la vie des autres en danger. Pour cela, on doit définir avec précision l'enveloppe de l'espace de travail comme on l'a vue dans le chapitre précédent. D'autre part, cela permet de positionner les plans de travail (pièces, tables, etc.) d'une manière accessible au bras manipulateur en évitant tout obstacle possible.

Pour cela, nous présentons l'enveloppe de l'espace de travail d'après les calculs effectués avec le Modèle Géométrique Direct du chapitre III, illustré dans les figures VI.3 et 4.

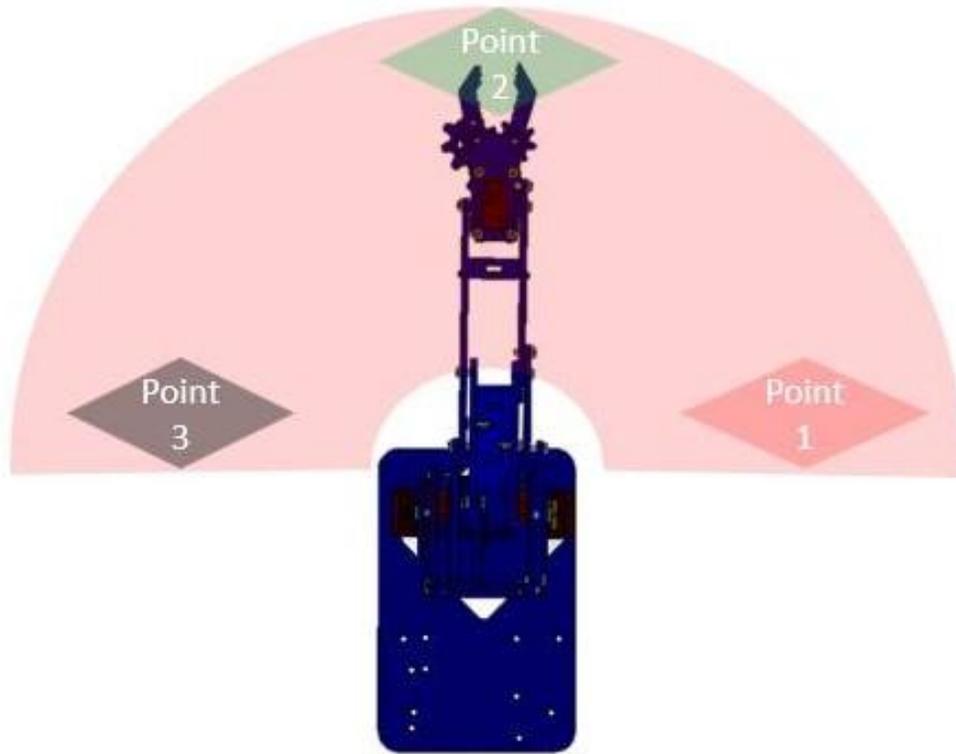


Figure VI.3 : Environnement de travail vu de dessous

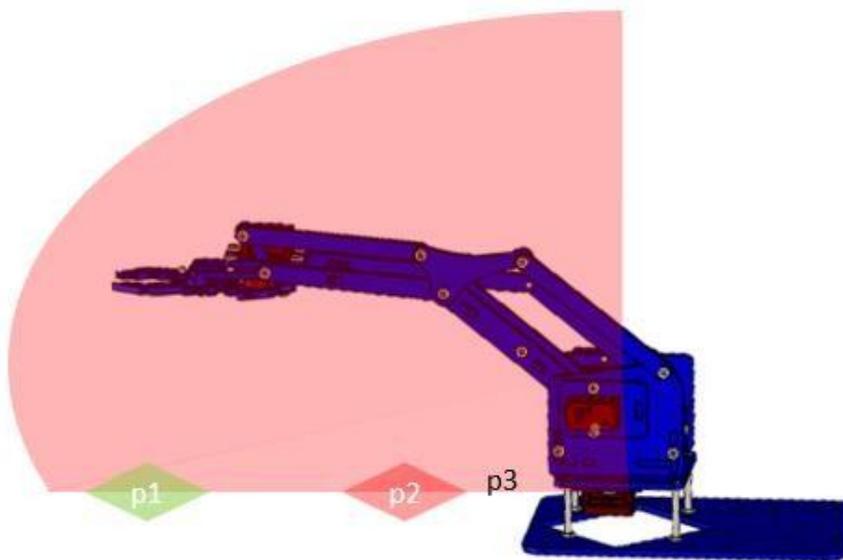


Figure VI.4 : Environnement de travail vu de droite

VI.4. Partie I : Commande manuelle du bras manipulateur par potentiomètres

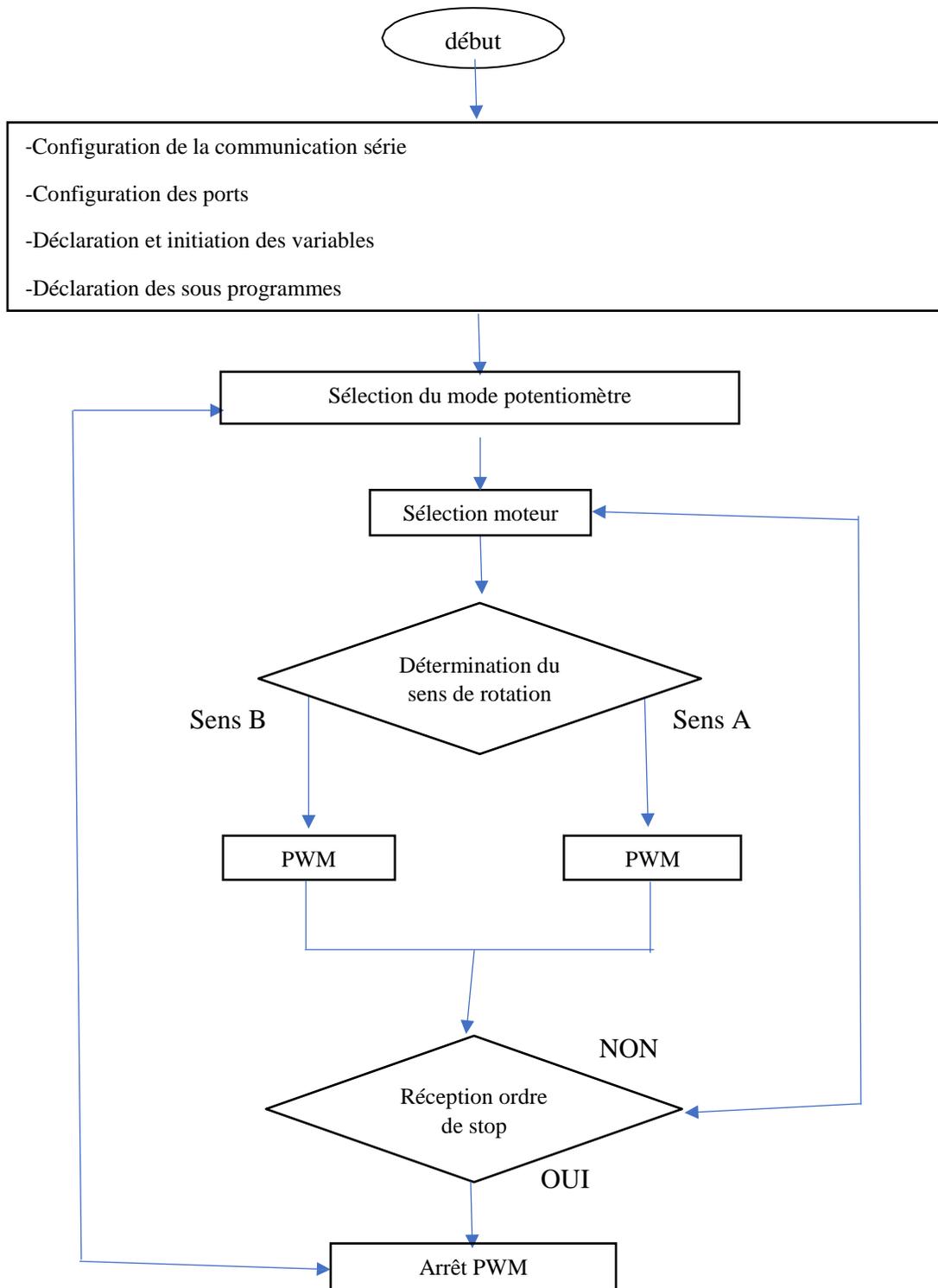


Figure VI.5 : Organigramme de l'algorithme de commande manuelle du robot

Dans cette partie nous allons décrire le programme qui permet de contrôler le bras manipulateur par l'utilisation de potentiomètres. Cette commande manuelle permet à l'opérateur de contrôler avec précision le robot afin d'effectuer des tâches difficiles ou bien dans l'idée d'apprentissage du robot.

La figure VI.5 nous montre l'organigramme du programme de commande manuelle du bras manipulateur avec potentiomètre. Ce programme a été transcrit en langage Arduino (voir Annexe B) et en le téléchargeant dans la mémoire du microcontrôleur de la carte programmable Arduino UNO, il permet au Shield Arduino qui contient les potentiomètres de contrôler les servomoteurs du bras manipulateur suivant la connexion préétablie. Au final, ce programme permet de faire bouger le bras manipulateur et la pince en fonction des variables articulaires (4 degrés de liberté).

Afin d'établir une commande manuelle au bras manipulateur, on a créé un programme qui permet de le commander par le biais de potentiomètres.

La connexion entre la carte électronique et l'environnement de programmation se fait via une liaison USB. On commence, tout d'abord, par déclarer la bibliothèque <Servo. H>, qui est une fonction prédéfinie qui nous permet d'utiliser les servomoteurs dans notre programme. On a utilisé la fonction *analogRead ()* qui permet de lire les entrées analogiques. Elle est définie comme suit :

```
Valpot1 = analogRead (pot1);
```

On a également utilisé la fonction *map ()* qui permet la conversion (ou le redimensionnement) des valeurs lues pour les adapter aux valeurs des servomoteurs (0 : 180).

```
ValPot1 = map (ValPot1, 0 , 1023 , 0 , 108 );
```

La transcription de l'organigramme en langage Arduino suit les étapes suivantes :

Début

1-Faire appel à la bibliothèque Servo

2-Définir les servomoteurs

3-Définir les potentiomètres

4-Entre les variables utiliser pour lecteur la broché analogique potentiomètres par la commande :

int valpot1 ;

5-Attacher les servomoteurs Aux pince qui fournissent PWM

6-Lire la valeur de potentiomètre (valeur entre 0 et 1023) par la commande :

valpot1 =analogRead (pot1)

7-On a également utilisé la fonction map() qui permet la conversion (ou le

Redimensionnement) des valeurs lues pour les adapter aux valeurs des servomoteurs (0 : 180).

ValPot1 = map (ValPot1, 0 , 1023 , 0 , 108) ;

Fin

VI.4. Partie II : Commande à distance du bras manipulateur par Bluetooth et application Android

Afin d'établir une commande sans fil qui nous donnera un contrôle à distance pour notre réalisation, on a utilisé un récepteur Bluetooth. On a également conçu une application mobile Android grâce à l'outil de développement créé par Google **App Inventor** pour assurer la communication entre l'Arduino et l'interface réalisée.

Réalisation de l'application Android

La réalisation de l'application sous **App Inventor** se divise en deux parties : la partie design pour le graphique et la partie bloc pour la programmation.

VI.4.1. MIT Inventor :

C'est une application développée par Google. Il est actuellement maintenu par le Massachusetts Institute of Technology (MIT).

Il simplifie le développement d'applications sur Android et le rend disponible même pour les débutants et ceux qui ne sont pas familiers avec les langages de programmation. Il est basé sur une interface graphique similaire à Scratch et à l'interface Star Logo TNG (en). Avec son interface entièrement graphique et l'absence totale de ligne de code, il est particulièrement adapté aux enfants qui commencent à coder, dès l'école primaire.

VI.4.1.1. Design de l'interface de l'application :

La figure VI.6 montre l'interface du logiciel Web App Inventor qui permet de créer les interfaces graphiques des applications sous Android. Le logiciel est très simple d'utilisation avec des fonctionnalités très intuitives.

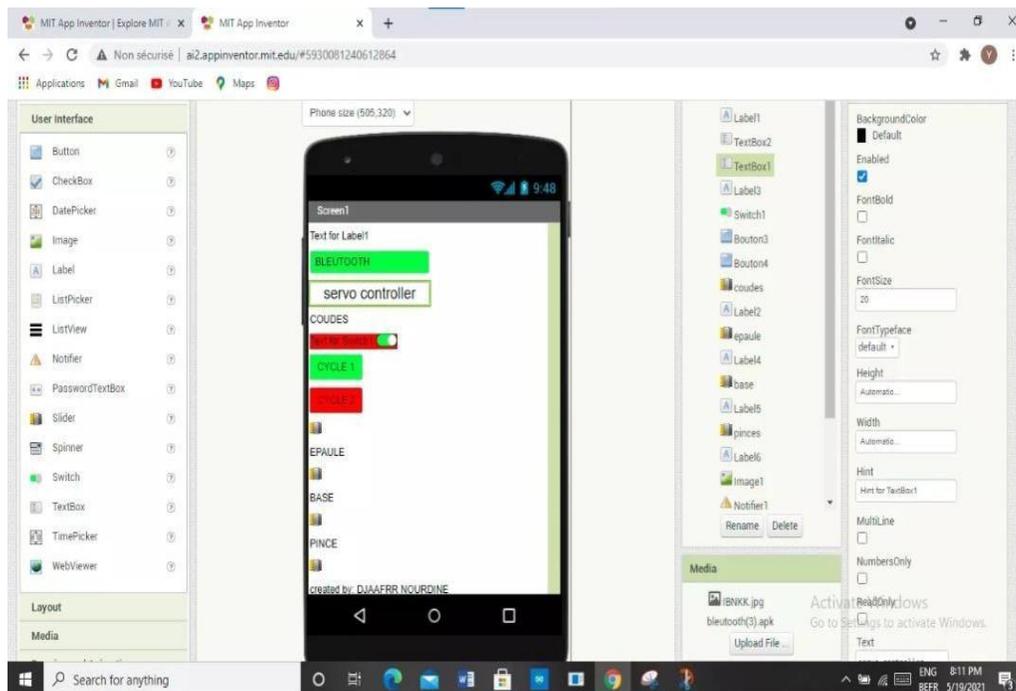


Figure VI.6 : Interface qui permet le design de notre application sous App Inventor

Le résultat final obtenu à savoir l'icône et l'interface finale de l'application sont illustrés dans la figure VI.7. Pour utiliser l'application, il faut établir la connexion Bluetooth avec

le module HC6 de la carte Arduino et par la suite le contrôle des servomoteurs du bras manipulateurs s'effectue grâce aux jauges suivant leurs appellations respectives.

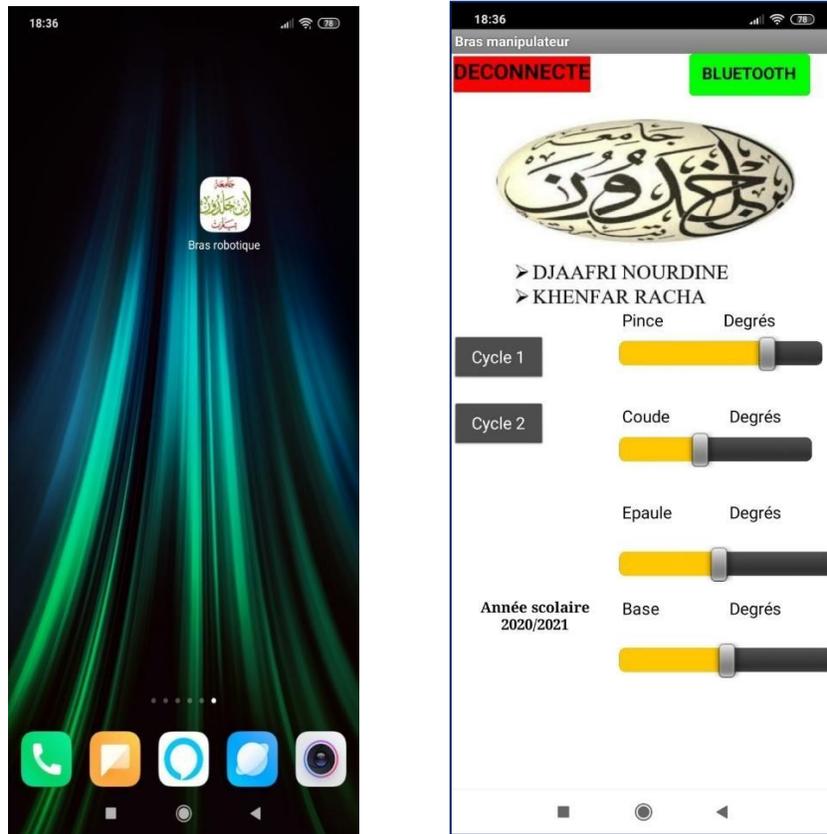


Figure VI.7 : Interface finale de l'application

VI.4.1.2. Programmation par block de notre application Android :

On utilise les blocks pour programmer cette application en utilisant le logiciel web MIT Inventer. Dans la partie bloc, on retrouve toutes les fonctions mathématiques, logiques, ou Evènementielles, ce rapportant aux objets utilisés dans la partie graphique. Cela nous permet de tirer profit d'une simplicité de programmation orientée objet sous la forme de blocs Préprogrammés et qu'ils ne restent plus qu'à organiser en fonction du programme souhaité. Lorsque ces outils sont utilisés correctement, on peut réaliser une application simple et Efficace, avec un design propre qui répond à nos besoins. Dans ce qui suit, on va expliquer le Fonctionnement des blocs de programme et leurs utilités dans le programme Arduino pour le Contrôle du bras.

Pour commencer, on définit les éléments de l'objet Bluetooth de type Bluetooth Liste (comme les périphériques Bluetooth disponibles). Ensuite on établit la connexion entre le périphérique et l'application.



Figure VI.8 : Les blocs de programmation application via Bluetooth

Ce sont les blocs qui contrôlent les quatre servomoteurs du bras manipulateur (pince, coude, épaule et base), voir figure VI.8. Si on change la position du curseur, en utilisant la fonction *Bluetooth.SendText*, on envoie un texte à l'Arduino. Ce texte se compose d'un préfixe qui indique quel curseur a été modifié (le préfixe « A » pour la pince jusqu'au

préfixe « D » pour la base) ainsi que la valeur actuelle du curseur qui est *thumbPosition*. [64]

VI.4.1.3. Programme Arduino :

La figure VI.9 représente l'organigramme de l'algorithme du programme de commande à distance avec Bluetooth du bras manipulateur. Le programme attend un message à travers la connexion Bluetooth, ce message ou préfixe va déclencher un sous-programme correspondant au servomoteur concerné. Le sous-programme indique la valeur de l'angle que va prendre l'articulation du bras manipulateur. (voir programme Annexe B)

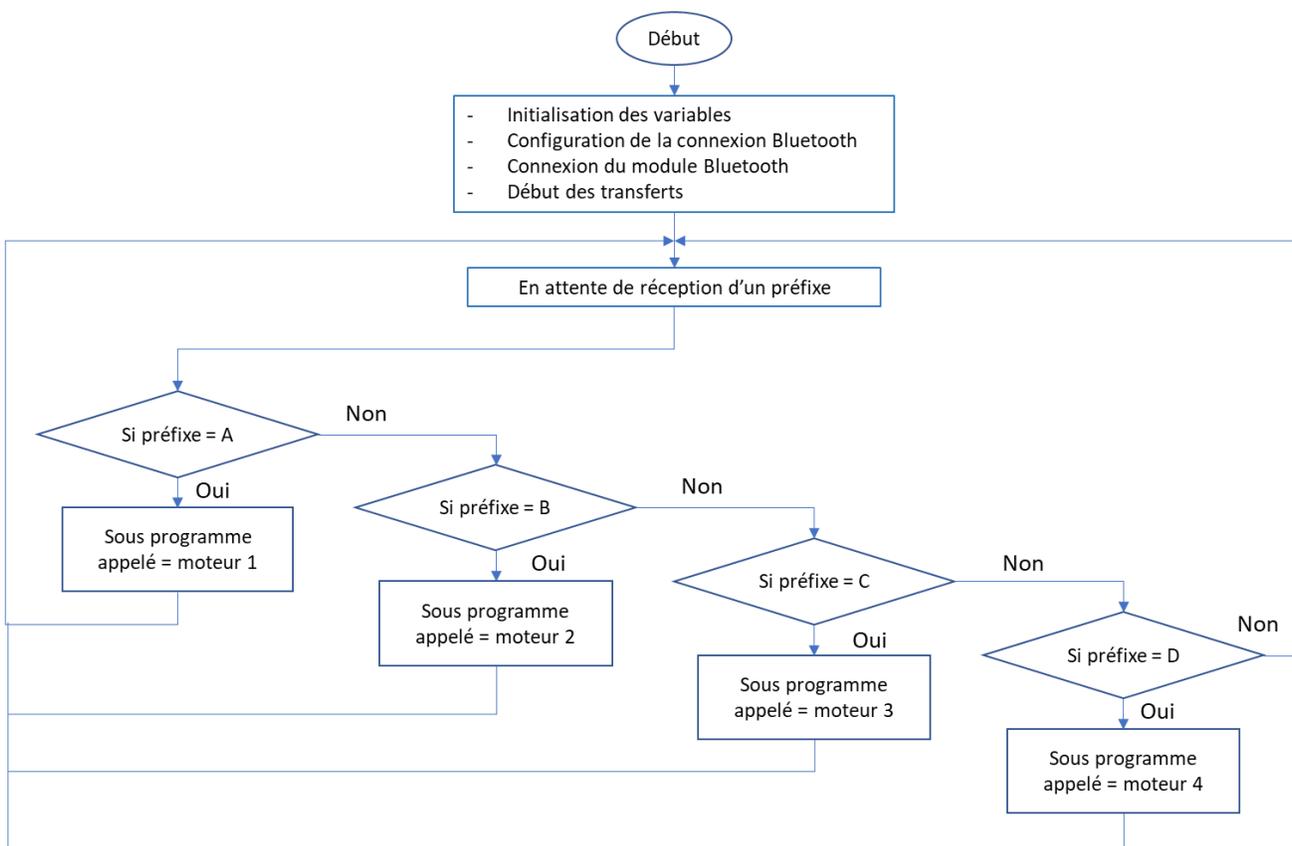


Figure VI.9 : Organigramme de l'algorithme de commande à distance du robot

VI.5. Application « Pick-and-Place » :

Les robots « Pick and Place » sont devenus monnaie courante dans l'environnement de fabrication d'aujourd'hui. Généralement relégués à des tâches simples, répétitives et

monotones dans lesquelles les robots excellent naturellement, les robots de sélection et de placement présentent un certain nombre d'avantages pour les fabricants.

Les robots « Pick and Place » sont généralement montés sur un support stable, stratégiquement positionné pour atteindre toute leur enveloppe de travail. Les systèmes de vision avancés leur permettent de saisir et de déplacer des objets sur une bande transporteuse, qui peut être utilisée de différentes manières.[65]

VI.5.1. Algorithme de la commande « Pick-and-Place » :

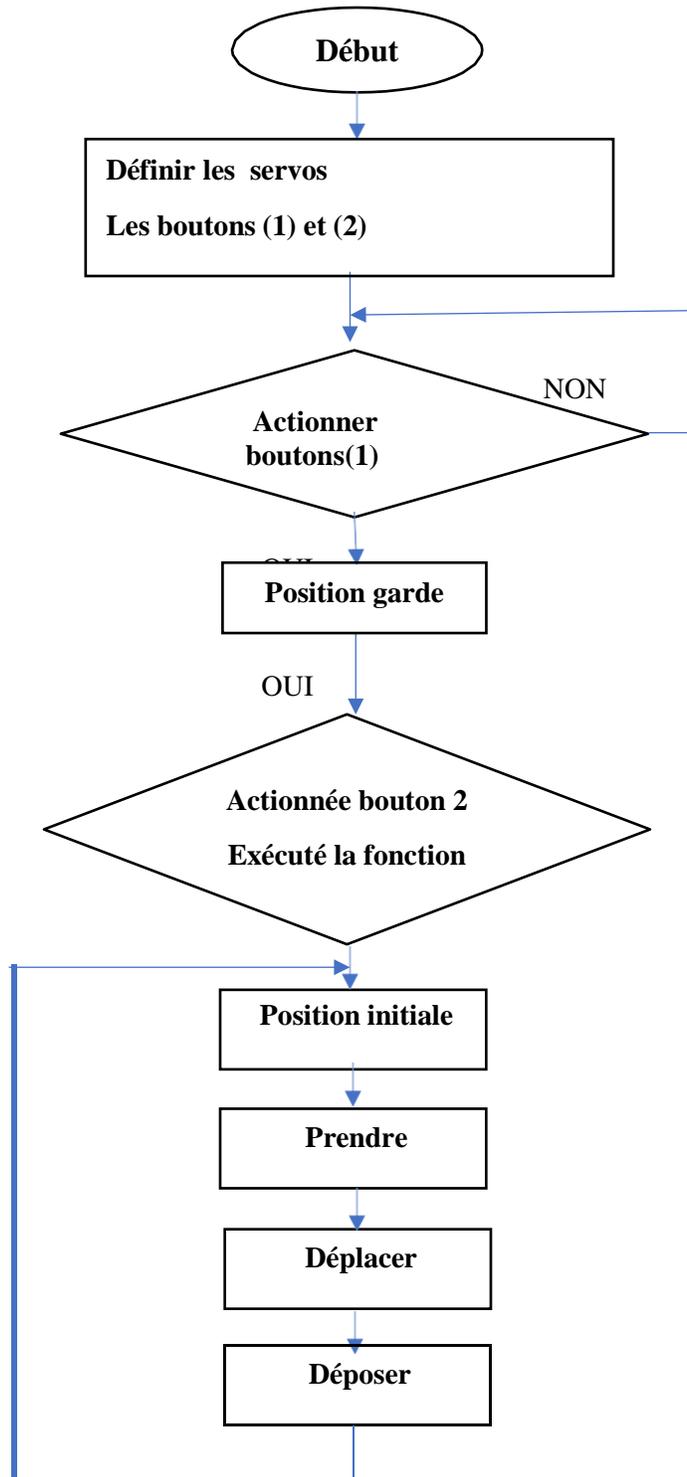


Figure VI.10 : Organigramme de l'algorithme de commande « Pick-and-Place » du robot

La figure VI.10 nous montre l'organigramme de l'algorithme de commande « Pick-and-Place » du bras manipulateur. L'idée de ce programme s'inspire du travail de Master de Mr Sahir et Mr Bouatmane [64] et qui réside dans l'utilisation de deux boutons :

1. Un pour la sauvegarde des positions
2. L'autre pour lancer le cycle automatique sauvegardé

VI.5.2. Programmation par block de l'application Android :

On ajoute au programme précédent les blocs des deux boutons. Ces derniers envoient un texte à l'Arduino. Ce texte se compose d'un l'état du bouton associé qui indique quelle action doit être exécutée.

Les deux boutons permettent d'exécuter une série d'instructions différentes. Celles-ci forment un cycle automatique, et peuvent être modifiés selon le besoin.

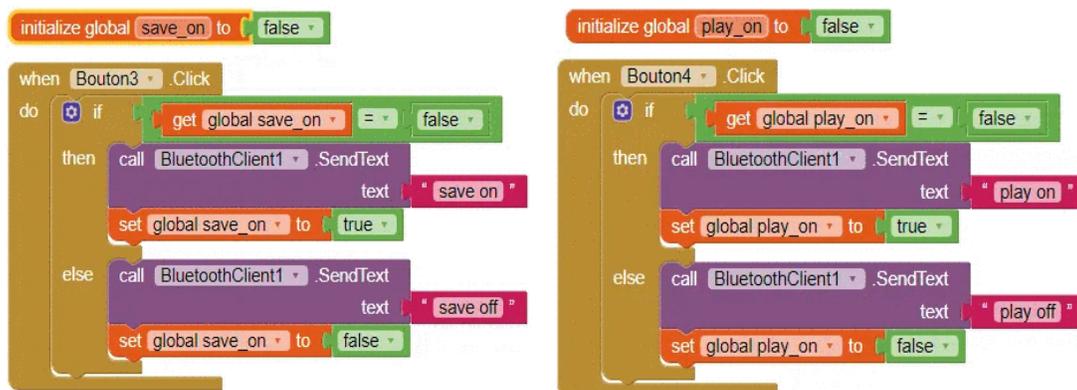


Figure VI.11 : les blocs de deux boutons sauvegarder et Lancer le cycle automatique

VI.6. Conclusion :

Dans ce dernier chapitre, nous avons implémenté plusieurs programmes pour faire fonctionner notre bras manipulateur en mode manuel ainsi qu'en mode distant. Nous avons présenté le logiciel App Inventor qui nous a permis de développer l'application Android qui communique avec notre Arduino à travers le module Bluetooth HC6.

Le dernier programme « Pick-and-Place » n'a pas fonctionné d'une manière stable et cela est dû à plusieurs paramètres, nous n'avons pas eu le temps nécessaire pour résoudre le problème de ce dernier vu le temps court qui nous a été donné pour finaliser notre mémoire.

CONCLUSION GÉNÉRALE

CONCLUSION GENERALE

Ce mémoire présente notre travail qui concerne l'étude et la réalisation d'un bras manipulateur à quatre degrés de liberté.

Après une exploration générale du domaine de la robotique, nous avons abordé le vif du sujet en étudiant les différents organes constituant notre bras manipulateur.

Par la suite, nous nous sommes penchés sur la modélisation avec le logiciel Solidworks du modèle CAO du robot et l'élaboration des modèles géométrique, cinématique et dynamique, qui ont pour rôle respectivement de générer des trajectoires de références, d'analyser les singularités et de commander et simuler le robot. Nous avons aussi présenté la possibilité de réaliser les pièces du bras manipulateur en impression 3D qui commence à se généraliser.

Enfin, nous avons procédé à l'étude pratique et l'implémentation des techniques de commande prévues sur notre réalisation à savoir la commande manuelle par potentiomètres, et la commande à distance avec application mobile et Bluetooth. La programmation a été faite à base du langage d'Arduino avec son environnement de développement spécifique.

Notre travail sur la thématique de la robotique nous a permis d'approfondir certaines notions abordées lors de notre formation mais qui n'ont pas pu être développées à cause du manque de temps et de matériels. Le fait d'avoir étudié le robot depuis la modélisation 3D jusqu'à la programmation des tâches de la commande avec la carte Arduino, nous a fait voir la difficulté à mettre en place un système robotisé et cela dans un contexte de formation didactique. Cette dimension aura encore plus d'importance dans un contexte industriel. Mais cela reste une expérience très enrichissante.

A la fin de notre mémoire, nous avons eu des problèmes de temps et de défaillance de matériels qui nous a empêché d'atteindre notre objectif à 100%, à savoir mettre en place le programme « Pick-and-place ». Nous espérons que notre travail aura une suite dans le futur proche pour explorer d'autres aspects de la robotique industriel.

RÉFÉRENCES
BIBLIOGRAPHIQUES

Références bibliographiques et Web-graphique

- [1] Latetitia Matignon INTRODUCTION A LA ROBOUOTIQUE université de cean france 2011/2012 [1]
- [2] cour robotique université Angers jean-Louis Boimond [2]
- [3] <https://mya-1997.wixsite.com/association-jsb/robotique> Association de Jeunes et Sciences Bousalem [3]
- [4] Bra manipulateur: journal GEEK Wire 5 juin 2019 photographe Alan Boyle[9]
- [5] company Direct industry by virtualexpo group
- [6] Ait dahmane kahina ,ait ziane meziane ; conception et realisation d'un bra manipulateur commande par API UNIVERSITE KHEMIS MELIANA option genie electrique 2014 / 2015_[11]
- [7] Cour robotique_ISTIA, Université Angers Jean-Louis Boimond[12]
- [8] Cour de robotique chapitre 1 moodle universite ibn khaldoune tiaret [13] [14]
- [9] la societe fanuc site de web : Fanuc.eu/pl/en/robots[15]
- [10] JOURNAL OF ROBOTICS M.J.MAHMOUD ABADI ARTICLE : inverse dynamics Based optimal fuzzy controller for a robot MANIPULATOR Via Particle Swarm Optimization 1 JANVIER 2019
- [11] <https://new.abb.com/products/robotics/industrial-robots/irb-2400/irb-2400-data> (visité la dernière fois 01/07/2021)
- [12] BOUAGAL FAWZI “commande vocale d’un robot manipulateur “ université de batna faculté de technologie department electronique 2013
- [13] <https://www.staubli.com/fr/> (visité la dernière fois 01/07/2021)
- [14] <http://dictionnaire.sensagent.leparisien.fr/Bras%20manipulateur/fr-fr/#anchorWiki>
- [15] conception et réalisation d’un bras manipulateur commande par API université khmis melianna réalisé par ait dahmene kahina et ait ziane meriane 2014-2015

[16] étude et réalisation d'un bras de robot a base de carte Arduino université A.Mira bejaia présenté par sahir koceila et bouatmane hakim 2018-2019

[17] http://techno.pvm.free.fr/images/Cours/3eme/4.DEFI_Robot/Dossier_Mbot.pdf

[18] étude et réalisation d'un bras de robot a base de carte Arduino université A.Mira bejaia présenté par sahir koceila et bouatmane hakim 2018-2019

[19] conception et réalisation d'un bras manipulateur a 3 degré de liberté université saad dahleb blida présenté par achour halim et mohamed seghir mounir 2016-2017

[20] livre **BUILD YOUR OWN COMBAT ROBOT** Pete miles Tom carroll page .243

[21] Dr. Pushkin kachroo Chair, Dr. Alynn abbott Dr. Hugh vanlandingham feedback control for a phath following robotic car April 2, 2002 blacksburg, virginia page 15-16

[22] https://www.academia.edu/8239651/1_Dossier_ARDUINO

[23] https://www.academia.edu/8239651/1_Dossier_ARDUINO

[24] <https://www.cours-gratuit.com/cours-arduino/introduction-a-la-carte-arduino-uno-tutoriel-pdf>

[25] <https://arduino.developpez.com/tutoriels/arduino-a-l-ecole/?page=projet-12-utiliser-un-servomoteur>

[26] <https://www.cours-gratuit.com/cours-arduino/arduino-bluetooth-module-hc-05-pdf-eng>

[27] <https://www.aranacorp.com/fr/votre-arduino-communique-avec-le-module-hc-05/>

[28] <https://www.aranacorp.com/fr/votre-arduino-communique-avec-le-module-hc-05/>

[29] <https://www.oh-kirei.com/23/un-programme-arduino-bluetooth.php>

[30] <https://www.techno-science.net/definition/6737.html>

[31] <https://www.generationrobots.com/fr/174-shield-arduino>

[32] « Cours robotique » cours jeans_louis boimond université Angers page 19

[33] « Cours robotique » cours jeans_louis boimond université Angers page 10

[34] <https://docer.pl/doc/xc0xs01>

[35] <https://docer.pl/doc/xc0xs01>

[36] <https://sciences-indus-cpge.papanicola.info/MEC2-Chaines-cinematiques>

[37] « Cours robotique » cours jeans_louis boimond université Angers page 12

[38] [BOKRETA Manal. GUELMAME Djamila], Étude cinématique d'un bras manipulateur et simulation des mouvements sur logiciel SOLIDWORKS. Étude de cas : Bras manipulateur RPP. Mémoire de master 2 UNIVERSITÉ IBN-KHALDOUN DE TIARET 2015-2016 page 2_3 chapitre2

[39] [José GUTIERREZ] , Modélisation et Identification géométrique de robots utilisés pour des opérations d'usinage. Thèse UNIVERSITE BLAISE PASCAL - CLERMONT II année 2016 page 41 -42-43

[40] [José GUTIERREZ] , Modélisation et Identification géométrique de robots utilisés pour des opérations d'usinage. Thèse UNIVERSITE BLAISE PASCAL - CLERMONT II année 2016 page 44 -45-46

[41] [BOKRETA Manal. GUELMAME Djamila], Étude cinématique d'un bras manipulateur et simulation des mouvements sur logiciel SOLIDWORKS. Étude de cas : Bras manipulateur RPP. Mémoire de master 2 UNIVERSITÉ IBN-KHALDOUN DE TIARET 2015-2016 page 2_3 chapitre2

[42] [BOKRETA Manal. GUELMAME Djamila], Étude cinématique d'un bras manipulateur et simulation des mouvements sur logiciel SOLIDWORKS. Étude de cas : Bras manipulateur RPP. Mémoire de master 2 UNIVERSITÉ IBN-KHALDOUN DE TIARET 2015-2016 page 14 chapitre2

[43] [BOKRETA Manal. GUELMAME Djamila], Étude cinématique d'un bras manipulateur et simulation des mouvements sur logiciel SOLIDWORKS. Étude de cas : Bras manipulateur RPP. Mémoire de master 2 UNIVERSITÉ IBN-KHALDOUN DE TIARET 2015-2016 page 15 chapitre2

[44] Article scientifique : linear robotic solutions Pick and Place Robots: What Are They Used For and How Do They Benefit Manufacturers ? Robotics Online Marketing Team | 03/13/2018

ANNEXE A :
Mises en plan des
pièces du bras
manipulateur

4 3 2 1

F

F

E

E

D

D

C

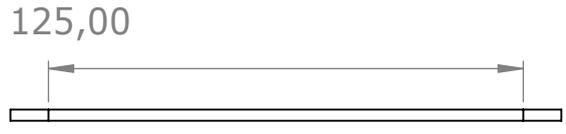
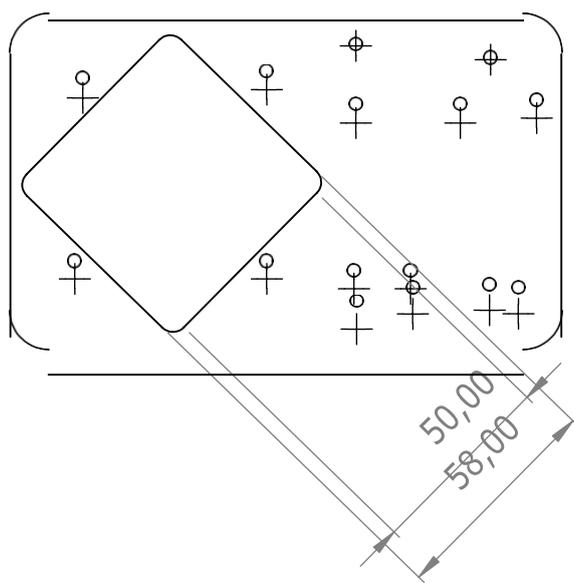
C

B

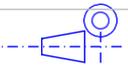
B

A

A



Echelle 1:2



A4

4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 002

3

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

1

4

3

2

1

4

3

2

1

F

F

E

E

D

D

C

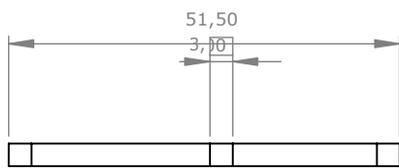
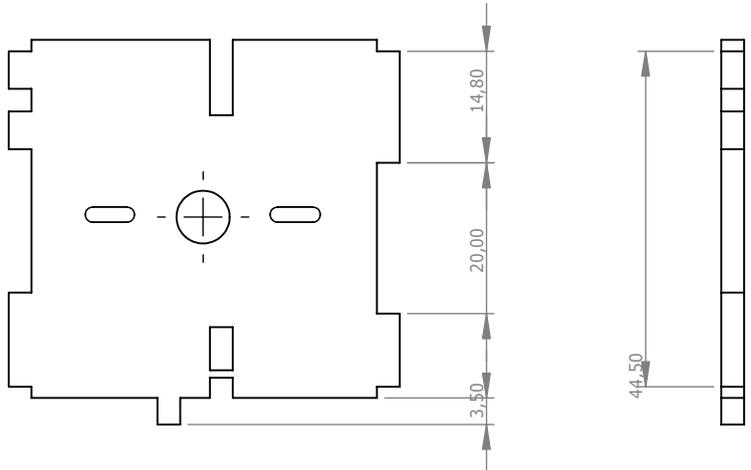
C

B

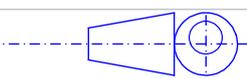
B

A

A



Echelle 1:1



Département de Génie Mécanique
Université Ibn Khaldoun

DJAAFRI
NOUREDDINE
KHENFAR RACHA

A4

pièce 003

26/06/2021

4

3

2

1

4

3

2

1

F

F

E

E

D

D

C

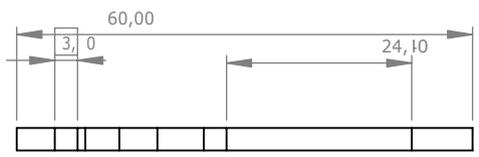
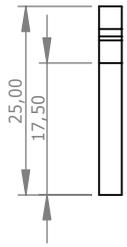
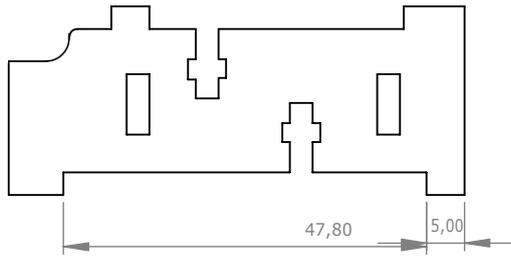
C

B

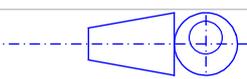
B

A

A



Echelle 1:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 004

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4

3

2

1

4

3

2

1

F

F

E

E

D

D

C

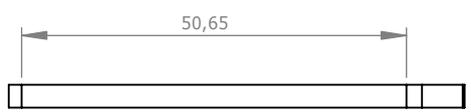
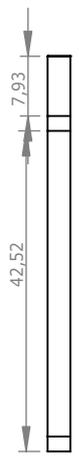
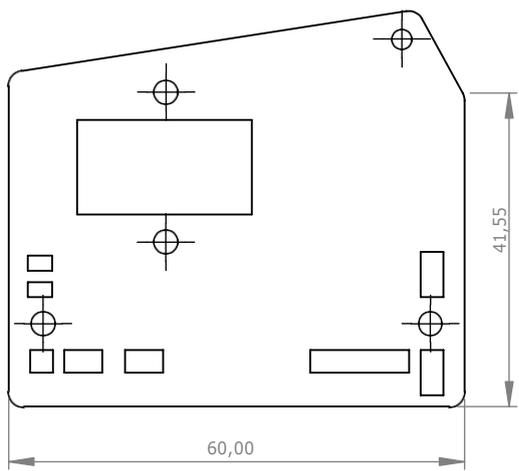
C

B

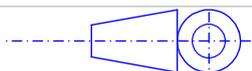
B

A

A



Echelle 1:1



Département de Génie Mécanique
Université Ibn Khaldoun

DJAAFRI
NOUREDDINE
KHENFAR RACHA

A4

pièce 005

26/06/2021

4

3

2

1

4 3 2 1

F

F

E

E

D

D

C

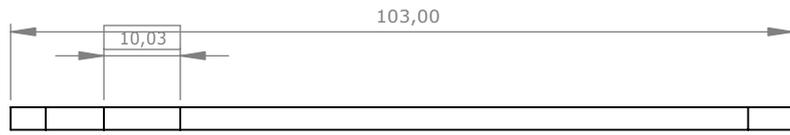
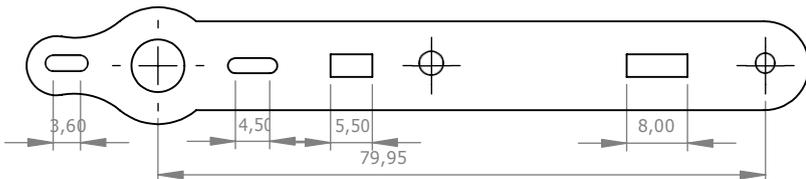
C

B

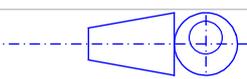
B

A

A



Echelle 1:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 006

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4 3 2 1

4

3

2

1

F

F

E

E

D

D

C

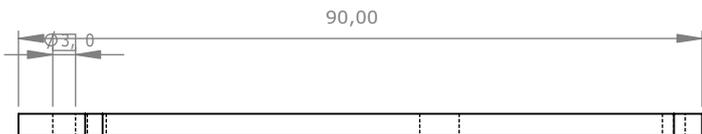
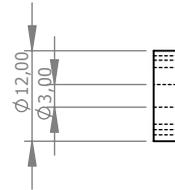
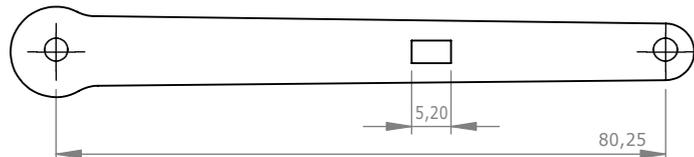
C

B

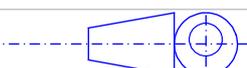
B

A

A



Echelle 1:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 007

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4

3

2

1

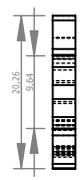
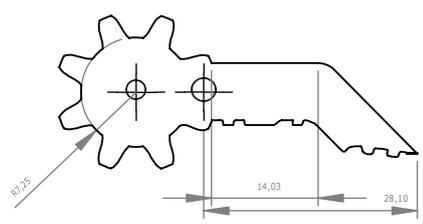
4 3 2 1

F

F

E

E

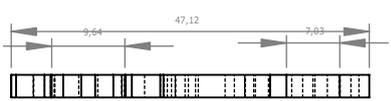


D

D

C

C



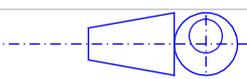
B

B

A

A

Echelle 1:1



Département de Génie Mécanique
Université Ibn Khaldoun

DJAAFRI
NOUREDDINE
KHENFAR RACHA

A4

pièce 008

26/06/2021

4 3 2 1

4

3

2

1

F

F

E

E

D

D

C

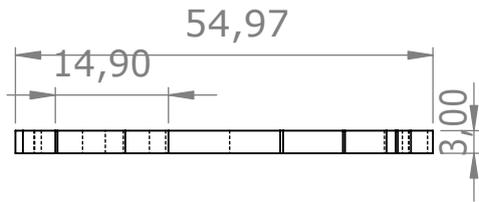
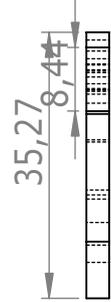
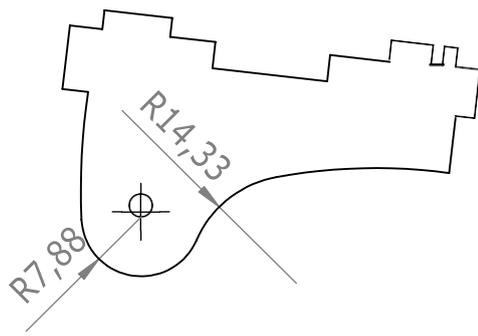
C

B

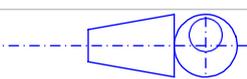
B

A

A



Echelle 1:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 010

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4

3

2

1

4

3

2

1

F

F

E

E

D

D

C

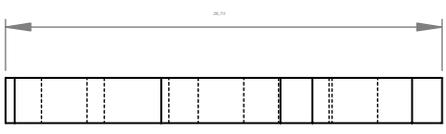
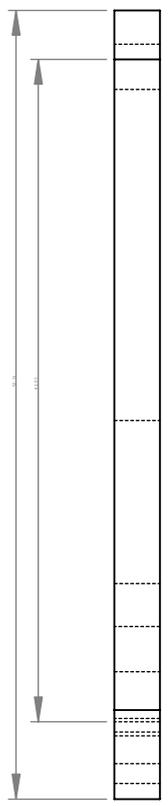
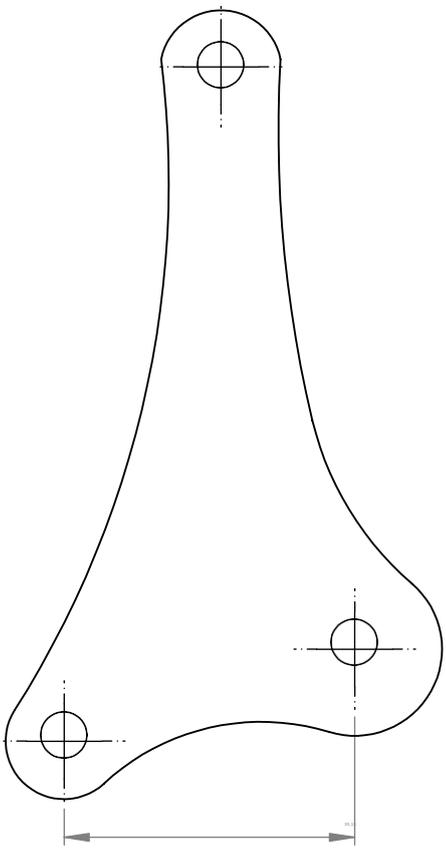
C

B

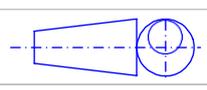
B

A

A



Echelle 1:1



A4

4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 011

3

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

1

2

4

3

2

1

F

F

E

E

D

D

C

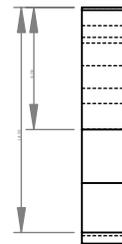
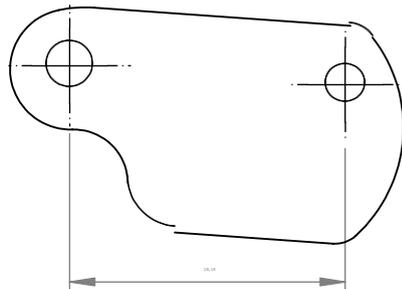
C

B

B

A

A



Echelle 1:1



A4

4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 012

3

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

2

1

4 3 2 1

F

F

E

E

D

D

C

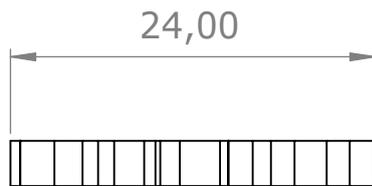
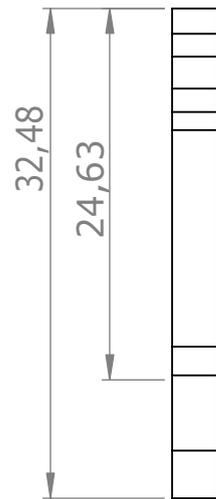
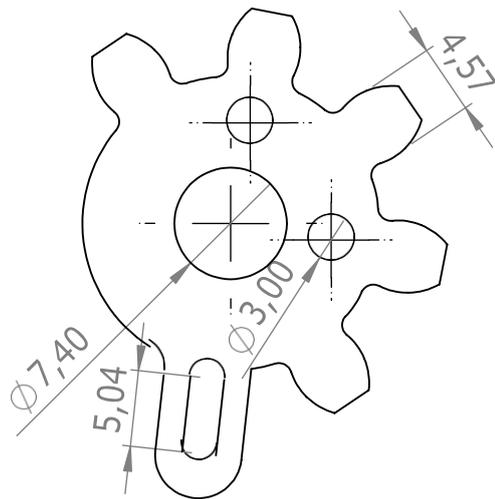
C

B

B

A

A



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 013

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4 3 2 1

4 3 2 1

F

F

E

E

D

D

C

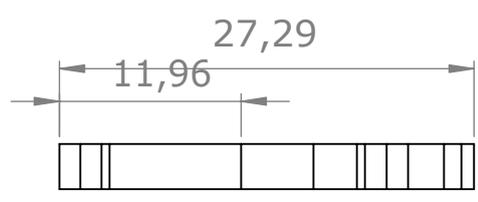
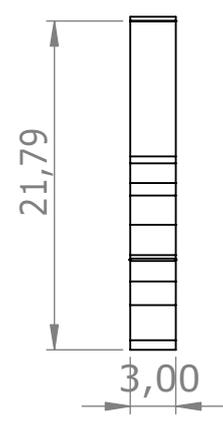
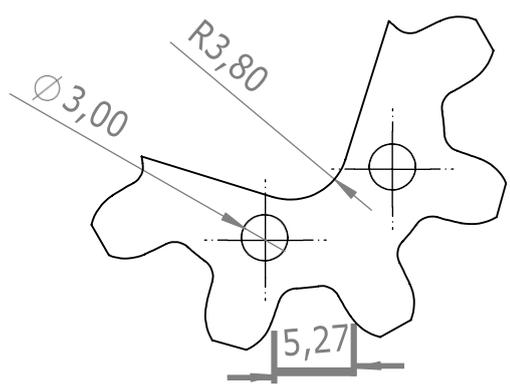
C

B

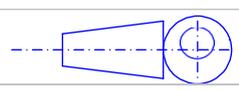
B

A

A



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 014

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4 3 2 1

4 3 2 1

F

F

E

E

D

D

C

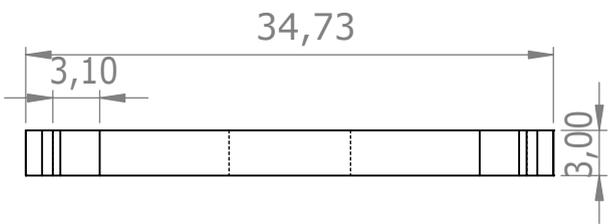
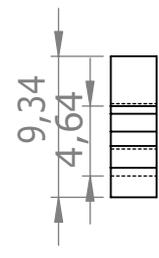
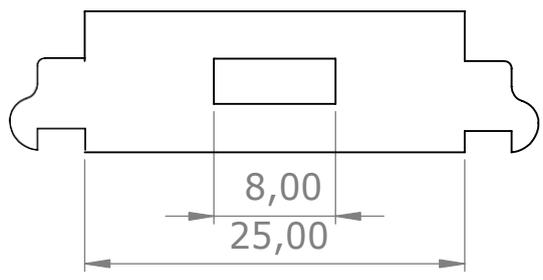
C

B

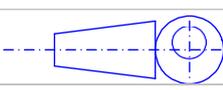
B

A

A



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 015

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4 3 2 1

4 3 2 1

F

F

E

E

D

D

C

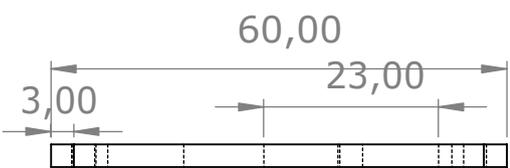
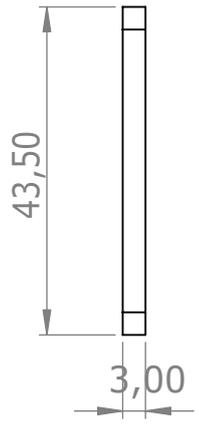
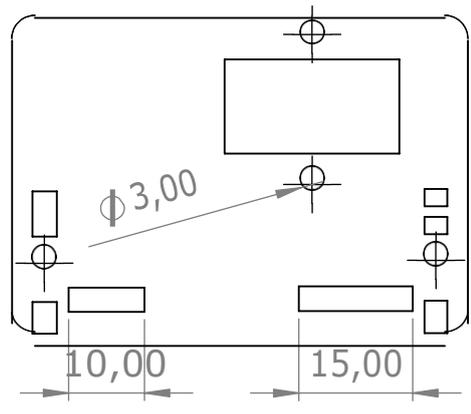
C

B

B

A

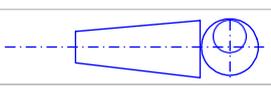
A



Echelle 2:1

Département de Génie Mécanique
Université Ibn Khaldoun

DJAAFRI
NOUREDDINE
KHENFAR RACHA



A4

pièce 016

26/06/2021

4 3 2 1

4

3

2

1

F

F

E

E

D

D

C

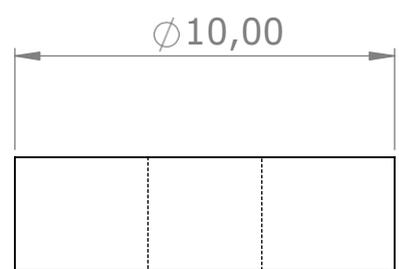
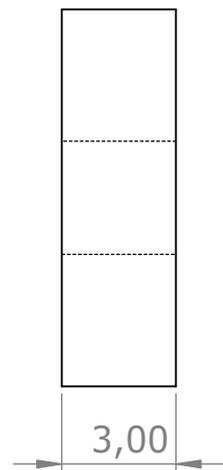
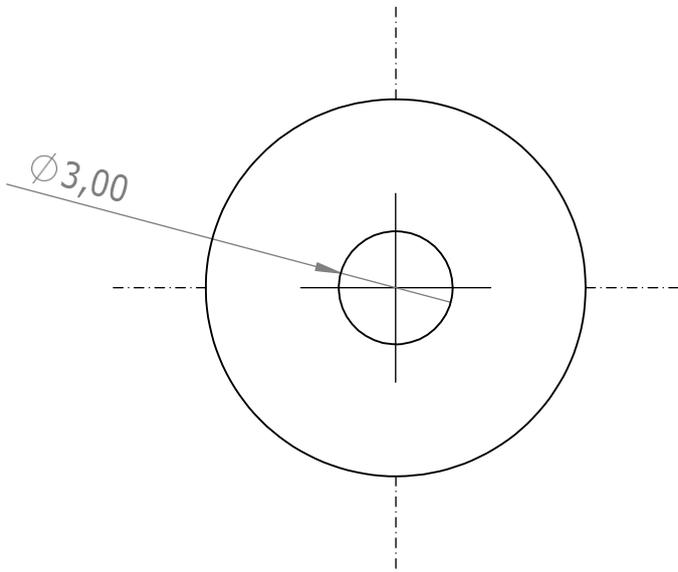
C

B

B

A

A



Echelle 5:1



Département de Génie Mécanique
Université Ibn Khaldoun

DJAAFRI
NOUREDDINE
KHENFAR RACHA

A4

pièce 017

26/06/2021

4

3

2

1

4

3

2

1

F

F

E

E

D

D

C

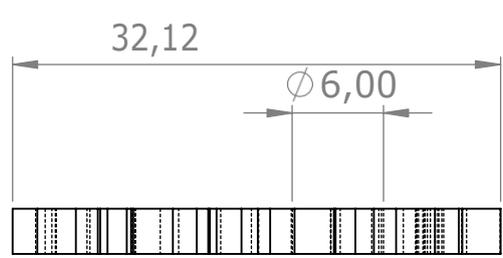
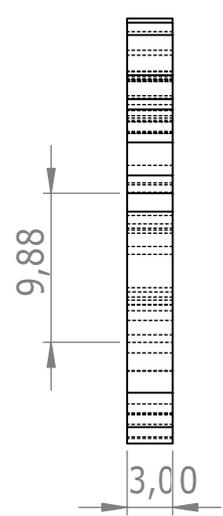
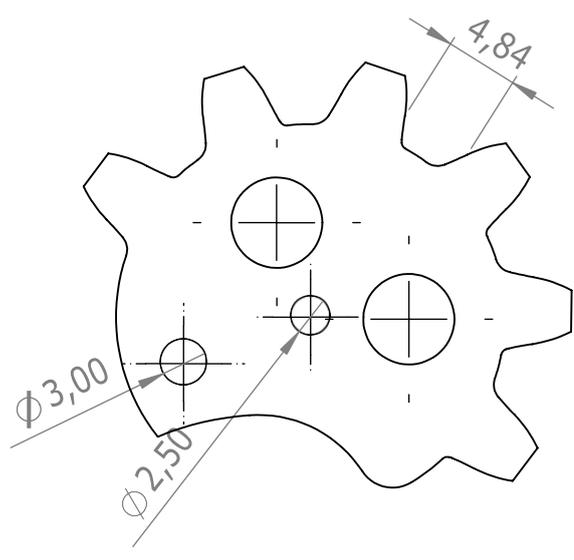
C

B

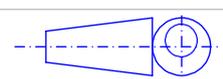
B

A

A



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 018

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4

3

2

1

4

3

2

1

F

F

E

E

D

D

C

C

B

B

A

A

Echelle 2:1

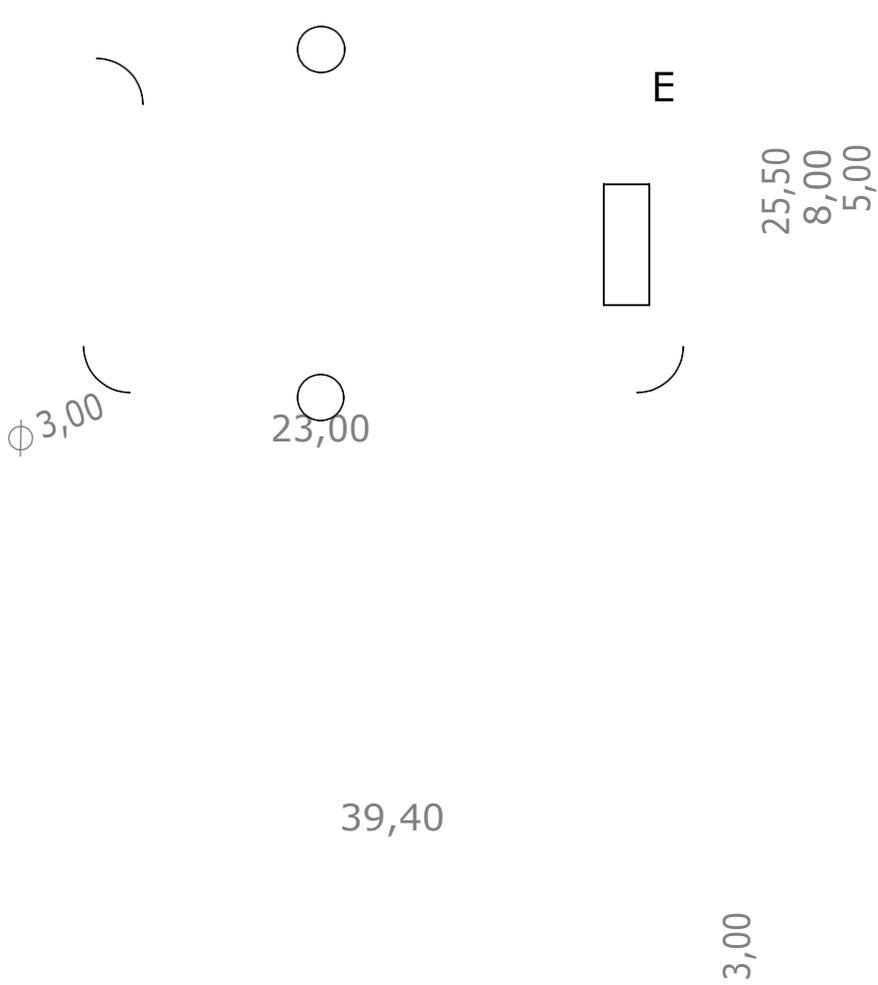
Département de Génie Mécanique
Université Ibn Khaldoun

DJAAFRI
NOUREDDINE
KHENFAR RACHA

A4

pièce 019

26/06/2021

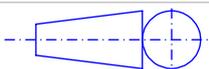
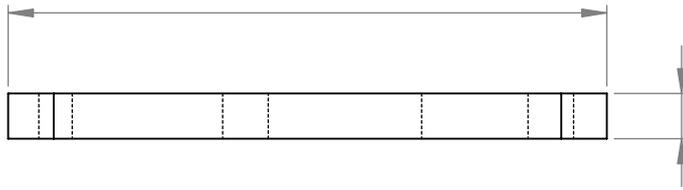
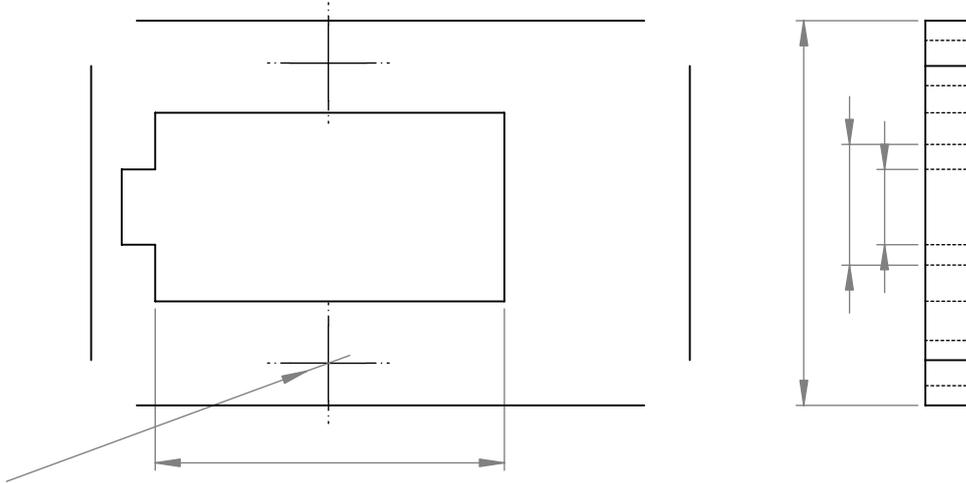


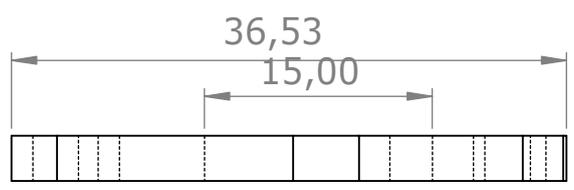
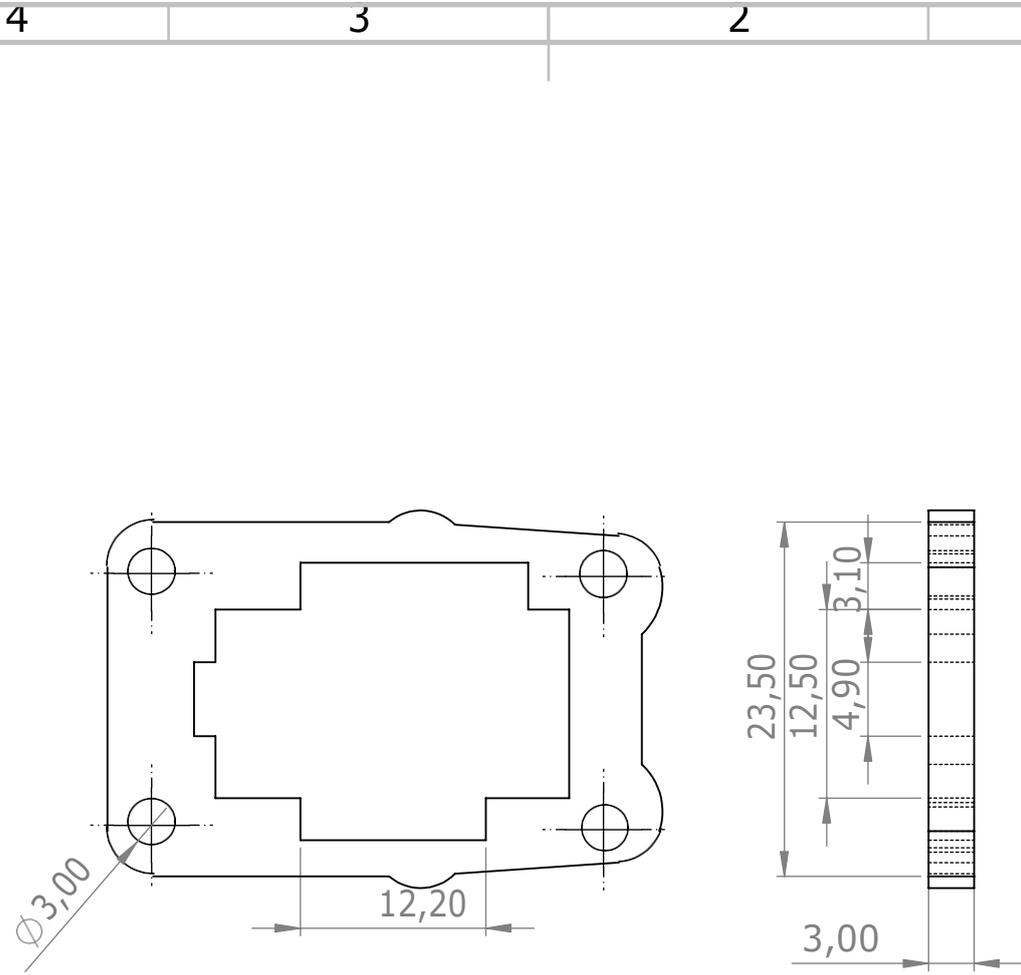
4

3

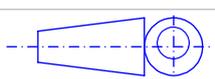
2

1





Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 020

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

F
E
D
C
B
A

F
E
D
C
B
A

4 3 2 1

4 3 2 1

4 3 2 1

F

F

E

E

D

D

C

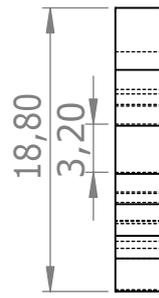
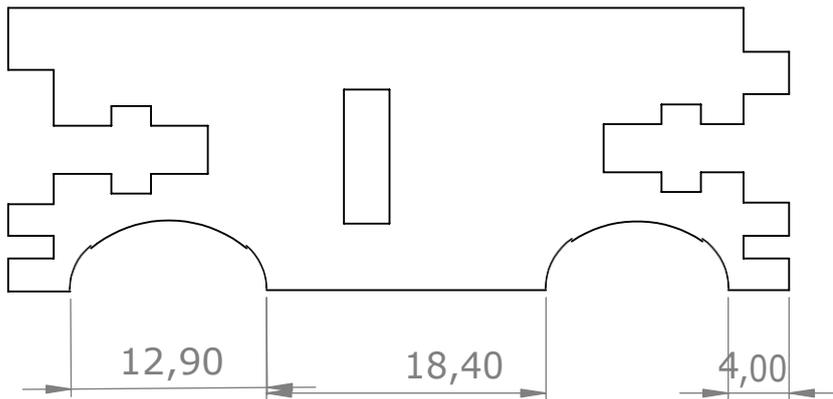
C

B

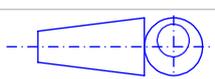
B

A

A



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 021

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4 3 2 1

4

3

2

1

F

F

E

E

D

D

C

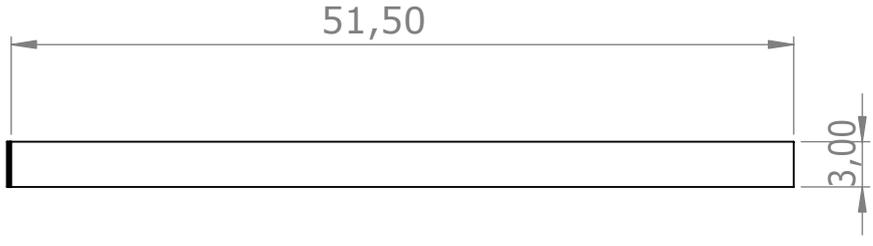
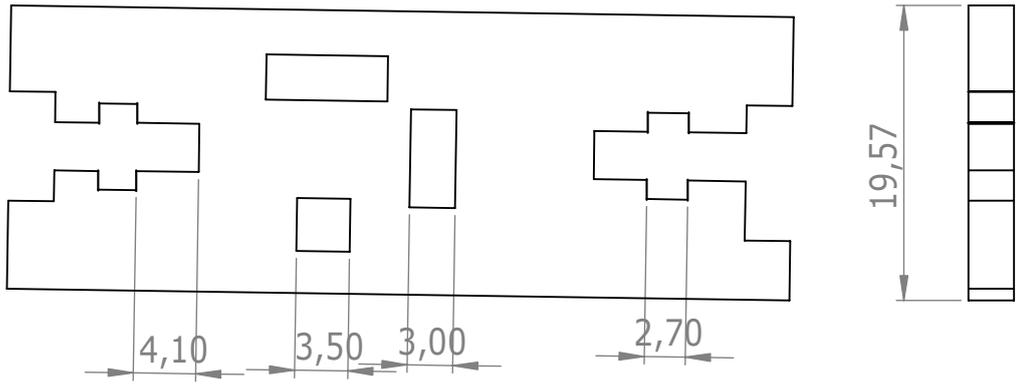
C

B

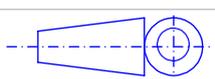
B

A

A



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 022

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4

3

2

1

4 3 2 1

F

F

E

E

D

D

C

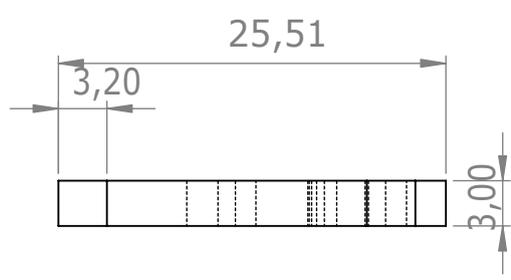
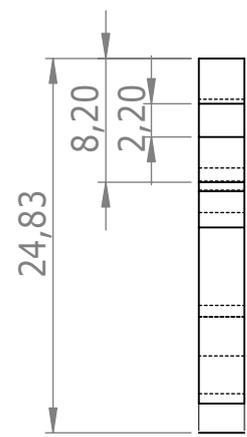
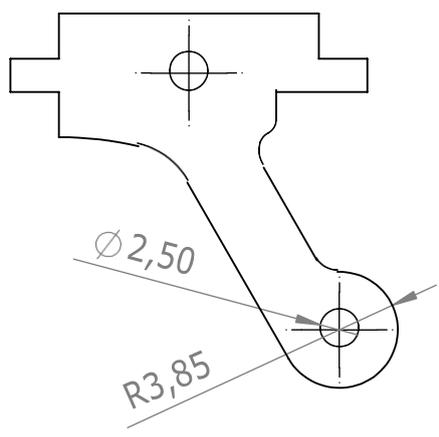
C

B

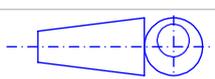
B

A

A



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 023

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4 3 2 1

4

3

2

1

F

F

E

E

D

D

C

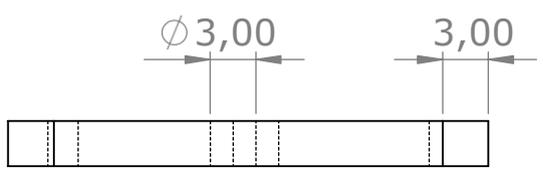
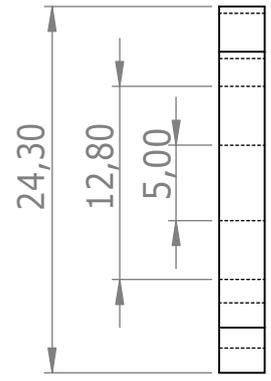
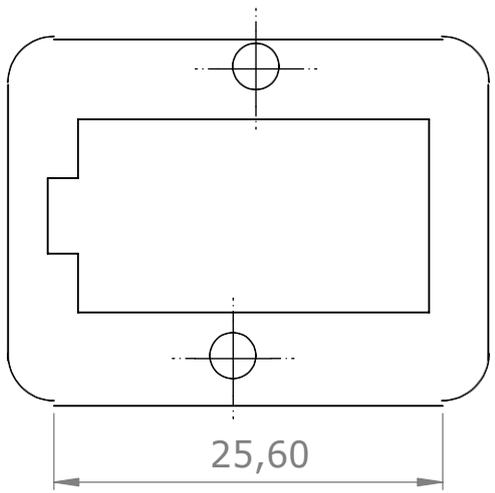
C

B

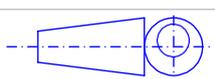
B

A

A



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 024

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4

3

2

1

4 3 2 1

F

F

E

E

D

D

C

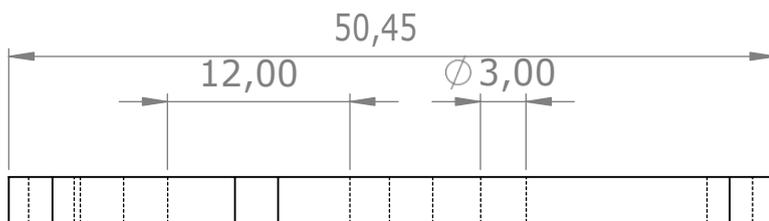
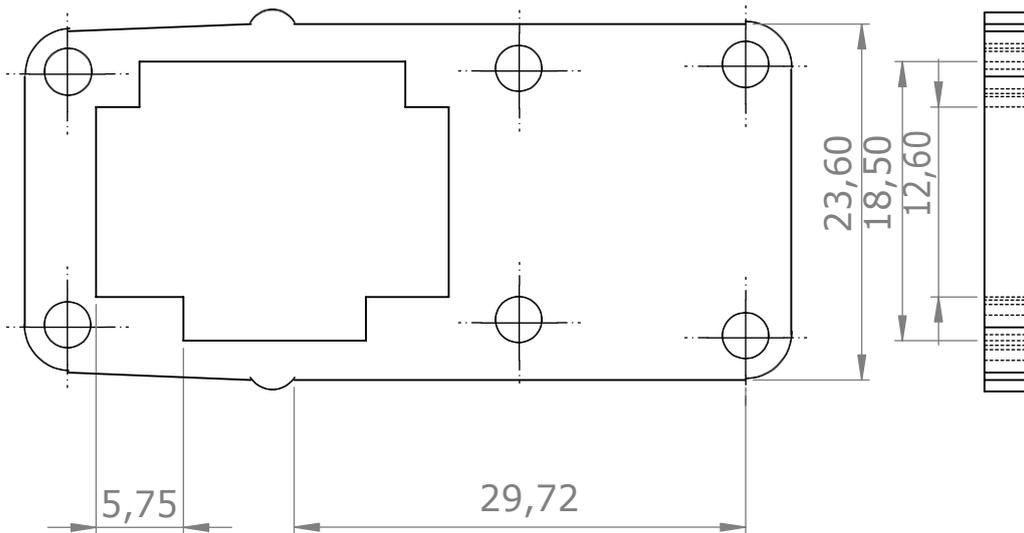
C

B

B

A

A



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 025

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4 3 2 1

4 3 2 1

F

F

E

E

D

D

C

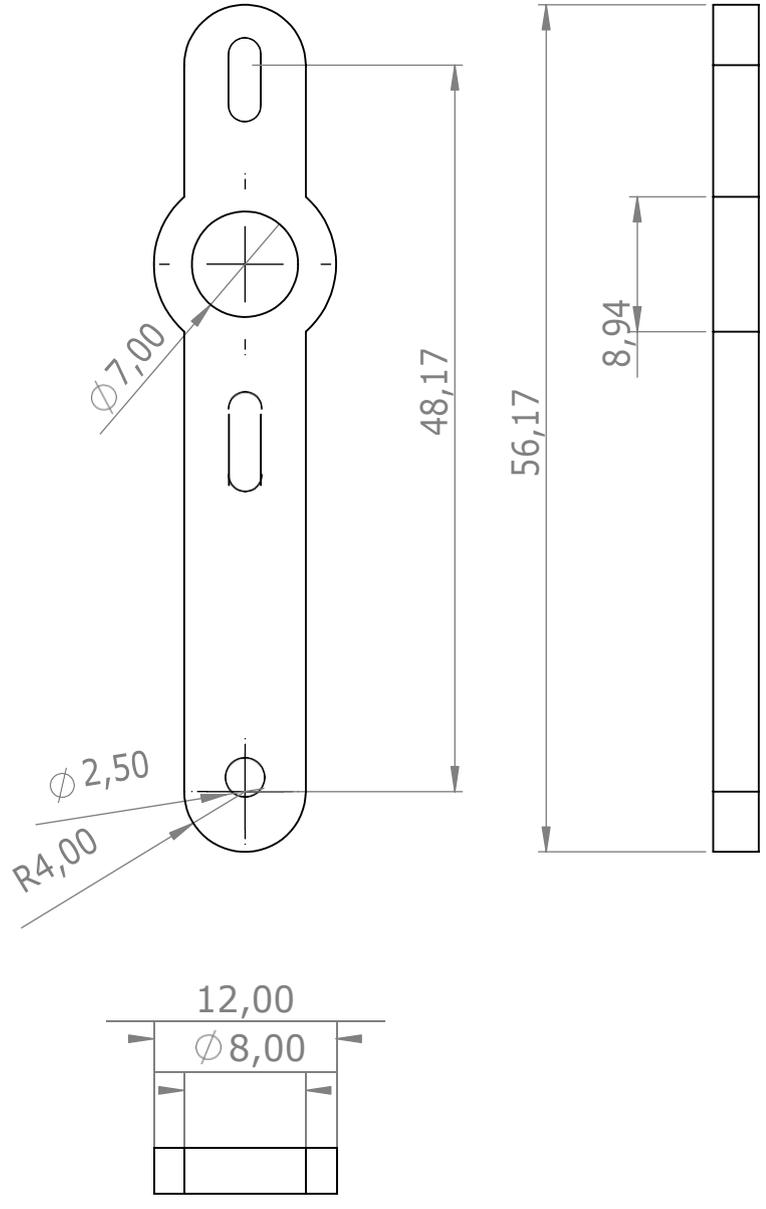
C

B

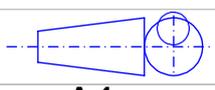
B

A

A



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

DJAAFRI
NOUREDDINE
KHENFAR RACHA

pièce 026

26/06/2021

4 3 2 1

4

3

2

1

F

F

E

E

D

D

C

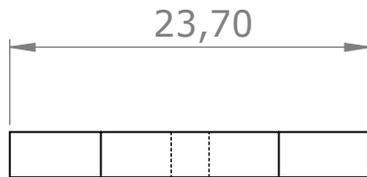
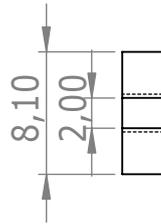
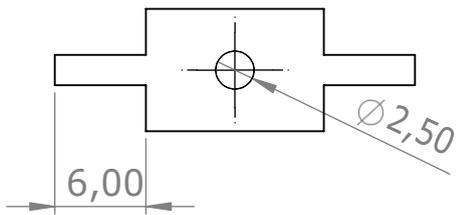
C

B

B

A

A



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 027

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4

3

2

1

4 3 2 1

F

F

E

E

D

D

C

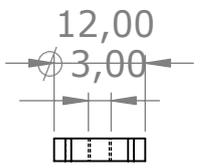
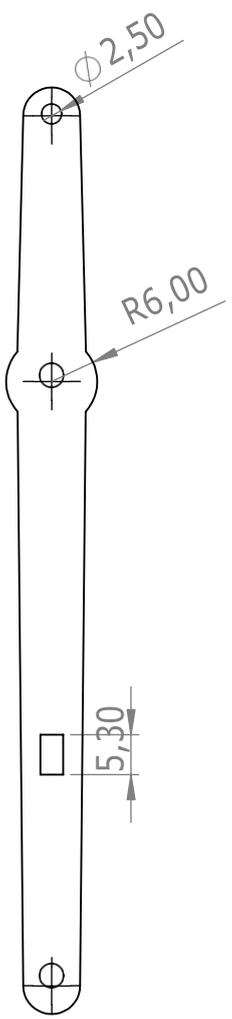
C

B

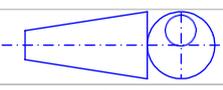
B

A

A



Echelle 1:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 028

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4 3 2 1

4

3

2

1

F

F

E

E

D

D

C

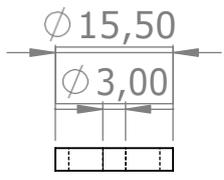
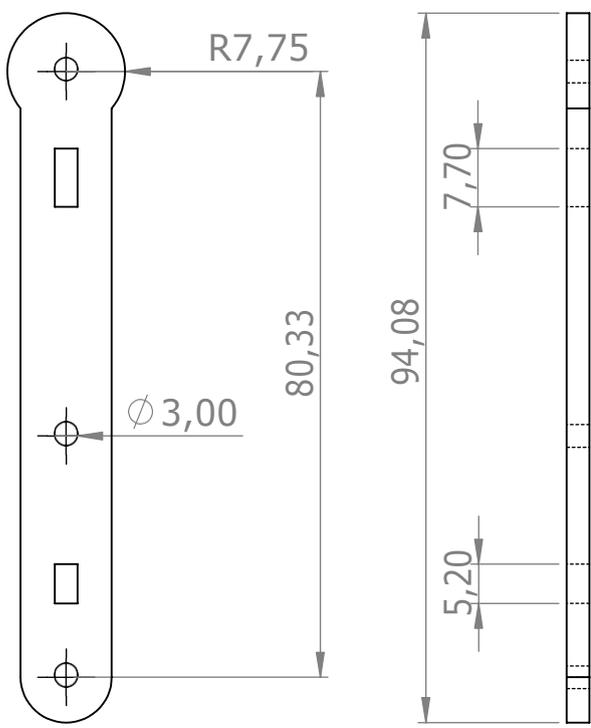
C

B

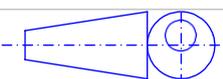
B

A

A



Echelle 1:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 029

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4

3

2

1

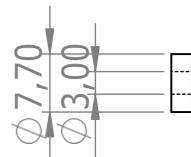
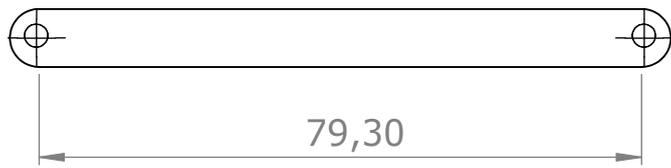
4 3 2 1

F

F

E

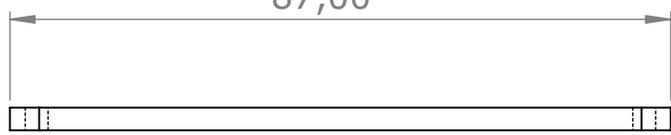
E



D

D

87,00



C

C

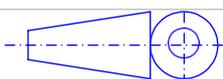
B

B

A

A

Echelle 1:1



A4

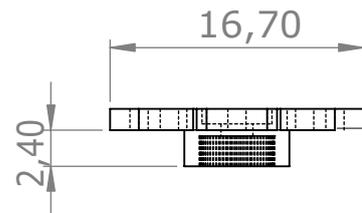
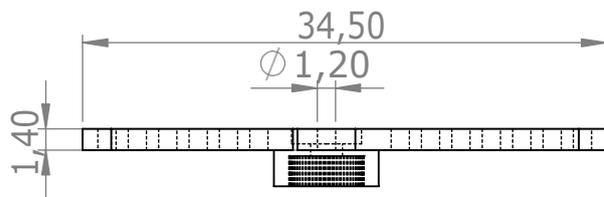
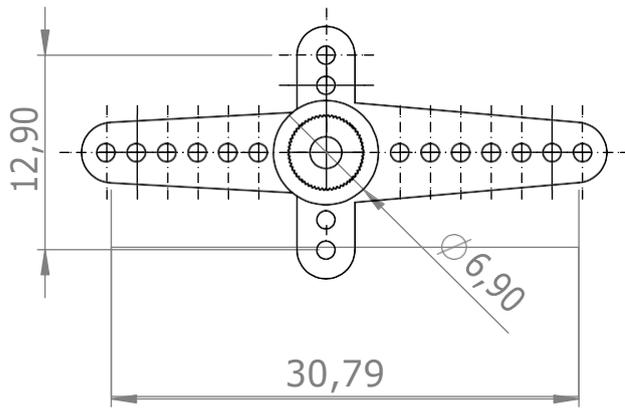
Département de Génie Mécanique
Université Ibn Khaldoun

pièce 030

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4 3 2 1



Echelle 2:1



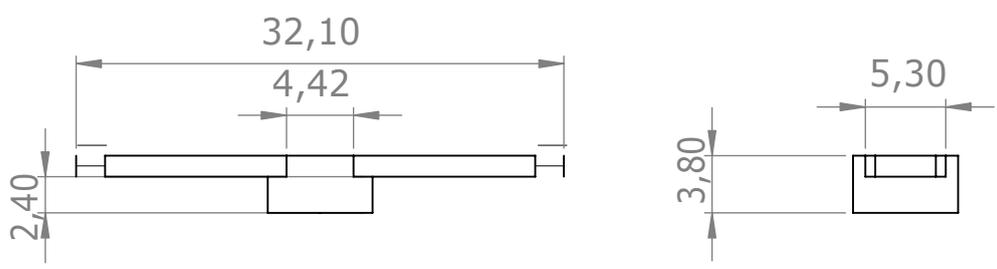
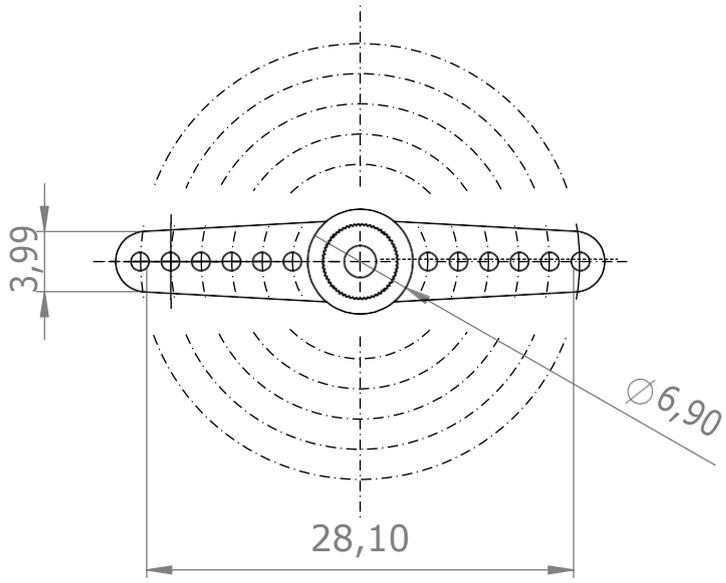
A4

Département de Génie Mécanique
Université Ibn Khaldoun

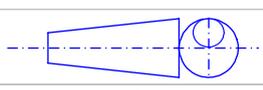
pièce 032

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 033

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4

3

2

1

4

3

2

1

F

F

E

E

D

D

C

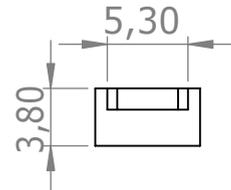
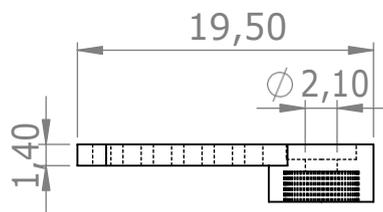
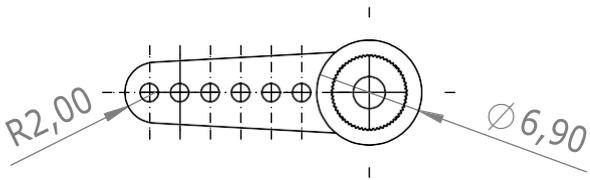
C

B

B

A

A



Echelle 2:1



A4

Département de Génie Mécanique
Université Ibn Khaldoun

pièce 034

DJAAFRI
NOUREDDINE
KHENFAR RACHA

26/06/2021

4

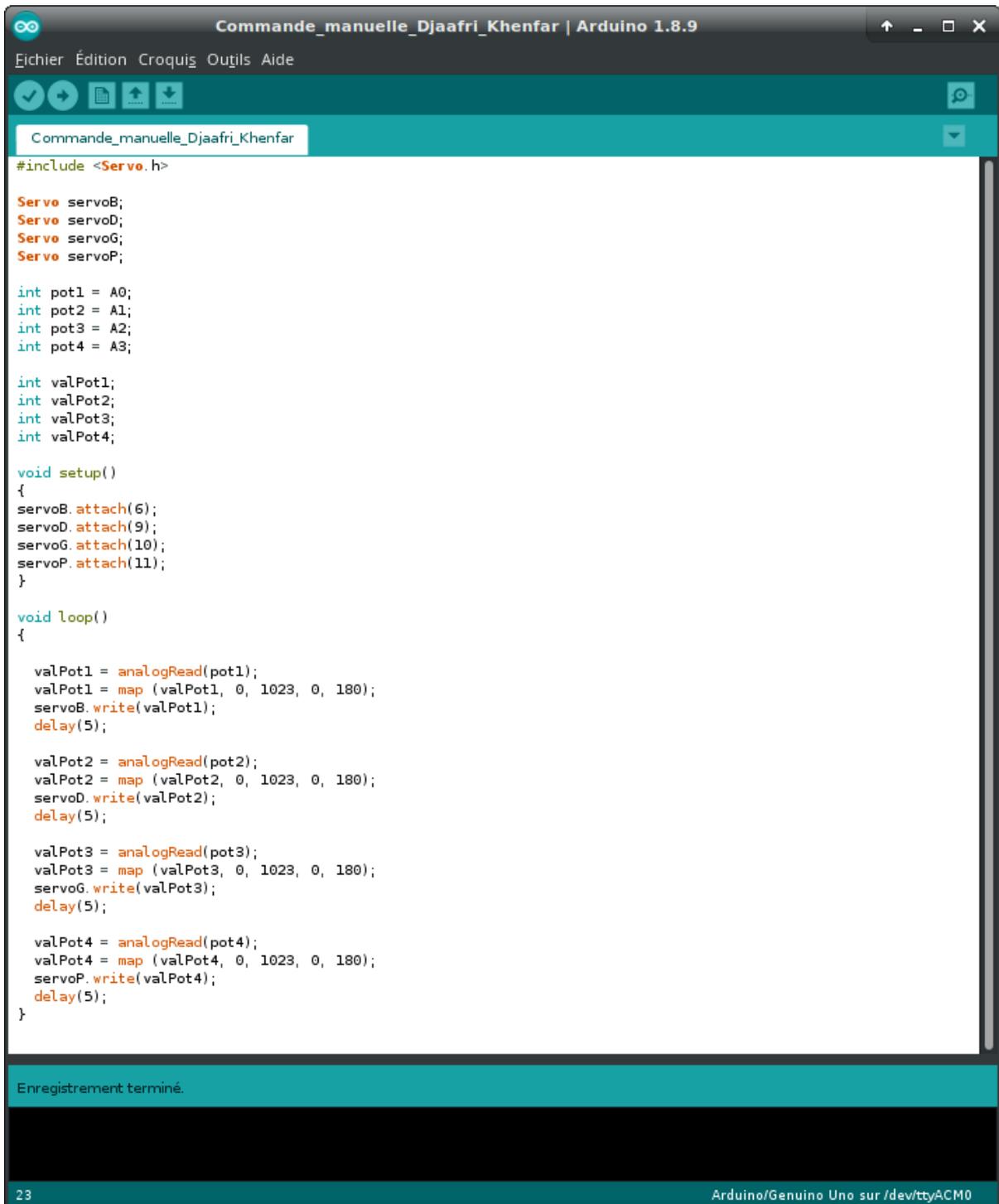
3

2

1

ANNEXE B :
Programmes
Arduino

I. Programme de la commande manuelle sur l'environnement de développement de Arduino



```
Commande_manuelle_Djaafri_Khenfar | Arduino 1.8.9
Fichier Édition Croquis Outils Aide

Commande_manuelle_Djaafri_Khenfar

#include <Servo.h>

Servo servoB;
Servo servoD;
Servo servoG;
Servo servoP;

int pot1 = A0;
int pot2 = A1;
int pot3 = A2;
int pot4 = A3;

int valPot1;
int valPot2;
int valPot3;
int valPot4;

void setup()
{
  servoB.attach(6);
  servoD.attach(9);
  servoG.attach(10);
  servoP.attach(11);
}

void loop()
{
  valPot1 = analogRead(pot1);
  valPot1 = map (valPot1, 0, 1023, 0, 180);
  servoB.write(valPot1);
  delay(5);

  valPot2 = analogRead(pot2);
  valPot2 = map (valPot2, 0, 1023, 0, 180);
  servoD.write(valPot2);
  delay(5);

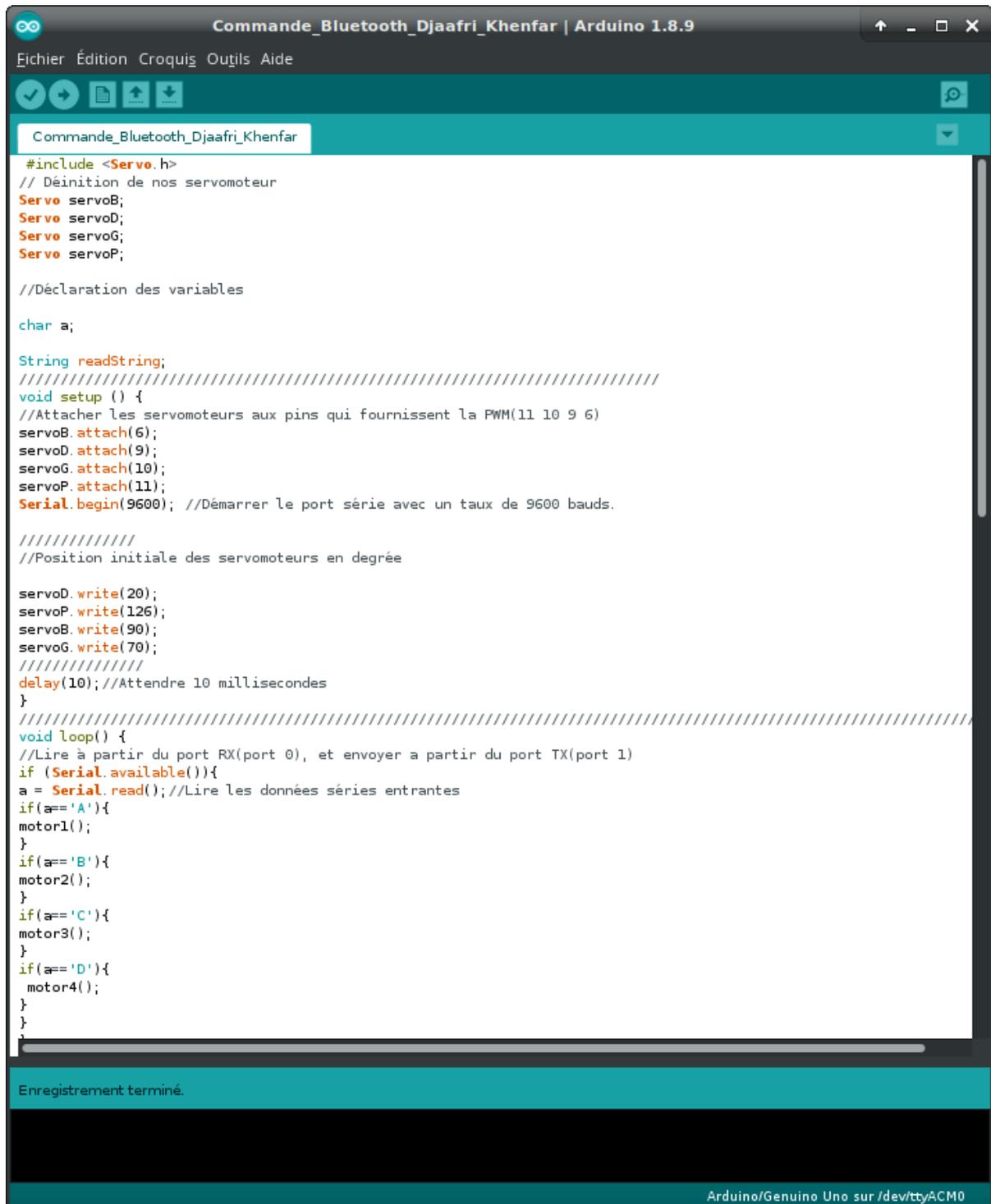
  valPot3 = analogRead(pot3);
  valPot3 = map (valPot3, 0, 1023, 0, 180);
  servoG.write(valPot3);
  delay(5);

  valPot4 = analogRead(pot4);
  valPot4 = map (valPot4, 0, 1023, 0, 180);
  servoP.write(valPot4);
  delay(5);
}

Enregistrement terminé.

23 Arduino/Genuino Uno sur /dev/ttyACM0
```

II. Programme de la commande à distance avec Bluetooth sur l'environnement de développement de Arduino



```
Commande_Bluetooth_Djaafri_Khenfar | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
Commande_Bluetooth_Djaafri_Khenfar
#include <Servo.h>
// Définition de nos servomoteur
Servo servoB;
Servo servoD;
Servo servoG;
Servo servoP;

//Déclaration des variables

char a;

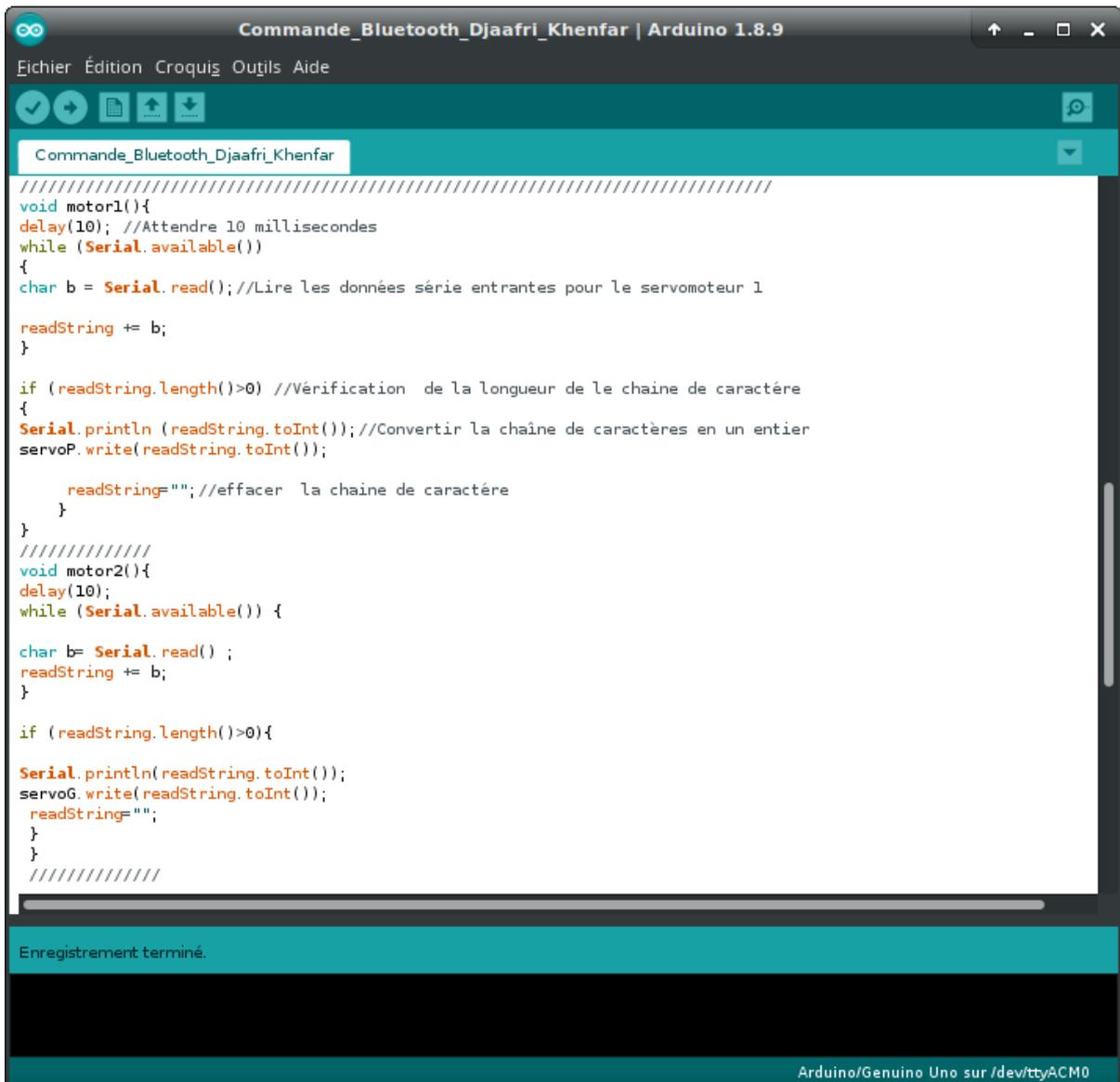
String readString;
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void setup () {
//Attacher les servomoteurs aux pins qui fournissent la PWM(11 10 9 6)
servoB.attach(6);
servoD.attach(9);
servoG.attach(10);
servoP.attach(11);
Serial.begin(9600); //Démarrer le port série avec un taux de 9600 bauds.

////////////////////////////////////////////////////////////////
//Position initiale des servomoteurs en degré

servoD.write(20);
servoP.write(126);
servoB.write(90);
servoG.write(70);
////////////////////////////////////////////////////////////////
delay(10); //Attendre 10 millisecondes
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void loop() {
//Lire à partir du port RX(port 0), et envoyer a partir du port TX(port 1)
if (Serial.available()){
a = Serial.read(); //Lire les données séries entrantes
if(a=='A'){
motor1();
}
if(a=='B'){
motor2();
}
if(a=='C'){
motor3();
}
if(a=='D'){
motor4();
}
}
}
}

Enregistrement terminé.

Arduino/Genuino Uno sur /dev/ttyACM0
```



```
Commande_Bluetooth_Djaafri_Khenfar | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
Commande_Bluetooth_Djaafri_Khenfar

void motor3(){
  delay(10);

  while (Serial.available () ) {

  char b = Serial.read();
  readString += b;
  }
  if (readString.length() >0) {
  Serial.println(readString.toInt());
  servoD.write(readString.toInt());

  readString="";
  }
  }
  //////////////////////////////////
void motor4(){
  delay(10);

  while (Serial.available()) {

  char b = Serial.read();
  readString += b;
  }
  if (readString.length()>0) {
  Serial.println(readString.toInt());
  servoB.write(readString.toInt());

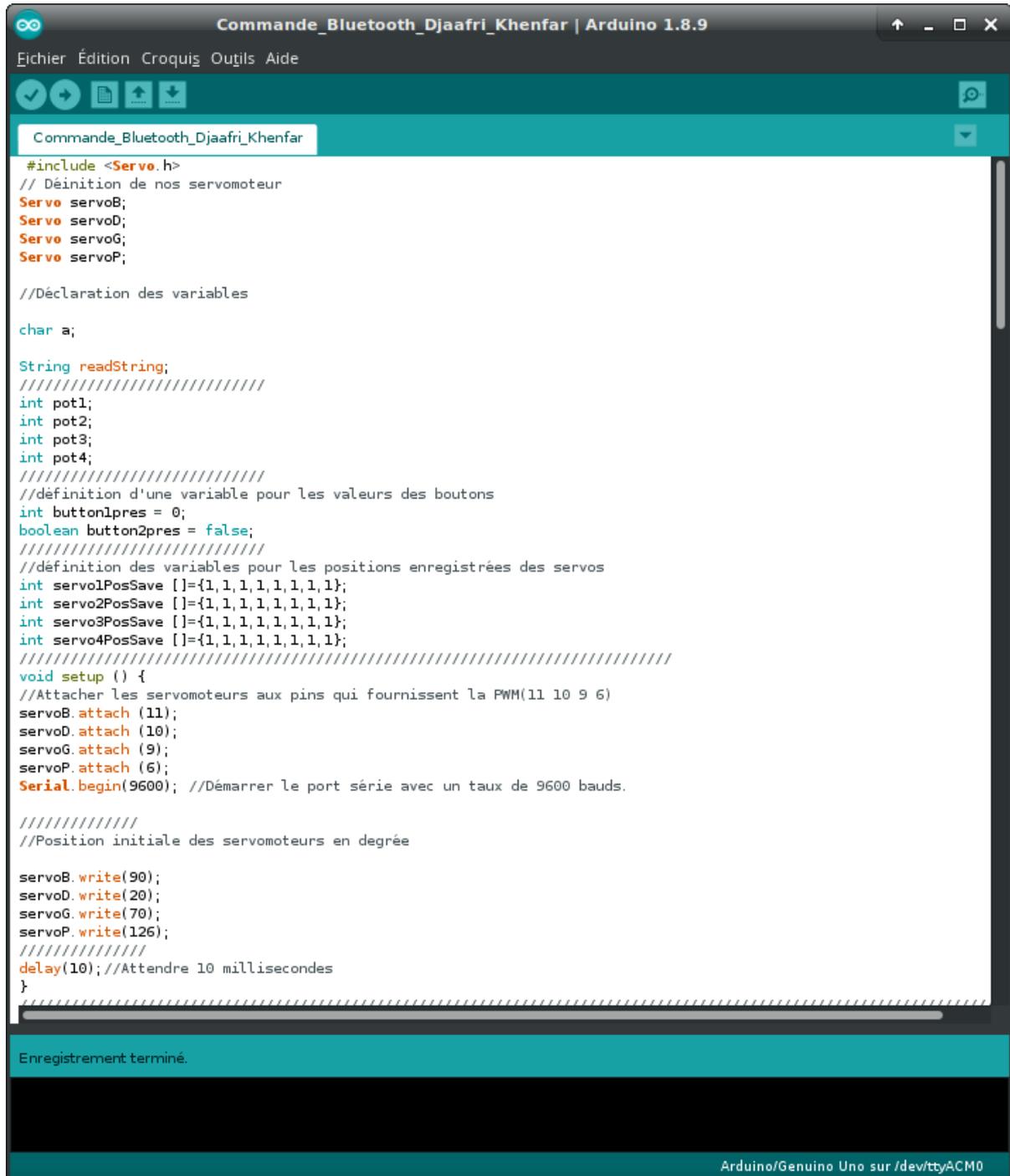
  readString="";
  }
  }
  //////////////////////////////////

```

Enregistrement terminé.

Arduino/Genuino Uno sur /dev/ttyACM0

III. Programme de la commande à cycle automatique sur l'environnement de développement de Arduino



```
Commande_Bluetooth_Djaafri_Khenfar | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
Commande_Bluetooth_Djaafri_Khenfar
#include <Servo.h>
// Définition de nos servomoteur
Servo servoB;
Servo servoD;
Servo servoG;
Servo servoP;

//Déclaration des variables

char a;

String readString;
////////////////////////////////////
int pot1;
int pot2;
int pot3;
int pot4;
////////////////////////////////////
//définition d'une variable pour les valeurs des boutons
int bouton1pres = 0;
boolean bouton2pres = false;
////////////////////////////////////
//définition des variables pour les positions enregistrées des servos
int servo1PosSave []={1,1,1,1,1,1,1,1};
int servo2PosSave []={1,1,1,1,1,1,1,1};
int servo3PosSave []={1,1,1,1,1,1,1,1};
int servo4PosSave []={1,1,1,1,1,1,1,1};
////////////////////////////////////
void setup () {
//Attacher les servomoteurs aux pins qui fournissent la PWM(11 10 9 6)
servoB.attach (11);
servoD.attach (10);
servoG.attach (9);
servoP.attach (6);
Serial.begin(9600); //Démarrer le port série avec un taux de 9600 bauds.

////////////////////////////////////
//Position initiale des servomoteurs en degré

servoB.write(90);
servoD.write(20);
servoG.write(70);
servoP.write(126);
////////////////////////////////////
delay(10); //Attendre 10 millisecondes
}
////////////////////////////////////
Enregistrement terminé.
Arduino/Genuino Uno sur /dev/ttyACM0
```

```
Commande_Bluetooth_Djaafri_Khenfar | Arduino 1.8.9
Fichier Édition Croquis Outils Aide

Commande_Bluetooth_Djaafri_Khenfar

////////////////////////////////////
void loop() {
//Lire à partir du port RX(port 0), et envoyer a partir du port TX(port 1)
if (Serial.available()){
  a = Serial.read();//Lire les données séries entrantes
  if(a == 'A'){
    motor1();
  }
  if(a == 'B'){
    motor2();
  }
  if(a == 'C'){
    motor3();
  }
  if(a == 'D'){
    motor4();
  }
  if(a == 'E'){
    button_Save();
  }
  if(a == 'F'){
    button_Play();
  }
}
}

////////////////////////////////////
void motor1(){
delay(10); //Attendre 10 millisecondes
while (Serial.available())
{
char b = Serial.read();//Lire les données série entrantes pour le servomoteur 1

readString += b;
}

if (readString.length()>0) //Vérification de la longueur de le chaine de caractère
{
Serial.println (readString.toInt());//Convertir la chaîne de caractères en un entier
servoP.write(readString.toInt());
pot1 = readString.toInt();

  readString="";//effacer la chaine de caractère
}
}

////////////////////////////////////
Enregistrement terminé.

Arduino/Genuino Uno sur /dev/ttyACM0
```

```
Commande_Bluetooth_Djaafri_Khenfar | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
Commande_Bluetooth_Djaafri_Khenfar $
////////////////////////////////
void motor2(){
  delay(10);
  while (Serial.available()) {

  char b= Serial.read() ;
  readString += b;
  }

  if (readString.length()>0){
Serial.println(readString.toInt());
    servoG.write(readString.toInt());
    pot2 = readString.toInt();
    readString="";
  }
  }
////////////////////////////////
void motor3(){
  delay(10);

  while (Serial.available () ) {

  char b = Serial.read();
  readString += b;
  }
  if (readString.length() >0) {
Serial.println(readString.toInt());
    servoD.write(readString.toInt());
    pot3 = readString.toInt();
    readString="";
  }
  }
////////////////////////////////
void motor4(){
  delay(10);

  while (Serial.available()) {

    char b = Serial.read();
    readString += b;
  }
  if (readString.length()>0) {
Serial.println(readString.toInt());
    servoB.write(readString.toInt());
    pot4 = readString.toInt();
    readString="";
  }
  }
}

Enregistrement terminé.

109 Arduino/Genuino Uno sur /dev/ttyACM0
```

```
Commande_Bluetooth_Djaafri_Khenfar | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
Commande_Bluetooth_Djaafri_Khenfar $
void button_Save(){
  delay(10);

  while (Serial.available()) {
    char b = Serial.read();
  }
  button1pres++;
  switch (button1pres){
    case 1:
      servo1PosSave[0] = pot1;
      servo2PosSave[0] = pot2;
      servo3PosSave[0] = pot3;
      servo4PosSave[0] = pot4;
      break;
    case 2:
      servo1PosSave[1] = pot1;
      servo2PosSave[1] = pot2;
      servo3PosSave[1] = pot3;
      servo4PosSave[1] = pot4;
      break;
    case 3:
      servo1PosSave[2] = pot1;
      servo2PosSave[2] = pot2;
      servo3PosSave[2] = pot3;
      servo4PosSave[2] = pot4;
      break;
    case 4:
      servo1PosSave[3] = pot1;
      servo2PosSave[3] = pot2;
      servo3PosSave[3] = pot3;
      servo4PosSave[3] = pot4;
      break;
    case 5:
      servo1PosSave[4] = pot1;
      servo2PosSave[4] = pot2;
      servo3PosSave[4] = pot3;
      servo4PosSave[4] = pot4;
      break;
    case 6:
      servo1PosSave[5] = pot1;
      servo2PosSave[5] = pot2;
      servo3PosSave[5] = pot3;
      servo4PosSave[5] = pot4;
      break;
  }
}

Enregistrement terminé.

109 Arduino/Genuino Uno sur /dev/ttyACM0
```

```
Commande_Bluetooth_Djaafri_Khenfar | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
Commande_Bluetooth_Djaafri_Khenfar $
    case 7:
        servo1PosSave[6] = pot1;
        servo2PosSave[6] = pot2;
        servo3PosSave[6] = pot3;
        servo4PosSave[6] = pot4;
        break;
    case 8:
        servo1PosSave[7] = pot1;
        servo2PosSave[7] = pot2;
        servo3PosSave[7] = pot3;
        servo4PosSave[7] = pot4;
        break;
    }
    delay(1500);
}

void button_Play(){
    delay(10);

    while (Serial.available()) {

        char b = Serial.read();
        button2pres = true;
    }
    if(button2pres){
        for(int i=0; i<8; i++){
            servoB.write(servo1PosSave[i]);
            servoD.write(servo2PosSave[i]);
            servoG.write(servo3PosSave[i]);
            servoP.write(servo4PosSave[i]);
            delay(2000);
        }
    }
    button2pres = false;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
Enregistrement terminé.
109 Arduino/Genuino Uno sur /dev/ttyACM0
```