الجمهورية الجزائرية الديمقراطية الشعبية

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ YAHIA FARÈS DE MÉDÉA
FACULTÉ DE LA TECHNOLOGIE
Département du Génie Mécanique

Projet Fin D'études

Présenté pour l'obtention du Diplôme

De Master en Génie Mécanique

Option : Mécatronique

THÈME :

Conception et réalisation d'un bras de robot piloté

Réalisé par :

RAHAL Fouad
ABDELLATIF Abdelmoniaame

Proposé et dirigé par :

Pr. ABUDURA Salam

Année scolaire : 2020/2021

REMERCIEMENTS

D’abord nous remercions Dieu notre créateur de nous avoir donné la santé, la force, la volonté et le courage de pouvoir réussir et d’accomplir ce modeste mémoire.

Nous tenons à exprimer nos sincères remerciements et notre reconnaissance à Pr. ABUDURA.S 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.

On ne saura clôturer ces remerciements sans exprimer toute notre affection pour les enseignants de l'Université de MÉDÉA, qui ont bien voulu partager leur savoir durant nos études.

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

DEDICACES

Je tiens à dédier ce modeste travail avant tout :

A mon très cher père.

A ma très chère mère.

A mes très chers frères.

A toute la famille.

A tous mes amis et mes collègues.

Et à tous ceux qui m'ont aidé de près ou de loin à accomplir ce travail.

ملخص

قمنا في مذكرة التخرج هذه بتصميم وإنجاز نموذج لذراع مناور ذو أربعة درجات للحرية، وتمت هذه العملية عبر مرحلتين هما الإدراك العملي والبرمجة.

في مرحلة الإدراك العملي قمنا بالتصميم عن طريق برنامج صوليدووركس وقمنا بتجميع كل القطع الخاصة بالتشغيل يدويا، ويتميز الذراع ببنية ميكانيكيه مفصلية تتحرك بالمحركات المؤازرة وتتحكم به بطاقة أرد وينو.

تمت البرمجة بالاستعمال بطاقة اردوينو استنادا إلى لغة البرمجة++c ، ويتم التحكم في الذراع باستخدام تطبيق اندرو يد الذي انجزناه.

Résumé

Dans cette note de fin d'études, nous avons conçu et réalisé un modèle de bras manipulateur à quatre degrés de liberté, et ce processus s'est déroulé en deux étapes : réalisation pratique et programmation.

Au stade de la réalisation pratique, nous avons conçu à travers le programme SolidWorks et assemblé toutes les pièces pour un fonctionnement manuel. Le bras est caractérisé par une structure mécanique articulée qui se déplace par des servomoteurs et est contrôlée par une carte Arduino.

Il a été programmé à l'aide de la carte Arduino basée sur le langage de programmation C++, et le bras est contrôlé à l'aide de l'application Android Hand que nous avons réalisée.

ABSTRACT

In this graduation note, we designed and produced a model of a manipulator arm with four degrees of freedom, and this process took place in two stages: practical realization and programming.

At the stage of practical realization, we designed through the SolidWorks program and assembled all the parts for manual operation. The arm is characterized by an articulated mechanical structure that moves by servomotors and is controlled by an Arduino board.

It was programmed using the Arduino board based on the C ++ programming language, and the arm is controlled using the Android Hand app that we made.

Liste des figures

Chapitre I : Généralités sur la robotique

Figure. I.1.Nombre d'installations annuelles de robots industriels dans le monde et perspectives d'ici à 2022^{E} (en milliers).....5

Figure I.2. Robots mobiles.....6

Figure I.3. Robot manipulateur.....6

Figure I.4.Différents Les éléments constitutifs d'un système robotique.....7

Figure I.5.Structure sérielle.....8

Figure I.6.Structure fermée.....8

Figure I.7**.** Manipulateur à commande manuelle.....9

Figure I.8.Manipulateur automatique.....9

Figure I.9.Robot programmable.....10

Figure I.10.Robots intelligents.....10

Figure I.11. Structure d'un Robot cartésien.....11

Figure I.1**2**. Structure d'un robot cylindrique.....11

Figure I.13. Structure d'un robot sphérique.....11

Figure I.14. Structure d'un robot 3R.....12

Figure I.15. Structure dite SCARA.....12

Chapitre II : Conception et réalisation du bras manipulateur.

Figure II.1.Logo SolidWorks.....16

Figure II.2.fenêtre principale de SolidWorks.....16

Figure II.3.plateau inférieur du la bas sous SolidWorks.....18

Figure II.4.les piliers et plateau inférieur du la bas sous SolidWorks.....18

Figure II.5.les piliers et plateaux du la bas sous SolidWorks.....18

Figure II.6.Disque avec support servomoteur sur SolidWorks.....19

Figure II.7.Morceau de bras vertical sur SolidWorks.....19

Figure II.8.Bras vertical du L'épaule sur SolidWorks.....20

Figure II.9.Morceau de bras vertical sur SolidWorks.....20

Figure II.10.Bras vertical du L'épaule sur SolidWorks.....21

Figure II.11.poignet avec pince sur SolidWorks.....21

Figure II.12.poignet avec pince sur SolidWorks.....22

Figure II.13.Base avec épaule sur SolidWorks.....22

Figure II.14.Le bras manipulateur que nous avons sur SolidWorks.....23

Figure II.15.La forme finale de notre du bras manipulateur sur SolidWorks.....23

Figure II.16.Réalisation final et les différents organes du bras manipulateur.....24

Chapitre III : Etude technique des composants du bras manipulateur.

Figure III.1.Schémas des éléments non visibles d'un servomoteur.....26

Figure III.2.Certains types du fils de servomoteur.....27

FigureIII.3.Principe d'un servomoteur.....28

FigureIII.4.Signaux et positions d'un servomoteur.....28

FigureIII.5.Servomoteurs MG995 utilisés dans le bras manipulateur.....28

FigureIII.6.Servomoteurs MG90S utilisés dans le bras manipulateur.....29

FigureIII.7.Emplacement du servomoteur sur le bras robots.....29

FigureIII.8.module Bluetooth (HC-06).....31

FigureIII.9.Conclusion Schéma synoptique simplifié du système.....32

FigureIII.10.Schéma de la communication avec le module Bluetooth HC-06.....32

FigureIII.11.Types des cartes arduino.....33

FigureIII.12.La carte ArduinoUNO.....34

FigureIII.13.Montage Arduino Uno + servomoteur.....35

FigureIII.14.câblage du Module Bluetooth HC06 sur Arduino Uno.....36

FigureIII.15.Schéma électrique globale du Bras manipulateur.....36

FigureIII.16.carte de commande d'un bras manipulateur.....37

FigureIII.17.circuit de la carte commande d'un bras manipulateur.....37

Chapitre IV: Commande du bras manipulateur.

FigureIV.1: Environnement de développement Arduino.....39

Figure IV.2.Fenêtre de moniteur série.....40

Figure IV.3.Plan de programmation planifié.....41

Figure IV.4.Partie d'déclaration des variables.....42

Figure IV.5.Fonction d'attache et position.....42

Figure IV.6.Fonction d'affichage à moniteur série.....43

Figure IV.7.Partie configuration.....43

Figure IV.8.Fonction rotation de base.....44

Figure IV.9.Fonction rotation d'épaule.....44

Figure IV.10.Fonction rotation de coude.....45

Figure IV.11.Fonction rotation de pincette.....45

Figure IV.12.Fonction SAVE.....46

Figure IV.13.Fonction RUN.....46

Figure IV.13.Fonction RESET.....47

Figure IV.14.Fonction. Home0.....47

Figure IV.15.Fonction. Home.....48

Figure IV.16.Fonction. Home2.....48

Figure IV.17.Fonction **testRunCounter**.....50

Figure IV.18.Fonction **testrecCounter**.....50

Figure IV.19**.**Logo App Inventor.....51

Figure IV.20.Partie design de notre application sous AppInventor.....51

Figure IV.21**.** Partie blocks de AppInventor.....52

Figure IV.22**.** Définition des éléments de l'objet Bluetooth.....52

Figure IV.23. Block Bluetooth.....53

Figure IV.24..Blocs de contrôle des curseurs.....53

Figure IV.25.Block bouton SAVE.....54

Figure IV.26.Block bouton RUN.....54

Figure IV.27.Block bouton RESET.....55

Figure IV.28.Block bouton Home.....55

Figure IV.29.Interface finale de l'application.....56

Liste des tableaux

Chapitre I : Généralités sur la robotique.

Chapitre II : Conception et réalisation du bras manipulateur.

Chapitre III : Etude technique des composants du bras manipulateur.

Tab.III.1.Tableau des caractéristiques des servomoteurs du bras……30
Tab.III.2.Tableau des limites de rotation du bras……..30
Tab.III.3. Tableau des caractéristiques du récepteur Bluetooth du bras……31
Tab.III.4. Caractéristiques de l'Arduino UNO ……..34

Chapitre IV: Commande du bras manipulateur.

Sommaire

REMERCIEMENTS I
DEDICACES II
Résumé III
ABSTRACT III
Liste des figures IV
Liste des tableaux VIII
Introduction Générale 1

Chapitre I : Généralités sur la robotique.

1.1. Introduction 3
I.2. Définitions : 3
I.2.1. Définition d'un robot : 3
I.2.2. Définition de la robotique : 3
I.3. L'histoire de la robotique : 3
I.4. Types de robots 5
I.4.1. Robots mobiles 6
I.4.2. Robot manipulateur 6
I.5. Différents Les éléments constitutifs d'un système robotique : 7
I.5.1. Unité informationnelle : 7
I.5.2. Unité opérationnelle : 7
I.6. Structure mécanique articulée : 7
I.6.1. Structure mécanique articulée à chaîne cinématique simple : 8
I.6.2. Structure mécanique articulée à chaîne cinématiques fermée 8
I.7. Classification des robots 9
I.7.1. Classification fonctionnelle 9
I.7.1.1. Manipulateur à commande manuelle : 9
I.7.1.2. Manipulateur automatique : 9
I.7.1.3. Robots programmables : 10
I.7.1.4. Robots intelligents 10
I.7.2. Classification géométrique 11
I.8. Caractéristiques d'un robot 12
I.9. Domaines d'applications 13
I.10. Avantages et inconvénients des robots 13

I.10.1. Avantages 13
I.10.2. Inconvénients 14
I.11.Conclusion 14

Chapitre II : Conception et réalisation du bras manipulateur.

II.1. Introduction 15
II.2. Choix de l'outil informatique de CAO.....15
II.3.logiciel SolidWorks 15
II.3.1. Présentation du logiciel SolidWorks 15
II.3.2.Fonctionnement 16
II.3.2.1. Conception des pièces 16
II.3.2.2. L'assemblage 17
II.3.2.3. La mise en plan 17
II.4.Conception des pièces et des différentes articulations du bras.....17
II.4.1. la base 17
II.4.2L'épaule 19
II.4.3.Le coude.....20
II.4.4.poignet avec la pince 21
II.5.Assemblage des pièces du bras manipulateur.....22
II.6.Réalisationréelle de la Structure mécanique du bras manipulateur 24
II.7.Conclusion.....25

Chapitre III : Etude technique des composants du bras manipulateur.

III.1.Introduction 26
III.2.Servomoteurs.....26
III.2.1.Présentation du servomoteur.....26
III.2.2.Les paramètres techniques des servomoteurs utilisés 29
III.2.3.Espace opérationnel.....30
III.3.Le module Bluetooth 31
III.3.1.Présentation du récepteur Bluetooth (HC-06) :.....31
III.3.2.Les paramètres techniques du récepteur Bluetooth (HC-06) :.....31
III.3.3.Communication de l'application avec l'Arduino.....32
III.4. Carte Arduino 32
III.4.1 La carte Arduino UNO : 33
III.4.2.Les paramètres techniques de la carte Arduino UNO : 34

III.5.Schéma de câblage des composants du bras manipulateur.....35
III.6. .Réalisation du lacarte de commande pour le bras manipulateur.....36
III.7. Conclusion.....38

Chapitre IV: Commande du bras manipulateur.

IV.1.Introduction.....39
IV.2.logiciel Arduino.....39
IV.2.1.Présentation.....39
IV.2.2.Explication du Structure de la programmation.....40
IV.3. Le concept d'App Inventor :.....51
IV.3.1.Présentation.....51
IV.3.2.1. Réalisation de l'application Android.....51
IV.3.2. Partie design.....51
IV.3.2.1partie bloc.....52
IV.4. Conclusion.....57
Conclusion générale.....58
Bibliographie..... XII

Introduction Générale

Le monde d'aujourd'hui connaît un énorme développement. La technologie robotique marque le développement du monde technologique, car de nombreuses technologies basées sur la technologie moderne incluent des circuits électromécaniques contrôlés par des unités pilotes programmables pour améliorer les performances dans de multiples domaines: industriel, militaire, médical……et bien d'autres. Avec l'avancée de la robotique. Les robots actuels ont été dotés d'une "intelligence" et d'une certaine autonomie, ce qui leur permettra de se développer dans de nouveaux domaines. Le bras robotique deviendra davantage un outil pour effectuer des tâches répétitives dans un espace sûr, mais un partenaire des activités quotidiennes et Les contraintes du bras robotique sont liées à la sécurité et au confort. La plupart des chercheurs ont mis l'accent sur le contrôle des robots et La conception des robots est généralement Parce que ces deux procédés sont sujets à renouvellement et à développement permanent.

Le bras robotisé est un système de positionnement dans lequel la force agissant sur l'articulation est générée par l'actionneur. Contrôler un robot manipulateur consiste à l'amener à un état souhaité ou à effectuer des tâches très spécifiques avec une grande précision dans un environnement réel. La phase de programmation est une phase très importante car elle permet de générer les trajectoires nécessaires à la réalisation de ces tâches.

Les principaux objectifs de nos travaux incluent la recherche théorique et la réalisation effective d'un bras mécanique à quatre degrés de liberté, utilisant des servomoteurs pour permettre à ces derniers d'effectuer des mouvements de haute précision. Après la mise en place, nous continuons à programmer, notamment la configuration Bluetooth et le contrôle à distance avec le smartphone via l'application Android.

Afin de mieux illustrer notre travail, le document est divisé en quatre chapitres différents, organisés comme suit :

Le premier chapitre traite de la généralité. Nous introduisons ici la définition de la robotique, son histoire et les différents types de robots, ainsi que la classification des robots et leurs domaines d'utilisation.

Dans le deuxième chapitre, nous avons présenté la conception du bras robotique et la réalisation de nos réalisations réelles dans le bras.

Dans le troisième chapitre, nous effectuerons des recherches techniques et des recherches pour sélectionner les composants qui nous satisfont du bon fonctionnement du bras, notamment : la carte **ArduinoUno**, le servomoteur et le module Bluetooth HC_06. Nous verrons leurs caractéristiques et définitions respectives et nous montrerons comment les assembler en montrant le circuit électrique que nous avons réalisé.

Le quatrième chapitre explique notre programme de commande et de contrôle du bras robotique. Focus sur la présentation de la carte de contrôle et son fonctionnement, et les types de contrôles pouvant être utilisés dans notre implémentation. Le traitement de la simulation du bras contrôlé par le programme **arduino** et l'application que nous avons créée sur logiciel Inventeur du MIT.

Enfin, le mémoire se termine avec une conclusion résumant les connaissances acquises tout au long de ce projet, ainsi que les perspectives pouvant venir améliorer notre bras manipulateur et compléter notre travail dans le futur. Et permettra nous d'acquérir une compétence certaine, qui contribuera à un gain énorme sur plusieurs niveaux Dont dans notre domaine dans le domaine de la mécatronique.

Chapitre I : Généralités sur la robotique

1.1. Introduction

De nos jours, l'utilisation des systèmes robotiques dans l'industrie ou la recherche est de plus en plus importante, notamment l'utilisation de la robotique. Celle-ci permet de manipuler, avec beaucoup de précision et donc de manière plus sûre, les robots sont utilisé d'une manière distinctive qui facilite les tâches humaines et améliore les performances.

Le secteur d'activité des systèmes robotiques s'est élargi de façon importante en particulier dans le domaine médical, spatial, industriel, militaire et dans le domaine de l'agriculture.

Dans ce premier chapitre, est dédié principalement à des généralités à la robotique on va donner un aperçu sur les robots, un bref historique sur l'évolution de la robotique industrielle et présenter les différents types dérobons et les éléments constitutifs de ces derniers.

I.2. Définitions :

I.2.1. Définition d'un robot :

Un robot est un agent physique dans un système mécanique multi-articulaire entraîné par un actionneur. C'est un manipulateur à plusieurs degrés de liberté, qui peut être contrôlé automatiquement et peut manipuler des objets ou effectuer des opérations selon un programme fixe, modifiable ou ajustable. Effectuer diverses tâches dans l'environnement dans lequel il évolue. [1]

I.2.2. Définition de la robotique :

La robotique est l'ensemble des technologies dans les domaines scientifiques et industriels liés à la production de robots et de machines. C'est un rêve humain de les utiliser dans des tâches difficiles et de les développer et de les ajuster en fonction de l'objectif.[2]

I.3. L'histoire de la robotique :

L'origine du mot robot provient de la langue tchèque dans laquelle sont ancêtre "rabota" signifie travail forcé.

Le terme de robotique quant à lui est apparu en 1942 dans le cycle universellement connu rédigé par *Isaac Asimov* et intitulé "Les robots".

Voici une définition concise de ce que peut être un robot : " Machine programmable qui imite des actions d'une créature intelligente."

Au cours de l'histoire on peut distinguer 3 types de robots correspondant en quelques sortes à l'évolution de cette "espèce" créée par l'Homme.

Le premier type de machine que l'on peut appeler robot correspond aux "Automates". Ceux-ci sont généralement programmés à l'avance et permettent d'effectuer des actions répétitives.[3]

Le second type de robot correspond à ceux qui sont équipés de capteurs (en fait les sens du robot). On trouve des capteurs de température, photoélectron que, à ultrasons pour par exemple éviter les obstacles et/ou suivre une trajectoire. Ces capteurs vont permettent au robot une relative adaptation à son environnement afin de prendre en compte des paramètres aléatoires qui n'aurait pu être envisagés lors de leur programmation initiale. Ces robots sont donc bien plus autonomes que les automates mais nécessitent un investissement en temps de conception et en argent plus conséquent..[3]

Enfin le dernier type de robot existant correspond à ceux disposant d'une intelligence dite "artificielle" et reposant sur des modèles mathématiques complexes tels que les réseaux de neurones. En plus de capteurs physiques comme leurs prédécesseurs, ces robots peuvent prendre des décisions beaucoup plus complexes et s'appuient également sur un apprentissage de leurs erreurs comme peut le faire l'être humain. Bien sûr il faudra attendre encore longtemps avant que le plus "intelligent" des robots ne soit égal, tant par sa faculté d'adaptation que par sa prise de décisions, à l'Homme.[3]

C’est l'une des dates les plus importantes connues dans les blogs :

- En 1947 : L’apparition de premier manipulateur électrique télé-opéré.
- En 1945 : Invention du premier robot qui peut être programmé.
- En 1961 : Montage d'un robot sur une chaîne de fabrication de General Motors.
- En1961 : Premier robot avec contrôle en effort.
- En 1963 : Le premier contrôle du robot avec vidéo
- En 1973 : La Société Hitachi a développé le premier robot avec un système de traitement d'image pour des boulons de fixation à une forme mobile.

- En 1974 : La société ASEA a livré les premiers robots tout-électriques, industriels contrôlés par microprocesseur.
- En 1978 : la société Urination a développé le robot PUMA (Programmable Universel Machine for Assemblé).
- En 1981 : TakeoKanade construit le premier robot muni de moteurs installés directement aux articulations du bras. Cela a permis des robots plus rapides et plus précis qu'auparavant
- En 1998 : ABB, Suède, a développé le FlexPicker, le robot de préparation de commandes le plus rapide au monde basé sur le robot delta développé par Reymond Clavel, Institut fédéral de technologie de Lausanne (EPFL).
- En 2003 : Première mission d'exploration de la planète Mars impliquant deux robots mobiles, Spirit et Opportunité.[3]

Ain, la Fédération Internationale de la Robotique (IFR) anticipe près de 2 millions d'installations de robots industriels entre 2019 et 2022E.[4]

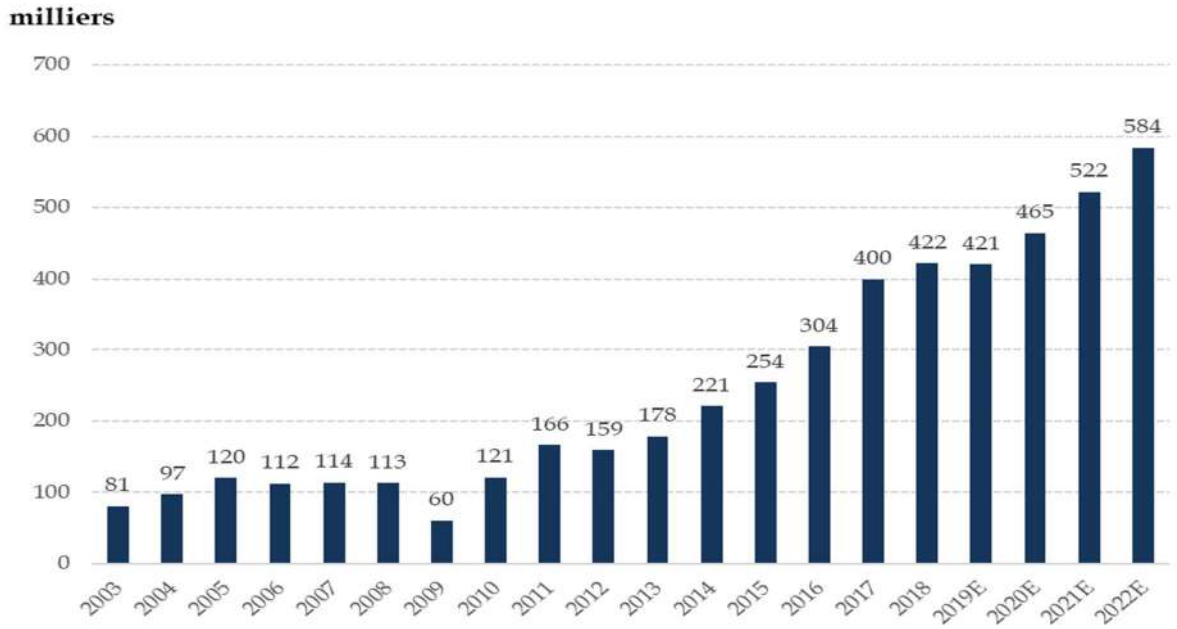


Figure. I.1.Nombre d'installations annuelles de robots industriels dans le monde et perspectives d'ici à 2022^{E} (en milliers).[4]

I.4. Types de robots

Dans le domaine des robots nous trouvons deux types de robots populaires **:** robots mobiles et robots manipulateurs. [5].

I.4.1. Robots mobiles

Ce sont des robots mobiles conçu et programmés pour se déplacer en fonction des caractéristiques de chaque environnement. [5].



FigureI.2. Robots mobiles.

I.4.2. Robot manipulateur

Selon la RIA (Robot Institute of American) , il s'agit d'un manipulateur reprogrammable multifonctionnel conçu pour déplacer des matériaux, des pièces, des outils ou tout autre équipement spécialisé à travers une série de mouvements programmés et effectuer diverses autres tâches. L'ISO (Organisation internationale de normalisation) la définit comme une machine entraînée par un mécanisme qui comprend plusieurs degrés de liberté, généralement avec l'apparence d'un ou plusieurs bras, se terminant par le poignet, capable de contenir des outils, des pièces ou des équipements d'inspection. [5].



Figure. I.3. Robot manipulateur.

I.5. Différents Les éléments constitutifs d'un système robotique :

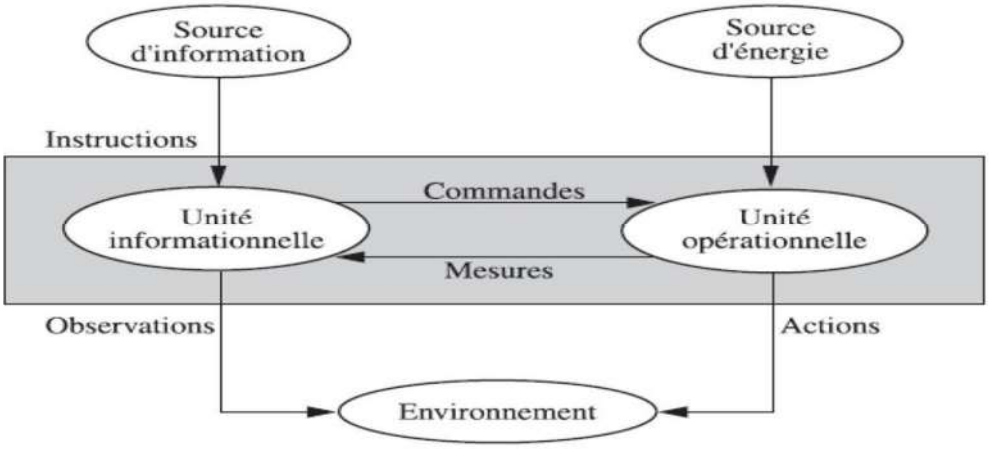


Figure. I.4.Différents Les éléments constitutifs d'un système robotique.[6]

I.5.1. Unité informationnelle :

Recevez des instructions décrivant les tâches à effectuer, les mesures liées à l'état interne de la structure mécanique qui constitue le bras robotique et les observations de son environnement. Par conséquent, elle a développé le contrôle de diverses articulations pour effectuer ces tâches. Le système actuel fonctionne de manière interactive permanente selon le cycle information-décision-action.[6].

I.5.2. Unité opérationnelle :

Effectuez une action commandée en empruntant l'énergie nécessaire à la source d'énergie. Cette partie constitue un robot physique, qui intègre des structures mécaniques (segments, articulations, structures, etc.), des modules d'énergie (amplificateurs, variateurs, servovalves, etc.), des convertisseurs d'énergie (moteurs, vérins, etc.), une cinématique de transmission mécanique des chaînes (réducteurs, vis à billes, courroies crantées, etc.), des capteurs proprioceptifs placés sur chaque axe pour mesurer en continu leur position et leur vitesse, et enfin les effecteurs ou organes terminaux qui interagissent avec l'environnement. [6].

I.6. Structure mécanique articulée :

Un robot manipulateur est constitué généralement de deux sous-ensembles distincts :

Un organe terminal, qui est un dispositif utilisé pour manipuler la structure mécanique des objets et des articulations (SMA).Il se compose d'un groupe de solides connectés,

généralement l'un après l'autre, où chaque solide est mobile par rapport au précédent. Cette mobilité s'exprime en degrés de liberté (ddl), qui est par définition le nombre de mouvements indépendants possibles de l'entité C1 par rapport à l'entité C2 qui lui est directement connectée. [6]

Une structure mécanique articulée peut être exprimée comme une architecture composée de multiples chaînes de corps rigides assemblées par des liaisons appelées articulations. Les chaînes peuvent être dites ouvertes ou connectées en série, dans lesquelles tous les objets ont au plus deux liaisons, ou en forme d'arbre, dans lesquelles au moins un objet a plus de deux liaisons. Des chaînes peuvent également être fermées, où les organes d'extrémité sont reliés au bas du mécanisme par plusieurs chaînes. [6]

I.6.1. Structure mécanique articulée à chaîne cinématique simple :

C'est une sorte de chaîne cinématique dans laquelle le degré de liaison (le nombre de maillons mécaniques) de chaque élément est inférieur ou égal à 2. Le robot tandem est composé d'une chaîne cinématique simple, le degré de connexion de sa base et de son effecteur est de 1 (c'est-à-dire qu'il n'est connecté qu'à un seul corps), et le degré de connexion des autres composants est de 2. [6]



Figure. I.5.Structure sérielle.

I.6.2. Structure mécanique articulée à chaîne cinématiques fermée

C'est une sorte de chaîne cinématique dans laquelle le degré de connexion d'un élément différent de la matrice est supérieur ou égal à trois. [6]



FigureI.6.Structure fermée.

I.7. Classification des robots

Les robots peuvent être classés d'un point de vue fonctionnel ou selon leur structure géométrique.[7]

I.7.1. Classification fonctionnelle

I.7.1.1. Manipulateur à commande manuelle :

Cette figure représente les manipulateurs à commande manuelle :



FigureI.7. Manipulateur à commande manuelle.

I.7.1.2. Manipulateur automatique :

Cette figure montre un bras manipulateur qui exerce des mouvements de soudure sans l'intervention de l'homme.



FigureI.8.Manipulateur automatique.

I.7.1.3. Robots programmables :

Ils répètent des actions qu'ils ont enseignées ou programmées sans aucune information sur l'environnement ou la tâche en cours. On peut également distinguer les robots "playback" qui reproduit les tâches d'apprentissage et les robots CN qui peuvent être programmés hors ligne.

Pour certains robots, comme les robots peintres, ils doivent suivre une trajectoire complexe difficile à exprimer en mathématiques. Un opérateur humain spécialisé dans cette tâche utilise une « marionnette » pour guider le bras du robot pour exécuter la trajectoire, et toute la trajectoire est mémorisée par le robot [7].



FigureI.9.Robot programmable.

I.7.1.4. Robots intelligents

On trouve actuellement des robots de seconde génération qui sont capables d'acquérir et d'utiliser certaines informations sur leur environnement (systèmes de vision, détecteurs de proximité, capteurs d'efforts,...) comme le montre la Figure I.10. Les robots de troisième génération sont capables de comprendre un langage oral proche du langage naturel et de se débrouiller de façon autonome dans un environnement complexe grâce à l'utilisation de l'intelligence artificielle. [7].

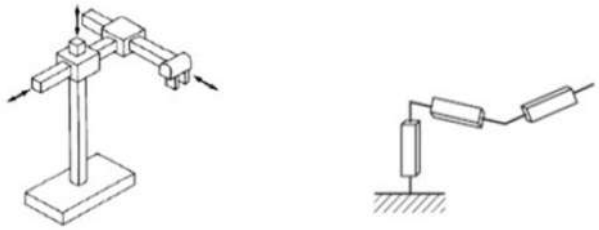


FigureI.10.Robots intelligents.

I.7.2. Classification géométrique

On peut aussi classer le robot selon sa configuration géométrique, c'est-à-dire la structure du porteur. Les 3 premiers ddm du robot peuvent être obtenus par un grand nombre de combinaisons de translation (maximum 3T) et de rotation (maximum 3R), autrement dit, par prisme (P) ou articulations de rotation (R) ; en pratique, seulement 4 d'entre eux sont utilisés Ou 5: [8].

-porteur cartésien (PPP) : les 3 axes sont animés d'un mouvement de translation.[8].



FigureI.11. Structure d'un Robot cartésien

-porteur en coordonnées cylindriques (RPP) : un mouvement de rotation et une translation axiale, complétées par une translation radiale [8].

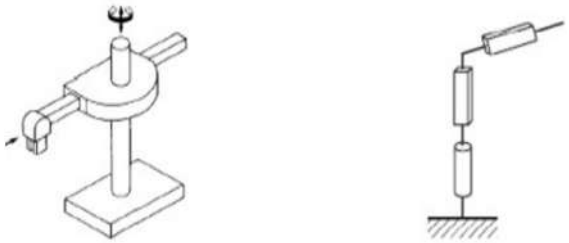


Figure.I.12. Structure d'un robot cylindrique.

- porteur en coordonnées polaires ou sphériques (RRP) : deux rotations (longitude et latitude) autour d'axes orthogonaux, complétées par une translation radiale comme le montre [8].

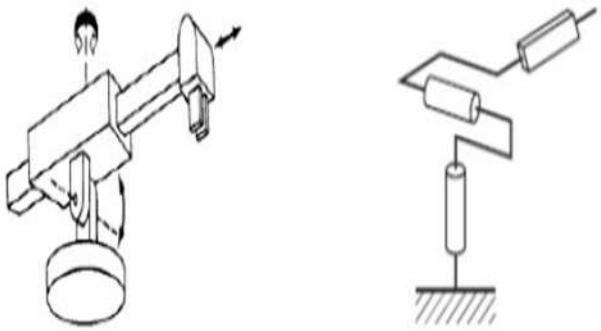
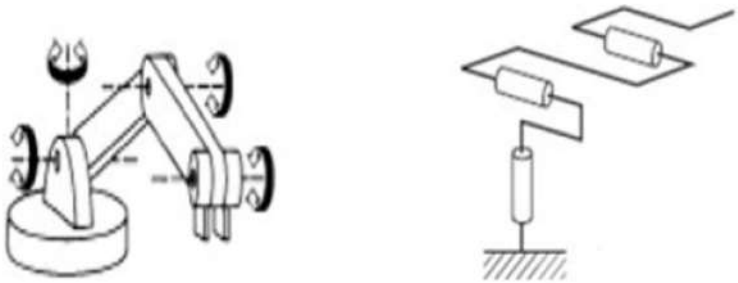


Figure.I.13. Structure d'un robot sphérique.

- porteur en coordonnées universelles, appelé aussi configuration poly articulée ou anthropomorphe (RRR), trois rotations dont les deux ornières se font autour d'axes parallèles orthogonaux au premier, les trois articulations correspondant respectivement au tronc (base), à l'épaule et au coude d'un être humain.[8].



FigureI.14. Structure d'un robot 3R.

-La structure dite SCARA Les caractéristiques d'une structure SCARA (Sélective Complaisance Articulâtes Robots pour l'assemblage) sont : 3 axes en série cylindrique (RRP) ayant trois degrés de liberté. Les particularités de cette structure sont qu'elles sont précises et très rapides [8].

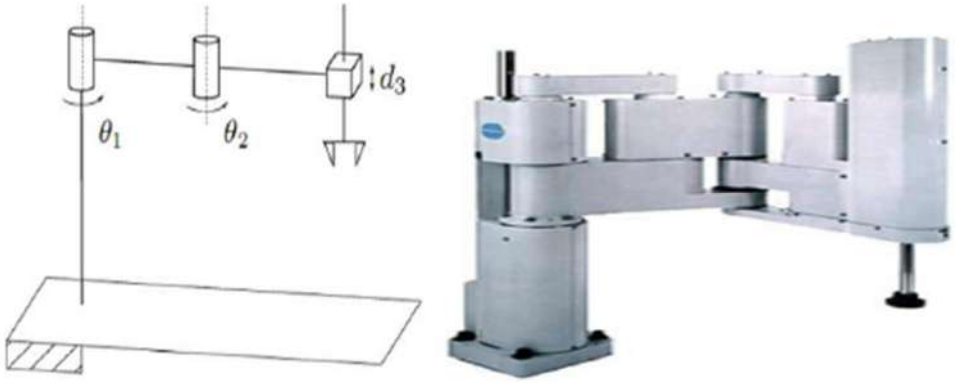


Figure I.15.structure dite SCARA

I.8. Caractéristiques d'un robot

Un robot doit être choisi en fonction de l'application qu'on lui réserve. Voici quelques paramètres à prendre, éventuellement, en compte :

- La charge utile transportable par le robot.
- L'espace de travail : c'est l'ensemble des situations de l'espace que l'organe terminal du robot peut atteindre.

- Les vitesses et les accélérations maximales qui conditionnent le temps du cycle.
- Les performances (l'écart entre une situation commandée et la moyenne des situations s'atteinte).

I.9. Domaines d'applications

Les robots industriels peuvent être utilisés dans plusieurs domaines et peuvent accomplir différentes tâches.

- **Domaine de la production ;**
  - Automatisation de la production. °Meilleur qualité. ° Adaptation au volume de production. °Constance de la qualité.
  - Flexibilité
- **Domaine de l'exploration ;** (BOUZIANE Fatima Zohra 2013)
  - Accès difficile (nettoyage, démantèlement).
  - Milieu hostile (déminage…)
- **Domaine de divers service et médicale.**

I.10. Avantages et inconvénients des robots

Un système robotique consiste non seulement des robots mais aussi d'autres dispositifs et systèmes qui sont utilisés avec le robot pour effectuer la tâche nécessaire.

I.10.1. Avantages

- Robotique et automatisation peut dans de nombreuses situations d'accroître la productivité, la sécurité, l'efficacité, la qualité et la cohérence des produits ;
- Robots peuvent travailler dans un environnement dangereux, sans le besoin de soutien de la vie, ou les préoccupations concernant la sécurité ;
- Robots n'ont pas besoin de l'éclairage, la climatisation, de ventilation et de protection contre le bruit ;
- Robots travail continuellement, sans ressentir une fatigue ou l'ennui, et ne nécessitent pas une assurance médicale ou de vacances ;
- Robots sont de précision répétable à tous les moments, sauf si quelque chose arrive à eux ou ils s'usent ;

- Robots peuvent être beaucoup plus précis que les humains. Précision linéaire d'un robot typiquement est de 20 à 10 microns ;

I.10.2. Inconvénients

L'inconvénient des robots est qu'ils manquent de capacité de réagir en cas d'urgence, à moins que les situations comprises et les réponses sont inclut dans le système. Les mesures de sécurité nécessaires pour s'assurer qu'ils ne lèsent pas les opérateurs et n'endommagent les machines qui travaillent avec eux. °Réponse inadéquate ou mal ; °Manque de pouvoirs prendre une décision ; ° Consommation de l'énergie ;

- Peuvent causer des dommages à des autres appareils, et la blessure de l'homme ;
- Bien que les robots ont de bonnes certaines caractéristiques mais aussi ont ces caractéristiques limités comme la capacité à degré de liberté, la dextérité, capteurs, système de vision et la réponse en temps réel.

Robots sont coûteux en raison du: coût initial de l'équipement, le coût d'installation, le besoin de périphériques, le besoin de formation et la nécessité de la programmation.

I.11.Conclusion

Le robot est un manipulateur à plusieurs degrés de liberté, à commande automatique, reprogrammable, multi-applications, mobile ou non, destiné à être utilisé dans les applications d'automatisation industrielle.

Le robot est un système articulé rigide est caractérisé par une structure arborescente articulée simple ou multiple dont les liaisons sont mobiles les unes par rapport aux autres. Ce système a pour objectif de mener l'organe terminal vers un lieu géométrique imposé par la tâche.

Chapitre II : Conception et réalisation du bras manipulateur

II.1. Introduction

Après avoir exposé les généralités sur les robots, nous présenterons dans ce chapitre une étude de la conception de pièces mécaniques sur le bras manipulateur réalisé, tous les matériaux utilisés et quelques explications de choses complexes.

La conception mécanique de ces pièces se fera par des outils informatiques de CAO dans ce cas on va utiliser SolidWorks. On va donner les étapes de modélisation de chaque pièce (au moins 5 pièces détaillées vue le nombre important de pièces). Après on a passé à la réalisation mécanique réelle.

Avant de passer à la rétro conception de ces pièces on va d'abord présenter un rappel sur la modélisation géométrique d'une façon générale et l'outil CAO utilisé.

II.2. Choix de l'outil informatique de CAO

Pour la modélisation géométrique (conception assistée par ordinateur) des différentes pièces du bras du robot on va utiliser le logiciel de CAO SolidWorks à cause de la disponibilité de ce dernier et les connaissances acquises durant notre formation.

II.3.logiciel SolidWorks

II.3.1. Présentation du logiciel SolidWorks

Le logiciel de conception mécanique SolidWorks est un outil de conception de modélisation solide paramétrique basé sur des fonctionnalités, qui utilise les fonctionnalités de Windows TM connues pour leur facilité d'utilisation. Des modèles solides 3D entièrement intégrés peuvent être créés avec ou sans contraintes, tout en utilisant des relations automatiques ou définies par l'utilisateur pour capturer l'intention de conception.

Le modèle SolidWorks est entièrement intégré aux dessins techniques et aux assemblages qui s'y réfèrent. Les modifications apportées au modèle sont pleinement reflétées dans les dessins et les assemblages qui lui sont associés. Inversement, si vous apportez des modifications dans le contexte d'un dessin ou d'un assemblage, ces modifications seront répercutées dans le modèle. Les relations géométriques (telles que les relations parallèles, verticales, horizontales, verticales, concentriques et coïncidentes) sont des exemples de

contraintes supportées par SolidWorks. Les équations peuvent également être utilisées pour établir des relations mathématiques entre les paramètres. De cette manière, nous pouvons garantir que le concept sera établi et préservé. [9]

Figure II.1:Logo SolidWorks

II.3.2.Fonctionnement

SolidWorks est un modeleur 3D utilisant la conception paramétrique. Il génère 3types de fichiers relatifs à trois concepts de base : la pièce, l'assemblage et la mise en plan. Ces fichiers sont en relation. Toute modification à quelque niveau que ce soit est répercutée vers tous les fichiers concernés. [9].



FigureII.2: fenêtre principale de SolidWorks

II.3.2.1. Conception des pièces

Les pièces se conforment à partir d'une esquisse dessinée sur un plan. Cette esquisse 2D est obtenue par des formes géométriques simples comme lignes et cercles. Une fois l'esquisse prête, le logiciel offre plusieurs fonctions à exécuter. On peut la prolonger vers une direction (extrusion), la faire tourner autour d'un axe ou réaliser plusieurs autres opérations.

Le dessin est paramétré, c'est-à-dire qu'on peut modéliser la forme souhaitée sans se préoccuper des dimensions exactes. Les dimensions des différentes pièces peuvent être liées de façon que lorsqu'une change, les autres changent aussi.[9].

II.3.2.2. L'assemblage

A mesure qu'on réalise les pièces, on peut les assembler dans un nouveau fichier qui contiendra seulement les contraintes et rapports qu'on a imposé aux pièces. Les contraintes définissent la position des pièces dans l'ensemble et les liaisons entreillés. On peut donc imposer que deux pièces soient concentriques, que deux surface soient coplanaires ou par exemple que la distance entre deux surfaces soit déterminée.

On peut créer un assemblage d'assemblages et comme cela construire des grands ensembles de façon structurée et ordonnée.[9].

II.3.2.3. La mise en plan

Pour que l'atelier puisse réaliser les pièces, il à besoin de plans en 2D plutôt que d'un dessin tridimensionnel dont la cotation serait assez confuse .SolidWorks possède un module capable de projeter des vues de pièces ou d'ensembles sur un plan. Ensuite on peut faire des coupes et d'autres opérations pour mieux représenter la pièce sur le plan.

La cotation est automatique même si elle n'est pas toujours optimisée pour l'usinage. Un changement sur une cote du plan entraîne la modification automatique de cette dimension sur la pièce d'origine et vice-versa.[9].

II.4.Conception des pièces et des différentes articulations du bras

La conception du robot est une phase très importante. Elle doit se faire de façon a réponde aux spécifiés de la feuille de route adoptée tout en assurant le minimum de contraintes aux phases qui en découleront-réalisation, modélisation et commande. Dans cette partie, nous expliquerons toutes les mesures et les dessins auxquels nous sommes parvenus.

II.4.1. la base

La base est le socle fixe du bras, elle contient deux plateaux de forme différente et entre eux se trouvent des piliers soutenant la structure générale, et nous mettons un servomoteur dans l'espace entre eux et installons un disque dessus afin de contrôler et afin de placer le reste des autres articulations sur le disque.

- plateau inférieur

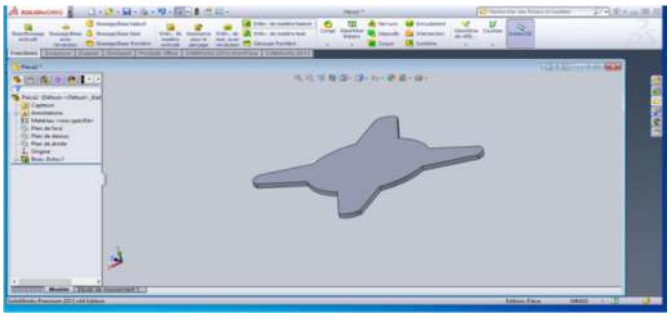


Figure II.3: plateau inférieur du la bas sous SolidWorks

Rayon du plateau inférieur =80 mm

Longueur de l'aile=35 mm

La distance entre chaque aile =60 mm

Epaisseur = 3 mm

- piliers

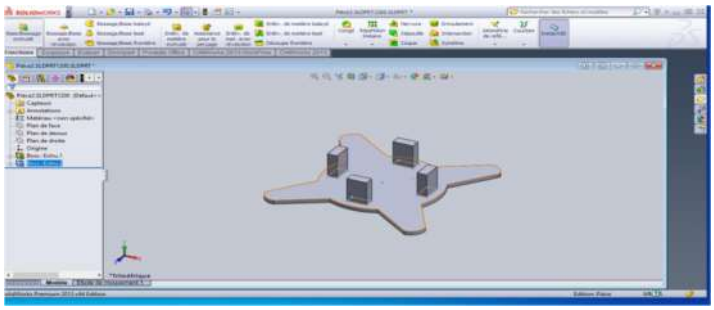


Figure II.4: les piliers et plateau inférieur du la bas sous SolidWorks

Hauteur du pilier = 53 mm

Longueur du pilier = 30 mm

Largeur du pilier = 15 mm

- plateaux supérieur

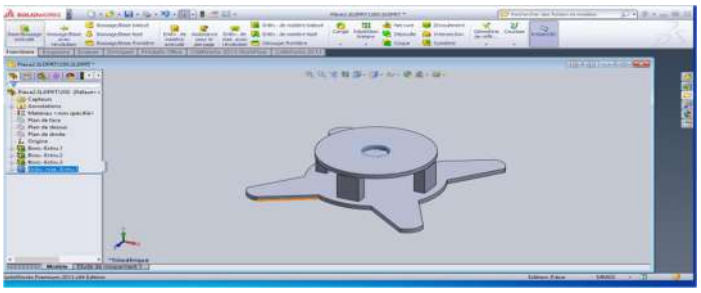


Figure II.5: les piliers et plateaux du la bas sous SolidWorks

Rayon du plateau supérieur = 75 mm

Epaisseur = 5 mm

II.4.2L'épaule

C'est la deuxième partie du bras manipulateur constitué d'un bras vertical et d'un disque rotatif qui est relié à une base. Ilya deux supports pour le servomoteur sur le disque. le lien entre le servomoteur qui lui permet de bouger.

- disque rotatif avec support servomoteur

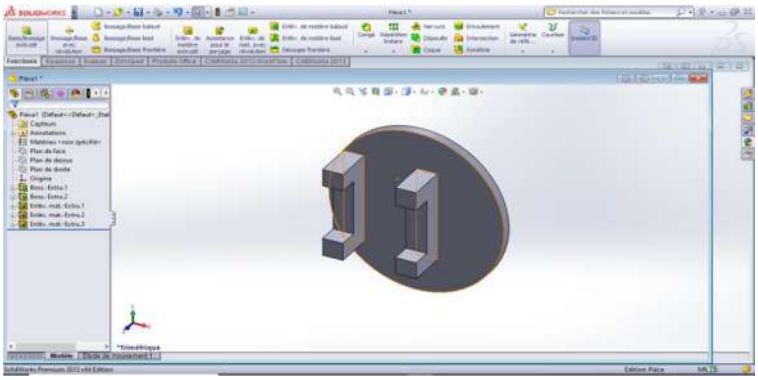


Figure II.6:Disque avec support servomoteur sur SolidWorks

Rayon du disque= 57 mm

Epaisseur = 5 mm

Hauteur du support = 35 mm

Longueur du support = 60 mm

Largeur du support = 5 mm

- bras vertical

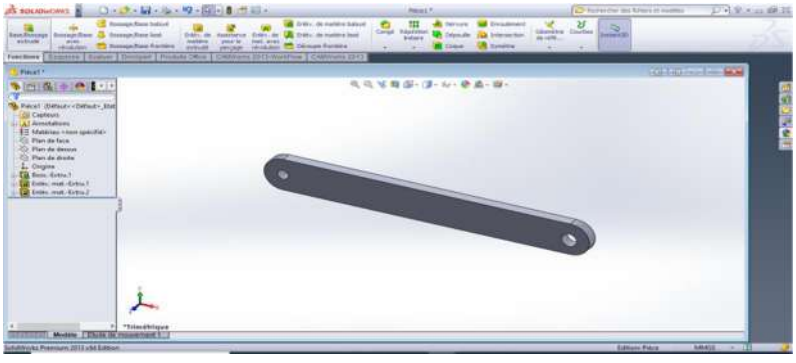


Figure II.7: Morceau de bras vertical sur SolidWorks

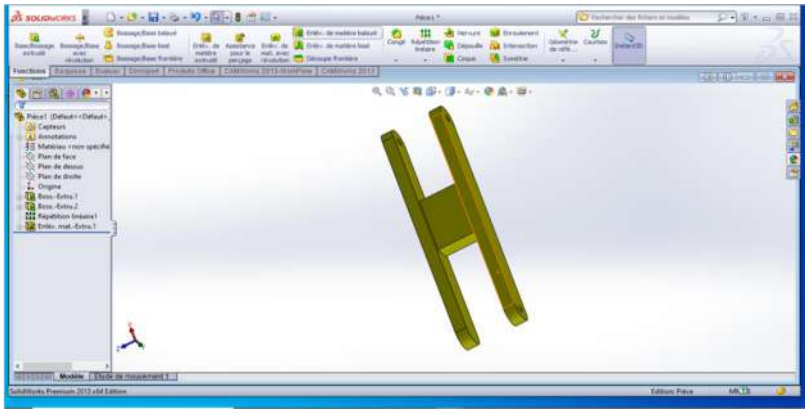


Figure II.8:Bras vertical du L'épaule sur SolidWorks

Hauteur du bras = 35 mm

Longueur du bras = 175 mm

Largeur du bras = 5 mm

Rayon du trou = 4 mm

Hauteur du la pièce entre les bras = 55 mm

Longueur du la pièce entre les bras= 55 mm

Largeur du la pièce entre les bras= 5 mm

II.4.3.Le coude

C'est la troisième partie du bras manipulateur constitué d'un bras horizontal auquel est fixé un servomoteur, ce qui lui permet de se déplacé de haut en bas.

- bras horizontal



Figure II.9: Morceau de bras vertical sur SolidWorks

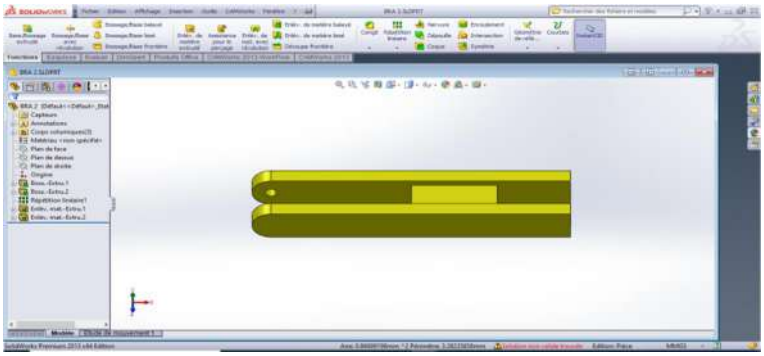


Figure II.10:Bras vertical du L'épaule sur SolidWorks.

Hauteur du bras = 32 mm

Longueur du bras = 155 mm

Largeur du bras = 10 mm

Rayon du trou = 4 mm

Hauteur du la pièce entre les bras = 35 mm

Longueur du la pièce entre les bras = 30 mm

Largeur du la pièce entre les bras = 10 mm

II.4.4.poignet avec la pince

C'est la dernière partie du bras manipulateur et nous en avons fait une partie fixe et il y a de la place pour le servomoteur de la pince.

La pine est un élément terminal constitué d'une pince avec une extrémité fixe et l'autre mobile au moyen d'un servomoteur qui commande son ouverture et sa fermeture.

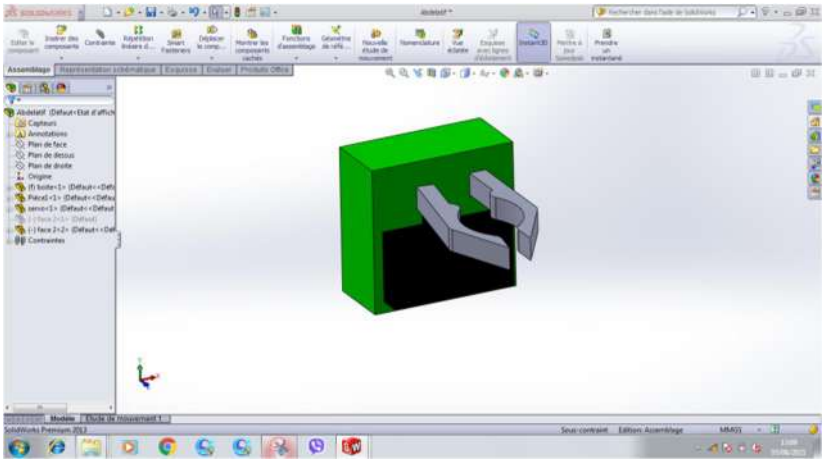


Figure II.10: poignet avec pince sur SolidWorks

II.5.Assemblage des pièces du bras manipulateur

Nous assemblerons les pièces de notre bras manipulateur que nous avons créé tous ses dessins sur SolidWorks 3D et la fonction d'assemblage des pièces nous permet d'analyser les processus mécanique des installations robotique et les forces physiques générées par celles-ci.

Nous avons divisé nos collections en plusieurs étapes:

Etape 1 :

On va installer les pièces de la base et placer le servomoteur entre deux plateaux, puis placez le disque rotatif d'épaule sur le lève-moteur du servomoteur.

Montage de deux servomoteurs sur support situées au-dessus du disque rotatif comme le montre la figure II.11.

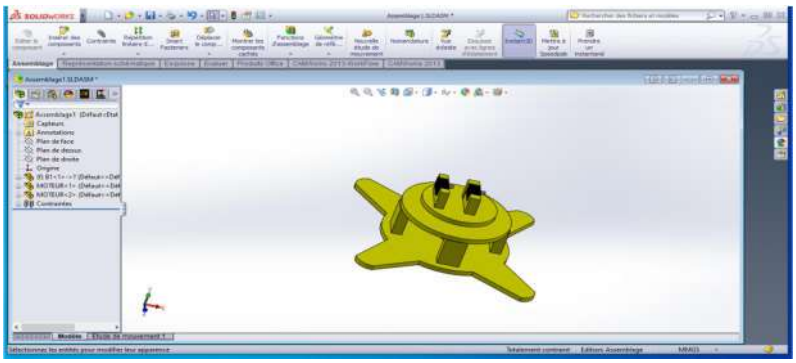


Figure II.11: poignet avec pince sur SolidWorks

Etape 2 :

On va installer bras vertical dans les élévateurs des deux servomoteurs. Ensuite nous obtenons une épaule attachée à la base comme le montre la figure II.12.

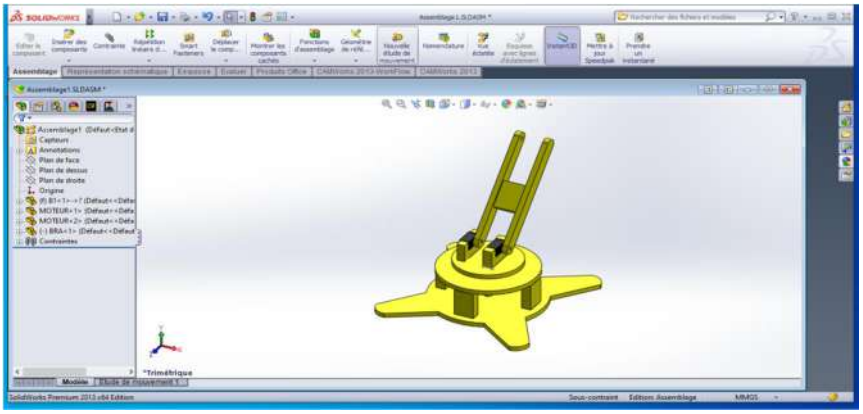


Figure II.12: Base avec épaule sur SolidWorks

Etape 3 :

On va installer bras horizontal avec le bras vertical et connectez a' leur point de jonction un servomoteur et nous obtenons une coude avec une base et une épaule comme le montre la figure II.13

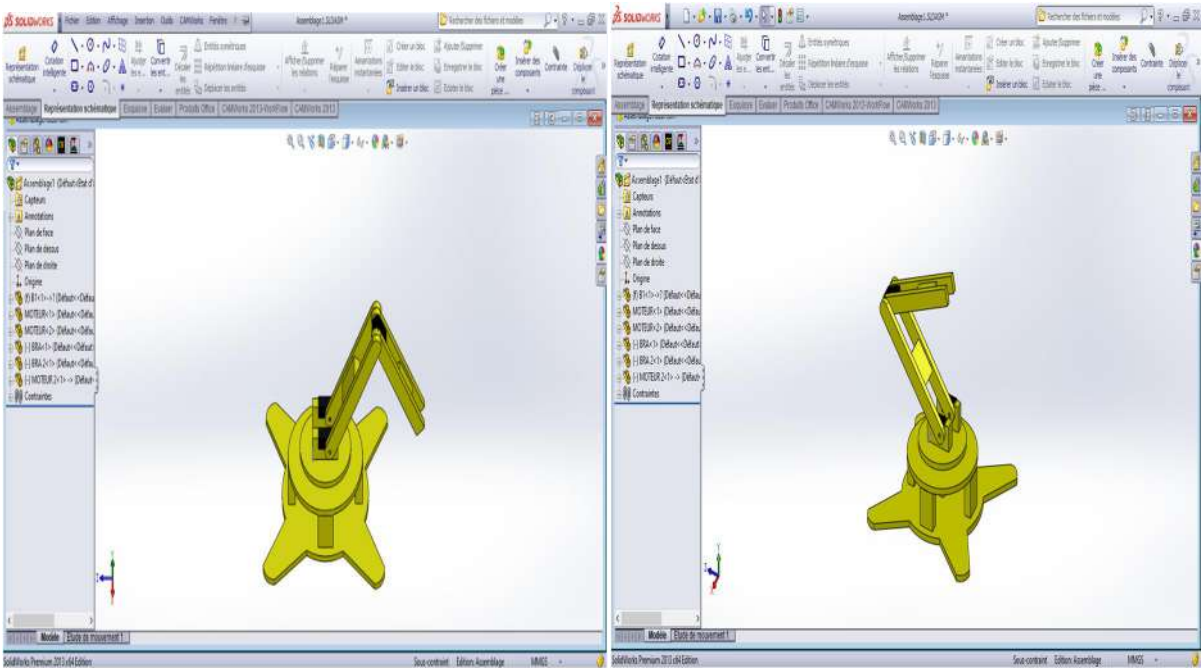


Figure II.13: Le bras manipulateur que nous avons sur SolidWorks

Etape 4 :

On va installer la pièce composite du poignet et le pince avec le bras que nous avons afin d'obtenir la fourme finale a réalisé après comme le montre la figure II.14

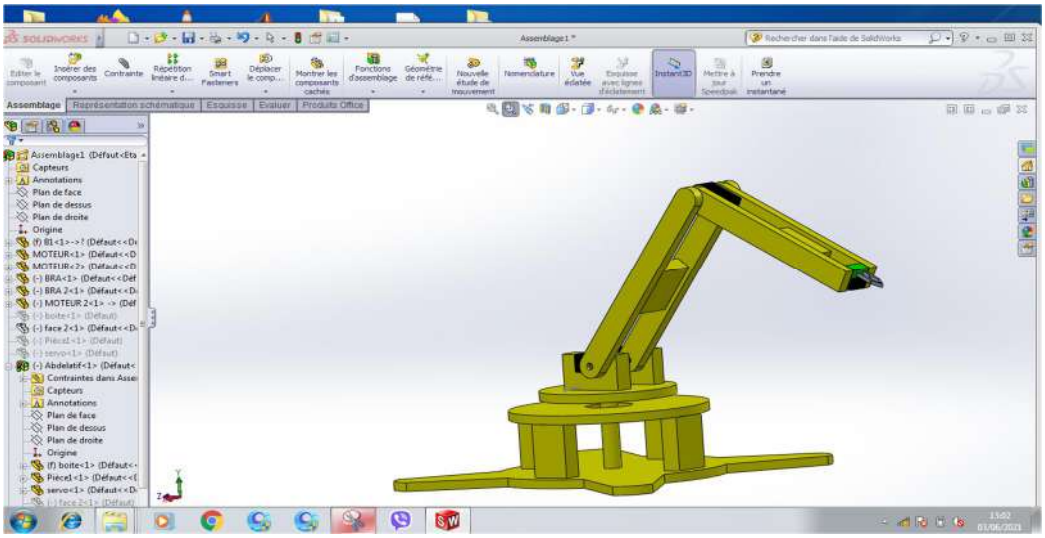


Figure II.14: La forme finale de notre du bras manipulateur sur SolidWorks

II.6.Réalisationréelle de la Structure mécanique du bras manipulateur

Après le succès de notre travail sur la conception et l'assemblage nous sommes passés à la réalisation que ces opération nous facilitaient la réalisation car nous choisissions l'impression 3D, nous ne prendrions que les fichiers. Pour le coût de cette technologie, et fait notre travail manuel et choisi le matériau bois en raison de sa disponibilité et de sa faible densité, nous l'avons rendu solide pour résister aux chocs et assez léger pour faciliter le transport. Nous avons conçu une structure mécanique selon les étapes que nous suivies dans la conception et l'assemblage. On a obtenu le bras manipulateur et la figure suivante montre notre travail.

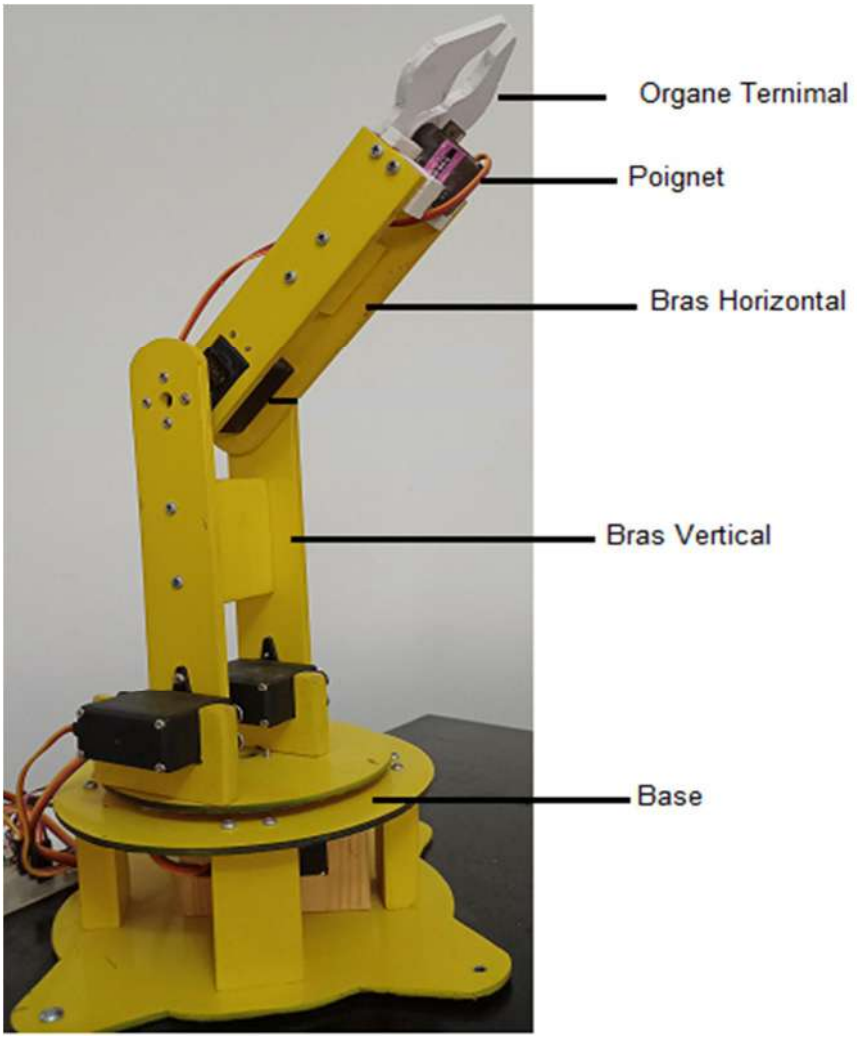


Figure II.15: Réalisation final elles différents organes du bras manipulateur

II.7.Conclusion

L’étape de conception est généralement la première car elle varie selon l'environnement de travail, et nous avons recherché toujours le modèle le plus simple qui puisse atteindre et expliquer notre imagination et nos objectifs, et nous avons utilisé tous les programmes et méthodes simples pour dessiner et réalise sur le bras manipulateur. Quant à l'étape de réalisation, c'était selon nos capacités et c'était une étape pleine d'obstacles et de problèmes qui nous ont retardés dans terminer le projet.

Chapitre III :
Etude technique des composants du bras manipulateur

III.1.Introduction

L'étude technique des composants constitue l'une des étapes de la conception et de l'analyse de faisabilité d'un projet. Elle représente un préalable dans le sens où elle jette les premières bases du projet. Toutefois, elle est amenée à évoluer en fonction des différentes informations et des conseils qui seront reçus tout au long de la préparation du projet. Elle doit apporter une clarification à un ensemble d'interrogations d'ordre technique.

Dans ce chapitre, nous allons expliquer et définir tous les appareils que nous avons utilisés et clarifier toutes l'opération liée aux schémas Schéma électrique (Montage, câblage…etc.) Des appareils en théorie, puis passer à l'étape de réalisation et d'assemblage ces appareils sur la structure de notre bras que nous avons réalisée.

III.2.Servomoteurs

III.2.1.Présentation du servomoteur

❖ Les servomoteurs sont un système électrique qui peut atteindre des positions prédéterminées et les maintenir. La position est : dans le cas d'un moteur rotatif, la valeur de l'angle, dans le cas d'un moteur linéaire, la distance. Utiliser des moteurs électriques (continus, asynchrones, à bagues) et des moteurs hydrauliques. Le démarrage et le maintien de la position prédéterminée sont commandés par le système de régulation. [10].

❖ Un servomoteur est un ensemble complet de mécanique et d'électronique, qui contient : [10].

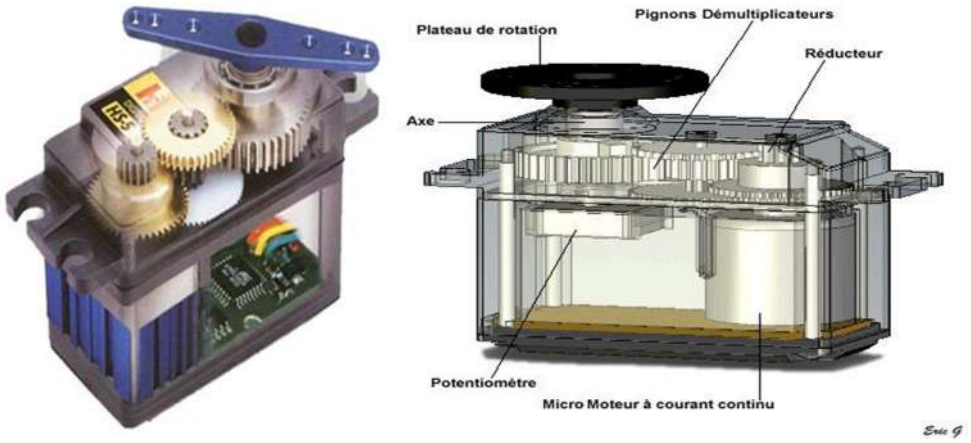


Figure III.1:Schémas des éléments non visibles d'un servomoteur.

- Les fils, qui sont au nombre de trois (comme la Figure III.2. montre)
  - rouge : pour l'alimentation positive (4.5V à 6V en général)
  - noir ou marron : pour la masse (0V)
  - orange, jaune, blanc, G : entrée du signal de commande.

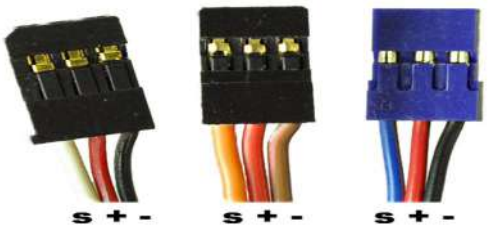


Figure III.2: Certains types du fils de servomoteur.

- L'axe de rotation sur lequel est monte un accessoire en plastique ou en métal
- Le boitier qui le protège.
- un moteur à courant continu.
- des engrenages pour former un réducteur (en plastique ou en métal).
- un capteur de position de l'angle d'orientation de l'axe (un potentiomètre bien souvent)
- une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu.

❖ Le servomoteur est contrôlé par un câble à trois conducteurs, qui sert à alimenter le moteur et à lui envoyer des commandes de position sous forme de signaux codés en largeur d'impulsion, généralement appelés PWM (modulation de largeur d'impulsion, modulation de largeur d'impulsion). Cela signifie que la durée de l'impulsion détermine l'angle absolu de l'arbre de sortie et la position du bras de commande de l'actionneur. Ce signal se répète périodiquement, généralement toutes les 20 millisecondes, ce qui permet au dispositif électronique de contrôler et de corriger en permanence la position angulaire de l'arbre de sortie, qui est mesurée par un potentiomètre. Lorsque le moteur tourne, la position de l'arbre du servomoteur change, modifiant ainsi la résistance du potentiomètre. La fonction du composant électronique est de contrôler le moteur pour que la position de l'arbre de sortie soit cohérente avec la commande reçue, qui est une sorte d'asservissement. [10]

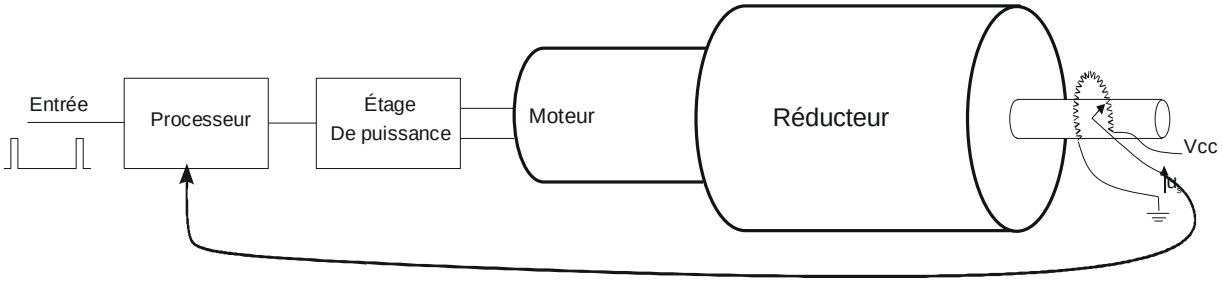


Figure III.3: Principe d'un servomoteur

❖ Le signal de commande est donné sous forme d'impulsions de largeur variable, correspondant à l'angle de rotation souhaité. Afin de maintenir une position donnée, le signal doit être périodique avec une fréquence comprise entre 50 et 100 Hz. L'angle de rotation maximum peut varier d'un modèle à l'autre, mais il est généralement compris entre [0°, 180°]. la figure 3.4 donne des exemples de signaux pour 3 positions de l'axe de rotation.

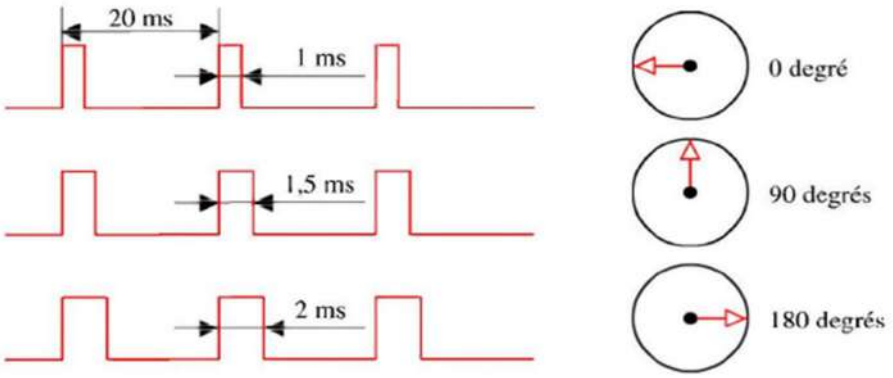


Figure III.4: Signaux et positions d'un servomoteur

❖ Dans notre réalisation nous avons utilisé deux types différents des servomoteurs qui sont illustrés ci-dessous et placés sur notre bras manipulateur comme indiqué sur la figure III.7.

- types1: Servomoteur MG995



Figure III.5: Servomoteurs MG995 utilisés dans le bras manipulateur

- types 2: Servomoteur MG90S



Figure III.6: Servomoteurs MG90S utilisés dans le bras manipulateur

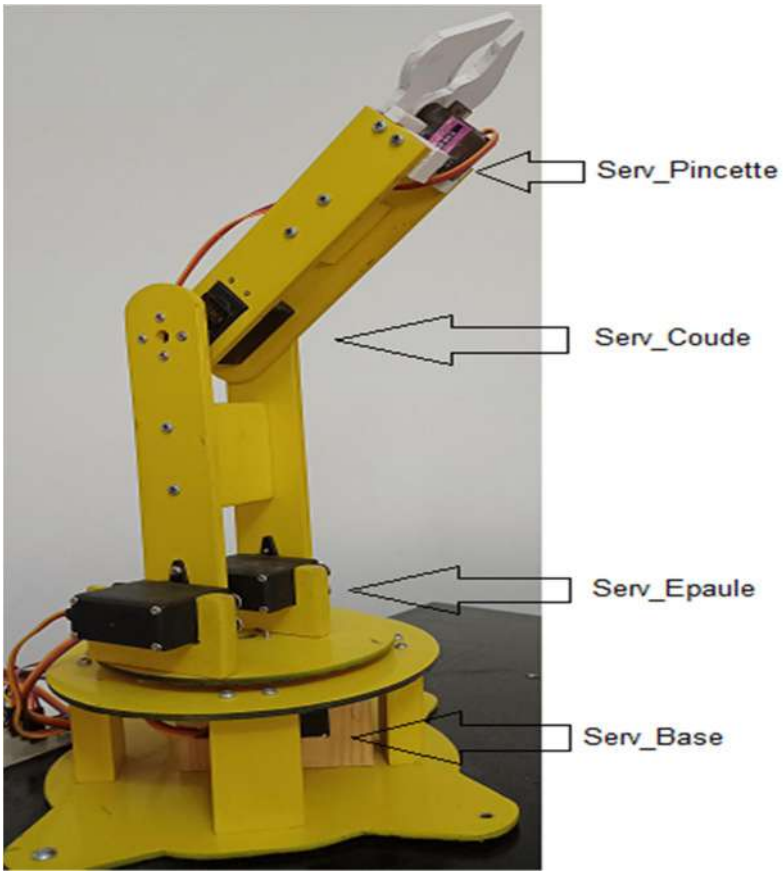


Figure III.7: Emplacement du servomoteur sur le bras robots

III.2.2.Les paramètres techniques des servomoteurs utilisés

Les paramètres techniques des servomoteurs utilisés sont donnés par **Tab.III.1**

Tab.III.1. Tableau des caractéristiques des servomoteurs du bras

Références / Caractéristiques	MG995	MG90S
poids	60g	13.5 g
Dimensions	40 x 20 x 36,5 mm	32 x 12 x 32 mm
Couple d'arrêt	11Kg/cm sous 4.8V - 13Kg/cm sous 6.0V	2.5 kgf.cm
Tension De fonctionnement	4.8V - 6V	4.8V-6V
Vitesse	0.16 sec/60°	0.1s/60°
Angles de rotation	180°	180°

III.2.3.Espace opérationnel

Après avoir installé les servomoteur et réalisé quelques études, nous avons pu déduire les limites de rotation maximale et minimale pour chaque organe du bras et nous a également permis de connaitre la circonférence qu'il couvre dans son travail, et la fixation de ces limites est une partie essentielle de la fabrication d'un bras manipulateur, comme il s'agit d'une protection contre les collisions et interférences, le tableau suivant indique les angles de rotation pour chaque organe de notre bras.

Tab.III.2. Tableau des limites de rotation du bras

ROTATION	MIN	MAX
BASE	0°	180°
EPAUL	100°	20°
COUDE	170°	110°
PINCE	25°	70°

III.3.Le module Bluetooth

III.3.1.Présentation du récepteur Bluetooth (HC-06) :

HC-06 est un module Bluetooth conçu pour établir une connexion de données sans fil à courte portée entre deux microcontrôleurs ou systèmes. L'appareil fonctionne sur le protocole de communication Bluetooth 2.0 et ne peut fonctionner qu'en tant qu'appareil esclave. C'est le moyen le moins cher et le plus flexible de transférer des données sans fil par rapport à d'autres méthodes et il peut également transférer des fichiers à des vitesses allant jusqu'à 2,1 Mo / s.

HC-06 utilise la technologie de propagation du spectre à sauts de fréquence (FHSS) pour éviter les interférences avec d'autres appareils et pour la transmission en duplex intégral. L'appareil fonctionne sur une gamme de fréquences de 2,402 GHz à 2,480 GHz.[11]

On a utilisé un récepteur Bluetooth (HC-06) afin de pouvoir commander ses déplacements à l'aide d'une application mobile.



Figure III.8: .Bluetooth (HC-06)

III.3.2.Les paramètres techniques du récepteur Bluetooth (HC-06) :

Les spécifications techniques du composant Bluetooth utilisé sont données par **Tab.III.2**

Tab.III.3. Tableau des caractéristiques du récepteur Bluetooth du bras.

Caractéristiques	**Bluetooth HC-06**
Sensibilité typique	-84dBm
Puissance d'émission RF	Jusqu'à 4dBm
Tension de fonctionnement	3.3 (2.7V à 4.2V)
Couverture du signal	30 pieds
Dimensions	15,5 x 39,8mm

III.3.3.Communication de l'application avec l'Arduino

Après l'installation de notre application Android et après avoir établie la connexion avec le module Bluetooth, l'application envoie une information d'état au Bluetooth et qui lui-même, par la suite, la transmettra à l'Arduino à travers une liaison série. [11]

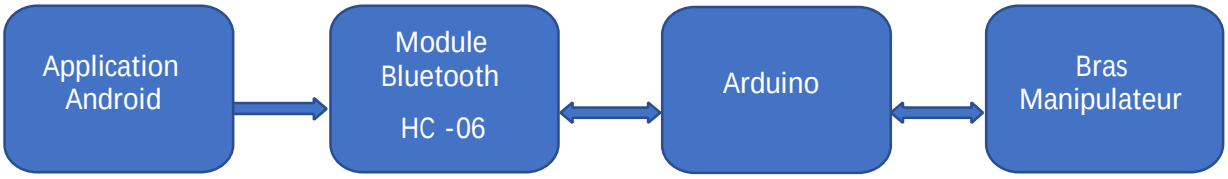


Figure III.9.Conclusion Schéma synoptique simplifié du système

Afin de gérer la communication avec le module HC-05, on connecte les broches Rx et Tx de la carte aux broches d'émission/réception de notre module Bluetooth. La broche Rx du module est branchée sur la broche Tx de l'Arduino et inversement.

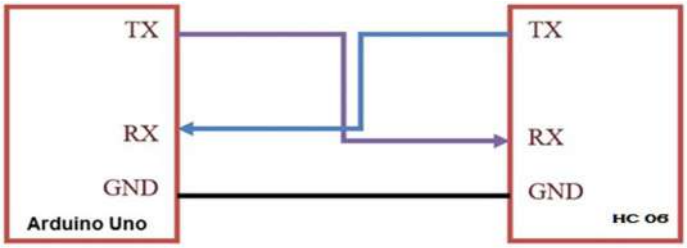


Figure III.10.Schéma de la communication avec le module Bluetooth HC-06

III.4. Carte Arduino

Arduino est une carte programmable qui peut être connectée à des capteurs (pression, luminosité, mouvement, etc.) pour déclencher les actions de moteurs, diodes, écrans d'affichage, etc. Les algorithmes et les activités de programmation peuvent trouver des applications spécifiques dans la robotique, les expériences scientifiques ou les réalisations artistiques. Les réalisations autour d'Arduino peuvent être très simples : jeux de lumière, télémètres sonar ou instruments de musique ne nécessitent aucune compétence technique, mais nécessitent de travailler sur des algorithmes qui s'inscrivent entièrement dans le cadre d'un nouveau logiciel. Dans le cadre d'options EPI, ICN exploration, ou ISN, nous pouvons construire des projets plus ambitieux, comme des drones, des stations météo ou des systèmes domotiques. [12]

Arduino présente de nombreux avantages :

- Grâce à l'interface de programmation intuitive par blocs, il est utilisé pour les petits projets que tout le monde peut réaliser.
- elle permet de construire des projets ambitieux car elle dispose de nombreuses entrées/sorties et peut se programmer avec un langage très complet ;
- elle est très populaire et l'Internet fourmille de tutoriels et d'idées de projets ;
- elle est solide ;
- elle est peu onéreuse.



Figure III.11:Types des cartes arduino

III.4.1 La carte Arduino UNO :

Arduino UNO est une carte de développement avec microcontrôleur Atmel ATMEGA328P, couramment utilisée par le grand public. [13]

L'avantage de cette carte est d'éviter l'utilisation de composants nus. Les composants périphériques nécessaires existent déjà, il n'est donc pas nécessaire de les souder. De plus, ATMEGA est très complet et simple d'utilisation. Enfin, il est aussi très bon marché et très réussi. [13]

Le microcontrôleur contient notamment :

• Une CPU (Central Processing Unit). • Mémoire ROM.

• Mémoire RAM. • entrée sortie.

La figure III.12 suivant pressante La carte ArduinoUNO

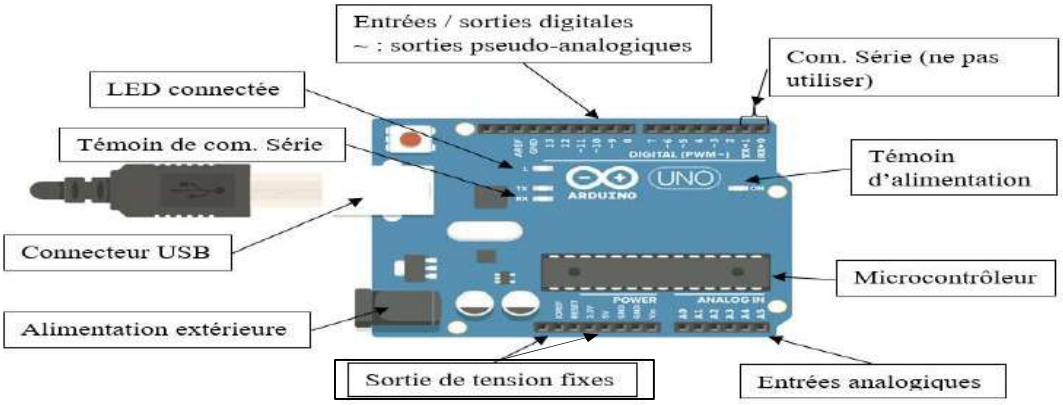


Figure III.12: La carte ArduinoUNO [13].

III.4.2.Les paramètres techniques de la carte Arduino UNO :

Les paramètres techniques donnes par **Tab.III.4**

Tab.III.4. Caractéristiques de l'Arduino UNO

Microcontrôleur	ATmega328
Tension de fonctionnement	5V
Tension d'alimentation	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	14 (dont 6 fournissent une sortie PWM)
Broches d'entrées analogiques	6
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	40 mA
Mémoire programme Flash	32 KB (dont 0.5 KB utilisés par le chargeur de démarrage)
Mémoire SRAM (mémoire volatile)	2 KB
Mémoire EEPROM (mémoire non volatile)	1 KB
Vitesse d'horloge	16 MHz
Poids	25 g
Dimensions	68,6x53, 4 mm

III.5.Schéma de câblage des composants du bras manipulateur

Connecter les appareils du bras manipulateur entre eux se fait en suivant les étapes et l'instruction correctes, es tout erreur s'est produite qui est loin de la cible prévue et endommage l'appareil. Dans ce toute partie, nous expliquerons comment connecter les fils de l'appareil de notre bras manipulateur les uns aux autres et mettre des schémas expliquant notre travail

- câblage du servomoteur sur la carte Arduino UNO
  - Les valeurs nominales actuelles indiquent que nous ne pouvons pas connecter directement ce servomoteur à l'Arduino, mais nous devons utiliser une alimentation séparée pour celui-ci.
  - Nous devons simplement connecter la broche de contrôle du servomoteur à une broche numérique de la carte Arduino, connecter la terre et les fils positifs à l'alimentation externe 5 V, et également connecter la terre Arduino à la masse du servomoteur.

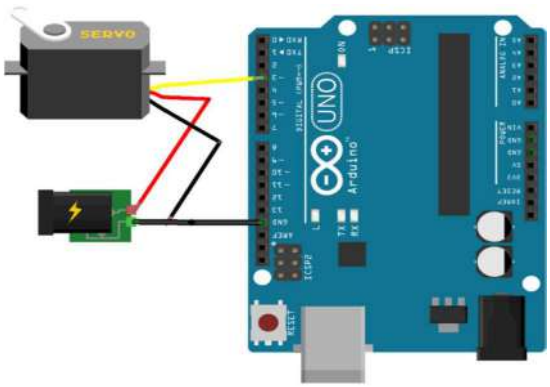


Figure III.13: Montage ArduinoUno + servomoteur

- câblage du Bluetooth (HC-06) sur la carte Arduino UNO
  - Le module HC-06 est un module Bluetooth SPP (Serial Port Protocol), ce qui signifie qu'il communique avec l'Arduino via la communication série.
  - Le peut être alimenté de 2.6 à 4.2 volts, car il vient sur une carte de dérivation contenant un régulateur de tension. Cependant, le niveau de tension logique des broches de données est de 3.3V. Ainsi, la ligne entre l'Arduino TX (broche de transmission, qui a une sortie 5V) et le module Bluetooth RX (broche de réception, qui supporte seulement 3.3V) doit être connectée via un diviseur de tension afin de ne pas graver le module. D'autre part, la ligne entre la broche TX du module Bluetooth et la

broche Arduino RX peut être connectée directement car le signal 3,3 V du module Bluetooth est suffisant pour être accepté comme une logique haute sur la carte Arduino

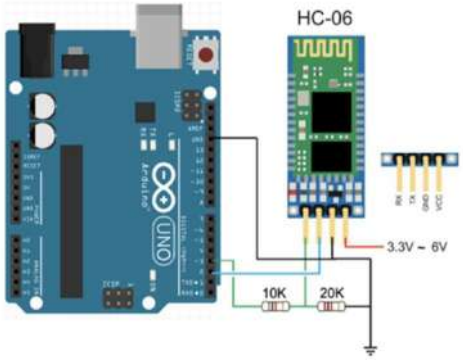


Figure III.14: Câblage du Module Bluetooth HC06 sur ArduinoUno

- câblage du le montage général du Bras manipulateur

Donc pour résumer nous utiliserons 3 servomoteursMG995, et une micro servomoteur MG90S, un module Bluetooth HC-06, une carte ArduinoUno, le tout connecté à notre application Android pourle contrôle. Ci-dessous le schéma électrique complet.

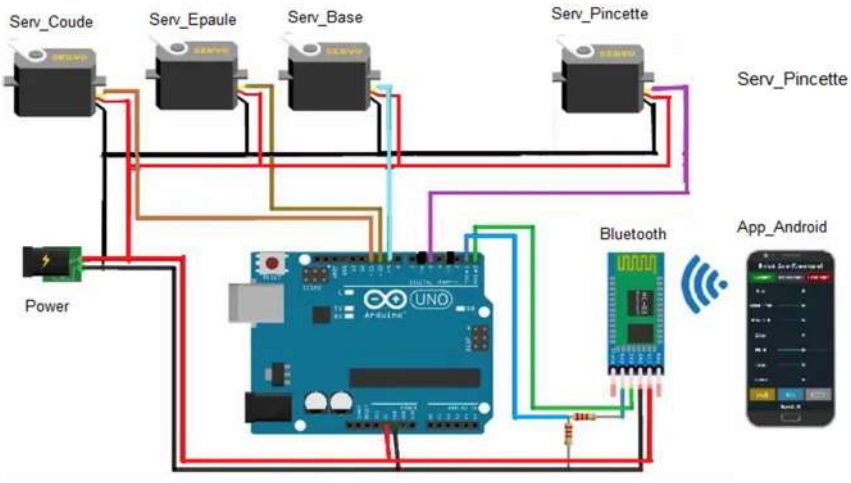


Figure III.15.Schéma électrique globale du Bras manipulateur

III.6. .Réalisation du la carte de commande pour le bras manipulateur

La figure suivant pressante la réalisation final d'un carte de commande (partie électronique) d'un bras manipulateur à constitue d'un arduino Uno, module Bluetooth HC_06, alimentions externe….etc.

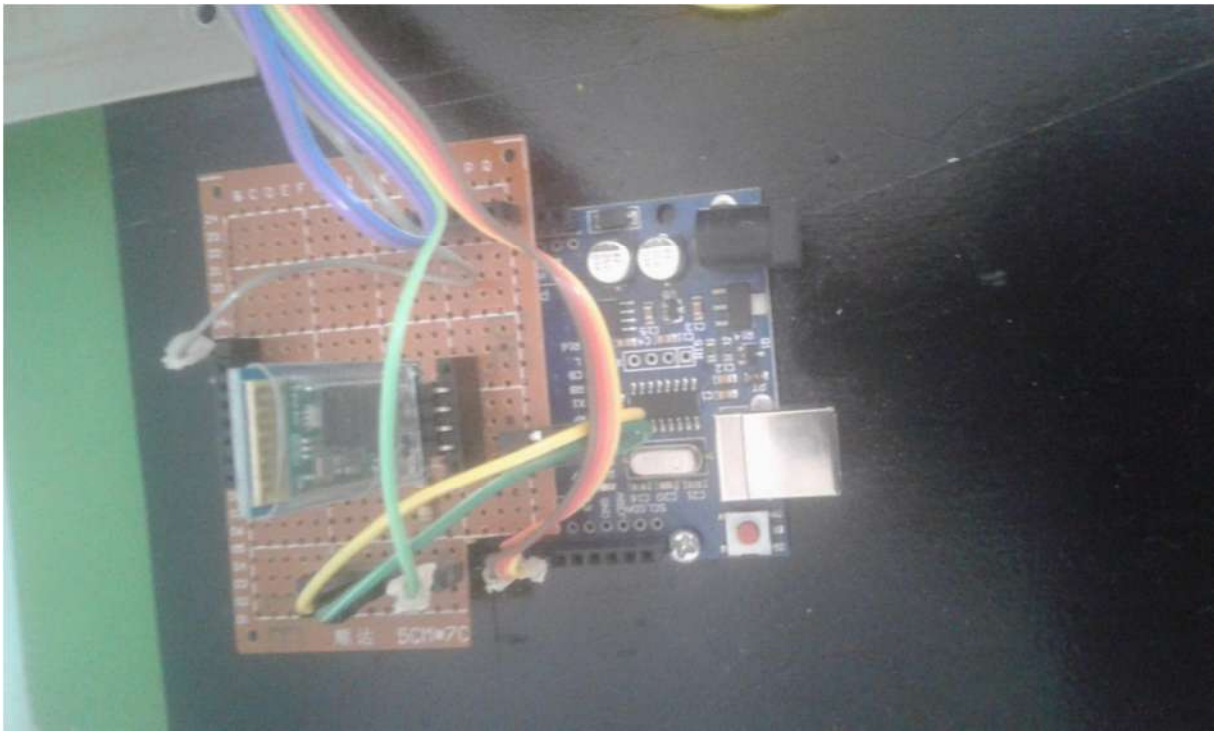


Figure III.16.Carte de commande d'un bras manipulateur.

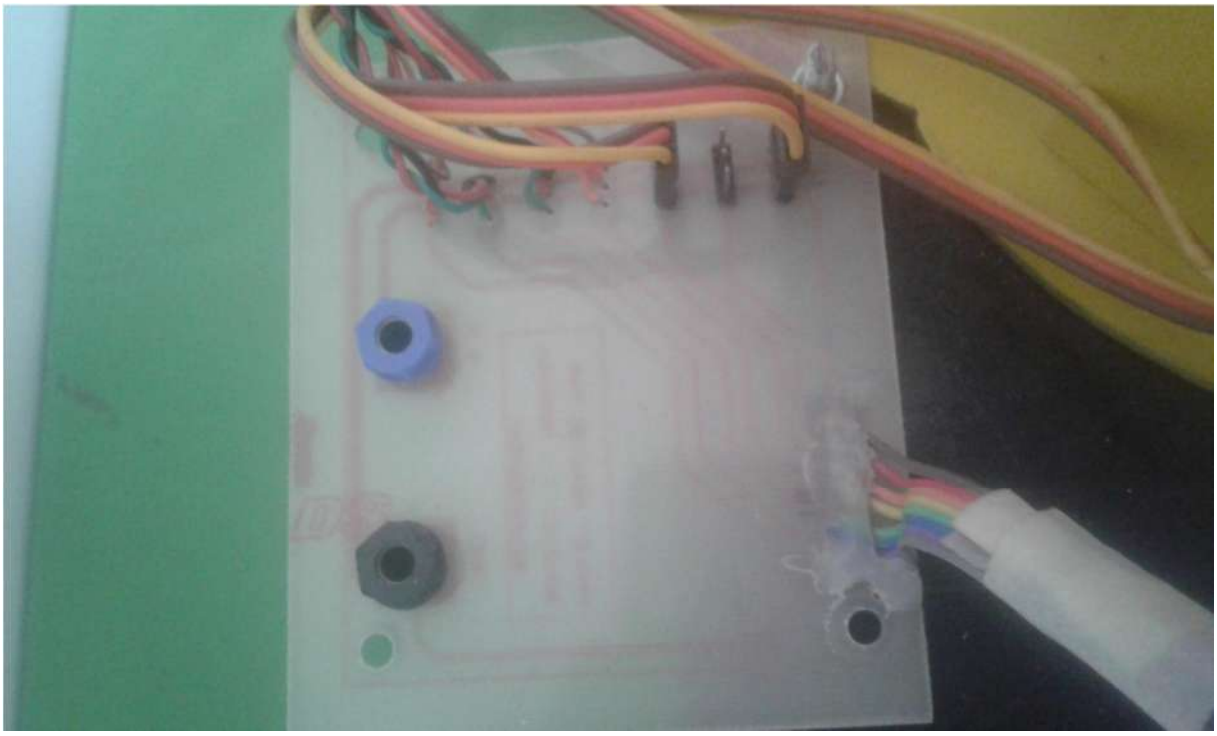


Figure III.17.Circuit de la carte commande d'un bras manipulateur.

III.7. Conclusion

Après des études et des aperçus, nous avons atteint la possibilité de choisir tous les appareils qui répondent à nos besoins et que nous compilons sur la structure et mettons une carte électronique, notamment en contrôlant le matériel et en fonctionnant avec succès sur la compilation des appareils où les programmes sont après l'introduction et l'entraînement .

Chapitre IV : Commande du bras manipulateur

IV.1.Introduction

Dans ce chapitre nous allons aborder la programmation de notre carte Arduino Uno, pour la commande du contrôle notre bras a l'aide d'une l'application mobile que nous présenterons comment concept et programmer sur le site de logiciel MITAPP INVENTOR .en plus de la façon d'utiliser le contrôle d'moniteur série que nous avons fait comme contrôle secours après l'application.

Nous allons expliquer et montrer toutes les étapes de travail sur la programmation que nous avons prévu dans le programme Arduino, afin de réussir l'objectif de ce chapitre qui est de faire fonctionner le bras manipulateur.

IV.2.logiciel Arduino

IV.2.1.Présentation

➢ L'interface de l'IDE Arduino est plutôt simple, il offre une interface minimale et épurée pour développer un programme sur les cartes Arduino. Il est doté d'un éditeur de code avec coloration syntaxique et d'une barre d'outils rapide. Ce sont les deux éléments les plus importants de l'interface, c'est ceux que l'on utilise le plus souvent. On retrouve aussi une barre de menus, plus classique qui est utilisée pour accéder aux fonctions avancées de l'IDE. Enfin, une console pour afficher les résultats de la compilation du code source, des opérations sur la carte, etc.[14]

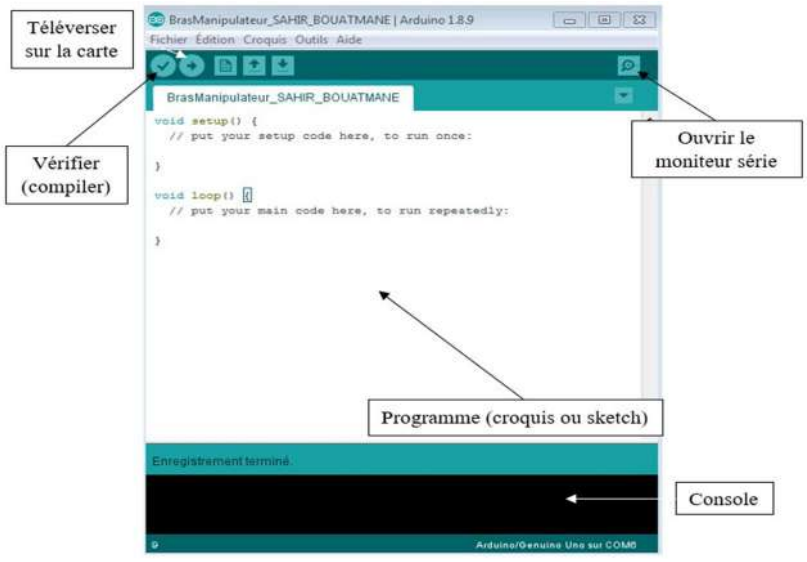


Figure.IV.1: Environnement de développement Arduino[12].

➢ Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des similarités avec le **C**, le **C++**, le **Java** et le **Processing**. Le langage impose une structure particulière typique de l'informatique embarquée.[14].

➢ La fonction **« Setup »** contiendra toutes les opérations nécessaires à la configuration De la carte (directions des entrées sorties, débits de communications série, etc.).

➢ La fonction « **Loop**», elle est exécutée en boucle après l'exécution de la fonction Setup. Elle continuera de boucler tant que la carte n'est pas mise hors tension, redémarrée (par le bouton reset). Cette boucle est absolument nécessaire sur les microcontrôleurs étant donné qu'ils n'ont pas de système d'exploitation. En effet, si l'on omettrait cette boucle, à la fin du code produit, il sera impossible de reprendre lamain sur la carte Arduino qui exécuterait alors du code aléatoire.

➢ Le moniteur série intègre de l'environnement Arduino assure une communication entre l'ordinateur et la carte Arduino. Le lancement du Moniteur série se fait simplement _a partir de la barre d'outils en sélectionnant le même débit en bauds que celui configuré dans le programme et permet de d'afficher des messages textes émis par la carte Arduino et d'envoyer des caractères vers la carte Arduino (en phase de fonctionnement) [3].

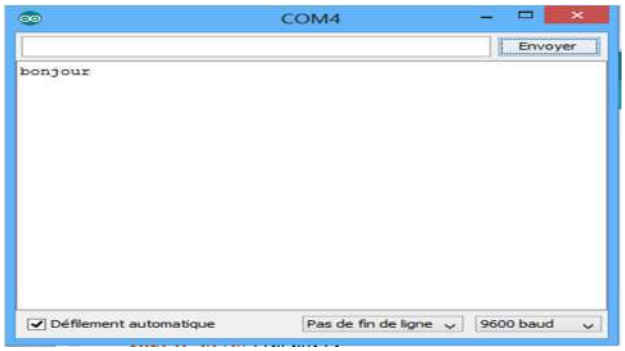


Figure IV.2.Fenêtre de moniteur série

IV.2.2.Explication du Structure de la programmation

Travailler sur la programmation est difficile et compliqué car les lois de programmation Arduino doivent être respectées. Et le travail qui était à notre programme, nous avons cherché à le mettre sans erreurs et à le développer selon nos idées. Dans cette partie, nous allons expliquer comment nous travaillons sur les étapes et les équations, et la figure. IV.3. montre la structure des étapes du programme.



Figure IV.3.Plan de programmation planifié

❖ étape1: partie déclarative

Dans cette partie nous devons :

- Inclure la bibliothèque **SoftwareSerial** pour la communication série du module. Bluetooth ainsi que la bibliothèque **Servo**, qui est incluses dans l'IDE Arduino.
- Déclaration des noms pour l'utilisation des servomoteurs.
- Déclaration les valeurs initiales du chaque servomoteur.
- Déclaration des variables globales utilisées pour le positionnement des servomoteurs
- Définir le module Bluetooth HC-06 de type **SoftwarSerial** en lui attribuant les broches **RX** et **TX.**
- Définir les tableaux de type **int** pour stocker la position actuelle, précédente des **servos.**
- Et enfin les variables pour la vitesse de déplacement **v,** et de donnée reçue via Bluetooth **Data** de type **String.**

```
#include <SoftwareSerial.h>
#include <Servo.h>
Servo base;    // Base
Servo epaule;    // Epaule
Servo coude;       // Coude
Servo pincette;       // pincette
SoftwareSerial Bluetooth(3, 4);
int valBase = 115;
int valepaule = 20;
int valcoude = 145;
int valpincette = 25;
int valactuelle_base=valBase;
int valactuelle_epaule=valepaule;
int valactuelle_coude=valcoude;
int recCounter = 0;
int runCounter = 0;
int num_servo;
int pick[3]={115,20,145};
int dep[3]={115,20,145};
String donnees = "";
char Data="";
char buf[2];
bool  stateRun=true;
int hold = 500;
```

Figure.IV.4.Partie d'déclaration des variables.

❖ étape 2: partie de la fonction d'initialisation setup ( ) :

➢ Déclaration les attachements, à l'aide de la fonction **attach () pour les sarvo**

➢ Déplacer le bras du robot dans sa position initiale, en utilisant la fonction **write ()** qui déplace simplement le **servo** dans n'importe quelle position de 0 à 180 degrés

```
void setup() {
  base.attach(9);
  epaule.attach(10);
  coude.attach(11);
  pincette.attach(6);
  pincette.write(valpincette);
  coude.write(valcoude);
  epaule.write(valepaule);
  base.write(valBase);
  delay(100);
```

Figure.IV.5.Fonction d'attache et position.

➢ Sans oublier l'initialisation du module Bluetooth et moniteur série voici ce que l'on obtient:

➢ l'écriture dans le moniteur série, des mots qui facilitent la compréhension du contrôle des bras, grâce à la fonction **serial.println ()**

```
  Serial.begin(9600);
  Bluetooth.begin(9600);
  Bluetooth.setTimeout(10);
  Serial.println("          ..................................................");
  Serial.println("          |            Titre du projet :                   |");
  Serial.println("          |  etude de rpogrammation du Bras Robot          |");
  Serial.println("          ..................................................");
  delay(20);
}
```

Figure.IV.6.Fonction d'affichage à moniteur série.

❖ Étape 3: partie de la boucle principale, la fonction **loop** ( ) :

- Vérifier en permanence, s'il existe des données entrantes en provenance du moniteur série et Smartphone avec la fonction **available ().**
- Utiliser la fonction **readString ()**, pour lire les données sous forme de chaîne et les stocker dans la variable **Data.**

```
void loop() {
    donnees=" ";
 if (Bluetooth.available()> 0 || Serial.available() > 0)   {  //
     if (Bluetooth.available() > 0)   donnees = Bluetooth.readString();    //Data = Bluetooth.read();
     if (Serial.available() > 0) donnees = Serial.readString();
     Serial.println("-----------------------------------------");
     Serial.print("donnees Recue : ");Serial.println(donnees);
```

Figure .IV.7.Partie configuration.

Explication suivent pour les trois sarvos (base, épaule, coude) :

- À l'aide de la fonction **startsWith ()**, nous vérifions le préfixe de chaque donnée entrante afin de savoir ce quel moteur contrôler.
- Nous extrayons le degré envoyé a partir de la variable Data à l'aide de la fonction **substring(),** et nous la convertissons en entier avec la fonction **toInt().**
- On a utilisé la boucle if et **elseif** pour le control de sarvo.
- Pour finir nous exécutons la fonction **rotation()** décrite ci-dessous « **rotation(**int**num**,int**deg)** », prend en paramètre deux donnée, le numéro du servo a contrôler, et la position souhaitée.
- D'abord, nous stockons **deg** dans notre tableau de position **donnees.**

- Ensuite nous vérifions si la position actuel **ps[]** et plus grande ou plus petite que la valeur **k.**
- En conséquence nous incrémenterons, ou décrémenterons un indice dans une boucle **for**, et avec la fonction **write()** de l'objet **Servo,** pour Contrôler les moteurs**,** suivie de la fonction **delay(v)** avec **v** une valeur En **ms**, pour contrôler sa vitesse.
- Enfin nous définissons la position actuelle comme position Précédente.

```
// ............. Servo base.............................
  if (donnees.startsWith("S1")) {
     valBase = donnees.substring(2,donnees.length()).toInt();
     Serial.print("valbase: ");Serial.println(valBase);
             if(valBase >= valactuelle_base){
                  for(int k=valactuelle_base;k <= valBase;k++){
                     base.write(k);
                     delay(20);
                  }
              }
              else if(valBase < valactuelle_base){
                   for(int k=valactuelle_base ; k >= valBase ; k--){
                     base.write(k);
                     delay(20);
                   }
              }
    valactuelle_base=valBase;
     donnees=" ";
  }
```

Figure.IV.8.Fonction rotation de base

```
// ............. Servo epaule.............................
if (donnees.startsWith("S2")) {
    valepaule = donnees.substring(2,donnees.length()).toInt();
   Serial.print("valepaule: ");Serial.println(valepaule);
          if(valepaule >= valactuelle_epaule){
                for(int k=valactuelle_epaule;k <= valepaule;k++){
                   epaule.write(k);
                   delay(20);
                }
           }
           else if(valepaule < valactuelle_epaule){
                for(int k=valactuelle_epaule ; k >= valepaule ; k--){
                   epaule.write(k);
                   delay(20);
                }
           }
  valactuelle_epaule=valepaule;
   donnees=" ";
}
```

Figure.IV.9.Fonction rotation d'épaule.

```
// ..... Servo coude.....
if (donnees.startsWith("S3")) {
   valcoude = donnees.substring(2,donnees.length()).toInt();
   Serial.print("valcoude: ");Serial.println(valcoude);
   if(valcoude >= valactuelle_coude){
              for(int k=valactuelle_coude;k <= valcoude;k++){
                 coude.write(k);
                 delay(20);
              }
           }
           else if(valepaule < valactuelle_coude){
                for(int k=valactuelle_coude ; k >= valcoude ; k--){
                 coude.write(k);
                 delay(20);
                }
           }
  valactuelle_coude=valcoude;
   donnees=" ";
}
```

Figure.IV.10.Fonction rotation de coude.

- Pour le **sarvo** pincette a utilisé boucle de fonctionnement simple, car il s'ouvre et se ferme toujours . nous exécutons la fonction rotation() décrite ci-dessous

```
if (donnees.startsWith("S4")) {
   valpincette = donnees.substring(2,donnees.length()).toInt();
   Serial.print("valpincette: ");Serial.println(valpincette);
   pincette.write(valpincette);
   donnees=" ";
}
```

Figure.IV.11.Fonction rotation de pincette.

Explication suivent pour la fonction **SAVE**:

- Ensuite viens la fonction **Save (),** associée au bouton et **SAVE** de l'application ou associée au moniteur série.
- En appuyant sur le bouton **SAVE** l'application envoie le préfix **'**SAVE**'**ou envoyé par moniteur série.
- Toujours avec la fonction **statsWith (),** nous vérifions le contenue du préfix, si ilcorrespond, nous exécutons la fonction **recCounter++,** et **testrecCounter ().**

```
// ..............donnees --> "SAVE".................
if (donnees.startsWith("SAVE")) {
    Serial.println("Mode SAVE ");
    recCounter++;
    testrecCounter();
 }
```

Figure.IV.12.Fonction SAVE.

Explication suivent pour la fonction **RUN**:

- Pour la fonction **RUN**, associé au bouton **RUN** de l'application ou associée au moniteur série.

  En appuyant sur le bouton **RUN**, l'application envoie le préfix **'**RUN**'** ou envoyé par moniteur série.
- Toujours avec la fonction **statsWith (),** nous vérifions le contenue du préfix, si il Correspond nous exécutons la fonction **Home0 (),** et **recCounter++,** et **testrecCounter.**

```
// ..............donnees --> "RUN".................
if (donnees.startsWith("RUN")) {

   if(pick[0] != 0 && pick[1] != 0 && pick[2] != 0 ){
       Serial.println("Mode RUN ");
       Home0();
       runCounter++;
       testRunCounter();
   }
}
```

Figure.IV.13.fonction RUN.

Explication suivent pour la fonction **RESET**:

- Pour la fonction **RESET** associé au bouton **RESET** de l'application ou associée au moniteur série.
- En appuyant sur le bouton **RESET**, l'application envoie le préfix **'**RESET**'**ou envoyé par le moniteur série.
- Toujours avec la fonction **statsWith (),** nous vérifions le contenue du préfix, si il correspond nous exécutons les fonctions **Home0 (),** et avec la suppression de toutes les donnée liée aux deux modes **pick []** et **dep []**.

```
// ..............donnees --> "RESET".................
if (donnees.startsWith("RESET")) {
   Serial.println("Mode Arret ");
   Home0();
    pick[0] = 0;
    pick[1] = 0;
    pick[2] = 0;
    dep[0] = 0;
    dep[1] = 0;
    dep[2] = 0;
}
}
```

Figure.IV.13.Fonction RESET.

- Explication suivent pour les fonctions que nous avons programmées et intégrée avec **voidloop (),** de ces fonctions nous trouvons **[voidHome (), voidHome0 (),voidHome2 (), voidtestrecCounter (), voidtestRunCounter ()].**

- Pour la fonction**voidHome0 ().** Ce sont les coordonnées de la position initiale de chaque **sarvo,** qui est le point par lequel il passe à chaque étape

```
void Home0() {
  epaule.write(20);
  delay(200);
  coude.write(145);
  delay(200);
  base.write(115);
  delay(200);
}
```

Figure.IV.14.fonction. Home0**.**

- Pour la fonction **voidHome ().**C'est une boucle spéciale pour le retour du bras de l'étape de **pick[]**en ramenant chaque **sarvo** à la position initial après avoir porté la chose.

```
void Home() {
  for ( int j =pick[1] ; j >=20 ; j--) {
          epaule.write(j);
          delay(10);
        }
  delay(hold);
  if(pick[2] > 145){
      for ( int j =pick[2] ; j >=145 ; j--) {
              coude.write(j);
              delay(25);
            }
  }else if(pick[2] < 145){
      for ( int j =pick[2] ; j <=145 ; j++) {
              coude.write(j);
              delay(25);
            }
  }
  delay(hold);
  for ( int j =pick[0]; j <= 115 ; j++) {
          base.write(j);
          delay(10);
  }
  delay(hold);
```

Figure.IV.15.Fonction. Home.

- Pour la fonction**voidHome2 ().**C'est une boucle spéciale pour le retour du bras de l'étape de **dep []**en ramenant chaque **sarvo** à la position initial après l'avoir mis la chose.

```
void Home2() {
  for ( int j = dep[1]; j >= 20; j--) {
          epaule.write(j);
          delay(10);
  }delay(hold);
  if(dep[2] > 145){
      for ( int j =dep[2] ; j >=145 ; j--) {
              coude.write(j);
              delay(25);
            }
  }else if(dep[2] < 145){
      for ( int j =dep[2]; j <=145 ; j++) {
              coude.write(j);
              delay(25);
            }
  }
  for ( int j = dep[0]; j >= 115; j--) {
          base.write(j);
          delay(10);
  }delay(hold);
}
```

Figure.IV.16.Fonction. Home2.

- Pour la fonction **voidtestRunCounter (),** si **donnees RunConter** est impaire on a activé le mode RUN avec confirmer qui communique avec Bluetooth et le **donnees** est n'pas **PAUSE,** si donnée **PAUSE** est sorti d'une boucle par **break.**
- **A**près avoir vérifié les condition En conséquence nous incrémenterons, ou décrémenterons un indice dans une boucle **for** pour les position **pick[].** et **dep[],** et avec la fonction **write**() de l'objet Servo, pour contrôler les moteurs, suivie de la fonction **delay** (v) avec v une valeur en ms, pour contrôler sa vitesse.

avec l'utilisation des fonction **[Home (), Home0 (), Home2 (),]** expliqué précédemment**.** Enfin nous supprimons la donnée existante.

```
//.........................test runCounter.....................
void testRunCounter(){
    if (runCounter % 2 == 1) {         //  && stateRun==true
      Serial.println("Automatic RUN");
      while (runCounter % 2 == 1) {
        if (Bluetooth.available() > 0 || Serial.available() > 0 ){
            if (Bluetooth.available() > 0)    donnees = Bluetooth.readString();
            if (Serial.available() > 0) donnees = Serial.readString();
        }
        if ( donnees.startsWith("PAUSE")) {
          runCounter++;
//          Serial.println(runCounter % 2);
          if (runCounter % 2 == 0) {
            break;
            }
        }
        //Home();
        delay(500);
       for ( int j =115; j >=  pick[0]; j--) {
          base.write(j);
          delay(10);
       }
        pincette.write(70);
        delay(800);
        for ( int j =20 ; j <= pick[1]; j++) {
          epaule.write(j);
```

```
            delay(25);
          delay(hold);
          pincette.write(25);
          delay(900);
          Home();
          delay(hold);
        for ( int j =115; j <=  dep[0]; j++) {
            base.write(j);
            delay(10);
          delay(hold);
        for ( int j =145; j <=  dep[2]; j++) {
            coude.write(j);
            delay(10);
          delay(hold);
        for ( int j =20; j <=  dep[1]; j++) {
            epaule.write(j);
            delay(10);
          delay(1000);
          pincette.write(70);
          delay(800);
          pincette.write(25);
          delay(800);
          Home2();
       Serial.println("Automation Pause");
          Home0();
          donnees=" ";
    }
```

Figure. IV.17.Fonction **testRunCounter**

- Pour la fonction **voidtestrecCounter** c'est pour afficher sur le moniteur série des positions avec des angles et on utilise tout ce qui a été expliqué précédemment

```
//.........................test bouton Record..............
void testrecCounter(){
    if (recCounter % 2 == 1) { // Here we write cords for pick place in array.
      pick[0] = valBase;
      pick[1] = valepaule;
      pick[2] = valcoude;
      Serial.print("pick = {"); Serial.print("   ");
      Serial.print(pick[0]);Serial.print("   ");
      Serial.print(pick[1]);Serial.print("   ");
      Serial.print(pick[2]); Serial.println("  }");
      stateRun=true;
    }
    if (recCounter % 2 == 0) { // Here we write cords for deposit place in array.
      dep[0] = valBase;
      dep[1] = valepaule;
      dep[2] = valcoude;
      Serial.print("dep = {");Serial.print("   ");
      Serial.print(dep[0]);Serial.print("   ");
      Serial.print(dep[1]);Serial.print("   ");
      Serial.print(dep[2]);Serial.println("  }");
      stateRun=true;
    }
```

Figure .IV.18.Fonction **testrecCounter.**

IV.3. Le concept d'App Inventor

IV.3.1.Présentation

La programmation s'accomplit en ligne à l'aide d'un langage de bloc d'instructions et sous un environnement logiciel au choix (Mac, Linux ou Windows). Les informations sont stockées sur des serveurs distants. La seule contrainte à laquelle nous sommes confrontés est de devoir posséder un compte Gmail pour pouvoir y accéder. [13]

Figure.IV.19. LOGO App Inventor

IV.3.2.1. Réalisation de l'application Android

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

IV.3.2. Partie design

Dans la partie design, on dispose de divers blocs graphiques comme des boutons, des champs de texte, des images, des curseurs, mais aussi des lay-out qui servent à délimiter les zones de notre application. Il existe également quelques autres fonctionnalités utiles de multimédia, de GPS, et de communication Bluetooth.

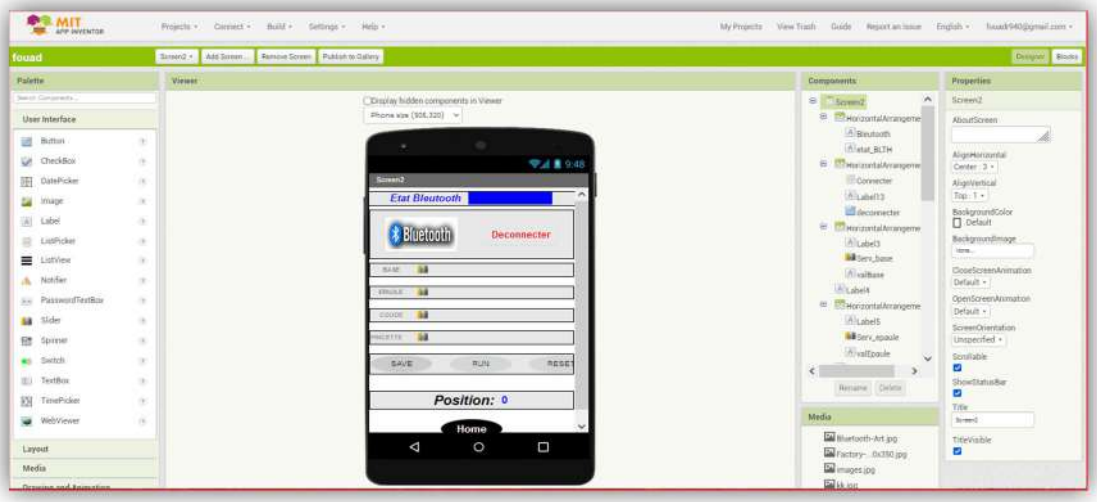


Figure.IV.20.Partie design de notre application sous AppInventor

IV.3.2.1partie bloc

Dans la partie bloc, on retrouve toutes les fonctions mathématiques, logiques, ou évènementielles, 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.

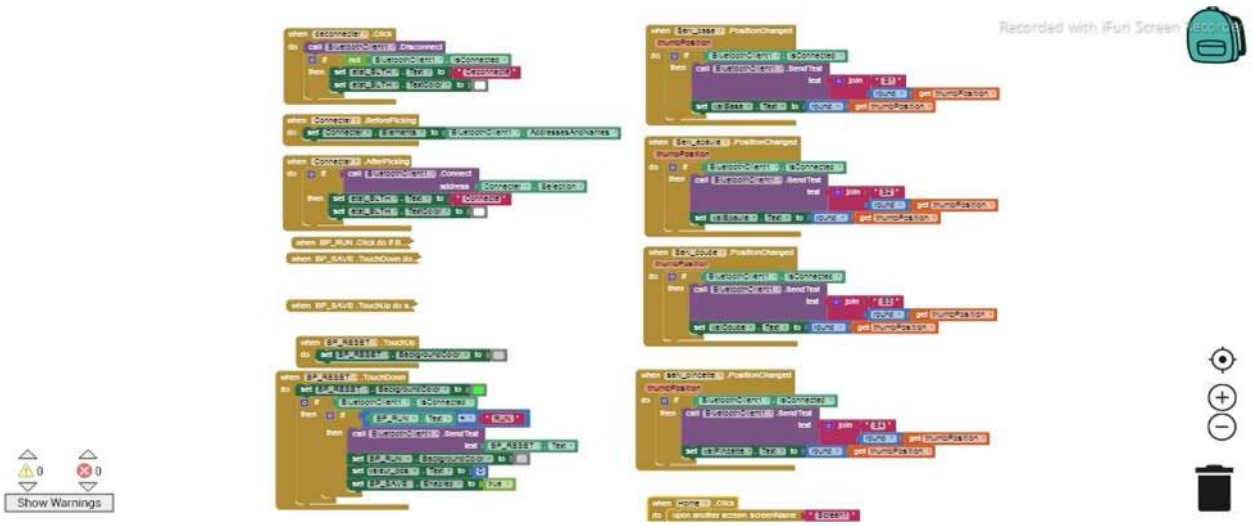


Figure.IV.21**.** Partie blocks de AppInventor.

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.

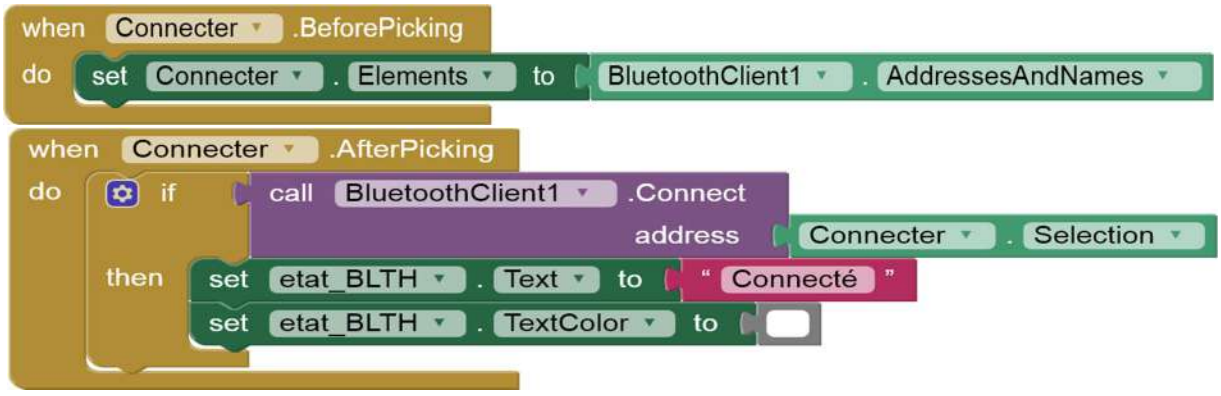


Figure.IV.22**.** Définition des éléments de l'objet Bluetooth.

- Ici nous indiquons seulement l'état de connexion de notre appareil en changeant le contenue et la couleur du texte et Quand nous appuyons sur le bouton « déconnecte », le Bluetooth se déconnecte.

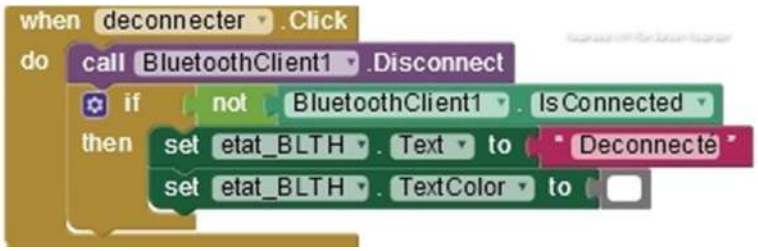


Figure.IV.23. Block Bluetooth

- S'il y a une erreur de connexion avec l'appareil, l'état est mis à déconnecte.

❖ Ensuite, on définit les blocs de contrôle des curseurs pour les variations d'angles des servomoteurs :

- Ce sont les blocs qui contrôlent les quatre servomoteurs du bras manipulateur (pince, coude, épaule et base). 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, dans ce cas « S1 » qui est l'indice du moteur **serv_base,** ainsi que la valeur actuelle du curseur qui est « thumbPosition ».

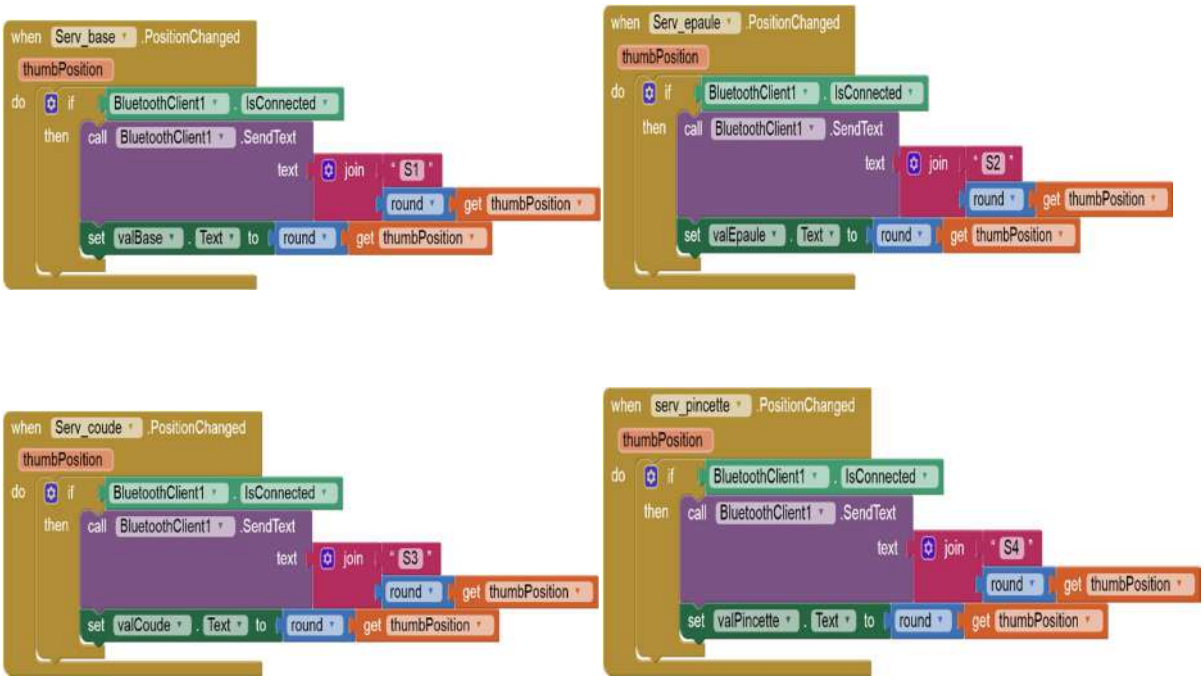


Figure.IV.24..Blocs de contrôle des curseurs.

❖ Pour finir les blocks de fonctionnalité des boutons SAVE, RUN, et RESET

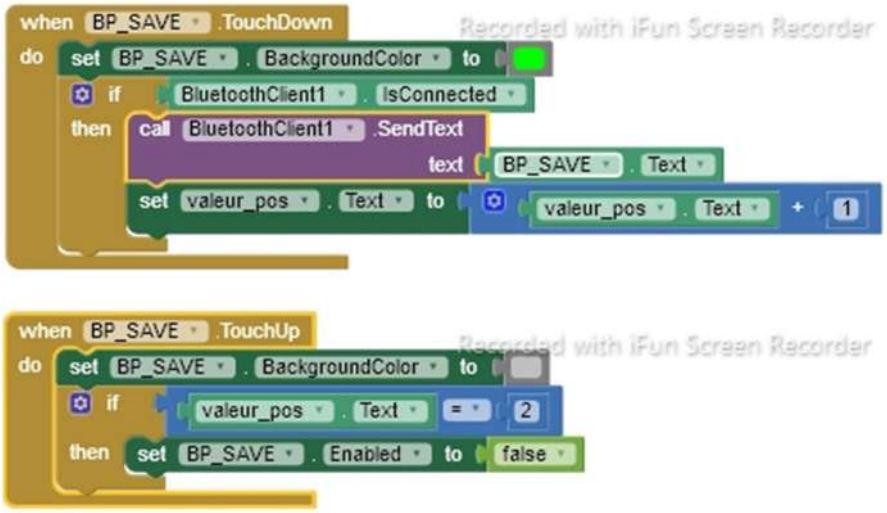


Figure.IV.25.Block bouton SAVE.

- Quand le bouton « save » est appuyé divisé en deux opérations (TouchDown,Touch Up) , nous changeons la couleur du bouton et nous envoyons le préfixe « SAVE » à l'Arduino, et nous incrémentons le nombre de positions sauvegardées .

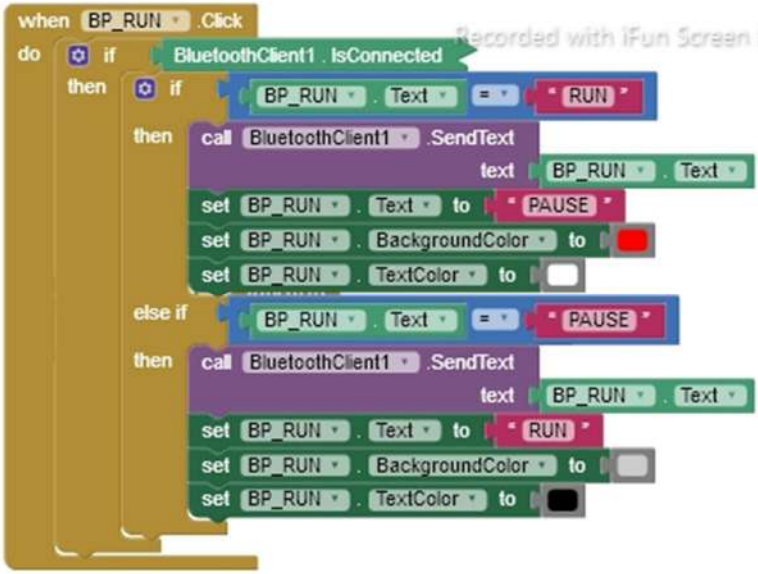


Figure.IV.26.Block bouton RUN

- Quand nous cliquons sur le bouton «RUN», nous changeons la couleur du bouton et nous envoyons le préfixe « PAUSE » ou « RUN » a l'Arduino, et si il est initialement sur «PAUSE», le texte est modifié en « RUN », sinon, si il est initialement en « RUN », le texte est modifié en «PAUSE».

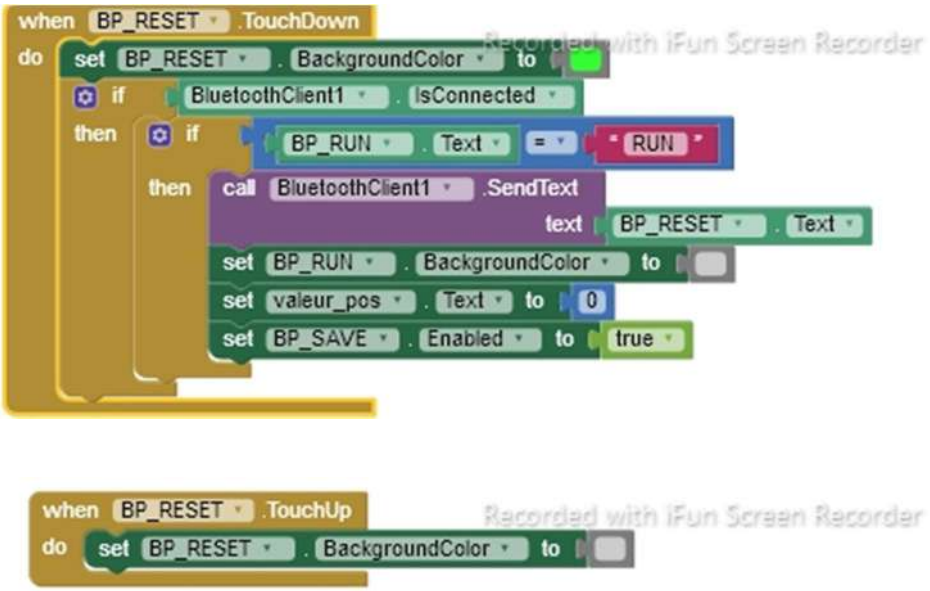


Figure.IV.27.Block bouton RESET

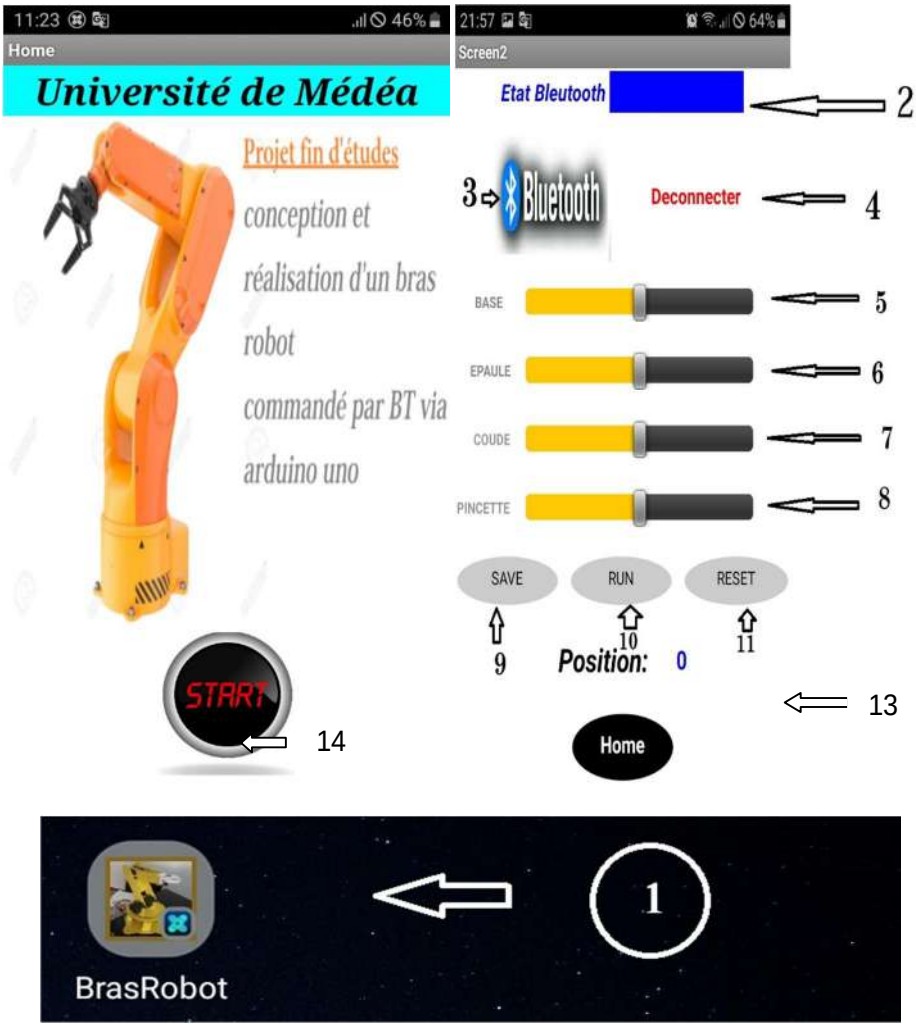
- Ici en appuyons sur le bouton « RESET » divisé en deux opérations (TouchDown,Touch Up) , nous envoyons le préfix « RESET » à l'Arduino, le texte et la couleur du bouton « run » est réinitialisé aux valeurs initiales, et le nombres de positions sauvegardées est remis à 0



Figure.IV.28.Block bouton Home.

Quand le bouton « Home »est appuyé, il retourne à«screen1».

❖ Après le design et la programmation par bloc, on obtient l'application finale :



1: Icône de l'application.

2: Etat du Bluetooth (connecté ou déconnecté).

3: Connexion au Bluetooth.

4: Déconnecter au Bluetooth.

5 : Angle de base.

6 : Angle de l'épaule.

7: Angle du coude.

8: Angle de la pince.

9: Angle de la pincette.

10:Bouton RUN.

11:Botton RESET.

12:Bouton Home.

13:Bouton Start.

14:Bouton Home.

Figure.IV.29.Interface finale de l'application.

IV.4. Conclusion

En passant par la réalisation, impression et conception mécanique et après le montage de tout le composant de notre bras, la création d'une application Android pour le contrôle de ce dernier, et développé un programme dans logiciel arduino spécial et conformément à notre application.

Après des études expériences que nous avons travaillé sur les programmes que nous avons obtenu notre objectif et couronne le bras selon donnée.

Conclusion générale

Ce mémoire présente notre travail, qui implique la recherche et la réalisation d'un bras robotique à quatre degrés de liberté, en supposant que l'environnement de travail est sans obstacle. Après avoir exploré de manière approfondie le domaine de la robotique, nous considérons le cœur du problème en étudiant les différents organes qui composent notre bras robotique. Par la suite, nous nous sommes concentrés sur la conception et le développement de modèles géométriques, cinématiques et dynamiques, qui ont respectivement pour fonction de générer des trajectoires de référence et d'analyser la fixité, ce qui nous permet de mettre en œuvre et d'assembler plus facilement la structure du bras. Ensuite, nous avons fait une recherche et une étude sur la façon de choisir et d'acquérir les pièces des appareils électroniques pour le fonctionnement et le contrôle qui complètent la production du bras. Enfin, nous avons mené une recherche et une mise en œuvre réelles de la technologie de programmation à travers les commandes et la technologie de commande fournies dans notre production (via des applications mobiles et des moniteurs série).

Pour la réalisation pratique, le cadre est conçu par travail manuel et découpe laser du châssis. La programmation est faite à base du langage C++ et a fait l'objet d'une implémentation sur la carte Arduino UNO, Le contrôle du bras manipulateur a été divisé en deux à l'aide d'une application mobile réalisée (sans fil) et grâce au module Bluetooth, ou contrôler le moniteur série l'Arduino logiciel.

Le manuscrit a été organisé en quatre chapitres. Le premier présente des généralités. La seconde porte sur Conception et réalisation du bras manipulateur étudié. Le troisième chapitre expose Etude technique des composants du bras manipulateur, en quatrième chapitre nous avons travaillé sur Commande du bras manipulateur,

La recherche et la mise en œuvre de ce projet nous ont permis d'approfondir les connaissances théoriques et pratiques acquises au cours de la formation, notamment en robotique et systèmes embarqués. Nous pouvons comprendre différents points clés, tels que le travail d'équipe, la conception et la mise en œuvre de projets, et comment gérer et résoudre les problèmes en fonction de chaque étape du travail.

Conclusion générale

Bien que cet humble travail ait atteint ses objectifs, je pense qu'il peut encore être amélioré. En fait, le travail décrit dans cet article peut être effectué de diverses manières. Premièrement, il serait intéressant de considérer un environnement de travail handicapé, mais ce n'est pas le cas dans cette étude. Il est également intéressant d'implémenter des commandes en boucle fermée et est plus utile lorsqu'il y a des interférences. C'est une belle extension de ce projet. Pour la réalisation du bras, il serait avantageux d'optimiser la construction en utilisant des matériaux plus résistants. Des servomoteurs avec un couple moteur plus élevé peuvent être utilisés pour soulever des objets plus lourds.

Bibliographie

[1]Achour Halim, Mohammed Sghir Mounir. *conception et réalisation d'un bras manipulateur a 3 dégrée de liberté.* blida: université saad dahlab, 2017.

[2]Moussaoui Amira. *Conception et réalisation d'un bras manipulateur commandé.* Boumerdes: Université M'hamed Bougara, 2017.

[3]Abed sadek , Mehiaoui Mohammed. *Intreface graphique pour piloter un bras de robot à travers une carte arduino.* Tlemcen: Université Abou belkaid , 2015.

[4]https://www.cafedelabourse.com/secteur-robotique-thematique-gagnante-bourse

[5] Haouri mohammed ridha,djennane mohammed amine. *Modélisation et commande non linéaire de position applique à un robot de manipulation* . Saïda : Université dr.tahar moulay, 2019.

[6]Hamdache Zine Eddine, Touil Farid. *Réalisation et commande d'un bras robotique à quatre degrés de liberté.* BOUIRA: UNIVERSITE AKLI MOHAND OULHADJE, 2019.

[7]Chaale Merouane . *Modélisation cinématique d'un robot manipulateur à chaine continue ouverte.* OUARGLA: UNIVERSITE KASDI MERBAH, 2013.

[8]Ait Dahmane Kahina, Ait Ziane Meziane. *Conception et Réalisation d'un Bras Manipulateur Commandé par API.* AIN DAFELA: Université de KHEMIS MILIANA, 2015.

[9]OUZIANE Fatima Zohra. *Rétro-conception du bras horizontal de robot manipulateur de lacellule flexible.* TLEMCEN: UNIVERSITE ABOU BEKR BELKAID-, 2013.

[10]MANSOURI Rafik, KHELID Ali. *Conception et réalisation d'un bras manipulateur à trois degrés de liberté commandé par la carte Arduino UNO.* TIZI-OUZOU: UNIVERSITE MOULOUD MAMMERI , 2015.

[11]YACINE TAZEROUT. *COMMANDE D'UN BRAS ROBOTISE AVEC UNE APPLICATION ANDROID.* Nancy, France: Université de Lorraine , 2019.

[12] Jules Sery ,Jordan Vocisano. *Le Bras Robotisé.* Marseille, France: Lycée Jean Perrin, 2012.

[13]Sahir koceila, Bouatmane hakim. *Etude et réalisation d'un bras de robot à base de carte arduino.* Bejaia: Université A.Mira, 2019.

[14]IAMRACHE MASSINISSA, BENTOUMI NAWALE. *Etude et réalisation d'un bras robotique.* Tizi-Ouzou: Université Mouloud Mammeri , 2017.