



Département de Maintenance en Instrumentation

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industriel

Spécialité : Ingénierie de la Maintenance en Instrumentation

Thème

**Réalisation et commande d'un bras manipulateur embarqué
Sur un robot mobile via une application Android**

Présenté et soutenu publiquement par :

KHELIFA Oussama et BEN AISSA Abdelkader

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
ADJELOUA ABDELAZIZ	MAA	IMSI-Univ. D'Oran2	Président
MEKKI Ibrahim El Khalil	MCA	IMSI-Univ. D'Oran2	Encadreur
ROUAN SERIK MEHDI	MCB	IMSI-Univ. D'Oran2	Examinateur

Remerciements

En préambule à ce mémoire nous remercions **ALLAH** qui nous a aidé et nous donné la patience et le courage durant ces longues années d'études.

Nous tenons aussi à remercier nos très **chers parents** qui se sacrifient chaque jour pour nous afin de nous voir réussir dans notre vie.

Nous voudrions exprimer notre profonde gratitude à notre encadreur, Monsieur

Mekki Ibrahim El khalil.

Nous vous remercions de nous avoir encadrés, orienté, aidé et conseillé.

On ne manquera pas de remercier les examinateurs :

ADJELOUA ABDELAZIZ

ROUAN SERIK MEHDI

Pour avoir accepté d'examiner notre travail et pour leur participation

Au jury.

Enfin, nous adressons notre gratitude à tous mes **proches et amis**

Pour leur soutien moral et matériel.

KHELIFA Oussama et BEN AISSA Abdelkader

Dédicace

Je dédie du profond de mon cœur ce modeste travail à :

A mes **chers parents** aucune dédicace ne saurait exprimée l'amour et le respect que j'ai toujours eu pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de vos sacrifices que vous avez consentis pour mon éducation et ma formation le long de ces années.

Je dédie ce travail aussi à ma sœur **Asmaa**
Et mes frères **Mohammed** et **Islam** pour leur encouragement durant mon parcours universitaire.

A mes **grand parents** que Dieu les protège pour nous.

A tous les membres de ma famille : **KHELIFA** et **Boushaba** petits et grands.

A mon amie **Hamichi Islam** et **Ramzi** en souvenir de notre sincère et profonde amitié.

KHELIFA Oussama

Dédicace

Je dédie du profond de mon cœur ce modeste travail à :

A mes **chers parents** aucune dédicace ne saurait exprimée l'amour et le respect que j'ai toujours eu pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de vos sacrifices que vous avez consentis pour mon éducation et ma formation le long de ces années.

Je dédie ce travail aussi à ma sœur **Manel**
Et mes frères **Anis** et **Khayr Eddine** pour leur encouragement durant mon parcours universitaire.

A mes **grand parents** que Dieu les protège pour nous.

A tous les membres de ma famille : **Ben Aissa** et **Ben nacer** petits et grands.

A mon amie **Hadjane Djamel** et **Ben Aissa Sadik** en souvenir de notre sincère et profonde amitié.

BEN AISSA Abdelkader

Résumé:

Un robot est une machine contrôlée équipée d'un ordinateur qui l'aide à effectuer des tâches préalablement programmées. Avec les nouveaux développements, le robot entre dans le monde des services à domicile, commerciaux et médicaux. Après être entré dans le monde de l'industrie lourde.

Alors nous avons choisi ce projet, dans lequel nous sommes occupés de l'étude d'un bras manipulateur embarqué à un robot mobile commandé par une application Android.

L'utilisation de ce bras est basée sur la technologie Bluetooth. Cette dernière est une technologie sûre et fiable et nous permet de communiquer à une distance allant jusqu'à 12 mètres. Il peut également répéter les tâches demandées en continu après les avoir enregistrées.

ملخص:

الروبوت الة متحكممة مزودة بحاسوب يساعدها على القيام بأعمال تمت برمجتها من قبل. ومعا لتطوراتا لجديدة يدخل الروبوت عالم الخدمات المنزلية والتجارية والطبية. بعدما دخل عالم الصناعات الثقيلة . ومن اجل هذا مشروع تخرجنا تناولنا فيه دراسة ذراع مناور مدمج في روبوت متحرك يأمر من طرف تطبيق (android). لكثرة فوائده واهمية عمله الدقيقة وتوفيره الكثير من الوقت والجهد وعلاوة على ذلك قادر على الوصول الى اماكن الضيقة والخطيرة ويستند استخدام هذه الذراع على تقانيه بلوتوث هذا الاخير هو تقنية امنة وموثوقة ويتيح لنا التواصل على مسافة تصل الى 12 مترا. كما يمكنه أيضا تكرار المهام المطلوبة بشكل مستمر بعد حفظها.

Abstract:

A robot is a controlled machine equipped with a computer that helps it perform previously programmed tasks. With the new developments, the robot enters the world of home, commercial and medical services. After entering the world of heavy industry.

For this project, we graduated, in which we dealt with the study of a manipulator arm integrated into a mobile robot commanded by an application android.

The use of this arm is based on Bluetooth technology. The latter is a safe and reliable technology and allows us to communicate at a distance of up to 12 meters. It can also repeat requested continuously after saving them.

Sommaire

Introduction générale :	1
CHAPITRE I : Généralité sur les robots	2
I.1.Introduction:	3
I.2.Définitions de la robotique et du robot	3
I.2.1.Définition de la robotique	3
I.2.2.Définition d'un robot	3
I.3.L'histoire de la robotique	3
I .4.Types de robots	5
I .4.1.Robots mobiles	5
I .4.2.Robots manipulateurs	5
I .5.Domainses d'application des robots	6
I .5.1.Robots industriels	6
I .5.2.Robots médicaux	6
I .5.3.Robots militaires	6
I .5.4.Robots stationnaires	7
I .5.4.1. Robots articulés	7
I .5.4.2.Robots cylindriques	7
I .5.4.3. Robots sphériques	7
I .5.4.4. Robots à roues	8
I .5.4.5.Robots logiciels	8
a.Intelligence artificielle :	8
b.Télégramme Bot	9
I .5.5.Les Robots volants	9
I .5.6. Robots modulaires	10
I.6. conception d'un robot	11
I.6.1. L'unité informationnelle	11
I.6.2.L'unité opérationnelle	11
I.7.Topologie des robots manipulateurs	11
I.7.1.Une base	12
I.7.2.Porteur	12
I.7.2.1.Un segment	12
I.7.2.2.L'articulation	12
I .7.3.Actionner	13
I .7.4.Organe terminal	13
I .8.Classification des robots	13

I .8.1.Classification fonctionnelle.....	13
I .8.1.1.Manipulateur à commande manuelle.....	13
I .8.1.2.La manipulatrice automatique.....	14
I .8.1.3.Robots programmables.....	14
I .8.1.4.Les robots intelligents.....	14
I.8.2.La classification géométrique.....	15
I.8.2.1.La structure cartésienne (ppp).....	15
I.8.2.2.La structure cylindrique (rpp) ou (prp).....	16
I.8.2.3.La structure 3r (Anthropomorphe).....	16
I.8.2.4.La structure parallèle.....	16
I.9.Les critères de choix d'un robot.....	17
I.10.Conclusion.....	18
CHAPITRE II : Description d'un bras manipulateur embarqué sur robot mobile.....	19
II .1.Introduction.....	20
II .2.Description mécanique du bras manipulateur.....	20
II .2.1.La structure mécanique du bras.....	20
II .2.1.1.La base.....	21
II .2.1.2.Le bras.....	21
II .2.1.3.La pince.....	22
II .2.1.4.Le robot.....	22
II .2.2.Description des parties constitutives du bras manipulateur.....	23
II .2.2.1.Partie électronique.....	23
II .2.2.2.Les actionneurs du bras manipulateur.....	23
II .2.2.3.Les paramètres techniques des servomoteurs.....	24
II .2.2.4.La commande des servomoteurs.....	24
II .2.2.5.Transmission entre articulations : les engrenages.....	25
II .2.2.6.Les capteurs.....	25
II .2.2.7.Les paramètres techniques du récepteur Bluetooth.....	26
II .2.3.Description du kit de robot.....	26
II .3.La Réalisation de robot mobile.....	27
II .4.Conclusion.....	32
CHAPITRE III: Interface de l'Arduino et programmation.....	33
III .1.Introduction.....	34
III .2.Partie programme :.....	34
III .3.L'environnement de développement Arduino :.....	34
III .4. Interface Arduino:.....	35
III .5.Structure générale du programme (IDE ARDUINO) :.....	35

III.6.Injection du programme :	36
III.6.1.Description du programme :	36
III.6.2.Les étapes de téléchargement du programme:	37
III.7. Les accessoires de la carte ARDUINO :	38
III.8.Communication Bluetooth:	38
III.9.Le code Arduino:	42
III.10.Conclusion	46
CHAPITRE IV: Réalisation d'une application Android	49
IV.1.Introduction	50
IV.2 Présentation de l'Android.	51
IV.3.Présentation d'App INVENTOR	51
IV.4. La Création un projet :	52
IV.4.1.Le concept d'App INVENTOR	53
IV.5.L'environnement de développement et les outils d'application	54
IV.5.1Accès à l'application et création d'un nouveau projet	54
IV.5.2.Description de l'espace Designer	54
IV.5.3.Description de l'environnement de programmation	55
IV.5.4.Description de l'émulateur	56
IV.6. Application Android 'Ouss bras manipulateur'	56
IV.6.1.Réalisation de l'application Android	56
IV6.2.L'interface finale de notre application	61
IV.7.Communication de l'application avec l'Arduino	62
IV.8.Téléchargement de l'application	62
IV.9.Conclusion	63

La liste des figures

Figure I.1 : Unimate : le premier robot industriel.....	4
Figure I.2 : Opportunit.....	5
Figure I.3 : Robonaut.....	5
Figure I.4 : Robot Mobile.....	5
Figure I.5 : Bras Manipulateur.....	6
Figure I.6 : Robot Industriel.....	6
Figure I.7 : Robot Medicale.....	7
Figure I.8 : Robot Militaire.....	7
Figure I.9 : Bras Robotique.....	7
Figure I.10 : Robot Sphérique.....	8
Figure I.11 : Robot a plusieurs Roues.....	8
Figure I.12 : L'intelligence Artificielle.....	9
Figure I.13 : Telegramme Bot.....	9
Figure I.14 : Les Robots Volants.....	10
Figure I.15 : Robot Modulaires.....	10
Figure I.16 : Les Caractéristiques du robot mobile.....	11
Figure I.17 : Topologies d'un robot manipulateur.....	11
Figure I.18 : Représente les actions d'une articulation rotoïde.....	12
Figure I.19 : Manipulations de bobines de papier ou de film plastique.....	13
Figure I.20 : Un bras manipulateur sans l'intervention humaine.....	14
Figure I.21 : Robot mobile programmable.....	14
Figure I.22 : Robot avec système d'artificielle intelligence.....	15
Figure I.23 : La structure cartésienne (ppp).....	15
Figure I.24 : La structure cylindrique (rpp) ou (prp).....	16
Figure I.25 : La structure 3r (Anthropomorphe).....	16
Figure I.26 : Structure Parallèle Delta.....	17
Figure II.1 : Les pièces constituant le robot.....	20
Figure II.2 : Base du robot conçu sur 3D Builder.....	21
Figure II.3 : Bras du robot conçu sur 3D Builder.....	21
Figure II.4 : Pince du robot conçu sur 3D Builder.....	22
Figure II.5 : Bras robot entier conçu sur 3D Builder.....	22
Figure II.6 : Arduino UNO R3.....	23
Figure II.7 : Schéma d'un actionneur.....	23
Figure II.8 : Servomoteurs.....	24
Figure II.9 : Exemple de signaux en position pour un servomoteur.....	25
Figure II.10 : Schéma d'un engrenage.....	25
Figure II.11 : Module Bluetooth.....	26
Figure II.12 : Le matériel utilisé pour la réalisation du robot.....	27
Figure II.13 : moteur à courant continu.....	27
Figure II.14 : fixation des moteurs au châssis.....	28
Figure II.15 : Installation la plaque des moteurs sur le châssis.....	28
Figure II.16 : Moteur Shield L293D.....	29
Figure II.17 : Câblage de l'alimentation.....	29
Figure II.18 : Installation du module Bluetooth.....	30
Figure II.19 : La fixation du servomoteur de la base.....	30
Figure II.20 : Fixation du servomoteur du côté droite et gauche.....	30
Figure II.21 : Fixation du servomoteur de la pince.....	31
Figure II.22 : La forme finale du bras manipulateur.....	31
Figure II.23 : Bras manipulateur embarqué sur un robot mobile.....	32
Figure III.1 : Environnement de développement arduino.....	35
Figure III.2 : Interface IDE Arduino.....	36

Figure III.3 : Paramètre de la carte.....	36
Figure III.4 : Description du programmesur Arduino.....	37
Figure III.5 : Les étapes de téléchargement du code.....	38
Figure III.6 : Schéma de la communication avec le module Bluetooth HC-05.....	39
Figure III.7 : Logigramme résumant la commande à angles sauvegarde.....	43
Figure IV.1 : Android.....	51
Figure IV.2 : Logo mit app inventor.....	51
Figure IV.3 : Crée un nouveau projet.....	52
Figure IV.4 : App INVENTOR LOGO.....	53
Figure IV.5 : Fenêtre d'accès au logiciel.....	54
Figure IV.6 : Les zones du logiciel.....	55
Figure IV.7 : Interface de programmation.....	55
Figure IV.8 : Simulation de l'application sur smartphone.....	56
Figure IV.9 : Partie1 design de notre application sous App INVENTOR.....	57
Figure IV.10 : Partie2 design de notre application sous App INVENTOR.....	57
Figure IV.11 : Définition des éléments de l'objet Bluetooth.....	58
Figure IV.12 : Blocs de contrôle des curseurs.....	58
Figure IV.13 : Configuration du block pour les boutons de robot mobile.....	59
Figure IV.14 : Blocs des trois boutons.....	60
Figure IV.15 : Interface finale de l'application.....	61
Figure IV.16 : Schéma synoptique simplifié du système.....	62
Figure IV.17 : téléchargement de l'application sur smartphone	62

La liste des tableaux

Table II.1. Tableau des caractéristiques des servomoteurs du bras	24
Table II .2. Tableau des caractéristiques du récepteur Bluetooth du bras	26

Abréviations

RX : Stands for Receiver « Réception ».

TX : Stands for Transmitter «Transmission ».

GND : Ground « masse ».

USB : Universal Serial Bus « bus série universel »

E/S: Entrées /Sorties.

EEPROM : Electrically Erasable Programmable Read-Only Memory

JC: Jesus Christ

JSC: Johnson Space Center

MCC : Moteur à Courant Continu

PPP : Trois articulations prismatiques consécutives

PWM : Pulse With Modulation

ISO : International Organizations for Standardization

MHz : MegaHertz

SMS : short message service

Mbps : Megabits par seconde

Kbit/s : kilobits par seconde

MIT : Massachusetts Institute of Technology

C++ : langage de programmation

Apk : Androïde Package Kit

GPS : Global Positionning System.

Mo : Mégaoctet

mm : millimètre.

mA : milli-Ampère

PC : Personal Computer.

PWM : Pulse Width Modulation « La modulation de largeur d'impulsions (MLI) ».

Introduction générale :

Notre projet de fin d'études porte sur la réalisation d'un bras manipulateur embarqué sur un robot mobile commandé via une application Android. Nous avons choisi ce sujet à cause de l'utilisation la plus courante de ces bras d'un côté, et d'autre côté l'importance de son travail y compris leur force qui dépasse celle du corps humain. De plus, il facilite les tâches avec une haute précision. Il est capable d'accéder dans des endroits dangereux. L'utilisation de ce bras se base sur une technologie Bluetooth.

Cette dernière est une technologie sécurisée, fiable et nous permet de communiquer à des distances allant jusqu'à 10m.

Afin d'atteindre notre objectif, ce mémoire est organisé en quatre chapitres comme suit :

Le premier chapitre présente une définition de robot et robotique, l'historique des bras manipulateurs, les types des bras, l'architecture et la classification des bras et leurs domaines d'application.

Le deuxième chapitre en lui-même est divisé en trois parties :

La première concerne une présentation de la carte Arduino Uno, les servomoteurs, le Motor Shield L293D, Bluetooth HC-05.

En revanche, la deuxième c'est la partie électronique ; elle contient le côté « *hardware* » (les composants) et la réalisation détaillée des côtes du bras, et elle montre comment faire le branchement exact des servomoteurs avec la carte Arduino.

Le troisième chapitre, interface de l'arduino et programmation

Le quatrième chapitre, nous avons fait la réalisation d'une application Android

CHAPITRE I :

Généralité sur les robots

I.1.Introduction:

Le système robotique a apporté d'énormes changements dans divers aspects socioéconomiques de la société humaine au cours des dernières décennies. D'après Per Guoqiangetal., les robots manipulateurs industriels ont été largement déployés et utilisés dans toutes sortes d'industries pour effectuer des tâches répétitives, fastidieuses, critiques et ou dangereuses, telles que l'assemblage de produits, la peinture automobile, l'emballage de boîtes et le soudage. Ces robots préprogrammés ont toujours été très performants dans plusieurs applications industrielles structurées en raison de leur grande précision, endurance et rapidité. Les technologies robotiques ont été intégrées aux technologies de réseau existantes pour étendre la gamme de valeurs fonctionnelles de ces robots lorsqu'ils sont déployés dans des environnements non structurés, tout en favorisant l'émergence de la robotique en réseau au cours des années 90.

I.2.Définitions de la robotique et du robot

I.2.1.Définition de la robotique

Comme son nom l'indique, la robotique est une science qui s'intéresse aux robots. En fait, il s'agit d'une intégralité des domaines techniques (informatique, électronique, automatique, mécanique) exposée autour d'un objectif commun. Cet objectif est l'automatisation flexible d'abondants secteurs de l'activité humaine présumée jusqu'à présent comme indispensable de la présence de l'homme, le robot est un dispositif universel, dont l'homme désire depuis toujours le supplanter dans les tâches complexes [1].

I.2.2.Définition d'un robot

Un robot est un système mécanique poly-articulé mû par des actionneurs et commandé par un ordinateur qui est voué à effectuer une large pluralité de tâches. Il est autant défini comme étant une machine apte d'exercer des tâches et de manier des objets selon un programme de manière automatique. Les robots sont habituellement utilisés pour succéder les humains dans des situations où ces derniers sont impuissants de pratiquer le travail, des circonstances plus délicates, d'une importante exactitude ou répétitive [2].

I.3.L'histoire de la robotique

Depuis la nuit des temps, l'homme a cherché à se faire remplacer pour des tâches spécifiques pouvant être dangereuses, fastidieuses ou longues. Le concept du robot a été établi par de nombreuses réalisations historiques créatives, tels que : l'horloge à l'eau de Clepsydre introduite par les Babyloniens (1400 avant JC), le théâtre de l'automate du Héron d'Alexandrie (une centaine d'années après JC) et les automates de l'arabe Al-Jazari (1200 après JC) [3].

Un de ces spécimens très évolués fut présenté par Jacques de Vaucanson en 1738 : il représentait un homme jouant d'un instrument de musique à vent. Jacques de Vaucanson créa également un automate représentant un canard mangeant et refoulant sa nourriture après ingestion de cette dernière.

Cependant, l'apparition du robot physique a dû attendre l'arrivée de technologies avancées et

plus adaptées au cours du XXe siècle.

- 1920 : L'origine du mot robot provient de la langue tchèque dans laquelle son ancêtre "robot" signifie travail forcé. Il a été introduit, en 1920, par l'écrivain tchèque Karel Capek dans la pièce de théâtre « Rossum's Universal Robots » [4].
- En 1954, George Charles Devol invente le premier robot industriel programmable. Il fonde en 1956 la première entreprise de robot, Unimation, avec l'ingénieur Joseph Engelberger. Leur robot, appelé Unimate, était capable de manipuler des objets avec des actionneurs hydrauliques.

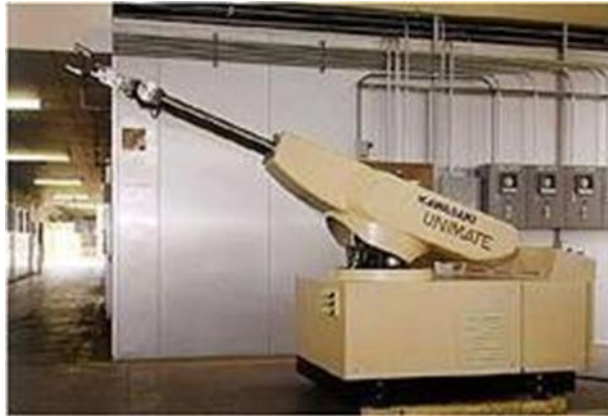


Figure I.1 : Unimate : le premier robot industriel [5]

- En 1969, Victor Scheinman de l'Université de Stanford inventa un bras articulé sur 6 axes. Scheinman vendit son concept à Unimation plus tard [5].
- 1973: KUKA, une entreprise allemande crée le premier robot industriel avec 6 axes contrôlés électromécaniquement : le Famulus [5].
- 1978 : Unimation propose le PUMA, un robot d'assemblage encore utilisé de nos jours dans plusieurs laboratoires de recherche [5].
- 1981 : Takeo Kanade construit le premier robot muni de moteurs installés directement aux articulations du bras. Cela a permis aux robots d'aller rapides et plus précis qu'auparavant [5].
- 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) [5].
- 2003 : Première mission d'exploration de la planète Mars impliquant deux robots mobiles, Spirit et Opportunity [5].



Figure I.2: Opportunity [5]

- 2011 : Robonaut (R2B) premier robot humanoïde envoyé dans l'espace construit et conçu

par la NASA au Johnson Space Center (JSC) à Houston (Texas), en collaboration avec General Motors (GM) et Oceaneering [5].



Figure I.3 : Robonaut [5]

I.4.Types de robots

Il existe deux types de robots : les robots mobiles et les robots manipulateurs

I.4.1.Robots mobiles

En général, on appelle robots mobiles l'ensemble des robots à base mobile. L'usage veut cependant que l'on montre le plus souvent par ce terme les robots mobiles à roues. Les autres robots mobiles sont en effet le plus souvent désignés par leur type de locomotion, qu'ils soient marcheurs, sous-marins ou aériens.



Figure I.4: Robot Mobile [6]

I.4.2.Robots manipulateurs

Les robots manipulateurs sont des robots intégrés dans un espace propre destiné au transfert de produits semi-finis entre les lignes de production. Le robot est généralement suspendu pour des questions d'encombrement et par contraintes d'implantation.

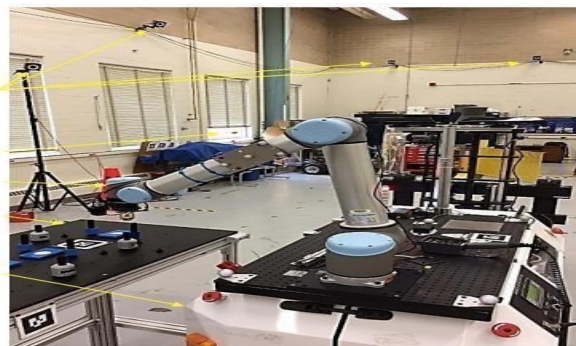


Figure I.5 : Bras Manipulateur [7]

I.5. Domaines d'application des robots

I.5.1. Robots industriels

Un robot industriel est défini par l'ISO comme un manipulateur polyvalent, reprogrammable, à commande automatique et programmable sur trois axes ou plus. Le domaine de la robotique peut être plus spécifiquement défini comme l'étude, la conception et l'utilisation de systèmes robotiques pour la production (une définition de haut niveau basée sur la définition précédente du robot).

Les applications typiques des robots comprennent la soudure, la peinture, l'assemblage, le "pick and place" (comme l'emballage, la palettisation et le SMT), l'inspection et le test des produits ; toutes ces opérations sont réalisées avec une grande robustesse, rapidité et précision. [8]



Figure I.6 : Robot Industriel [8]

I.5.2. Robots médicaux

Les robots médicaux sont des robots qui permettent aux chirurgiens d'accéder plus facilement aux zones opérées en utilisant des méthodes plus précises et moins invasives. On les trouve dans la plupart des télémanipulateurs, qui utilisent les actions du chirurgien d'une part pour contrôler l'effecteur" d'autre part [8].



Figure I.7 : Robot Médicale [8]

I.5.3. Robots militaires

Les robots militaires sont des robots autonomes ou des dispositifs télécommandés conçus pour des applications militaires. Plusieurs armées étudient actuellement de tels systèmes. L'armée a toujours été à la pointe de la technologie, il n'est donc pas surprenant que les robots les plus avancés au monde soient construits en vue d'applications militaires. Si l'idée de machines autonomes portant des armes lourdes peut rendre les gens un peu nerveux, elles ont le potentiel de réduire considérablement les pertes de vies humaines en permettant aux

soldats de localiser des endroits sûrs ou de pénétrer dans des sites ennemis. Beaucoup sont même conçus à des fins de soutien, plutôt que pour éliminer les menaces. Alors que de nombreux projets militaires dans le monde sont naturellement entourés de secret, certains émergent, notamment des projets de robotique [15].



Figure I.8 : Robot Militaire [8]

I.5.4. Robot stationnaires

I.5.4.1. Robots articulés

Un robot articulé est un robot à articulations rotatives (par exemple, un robot à pattes ou un Robot industriel). Les robots articulés peuvent varier de simples structures à deux articulations à des systèmes à 10 articulations ou plus en interaction. Ils sont alimentés par divers moyens, dont des moteurs électriques [8].



Figure I.9 : Bras Robotique [8]

I.5.4.2. Robots cylindriques

Les robots cylindriques sont des robots dont les axes forment un système de coordonnées cylindriques. [15] Il est utilisé pour :

- a. Opérations d'assemblage.
- b. Manipulation sur les machines-outils.
- c. Soudage par points.
- d. Fonctionnement des machines de coulée sous pression.

I.5.4.3. Robots sphériques

Un robot sphérique est un robot doté de deux articulations rotatives et d'une articulation prismatique, c'est-à-dire de deux axes rotatifs et d'un axe linéaire. Les robots sphériques ont un bras qui forme un système de coordonnées sphériques [8].



Figure I.10 : Robot Sphérique [8]

I.5.4.4. Robot à roues

La conception devient beaucoup plus complexe, surtout lorsqu'on utilise des roues plus puissantes, car chaque roue doit tourner à la même vitesse lorsque le robot doit avancer. Les différences de vitesse entre les roues gauche et droite des robots à direction différentielle font que le robot se déplace latéralement au lieu d'aller tout droit. La différence de vitesse entre les roues d'un même côté fait que la roue la plus lente tourne. [8]



Figure I.11 : Robot a plusieurs Roues [8]

I.5.4.5. Robot logiciel

I.5.4.5.1. Intelligence artificielle :

C'est l'intelligence dont font preuve les machines. En informatique, le domaine de la recherche en IA est défini comme l'étude des "agents intelligents" : tout dispositif qui perçoit son environnement et prend des mesures qui maximisent ses chances de succès dans un objectif donné. Dans le langage courant, le terme "intelligence artificielle" s'applique lorsqu'une

Machine imite les fonctions "cognitives" que les êtres humains associent à d'autres esprits humains, telles que "l'apprentissage" et "la résolution de problèmes". À mesure que les machines deviennent de plus en plus puissantes, les structures mentales, dont on pensait qu'elles nécessitaient de l'intelligence, sont retirées de la définition [8].



Figure I.12 :l'intelligence Artificielle [8]

I.5.4.5.2. TélégrammeBot

Il s'agit d'un service de messagerie instantanée gratuit basé sur le Cloud. Il existe des clients télégraphiques pour les systèmes mobiles (Android, iOS, Windows Phone, Ubuntu Touch) et de bureau (Windows, MacOS, Linux). Les utilisateurs peuvent envoyer des messages et échanger des photos, des vidéos, des autocollants, des fichiers audio et toutes sortes de fichiers. Le télégramme offre également la possibilité d'envoyer des messages cryptés d'un bout à l'autre. Le télégramme est soutenu par l'entrepreneur russe Pavel Durov. Son code côté client est un logiciel libre mais contient des blocs binaires, et le code source des versions récentes n'est pas toujours publié immédiatement, tandis que son code côté serveur est fermé et propriétaire [8].



Figure I.13 : Télégramme Bot [8]

I.5.5. Les Robots volants

Il n'est pas facile de regrouper les robots volants en sous-groupes. Comme les robots volants utilisent différents types de technologie, nous regrouperons ces robots en fonction de leur système de vol.

Pour l'instant, nous avons défini les types suivants (d'autres types seront ajoutés prochainement).

1. Robot-ballon : ces types de robots volants utilisent une montgolfière pour flotter dans le ciel.
2. Robot à voile tournante : Robots volant comme un hélicoptère (Quad rocoptères).
3. Robots à battements d'ailes : Ces robots volent comme un oiseau avec des ailes.
4. Robots d'avion : Des robots qui volent comme un avion [8].



Figure I.14 : Les Robots Volants [8]

I.5.6. Robots modulaires

Les systèmes robotiques modulaires à reconfiguration automatique ou les robots modulaires à reconfiguration automatique sont des machines cinématiques autonomes à morphologie variable. Au-delà de la conduite, de la détection et du contrôle classiques, typiques des robots à morphologie fixe, les robots auto configurant sont également capables de modifier délibérément leur forme en réorganisant la connectivité de leurs parties pour s'adapter à de nouvelles circonstances, accomplir de nouvelles tâches ou se remettre de dommages [9].



Figure I.15 : Robot Modulaires [9]

Un robot peut être décrit comme un système électromécanique ou une machine qui peut être commandée au moyen d'un programme informatique ou d'un équipement électronique (microprocesseur, microcontrôleur...); il existe de nombreuses définitions du robot, mais il présente certaines caractéristiques importantes telles que:

-Détection: à l'aide de divers capteurs tels que les capteurs de lumière, les capteurs tactiles, le sonar, il doit détecter le milieu environnant.

-Locomotion: Il est nécessaire de se déplacer dans l'environnement de travail ou dans le milieu de travail à l'aide de roues ou de jambes.

-Énergie: pour réaliser la tâche souhaitée, il est nécessaire de consommer de l'énergie. Il peut donc être alimenté par des piles, de l'électricité ou de l'énergie solaire, selon le cas.

-Intelligence: comme nous, elle permet aux robots de recueillir des informations sur leur environnement et leur permet de commencer à prendre des décisions pour eux-mêmes [9].

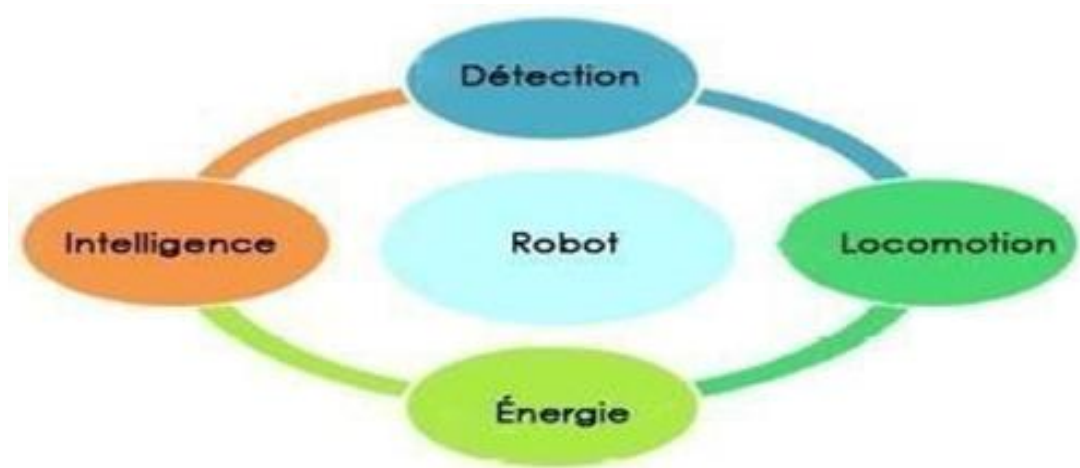


Figure I.16 : Les Caractéristiques du robot mobile [9]

I.6. Éléments conception d'un robot

On distingue deux ensembles interactifs dans un robot en fonctionnement: l'unité informationnelle et l'unité opérationnelle.

I.6.1. L'unité informationnelle

Consiste à recevoir les instructions représentant la tâche à accomplir, les mesures relatives à l'état interne de la structure mécanique qui constitue le bras manipulateur et les observations concernant son environnement. Elle élabore en conséquence les commandes de ses différentes articulations en vue de l'exécution de ses tâches.

Les systèmes actuels fonctionnent en interaction constante selon le cercle d'information-décision- action [10].

I.6.2.L'unité opérationnelle

Cette unité symbolise les actions commandées en imitant la puissance nécessaire à la source d'énergie.

En effet, ça constitue le robot physique, qui intègre la structure mécanique (segments, articulations, architecture,...), les modules d'énergie (amplificateurs, variateurs, servovalves....), les convertisseurs d'énergie (moteurs, vérins...), les chaînes cinématiques de transmission mécanique (réducteurs, vis à billes, courroies crantées...), les capteurs proprioceptifs placés sur chaque axe pour mesurer en permanence leur position et leur vitesse, et enfin l'effecteur, ou organe terminal, qui interagit avec le monde extérieur [9].

I.7.Topologie des robots manipulateurs

Le robot manipulateur est constitué principalement de la base, des segments, des articulations, des actionneurs et de l'organe terminal.

L'architecture générale d'un robot manipulateur est illustrée dans la figure I .17

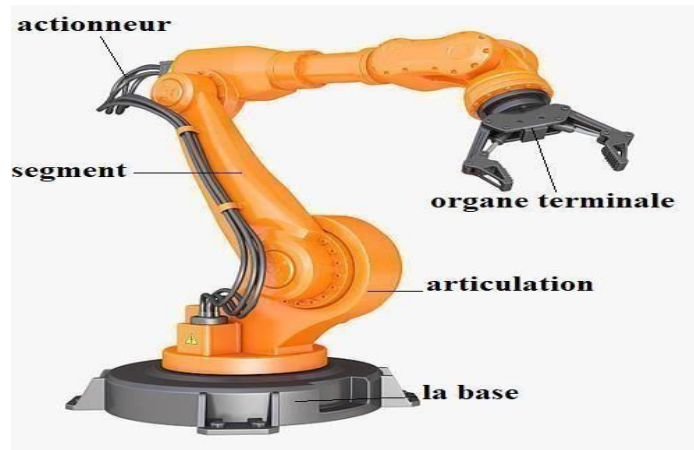


Figure I.17 : Topologies d'un robot manipulateur

I.7.1. Une base

La base du manipulateur est immobile sur la zone du travail. Ceci est en effet cas de la majorité des robots industriels. [10]

I.7.2. Porteur

Un porteur représente la partie vitale du système mécanique articulé, il a pour rôle d'amener l'organe terminal dans une situation donnée imposée par la tâche (la situation d'un corps peut être définie comme la position et l'orientation d'un repère attaché à ce corps par rapport à un repère de référence) [10]. Il est constitué de :

I.7.2.1. Un segment

C'est un Corps solide rigide apte d'être en mouvement par rapport à la base du porteur, et les uns par rapport aux autres [11].

I.7.2.2. L'articulation

Une articulation relie deux corps successifs en appliquant un nombre de degrés de liberté restreint, de l'un par rapport à l'autre. Soit m le nombre de degrés de liberté résultant, encore appelé mobilité de l'articulation. La mobilité d'une articulation est telle que $0 \leq m \leq 6$, lorsque $m=1$, ce qui est régulièrement le cas en robotique, l'articulation est dite simple : rotonde (R) ou prismatique (P) [11].

➤ Articulation rotoïde (R),

C'est une question d'une articulation de type pivot, notée R, réduisant le mouvement entre deux corps à une rotation autour d'un axe commun. La situation relative entre les deux corps est précisée par l'angle autour de cet axe.

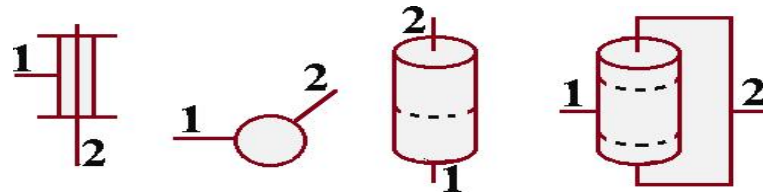


Figure I.18 : Représente les actions d'une articulation rotoïde

I.7.3.Actionner

Les actionneurs emploient fréquemment des moteurs électriques à aimant permanent, à courant continu, à commande par l'induit. On trouve de plus en plus de moteurs à commutation électronique (sans balais), ou, pour de petits robots, des moteurs pas-à-pas.

Pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique), les actionneurs sont le plus souvent hydrauliques, agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique) [11].

I.7.4.Organe terminal (effecteur)

Ce rassemblement tout dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, à dépression...), ou à les transformer (outils, torche de soudage, pistolet de peinture...). En d'autres termes, il s'agit d'une interface permettant au robot d'interagir avec son environnement [11].

I.8.Classification des robots

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

I.8.1.Classification fonctionnelle

I.8.1.1.Manipulateur à commande manuelle

Un manipulateur à commande manuelle est réalisé pour les manipulations particulièrement difficiles, répond spécifiquement aux exigences liées à la manutention de charges élevées, dans tous les points du volume de travail sans effort et dans des conditions optimales d'ergonomie et de sécurité [12].



Figure I.19 : Manipulations de bobines de papier ou de film plastique

I.8.1.2. La manipulatrice automatique

Une commande peut se faire par automate programmable on peut distinguer entre manipulateurs à cycle fixe et manipulateur à cycle programmable



Figure I.20: Un bras manipulateur sans l'intervention humaine [13]

I.8.1.3. Robots programmables

Ces robots refont les mouvements qu'on leur a appris par la programmation sans informations sur l'environnement ou la tâche effectuée. On peut aussi faire la distinction entre robots «Playback» qui reproduit la tâche apprise et robots à commande numérique qui peuvent être programmés hors-ligne [13].



Figure I.21 : Robot mobile programmable

I.8.1.4. Les robots intelligents

Il y a 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...). 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 [13].

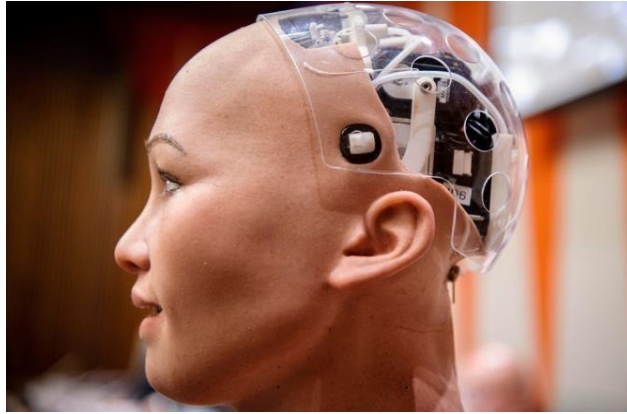


Figure I.22 : Robot avec système d'artificielle intelligence

I.8.2.La classification géométrique

Il y a possibilité aussi de classer les robots suivant leur configuration géométrique, autrement dit l'architecture de leur porteur. Ci-dessous, nous soulignerons les caractéristiques les plus importantes pour différentes figures de structure.

I.8.2.1.La structure cartésienne (ppp)

- 3 axes \perp 2 à 2, série, PPP, 3 DDL.
- Très bonne précision.
- Lent. [9]



Figure I.23 : La structure cartésienne (ppp)

I.8.2.2. La structure cylindrique (rpp) ou (prp)

- 3 axes, série, RPP/ PRP, 3 DDL.
- Espace de travail cylindrique.
- Très rapide. [9]



Figure I.24 : La structure cylindrique (rpp) ou (prp)

I.8.2.3. La structure 3r (Anthropomorphe)

- Reproduit la structure d'un bras humain.
- 3 axes, série, 3R, 3DDL. [9]



Figure I.25 : La structure 3r (Anthropomorphe)

I.8.2.4. La structure parallèle [14]

- Plusieurs architectures parallèles sont possibles (structure Delta, structure plane, structure hexapode)
- Plusieurs chaînes cinématiques en parallèle.

- Espace de travail réduit.
- Précis (grande rigidité de la structure).
- Rapide.

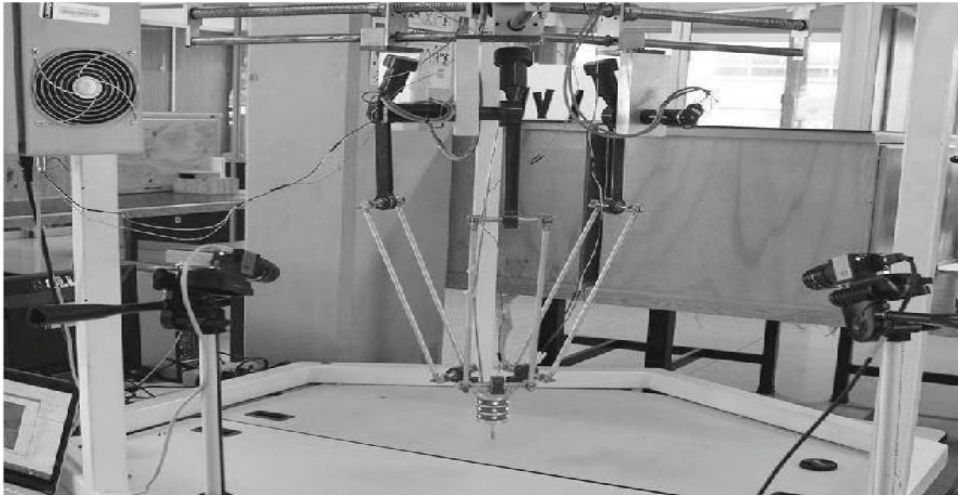


Figure I.26: Structure Parallèle Delta.

I.9. Le critère de choix d'un robot

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

- ❖ Une charge maximum transportable (de quelques kilos à quelques tonnes) a déterminé dans les conditions les plus défavorables (en élongation maximum).
- ❖ Une topologie mécanique articulée, le choix est guidé par la tâche à réaliser.
- ❖ Un volume de travail est accessible par l'outil du robot qui peut balayer l'outil, le volume dépend de la géométrie du robot, de la longueur des segments et du débattement des articulations.
- ❖ Un positionnement absolu, représentant l'erreur entre un point souhaité (réel)
- ❖ Défini par une position et une orientation dans l'espace cartésien—et le point atteint et calculé via le modèle géométrique inverse du robot. Cette erreur est due au modèle utilisé, à la quantification de la mesure de position, à la flexibilité du système mécanique. En général, l'erreur de positionnement absolu, également appelée précision, est de l'ordre de 1 mm.
- ❖ La répétitivité d'un robot est l'erreur maximale de positionnement répété de l'outil en tout point de son espace de travail. En général, la répétitivité < 0.1 mm
- ❖ La maintenance
- ❖ La masse du robot.
- ❖ Le coût du robot.
- ❖ La vitesse de déplacement (vitesse maximum en élongation maximum) et d'accélération.

I.10.Conclusion

Dans ce chapitre nous avons donné une vision générale sur le ressort de la robotique en passant par l'explication de la robotique et le robot ainsi que leur historique, les types des robots, les éléments constitutifs d'un robot, l'architecture des robots manipulateurs, la classification des robots telles que la classification fonctionnelle et géométrique, nous avons cité également les domaines d'applications les plus répondues.

La robotique devient donc plus en plus importante et les avancées technologiques et l'intelligence artificielle permettent toujours des avancements que ce soient dans les fonctionnalités et l'autonomie ou encore dans les architectures et la composition.

CHAPITRE II :

Description d'un bras manipulateur embarqué sur un robot mobile

II.1.Introduction

L'une des premières étapes dans la réalisation d'un robot consiste à définir le mode de locomotion à mettre en œuvre ainsi que les différents éléments qui le composent. Dans ce second chapitre, on présentera l'architecture mécanique générale du bras manipulateur et le robot mobile on abordera les différents actionneurs et capteurs le constituant. On donnera l'embarquement de robot.

II.2.Description mécanique du bras manipulateur

II.2.1.La structure mécanique du bras

Pour la réalisation de notre robot, plusieurs choix s'offrent à nous. Notre premier choix s'est porté sur la technique à utiliser pour la fabrication du châssis. On a le choix entre deux techniques : l'impression en 3D ou la découpe laser. L'impression en 3D est rare, car les imprimantes 3D sont peu disponibles et ses bobines de recharge sont coûteuses. C'est donc pourquoi on a choisi la découpe laser.

Pour le matériau, on a opté pour du plexiglass, car celui-ci est connu pour son prix abordable et sa grande disponibilité.

On a utilisé un modèle de bras en format .dxf pour découpe laser en téléchargement sur internet (<https://microbotlabs.com>) et on a procédé à la découpe des pièces (**Figure II.1**).

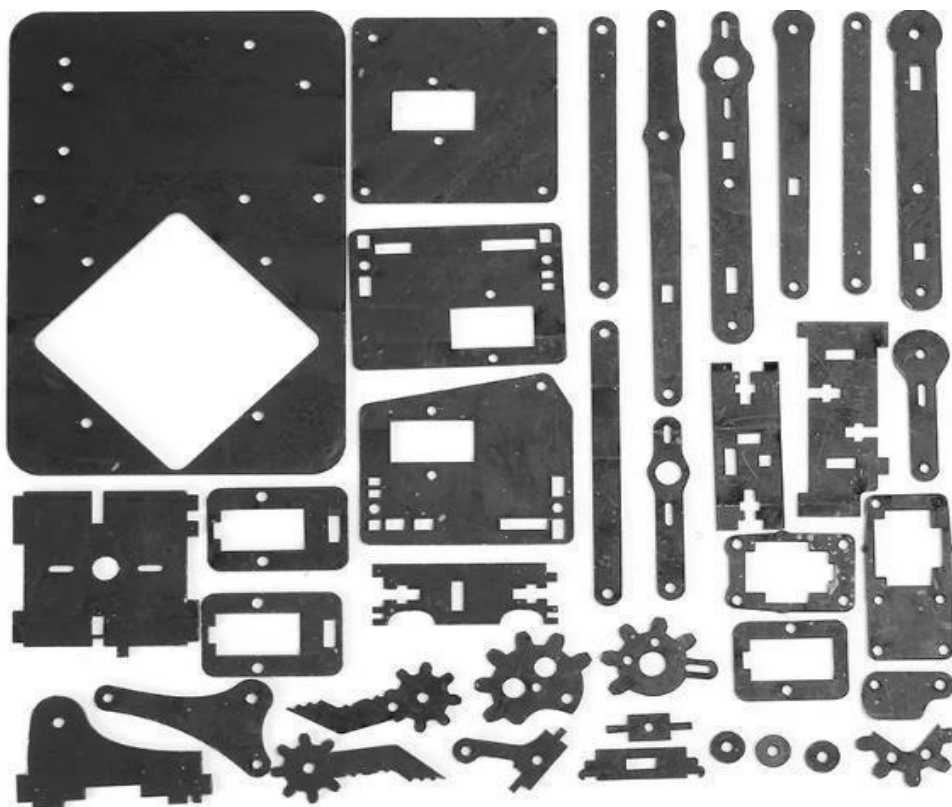


Figure II.1 : Les pièces constituant le robot

II.2.1.1. La base

La base de notre robot est fixe. La première articulation possède un servomoteur qui permet au bras de se déplacer sur un angle de 180° , autour de la verticale.

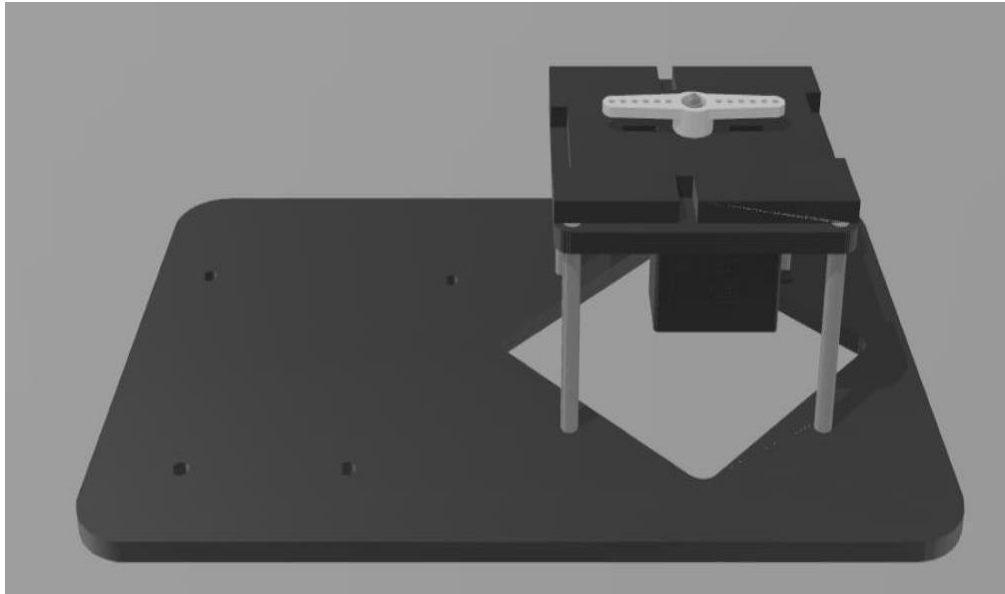


Figure II.2 : Base du robot conçue sur 3D Builder

II.2.1.2. Le bras

Il relie la base à la pince. Il se constitue d'une multitude de pièces en plexiglass reliées par des vis et de deux servomoteurs. Ils se trouvent sur les côtés comme l'illustre la Figure II .3. Ceux-ci contrôlent les mouvements avant/arrière et haut/bas du bras.



Figure II.3 : Bras du robot conçu sur 3D Builder

II.2.1.3. La pince

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

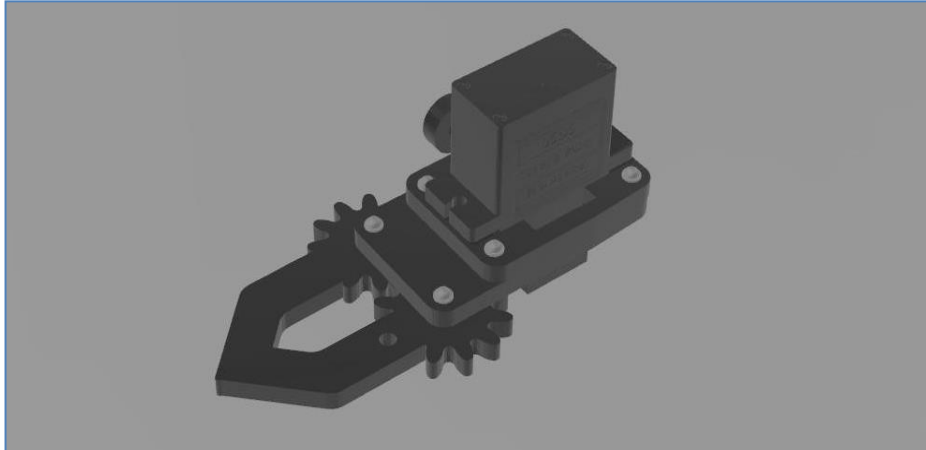


Figure II.4 : Pince du robot conçu sur 3D Builder

II.2.1.4. Le robot

La structure globale du bras manipulateur avec pince est illustrée par la Figure. II.5.



Figure II.5 : Bras robot entier conçu sur 3D Builder

II.2.2. Description des parties constitutives du bras manipulateur

Dans cette partie, on va parler des types d'actionneurs, capteurs et partie électronique que nous avons utilisée pour notre robot.

II.2.2.1. Partie électronique

La partie électronique a pris une place très importante dans notre projet car elle fait le lien entre l'application sur notre smartphone et les composants en lien direct avec l'extérieur (Arduino, servomoteurs...).

Tout d'abord, le cerveau du robot est l'Arduino UNO R3 présenté par la figure II.6.



Figure II.6 : Arduino UNO R3 [16]

Dans cette partie, nous détaillons les montages électroniques que nous avons pu réaliser étape par étape ainsi que le rôle de chaque module électronique utilisé pour bien apprendre comment le robot fonctionne. Pour assurer le déplacement du robot, on a utilisé les servomoteurs.

II.2.2.2. Les actionneurs du bras manipulateur

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

L'asservissement est un moyen de gérer, corriger une commande en fonction d'une consigne et d'un capteur de position. Pour pouvoir réaliser la correction de l'angle de l'actionneur, le servomoteur utilise une électronique d'asservissement. Cette électronique est constituée d'un comparateur qui compare la position du bras du servomoteur à la consigne. Après une rapide comparaison entre la consigne et la valeur réelle de la position du bras, le servomoteur applique une correction si le bras n'est pas orienté à l'angle imposé par la consigne [22].

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

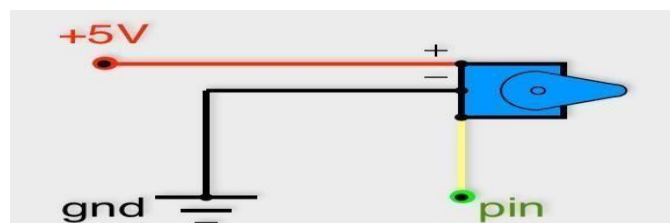


Figure II.7: Schéma d'un actionneur



Figure II.8 : Servomoteurs :
 (a) servomoteur du type MG90S [23], (b) servomoteur du Type SG90 [24].

II.2.2.3. Les paramètres techniques des servomoteurs

Les paramètres techniques des servomoteurs utilisés sont donnés par **Table II.1**.

Table II.1. Tableau des caractéristiques des servomoteurs du bras [23], [24].

Références	MG90S	SG90
Caractéristiques		
Poids	13,4g	14,7g
Dimensions	22,5 x 12 x 35,5 mm	32 x 12 x 32 mm
Couples d'arrêt	1,8 kgf.cm (4,8V), 2,2 kgf.cm (6V)	2,5 kgf.cm
Tensions de fonctionnement	4,8V – 6V	4,8V – 6V
Vitesses	0,1s/60° (4,8V), 0,08s/60° (6V)	0,1s/60°
Angles de rotation	180°	180°

II.2.2.4. La commande des servomoteurs

La consigne envoyée au servomoteur est un signal électronique de type PWM. Il dispose cependant de deux caractéristiques indispensables pour que le servomoteur puisse fonctionner :

- La fréquence fixe : le signal qu'on va devoir générer doit avoir une fréquence de 50Hz,
- La durée de l'état haut : cette durée indique au servomoteur l'angle précis qui est souhaité par l'utilisateur [22].

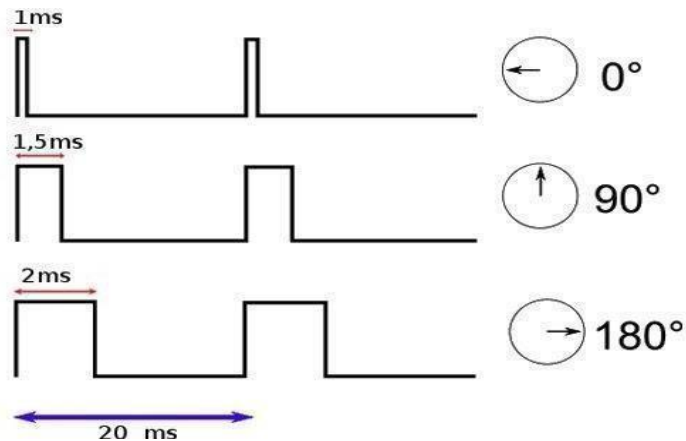


Figure II.9 : Exemple de signaux en position pour un servomoteur [22].

II.2.2.5. Transmission entre articulations: les engrenages

On trouve, au niveau de la pince, des engrenages reliés au servomoteur. Les transmissions par engrenages sont des organes mécaniques utilisés pour transformer et transmettre à un organe récepteur le couple et le mouvement de rotation générés par un moteur [25].

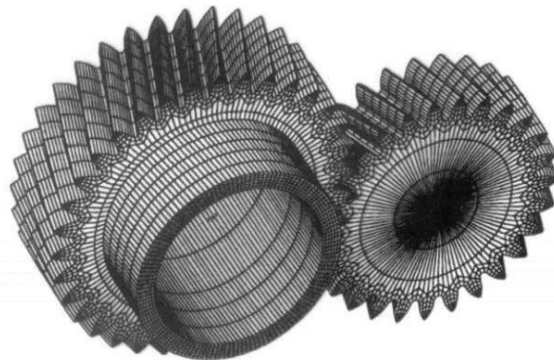


Figure II.10 : Schéma d'un engrenage [25]

II.2.2.6. Les capteurs

Le rôle des capteurs est de gérer les relations entre le robot et son environnement. Ils permettent de contrôler plus facilement les tâches qu'on lui ordonne.

Il est impor

tant de distinguer deux types de capteurs :

- Les capteurs proprioceptifs qui mesurent l'état mécanique interne du robot,
- Les capteurs extéroceptifs qui recueillent des informations sur l'environnement.

Dans notre projet, on suppose qu'on travaille dans un espace sans obstacle. L'utilisation de capteurs de détection n'est donc pas nécessaire. On n'a aussi pas besoin de rajouter des capteurs de position, car ceux-ci sont déjà intégrés dans les servomoteurs. En revanche, on a utilisé un récepteur Bluetooth afin de pouvoir commander ses déplacements à l'aide d'une application mobile

II.2.2.7. Les paramètres techniques du récepteur Bluetooth

Les spécifications techniques du composant Bluetooth utilisé sont données par **Table II.2**.

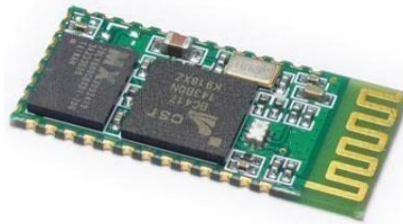


Figure II.11: Module Bluetooth [26]

Table II.2. Tableau des caractéristiques du récepteur Bluetooth du bras [26].

Caractéristiques	Bluetooth HC-05
Sensibilité typique	-80dBm
Puissance d'émission RF	Jusqu'à +4 dBm
Tension de fonctionnement	1,8V - 3,6V

II.2.3. Description du kit de robot

Cette partie ne fut que très rapidement abordée. En effet, au vu des composants disponibles, le châssis est livré en pièces détachées qui doivent être assemblées. Des instructions de montage détaillées sont incluses dans le kit.

La plate-forme utilisée est très simple capable de porter les composants du système, composée de quatre moteurs à courant continu (CC), quatre roues, d'autres modules que nous allons voir par la suite, et toutes les pièces nécessaires pour assembler le châssis.

Il comprend également un boîtier où peuvent se loger deux piles.

Ce qui signifie que l'ensemble du projet peut être réalisé sans aucune soudure (l'outil le plus difficile à utiliser ne sera donc qu'un ou revis)

II.3.La Réalisation de robot mobile

➤ **Installation Matérielle**

➤ Etape 1 :

Pour commencer, nous allons présenter le matériel utilisé pour la réalisation du robot.



Le chassis



Moteur Shield L293D



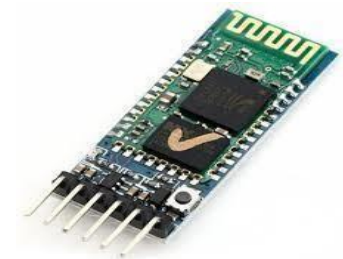
Arduino UNO R3



Les fils



Support de batterie



Module bluetooth HC-05

Figure II.12 : le matériel utilisé pour la réalisation du robot.

➤ Etape 2 :

Tout d'abord, on fixe les quatre supports plastiques (ce sont des blocs circulaires en noirs) à chaque moteur à l'aide de deux longs boulons et de deux écrous.

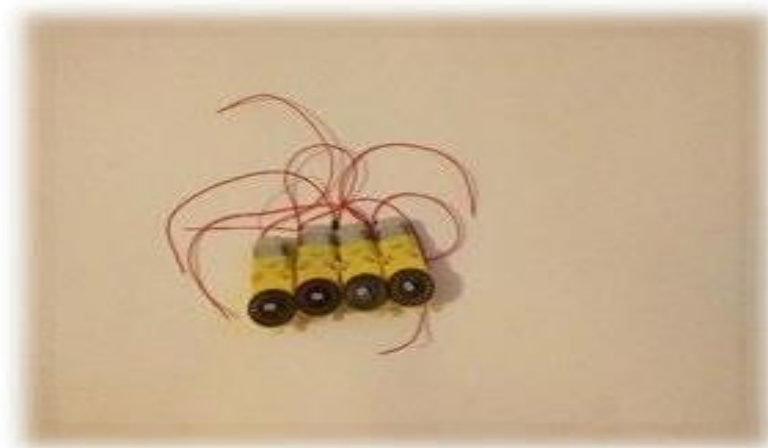


Figure II.13 : moteur à courant continu

➤ Etape 3 :

Maintenant, chaque moteur peut être fixé au châssis en utilisant deux boulons courts dans le bas de chaque support plastique. Voici une vue de la partie inférieure du châssis afin que nous puissions voir où les boulons doivent être placés.

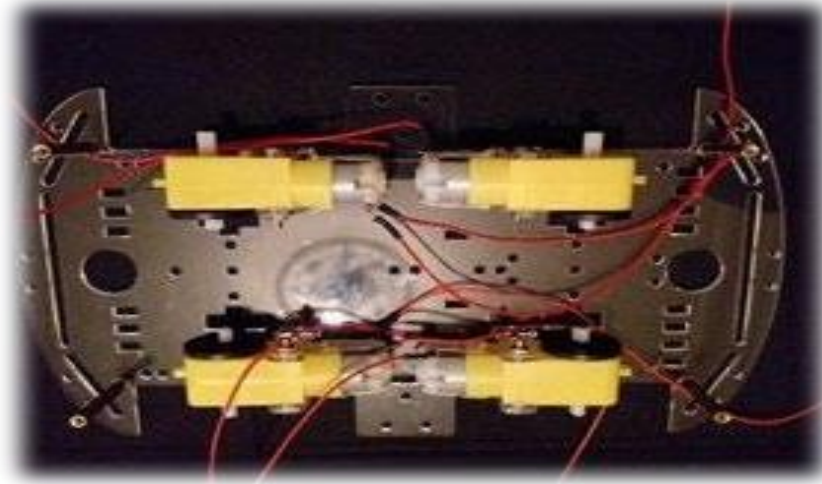


Figure II.14 : fixation des moteurs au châssis

➤ Etape 4 :

On installe la plaque d'entraînement des moteurs Shield L293D : sur la plaque Arduino UNO
Remarque : il est recommandé d'attendre que tous les fils soient fixés au pont en H avant de procéder à cette opération



Figure II.15 : Installation la plaque des moteurs sur le châssis

- Etape 5 :
Connectez le câble moteur, diagramme de circuit.

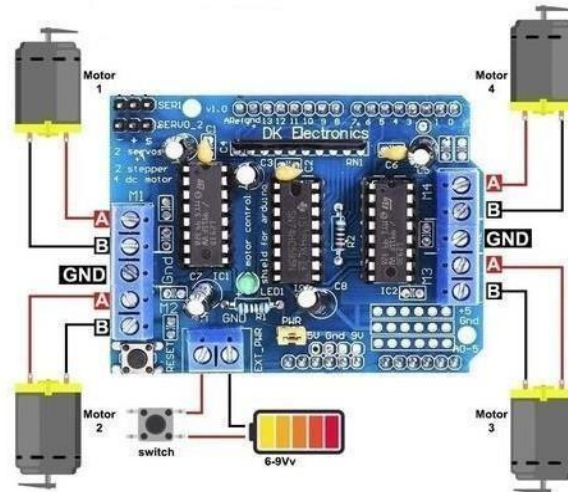


Figure II.16 : Moteur Shield L293D

- Etape 6 :
Câblage de l'alimentation (boîte à batterie) sur la partie supérieure du châssis. Fixation des roues : On fait attacher les roues sur les arbres du moteur et on fixe les quatre arbres aux endroits indiqués sur la photo ci-dessous. Ce robot commence à prendre forme.

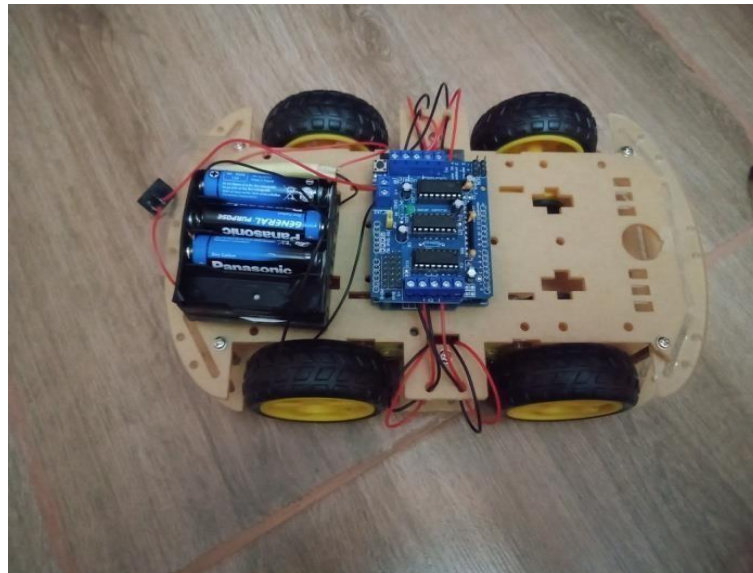


Figure II.17 : Câblage de l'alimentation

➤ Etape 7 :

On installe le module Bluetooth et on monte le moteur Drive L293D à la partie supérieure pour faciliter le travail.

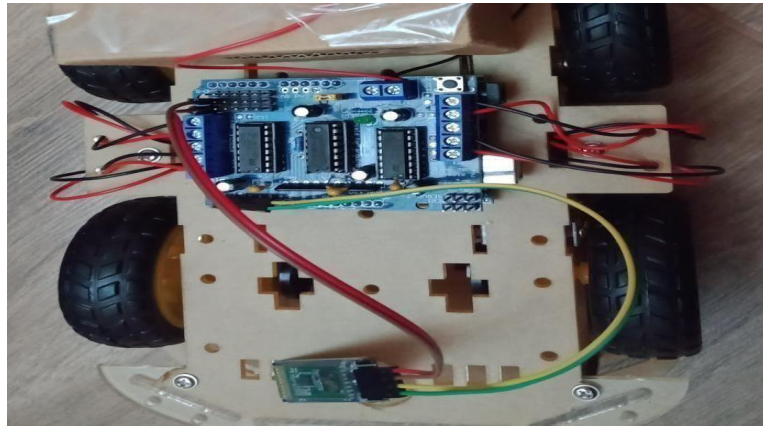


Figure II.18 : Installation du module Bluetooth

➤ Etape 8 :

Fixation du servomoteur de la base à l'aide de quatre longs boulons et de deux écrous



Figure II.19 : La fixation du servomoteur de la base

➤ Etape 9 :

Fixé le servomoteur de la côté droite et gauche à l'aide de deux longs boulons et de deux écrous

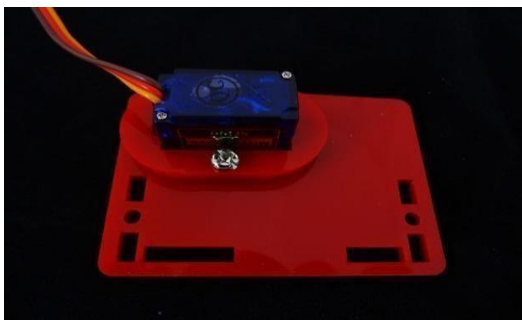


Figure II.20 : Fixation du servomoteur du côté droite et gauche

➤ Etape 10 :

Fixation du servomoteur de la pince à l'aide de quatre longs boulons et de deux écrous.



Figure II.21 : Fixation du servomoteur de la pince

➤ Etape 11:

La forme finale du bras manipulateur après avoir terminé l'assemblage de la base et du servomoteur du côté droite et du gauche

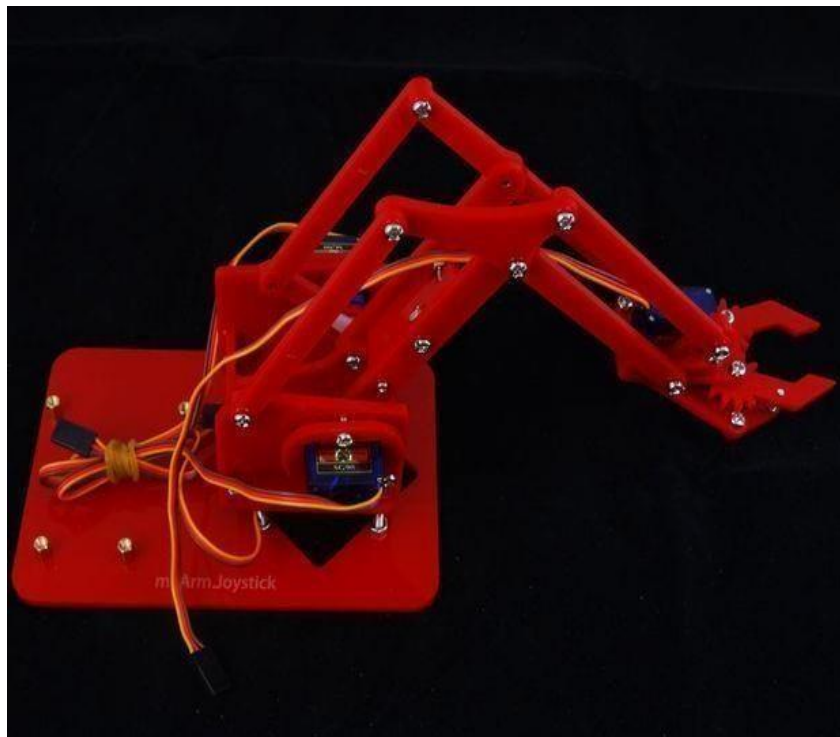


Figure II.22 : La forme finale du bras manipulateur

➤ Etape 12 :

Embarquer le bras manipulateur sur le robot mobile et c'est ça la Forme finale après avoir terminé l'assemblage.



Figure II.23 : Bras manipulateur embarqué sur un robot mobile

II.4.Conclusion

Dans ce chapitre nous avons présenté la conception et les différents composants de notre projet ; ainsi que le principe de fonctionnement du robot mobile, comme on a étudié les différents blocs de ce système tels que :

- Bloc de commande
- Bloc de communication
- Bloc d'alimentation

On a aussi présenté des différentes étapes possibles réalisées par notre projet. Dans ce qui suit, nous allons procéder à la programmation de notre projet selon les différents blocs.

CHAPITRE III :

Interface de l'Arduino et programmation

III.1.Introduction

Aujourd'hui, l'électronique est en plus remplacée par de l'électronique programmée. On parle aussi le système embarquée ou d'informatique embarquée. Dans le but Est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit.

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. L'électronique est devenue accessible à toutes personnes en ayant l'envie : ce que nous allons apprendre dans ce travail est un mélange d'électronique et de programmation. On va en effet parler d'électronique embarquée qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance de L'électronique.

III.2.Partie programme :

La carte d'acquisition qui se base dans sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme celle de cas de notre carte.

L'environnement de programmation open-source pour ARDUINO peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

III.3.L'environnement de développement Arduino :

Un IDE (environnement de développement) libre et gratuit est distribué sur le site d'Arduino (compatible Windows, Linux et Mac) [21].Langage Arduino est basé sur le langage C/C++, avec des fonctions et des bibliothèques spécifiques à l'Arduino. Les opérations de compilation et de chargement dans la mémoire du microcontrôleur étant ramenées à des clics sur des boutons dans l'IDE (très simple). La communication entre le PC et la carte se fait via le port USB, moyennant installation d'un driver adapté (fourni par ARDUINO). [22]

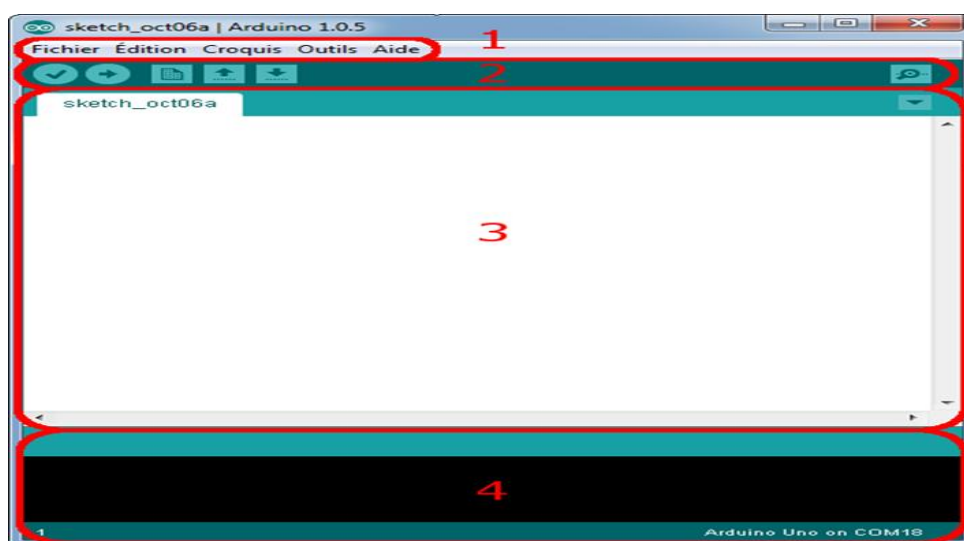


Figure III.1 : environnement de développement arduino

III.4. Interface Arduino :

- **Le cadre 1** : ce sont les options de configuration du logiciel (Barre de menu).
- **Le cadre 2** : contient des boutons qui vont nous servir lorsque l'on va programmer nos cartes (Barre des boutons).
- **Le cadre 3** : ce bloc va contenir le programme que nous allons créer (Zone de message des actions encours).
- **Le cadre 4** : le débogueur corrigé les fautes et affiche le message (le Console d'affichage des messages de compilation).

III.5. Structure générale du programme (IDE ARDUINO) :

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation.

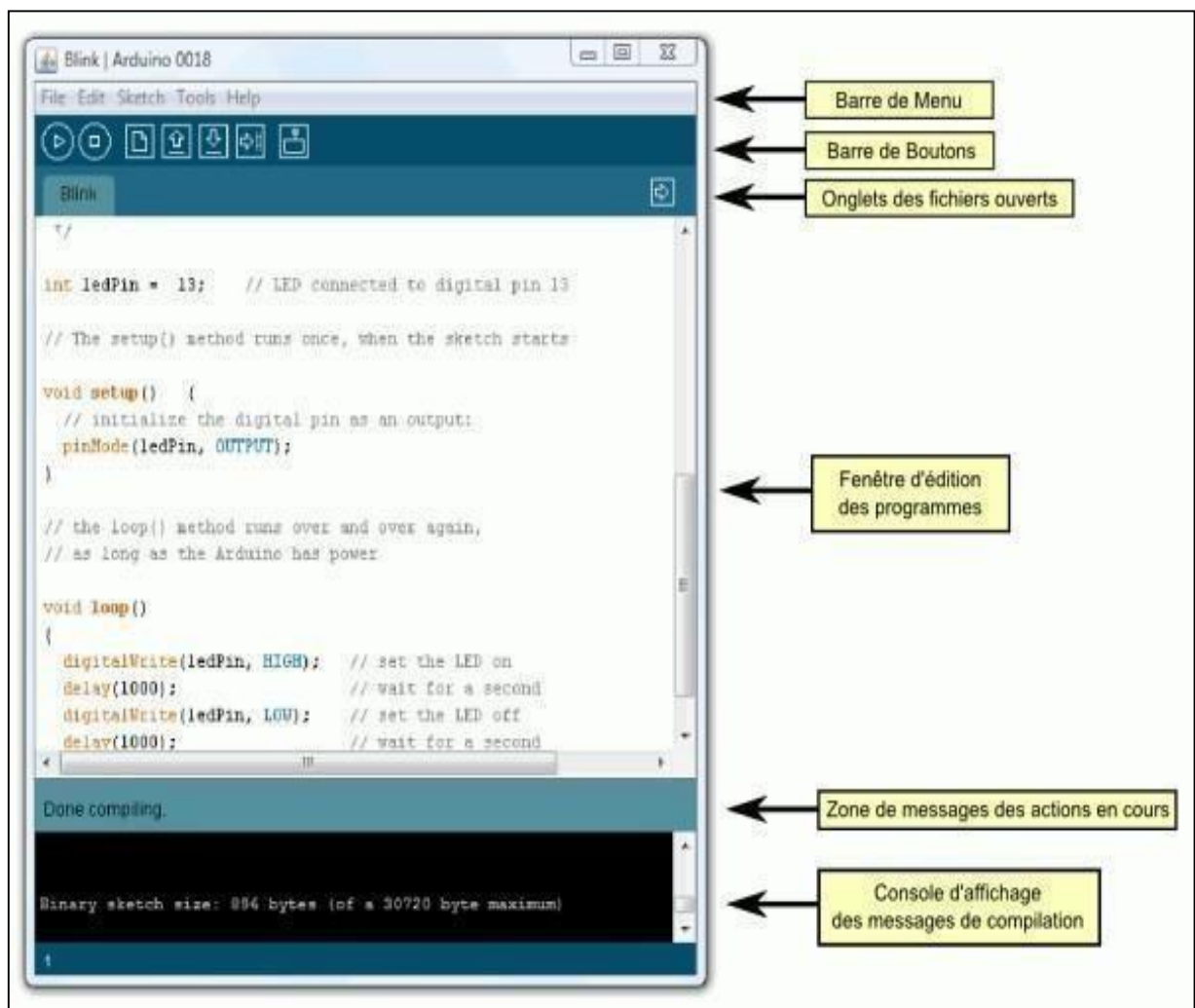


Figure III.2 : Interface IDE Arduino

III.6.Injection du programme :

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (ARDUINO UNO) et le numéro de port USB(COM12) comme à titre d'exemple cette figure suivante.

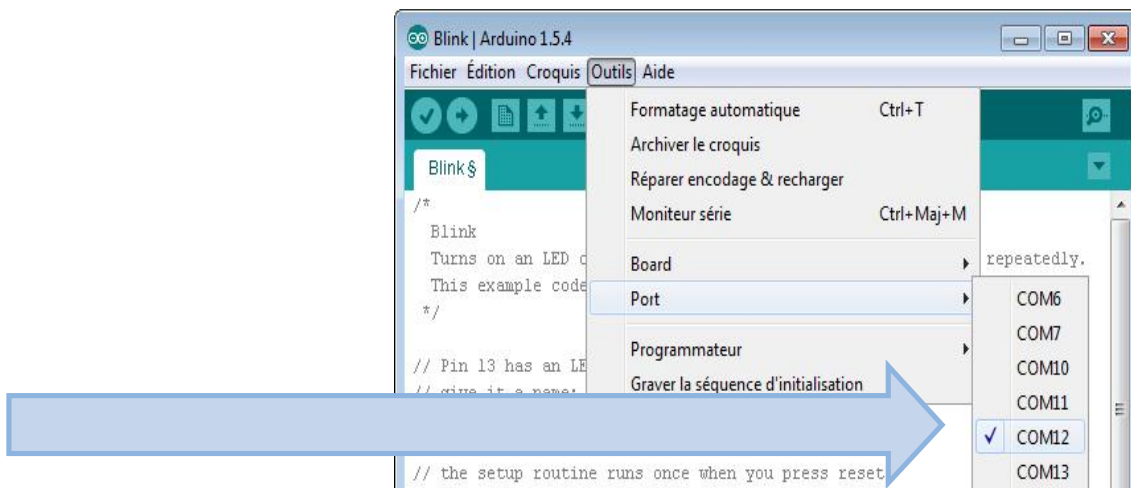


Figure III.3 : Paramètre de la carte.

III.6.1.Description du programme :

Un programme ARDUINO est une suite d'instruction salement aires sous forme textuelle (ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de codes.

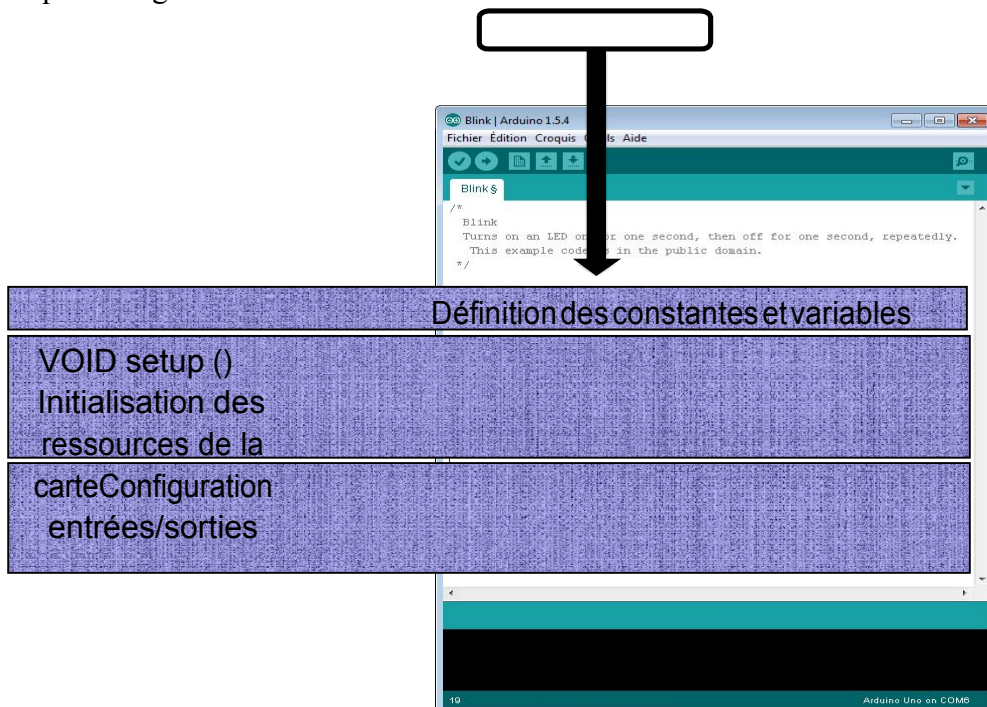


Figure III.4 : Description du programme sur Arduino

III.6.2. Les étapes de téléchargement du programme:

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la carte ARDUINO via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE ARDUINO.
2. On vérifie ce programme avec le logiciel ARDUINO (compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On charge le programme sur la carte.
5. On câble le montage électronique.
6. L'exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (pile 9 volts par exemple).
8. On vérifie que notre montage fonctionne.

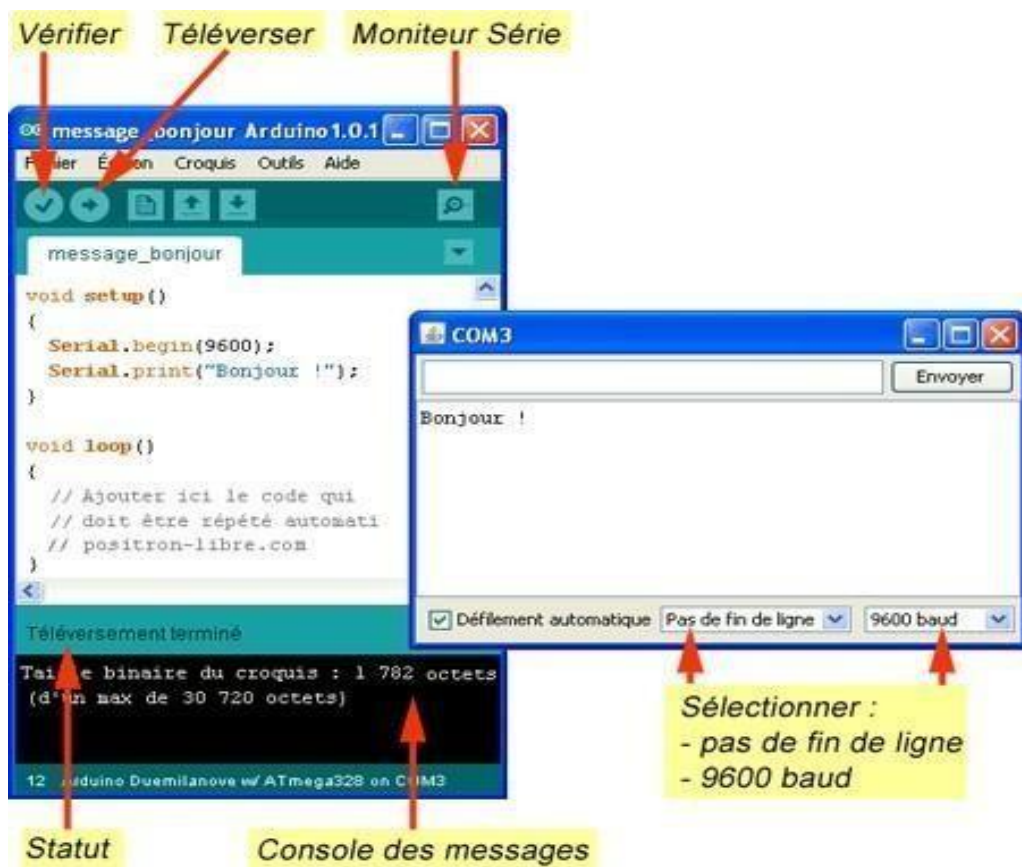


Figure III.5 : Les étapes de téléchargement du code.

III.7. Les accessoires de la carte ARDUINO :

La carte ARDUINO généralement est associée aux accessoires qui simplifient les réalisations.

III.8.Communication Bluetooth:

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.

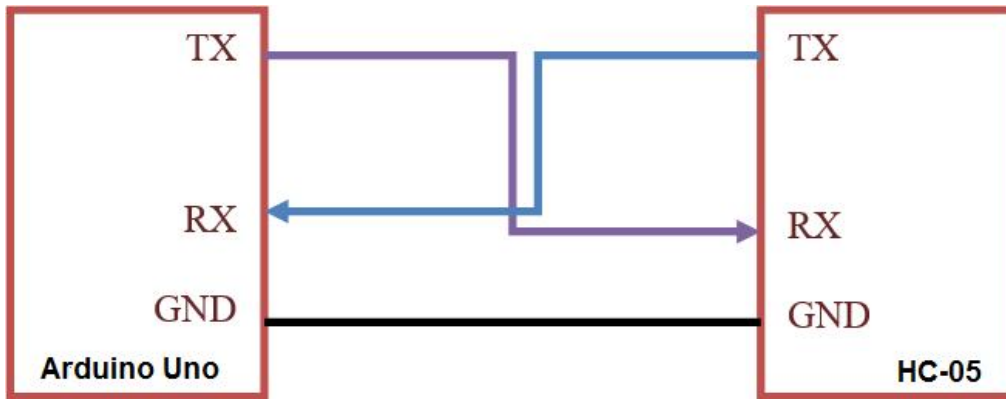


Figure III.6 : Schéma de la communication avec le module Bluetooth HC-05

Ainsi, chaque fois qu'un caractère/texte sera reçu, il sera envoyé sur la voie série de la carte de contrôle. Pour faire le lien avec l'application, on doit procéder par programmation en C++ qui permet de traduire les chaînes de caractères reçues via l'application en des informations pour les actionneurs du bras manipulateur.

Dans le programme, après avoir fait l'initialisation des servomoteurs et du module Bluetooth, on place le bras du robot dans sa position initiale en utilisant la fonction *write ()* qui permet de déplacer simplement le servomoteur dans n'importe quelle position désirée.

On organise, ensuite, notre programme en un ensemble de fonctions d'exécution pour chaque servomoteur (moteur 1, moteur 2, moteur 3 et moteur 4), auquel on fait appel grâce à l'outil conditionnel « if », qui teste à chaque fois le préfixe envoyé par l'application et qui indique la position actuelle et le curseur modifié du servomoteur sélectionné. Le sous-programme est défini comme suit :

```

////////////////////////////////// SERVO 1 - BASE ////////////////////////////////////
if(bt.startsWith("s1")){           // comprueba si la cadena comienza con "s1"
    btS = bt.substring(2,bt.length()); // extraigo los caracteres desde la posición 2 en adelante
    s1Act = btS.toInt();             // convierto de string a entero
    //MUEVO EL SERVOMOTOR CON UN BARRIDO
    if(s1Ant > s1Act){
        for(int j=s1Ant; j>=s1Act; j--){
            servol.write(j);
            delay(s1Vel);
        }
    }
}

```

```

////////////////////////////////////// SERVO 2 - MOTOUR gauche //////////////////////////////////
if(bt.startsWith("s2")){
  btS = bt.substring(2, bt.length());
  s2Act = btS.toInt();
  Serial.println(s2Act);
  if(s2Ant > s2Act){
    for(int j=s2Ant; j>=s2Act; j--){
      servo2.write(j);
      servo3.write(180-j);
      delay(s2Vel);}}

////////////////////////////////////// SERVO 5 - moteur droite //////////////////////////////////
if(bt.startsWith("s5")){
  btS = bt.substring(2, bt.length());
  s5Act = btS.toInt();

  if(s5Ant > s5Act){
    for(int j=s5Ant; j>=s5Act; j--){
      servo5.write(j);
      delay(s5Vel);}}

////////////////////////////////////// SERVO 6 - pince //////////////////////////////////
if(bt.startsWith("s6")){
  btS = bt.substring(2, bt.length());
  s6Act = btS.toInt();
  Serial.println(s6Act);
  if(s6Ant > s6Act){
    for(int j=s6Ant; j>=s6Act; j--){
      servo6.write(j);
      delay(s6Vel);}}
  else{
    for(int j=s6Ant; j<=s6Act; j++){
      servo6.write(j);
}

```

La fonction *Serial.available()* permet d'obtenir le nombre d'octets (caractères) disponibles pour la lecture à partir du port série. La fonction *serial.read()* nous permet de lire ces dernières. Chaque fonction doit être programmée de manière à convertir la chaîne de caractères reçue en un entier qui désignera l'angle de déplacement du servomoteur. Le sous-programme de la première fonction est donné ci-après :

```

void setup() {
  Serial.begin(9600);
  pinMode(Trig, OUTPUT);
  pinMode(Echo, INPUT);
  servo.attach(motor);
}
void Bluetoothcontrol() {
  if (Serial.available() > 0) {
    value = Serial.read();
    Serial.println(value);
  }
}

```

La fonction *Serial.println(readString.toInt())* nous permet la conversion de la chaîne de caractères en un entier. La fonction *write(readString.toInt())* nous permet le déplacement du servomoteur vers l'entier (angle) obtenu après conversion.

Pour les trois boutons, on ordonne notre programme qui fonctions d'exécution cycle automatique à qui on fait appel (une fois le bouton actionné) avec l'outil conditionnel « if ». Chaque bouton est créé selon une trajectoire choisie. Le sous-programme est défini comme suit :

```

//////////////////////////////////// SAVE //////////////////////////////////////
if(bt.startsWith("SAVE")){
  s1[index]= s1Ant;
  s2[index]= s2Ant;
  s4[index]= s4Ant;
  s5[index]= s5Ant;
  s6[index]= s6Ant;
  s7[index]= s7Ant;
  index++;}

//////////////////////////////////// RESET //////////////////////////////////////
if(bt.startsWith("RESET")){
  //memset se utiliza para rellenar un bloque de memoria con un valor determinado
  //memset("matriz", "dato a rellenar", "tamaño de la matriz")
  memset(s1,0,sizeof(s1)); //rellena con el valor de "0" todas las posiciones de la matriz
  memset(s2,0,sizeof(s2));
  memset(s4,0,sizeof(s4));
  memset(s5,0,sizeof(s5));
  memset(s6,0,sizeof(s6));
  memset(s7,0,sizeof(s7));
  index=0;}

//////////////////////////////////// RUN //////////////////////////////////////
if(bt.startsWith("RUN")){
  //Ejecuta los pasos una y otra vez hasta que se pulse el botón "RESET" o espera hasta que se vuelva a pulsar "RESET"
  while(bt.startsWith("RESET") != true){//MIENTRAS NO SE PRESIONE 'RESET' SE MANTIENE EN EL BUCLE Y SOLO FUNCIONA BOTON "PAUSE"
    for(int i=0; i<= index-2;i++){
      if(Bluetooth.available()>0){
        bt = Bluetooth.readString();
        if(bt.startsWith("PAUSE") == true){
          //Ejecuta los pasos una y otra vez hasta que se pulse el botón "RUN" o espera hasta que se vuelva a pulsar "RUN"
          while(bt.startsWith("RUN") != true){//MIENTRAS NO SE PRESIONE 'RUN' SE MANTIENE EN EL BUCLE Y SOLO FUNCIONA BOTON "RESET"
            if(Bluetooth.available()>0){
              bt = Bluetooth.readString();
              if(bt.startsWith("RESET") == true){
                break;}
            }
          }
        }
      }
    }
  }
}

```

Comme précisé précédemment, le programme exécute directement les angles désirés sauvegardées dans chaque articulation. L'inclusion des retards est faite afin d'éviter les déplacements rapides entre les différents organes du robot qui peuvent causer des dégâts matériels.

Le programme détaillé est donné dans l'annexe D.

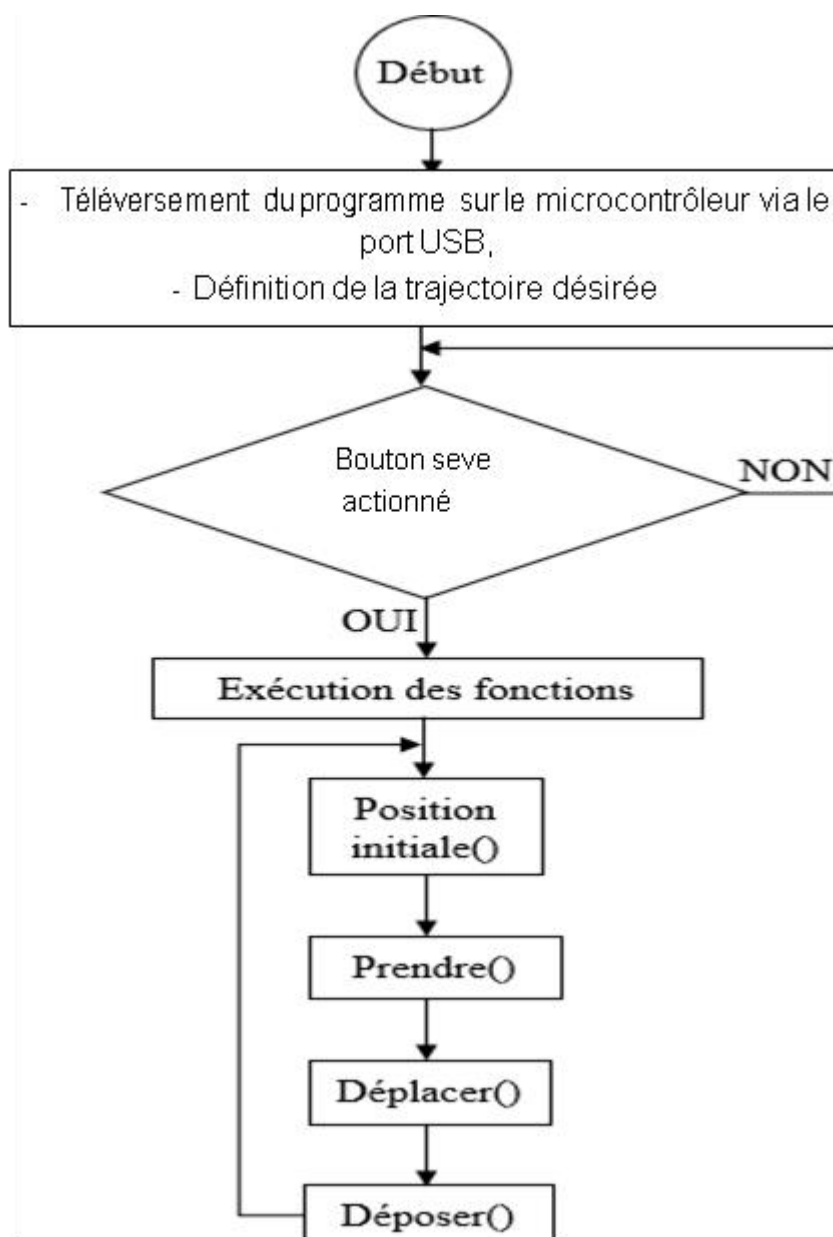


Figure III.7 : Logigramme résumant la commande à angles sauvegarde

III.9. Le code Arduino:

```

/*
  DIY Arduino Robot Arm Smartphone Control
  by Dejan, www.HowToMechatronics.com
*/

#include <SoftwareSerial.h>
#include <Servo.h>
#include <AFMotor.h>
#define motor 10
#define Speed 170
#define spoint 103

Servo servo01;
Servo servo02;
Servo servo05;
Servo servo06;

SoftwareSerial Bluetooth(10, 11); // Arduino(RX, TX) - HC-05 Bluetooth (TX, RX)
char value;
int distance;
int Left;
int Right;
int L = 0;
int R = 0;
int L1 = 0;
int R1 = 0;
AF_DCMotor M1(1);
AF_DCMotor M2(2);
AF_DCMotor M3(3);
AF_DCMotor M4(4);
int servo1Pos, servo2Pos, , servo5Pos, servo6Pos; // current position
int servo1PPos, servo2PPos,, servo5PPos, servo6PPos; // previous position
int servo01SP[50], servo02SP[50], servo05SP[50], servo06SP[50]; // for storing positions/steps
int speedDelay = 20;
int index = 0;
String dataIn = "";

void setup() {
  servo01.attach(5);
  servo02.attach(6);
  servo05.attach(9);
  servo06.attach(3);
  Bluetooth.begin(38400); // Default baud rate of the Bluetooth module
  Bluetooth.setTimeout(1);
  delay(20);
  M1.setSpeed(Speed);
  M2.setSpeed(Speed);
  M3.setSpeed(Speed);
  M4.setSpeed(Speed);
  // Robot arm initial position
  servo1PPos = 90;
  servo01.write(servo1PPos);
  servo2PPos = 150;
  servo02.write(servo2PPos);
  servo5PPos = 85;
  servo05.write(servo5PPos);
  servo6PPos = 80;
  servo06.write(servo6PPos);
}

void loop() {
}

void Bluetoothcontrol() {
  if (Serial.available() > 0) {

```

```

    value = Serial.read();
    Serial.println(value);
}
if (value == 'F') {
    forward();
} else if (value == 'B') {
    backward();
} else if (value == 'L') {
    left();
} else if (value == 'R') {
    right();
} else if (value == 'S') {
    Stop();
}
// Check for incoming data
if (Bluetooth.available() > 0) {
    dataIn = Bluetooth.readString(); // Read the data as string

    // If "Waist" slider has changed value - Move Servo 1 to position
    if (dataIn.startsWith("s1")) {
        String dataInS = dataIn.substring(2, dataIn.length()); // Extract only the number. E.g. from "s1120" to "120"
        servo1Pos = dataInS.toInt(); // Convert the string into integer
        // We use for loops so we can control the speed of the servo
        // If previous position is bigger than current position
        if (servo1PPos > servo1Pos) {
            for (int j = servo1PPos; j >= servo1Pos; j--) { // Run servo down
                servo01.write(j);
                delay(20); // defines the speed at which the servo rotates
            }
        }
        // If previous position is smaller than current position
        if (servo1PPos < servo1Pos) {
            for (int j = servo1PPos; j <= servo1Pos; j++) { // Run servo up
                servo01.write(j);
                delay(20);
            }
        }
        servo1PPos = servo1Pos; // set current position as previous position
    }

    // Move Servo 2
    if (dataIn.startsWith("s2")) {
        String dataInS = dataIn.substring(2, dataIn.length());
        servo2Pos = dataInS.toInt();

        if (servo2PPos > servo2Pos) {
            for (int j = servo2PPos; j >= servo2Pos; j--) {
                servo02.write(j);
                delay(50);
            }
        }
        if (servo2PPos < servo2Pos) {
            for (int j = servo2PPos; j <= servo2Pos; j++) {
                servo02.write(j);
                delay(50);
            }
        }
        servo2PPos = servo2Pos;
    }

    // Move Servo 5
    if (dataIn.startsWith("s5")) {
        String dataInS = dataIn.substring(2, dataIn.length());
        servo5Pos = dataInS.toInt();
        if (servo5PPos > servo5Pos) {
            for (int j = servo5PPos; j >= servo5Pos; j--) {
                servo05.write(j);
            }
        }
    }
}

```

```

    delay(30);
  }
}
if (servo5PPos < servo5Pos) {
  for ( int j = servo5PPos; j <= servo5Pos; j++) {
    servo05.write(j);
    delay(30);
  }
}
servo5PPos = servo5Pos;
}
// Move Servo 6
if (dataIn.startsWith("s6")) {
  String dataInS = dataIn.substring(2, dataIn.length());
  servo6Pos = dataInS.toInt();
  if (servo6PPos > servo6Pos) {
    for ( int j = servo6PPos; j >= servo6Pos; j--) {
      servo06.write(j);
      delay(30);
    }
  }
  if (servo6PPos < servo6Pos) {
    for ( int j = servo6PPos; j <= servo6Pos; j++) {
      servo06.write(j);
      delay(30);
    }
  }
  servo6PPos = servo6Pos;
}
// If button "SAVE" is pressed
if (dataIn.startsWith("SAVE")) {
  servo01SP[index] = servo1PPos; // save position into the array
  servo02SP[index] = servo2PPos;
  servo05SP[index] = servo5PPos;
  servo06SP[index] = servo6PPos;
  index++; // Increase the array index
}
// If button "RUN" is pressed
if (dataIn.startsWith("RUN")) {
  runservo(); // Automatic mode - run the saved steps
}
// If button "RESET" is pressed
if ( dataIn == "RESET") {
  memset(servo01SP, 0, sizeof(servo01SP)); // Clear the array data to 0
  memset(servo02SP, 0, sizeof(servo02SP));
  memset(servo05SP, 0, sizeof(servo05SP));
  memset(servo06SP, 0, sizeof(servo06SP));
  index = 0; // Index to 0
}
}
}

// Automatic mode custom function - run the saved steps
void runservo() {
  while (dataIn != "RESET") { // Run the steps over and over again until "RESET" button is pressed
    for (int i = 0; i <= index - 2; i++) { // Run through all steps(index)
      if (Bluetooth.available() > 0) { // Check for incoming data
        dataIn = Bluetooth.readString();
        if ( dataIn == "PAUSE") { // If button "PAUSE" is pressed
          while (dataIn != "RUN") { // Wait until "RUN" is pressed again
            if (Bluetooth.available() > 0) {
              dataIn = Bluetooth.readString();
              if ( dataIn == "RESET") {
                break;
              }
            }
          }
        }
      }
    }
  }
}
}
}

```

```

}
// If speed slider is changed
if (dataIn.startsWith("ss")) {
  String dataInS = dataIn.substring(2, dataIn.length());
  speedDelay = dataInS.toInt(); // Change servo speed (delay time)
}
}
// Servo 1
if (servo01SP[i] == servo01SP[i + 1]) {
}
if (servo01SP[i] > servo01SP[i + 1]) {
  for (int j = servo01SP[i]; j >= servo01SP[i + 1]; j--) {
    servo01.write(j);
    delay(speedDelay);
  }
}
if (servo01SP[i] < servo01SP[i + 1]) {
  for (int j = servo01SP[i]; j <= servo01SP[i + 1]; j++) {
    servo01.write(j);
    delay(speedDelay);
  }
}
}

```

```

// Servo 2
if (servo02SP[i] == servo02SP[i + 1]) {
}
if (servo02SP[i] > servo02SP[i + 1]) {
  for (int j = servo02SP[i]; j >= servo02SP[i + 1]; j--) {
    servo02.write(j);
    delay(speedDelay);
  }
}
if (servo02SP[i] < servo02SP[i + 1]) {
  for (int j = servo02SP[i]; j <= servo02SP[i + 1]; j++) {
    servo02.write(j);
    delay(speedDelay);
  }
}
}

```

```

// Servo 5
if (servo05SP[i] == servo05SP[i + 1]) {
}
if (servo05SP[i] > servo05SP[i + 1]) {
  for (int j = servo05SP[i]; j >= servo05SP[i + 1]; j--) {
    servo05.write(j);
    delay(speedDelay);
  }
}
if (servo05SP[i] < servo05SP[i + 1]) {
  for (int j = servo05SP[i]; j <= servo05SP[i + 1]; j++) {
    servo05.write(j);
    delay(speedDelay);
  }
}
}

```

```

// Servo 6
if (servo06SP[i] == servo06SP[i + 1]) {
}
if (servo06SP[i] > servo06SP[i + 1]) {
  for (int j = servo06SP[i]; j >= servo06SP[i + 1]; j--) {
    servo06.write(j);
    delay(speedDelay);
  }
}
}

```



```
}  
}  
if (servo06SP[i] < servo06SP[i + 1]) {  
  for ( int j = servo06SP[i]; j <= servo06SP[i + 1]; j++) {  
    servo06.write(j);  
    delay(speedDelay);  
  }  
}
```

III.10.Conclusion

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 se fait à l'aide d'une application mobile réalisée (sans fil) et grâce au Shield

CHAPITRE IV :

Réalisation d'une application Android

IV.1.Introduction

L'Android est parmi les derniers systèmes d'exploitation qui développent les exigences des téléphones intelligents. La plateforme Android de smartphone devient de plus en plus importante pour les réalisateurs de logiciel, en raison de ses puissantes possibilités et opensource.

Lors des années précédentes, le traitement des données informatiques se fait par des ordinateurs, en revanche le smartphone a des avantages qui ont les mêmes fonctions que l'outil informatique, ce dernier porte l'intérêt de l'ordinateur grâce à l'Android.

La téléphonie mobile a connu une explosion dans les années 2000 mais aucune révolution n'a semblé arriver de puisque les appareils se ressemblent. Les innovations n'avaient plus vraiment de saveur, les applications étaient difficiles d'accès de par leur mode de distribution et souvent peu performantes à cause des faibles capacités des appareils.

Depuis quelques mois, les smartphones sont dotés d'une puissance plus importante et d'espaces de stockage conséquents. Les téléphones tendent à devenir des objets artistiques, presque de reconnaissance sociale, et possèdent des fonctionnalités qu'aucun téléphone ne pouvait espérer au pare avant : connexion haut débit, localisation GPS, boussole, accéléromètre, écran tactile souvent multipoint, marché d'applications en ligne. Autant de qualités permettant de créer des applications innovantes et de les distribuer en toute simplicité.

La plate-forme Android apporte tout cela au consommateur, mais surtout, elle affranchit le développeur de nombreuses contraintes. Par son ouverture, elle permet à n'importe quel développeur de créer ses applications avec un ticket d'entrée quasi nul. Le Framework et le système d'exploitation et outils associés ont un code source ouvert, leur accès est gratuit et illimité. Plus besoin de

Négocier avec le constructeur du téléphone pour qu'il vous laisse développer sur sa plate-forme.

Tous les développeurs sont ainsi sur un même pied d'égalité, tous peuvent ajouter de la mobilité à des applications existantes.

Cette partie de notre étude ne nous donnera pas de bons résultats si on néglige

Certains paramètres, donc le bon fonctionnement de notre système se base essentiellement sur une bonne démarche et une bonne réflexion de notre programme.

IV.2.Présentation de l'Android



Figure IV.1 : Android

Android est un système d'exploitation libre qui utilise les téléphones mobiles modernes nombre (smartphones=smartphones), est basé sur Linux. Android est détenue par Google. Il existe d'autres systèmes d'exploitation de téléphone comme IOS d'Apple (20%) et Microsoft Windows Phone (2%), Android (78%).

L'Android vous pouvez facilement programmer et effectuer des programmes de formules mathématiques, des jeux, emplacement sur Google Maps, le déplacement d'un robot via Bluetooth, réponse automatique de SMS ... [23]

IV.3.Présentation d'App INVENTOR



Figure IV.2 : logo mit app inventor

APP INVENTOR est un IDE (environnement de développement intégré) qui permet la création d'applications destinées à des systèmes équipés de plateformes Android tels que les téléphones (tactiles ou non) et les tablettes (ACER, SAMSUNG...).

A l'origine le projet APP INVENTOR a été créé par le Google Labos puis abandonné. Aujourd'hui ce concept est repris par le MIT (Massachusetts Institute of Technologie). Le téléchargement et l'utilisation de APP INVENTOR nécessite un compte Google(Gmail). [24]

Cet environnement de programmation permet une programmation graphique, basée sur l'assemblage de blocs (langage Scratch). Des connaissances en programmation orientée objet sont toutefois nécessaires. Cependant, on pourra s'affranchir complètement de la connaissance des noms des propriétés et méthodes liées aux objets.

En effet des blocs de propriétés, méthodes et événements seront directement proposés dès la création d'un objet.

L'IDE est formé de deux composantes logicielles :

Un site Web permettant de créer l'interface homme machine (aspect graphique) de la future application ;

Un programme résidant sur l'ordinateur, qui permet par association des blocs liés aux objets Précédemment définis dans l'espace web, de créer le comportement de l'application.

Il permet également le test de l'application sur un émulateur ou le téléchargement vers le média désiré (tablette, téléphone...).

IV.4.La Création un projet :

Depuis un navigateur internet ouvrir le compte Google.

Aller sur le site du MIT App INVENTOR : <http://mit App inventor>.



Figure IV.3 : crée un nouveau projet

IV.4.1. Le concept d'App INVENTOR

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.



Figure IV.4 : App INVENTOR LOGO

Il faut donc une liaison internet et un navigateur internet comme Mozilla Firefox ou Chrome.

App Inventor est une application web open-source à l'origine fournie par Google, et maintenant maintenue par le Massachusetts Institute of Technology (MIT). Cette plate-forme de développement est utilisable par toutes et tous à condition de posséder un compte Gmail. La programmation est réalisée sans taper une seule ligne de code, mais simplement en associant et en paramétrant des briques logicielles toutes faites (langage Scratch).

Des connaissances en programmation orientée objet peuvent toutefois être nécessaires. Cependant, on pourra s'affranchir complètement de la connaissance des noms des propriétés et méthodes liées aux objets. En effet des blocs de propriétés, méthodes et événements seront directement proposés dès la création d'un objet sans que l'on s'en rende compte ce qui simplifie la programmation [25].

IV.5. L'environnement de développement et les outils d'application

IV.5.1 Accès à l'application et création d'un nouveau projet

En suivant les instructions suivantes

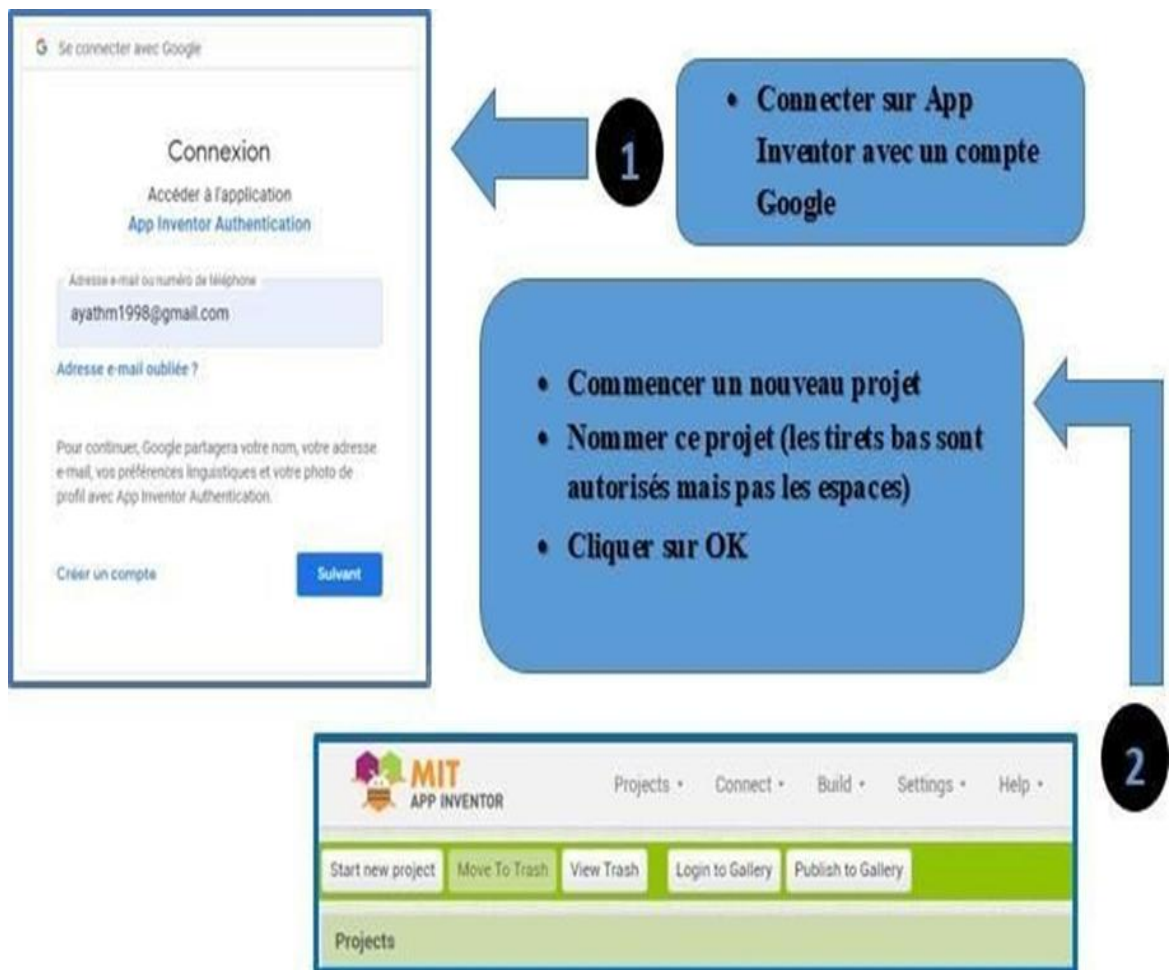


Figure IV.5 : Fenêtre d'accès au logiciel

IV.5.2. Description de l'espace Designer

En utilisant Design Windows, ou "Designer", vous pouvez visualiser l'apparence de votre application et sélectionner les composants qu'elle doit avoir. Vous pouvez aussi choisir des options pour l'interface utilisateur telles que : boutons, images et des fonctionnalités telles que caméra, capteurs et GPS.



Figure IV.6 : Les zones du logiciel

IV.5.3. Description de l'environnement de programmation

Le Blocks Editor permet de programmer le comportement de l'application. Ce sont des blocs intégrés pouvant gérer des choses comme les maths, la logique et le texte.

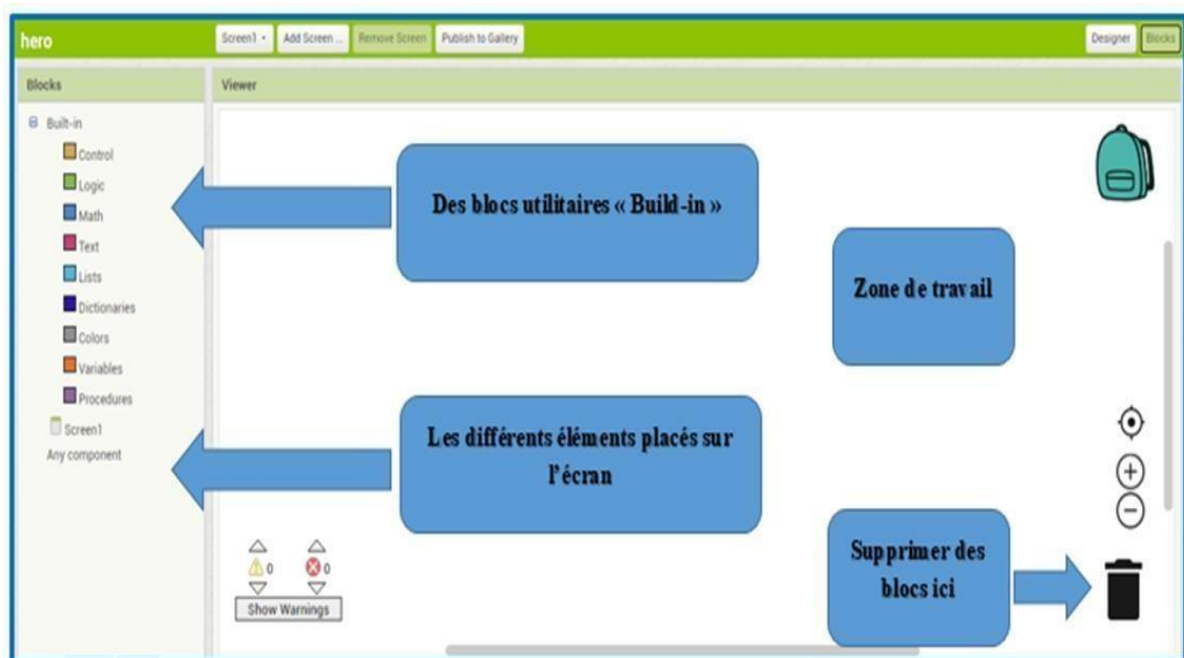


Figure IV.7 : Interface de programmation

IV.5.4 Description de l'émulateur

Afin de tester l'application vous pouvez choisir le test sur :

- L'émulateur : un écran s'affichera sur l'ordinateur,
- AI Companion : dans ce cas, la connexion se fera directement sur le smartphone en wifi,
- USB : la connexion se fera sur le smartphone via un câble USB.

Sur le smartphone, l'application MIT AI2 Companion doit être installée à télécharger sur le Play Store. Sur le PC, l'application MIT Emulator in AI2 doit être installée.

En choisissant AI Companion : un Code de 6 caractères est généré ainsi qu'un QR Code. Il Suffit de saisir le code ou de scanner le QR Code pour que l'application soit visible sur le smartphone [26].

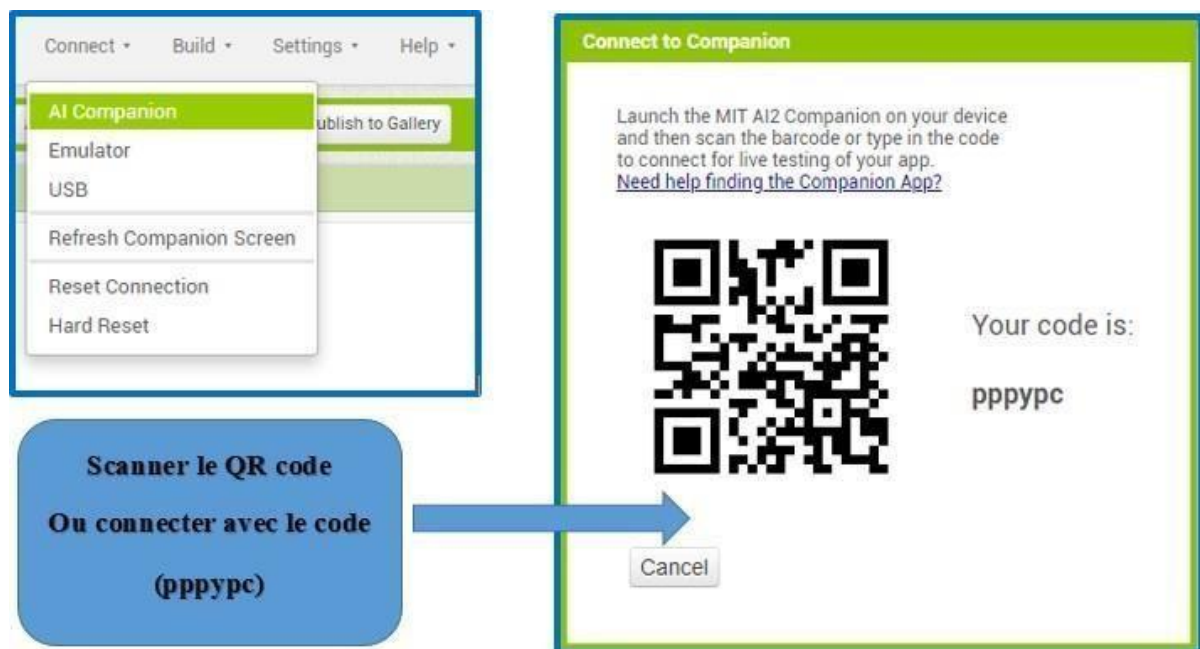


Figure IV.8 : Simulation de l'application sur smartphone

IV.6. Application Android 'Ouss bras manipulateur'

Notre application «Ouss bras robotique » est constituée d'une seule interface.

IV.6.1. Réalisation de l'application Android

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

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.

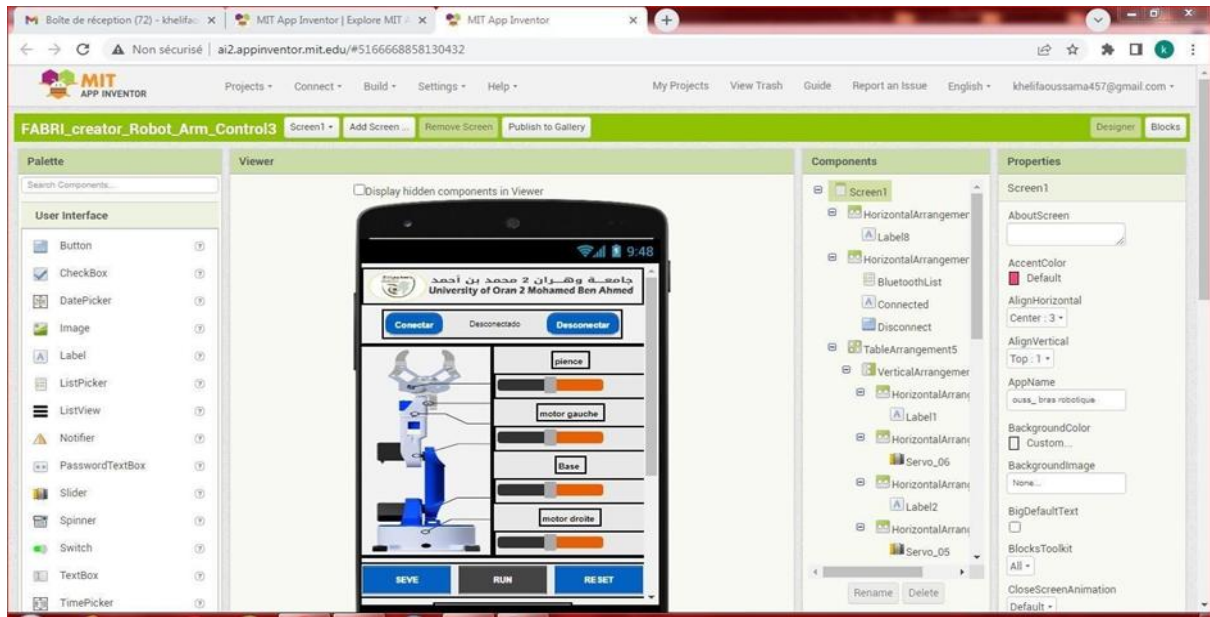


Figure IV.9 : Partie1 design de notre application sous App INVENTOR

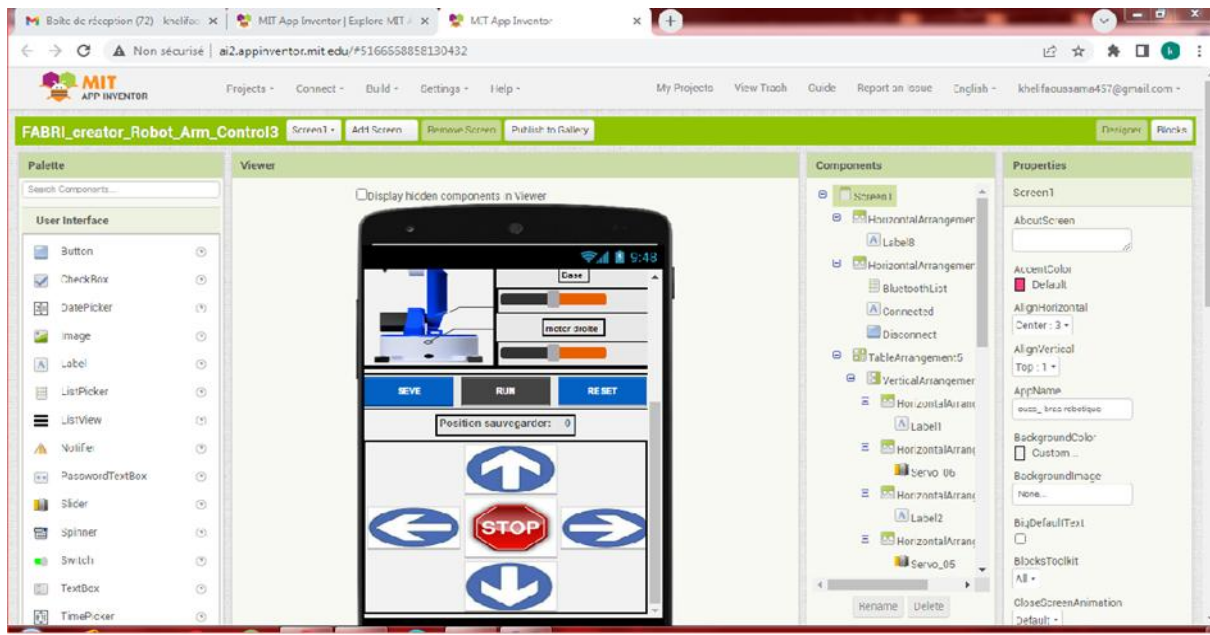


Figure IV.10 : Partie2 design de notre application sous App INVENTOR

Dans la partie bloc, on retrouve toutes les fonctions mathématiques, logiques, ou événementielles, ce rapportant aux objets utilisés dans la partie graphique. Cela nous permet de tirer profit d'une simplicité de programmation orientée objet sous la

Forme de blocs préprogrammés et qu'ils ne restent plus qu'à organiser en fonction du programme souhaité.

Lorsque ces outils sont utilisés correctement, on peut réaliser une application simple et efficace, avec un design propre qui répond à nos besoins. Dans ce qui suit, on va expliquer le fonctionnement des blocs de programme et leurs utilités dans le programme ARDUINO pour le contrôle du bras.

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



Figure IV.11 : Définition des éléments de l'objet Bluetooth

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



Figure IV.12 : Blocs de contrôle des curseurs

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 .Send Text*, on envoie un texte à l'ARDUINO no. Ce texte se compose d'un préfixe qui indique quel curseur a été modifié (le préfixe « A » pour la pince jusqu'au préfixe « D » pour la base) ainsi que la valeur actuelle du curseur qui est *thumb Position*.

Comme nous avons faire avec les curseurs, on doit initialise une valeur initiale pour que notre robot mobile soit en mode stop dans le démarrage de l'application et lorsqu'on clique sur l'un des boutons on active une fonction. Chaque bouton correspond à une fonction différente, lorsqu'on relâche, le bouton réinitialiser sur le mode stop.

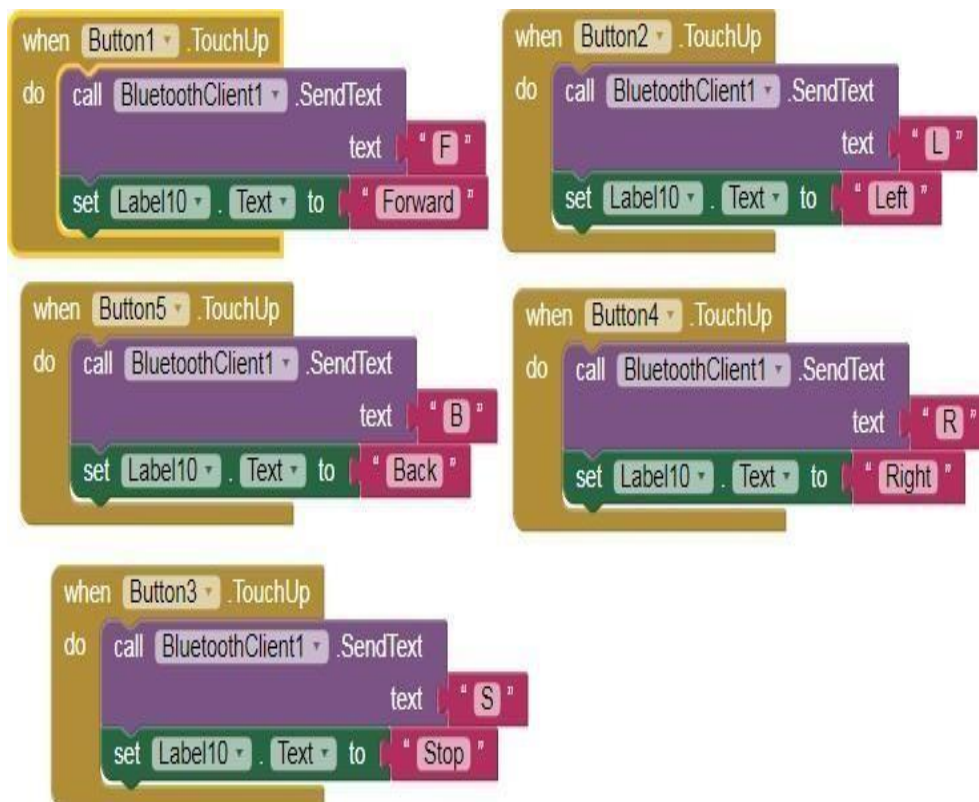


Figure IV.13 : configuration du block pour les boutons de robot mobile

Enfin on définit les blocs des trois boutons qui permettent d'exécuter une série d'instructions différentes. Celles-ci forment un cycle automatique, et peuvent être modifiés selon le besoin .qui peuvent être aussi un cycle répétitive

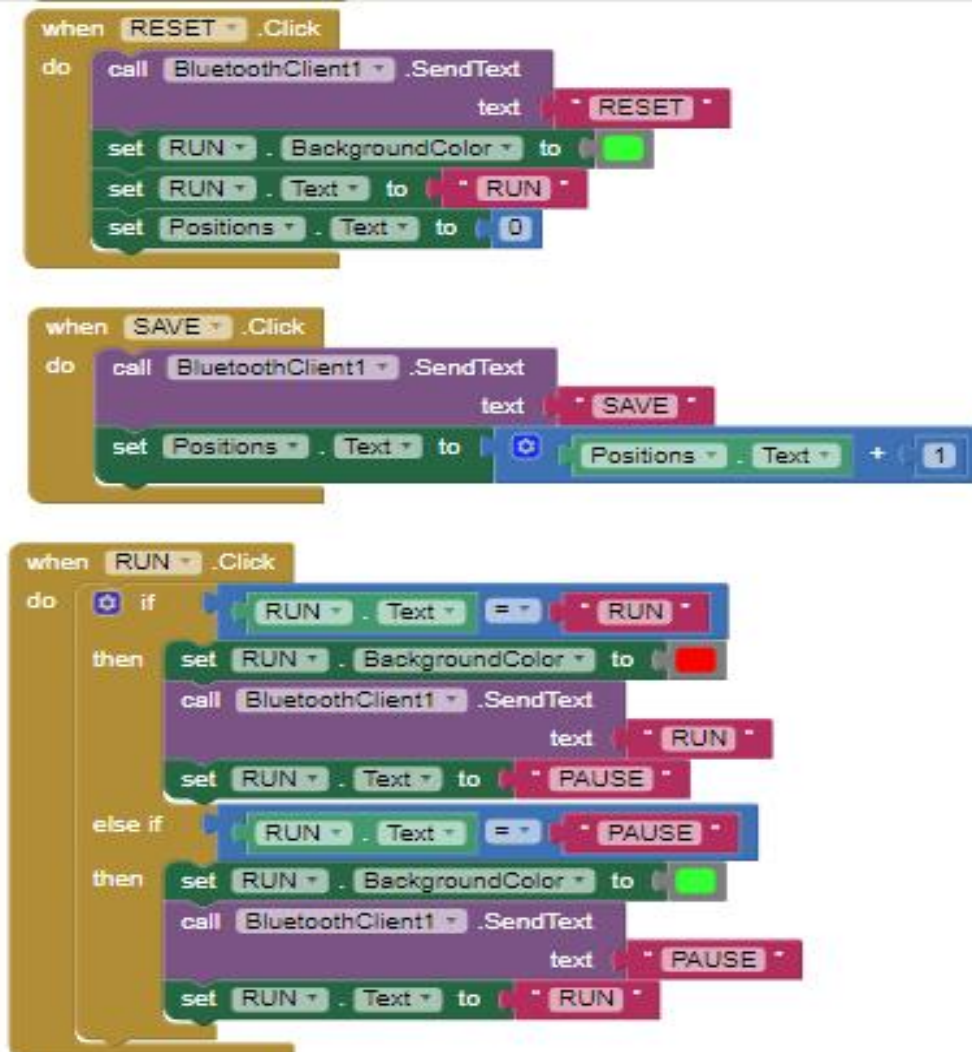


Figure IV.14 : Blocs des trois boutons

IV.6.2.L'interface finale de notre application

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



Figure IV.15 : Interface finale de l'application

- 1: Icône de l'application
- 2: Connexion au Bluetooth
- 3: Déconnecté au Bluetooth
- 4: Angle de la pince
- 5: Angle de moteur gauche
- 6: Angle de la base
- 7: Angle de moteur droite
- 8: Position sauvegardé
- 9: Bouton Run
- 10 : Bouton sauvegardé
- 11: Bouton avant
- 12: Bouton Stop
- 13: Bouton gauche
- 14: Bouton arrière
- 15: Bouton droit
- 16: Etat du Bluetooth (connecté ou déconnecté)
- 17: Bouton Reset

IV.7.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.

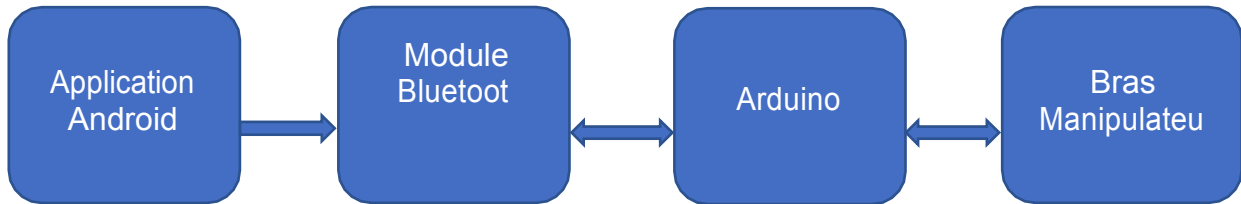


Figure IV.16 : Schéma synoptique simplifié du système

IV.8.Téléchargement de l'application

Après avoir fini de faire l'application nous avoir téléchargé Sur le smartphone, l'application MIT AI2 Companion doit être installée à télécharger sur le Play Store. Sur le PC, l'application MIT Emulator in AI2 doit être installée.

En choisissant build et click sur android app (.apk) : il montre nous un Code de 6 caractères est généré ainsi qu'un QR Code. Il Suffit de saisir le code ou de scanner le QR Code pour que l'application soit téléchargée sur le smartphone.

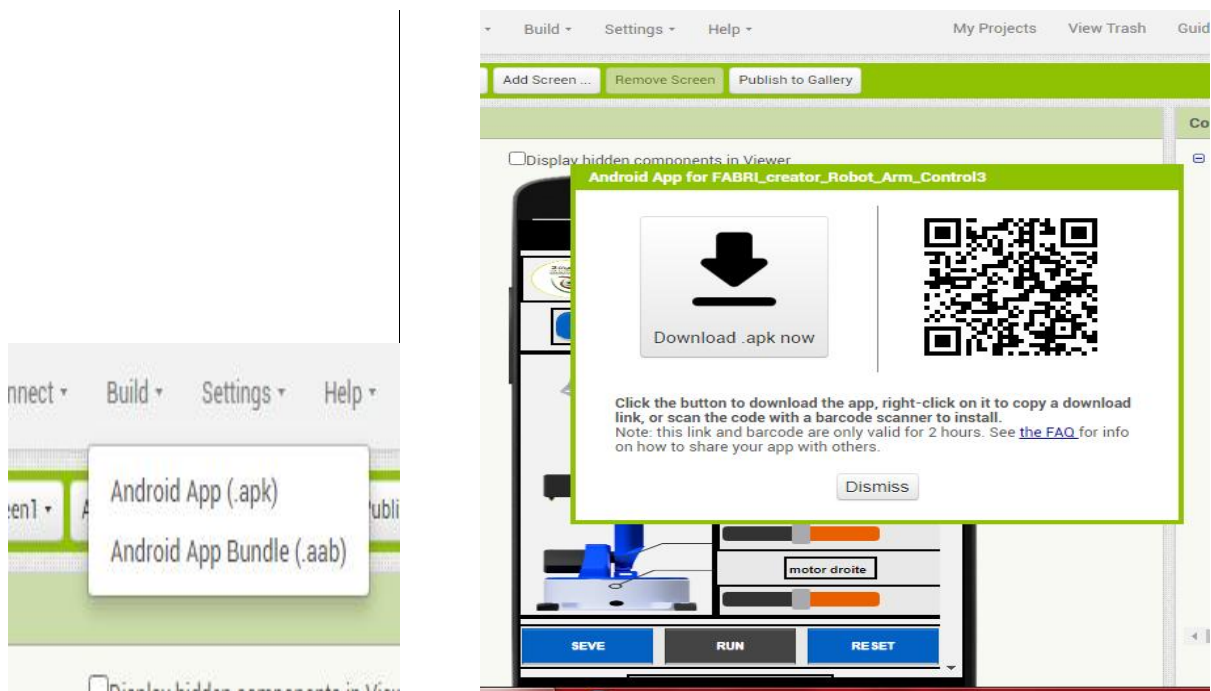


Figure IV.17 : téléchargement de l'application sur smartphone

IV.9.Conclusion

Dans ce chapitre nous avons présenté notre application constituée d'une seule interface importante, l'une commande le bras manipulateur et le robot mobile. et faire le cycle automatique grâce à l'appelle des trois boutons Save, Run, reset et faire la répétition des mouvements sauvegarde

Conclusion générale et perspectives

Conclusion générale et perspectives

Conclusion générale :

Un robot est une machine contrôlée équipée d'un ordinateur qui l'aide à effectuer des tâches préalablement programmées. Avec les nouveaux développements, le robot entre dans le monde des services à domicile, commerciaux et médicaux. Après être entré dans le monde de l'industrie lourde

Nous avons choisi ce projet, dans lequel nous sommes occupés de l'étude d'un bras manipulateur embarqué à un robot mobile commandé par une application Android. L'utilisation de ce bras est basée sur la technologie Bluetooth. Cette dernière est une technologie sûre et fiable et nous permet de communiquer à une distance allant jusqu'à 12 mètres. Il peut également répéter les tâches que nous lui avons demandées en continu après les avoir enregistrées.

C'était intéressant pour nous la réalisation de ce projet était intéressante pour nous car, elle nous a permis d'appliquer les connaissances acquises au cours de nos études universitaires, elle nous a amené à découvrir des nouveaux secteurs et nous a donné la chance d'avoir des connaissances pertinentes dans le domaine de programmation et de développement des applications Android.

Enfin, le prototype conçu et réalisé doit encore être amélioré : il faut poursuivre sa miniaturisation et son intégration dans un format plus pratique. Ces perfectionnements sont particulièrement importants en raison de l'influence de la praticité du système sur les expérimentations, et surtout sur sa future utilisation par les personnes malvoyantes et aveugles, qui devraient être les premières bénéficiaires.

Perspectives :

- Exploitation de l'internet des objets pour pouvoir télécommande le robot à diverses applications et plates-formes
- Ajoutez ou remplacez une batterie qui a une longue durée de vie
- L'ajout d'une caméra pour une détection visuelle précise.

Annexes

Annexe 1 :

❖ Les caractéristiques du servomoteur :

- Tension d'alimentation : 4.8 _ 16 V
- Courant de fonctionnement : -500 mA
- Longueur du câble : 180 mm
- Dimensions : 22 mm x 12.5 mm x 29.5 mm

Annexe 2 :

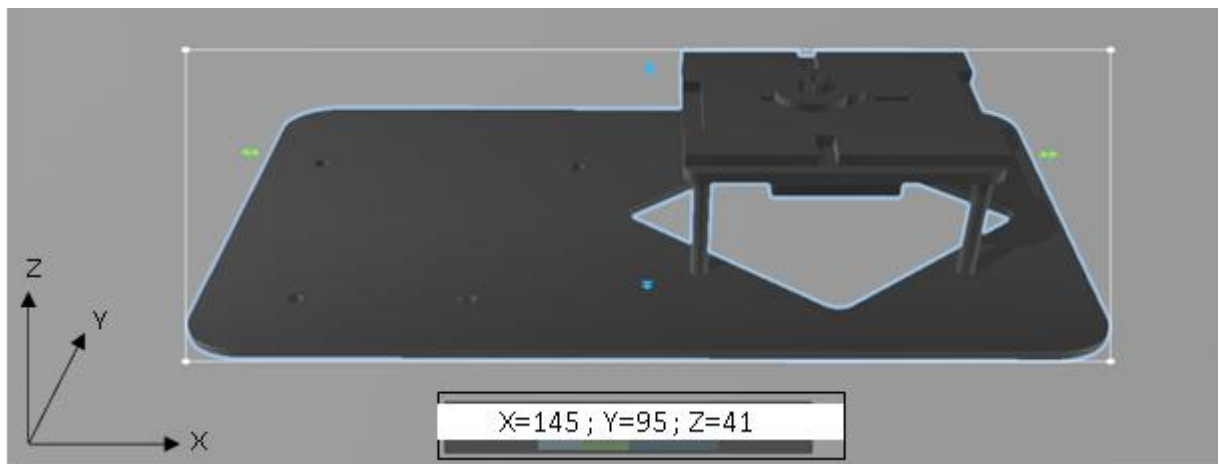


Fig.E.1. Dimensions de la base du bras manipulateur en millimètre

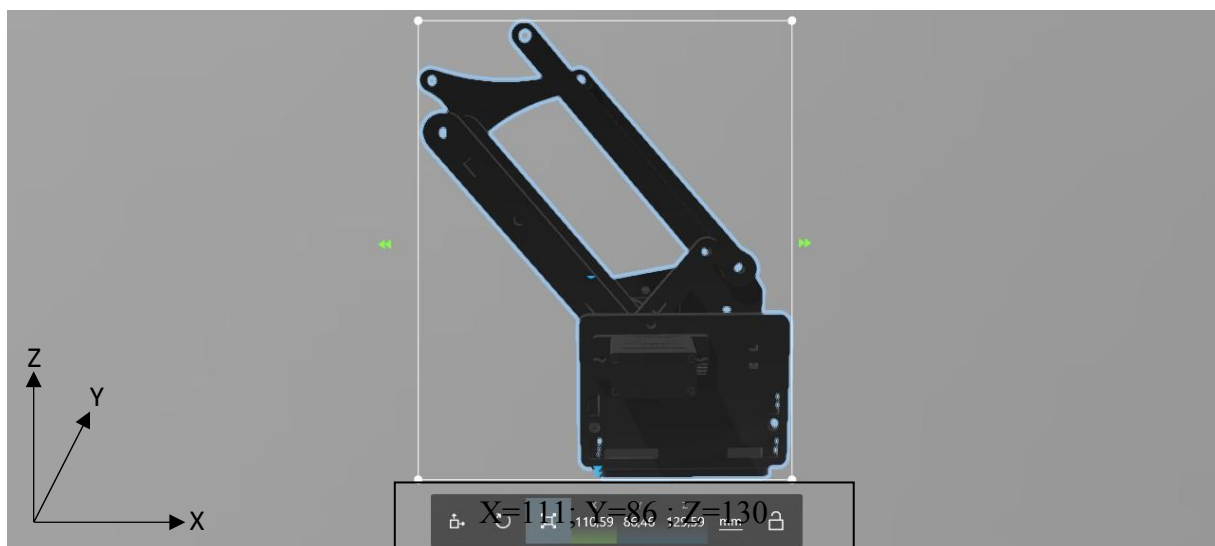


Fig.E.2. Dimensions du coude du bras manipulateur en millimètre

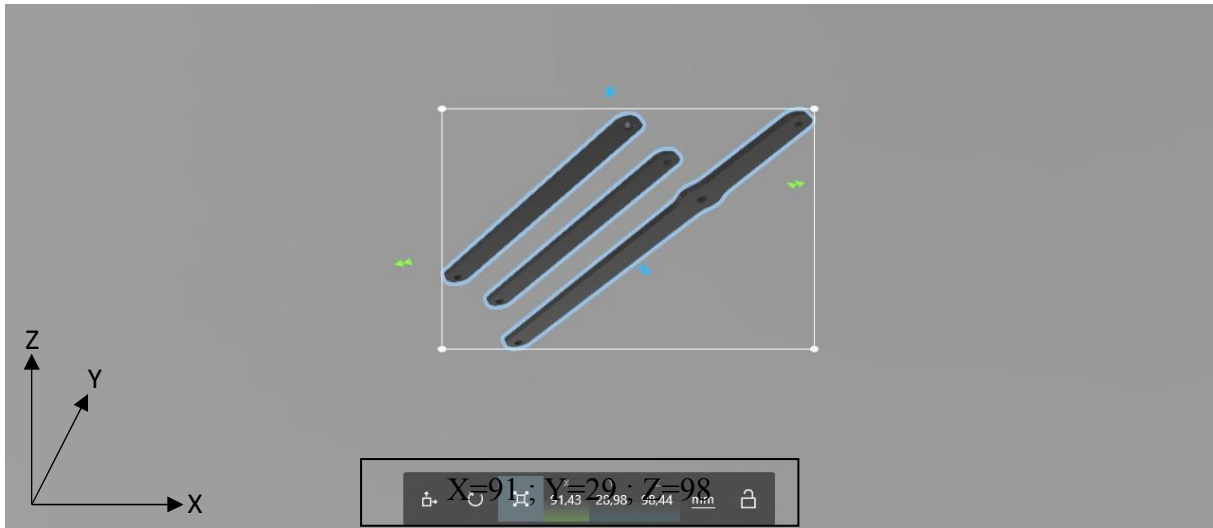


Fig.E.3. Dimensions de l'épaule du bras manipulateur en millimètre

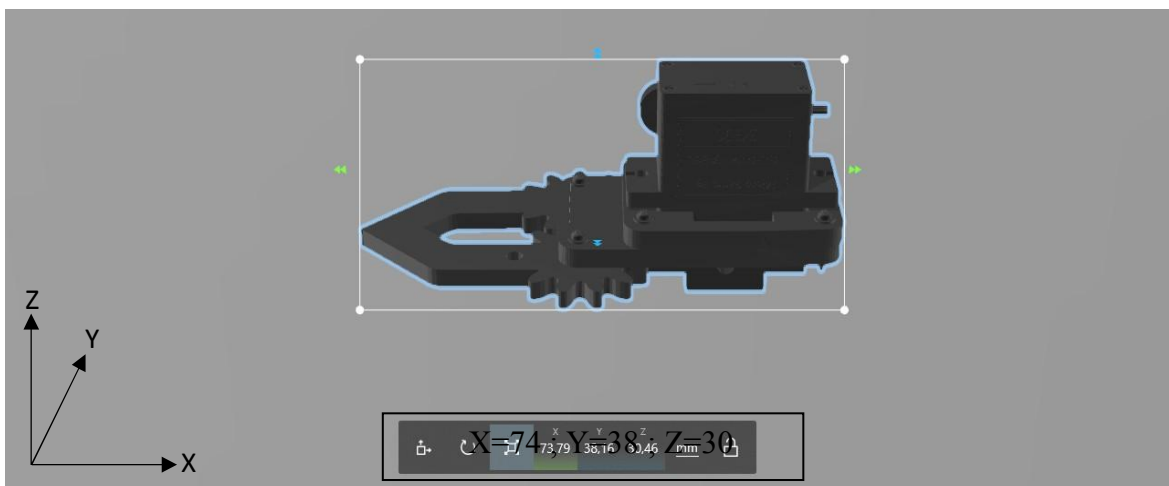


Fig.E.4. Dimensions de la pince du bras manipulateur en millimètre

Références bibliographiques :

- [1] Kahina, A. D., Meziane, A. Z., & Hocine, A. (2014). Conception et Réalisation d'un Bras Manipulateur Commandé par API. Université de Khemis Miliana, Algérie 2015. RICH A
- [2] Brahim et BOUYEKHF Mohamed Redah, Université de Biskra Algérie 2017
- [3] EYRAUD, Charles-Henri. « Horloges astronomiques au tournant du XVIIIe siècle : de la peu-près à la précision ». Thèse de doctorat. Lyon 2, 2004.
- [4] International Federation of Robotics (IFR). (2018). Robots and the Workplace of the Future.
- [5] EL HAMZAOUI Oussama « Localisation et Cartographie Simultanées pour un robot mobile équipé d'un laser à balayage : CoreSLAM », Thèse de Doctorat, École nationale supérieure des mines de Paris, 25 septembre 2012.
- [6] BOSTELMAN Roger « PERFORMANCE MEASUREMENT OF MOBILE MANIPULATORS », Thèse de Doctorat, UNIVERSITY OF BURGUNDY, 16 mars 2018.
- [7] SABRINA, HABOUSSI. Etude cinématique et géométrique d'un robot à 6 ddl. Université de Biskra Algérie 2016
- [8] MIGAN, A., SOGBOHOSSOU, M., HOUNGAN, K. T., KIKI, P., & MONTEIRO, L. (2021). Commande floue d'un robot mobile à roues suiveur de ligne.
- [9] BENSIKHALED SOFIANE, Hadiouche Youba. Réalisation et commande d'un bras robotique avec application sur smart phone. Université de Biskra Algérie 2016
- [10] BENSIKHALED SOFIANE, Hadiouche Youba. Réalisation et commande d'un bras robotique avec application sur smart phone. Université de Biskra Algérie 2016
- [11] <https://www.indevagroup.com/solutions/handling-reels-paper-plastic.>, 2019
- [12] <http://www.machine-outil.com/actualites/t157/a4676-robot-de-soudage-a-l-arc-kuka-tolexpo->, 2019
- [13] SMAHI, Mohammed Ismail, HADJILA, Fethellah, TIBERMACHINE, Chouki, *et al.* An encoder-decoder architecture for the prediction of web service QoS. In: European conference on service-oriented and cloud computing. Springer, Cham, 2018. p. 74-89.
- [14] Koubaa, H. Bennaceur, I. Chaari, S. Trigui, A. Ammar, M.-F. Sriti, M. Alajlan, O. Cheikhrouhou, and Y. Javed, Robot Path Planning and Cooperation Foundations, Algorithms and Experimentations, vol. XXII. 2018.
- [15] <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQIK-O88FG-1n2cyzU5XrNiUPNuhtgfn8oUog&usqp=CAU>
- [16] FERRERO F. « Moteurs et transistors MOS », Cours, Université de Nice, 2017.
- [17] DATASHEET « MG90S Metal Gear Servo », Components 101, 2019.
- [18] DATASHEET « SG90 Servo Motor », Components 101, 2017.
- [19] BETTAIEB Mohamed Nizar « Modélisation du comportement mécanique d'engrenages. Couplages entre engrenement et éléments de structures flexibles », Thèse de Doctorat, Ecole Doctorale Des Sciences Pour L'Ingénieur De Lyon, 16 juin 2006.
- [20] DATASHEET « HC-05 Bluetooth to Serial Port Module », Components 101, 20

- [21] LECHALUPÉ Julien <<Cours d'initiation à Arduino>> Université Paul Sabatier Mai 2014
- [22] B. Cottenceau B311 ISTIA< bertrand.cottenceau@univ-angers.fr<<Carte ARDUINO UNO Microcontrôleur ATmega328>>
- [23] <http://www.centreantipoisons.be/monoxyde-de-carbone/le-monoxyde-de-carbone-co-en-d-tail/quelles-sont-les-concentrations-toxiques-de>
- [24] https://fablab.univ-tlse3.fr/wiki/images/9/.../Cours_arduino_v0.2.pdf
- [25] <https://www.cours-gratuit.com/cours-android/utiliser-app-inventor-pour-programmer-une-application-android-tutoriel-pdf>
- [26] http://sig.fgranotier.info/IMG/pdf/debuter_app_inventor.pdf