REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE YAHIA FARES DE MEDEA



Faculté de Technologie

Mémoire de fin d'études de Master

Filière : Génie électrique

Spécialité : Automatique et systèmes

Conception et réalisation d'un bras Robot manipulateur piloté par application Android

---

Présenté par :

Sadoudi Abdelhamid

Douma Foudil

Proposé et Dirigé par

Dr. Battache Farouk

Promotion 2020

DEDICACE

Nous dédions ce modeste travail et nos profondes gratitudes à :

Nos mères, sources de tendresse et d'amours pour leurs soutiens tout le long de notre vie scolaire.

Nos pères, qui nous ont toujours soutenus et qui ont fait tout possible pour nous aider.

Nos frères et nos sœurs, que nous aimons beaucoup.

Notre grande famille.

Nos cher amis, et enseignants, administratif, et le personnel de la faculté de technologie de Médéa.

Tout qu'on collaboré de près ou de loin à l'élaboration de ce travail.

Que dieu leur accorde santé et prospérité.

Remerciements

Avant tout remerciement, louange à ALLAH

C'est avec le plus grand honneur que nous avons réservé l'ouverture de notre projet en signe de gratitude et de reconnaissance à l'égard de tous ceux qui nous ont aidés, de près ou de loin, à la réalisation de ce projet fin d'étude.

Nous tenons à adresser nos vifs remerciements à notre encadrant Monsieur. BattacheFarouk enseignant à l'université de Médéa pour sa présence, son encadrement, ses conseils fournis de façon efficace tout au long de la période de réalisation.

Nous voudrions aussi exprimer nos remerciements sincères à tous les professeurs de l'université de Médéa qui ont fait beaucoup d'efforts pour nous transmettre leurs connaissances. Vos compétences incontestables ainsi que vos qualités humaines vous valent l'admiration et le respect de tous. Nos sincères remerciements s'adressent aussi pour votre patience et votre encadrement durant toutes ces années.

Nous remercions le président pour avoir accepté de juger ce travail et les membres du jury qui ont pris de leurs temps pour lire et de nous avoir accordé l'honneur d'examiner, juger, et de valoriser notre travail.

Je remercier également Monsieur. Mahfoud Zouaouine.

Enfin, que nos parents, nos familles, nos enseignants et collègues à l'Université de Médéa, et tous ceux qui ont collaboré de près ou de loin à la réussite de ce travail trouvent à travers ces quelques lignes l'expression de notre profonde gratitude pour leur soutien et leurs encouragements de tous les instants.

On vous en remercie chaleureusement.

ملخص

يتضمن هذا العمل دراسة ل ذراع روبوت متكون من أربعة درجات من الحرية، ولتحقيق ذلك وجب علينا تقسيم العمل الى شطرين يتمثلان في دراسة نظرية وإدراك عملي.

الدراسة النظرية، يتم فيها عرض العاميات المتعلقة بأسس الروبوتات وخصائص مكونات الذراع المدروس.

ثم تناولنا النماذج التي تميزه (الهندسية، الحركية و الديناميكية).

الإدراك العملي، قمنا فيه بإجراء البرمجة استنادا إلى لغة البرمجة وتم تنفيذها على بطاقة أردوينو (لاسلكي) ومقياس الجهد الإنزلاقي مع إنجاز تطبيق أندرويد الذي يتحكم بالذراع عن بعد.

الكلمات المفتاحية :ذراع روبوت، اردوينو ، التطبيق أندرويد

RESUME

Ce projet de Master concerne la conception et la réalisation d'un bras manipulateur à quatre degrés de liberté à base d'une carte Arduino nano. Après une exploration générale du domaine de la robotique, on a abordé le vif du sujet en étudiant les différents organes constituant notre bras manipulateur. Par la suite, Enfin la commande du bras robot à travers l'application Android. Pour la réalisation pratique. La programmation est faite à base du langage C. Le contrôle du bras manipulateur se fait à l'aide d'une application mobile réalisée (sans fil) et grâce a le potentiomètres sur la carte de commande.

Mots-clés : Bras Robotique ; Arduino, Application Android,

ABSTRACT

This Master's project concerns the design and production of a manipulator arm with four degrees of freedom based on an Arduino nano board. After a general exploration of the field of robotics, we approached the heart of the matter by studying the different organs constituting our manipulator arm. Finally, The programming is done using the C language. The manipulator arm is controlled using a mobile application (wireless) and thanks to the the potentiometers.

Keywords: Robotic arm; Arduino, Android Application,

LISTE DES ABREVIATIONS

R.U.R : Rossum's Universal Robot

PUMA : Programmable Universal Machine for Assembly

SCARA : Selective Compliance Articulated Robot for Assemblage

ISO : Organisation international de normalisation

SMA : Système mécanique articulé

IDE : environnement de développement intégré

MIT2 : App-Inventor-2

AVR : Régulateur automatique de tension

V : Le volt (V) est l'unité de mesure de la tension électrique dans un circuit

MHz : Mégahertz

mA : Unité de mesure d'intensité du courant électrique du Système international (SI), valant 10 −3 ampère, et dont le symbole est mA

Kb : kilobyte

ICSP : In-Circuit Serial Programming

I2C : Circuit intégré inter

SPI : Interface périphérique série

PWM : Pulse With Modulation

USB : Universal Serial Bus

SRAM : Mémoire statique à accès aléatoire

EEPROM : Electrically Erasable Programmable Read-Only Memory

Sommaire

Introduction générale.....1

Chapitre I

Généralité s sur les robots industriels

I.1.Introduction :……...3

I.2.Définition générale :……3

I.2.1. Le robot industriel :……3

I.2.2 La robotique :……3

I.3 L'histoire de la robotique industrielle :……3

I.4 Types de robots :…….....5

I.4.1 Robots mobiles : ……5

I.4.2 Les robots manipulateurs :……...5

I.4.2.1 Robots cylindriques :……..6

I.4.2.2 Robots rectilignes : ……6

I.4.2.3 Robots sphériques : ...……6

I.4.2.4. Robots articulés : ……..6

I.4.2.5. Robots SCARA: ……6

I.5. Caractérisation d'un robot :……..8

I.5.1. Morphologie : ……8

I.5.2. Espace de travail : …….....8

I.5.3. Précision:……8

I.5.4. Respectabilité : ……..…8

I.5.5. Vitesses et accélérations max : ……9

I.5.6. Charge utile: ……..9

I.6.Domaines d'application des robots manipulateurs : ……...9

I.7. Constitution d'un robot manipulateur : ……..9

I.8.Classification des robots : ……10

I.8.1.Classification fonctionnelle : ……...14

I.8.1.1. Manipulateur à commande manuelle :……..14

I.8.1.2. Manipulateur automatique : ……..15

I.8.1.3. Robots programmables :……15
I.8.1.4. Robots intelligents : ……16
I.8.2. Classification géométrique : ……...16
I.9.Objectifs de la robotisation : ……...17
I.10. Conclusion : ……18

Chapitre II

La Description des parties matériel et logiciel

II.1 Introduction:……....19
II.2Partie matériel:..……19
II.2.1L'arduino: ……..19
II.2.1.1 Définition: ……...……19
II.2.1.2 Arduino Nano: ……...……19
II.2.2 Module BluetoothHC-05: ……21
II.2.3 Capteur de couleurTCS3200: ……22
II.2.4 Les servomoteurs: ……....……25
II.2.4.1 Définition : ……25
II.2.4.2Fonctionnement des servos : ……25
II.3PartieLogiciel: ……..26
II.3.1 ArduinoIDE: ……26
II.3.2 Proteus professionnel : ……27
II.3.3Fritzing : ……28
II.4 ApplicationAndroid : ……..…28
II.4.1 MIT App Inventor 2 : ……..28
II.4.2 Environnement de travail : ……..……30
II.5 Conclusion : ……31

Chapitre III

La Description des parties matériel et logiciel

III.1 Introduction : ……...32
III.2 Un bref aperçu de notre bras manipulateur : ……32
III.3 Partie mécanique: ……33
III.3.1 conception du robot: ……..34
III.3.2 Fabrication des pièces: ……..35
III.3.3 Le montage des pièces: ……..37
III.4 La carte de commande : ……..40
III.4.1 Réalisation de la carte de commande: ……40
III.5 Partie Commande: ……43
III.5.1 Commande par application Android : ……43
III.5.1.1L'écran d'accueil : ……...45
III.5.1.2 L'écran de connexion: ……46
III.5.1.3 L'écran de Fonctionnement: ……..48
III.5.2 Commande par PC: ……..57
III.5.3 Commande par un potentiomètre : ……..58
III.6 Fonctionnement et résultats : ……...59
III.7 Conclusion :……64

LISTE DES FIGURES

Chapitre I

Figure 1 Robotique industrielle (unimate, puma) …….2
Figure 2.robot mobile koala 2.5 …….3
Figure 3.L'espace de travail de robot cylindrique ……4
Figure 4.L'espace de travail de robot rectiligne ……5
Figure 5.L'espace de travail de robot sphérique ……...5
Figure 6 L'espace de travail de robot articulé ……...5
Figure 7L'espace de travail de robot SCARA ……..6
Figure 8Constituant d'un bras manipulateur ……...9
Figure 9 exemple de SMA ……11
Figure10 Principaux types d'articulations et leur représentation conventionnelle …….12
Figure 11Manipulateur à commande manuelle ……..15
Figure 12Manipulateur à cycle préréglé ……..15
Figure 13Robot programmable ……16
Figure 14Robot intelligent ……16

Chapitre II

Figure 15 Structure de la carte Arduino nano ……20
Figure 16 module Bluetooth HC-05 ……..22
Figure 17 Capteur de couleur TCS3200 ……..23
Figure 18 Organisation de la Matrice des photos détectrices ……24
Figure 19 Structure interne du capteur TCS3200 ……..24
Figure 20 Structure d'un servomoteur ……25
Figure 21 Logo du logiciel IDE Arduino …….....26
Figure 22 fenêtre de logiciel IDE ……..27
Figure 23 Logo de PROTEUS professionnel ……...28
Figure 24 logo de Fritzing ……28
Figure 25 Logo de MIT App inventor2 ……29
Figure 26 schéma synoptique de la réalisation d'une application ……29
Figure 27 l'espace de travail du concepteur ……....30
Figure 28 L'espace de travail des blocs ……31

Chapitre III

Figure 29 Organigramme de commande du bras……33
Figure 30 Les dimensions des pièces…….34
Figure 31 Dessin du bras assemblé……35
Figure 32 La base……35
Figure 33 L'épaule……..35
Figure 34 montage de l'avant épaule. ……...36
Figure 35 Le coude……..36
Figure 36 Pince……36
Figure 37 Panier de triage par couleur…….36
Figure 38 Matrice de triage Matriciel……...36
Figure 39 position de ramassage. …….37
Figure 40 Support de capteur de couleur. ……...37
Figure 41 Les pièces du bras final. ……...37
Figure 42 montage de base avec l'avant épaule et deux servomoteurs…….38
Figure 43 Montage de l'épaule avec un servomoteur. ……38
Figure 44 Montage de coude avec la pince et un servomoteur……...38
Figure 45 Montage de l'épaule avec le coude. …….39
Figure 46 Le montage de base avec l'épaule et le coude. ……39
Figure 47 Montage Finale du bras…….39
Figure 48 Schéma électrique de la carte de commande……...40
Figure 49 Vue depuis l'ISIS……41
Figure 50 Vue depuis l'ARES……42
Figure 51 Vue 3D du circuit imprimer sur Proteus……43
Figure 52 La carte de commande après l'impression……..43
Figure 53 La carte de commande après le soudage (final)……..43
Figure 54 schéma synoptique de réalisation d'une carte de commande……43
Figure 55 La communication entre l'application Android et la carte Arduino……44
Figure 56 La communication entre le module Bluetooth HC-05 et la carte Arduino…..44
Figure 57 L'écran d'accueil……45
Figure 58 le bloc de démarrage d'application……..46
Figure 59 Les blocs de l'écran d'accueil……...46
Figure 60 L'écran de connexion ……...47
Figure 61 les blocs du bouton «connecter»……...47

Figure 62 le bloc du notifiant…….48
Figure 63 les blocs des fonctions «connecté» et «déconnecté»……48
Figure 64 les blocs du bouton «déconnecté»…….49
Figure 65 L'écran de "Fonction"……...49
Figure 66 Les Boutons d'écran de Fonction…….50
Figure 67 L'écran de triage par couleur……...51
Figure 68 Les blocs des boutons «ouvrir» et «fermer»……52
Figure 69 Les blocs des glissières……...52
Figure 70 le bloc du bouton «record»……53
Figure 71 le bloc du bouton «StartRun»……...54
Figure 72 Le bloc du bouton «Stop»……..55
Figure 73 logigramme de fonctionnement de triage par couleur……...56
Figure 74 L'écran de triage par les angles……56
Figure 75 Les blocs des boutons de triage par les angles……57
Figure 76 logigramme de fonctionnement de triage par les angles……58
Figure 77 Les axes mécaniques de rotation……..61
Figure 78 : L'environnement de robot……..61
Figure 79Test du bras manipulateur pour trier par couleur……..63
Figure 80 Test du bras manipulateur pour trier par les angles des organes…….64

LISTE DES TABLEAUX

Chapitre II

Taleau 1 les caractéristiques de carte Arduino nano……..……..21

Tableau 2 Caractéristique du module Bluetooth HC-05….……22

Tableau 3.Les fréquences de sortie Tableau…….……24

Tableau4.Photo détecteur….……..24

Chapitre III

Tableau 5.Les angles max et min de chaque servomoteur…….61

Introduction Générale

Aujourd'hui, il existe plusieurs types de robots conçus pour des tâches bien spécifiques. Dans ce travail de mémoire, nous nous sommes intéressés aux robots manipulateurs pour des applications en robotique industrielle.

Utilisés dans l'industrie depuis maintenant plusieurs dizaines d'années, les robots industriels actuellement en service sont des robots de type manipulateur. Ils sont bien implantés dans les procédés de fabrication modernes et sont employés pour augmenter le volume de production et améliorer la qualité du produit. Dans les chaînes de montages de l'industrie automobile, ils remplacent les ouvriers dans les tâches pénibles, répétitives ou dangereuses (peinture, soudure…).

Le bras manipulateur est un système de positionnement ou les forces agissant au niveau des articulations sont produites par des actionneurs. Ces derniers peuvent être de types électrique, hydraulique ou pneumatique. Dans notre mémoire nous intéressons particulièrement au cas des servomoteurs

L'objectif principal de notre travail consiste en l'étude théorique, ainsi qu'en la réalisation pratique d'un bras robot manipulateur à quatre degrés de liberté, en utilisant des servomoteurs permettant à ce dernier d'avoir des déplacements d'une grande précision. Après la réalisation, on a procédé à la programmation, notamment la configuration du Bluetooth et l'implémentation d'une commande à distance avec Smartphone à travers une application Android. [1]

Pour mieux présenter notre travail, le mémoire est organisé en trois chapitres distincts. Le premier chapitre donne un aperçu sur les robots. Nous définissons le robot industriel et la robotique et nous donnons un bref historique sur la robotique industrielle et les différents types de robots. Ce chapitre analyse aussi les différents constituants d'un robot manipulateur en précisant leurs domaines d'utilisation.

Le deuxième chapitre nous avons donné une explication de la carte Arduino, le module Bluetooth, les servomoteurs et son principe de fonctionnement. Ensuite, Nous avons fourni une explication simple de la partie software (Proteus,Fritzing),et nous avons parlé en détail sur logiciel Arduino IDE et App Inventor 2 parce que l'une des pièces de base de cette recherche.

Le troisième chapitre on va voir la présentation mécanique du bras manipulateur qui traite de la conception à la construction des pièces et le montage, aussi la réalisation

de la carte de commande. Notamment, nous avons montré de différentes manières de commande du bras manipulateur.

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.

I.1.Introduction

L'utilisation des systèmes robotiques apparait aujourd'hui dans plusieurs domaines d'activités : la médecine, la défense, la recherche scientifique etc.… Les robots sont utilisés de manière privilégiée pour des missions où les objectifs sont quantifiables et clairement définis. Ils sont destinés à faciliter les tâches pour l'homme et à amplifier le rendement.

Dans ce premier chapitre, on va donner un aperçu non exhaustif sur les robots, un Bref historique sur l'évolution de la robotique industrielle et présenter les différents types des robots et les éléments constitutifs de ces derniers.

I.2.Définition générale :

I.2.1. Le robot industriel :

Le terme robot a été introduit pour la première fois par l'auteur Tchèque Capek en 1920 dans sa pièce de théâtre R.U.R. (Rossum's Universal Robot) et il est dérivé du mot Robota qui signifie travailleur (de force). [2]

Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) accomplissant automatiquement des tâches diverses. C'est une machine intelligente fonctionnelle qui nécessite une autonomie de mouvements.

L'Organisation Internationale de Normalisation définit le robot comme étant 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. [3]

I.2.2 La robotique :

La robotique est une science qui s'intéresse aux robots. En fait, il s'agit d'un ensemble de disciplines techniques (mécanique, électronique, automatique, informatique) articulées autour d'un objectif et d'un objet communs. Cet objectif est l'automatisation flexible de nombreux secteurs de l'activité humaine réputés jusqu'à très récemment comme ne pouvant se passer de la présence de l'homme, et l'objet est le robot, sorte de machine universelle dont l'homme rêve depuis toujours pour le remplacer dans les tâches difficiles. [4]

I.3.L'histoire de la robotique industrielle :

La robotique industrielle a connu un essore entre 1950-1970. Elle a vu le jour en 1954lorsque Georges DEVOL a pu réaliser son brevet sur la robotique. Dans ce brevet Devol a conçu un robot qu'il a intitulé Unimate.

En 1961, le premier Unimate fut utilisé dans les usines de GENERAL MOTORS.

En 1966 , l'entreprise Unimation continue de développer des robots et élaborent notamment des robots permettant de faire d'autres tâches, comme des robots de manipulation matérielle ou encore des robots conçus pour la soudure ou pour d'autres applications de ce genre.

En 1978 un nouveau robot est conçu par Unimation Inc avec l'aide de General Motors. Ensemble ils conçurent le robot PUMA 500. Le robot PUMA (Programmable Universal Machine for Assembly) a été conçu par Vic Schienman et fut financé par General Motors et par The Massachussetts Institute of Technology au milieu des années 70. Le système de ce robot est composé d'un bras manipulateur permettant d'assembler des composants industriels et de son ordinateur de commande. Ce robot est le robot d'assemblage le plus rependu dans l'industrie des années 70.

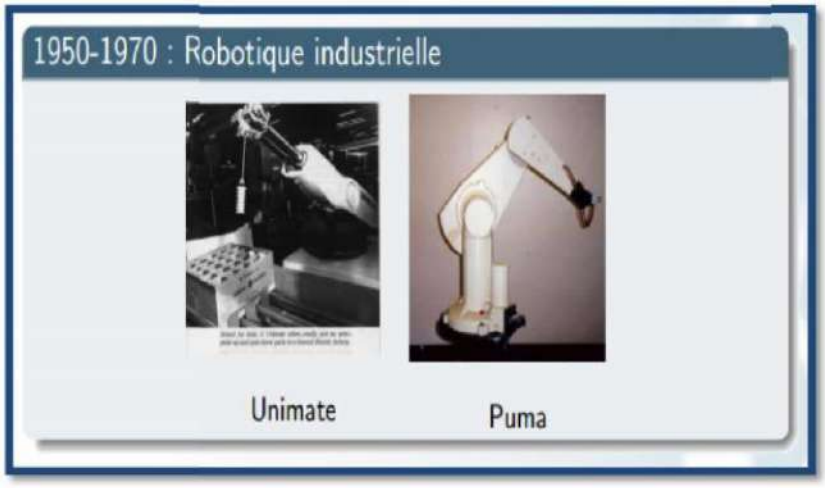


Figure 1 Robotique industrielle (unimate, puma)

En 1985, Reymond Clavel a imaginé le Robot Delta qui possède un bras de manipulation formé de 3 parallélogrammes. Son brevet tombe dans le domaine public en 2007 et différents constructeurs devraient alors sortir leur propre robot delta.

Le Jet Propulsion Laboratoire (JPL) développe un robot industriel hexapode (à 6 pattes) du nom de **Lemur**. Lemur aura pour mission de monter, assembler et réparer des installations spatiales. Pesant moins de 5 kg, il offre la possibilité innovante d'adapter différents outils sur chacun de ses membres. Selon l'étude robotique de la Fédération Internationale de Robotique (IFR) en 2012, il y a aumoins 1 153 000 robots industriels opérationnels fin 2011 dans le monde. Avec l'apparition de la robotique industrielle, les robots étaient conçus pour remplacer les ouvriers dans les tâches pénibles, répétitives ou dangereuses (peinture, soudure…). Aujourd'hui avec le développement de l'électronique, de l'informatique, de la mécanique et aussi de l'automatique, la technologie robotique a progressé. La recherche dans le domaine de la robotique est dirigée vers le développement de

robots dévoués à des tâches bien différentes que celles demandées par l'industrie. Par exemple des robots travaillant en mode automatique ou semi-automatique et qui ont souvent pour objectif d'interagir avec des humains et de les aider dans leurs tâches (surveillance, manutention d'objets lourds…). Ils sont dotés d'une intelligence qui leur donne une certaine autonomie.

Ainsi donc, le développement important de l'intelligence artificielle et de la robotique font que de nouveaux robots apparaissent constamment et l'utilisation de systèmes robotiques apparait aujourd'hui dans plusieurs domaines d'activité : la médecine, la défense, la recherche etc.… [5]

I.4. Types de robots :

Les types des robots : Il existe deux grandes familles de robots sont :

· Les robots mobiles.

· Les robots manipulateurs. [6]

I.4.1. Robots mobiles :

Ce sont des robots capables de se déplacer dans un environnement. Ils sont équipés ou non suivant leur utilisation, (les robots explorateurs, les robots de services….).



Figure 2.robot mobile koala 2.5 [7]

I.4.2. Les robots manipulateurs :

Un robot manipulateur est en forme d'un bras et se composent d'un certain nombre de segments qui est conçu pour manipuler ou déplacer des matériaux, outils et pièces sans contact humain direct. Ils sont des dispositifs qui permettent aux humains d'interagir avec des objets dans un environnement en toute sécurité. Les robots manipulateurs sont utilisés dans des applications industrielles pour s'effectuer efficacement des tâches telles que l'assemblage, soudage, traitement de surface, et le forage.

Les types des robots manipulateurs : les robots manipulateurs viennent sous plusieurs formes. Les formes se répartissent en en cinq grandes catégories :

- Robots cylindriques
- Robots rectilignes
- Robots sphériques
- Robots articulés
- Robots SCARA

I.4.2.1. Robots cylindriques :

Le robot cylindrique a deux axes de mouvement, un pour le mouvement en haut et bas. La rotation se fait par la jonction à la base. De plus, le bras horizontal peut se déplacer à l'intérieur et à l'extérieur, ce qui donne un troisième axe de mouvement limitée.

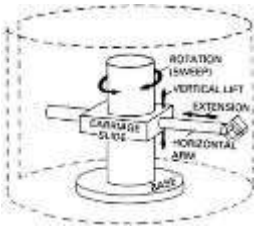


Figure 3.L'espace de travail de robot cylindrique

I.4.2.2. Robots rectilignes :

Les robots rectilignes a trois axes de mouvement (x, y, z). Pour cette raison, le robot rectiligne est parfois appelé Robot cartésien. Ces robots sont exploités par vérin pneumatique.

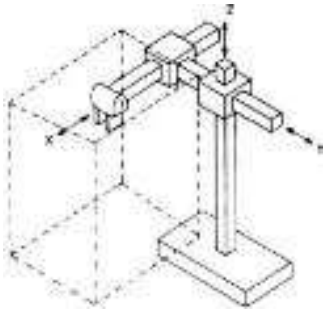


Figure 4.L'espace de travail de robot rectiligne

I.4.2.3. Robots sphériques :

Le robot sphérique est de grande taille avec un bras télescopique qui assure un mouvement à l'intérieur ou à l'extérieur. Les mouvements de base du robot sphérique sont de rotation (à la base) et angulairement en haut ou en bas (sur le bras).

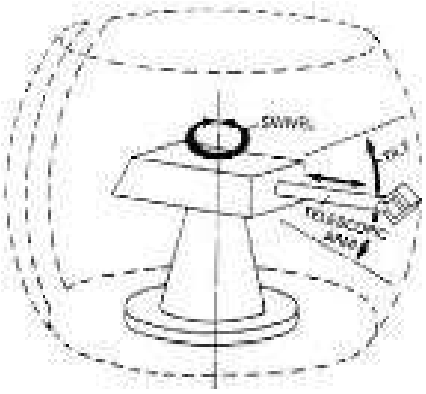


Figure 5.L'espace de travail de robot sphérique

I.4.2.4. Robots articulés :

Le bras articulé du robot ressemble à un bras humain. Il se compose de deux éléments, nommés l'avant-bras et le bras supérieur. Ce type de robot n'a généralement pas besoin d'un lieu séparé.

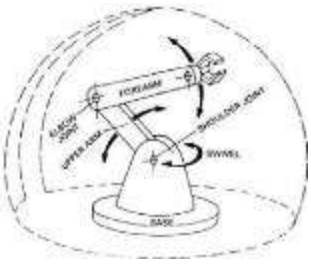


Figure 6 L'espace de travail de robot articulé

I.4.2.5. Robots SCARA:

Un robot SCARA est défini dans la norme ISO 8373: 1994, No.3.15.6, en tant que «robot comporte deux liaisons pivots parallèles pour fournir conformément à un plan sélectionné» .et peut être considéré comme un cas particulier d'un robot cylindrique. Le terme «SCARA» signifie «Bras de conformité sélective pour l'assemblage du robot». (Autre interprétation comprend «Conformité sélective articulée pour l'assemblage du robot».).

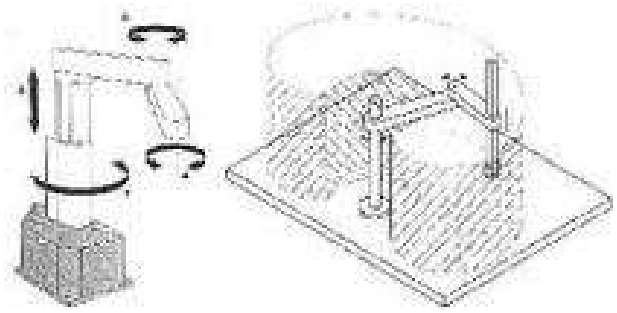


Figure 7L'espace de travail de robot SCARA

I.5. Caractérisation d'un robot :

L'intérêt d'une caractérisation est de mettre a la disposition des différents intervenants dans le domaine de la robotique, comparaison efficace des performances de plusieurs manipulateurs et de leur permettre ainsi d'effectuer le meilleur choix possible du robot.

Cependant, la versatilité (flexibilité d'utilisation), l'auto adaptabilité a l'environnement (Capacité d'initiative) ou encore la complexité technologique des robots rend leur caractérisation très difficile, le but essentiel est de caractéristiques permettent de réaliser cet objectif :

I.5.1. Morphologie :

Morphologie et est basé sur une combinaison de deux paramètres, le type de joints tournants ou prismatique et les angles que font les deux axes des applications consécutives (0 ou 90 °).

I.5.2. Espace de travail :

Appelé aussi volume accessible, c'est l'ensemble des points aux quels peut accéder l'organe terminal du robot.

I.5.3. Précision:

Les imprécisions (> 1 mm) sur le positionnement absolu sont du essentiellement aux erreurs dans le modèle géométrique, aux erreurs de quantification de la mesure de position ou encore aux flexibilités.

I.5.4. Respectabilité :

C'est l'erreur maximale de positionnement répète de l'outil en tout point de son espace de travail, elle est en général inférieure à 1 mm

I.5.5. Vitesses et accélérations max :

Elles dépendent de la qualité des actionneurs utilisés il existe deux vitesses : la vitesse maximale de translation ou de rotation de chaque axe, les constructeurs donnent souvent une

vitesse maximale de l'organe terminal. L'accélération maximale, est donnée pour chaque axe dans sa configuration la plus défavorable (charge maximale, inertie maximale), elle dépend fortement de l'inertie et donc de la position du robot.

I.5.6. Charge utile:

La charge utile est la charge maximale que peut porter le robot sans dégrader sa respectabilité, Elle est diffèrent de la charge maximale que peut porter le robot qui dépend directement des actionneurs, les méthodes et les essais à effectuer afin de déterminer ces caractéristiques sont décrits de manière détaillée dans des normes spécifiques. Pour notre, nous ne pouvons dans le cadre de ce travail faire une caractérisation du robot, d'une part, ceci n'est pas l'objet du projet, d'autre part, une caractérisation nécessite l'achat de normes et l'utilisation d'appareils de mesure Perfectionnés et dont nous ne disposons malheureusement pas.

I.6.Domaines d'application des robots manipulateurs :

La robotique permet de réaliser des tâches répétitives, pénible, dangereuse et dans des milieux hostiles, avec précision et fiabilité, dans tout environnement, a des cadences élevées et continue, tout en palliant au manque de main-d'œuvre, favorisant la haute productivité, et donnant ainsi à la robotique un fort intérêt humain, technique et économique. Ce qui lui a permis de se diversifier dans de nombreux domaines d'utilisation qui sont les suivants.

- Domaines industriels :
  - La transformation de pièces : moulage, usinage, perçage, rectification, enlèvement de matière
  - La préparation de surface : peinture, enduction, projection thermique/plasma, Grenaillage
  - L'assemblage : rivetage, vissage, soudage, insertion, collage
  - Les manutentions : manipulation de pièces, chargement/déchargement de machine, positionnement
  - La distribution
  - La palettisation et le conditionnement
  - L'inspection et le test
  - Le nettoyage
- Domaine médical :
  - Opérations
  - Diagnostique

- Radiothérapie
- Prothèses

· Domaine militaire :

- Désamorçage
- Déminage

· Autres domaines :

- Nucléaire

  Manipulation des produits radioactifs

- Spatial :

  Exploration des planètes

- Sous-marin :

  Forage

- Musique :

  Chef d'orchestre

- Dessin :

  Robographe (imitation de gestes humains)

I.7. Constitution d'un robot manipulateur :

La robotique est un domaine d'intégration ou interviennent plusieurs disciplines différentes : la mécanique, les actionneurs (électromagnétiques, hydrauliques, pneumatiques, etc.), les capteurs (position, vitesse, force, vision, etc.), l'électronique, l'automatique, l'informatique, la robotique « pure » (génération de trajectoires, planification, placement d'un robot, manipulabilité, etc.)., tous ces domaines jouent un rôle fondamental dans la constitution et la commande d'un robot manipulateur .

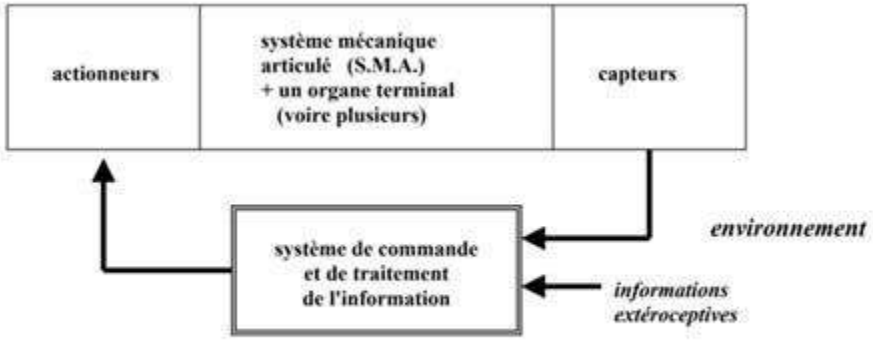


Figure 8Constituant d'un bras manipulateur

Les différents éléments d'un robot industriel sont :

-Une structure mécanique qui supporte l'organe terminale à situer.

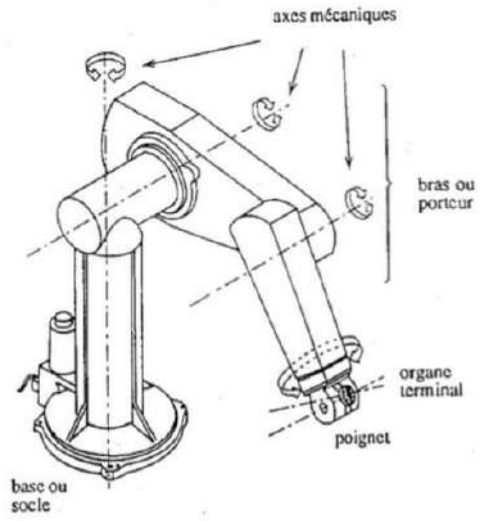


Figure 9 exemple de SMA

- Base : support sur lequel est situé le bras
- Porteur (bras) : chaine constituée d'élément de translation et/ou rotation
- Poignet : ensemble mécanique de l'extrémité du bras d'un robot
- Liaison (articulation) :
  - Rotoïde (ou pivot)
  - Prismatique (ou glissière)
  - Rotule S (sphérique)
  - Cardan U (joint universel)

Liaison active : liaison motorisée entre deux corps successifs d'un robot

Liaison passive : liaison non motorisée

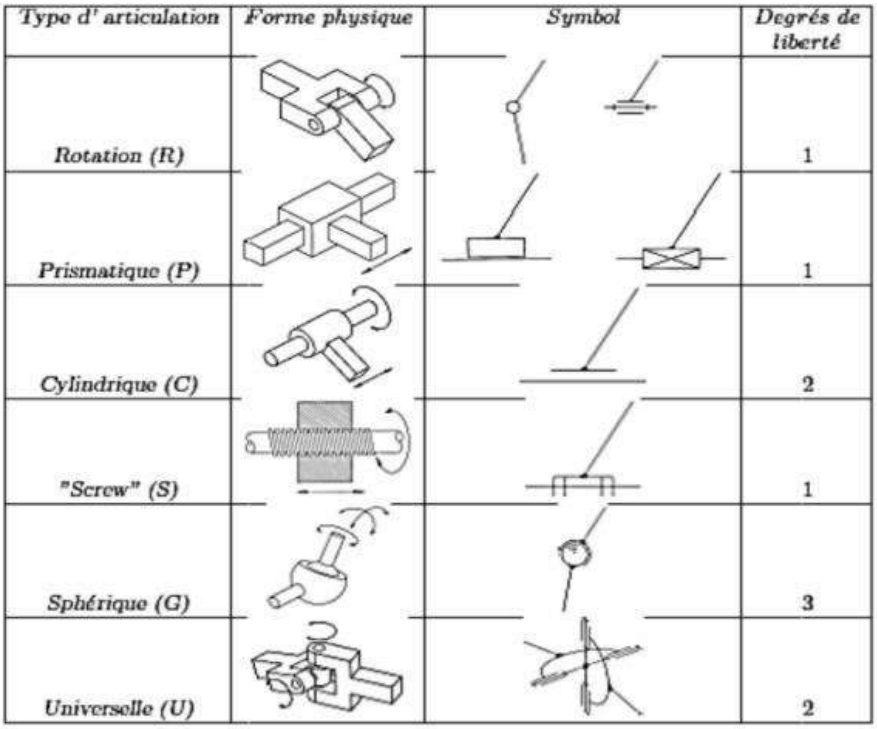


Figure10 Principaux types d'articulations et leur représentation conventionnelle.

-Des actionneurs qui servent à agir sur la structure mécanique pour changer la situation de l'organe terminal. (Ensemble de moteurs et de transmissions)

- Moteurs électriques
  - A aimant permanent
  - A courant continue
  - A commande par l'induit
  - A commutation électronique (sans balais)
  - Pas à pas
  - Servomoteur (Le terme « servo » induit un asservissement)
- Actionneur hydraulique
  - Vérin hydraulique pour la translation
  - Moteur hydraulique pour la rotation
- Actionneurs pneumatiques
  - Usage général pour le manipulateur a cycles (tout ou rien)

-Des capteurs divers nécessaires à la commande.

- Capteurs proprioceptifs : mesurent l'état interne du robot
  - Position
    - Linéaire : potentiométrique

- Angulaire : potentiométrique, capteur de contact, capteur magnétique, capteur optique incrémentaux et absolue

Vitesse

- Linéaire : tachymètres linéaires, tachymètres linéaires à ondes ;
- Angulaire : Tachymètres à impulsion, Génératrices tachymétriques, Gyromètres

Accélération

- Accéléromètre
- Gyroscope
- Magnétomètre

- Capteurs extéroceptifs : recueillent des informations sur l'environnement

Capteur de contact

- Détection de collision, température, texture, relief, forme, détection de dommages

Capteur sans contact

- Capteur de proximité
- Capteur d'image
- Caméras standards
- Caméras stéréoscopiques
- Caméras panoramiques

-Un système de commande qui pilote les actionneurs du robot manipulateur.

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

S'ajoutent à cela :

- L'interface homme-machine à travers laquelle l'utilisateur programme les tâches que le robot doit exécuter
- Le poste de travail, ou l'environnement dans lequel évolue le robot.

-Un système décisionnel qui assure la fonction de raisonnement et élabore le mouvement du robot manipulateur. (La programmation)

- Par apprentissage : les trajectoires et actions des robots sont apprises manuellement pendant d'apprentissage

- Hors-ligne (PHL) : un logiciel permet de programmer et de simuler le fonctionnement du robot dans la cellule virtuelle sur un ordinateur. Les programmes sont ensuite transférés sur la baie de commande

-Un système de communication qui gère le message transmis entre le système décisionnel et l'opérateur via une console de visualisation

O A distance : l'homme et le robot sont séparé spatialement ou temporellement

La télé opération ou le contrôle supervisé d'un robot placé à distance

La télémanipulation lorsqu'un robot manipulateur est utilisé pour réaliser des tâches de manipulation à distance

O Proche : l'homme et le robot sont colocalisé

L'interaction proche avec un robot mobile pouvant servir d'assistant

Interaction sociale, émotive ou cognitive où l'homme et le robot sont le plus souvent proches

O Interaction physique : la robotique collaborative

-Les moyens d'interaction :

De L'homme vers le robot : clavier, souris, bouton, joystick, capteur de mouvement, capteur de proximité

Du robot vers l'homme : écran, voyant, alarme

Dans les deux sensé : écran tactile, joystick a retour de force, dispositif haptique.

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 :

Le nombre de classe et les distinctions entre celles-ci varient de pays à pays (6 classes au Japon, 4 en France). L'A.F.R.I. distingue 4 classes illustrées ci-dessous :

I.8.1.1. Manipulateur à commande manuelle :

La Figure I.10 représente les manipulateurs à commande manuelle :



Figure 11Manipulateur à commande manuelle

I.8.1.2. Manipulateur automatique :

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



Figure 12Manipulateur à cycle préréglé

I.8.1.3. Robots programmables :

Ils répètent les mouvements qu'on leur a appris ou programmés sans informations surl'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 êtreprogrammés hors-ligne.

Pour certains robots, par exemple les robots de peinture, qui doivent suivre une trajectoire complexe qu'il est difficile d'exprimer mathématiquement, un opérateur humainspécialiste de la tâche effectue la trajectoire en guidant le bras du robot à l'aide d'un « pantin», et l'entièreté de la trajectoire est mémorisée par le robot et la figure I.12 représente lesrobots programmables. [4]



Figure 13Robot programmable

I.8.1.4. 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 deproximité, capteurs d'efforts,...) comme le montre la Figure I.13. Les robots de troisièmegénération sont capables de comprendre un langage oral proche du langage naturel et de sedébrouiller de façon autonome dans un environnement complexe grâce à l'utilisation del'intelligence artificielle. [4]

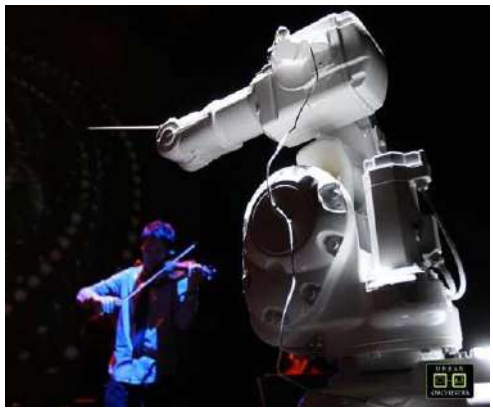


Figure 14Robot intelligent

I.8.2. Classification géométrique :

On peut aussi classer les robots suivant leur configuration géométrique, autrement dit l'architecture de leur porteur.

La structure cartésienne (PPP) :

C'est une structure à trois liaisons prismatiques et est la plus ancienne. Historiquement, elle découle logiquement de la conception traditionnelle d'une machine-outil à trois axes, type rectifieuse ou fraiseuse par exemple. Cette structure est relativement peututilisée sauf dans quelques applications particulières telles que robots pratiques, robots demagasinage.

La structure cylindrique (RPP) ou (PRP) :

Cette structure associe une rotation et deux translations. Elle présente l'inconvénient d'offrir un volume de travail faible devant un encombrement total important. Elle n'est pratiquement plus utilisée.

La structure sphérique ou polaire

C'est une structure quasiment abandonnée pour des raisons similaires à l'abandon dela structure cylindrique.

La structure dite SCARA :

A axes de rotation parallèles, elle est l'une des plus utilisées en particulier pour destâches de manutention ou d'assemblages très fréquents dans l'industrie. Ce succès commercial est lié au fait que le ratio entre le volume de travail et l'encombrement est très favorable et aussi au fait que la structure SCARA est très adaptée à ce type de tâches.

La structure 3R:

Elle permet d'amener un solide en un point de l'espace par trois rotations, généralement une à axe vertical et deux à axes horizontaux et parallèles. C'est le porteur généraliste par excellence pouvant se programmer facilement pour différents types de tâcheset disposant d'un volume de travail conséquent.

I.9.Objectifs de la robotisation :

Parmi les objectifs de la robotisation on peut citer :

- Adaptation a de l'environnement hostiles pour l'homme.
- Adaptation a des taches physiques ou intellectuelles pénibles pour l'homme.
- Augmentation de la sécurité.
- Adaptation a des contextes particuliers.
- Amélioration de la qualité du produit grâce à une meilleure répétitivité de la valeur ajoutée
- Amélioration de la flexibilité du produit

▪

I.10. Conclusion :

Nous avons donné dans ce chapitre une vue général sur les bras robotisé en passant par leur définition, leurs types, leur structure, à savoir les structures, nous avons cité les domaines d'applications les plus répondue, et décrit sa constitution, notamment sa structure mécanique ainsi que les différents capteurs et actionneurs utilisés dans le domaine des bras robotisé.

Chapitre II : La description des parties matériel et logiciel

II.1 Introduction:

Lechapitreque nous allonsentamerva nous permettre de détailler et savoir exploiter tous les composants que nous voyons essentiels pour notre réalisation, à savoir Arduino nano, servomoteurs et capteur de couleur.

Sans avoiroublier de faire un passage explicatif et détailléconcernant les logicielsutilisésdansceprojeton note pour cela le logiciel IDE Arduino, ainsique le logiciel MIT2 (App-Inventor-2) qui est un environnement de programmation concu pour des applications mobiles sous Android.

II.2Partie matériel:

II.2.1L'arduino:

II.2.1.1 Définition:

Arduino est une plate-forme électronique open source basée sur du matériel et des logiciels faciles à utiliser.

La carte Arduino est généralement construit autour d'un microcontrôleur Atmel AVR (ATmega328, ATmega32u4 ou ATmega2560 pour les versions récentes, ATmega168, ATmega1280 ou ATmega8 pour les plus anciennes), et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles).

La carte Arduino que nous allons utiliser pour notre projet est de type Arduino-nano, en effet nous aborderons en ce qui s'en suit le détaille de cette version.

II.2.1.2 Arduino Nano:

L'Arduino Nano est une petite carte complète et compatible, elle est basée sur le microcontrôleur l'ATmega328.Sa petite taille lui permet de s'adapter aux petits systèmes électroniques embarqués.

Arduino nano est dotée de 14 pins digitaux à travers lesquels elle fournit 40 mA maximum. En outre, elle intègre 2 pins analogiques de plus que l'ArduinoUno. Cela fait donc 8 pins analogiques pour Arduino Nano. En ce qui concerne la capacité, et il est dotée d'une mémoire Flash de 32Kb et 2Kb de mémoire dynamique. [7]

Nous tenons ànoterque le microcontrôleur ATmega328 de la carte Arduino Nano bénéficie d'un programmebootloaderlogédans la mémoire Flash. Celui-ci assure le téléchargement de nouveaux codes sans pour autantsolliciterunprogrammateurmatérielsupplémentaire. Toutefois, ilexisteune solution qui

permet de se passer du bootloaderafin de programmer le microcontrôleur grâce au connecteur In-Circuit Serial Programming (ICSP).

La **figure II.1** montre le brochage de la carte ArduinoNanoainsique les fonctionsessentiellesqu'ellecontient à savoir entrées analogiques,sorties PWM, les entrées /sorties numériques, liaison I2C et liaison SPI etc..

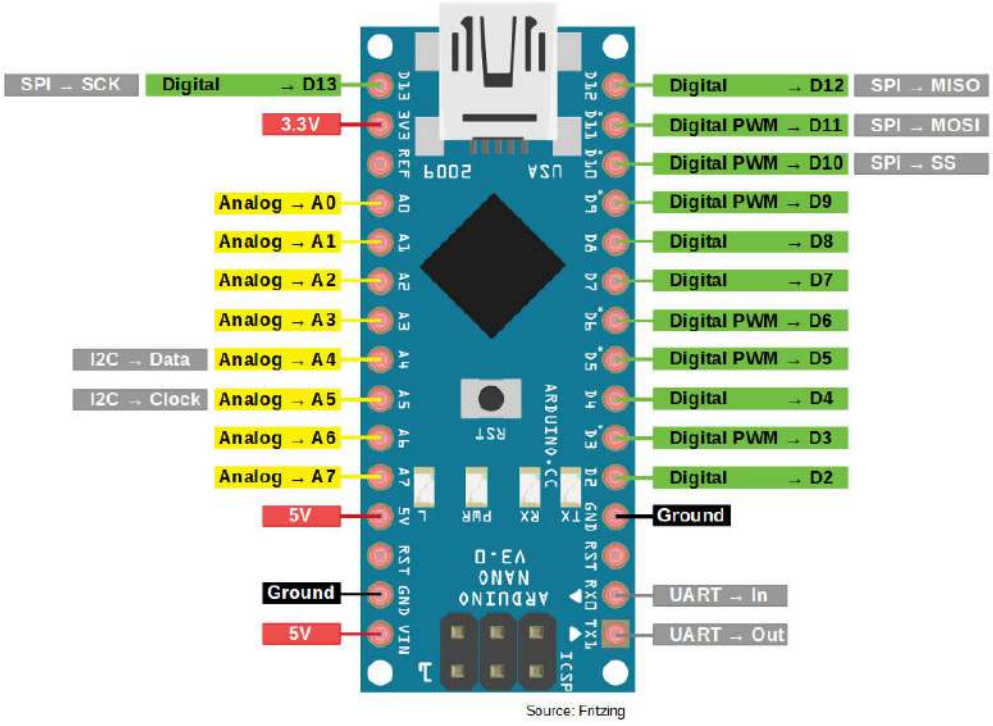


Figure 1 Structure de la carte Arduino nano

.

Microcontroller	Atmega328p/Atmega 168
Operating Voltage	5V
Input Voltage	7 – 12 V
Digital I/O Pins	14
PWM	6 out of 14 digital pins
Max. Current Rating	40mA
USB	Mini
Analog Pins	8
Flash Memory	16KB or 32KB
SRAM	1KB or 2KB
Crystal Oscillator	16 MHz
EEPROM	512bytes or 1KB
USART	Yes

Taleau 1 les caractéristiques de carte Arduino nano.[8]

II.2.2 Module BluetoothHC-05:

La communication en Bluetooth estunenormepermettantl'échangebidirectionnel de données à trèscourte distance en utilisant des ondes radio UHFsurunebande de fréquence de 2,4 GHz. Son objectifest de simplifier les connexions entre les appareilsélectroniques en supprimant des liaisons filaires.

Pour ce qui estdu module Bluetooth HC-05 est un module Bluetooth PPS (protocole de port série) facile à utiliser, conçu pour la configuration d'uneconnexionsériesans filtransparente. ce qui facilitel'interfaçage avec les microcontrôleurs via deuxbroches **Rx**et **Tx**, telle quela broche **Rx** de ce dernierdoitêtreraccordée à la broche **Tx** du module Bluetooth HC-05 et la broche **Tx**doitêtreraccordée à la broche **Rx** du module HC-05. **[9]**

Alimentation	3,6 à 6 Vcc (régulateur 3 Vcc intégré)
Bluetooth	2.0
Portée	jusqu'à 10 mètres
Liaison série	4800 à 1382400 bauds
Antenne	2,4 GHz intégrée
Sorties	Key, Vcc, Gnd, TXD, RXD et State.
Configuration	via commandes AT
Longueur du cordon	environ 20 cm
Dimensions	35 x 16 x 12 mm

Tableau 2 Caractéristique du module Bluetooth HC-05[9]

Selonla figure II.2 le HC-05 contient 6 broches:

- **VCC** broche d'alimentation. connectée à la broche 5V de l'Arduino.
- **GND** masse. connectée à la broche GND de l'Arduino
- **RX** broche de réception. connecté à la broche de transmission (TX) de l'Arduino
- **TX** broche de transmission. connecté à la broche de réception (RX) de l'Arduino
- **State** retourne 1 lorsque le module estconnecté
- **Key**ou**EN**doitêtrealimentée pour entrer dans le mode de configuration.

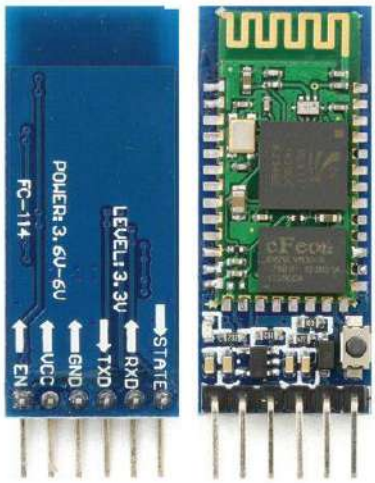


Figure 16 module Bluetooth HC-05.

II.2.3 Capteur de couleurTCS3200:

Uncapteur de couleurest un détecteur de couleursincluantun puce capteur TCS3200. Il estégalementcomposé de 4 LEDS blanches servant à éclairerl'objet exposé sinécessaire. Le capteur peut détecter et mesurer une gamme quasi illimitée de couleurs visibles. [10]

Il est caractérisé par:

- Tension d'entrée: (2,7 V à 5,5 V)
- Interface: TTL numérique.
- Conversion haute résolution de l'intensitélumineuse en fréquence
- Couleur programmable et fréquence de sortie pleineéchelle
- Pas besoind'ADC (peutêtredirectementconnecté aux brochesnumériques du microcontrôleur).
- Fonction de mise hors tension.
- Température de travail: -40oC à 85oC
- Taille: 28, 4 x 28, 4 mm (1,12 x 1,12 ").

Il dispose d'uneMatrice de photo détectrice de 8x8 comme le montre la **figure II.3,** CetteMatrice de photodétectriceestcomposée de filtres rouges, verts et bleus.En effetlorsqueon choisie unfiltre de couleur, il ne peutlaisser passer qu'unecouleurparticulière et empêche les autres de passer.

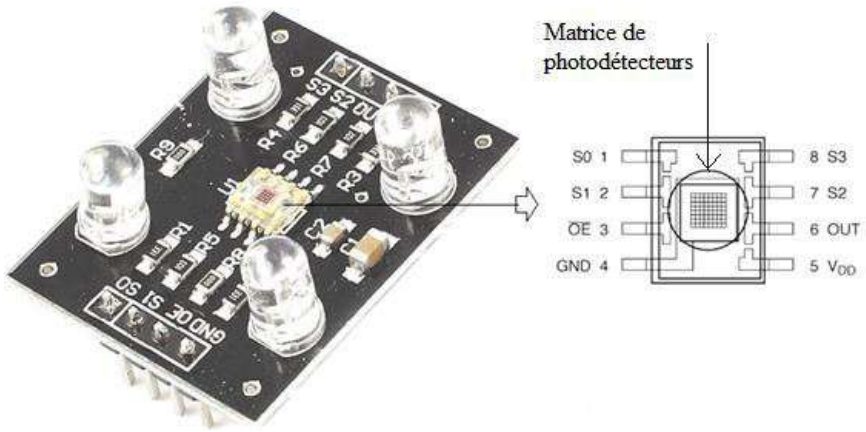


Figure 17 Capteur de couleur TCS3200

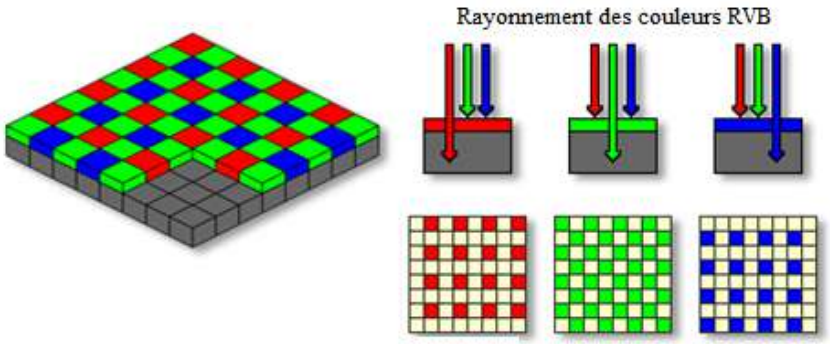


Figure 18 Organisation de la Matrice des photos détectrices

Dans le capteur, se trouveunoscillateur qui produit un signal dont la fréquenceestproportionnelle à l'intensité de la couleurchoisie. Ce signal estensuitetransmet à un convertisseur qui le convertie en fréquence, voir la **figure II.5.**

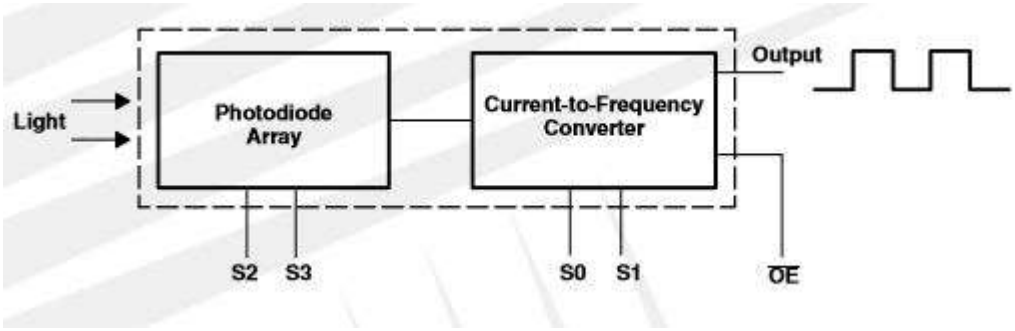


Figure 19 Structure interne du capteur TCS3200

Le capteurTCS3200 estconstituéde 8 pins comme le montre la figure II-3 telsque:

- S0-S1: sélection de mise à l'échelle des fréquences de sortie
- S2-S3: sélection du type des photos détectrices (Filtres)
- OUT: fréquence de sortie (signal carré)
- OE: broche d'activation de sortie (OE)
- Vcc-GND broches d'alimentation.

S0	S1	FREQUENCE DE SORTIE
L	L	Power down
L	H	2%
H	L	20%
H	H	100%

Tableau 1.Les fréquences de sortie Tableau

S2	S3	PHOTODETECTEURS
L	L	Red
L	H	Blue
H	L	Clear (no filter)
H	H	Green

Tableau2.Photo détecteur

II.2.4 Les servomoteurs:

II.2.4.1 Définition :

Unservomoteur (vient du latinservus qui signifie « esclave ») est unmoteur capable de maintenirunepositionà un effort statique etdont la position estvérifiée en continu et corrigée en fonction de lamesure. C'estdoncunsystèmeasservi.Les servos peuventactionner les parties mobiles d'un robot, drone, etc.…

Il estconstitué d'un ensemble mécanique et électroniquecomprenant :

- un moteur à courant continu.
- un réducteur en sortie de cemoteurdiminuant la vitessemaisaugmentant le couple ;
- un potentiomètre (faisantfonction de diviseurrésistif) qui génèreune tension variable, proportionnelle à l'angle de l'axe de sortie ;
- UN dispositiveélectroniqued'asservissement.
- un axe dépassant hors du boîtier avec différents bras ouroues defixation.

II.2.4.2Fonctionnement des servos :

Le servomoteur a besoin de troisfils deconnexion pour fonctionner. Deuxfilsservent à son alimentation, le dernier étantcelui qui reçoit le signal de commande :

• rouge : pour l'alimentation positive (4.5Và 6V en général)

• noiroumarron : pour la masse (0V)

• orange, jaune, blanc, … : entrée du signalde commande

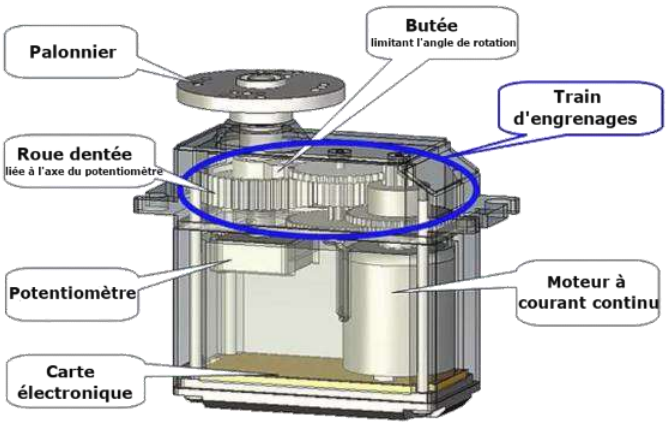


Figure 20 Structure d'un servomoteur

Le potentiomètreestfixésur le pignon final ousurl'arbre de sortie. Ainsi, lorsque le moteurtourne, le potentiomètretourneégalement, produisantainsiune tension liée à l'angleabsolu de l'arbre de sortie. Dans le circuit de commande, cette tension estcomparée à la tension provenant de la ligne de signal. Si nécessaire, le contrôleur active unpont en H intégré

qui permet au moteur de tournerdans un sensoudansl'autrejusqu'àceque les deuxsignauxatteignentunedifférence de zéro.

Unservomoteurestcontrôlé en envoyantuneséried'impulsions à travers la ligne de signal. La fréquence du signal de commandedoitêtre de 50Hz ouune impulsion doitavoir lieu toutes les 20 ms. La largeur de l'impulsiondétermine la position angulaire du servo et ce type de servo peutgénéralementpivoter à 180 degrés (ilsontunelimite physique de déplacement). **[11]**

II.3PartieLogiciel:

II.3.1 ArduinoIDE :

C’estunlogiciel open source téléchargeablegratuitementsur le site officield’Arduino. Lepuissant environnement de développementintégré (IDE) qu’ilfournitpermet aux programmeursle développement du logiciel. Ilfonctionne sous Windows, Mac OS X et Linux. L'environnementestécrit en Java et basésur Processing et autreslogiciels open source, Celogicielpeutêtreutiliséavecn'importequelle carte Arduino.



Figure 21 Logo du logiciel IDE Arduino.

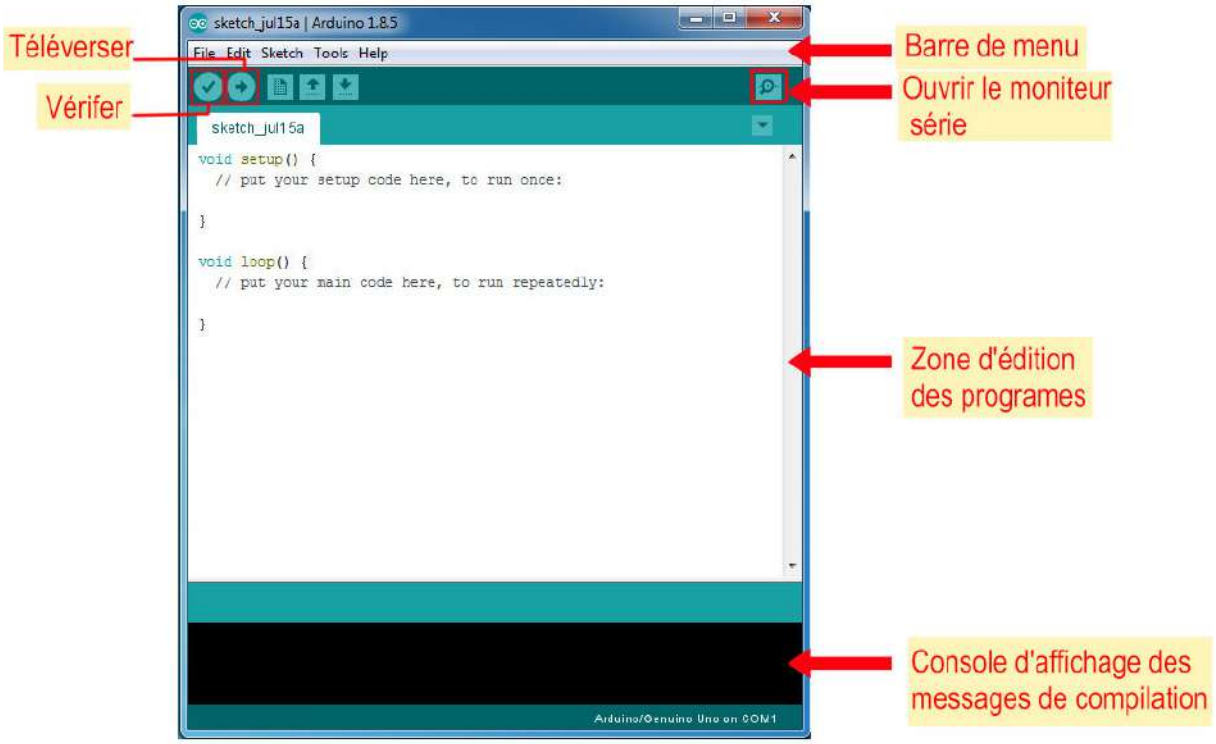


Figure 22 fenêtre de logiciel IDE

Il nous permetd'éditer des croquis en respectant la structure du code et les téléverserdansla mémoireArduino après la compilation du programmedans le langage machine. Ilpermetaussiune communication avec la carte par le terminal.**[12]**

II.3.2 Proteus professionnel :

Proteus est un logiciel permetant de faire des simulations des circuits électroniques,il est doté de deux editeurs ISIS et ARES.

ISIS est concu pour la simulation alors que ARES est un editeur consacré à la réalisation des circuits imprimés,

le processus de construction d'unecarte électronique à microcontrolleurs se compose de cinq étapes :

- Conception de circuits électroniques.
- Conception de circuits conçus.
- Construction du prototype fiscal: circuit imprimé et soudage des composants.
- Développement de logiciels.
- Test de fonctionnalité.**[13]**

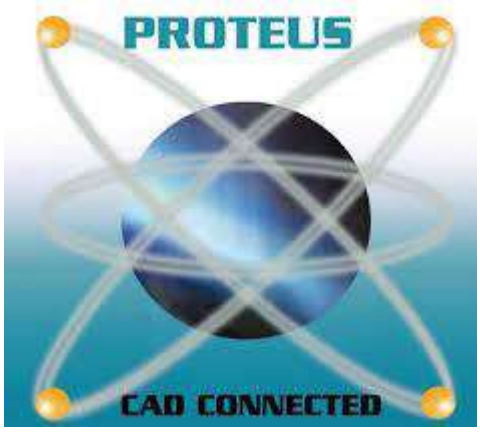


Figure 23 Logo de PROTEUS professionel.

II.3.3Fritzing :

Fritzing est une initiative matérielle open-source qui rend l'électronique accessible en tant que matériau créatif pour tous. Il propose un outil logiciel, un site Web communautaire et des services dans l'esprit de Processing et Arduino, favorisant un écosystème créatif qui permet aux utilisateurs de documenter leurs prototypes, de les partager avec d'autres, d'enseignerl'électronique dans une salle de classe, de mettre en page et de fabriquer des circuits imprimés professionnels. **[14]**



Figure 24 logo de Fritzing

II.4 ApplicationAndroid :

Nousallonscréernotre application en utilisant un navigateur Web qui nous fournit un créateur d'applications facile et très simple appelé MIT App Inventor.

II.4.1 MIT App Inventor 2 :

App Inventor vouspermet de développer des applications pour les téléphones Android à l'aide d'un navigateur Web et d'un téléphoneconnectéou d'un émulateur. Les serveurs App Inventor stockentvotre travail etvousaident à suivrevosprojets.



Figure 25 Logo de MIT App inventor2

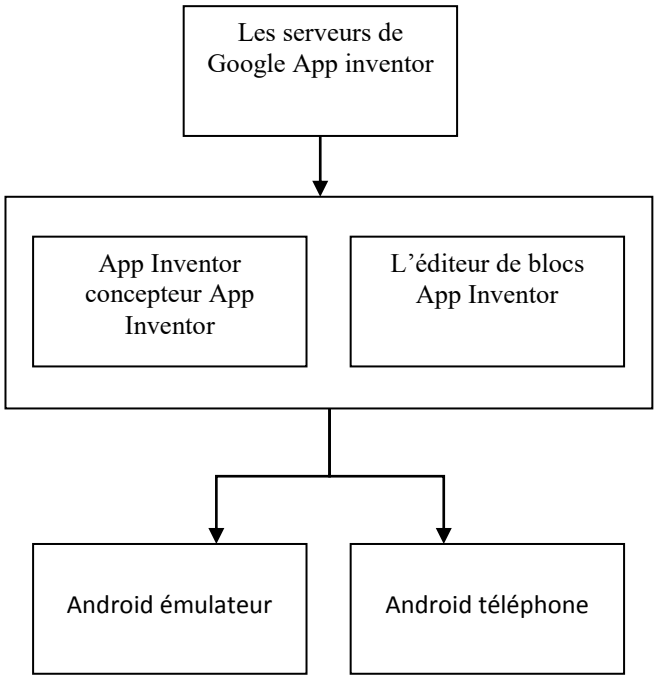


Figure 26 schéma synoptique de la réalisation d'une application.

Vouscréez des applications en travaillant avec:

- **Le concepteur App Inventor** :danslequelvoussélectionnez les composants de votre application.
- **L'éditeur de blocs App Inventor** :danslequelvousassemblez des blocs de programme qui spécifient le comportement des composants. Vousassemblez les programmesvisuellement, en assemblant les piècescomme les pièces d'un puzzle.

Votre application apparaîtsur le téléphoneétape par étape au fur et à mesurequevous y ajoutez des éléments, afinquevouspuissiez tester votre travail au fur et à mesurequevousconstruisez. Lorsquevousavezterminé, vouspouvezempaquetervotre application etproduireune application autonome à installer.

Si vousn'avez pas de téléphone Android, vouspouvezcréervos applications à l'aide de l'émulateurAndroid , un logiciel qui s'exécute sur votreordinateur et se comportecomme le téléphone.

L'environnement de développement App Inventor estpris en charge pour les systèmesd'exploitation Mac OS X, GNU / Linux et Windows, ainsique pour plusieursmodèles de téléphones Android populaires. Les applications créées avec App Inventor peuventêtreinstalléessurn'importequeltéléphone Android.**[15]**

II.4.2 Environnement de travail :

Pour utiliser l'application MIT App Inventor, les utilisateurs doivent créer leur propre comptepour accéder à l'application.

L'espace de travail est divisé par deux: l'espace de travail du concepteur et l'espace de travail des blocs :

- **l'espace de travail du concepteur :**

  Dans cette interface on trouve:

  - **la palette**, pour ajouter un composant.
  - **L'afficheur,** pour voir le design et les modifications
  - **Les composants,** pour voir les composants utilisés et les modifier.
  - **Les propriétés,** pour modifier les propriétés de chaque composant.



Figure 27 l'espace de travail du concepteur

- **l'espace de travail des blocs :**

  Dans cette interface on trouve:

  - **Les blocs :** pour ajouter des blocs.
  - **L'afficheur :** pour voir les blocs utilisés et les modifier.



Figure 28 L'espace de travail des blocs.

II.5 Conclusion :

Danscechapitre, nous avons exposé les logiciels utilisés et les composants de base pour démarrernotreprojet, nous les avons montrés et expliqué le fonctionnement, et cela nous mènera au chapitresuivant.

Chapitre III: La conception et la réalisation du bras robot

III.1 Introduction :

Dans ce chapitre, nous allons présenter la réalisation d'un bras robotique a quatre degrés de liberté.

Nous allons montrer l'ensemble du processus de construction, de la conception de pièces de robot de la connexion de composants électroniques et de la programmation de la carte Arduino, au développement de notre application Android pour le contrôle du bras.

La première étape a été de concevoir les pièces de braspour les construire, puis de les assembler avec les servomoteurs et de connecter le câblage électrique.

La deuxième étape consiste à vous familiariser avec le "MIT App Inventor" pour créer une application Android qui envoie les données nécessaires à l'Arduino via Bluetooth.

Pour terminer, la réalisation du circuit composé de l'Arduino, du module HC-05, des servomoteurs et de l'alimentation.

III.2 Un bref aperçu de notre bras manipulateur :

Nous avons conçu et réalisé un bras manipulateur à 4 degrés de liberté, ce bras est fixe et ses organes sont en série. Ses pièces sont: la base, l'épaule, le coude et le pince.

Le robot aura 4 servomoteurs, nous utiliserons le servomoteur de typeMG990 pour l'épaule et pour les autres servomoteurs nous utiliserons de type SG90, puis nous avonsprogrammé par la carte Arduino nano(ATmega328) en utilisant le logiciel IDE.

Ce robot fonctionne en recevant des commandes par l'utilisateur utilisent 3 manières différentes:

- Utilisation d’un Smartphone (sans fil).
- Utilisation du PC(liaison USB).
- Utilisation manuelleavec un potentiomètre.

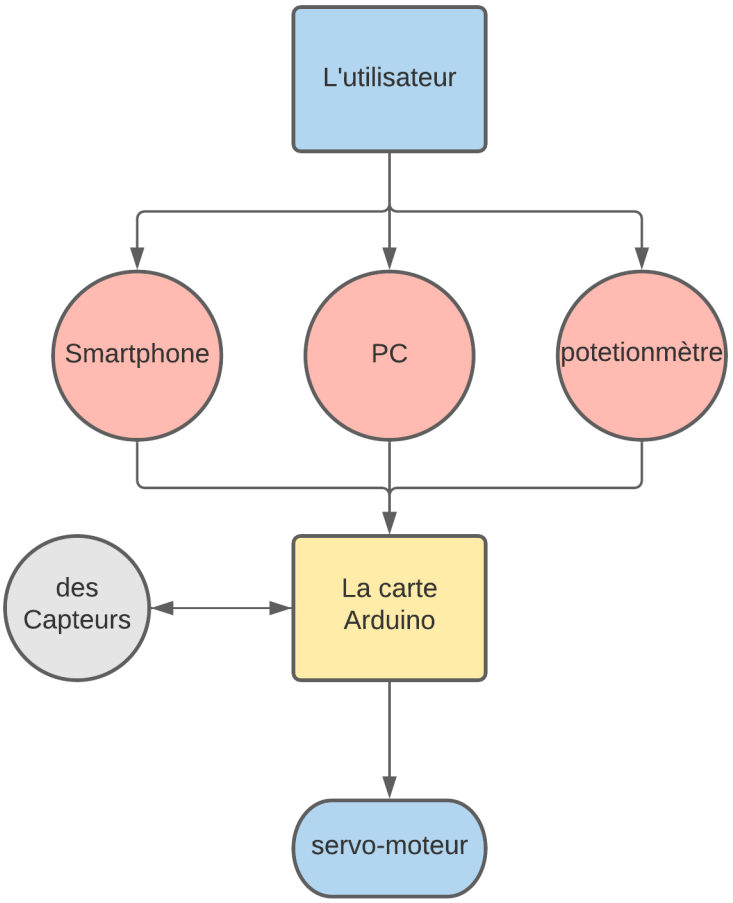


Figure 29 Organigramme de commande du bras.

Nous avons programmé ce bras pour deux tâches:

- Trier les pièces colorées en les mettant dans le bon panier (Triage par couleur).
- De mettre les pièces aux endroits que nous avons commandés, avec les angles des organes qui sont déjà programmés.

On peut aussi tourner les organes sans tâches avec le potentiomètre, et chaque moteur a certaines limites de rotation.

III.3 Partie mécanique:

Dans cette partie nous allons présenter la partie mécanique du bras, de la conception à la construction des pièces et le montage.

III.3.1 conception du robot:

Nous avons conçu chaque pièce du bras et nous avons choisi les mesures optimales des pièces pour éviter les problèmes mécaniques et physiques et les rendre compatibles avec chacun.

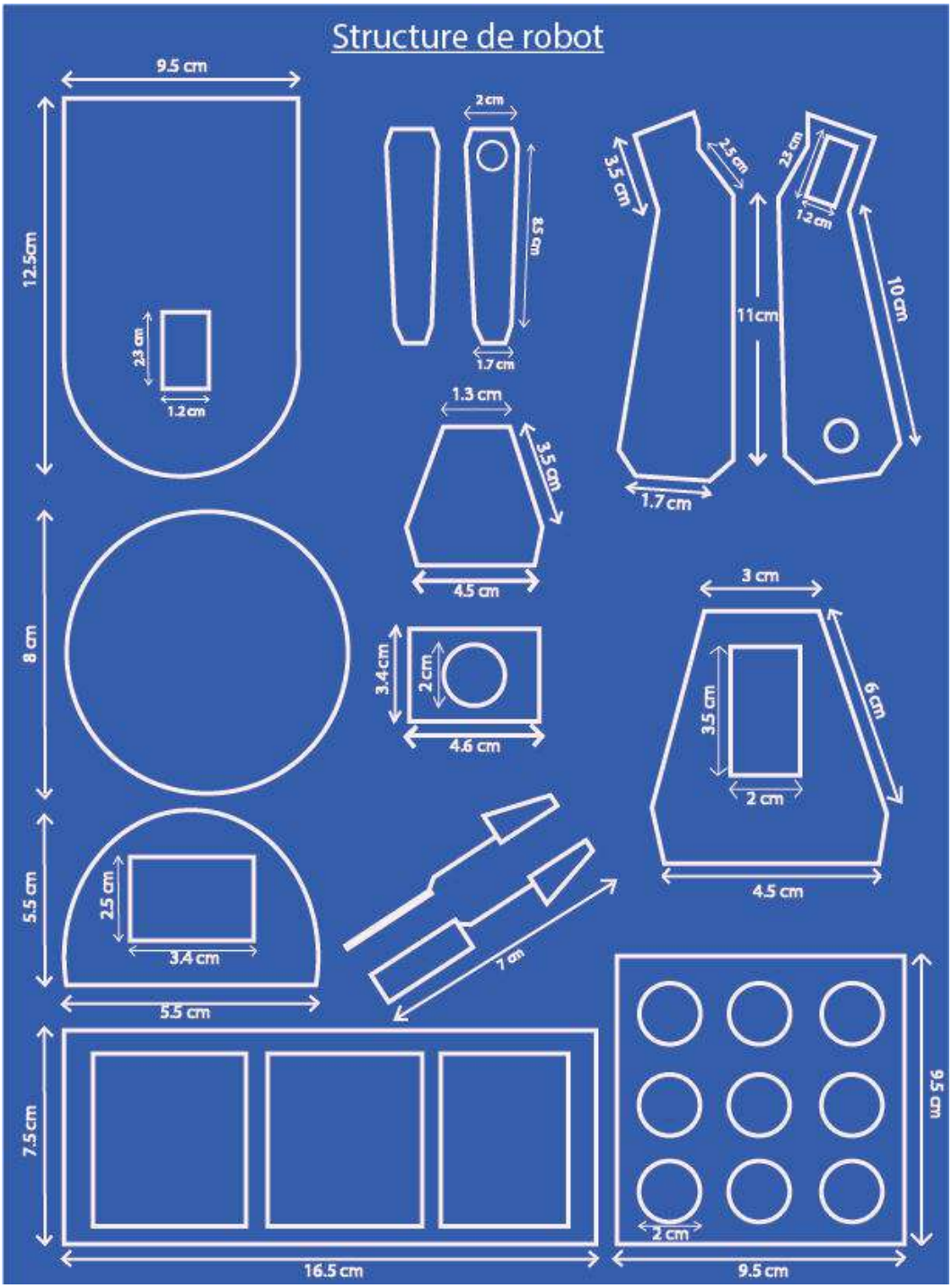


Figure 30 Les dimensions des pièces.

La **figure III.3** montre comment sera le robot après avoir assemblé ses pièces.

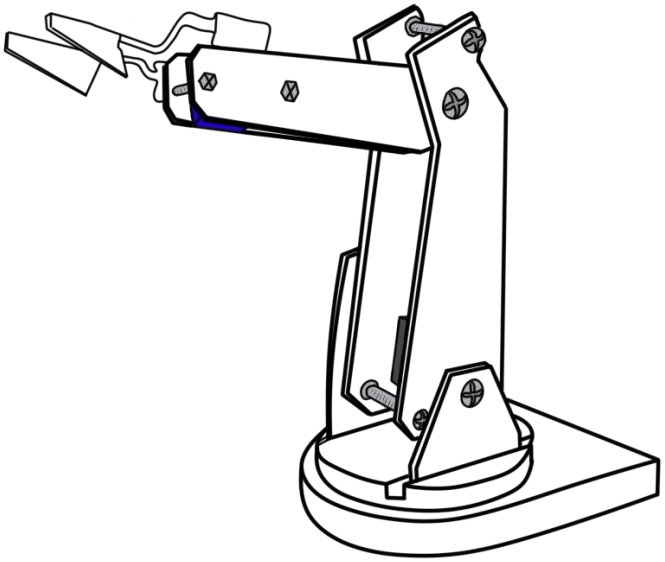


Figure 31 Dessin du bras assemblé.

III.3.2 Fabrication des pièces:

Le matériau utilisé pour construire les pièces du bras et les pièces de triage est le bois d'hêtre, sauf que la pince est en fer, les pièces sont :

- ✓ La base.
- ✓ L'épaule.
- ✓ Le coude.
- ✓ La pince.
- ✓ l'avant épaule.
- ✓ Le panier de triage.
- ✓ La matrice de triage.
- ✓ Position pour ramasser.
- ✓ Support de capteur de couleur.



Figure 32 La base.



Figure 33 L'épaule.



Figure 34 montage de l'avant épaule.



Figure 35 Le coude.



Figure 36 Pince



Figure 37 Panier de triage par couleur.



Figure 38 Matrice de triage Matriciel.





Figure 39 position de ramassage. **Figure 40 Support de capteur de couleur.**

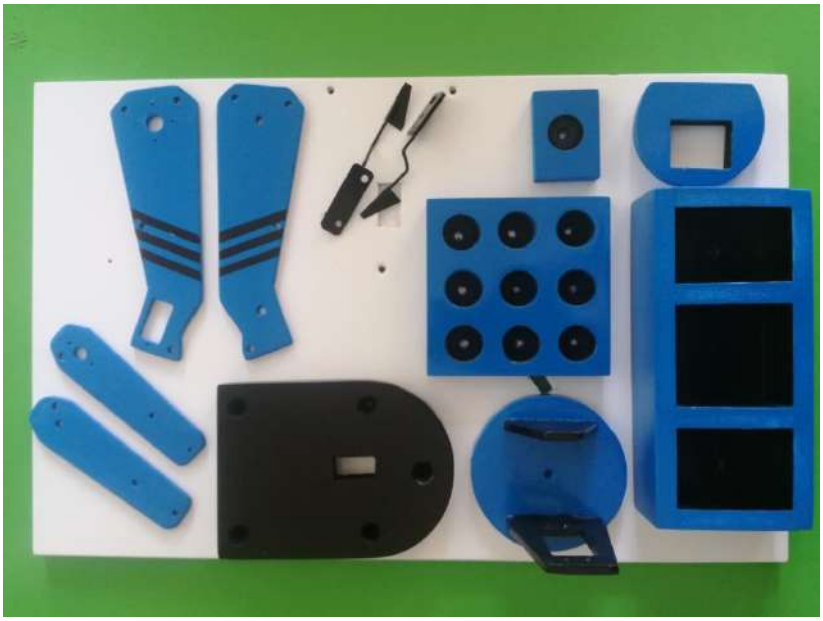


Figure 41 Les pièces du bras final.

III.3.3 Le montage des pièces:

Après avoir construit toutes les pièces, nous avons commencé à les assembler avec les moteurs, les figures ci-dessous montrent le montage étape par étape:

- ✓ Montage de l'avant épaule avec la base et deux servomoteurs.

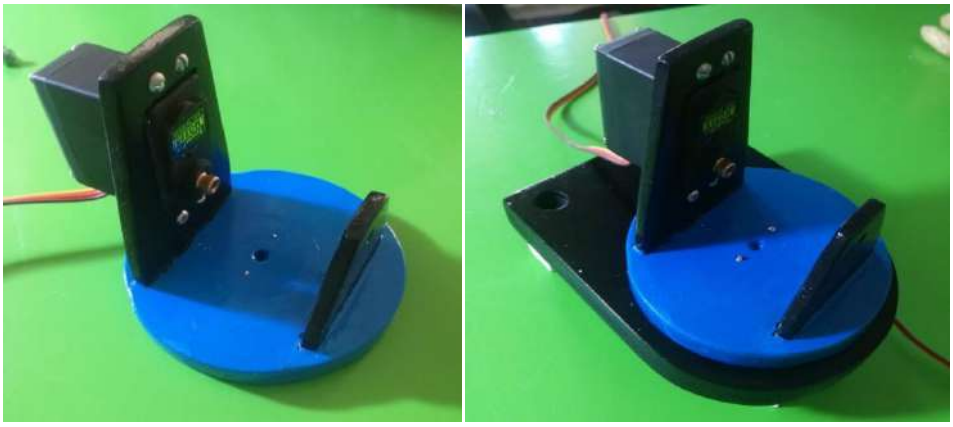


Figure 42 montage de base avec l'avant épaule et deux servomoteurs.

- ✓ Montage de l'épaule avec un servomoteur.



Figure 43 Montage de l'épaule avec un servomoteur.

- ✓ Montage de coude avec la pince et un servomoteur.

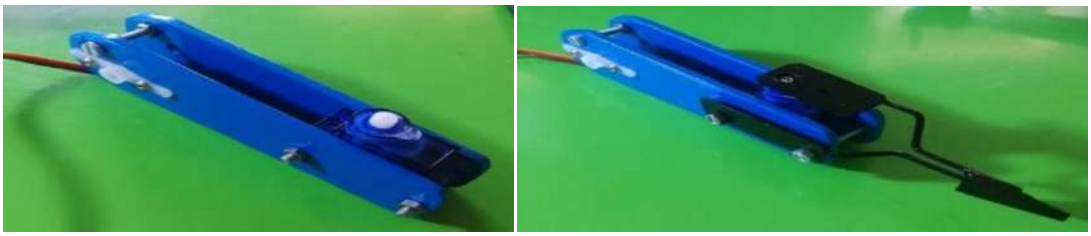


Figure 44 Montage de coude avec la pince et un servomoteur.

- ✓ Montage de l'épaule avec le coude.



Figure 45 Montage de l'épaule avec le coude.

- ✓ Montage de base avec l'épaule et le coude.

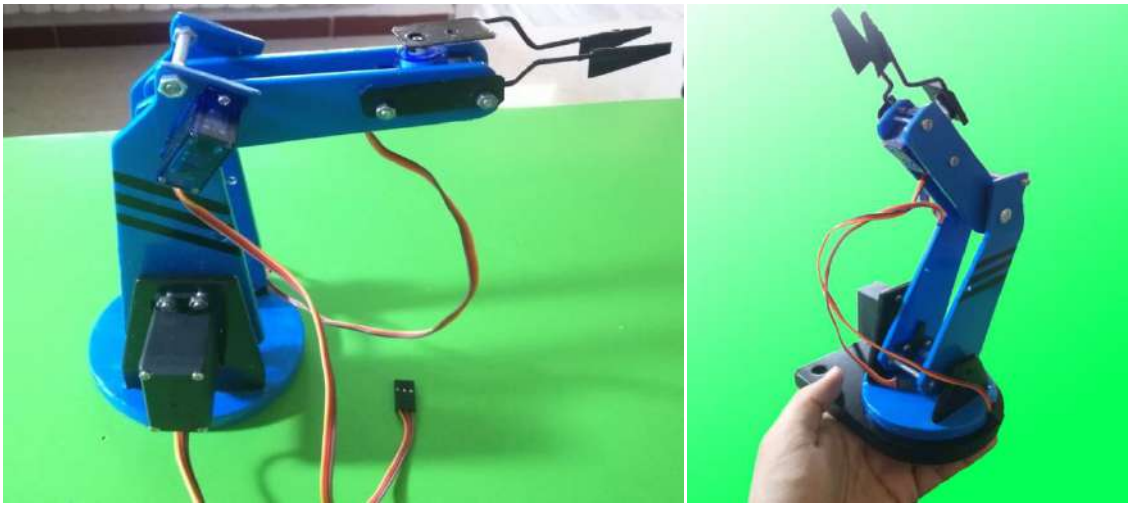


Figure 46 Le montage de base avec l'épaule et le coude.

- ✓ Montage des autres pièces avec une surface de travail avec le robot.

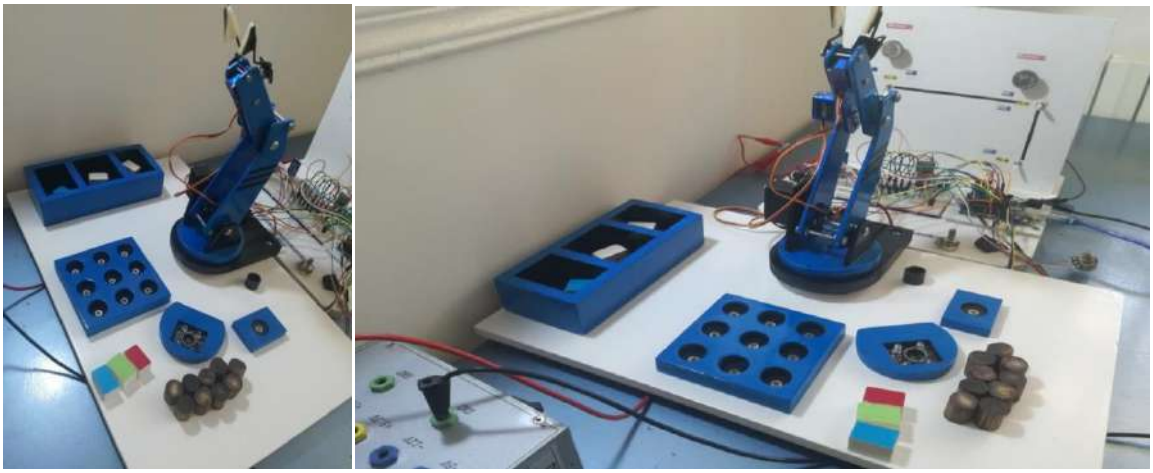


Figure 47 Montage Finale du bras.

III.4 La carte de commande :

Sur notre cartede commande nous avons ajouté :

- ✓ la carte Arduino nano.
- ✓ le module Bluetooth HC-05.
- ✓ Un potentiomètre.
- ✓ Six boutons.
- ✓ Deux LED.
- ✓ dix résistances.

Le schéma électrique est représenté sur la **Figure III.20.**

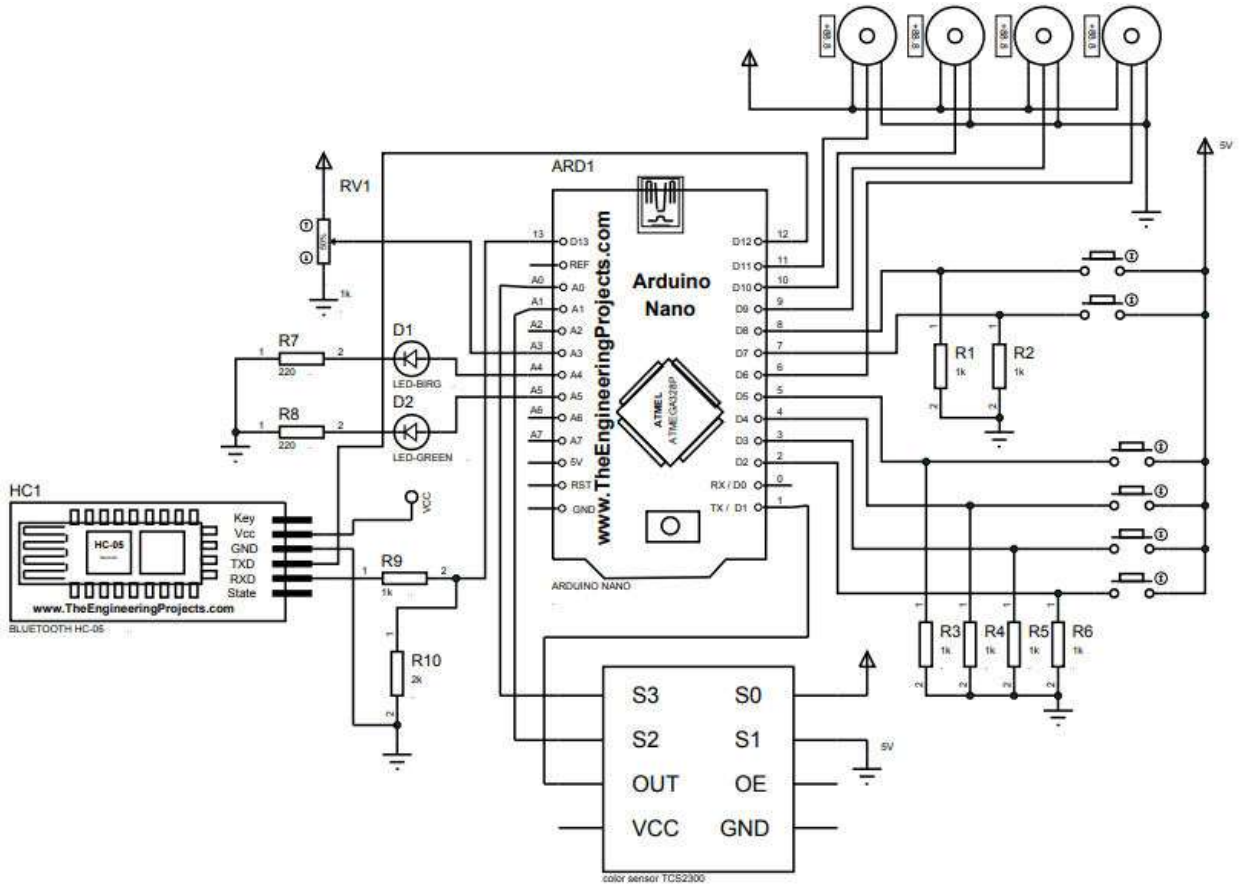


Figure 48 Schéma électrique de la carte de commande.

III.4.1 Réalisation de la carte de commande:

A partir de La **Figure III.20**, nous avons utilisé le logiciel Proteus pour créer le circuit imprimé,

Dans l'éditeur ISIS (l'onglet de capture schématique), nous avons créé le circuit en ajoutant tous les composants dont nous avons besoin et les avons liés ensemble exacte comme le schéma électrique afin de le faire passer à l'ARES (l'onglet de PCB layout)

L'ISIS nous permet de compiler le circuit afin de corriger nos erreurs avant de passer à l'étape suivante et c'est ce qui rend le circuit plus sûr.

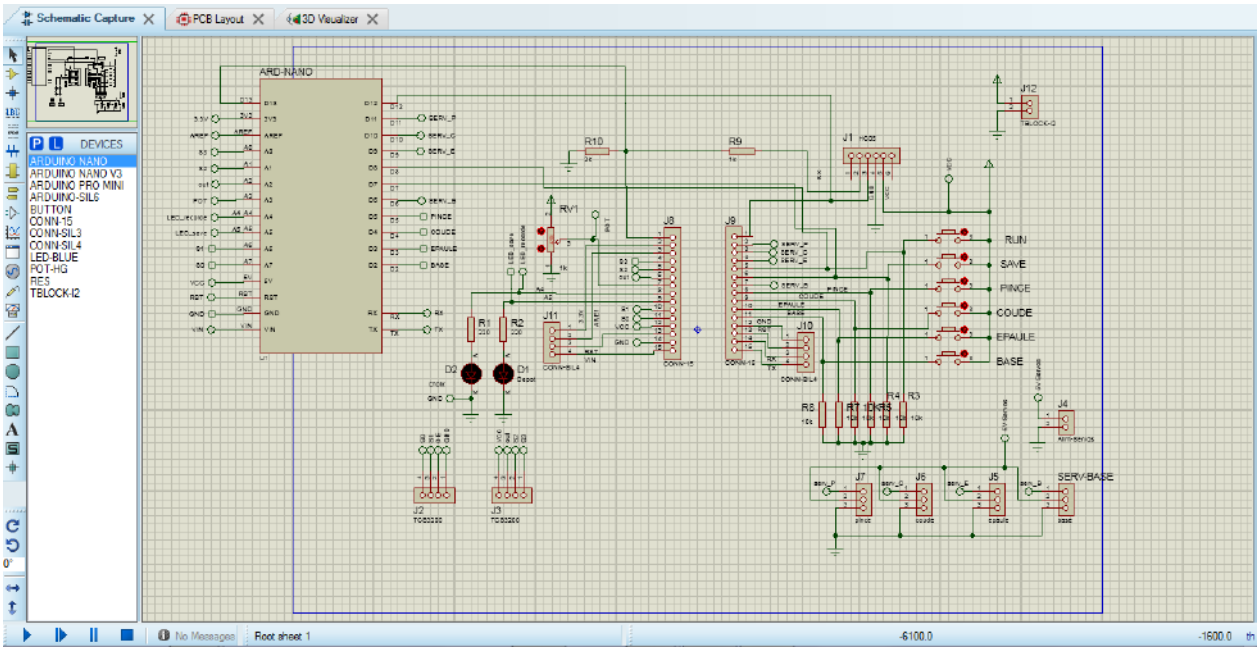


Figure 49 Vue depuis l'ISIS.

Avec l'ARES nous avons réussi à voir comment la carte allait être en organisant les positions des composants et en les reliant de manière à ce qu'elles ne se croisent pas, et nous avons essayé de nous assurer que les points de soudage ne sont pas très proches les uns des autres afin que le soudage soit plus facile et plus sûr en évitant les courts-circuits sur les deux côtés du cuivre supérieur et cuivre inférieur.

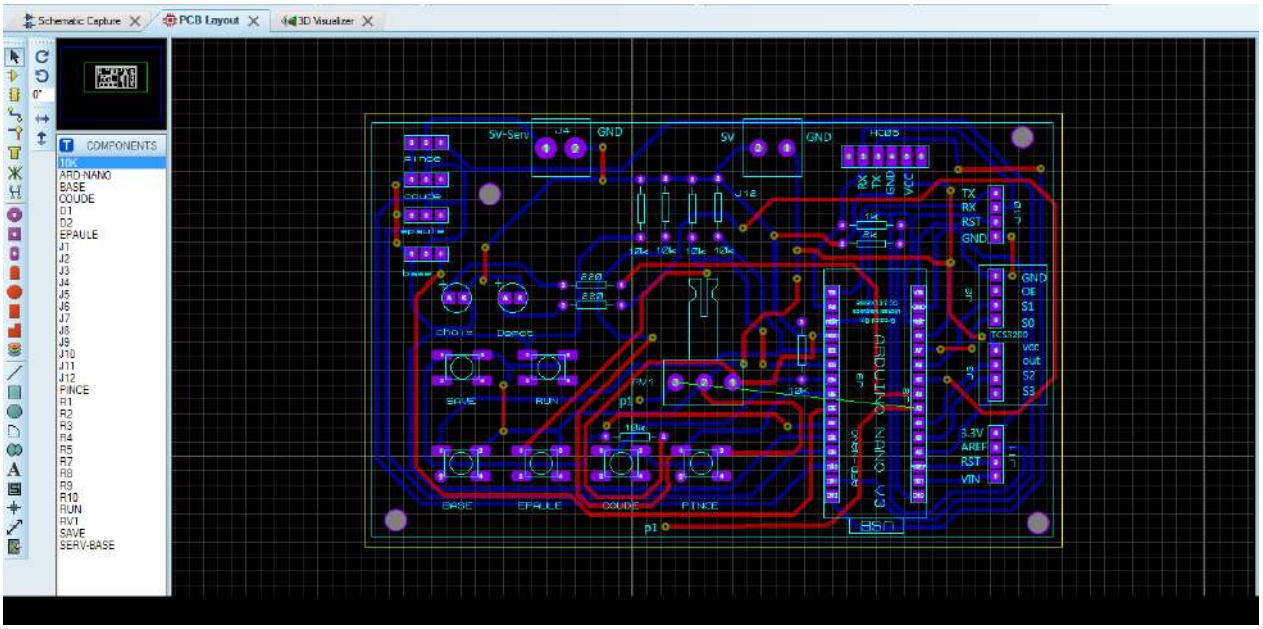


Figure 50 Vue depuis l'ARES.

Dans cette étape l'onglet «3Dvisiluazer» nous donne une vision 3D du circuit imprimé qui montre sa forme définitive.

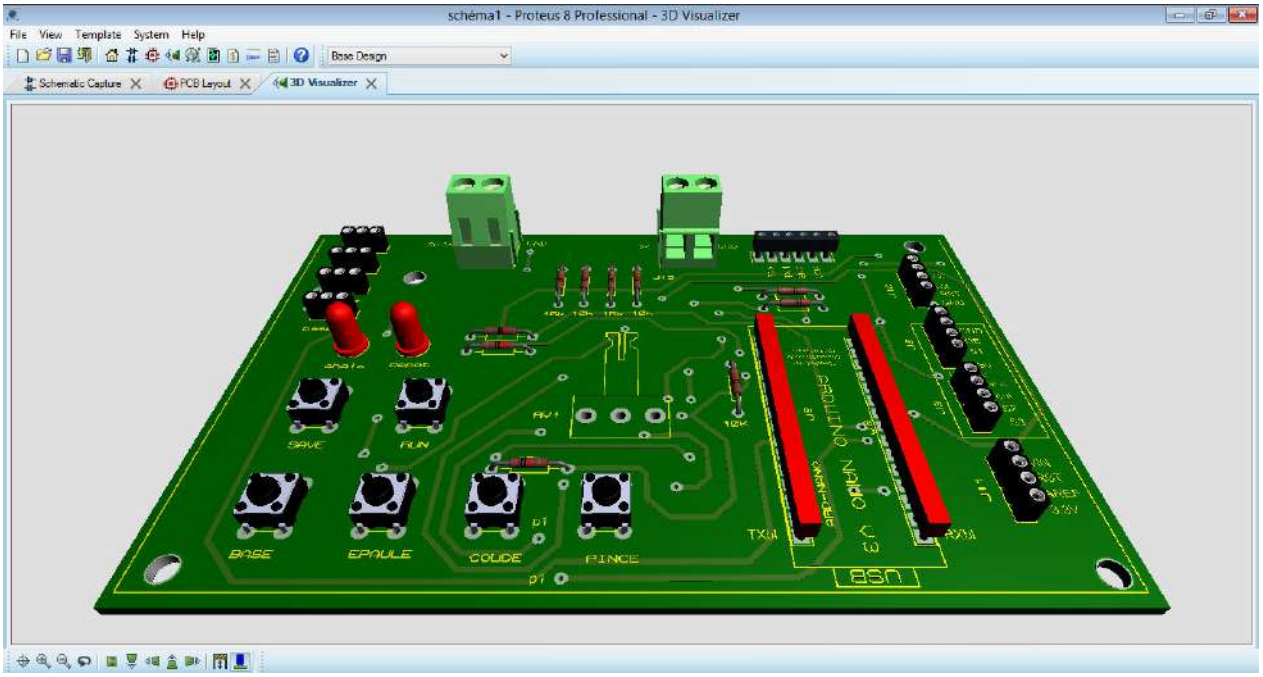


Figure 51 Vue 3D du circuit imprimer sur Proteus.

En utilisant une machine d'impression, nous avons réussi à imprimer notre carte de commande.

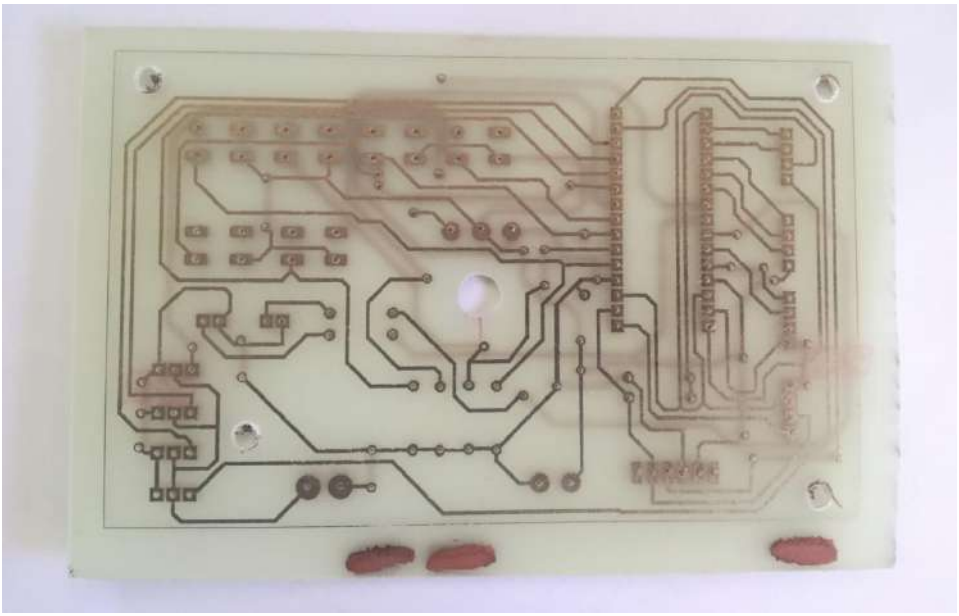


Figure 52 La carte de commande après l'impression.

Nous soudons soigneusement les composants avec la carte et évitons les courts-circuits.



Figure 53 La carte de commande après le soudage (final).

Le schéma synoptique de **figure III.26**résume la réalisationd'une carte de commande :

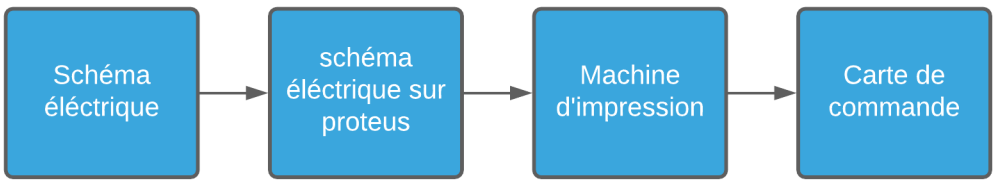


Figure 54 schéma synoptique de réalisation d'une carte de commande.

III.5 Partie Commande:

Dans ce partie nous allons expliquer les trois types de commande étape par étape :

- Commande via une application Android par Bluetooth (sans fil)
- Commande par PC avec les angles des organes du robot préprogrammés directement via le logiciel IDE par liaison USB.
- Commande par un potentiomètre sur la carte de commande.

On va également présenter la programmation de chaque type.

III.5.1 Commande par application Android :

En utilisant MIT App Inventor, nous avons créé notre application qui nous permet de commander notre robot, ce qui signifie que nous pourrons envoyer et recevoir des données d'une application Android vers le robot, cette application communique avec la carte Arduino par Bluetooth du Smartphone et le module Bluetooth HC-05.

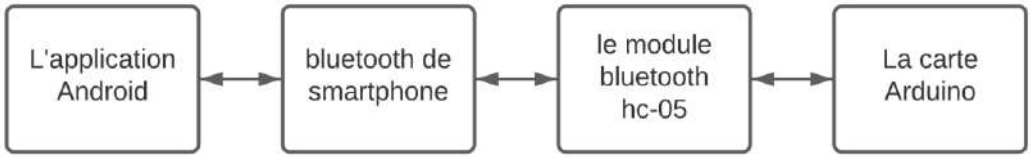


Figure 55 La communication entre l'application Android et la carte Arduino.

L'application envoie des codes déjà programmés à la carte Arduino qui sont reçu par le Bluetooth HC-05

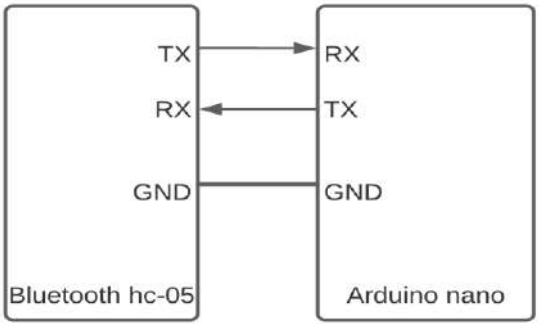


Figure 56 La communication entre le module Bluetooth HC-05 et la carte Arduino.

Pour lire les données reçues par Bluetoothsur la carte Arduino, le programme doit inclure la bibliothèque de Bluetoothet configurer ses TX et RX.

```
#include <SoftwareSerial.h>

SoftwareSerial Bluetooth(12, 13);
```

Puis nous avons lu les données reçues et les avons sauvegardées comme une chaîne de caractères

```
if (Bluetooth.available() > 0) {
    if (Bluetooth.available() > 0)    donnees = Bluetooth.readString();
    if (Serial.available() > 0) donnees = Serial.readString();
```

Nous avons expliqué la réalisation de notre application étape par étape des deux côtés, la conception et les blocs.

On a divisé notre application sur trois écrans :

- Ecran d'accueil.
- Ecran de connexion.
- Ecran de commande.

III.5.1.1L'écran d'accueil :

La conception de l'écran d'accueil :

Sur notre application le premier écran qui fait face à l'utilisateur est l'écran d'accueil qui donne des informations sur l'application.

Nous mettons trois boutons, bouton d'accueil, bouton de connexion et bouton de fonction.



Figure 57 L'écran d'accueil.

Les Blocs de l'écran d'accueil :

La**Figure III.31**montre quechaque bouton est programmé par blocs pour fonctionner comme:

- **Bouton connexion**: passe à l'écran Bluetooth pour se connecter à la carte Arduino
- **Bouton fonction**: passe aux écrans de fonctionnement afin de commander le robot
- **Bouton accueil**: passe à l'écran d'accueil.

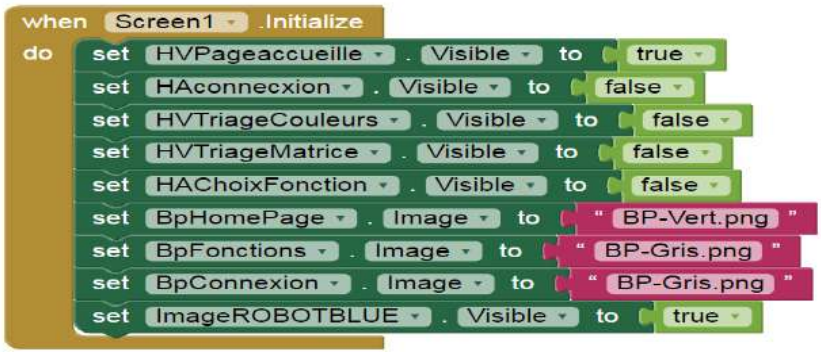


Figure 58 le bloc de démarrage d'application.



Figure 59 Les blocs de l'écran d'accueil.

III.5.1.2 L'écran de connexion:

Nous avons créé l'écran "connexion" pour se connecter à la carte Arduino en utilisant le Bluetooth.

La conception de l'écran de connexion :

Nous avons ajouté des boutons de connexion et de déconnexion appelés « connecter » et « déconnecter », et une étiquette pour montrer les caractéristiques du Bluetooth.

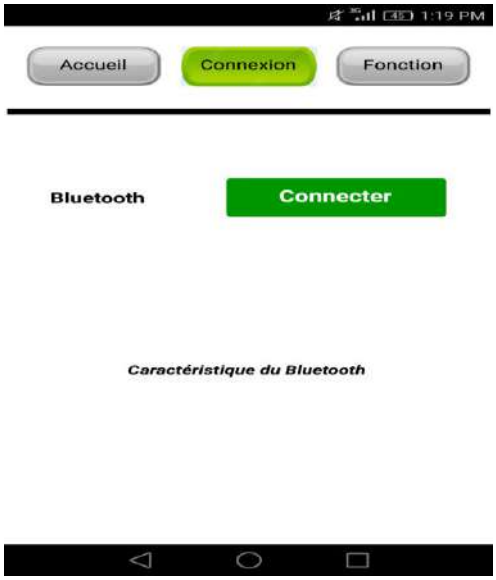


Figure 60 L'écran de connexion

Les blocs de l'écran de connexion :

Le type de bouton «connecter» est un sélecteur de liste, lorsque nous cliquons dessus, Bluetooth cherchera les adresses Bluetooth disponibles pour se connecter, puis il place les adresses dans une liste dans une nouvelle fenêtre pour en choisir une.

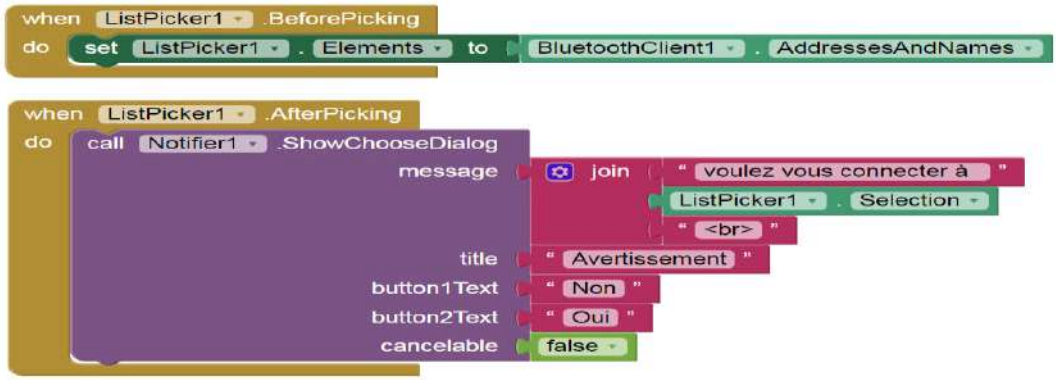


Figure 61 les blocs du bouton «connecter».

Juste après avoir sélectionné l'adresse, le notifiant ouvrira une fenêtre demandant si nous sommes sûrs de la sélection ou non, avec deux boutons "oui" et "non".

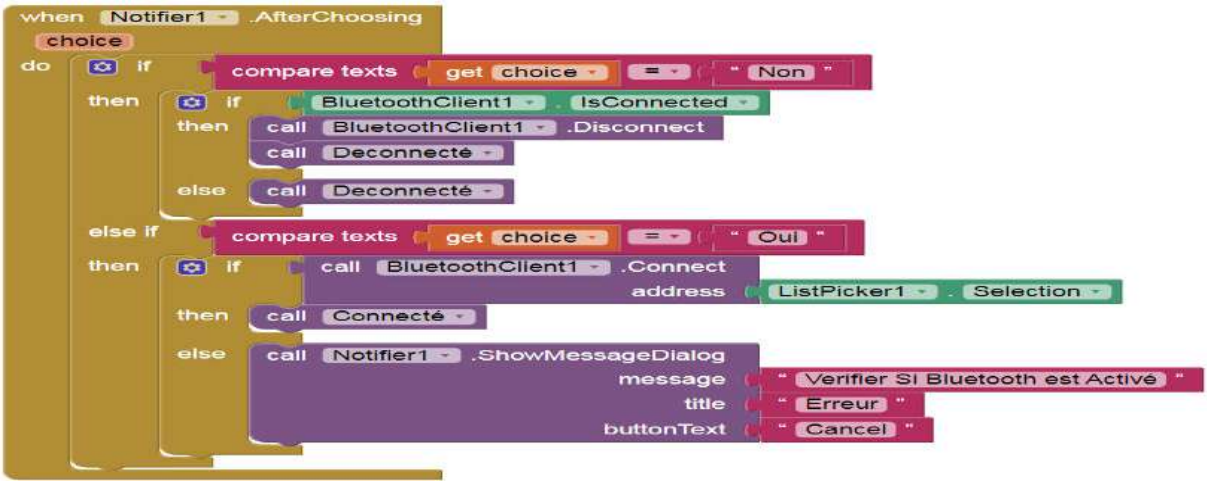


Figure 62 le bloc du notifiant.

Dans le bloc de notifiant nous avons utilisé deux fonctions "connecté" et "déconnecté", si on appelle la fonction "connecté" les caractéristiques Bluetooth et le bouton «déconnecter» apparaîtra, et si on appelle la fonction "déconnecté" le bouton «connecter» apparaîtra.

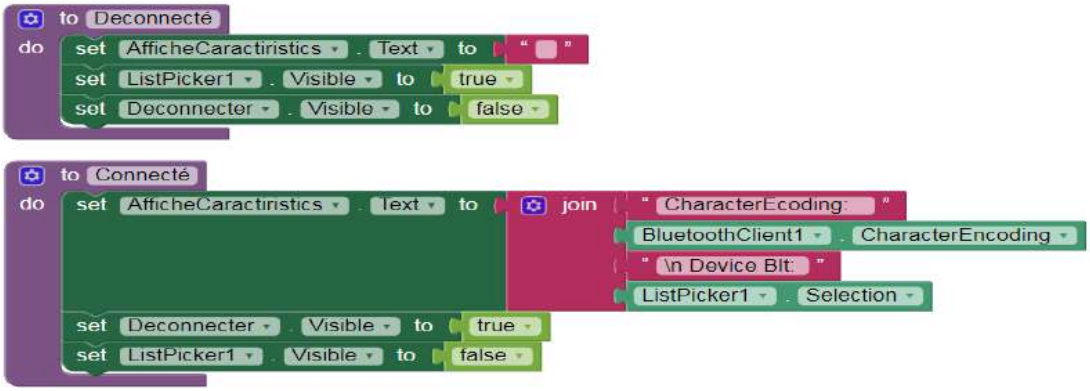


Figure 63 les blocs des fonctions «connecté» et «déconnecté».

Et le bouton «déconnecter» apparaîtra après la connexion, et si nous cliquons dessus, le Bluetooth se déconnectera et le bouton «connecter» réapparaîtra à sa place.

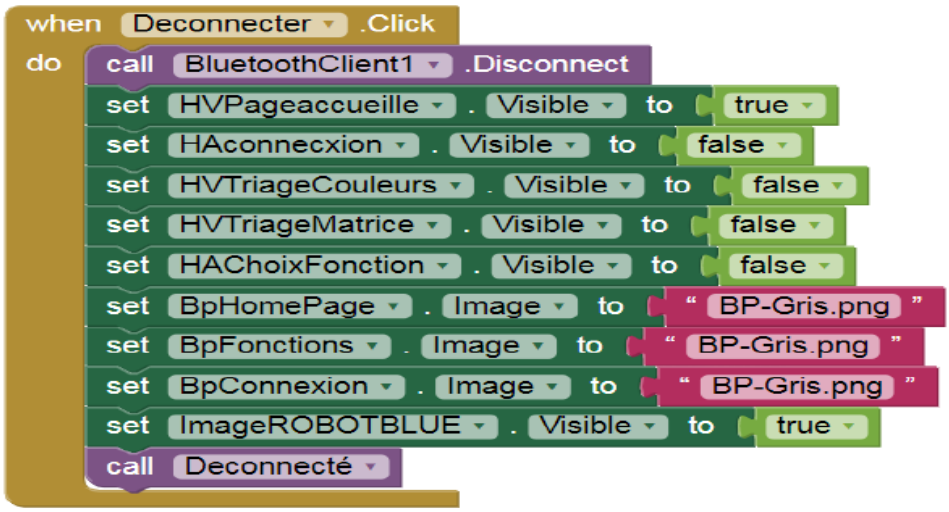


Figure 64 les blocs du bouton «déconnecté».

III.5.1. 3 L'écran de Fonctionnement:

Après la connexion à la carte Arduino, l'utilisateurentre à l'écran de fonctionnement afin de commander le robot.

La conception de l'écran de fonctionnement :

L'écran de fonctionnement contient deux boutons «TriageParCouleur» et «TriageMatriciel», pour choisir de quelle manière vous voulez commander le robot.

« triageMatriciel » fait référence au triage par les angles programmé.

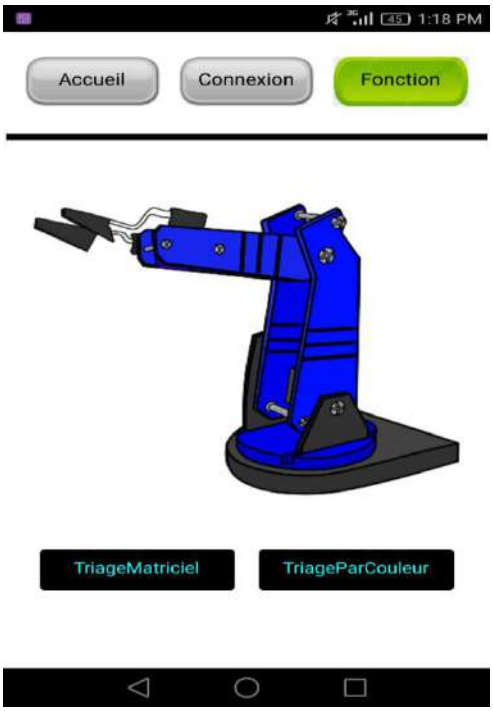


Figure 65 L'écran de "Fonction".

Les blocs de l'écran de fonctionnement :

Chaque bouton ouvre un autre écran différent:

- Le bouton « triageParCouleur » ouvre l'écran de triage par couleur
- Le bouton « triageMatriciel » ouvre l'écran de triage par les angles programmé

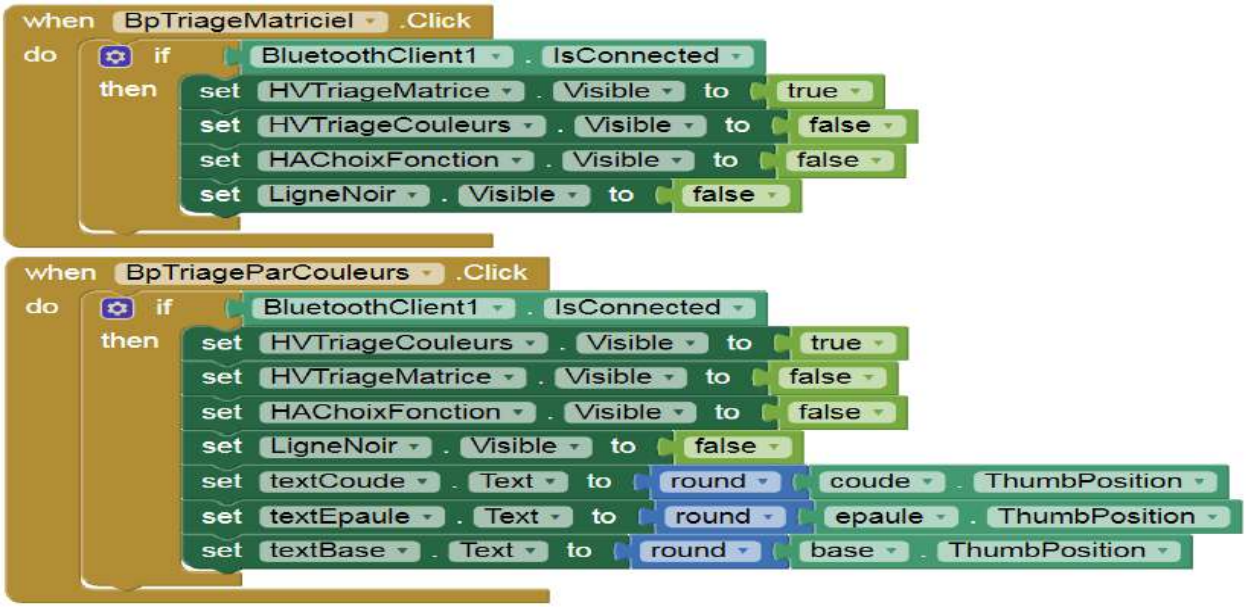


Figure 66 Les Boutons d'écran de Fonction.

III5.1.3.1 Triage par couleur :

La conception de l'écran de triage par couleur :

Cet écran contient:

- deux boutons pour ouvrir et fermer la pince.
- trois glissières pour commander la base, l'épaule et le coude.
- un bouton d'enregistrement pour enregistrer la position actuelle de chaque moteur.
- deux LED montrent quelle phase d'enregistrement sommes-nous.
- un bouton de démarrage pour démarrer le robot en mouvement avec les positions enregistrées.
- un bouton d'arrêt pour arrêter tout fonctionne.

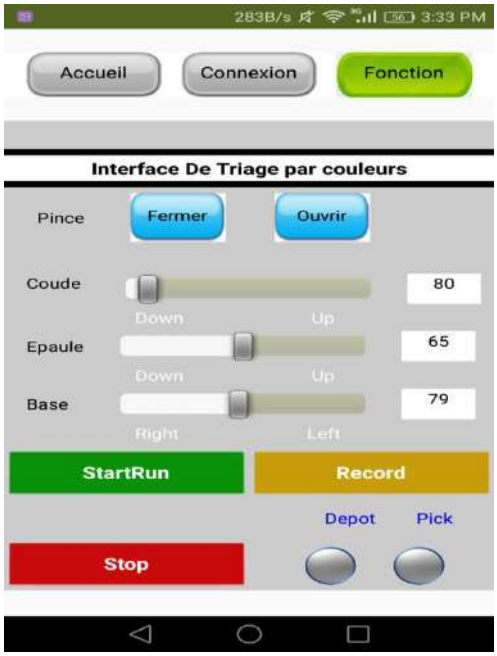


Figure 67 L'écran de triage par couleur.

Les Blocs de l'écran de triage par couleur :

- Lorsque nous cliquons sur le bouton «ouvrir», le Bluetooth envoie un code contenant "ouvrir" à la carte Arduino, et le dernier bouton deviendra vert et le bouton «fermer» deviendra gris.
- Lorsque nous cliquons sur le bouton "fermer", le Bluetooth envoie un code contenant "fermer" à l'Arduino, et le dernier bouton deviendra rouge.et le bouton "fermer" deviendra gris.

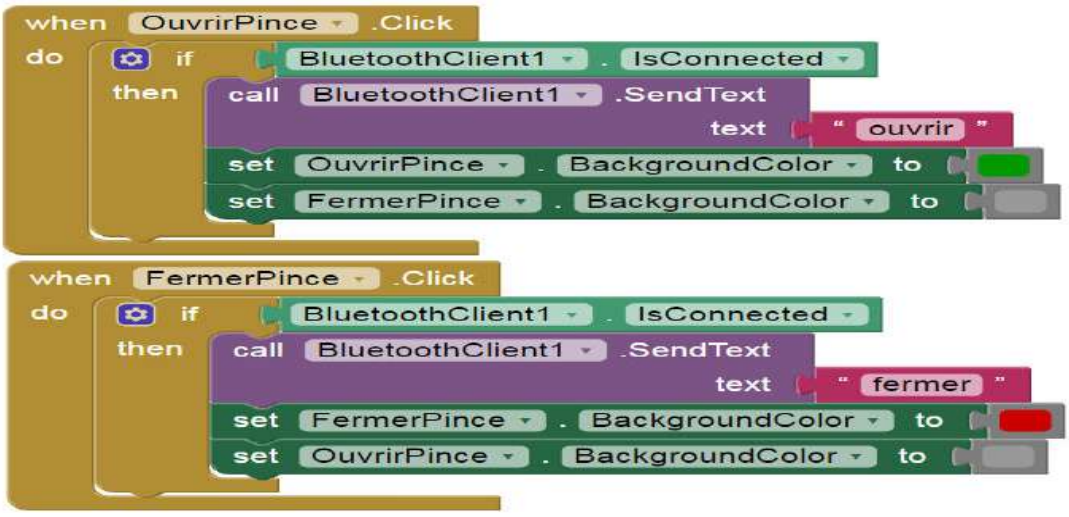


Figure 68 Les blocs des boutons «ouvrir» et «fermer».

Après que le programme a reçu le code, nous utilisons la fonction **write()**qui permet au servomoteur de tourner avec l'angle d'ouverture ou l'angle de fermeture qui sont déjà enregistrés

```
//.....................ouverture Servo pince..............
if (donnees.startsWith("ouvrir")) {
   Serial.println("ouverture de la pince ");
   pinch.write(50);
   donnees=" ";
}
//.....................fermeture Servo pince..............
if (donnees.startsWith("fermer")) {
   Serial.println("fermeture de la pince ");
   pinch.write(2);
   donnees=" ";
}
```

- Chaque glissière envoie à l'Arduino un code qui est transformé en deux parties qui sont: le préfixe S1, S2 ou S3 (le préfixe S1 fait référence à la base, S2 à l'épaule et S3 au coude) et lesautres chiffres sont la position de la glissière qui ne passe pas l'angle 180, par exemple si on change la position de glissière de coude à 180 degrés, il envoie le code "S2180"

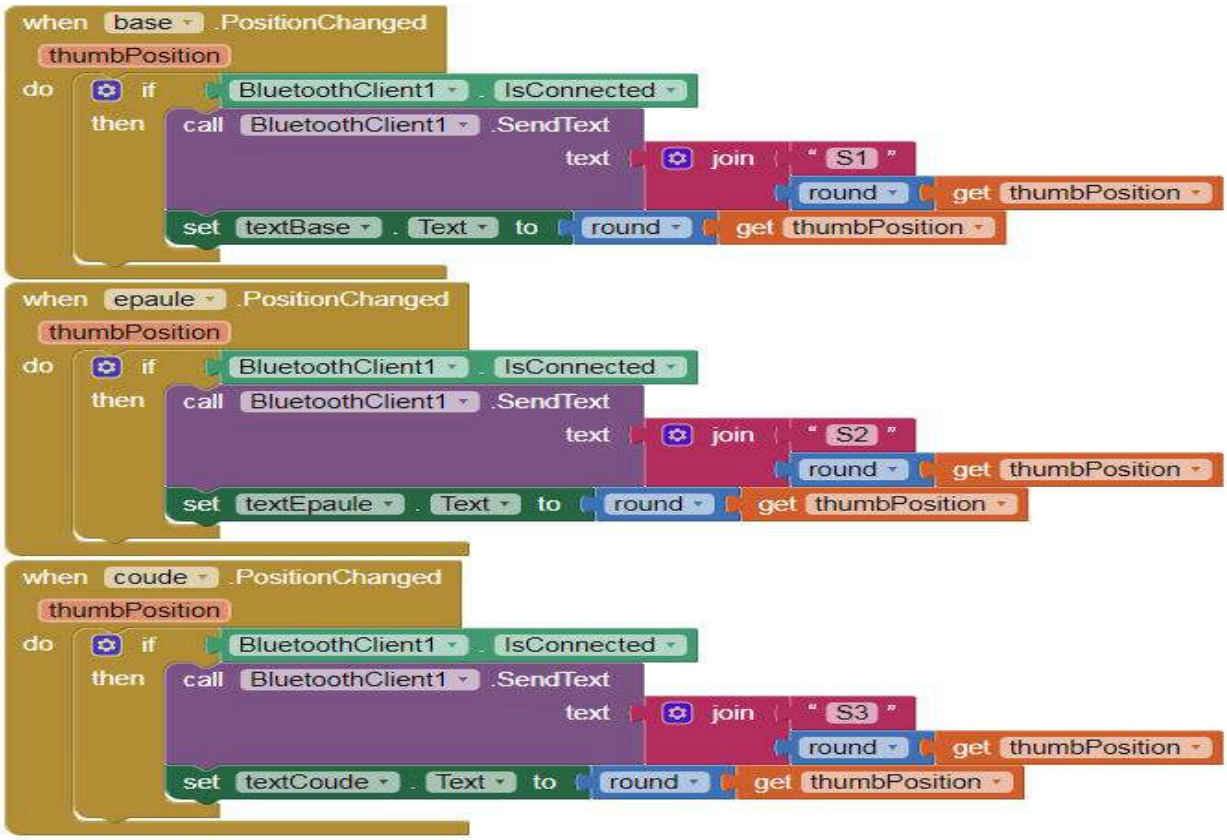


Figure 69 Les blocs des glissières.

Après que le programme a reçu le code, il vérifie le préfixe si il est S1, S2 ou S3 pour choisir le moteur spécifie, puis utiliser la fonction **write();** pour faire tourner le moteur avec l'angle reçu.

```
// ..... Servo base.....
  if (donnees.startsWith("S1")) {
     valBase = donnees.substring(2,donnees.length()).toInt();
     Serial.print("valbase: ");Serial.println(valBase);
     base.write(valBase);
     donnees=" ";
  }
  // ..... Servo epaule.....
  if (donnees.startsWith("S2")) {
     int valShoulder = donnees.substring(2,donnees.length()).toInt();
     Serial.print("valShoulder: ");Serial.println(valShoulder);
     shoulder.write(valShoulder);
     donnees=" ";
  }
  // ..... Servo coude.....
  if (donnees.startsWith("S3")) {
     int valElbow = donnees.substring(2,donnees.length()).toInt();
     Serial.print("valElbow: ");Serial.println(valElbow);
     elbow.write(valElbow);
     donnees=" ";
```

- Nous avons défini une variable appelée **REC** qui incrémente par +1à partir de 0, et onenvoyons le code "record" à la carte Arduino à chaque fois que nous cliquons sur le bouton «record», et lorsque le comptage atteint 3, il se réinitialise à 0.

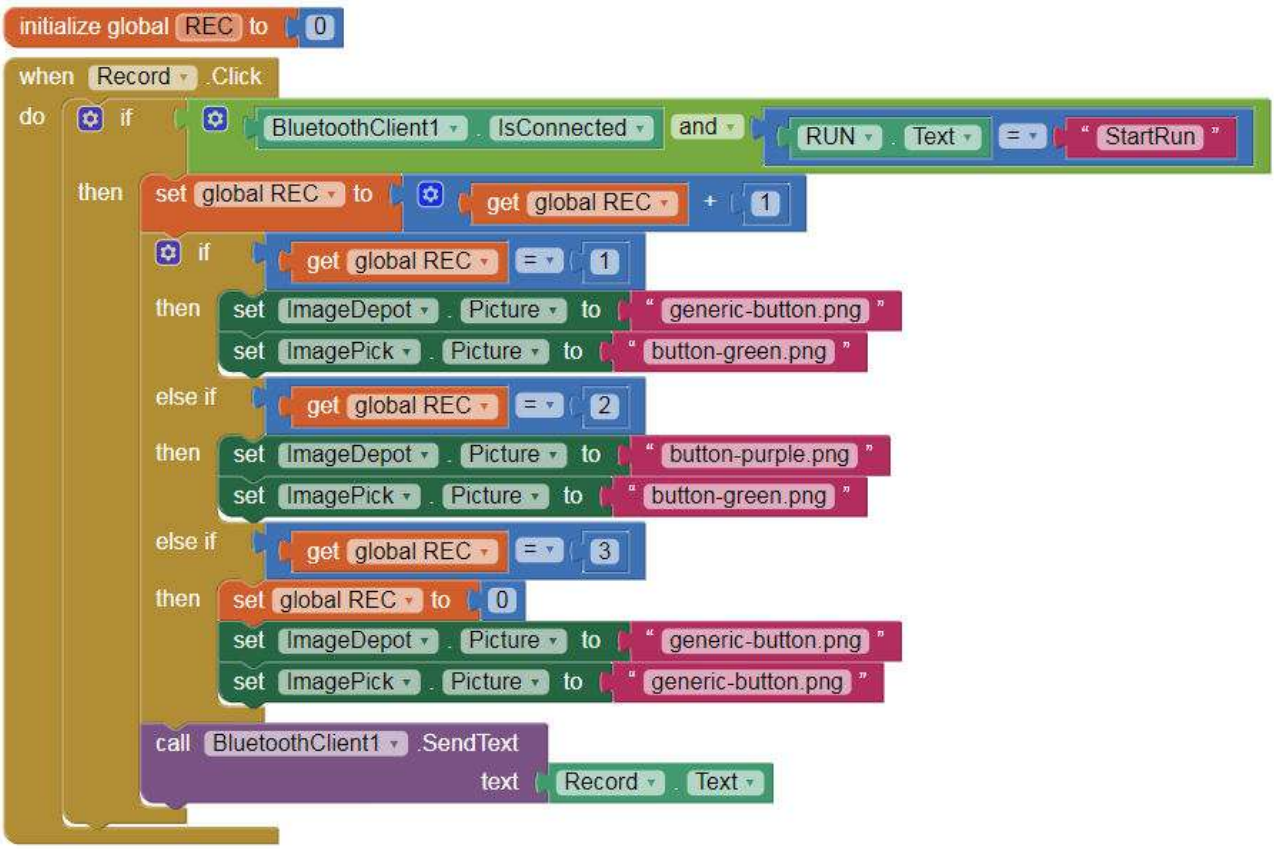


Figure 70 le bloc du bouton «record»

Sur le programme on a définit un entier appelé « recCounter » qui incrémente par +1 à chaque fois le programme reçoit le code « record ».

```
int recCounter = 0;
```

On a créé un sous programme « testrecCounter » qui vérifié la valeur de « recCounter » si il égale à 0,1ou 2 pour définie la phase d'enregistrement, et dans chaque phase nous enregistrons les positions de tous les angles des moteurs. Le programme détaillé de ce sous programme est donné dans l'annexe.

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

- Lorsque nous cliquons sur le bouton «StartRun» le bouton devient un bouton «Pause» et envoie à l'arduinole code "RUN", et si nous cliquons sur le bouton «pause», il devient un bouton «StartRun» et envoie le code contient "pause".

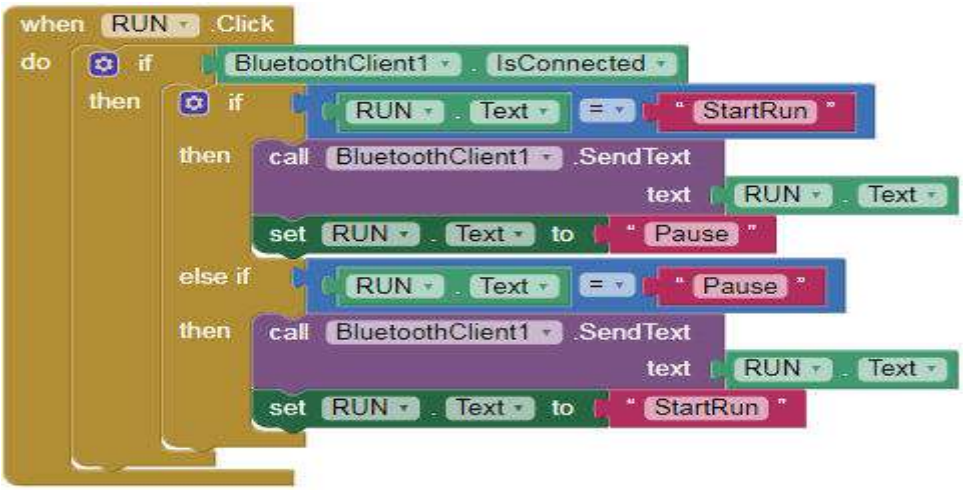


Figure 71 le bloc du bouton «StartRun»

Lorsque le code « StartRun » et reçu, on incrémente l'entier « runCounter » par +1. On a créé un sous programme « testRunCounter » qui exécute la fonction **write()** ; avec touts les moteurs pour les tourner avec les angles qui sont déjà enregistrés. Le programme détaillé de ce sous programme est donné dans l'annexe.

```
int runCounter = 0;

// ..............donnees --> "RUN".................
if (donnees.startsWith("StartRun")) {
   Serial.println("Mode run ");
   runCounter++;
   testRunCounter();
```

- Lorsque nous cliquons sur le bouton «Stop», il envoie à la carte Arduino le code « Stop », et les deux LED s'éteignent, et le bouton devient le bouton "StartRun".

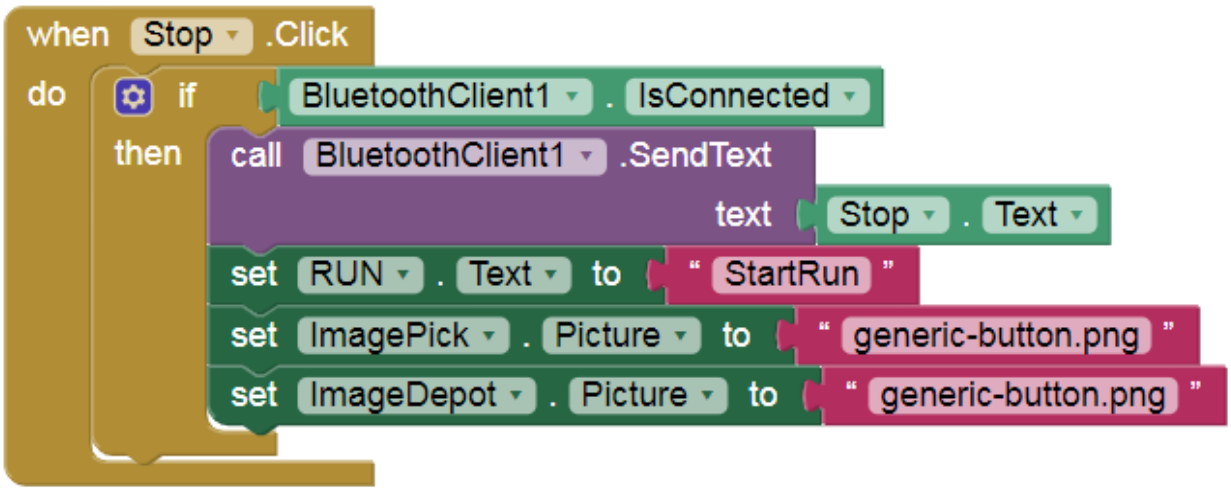


Figure 72 Le bloc du bouton «Stop».

Lorsque le code « Stop » et reçuon éteint les LED et tourne le bras dans sa position initiale avec la fonction **Home()** et on réinitialise l'enregistrement. Le programme détaillé est donné dans l'annexe.

```
// ..............donnees --> "STOP".................
if ( donnees == "Stop") {
 //Serial.println("Mode stop");
digitalWrite(ledPin1, LOW);
digitalWrite(ledPin2, LOW);
Home();
pick[0][0] = 80;
pick[0][1] = 20;
pick[0][2] = 120;
pick[0][3] = 0;
}
```

La figure III.45résume le fonctionnement de triage par couleursur un logigramme :

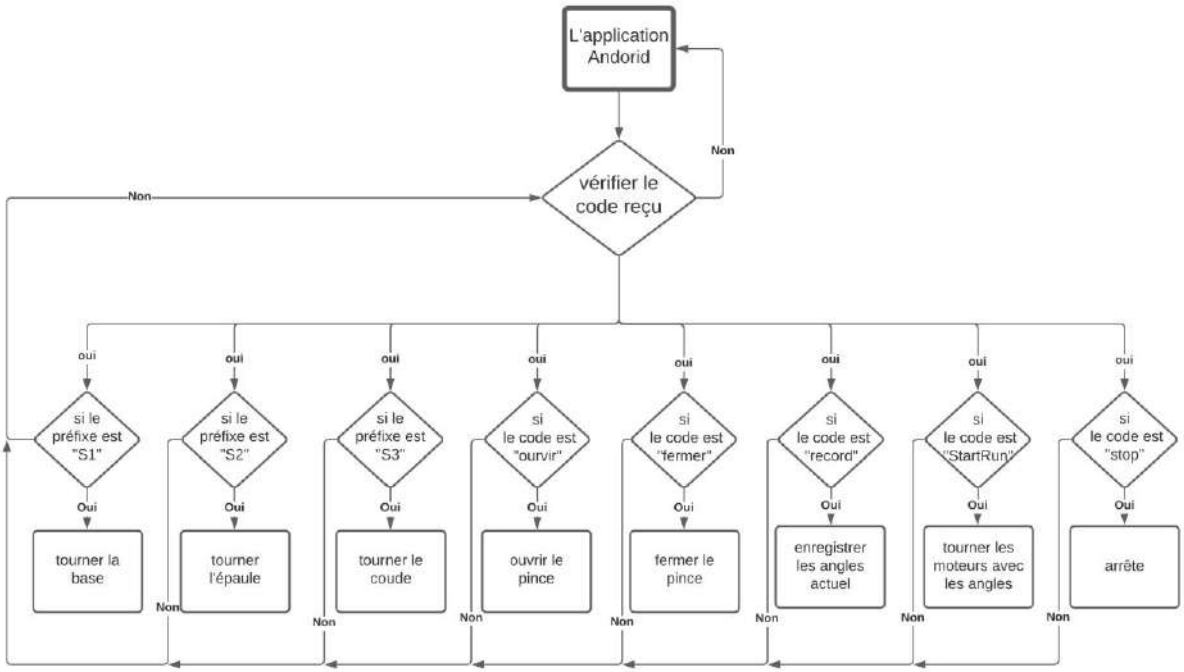


Figure 73 logigramme de fonctionnement de triage par couleur.

III.5.1.3.2Triage par les angles programmé :

La conception de l'écran de triage par les angles :

Cet écran contient 9 boutons (X, S, E, C, W, D, Z, A et Q).



Figure 1 L'écran de triage par les angles.

Les blocs de l'écran de triage par les angles :

Chaque bouton envoie un code contenant la lettre qui fait référence à sa position, si nous cliquons sur le bouton «X» il envoie le code « X » à l'Arduino. Et ainsi de suite avec les autres boutons.

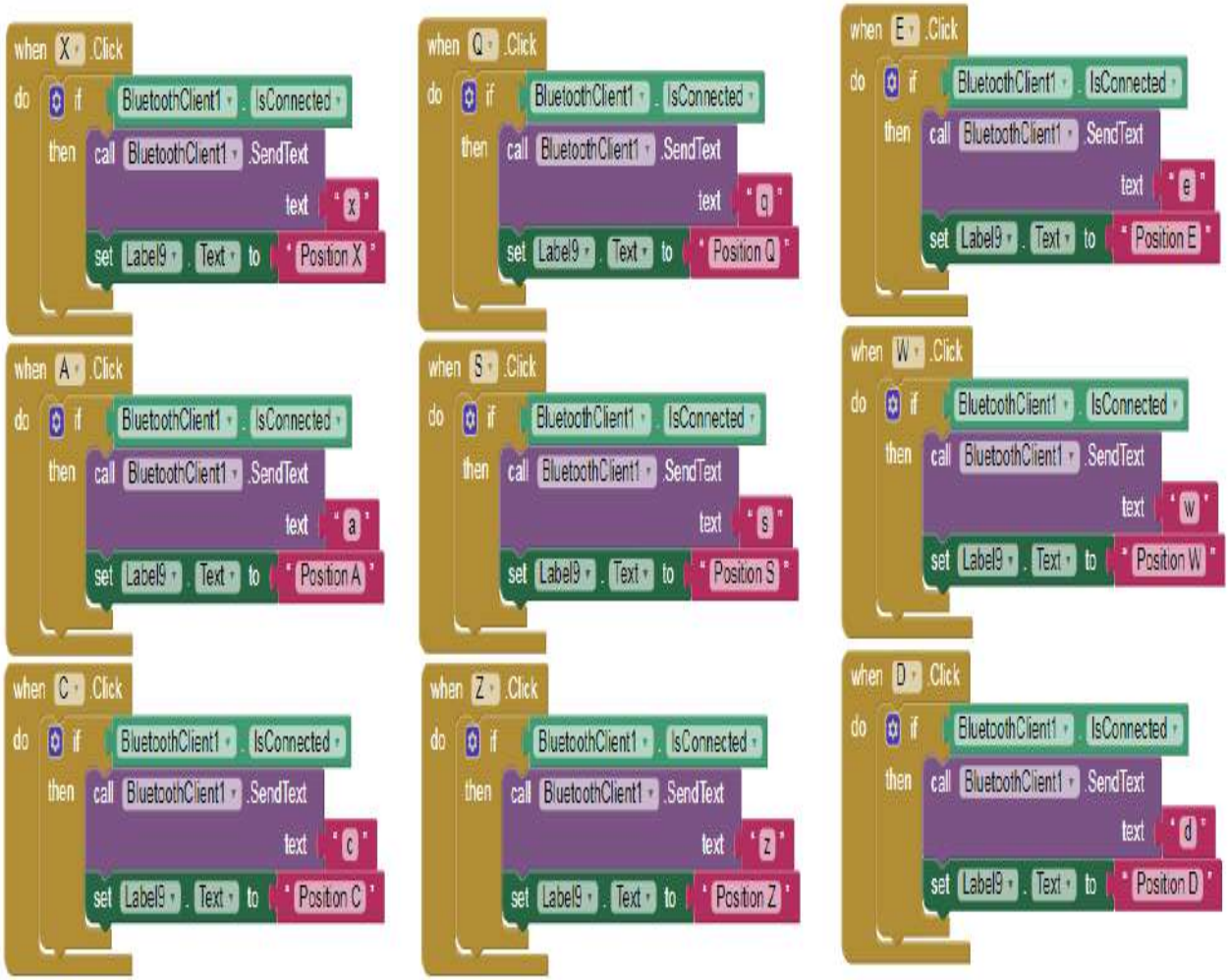


Figure 75 Les blocs des boutons de triage par les angles.

Après que le programme a reçu le code « X », il exécute la fonction **write ();** et des autres sous programme «Home» et «pick» pour tourner les moteurs avec des angles déjà sauvegardé pour la position X, avec une vitesse de 0.01s/degré. Le programme détaillé des sous programmes est donné dans l'annexe.

```
if(donnees=="x"){
    Serial.println("X Selectionne");
    Pick();
    delay(100);
    Home();
    delay(100);
    base.write(65);
    delay(hold);
    elbow.write(130);
    delay(hold);
    for (int i=35;i<=100;i++){
         shoulder.write(i);
         delay(10);
    }
    //shoulder.write(61);
    delay(200);
    pinch.write(16);
    delay(500);
    Home();
   }
```

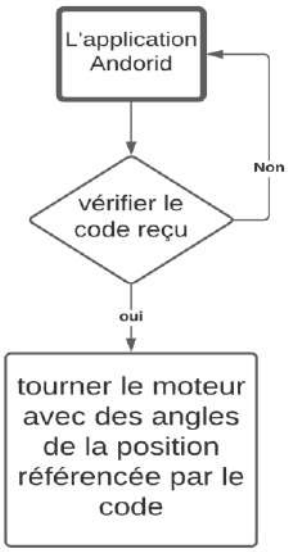


Figure 76 logigramme de fonctionnement de triage par les angles.

III.5.2 Commande par PC:

Pour commander le robot par pc, nous avons besoin d'une liaison USB avec la carte Arduino,et nous utilisons le logiciel IDE qui a une fenêtre appelée le moniteur série qui nous permet d'insérer des commandes sur la carte Arduino pour les exécuter. Pour permettre au moniteur série de communiquer avec la carte Arduino, notre programme doit inclure la commande "Serial.begin (9600);".

```
Serial.begin(9600);
```

La valeur 9600 est appelée «vitesse de transmission», C'est à quelle vitesse les données doivent être envoyées.

Les commandes que nous pouvons utiliser sont les mêmes codes que l'application envoie à la carte Arduino, alors ils ont le même fonctionnement et même programme qui est sur l'annexe :

- pour déplacer le robot vers n'importe quelle position on insère le codeen forme "S1105" :
  - "S1" fait référence à un moteur spécifique (S1 pour la base, S2 pour l'épaule, S3 pour le coude er S4 pour la pince).
  - Le reste des chiffres «105» représente le degré de l'angle souhaité.
- Pour déplacer le robot vers une position spécifique à l'intérieur de la matrice en insérant la lettre (X, S, E, C, W, D, Z, A ou Q) qui fait référence à la position souhaitée.

III.5.3 Commande par un potentiomètre :

Pour commander les servomoteurs, on utilise le potentiomètre et les 6 boutons sur la carte de commande:

- 4 boutons permettent le potentiomètre de faire tourner un servomoteur, et chaque bouton fait référence à un servomoteur (la base, l'épaule, le coude ou le pince). si nous appuyons un bouton le potentiomètre pourra déplacer le moteur qui est lié à ce bouton.
- 1 bouton « record » pour enregistrer les positions du chaque moteur.
- 1 bouton « Start » pour démarrer après l'enregistrement.

Avec deux LED (rouge et vert) pour montrer les phases d'enregistrement.

On a programmé la carte Arduino pour lire les valeurs du potentiomètre après nous cliquons sur un bouton de servomoteur.

On a également utilisé la fonction map() qui permet la conversion (ou le redimensionnement) des valeurs lues pour les adapter aux valeurs des servomoteurs (0 : 180), et nous avons terminé en tournant le moteur avec l'angle déjà converti avec la fonction write ().

```
while (digitalRead(basePin) == HIGH) {
  valBase = analogRead(potpin);
  valBase = map(valBase, 0, 1023, 0, 165);
 Serial.print(("valBase: "));
 Serial.println(valBase);
  base.write(valBase);
  delay(1);
}
```

Le programme détaillé des autres boutons des moteurs est donné dans l'annexe.

Pour les boutons record et « start » on utilise la fonction **recording()** qui nous avons crée.

```
recording();
```

Cette fonction détecte si le bouton de record est cliqué afin qu'il incrémente la valeur de REC et enregistre les angles actuels à l'aide de la fonction testrecCounter(), et détecte également si le bouton de « start » est cliqué pour faire tourner les moteurs avec les angles enregistrés en utilisant la fonction testrunCounter(). Ce fonction est bien détaillé dans l'annexe.

```
void recording() {
  recState = digitalRead(recPin);
  runState = digitalRead(runPin);
  if (recState != recLast) {
    if (recState == HIGH) {
      recCounter++;
//      Serial.println(recCounter % 3);
      delay(200);
      testrecCounter();
   }
  }
  recLast = recState;
  if (runState != runLast) {
    if (runState == HIGH) {
      runCounter++;
      delay(2000);
    }
    testRunCounter();
}
  runLast = runState;
}
```

III.6 Fonctionnement et résultats :

La **Figure III.49** montre les axes mécaniques de rotation, nous pouvons donc voir qu'il y a des positions qui sont dangereuses pour le robot ou le câblage. Le robot ne doit tourner que dans les positions nécessaires.

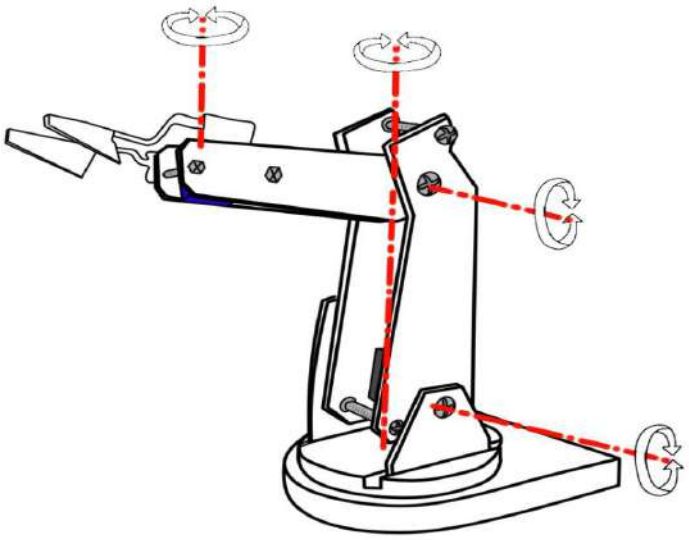


Figure 77 Les axes mécaniques de rotation.

Donc nous avons limité les angles de chaque moteur pour éviter ce problème.

Les organes	Degrés min	Degrés max
La base	0	165
L'épaule	40	90
Le coude	75	175
La pince	0	30

Tableau 1.Les angles max et min de chaque servomoteur.

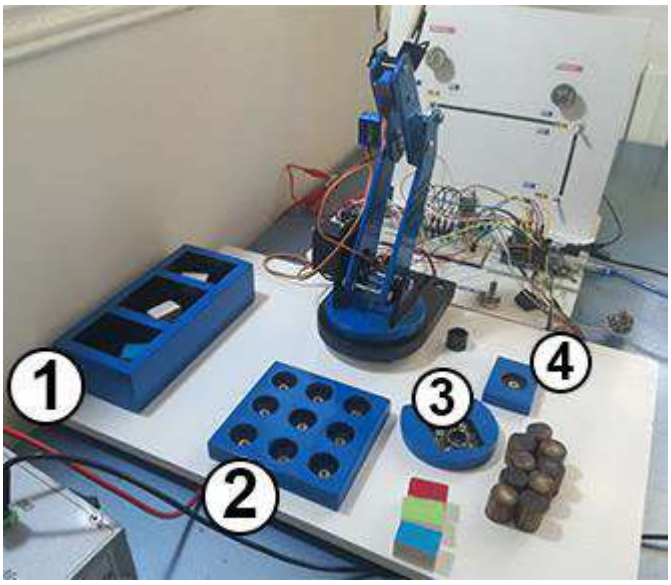


Figure 78 : L'environnement de robot

1. Le Panier.
2. Matrice des positions.
3. Le capteur de couleur.
4. La position pour ramasser les pièces.

Nous allons expliquer comment le robot peut effectuer les deux tâches avec différents moyens de contrôle:

Tâche 1 : Le triage par couleur

- **Par applications :**

1. Ouvrir l'application
2. Sur l'écran de triage par couleur, on utilise les glissières pour ramasser la pièce colorée.
3. Enregistrer les angles des organes en cliquant une fois sur le bouton "record".
4. Utiliser les glissières pour amener la pièce au capteur de couleur.
5. Enregistrer les angles des organes en cliquant la bouton « record » pour la deuxième fois.
6. Cliquer le bouton « StartRun » pour démarrer le cycle de triage automatiquement.

Le bras va détecter les couleurs des pièces et les placer sur le bon panier. Le programme de ce triage est détaillé en annexe.

- **Par PC :**

Entrer uniquement la commande « triagecouleur » sur le moniteur série et le bras va trier automatiquement les pièces.

- **Par potentiomètre :**

1. Utiliser les boutons du moteur pour permettre au potentiomètre de faire tourner les moteurs.
2. Ramasser la pièce.
3. Enregistrer les angles des organes en cliquant une fois sur le bouton "save".
4. amener la pièce au capteur de couleur.
5. Enregistrer les angles des organes en cliquant le bouton « save » pour la deuxième fois.
6. Cliquer le bouton « run » pour démarrer le cycle de triage automatiquement

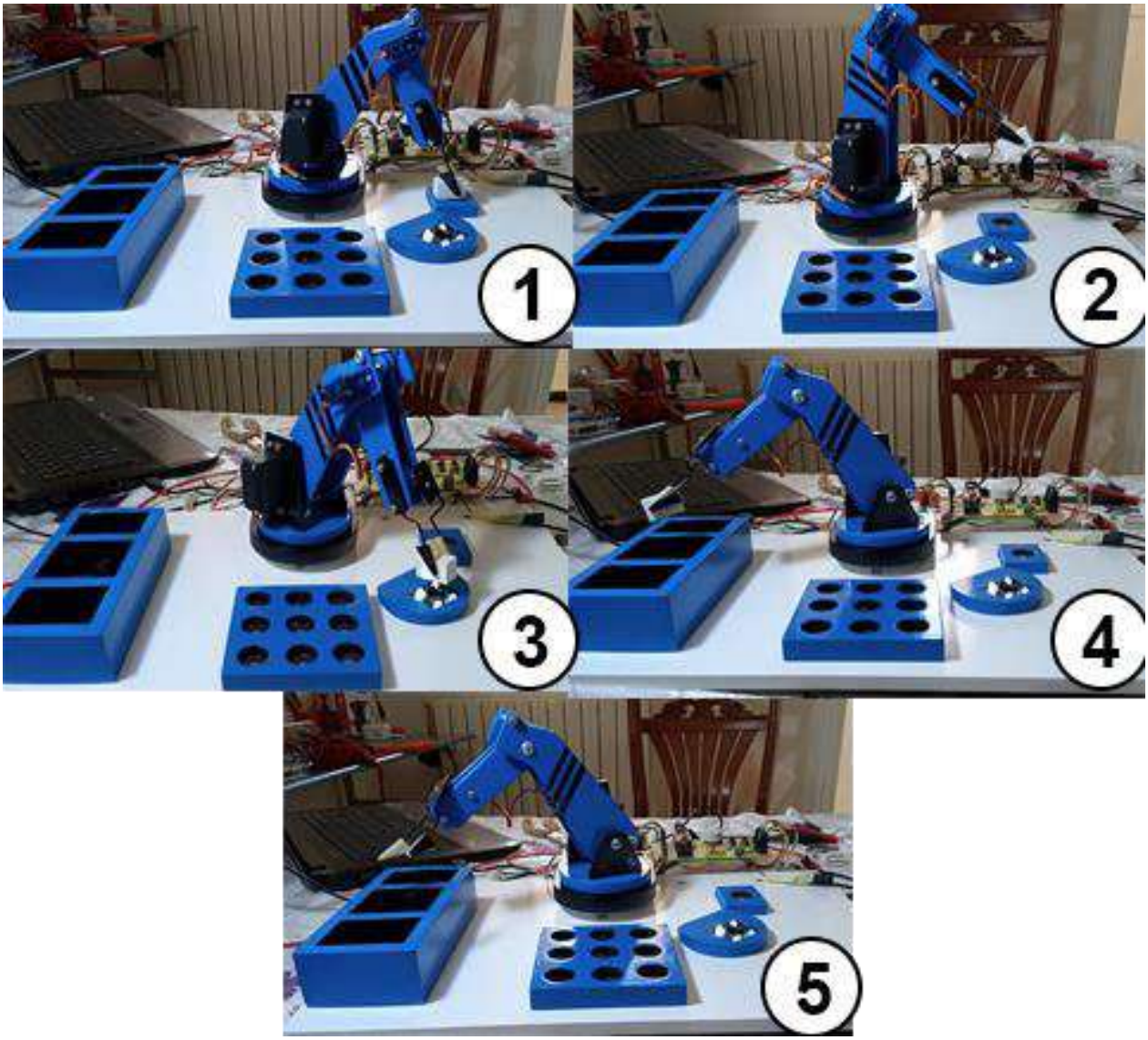


Figure 79Test du bras manipulateur pour trier par couleur.

Tâche 2 : triages par les angles

- **Par applications :**

1. Ouvrir l'application.
2. Sur l'écran de triage Matriciel, cliquez sur le bouton de position souhaitée.

Le bras va prendre la pièce et la mettre dans la position sélectionnée automatiquement.

- **Par PC :**

Insérer uniquement la position souhaitée (X, S, E, C, W, D, Z, A ou Q) sur le moniteur série et le bras va mettre la pièce automatiquement.

➢ Par potentiomètre :

1. Utiliser les boutons des moteurs pour permettre le potentiomètre de tourner les moteurs.
2. Ramasser la pièce.
3. Enregistrer les angles des organes en cliquant une fois sur le bouton « save ».
4. amener la pièce à la position souhaitée.
5. Enregistrer les angles des organes en cliquant le bouton « save » pour la deuxième fois.
6. Cliquer le bouton « run » pour démarrer le cycle automatiquement.

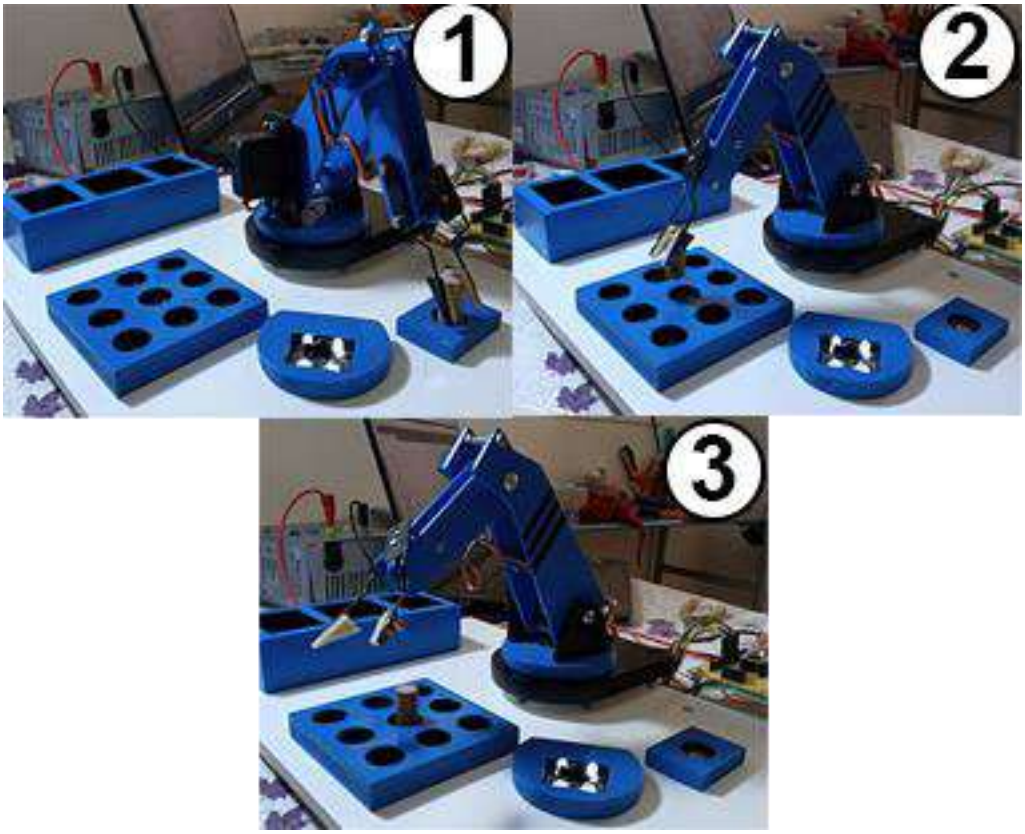


Figure 80 Test du bras manipulateur pour trier par les angles des organes.

Si nousvoulonsd'arrêter tous les tâches, nous pouvons cliquer sur le bouton « Arrêter » de l'application, ou cliquer pour la deuxième fois sur le bouton "run" sur la carte de commande, ou simplement entrer la commande « stop » sur le moniteur série

.

III.7 Conclusion :

Dans la première partie de ce chapitre, nous avons donné un bref aperçu de notre robot. Puis nous avons montré la réalisation mécanique du bras et la réalisationde la carte de commande, puis nous avons présenté la partie commande où nous avons expliqué les types de commande, et nous avons montré la réalisation de notre application étape par étape et sa relation avec le programme sur la carte Arduino, et nous avons terminé en montrant le fonctionnement du robot avec les résultats obtenus.

Conclusion Générale

Ce mémoire présente notre travail qui concerne l'étude et la réalisation d'un bras manipulateur à quatre degrés de liberté, avec l'hypothèse d'un environnement de travail sans obstacles. Après une exploration générale du domaine de la robotique, on a abordé le vif du sujet en étudiant les différents organes constituant notre bras manipulateur. Par la suite, on s'est penché l'élaboration des modèles géométrique et dynamique, qui ont pour rôle respectivement de générer des trajectoires de références, d'analyser les singularités et de commander et simuler le robot. Enfin, on a procédé à l'étude pratique et l'implémentation des techniques de commande prévues sur notre réalisation (par potentiomètres, application mobile, et triage par couleur).

Pour la réalisation pratique. La programmation est faite à base du langage C et a fait l'objet d'une implémentation sur la carte Arduino nano. Le contrôle du bras manipulateur se fait à l'aide d'une application mobile réalisée (sans fil), une commande manuelle par potentiomètres et triage par couleur.

Le manuscrit a été organisé en trois chapitres. Le premier présente des généralités. La deuxième porte sur la description des parties matériel et logiciel du bras manipulateur étudié. Le troisième chapitre expose la conception et la réalisation du bras manipulateur.

L'étude et la réalisation de ce projet nous ont permis d'approfondir nos connaissances théoriques et pratiques, notamment en robotique et en automatique. Nous avons pu appréhender différents points importants comme le fait de travailler en équipe et de concevoir et mener à bien un projet.

Bien que ce modeste travail ait atteint ses objectifs, on estime qu'il pourrait être encore amélioré. En effet, les travaux décrits dans ce mémoire peuvent se poursuivre sur plusieurs voies. Tout d'abord, il serait intéressant d'envisager un environnement de travail avec obstacles, ce qui n'est pas le cas dans cette étude. Cela constitue une bonne extension de ce projet. Pour la réalisation du bras, il serait avantageux d'optimiser la construction en utilisant un matériau plus solide. On pourrait utiliser des servomoteurs offrant un couple moteur plus important, dans le but de soulever des objets plus lourds.

Références bibliographiques

[1] ALMUSAWI Ahmed R. J., KAPUCU Sadettin, DÜLGER Lale Canan « Robotic Arm Dynamic and Simulation with Virtual Reality Model (VRM) » The 3rd International Conference on Control, Decision and Information Technologies, Malte, Volume: 3, Avril 2016.

[2] BOUZIANE Fatima Zohra « Rétro-conception du bras horizontal de robot manipulateur de la cellule flexible (Tlemcen)», UNIVERSITE ABOU BEKR BELKAID-TLEMCEN Option Ingénierie des systèmes mécaniques productiques,(2013)

[3] PRIEL MARC «les robots industriels: caractéristiques, performances et choix »: Edition AFNOR (1990)

[4] W .KHALIL, E. DOMBRE „Modélisation et commande des robots", Hermès, Paris,1988.

[5] EYRAUD, Charles-Henri. « Horloges astronomiques au tournant du XVIIIe siècle : de l'à-peu-près à la précision ». Thèse de doctorat. Lyon 2, 2004.

[6] BOIMOND, Jean-Louis. « ROBOTIQUE » Cours, ISTIA, Université Angers, 2017.

[7]- https://www.arduino.cc/en/Guide/Introduction

[8]- https://www.arduino.cc/en/Products/Compare

[9]- Guide de mise en marche du module Bluetooth HC-05

[10]- https://www.gotronic.fr/pj2-tcs3200-fr-1446.pdf

[11]- http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf

[12]- https://www.arduino.cc/en/Main/Software

[13]- https://www.hubor-proteus.com/proteus-pcb/proteus-pcb/2-proteus.html

[14]- https://fritzing.org

[15]- https://appinventor.mit.edu/explore/content/what-app-inventor.html

Le programme de commande sur la carte Arduino:

```
#include <SoftwareSerial.h>
#include <Servo.h>
Servo base;
Servo shoulder;
Servo elbow;
Servo pinch;
SoftwareSerial Bluetooth(12, 13);               // Arduino(RX, TX) ---> HC-05 Bluetooth (TX, RX)
int potpin = A3;
int ledPin1 = A4;
int ledPin2 = A5;
const int s2 = A1;
const int s3 = A0;
const int out =A2;
unsigned int red = 0;
unsigned int green = 0;
unsigned int blue = 0;
int valBase = 80;
int basePin = 2;
int valShoulder = 40;
int shoulderPin = 3;
int valElbow = 120;
int elbowPin = 4;
int valPinch = 0;
int pinchPin = 5;

int runPin = 8;
int recState;
int runState;
int recLast;
int runLast;
int recCounter = 0;
int runCounter = 0;
long previousMillis = 0;
int pick[1][4];
int dep[1][4];
String donnees = "";
String detectCouleur;
char Data="";
char buf[2];
bool  stateRun=true;
int hold = 800;
//.....................................................................................
void setup() {
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(out, INPUT);
  pinMode(7, INPUT);
  pinMode(8, INPUT);
   pinMode(2, INPUT);
   pinMode(3, INPUT);
   pinMode(4, INPUT);
   pinMode(5, INPUT);
   base.attach(6, 550, 2700);
```

```
  shoulder.attach(9);
  elbow.attach(10);
  pinch.attach(11);
  elbow.write(valElbow);
  shoulder.write(valShoulder);
  base.write(valBase);
  pinch.write(valPinch);
  delay(100);
  Serial.begin(9600);     //Serial.print("runCounter: ");Serial.println(runCounter);
  Bluetooth.begin(9600);
  Bluetooth.setTimeout(10);
  delay(20);
}
//.....................................................................
void loop() {
 // if (Bluetooth.available() > 0) {
 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: ");Serial.println(donnees);
 if (donnees.startsWith("x")|| donnees.startsWith("s") || donnees.startsWith("e")|| donnees.startsWith("c")
|| donnees.startsWith("c")|| donnees.startsWith("w")|| donnees.startsWith("d")
|| donnees.startsWith("z")|| donnees.startsWith("a") || donnees.startsWith("q")) {
        testMatriciel();   // le probleme se trouve la
        donnees=" ";
     }
   // ............. Servo base.............................
     if (donnees.startsWith("S1")) {
        valBase = donnees.substring(2,donnees.length()).toInt();
        Serial.print("valbase: ");Serial.println(valBase);
        base.write(valBase);
        donnees=" ";
     }
     // ............. Servo epaule.............................
     if (donnees.startsWith("S2")) {
        int valShoulder = donnees.substring(2,donnees.length()).toInt();
        Serial.print("valShoulder: ");Serial.println(valShoulder);
        shoulder.write(valShoulder);
        donnees=" ";
     }
     // ............. Servo coude.............................
     if (donnees.startsWith("S3")) {
        int valElbow = donnees.substring(2,donnees.length()).toInt();
        Serial.print("valElbow: ");Serial.println(valElbow);
        elbow.write(valElbow);
        donnees=" ";
     }
     //.....................ouverture Servo pince..............
```

```
  if (donnees.startsWith("ouvrir")) {
     Serial.println("ouverture de la pince ");
     pinch.write(50);
     donnees=" ";
  }
  //.....................fermeture Servo pince..............
  if (donnees.startsWith("fermer")) {
     Serial.println("fermeture de la pince ");
     pinch.write(2);
     donnees=" ";
  }
  // ..............donnees --> "SAVE".................
  if (donnees.startsWith("Record")) {
      Serial.println("Mode record ");
      recCounter++;
      testrecCounter();
   }
  // ..............donnees --> "RUN".................
  if (donnees.startsWith("StartRun")) {
     Serial.println("Mode run ");
     runCounter++;
     testRunCounter();
  }
  // ..............donnees --> "PAUSE".................
  if (donnees.startsWith("Pause")) {
      Serial.println("Mode pause ");
   }
   Home();
   // ..............donnees --> "STOP".................
   if ( donnees == "Stop") {
    Serial.println("Mode stop");
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, LOW);
    Home();
    pick[0][0] = 80;
    pick[0][1] = 20;
    pick[0][2] = 120;
    pick[0][3] = 0;
   }
}
//.....................triage par couleurs...............
  if (donnees.startsWith("triagecouleur")) {
     Serial.println(".........> La couleur detectee est: ");
     Serial.println("");
     DetecteCouleurPieces();
     donnees=" ";
  }
while (digitalRead(basePin) == HIGH) {
  valBase = analogRead(potpin);
  valBase = map(valBase, 0, 1023, 0, 165);
```

```
 Serial.print(F("valBase: "));
 Serial.println(valBase);
  base.write(valBase);
  delay(1);
}
while (digitalRead(shoulderPin) == HIGH) {
  valShoulder = analogRead(potpin);
  valShoulder = map(valShoulder, 0, 1023, 40, 100);
  Serial.print("valShoulder ");
  Serial.println(valShoulder);
  shoulder.write(valShoulder);
  delay(1);
}
while (digitalRead(elbowPin) == HIGH) {
  valElbow = analogRead(potpin);
  valElbow = map(valElbow, 0, 1023, 70, 170);
  Serial.print("valElbow ");
  Serial.println(valElbow);
  elbow.write(valElbow);
  delay(1);
}
while (digitalRead(pinchPin) == HIGH) {
  valPinch = analogRead(potpin);
  valPinch = map(valPinch, 0, 1023, 0, 60);
  pinch.write(valPinch);
    delay(1);
  }
  recording();
}
//.........................Recording.....................
void recording() {
  recState = digitalRead(recPin);
  runState = digitalRead(runPin);
  if (recState != recLast) {
    if (recState == HIGH) {
      recCounter++;
//      Serial.println(recCounter % 3);
      delay(200);
      testrecCounter();
   }
  }
  recLast = recState;
  if (runState != runLast) {
    if (runState == HIGH) {
      runCounter++;
      delay(2000);
    }
    testRunCounter();
}
  runLast = runState;
```

```
}
//.........................test runCounter.....................
void testRunCounter(){
    if (runCounter % 2 == 1 && stateRun==true) {
      Serial.println("Automation Starting");
      while (runCounter % 2 == 1) {
        runState = digitalRead(runPin);
        if (Bluetooth.available() > 0)  donnees = Bluetooth.readString();
        if ((runState == HIGH) || donnees.startsWith("Pause")) {
          runCounter++;
//          Serial.println(runCounter % 2);
          if (runCounter % 2 == 0) {
            break;
            }
        }
        Home();
        pinch.write(50);
        delay(hold);
        base.write(pick[0][0]);
        delay(hold);
        elbow.write(pick[0][2]);
        delay(hold);
        shoulder.write(pick[0][1]);
        delay(hold);
        pinch.write(pick[0][3]);
       delay(hold);
       Home();
       delay(1000);
       base.write(dep[0][0]);
       delay(hold);
       elbow.write(dep[0][2]);
       delay(  hold);
       shoulder.write(dep[0][1]);
       delay(hold);
       pinch.write(dep[0][3]);
       delay(hold);
      }
      Serial.println("Automation Stopped");
        Home();
        pinch.write(60);
        delay(hold);
        donnees=" ";
    }
}
//.........................test bouton Record..............
void testrecCounter(){
    if (recCounter % 3 == 0) {
      digitalWrite(ledPin1, LOW);
      digitalWrite(ledPin2, LOW);
      pick[0][0] = 80;
```

```
      pick[0][1] = 20;
      pick[0][2] = 120;
      pick[0][3] = 0;
      stateRun=false;
      Serial.println("Arret total");
    }
    if (recCounter % 3 == 1) { // Here we write cords for pick place in array.
      pick[0][0] = valBase;
      pick[0][1] = valShoulder;
      pick[0][2] = valElbow;
      pick[0][3] = valPinch;
      stateRun=true;
      Serial.println("prendre objet est acquit");
      digitalWrite(ledPin1, HIGH);
    }
    if (recCounter % 3 == 2) { // Here we write cords for deposit place in array.
      dep[0][0] = valBase;
      dep[0][1] = valShoulder;
      dep[0][2] = valElbow;
      dep[0][3] = valPinch;
      stateRun=true;
      Serial.println("Depot objet est acquit");
      digitalWrite(ledPin2, HIGH);
    }
}
//.....................................Home.....................
void Home() {
  shoulder.write(50);
  delay(200);
  elbow.write(130);
  delay(200);
  base.write(80);
  delay(200);
}
//..................................testmatriciel.....................
void testMatriciel(){
    if(donnees=="q"){
        Serial.println("Q selected");
        Pick();
        delay(100);
        Home();
        delay(100);
        base.write(89);
        delay(hold);
        elbow.write(167);
        delay(hold);
         for (int i=35;i<=85;i++){
            shoulder.write(i);
            delay(10);
        }
```

```
  //shoulder.write(69);
   delay(200);
   pinch.write(14);
   delay(500);
   Home();
}
if(donnees=="w"){
   Serial.println("W Selectionne");
   Pick();
   delay(100);
   Home();
   delay(100);
   base.write(74);
   delay(hold);
   elbow.write(150);
   delay(hold);
    for (int i=35;i<=89;i++){
        shoulder.write(i);
        delay(10);
   }
   delay(200);
   pinch.write(14);
   delay(500);
   Home();
}
if(donnees=="e"){
   Serial.println("E Selectionne");
   Pick();
   delay(100);
   Home();
   delay(100);
   base.write(55);
   delay(hold);
   elbow.write(166);
   delay(hold);
    for (int i=35;i<=80;i++){
        shoulder.write(i);
        delay(10);
   }
   delay(200);
   pinch.write(20);
   delay(500);
   Home();
}
if(donnees=="a"){
   Serial.println("A Selectionne");
   Pick();
   delay(100);
   Home();
   delay(100);
```

```
    base.write(88);
    delay(hold);
    elbow.write(151);
    delay(hold);
     for (int i=35;i<=90;i++){
         shoulder.write(i);
         delay(10);
    }
    delay(200);
    pinch.write(20);
    delay(500);
    Home();
 }
if(donnees=="s"){
    Serial.println("S Selectionne");
    Pick();
    delay(100);
    Home();
    delay(100);
    base.write(61);
    delay(hold);
    elbow.write(149);
    delay(hold);
    for (int i=35;i<=90;i++){
         shoulder.write(i);
         delay(10);
     }
     delay(1000);
     pinch.write(20);
     delay(500);
     Home();
   }
 if(donnees=="d"){
     Serial.println("D Selectionne");
     Pick();
     delay(100);
     Home();
     delay(100);
     base.write(71);
     delay(hold);
     elbow.write(176);
     delay(hold);
     for (int i=35;i<=83;i++){
          shoulder.write(i);
          delay(10);
     }
     //shoulder.write(51);
     delay(200);
     pinch.write(20);
     delay(500);
```

```
    Home();
 }
if(donnees=="z"){
    Serial.println("Z Selectionne");
    Pick();
    delay(100);
    Home();
    delay(100);
    base.write(88);
    delay(hold);
    elbow.write(132);
    delay(hold);
    for (int i=35;i<=97;i++){
         shoulder.write(i);
         delay(10);
    }
    //shoulder.write(62);
    delay(200);
    pinch.write(16);
    delay(500);
    Home();
  }
if(donnees=="x"){
    Serial.println("X Selectionne");
    Pick();
    delay(100);
    Home();
    delay(100);
    base.write(65);
    delay(hold);
    elbow.write(130);
    delay(hold);
    for (int i=35;i<=100;i++){
         shoulder.write(i);
         delay(10);
    }
    //shoulder.write(61);
    delay(200);
    pinch.write(16);
    delay(500);
    Home();
   }
 if(donnees=="c"){
    Serial.println("C Selectionne");
   Pick();
    delay(100);
    Home();
    //delay(100);
    base.write(76);
    delay(hold);
```

```
        elbow.write(133);
        delay(hold);
        for (int i=35;i<=98;i++){
             shoulder.write(i);
             delay(10);
        }
        //shoulder.write(62);
        delay(200);
        pinch.write(16);
        delay(500);
        Home();
       }
}
//.....................Lire couleur.............................
void lireCouleur(){
   //s2 et s3 à low pour lire la couleur rouge
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);
  red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  // s3 à HIGH pour lire la couleur bleu
  digitalWrite(s3, HIGH);
  blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

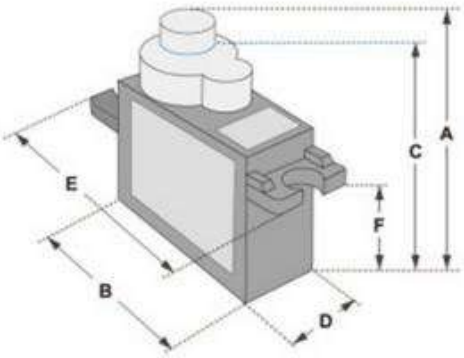
  //s2  à HIGH pour lire la couleur verte
  digitalWrite(s2, HIGH);
  green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

  Serial.print("R:");
  Serial.print(red, DEC);
  Serial.print(" G: ");
  Serial.print(green, DEC);
  Serial.print(" B: ");
  Serial.print(blue, DEC);
  delay(1500);
}
//.....................pos capteur de coiuleur TCS3200.........................
void TCS() {
  shoulder.write(85);
  delay(500);
  elbow.write(150); // elbow.write(174);
  delay(500);
  base.write(125);
  delay(500);
  shoulder.write(93);
  delay(200);
}
//.......................Pick.....................................
void Pick() { // This is fixed pick place.
  pinch.write(40);
  base.write(155);
```

```
  delay(700);
  shoulder.write(98);
  delay(200);
  elbow.write(154);
  delay(300);
  pinch.write(0);
  delay(100);
}
//.....................choisir couleur ..........................
void pickColor(){     // fonction permettant de positionner l'objet sur  le capteurTCS3200
  Pick();
  delay(500);
  TCS();
  lireCouleur();
  delay(100);
}
//.........................deposePieces............................
void DetecteCouleurPieces(){
  pickColor();
  if (red < blue && red < green && red<40 )                    //&& red<56
  {
   Serial.println(" - (....Red Color....)");delay(1500); RedColor();
  }
  else if (blue < red && blue < green && blue<57 )             //&& blue<50
  {
   Serial.println(" - (....Blue Color....)");delay(1500); BlueColor();
  }
  else if (green < red && green <= blue )                       //&& 40<green<60
  {
   Serial.println(" - (....Green Color....)");delay(1500); GreenColor();
  }
  else{
  Home();
  }
  }
//..................... couleur Rouge ..........................
void RedColor(){     // cette fonction nous permet juste l'essai de la position du capteurTCS3200
  //Home();
  shoulder.write(40);
  delay(200);
  base.write(16);
  delay(100);
  elbow.write(116);
  delay(500);
  shoulder.write(55);
  delay(500);
  pinch.write(60);
  delay(1000);
  Home();
}
```

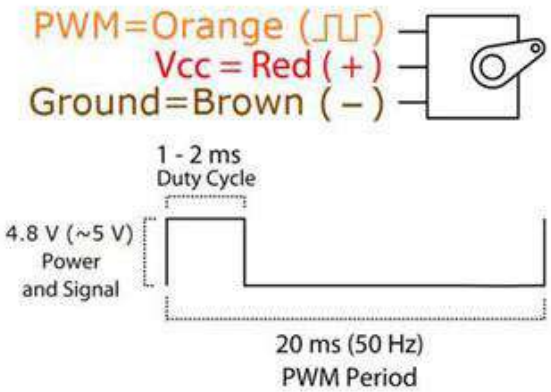
```
//..................... couleur Bleu ........................
void BlueColor(){     // cette fonction nous permet juste l'essai de la position du capteurTCS3200
  //Home();
  shoulder.write(40);
  delay(200);
  base.write(30);
  delay(100);
  elbow.write(98);
  delay(500);
  shoulder.write(70);
  delay(500);
  pinch.write(60);
  delay(1000);
  Home();
}
//..................... couleur Verte ........................
void GreenColor(){     // cette fonction nous permet juste l'essai de la position du capteurTCS3200
  //Home();
  shoulder.write(50);
  delay(100);
  elbow.write(132);
  delay(100);
  base.write(0);
  delay(500);
  shoulder.write(80);
  delay(500);
  pinch.write(60);
  delay(1000);
  Home();
}
```

Datasheet deservo moteur SG90:

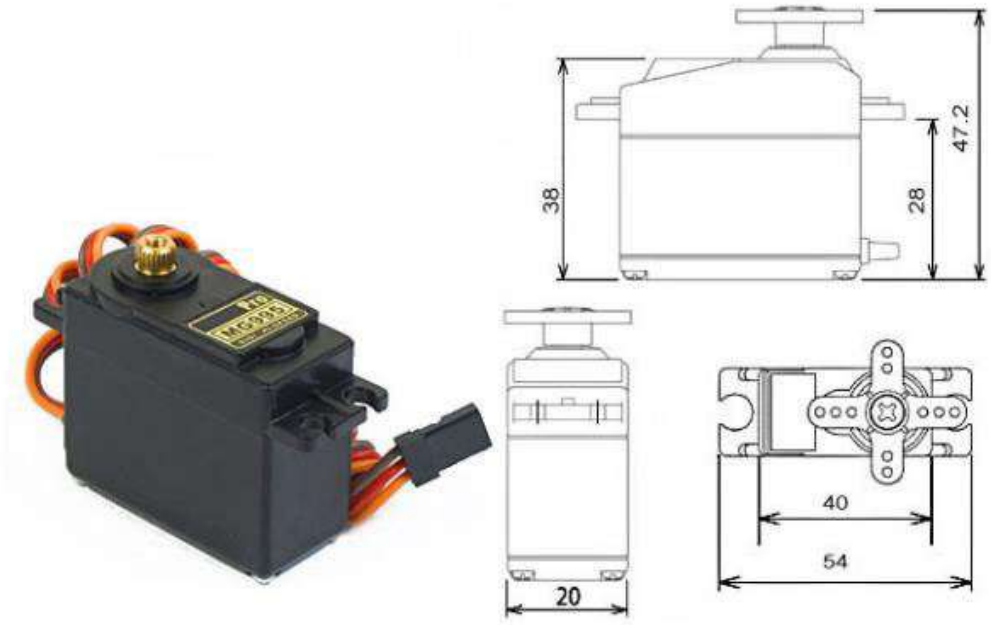


Dimensions & Specifications
A (mm) : 32
B (mm) : 23
C (mm) : 28.5
D (mm) : 12
E (mm) : 32
F (mm) : 19.5
Speed (sec) : 0.1
Torque (kg-cm) : 2.5
Weight (g) : 14.7
Voltage : 4.8 - 6

Position "0" (1.5 ms pulse) is middle, "90" (~2ms pulse) is middle, is all the way to the right, "-90" (~1ms pulse) is all the way to the left.



Datasheet de servomoteur MG990 :



The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec. This high-speed standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG995 Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

Specifications:

• Weight: 55 g

• Dimension: 40.7 x 19.7 x 42.9 mm approx.

• Stall torque: 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V)

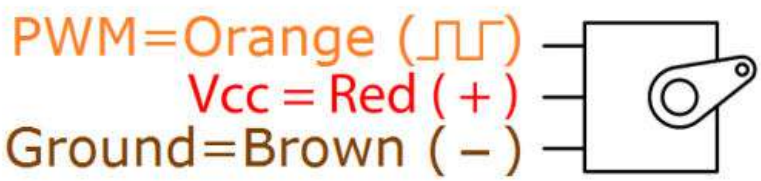
• Operating speed: 0.2 s/60º (4.8 V), 0.16 s/60º (6 V)

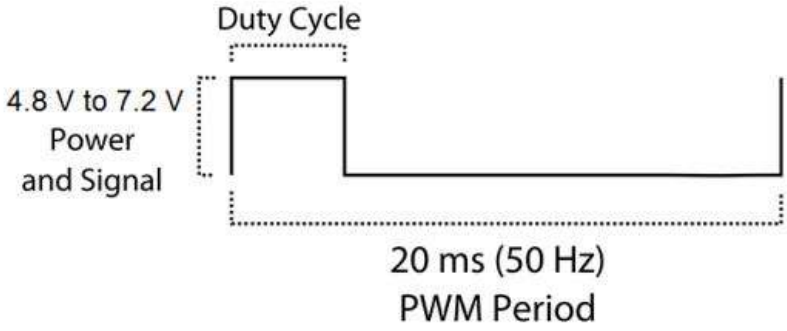
• Operating voltage: 4.8 V a 7.2 V

• Dead band width: 5 µs

• Stable and shock proof double ball bearing design

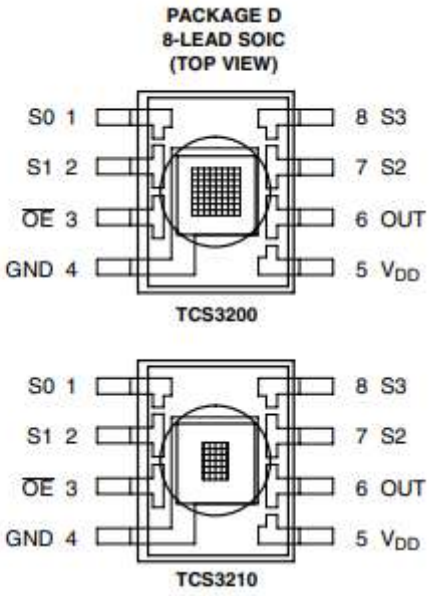
• Temperature range: 0 ºC – 55 ºC





Datasheet de capteur de couleur TCS2300:

- **High-Resolution Conversion of Light Intensity to Frequency**
- **Programmable Color and Full-Scale Output Frequency**
- **Communicates Directly With a Microcontroller**
- **Single-Supply Operation (2.7 V to 5.5 V)**
- **Power Down Feature**
- **Nonlinearity Error Typically 0.2% at 50 kHz**
- **Stable 200 ppm/°C Temperature Coefficient**
- **Low-Profile Lead (Pb) Free and RoHS Compliant Surface-Mount Package**



Description

The TCS3200 and TCS3210 programmable color light-to-frequency converters that combine configurable silicon photodiodes and a current-to-frequency converter on a single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance).

The full-scale output frequency can be scaled by one of three preset values via two control input pins. Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. Output enable ($\overline{OE}$) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line.

In the TCS3200, the light-to-frequency converter reads an 8 x 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters.

In the TCS3210, the light-to-frequency converter reads a 4 x 6 array of photodiodes. Six photodiodes have blue filters, 6 photodiodes have green filters, 6 photodiodes have red filters, and 6 photodiodes are clear with no filters.

The four types (colors) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance. All photodiodes of the same color are connected in parallel. Pins S2 and S3 are used to select which group of photodiodes (red, green, blue, clear) are active. Photodiodes are 110 µm x 110 µm in size and are on 134-µm centers.

Les dimensions de la carte Arduino nano:

