



Rapport Mini Projet

Robot Manipulateur



Élaboré par :

Nour Trabelsi

Ahmed Aziz Gaddour

Classe : L3AII

2021/2022

REMERCIEMENT

Au terme de ce projet nous tenons à exprimer nos respects et nos sincères remerciements à Madame. Chaouch Khadouja et Mr. Ben Abdallah Mohamed qui nous ont bien encadré à la réalisation de notre mini-projet ainsi de nous faire confiance.

Sommaire

Introduction générale	4
I. Etude de projet	5
1. Qu'est-ce qu'un robot Pick and place ?	5
2. Comment fonctionnent les robots pick and place	5
3. Types de robots.....	5
4. Applications pour les robots pick and place	6
5. Avantages des robots Pick and place	6
6. Nombre d'axes	6
7. Configuration et vision.....	7
8. Charge utile	7
II. Etude Pratique	8
1. Synthèse du projet	8
2. Les composants du robot.....	8
2.1 Motors pas à pas Nema 17.....	8
2.2. Stepper Driver A4988.....	9
2.3. PCA 9685.....	9
2.4. Servo Motors (MG996R/ SG90 Micro).....	10
2.5. Bluetooth HC 06.....	11
3. Conception du robot.....	11
3.1 . Le bras robotique	11
3.2. Le modèle de chariot.....	12
3.3. Le schéma électrique	13
4. Partie software.....	13
5. Conclusion.....	19

Listes des figures

Figure 1 : Motor pas à pas Nema 17	8
Figure 2 : Stepper Driver A4988.....	9
Figure 3 : câblage stepper et driver A4988.....	9
Figure 4 : CARTE PCA 9685.....	10
Figure 5 : Servo Motors.....	10
Figure 6 : câblage Bluetooth HC 06.....	11
Figure 7 : le schéma électrique de bras (proteus)	11
Figure 8 : la conception de bras.....	12
Figure 9 : Le conception de chariot en 3D	12
Figure 10 : le schéma ELECTRIQUE DE chariot.....	12
Figure 11 : Le schéma électrique	13
Figure 12 : le câblage électrique	13
Figure 13 : interface App Inventer	15
Figure 14 : code App Inventor.....	15
Figure 15 : application mobile (COMMANDE de bras)	16

Introduction générale

Dans le secteur industriel, le concept de pick and place désigne l'opération qui consiste à retirer de la marchandise d'un emplacement pour la placer à un autre endroit. Par exemple, emballer des chocolats au fur et à mesure qu'ils sortent des lignes de production.

C'est un processus généralement très fastidieux et répétitif. Cependant, nous sommes dans une époque où la logistique 4.0 et l'industrie 4.0 sont en pleine expansion, les entreprises se tournent vers la robotisation pour améliorer leur productivité. La raison en est que l'automatisation fait gagner du temps et élimine les processus à faible valeur ajoutée.

De ce fait, dans le cadre de notre mini-projet au sein de l'institut supérieur des études technologiques de Radès nous allons réaliser un robot manipulateur inspiré des robots pick and place.

Ce travail a été fait Pendant la première semestre de l'année universitaire 2021/2022, Il était assez bénéfique pour nous au niveau théorique et pratique.

I. Etude de projet

1. Qu'est-ce qu'un robot Pick and place ?

Les robots Pick and Place sont couramment utilisés dans les environnements de fabrication modernes. L'automatisation du prélèvement et du placement accélère le processus de prélèvement de pièces ou d'articles et de leur placement à d'autres emplacements. L'automatisation de ce processus permet d'augmenter les taux de production. Les robots de sélection et de placement gèrent des tâches répétitives tout en permettant aux travailleurs humains de se concentrer sur des travaux plus complexes.

2. Comment fonctionnent les robots pick and place

Généralement montés sur un support stable, ils sont positionnés pour atteindre différentes zones pour effectuer le travail. Ils utilisent des systèmes de vision avancés pour identifier, saisir et déplacer des objets d'un endroit à un autre. Avec une variété d'options de conception disponibles, les robots pick and place peuvent être configurés avec diverses options d'outillage en bout de bras pour une utilisation dans différentes applications, telles que l'assemblage, l'emballage ou le prélèvement de bacs. Par exemple, les robots pick and place peuvent être utilisés pour ramasser des articles pour une commande et les placer dans une boîte pour l'emballage, ou ils peuvent être utilisés pour ramasser les pièces nécessaires à l'assemblage et les déplacer vers l'emplacement suivant.

3. Types de robots

Il existe plusieurs types de robots pick and place, notamment :

Bras robotique – Les bras robotiques sont le type le plus courant de robots Pick and place. Un robot à bras robotisé à 5 axes peut être utilisé pour des applications de prélèvement et de placement standard où les objets sont prélevés et déplacés vers d'autres emplacements dans un seul plan.

Cartésien – Comme un bras robotique à 6 axes, les robots cartésiens fonctionnent dans plusieurs plans. Ces robots se déplacent selon trois axes orthogonaux (X, Y et Z) en utilisant des coordonnées cartésiennes. Ils peuvent être construits avec n'importe quel type d'actionneur linéaire et plusieurs types de mécanismes d'entraînement tels que des mécanismes à courroie, à billes ou à vis mère.

Delta – Souvent utilisés dans les applications où les robots prélèvent des éléments en groupes et les placent dans des modèles d'assemblage ou des conteneurs, les robots Delta disposent de technologies de vision avancées qui leur permettent de distinguer différentes tailles, formes et couleurs. Il existe plusieurs configurations de robots Delta, mais la plupart ont trois bras qui fonctionnent sur quatre axes. Ils ont des moteurs lourds fixés à un cadre, avec des bras légers connectés à des tiges de liaison avec des joints à chaque extrémité de chaque bras (généralement des rotules) pour permettre le mouvement.

Sélection rapide - Les robots de sélection rapide sont idéaux pour une utilisation dans des applications à volume moyen et élevé avec des références SKU à grande vitesse. Les robots de prélèvement rapide automatisent entièrement le processus de prélèvement, libérant ainsi la main-d'œuvre humaine pour qu'elle se concentre sur les activités à plus fort impact.

Collaboratif - Les robots collaboratifs augmentent le travail des humains en amenant les associés à choisir des emplacements et en guidant les associés à travers chaque tâche. En optimisant les itinéraires en temps réel et en gardant les associés sur leur tâche, les robots collaboratifs aident les associés à travailler plus efficacement.

4. Applications pour les robots pick and place

Les robots pick and place sont souvent utilisés dans la fabrication, mais sont également utilisés dans des applications telles que l'emballage, le ramassage de bacs et l'inspection. Voici un aperçu de quelques-unes des applications les plus courantes pour les robots pick and place et de leur utilisation.

Assemblage – Les robots de prélèvement et de placement utilisés dans les applications d'assemblage saisissent les pièces entrantes d'un emplacement, tel qu'un convoyeur, et placent ou fixent la pièce sur une autre pièce de l'article. Les deux pièces jointes sont ensuite transportées vers la zone d'assemblage suivante.

Emballage – Les robots de prélèvement et de placement utilisés dans le processus d'emballage saisissent les articles d'une source entrante ou d'une zone désignée et placent les articles dans un conteneur d'emballage.

Cueillette de bacs – Les robots de prélèvement et de placement utilisés dans les applications de ramassage de bacs saisissent des pièces ou des articles dans les bacs. Ces robots pick and place disposent généralement de systèmes de vision avancés leur permettant de distinguer la couleur, la forme et la taille pour choisir les bons articles, même dans des bacs contenant des articles mélangés de manière aléatoire. Ces pièces ou articles sont ensuite envoyés à un autre endroit pour assemblage ou emballage.

Inspection - Les robots Pick and Place utilisés pour les applications d'inspection sont équipés de systèmes de vision avancés pour ramasser des objets, détecter des anomalies et retirer des pièces ou des articles défectueux en les plaçant dans un endroit désigné.

5. Avantages des robots Pick and place

Les principaux avantages des robots Pick and place sont la vitesse et la cohérence. Les robots peuvent être personnalisés pour répondre à des exigences de production spécifiques, et ils sont facilement programmables afin de pouvoir être utilisés pour de multiples applications. Bien que les conceptions varient, les robots pick and place sont souvent de petite taille et légers, ce qui les rend idéaux pour une utilisation dans des applications où l'espace est limité.

Les robots de prélèvement et de placement contribuent à augmenter la production, en facilitant les processus de prélèvement et d'emballage dans les entrepôts. Ils offrent également un bon retour sur investissement aux fabricants. Dans les deux cas, les robots pick and place soulagent les employés et les opérateurs d'un travail monotone et répétitif, ce qui augmente la productivité tout en allégeant la pression physique sur les travailleurs humains qui résulte généralement de l'exécution de ces types de tâches.

6. Nombre d'axes

Le nombre d'axes détermine le degré de liberté et de mouvement du robot pick and place. Généralement, plus d'axes signifie plus de mouvement et de flexibilité, il est donc préférable d'avoir plus d'axes que moins. Pour les applications de traitement des commandes dans lesquelles les articles sont placés sur un convoyeur, un bac de prélèvement ou directement dans un conteneur d'emballage, le robot doit être équipé de quatre à cinq axes. Six axes ou plus sont préférables pour les applications dans lesquelles le robot doit tourner ou se déplacer linéairement pour gérer les tâches dans son enveloppe de travail.

7. Configuration et vision

Les robots pick and place peuvent être :

Un robot articulé ou scada (fixé avec des bras rotatifs sur trois axes verticaux)

Un robot sphérique fournissant un mouvement linéaire et deux mouvements de rotation

Un robot cylindrique fournissant un mouvement dans les axes horizontaux, verticaux et de rotation

Bien que la plupart des robots Pick and place soient flexibles, ils sont limités par leurs dimensions, les buses de manipulation des produits et la latitude de mouvement. Ces caractéristiques affectent l'endroit où ils peuvent être déployés et les éléments qu'ils peuvent gérer. De plus, le système de guidage visuel doit être suffisamment sophistiqué pour identifier plusieurs articles à partir d'un pool de SKU.

8. Charge utile

La charge utile est le poids maximum que le robot peut transférer d'un point à un autre. Il comprend le poids du produit à prélever et le poids de l'outillage en bout de bras du robot. Un robot pick and place doit être capable de soulever l'article le plus lourd de votre inventaire et d'étendre complètement son bras et de placer cet article avec précision.

II. Etude Pratique

1. Synthèse du projet

Dans cette partie, nous allons vous montrer comment nous avons créé notre robot manipulateur qu'il est composé d'un chariot mobile et un bras robotique imprimé en 3D. nous pouvons contrôler le robot manipulateur avec l'application Android personnalisée. En plus de cela, l'application dispose des boutons pour contrôler la vitesse de chariot et de bras séparément.

Néanmoins, la caractéristique distincte de ce robot est la possibilité de stocker les mouvements puis de les répéter automatiquement, En utilisant le bouton Enregistrer, nous pouvons simplement stocker les positions des moteurs pour chaque étape. Ensuite, il suffit de cliquer sur le bouton Exécuter et le robot répétera automatiquement les mouvements stockés encore et encore.



2. Les composants du robot

2.1 Motors pas à pas Nema 17

Un moteur pas à pas est un moteur, à haute durée de vie, qui se commande comme son nom nous l'indique « pas par pas », avec précision. Ces moteurs ont 200 pas par tour soit 1.8° par pas. Ce moteur pas à pas est un moteur à quatre fils : Rouge, Bleu, Vert, Noir. Il y a 4 demi-bobines, câblées en parallèles 2 à 2 pour former finalement 2 bobines : 2 fils par bobines, les fils "rouge" et "bleu" étant les extrémités d'une bobine, les fils "noir" et "vert" les extrémités de l'autre bobine.

Afin d'utiliser un moteur pas à pas il est nécessaire d'utiliser un "driver de moteur pas à pas". Ces drivers permettent de transmettre la puissance électrique au moteur afin de le faire tourner.

Généralement les drivers de moteurs pas à pas permettent de commander les moteurs en fractions de pas complet. En mode demi-pas, le moteur devra alors faire 400 demi-pas pour faire un tour complet soit 0.9° par demi-pas.

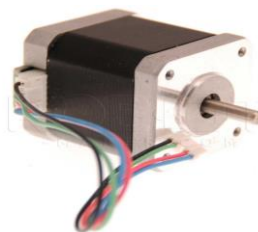


FIGURE 1 : MOTOR PAS A PAS NEMA 17

2.2. Stepper Driver A4988

Le A4988 est un pilote micro pas pour contrôler les moteurs pas à pas bipolaires qui a un traducteur intégré pour une utilisation facile. Cela signifie que nous pouvons contrôler le moteur pas à pas avec seulement 2 broches de notre contrôleur, ou une pour contrôler le sens de rotation et l'autre pour contrôler les étapes.



FIGURE 2 : STEPPER DRIVER A4988

Le pilote propose cinq résolutions de pas différentes : pas complet, demi-Pas, quart de pas, huit pas et seizième pas. En outre, il dispose d'un potentiomètre pour régler la sortie de courant, d'un arrêt thermique en cas de surchauffe et d'une protection contre les courants croisés.

Sa tension logique est de 3 à 5,5 V et le courant maximum par phase est de 2A si un bon refroidissement d'addition est prévu ou 1A de courant continu par phase sans radiateur ni refroidissement.

Minimum Logic Voltage:	3V
Maximum Logic Voltage:	5.5 V
Continuous current per phase:	1 A
Maximum current per phase:	2 A
Minimum Operating Voltage:	8 V
Maximum Operating Voltage:	35 V

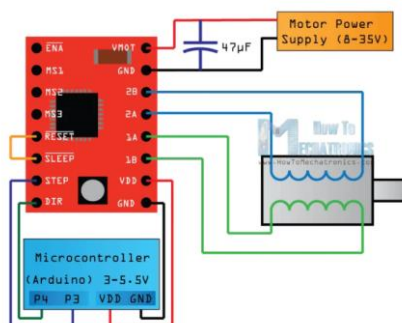


FIGURE 3 : CABLAGE STEPPER ET DRIVER A4988

2.3. PCA 9685

Le module PCA9685 est un contrôleur 16 canaux qui permet de piloter 16 sorties PWM via la communication I2C. Il permet entre autres de libérer des entrées sorties de votre microcontrôleur et piloter jusqu'à 16 LED ou servomoteurs (ou tout autre module prenant en entrée un signal PWM) à l'aide de deux broches (SCL et SDA) tout en conservant les broches de votre microcontrôleur pour d'autres modules comme des capteurs.

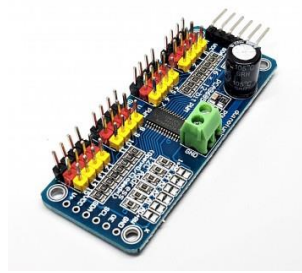
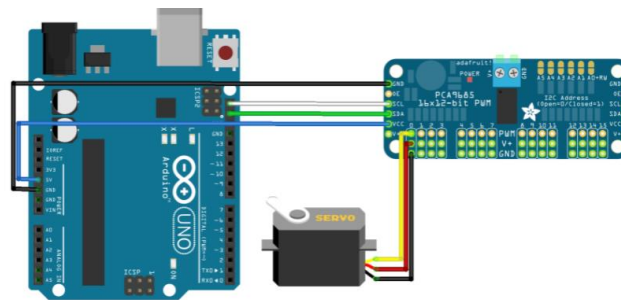


FIGURE 4 : CARTE PCA 9685

Le module est basé sur le contrôleur PCA9685 qui permet de piloter des sorties PWM à l'aide de la communication I2C et d'une horloge intégrée. Ce module comporte 6 ponts permettant de sélectionner l'adresse de la carte et ainsi de placer sur le même bus jusqu'à 62 contrôleurs pour un total de 992 servomoteurs (Adresses disponibles 0x40 à 0x7F).

Il permet de piloter des sorties PWM avec une fréquence ajustable et avec une résolution de 12 bits. Le module est compatible avec les microcontrôleurs 5V et 3.3V.



2.4. Servo Motors (MG996R/ SG90 Micro)

Le MG996R est un servomoteur à engrenages métalliques avec un couple de décrochage maximal de 11 kg/cm. Comme les autres servos RC, le moteur tourne de 0 à 180 degrés en fonction du rapport cyclique de l'onde PWM fournie à sa broche de signal.



FIGURE 5 : SERVO MOTORS

2.5. Bluetooth HC 06

Nous avons communiqué notre carte Arduino par liaison Bluetooth en utilisant un module HC-06 (esclave). Cela nous permet de connecter la carte Arduino à un autre système (Smartphone, ordinateur ou un autre microcontrôleur) afin d'envoyer et de recevoir des données. La communication par Bluetooth permet, de piloter notre robot via une application Smartphone ou récupérer les mesures d'un capteur sur pc.

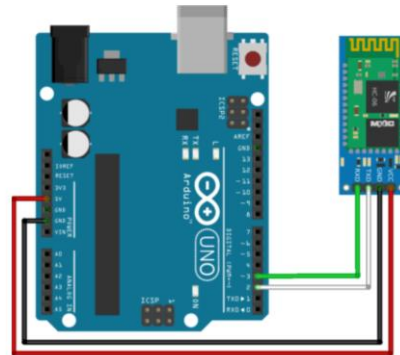


FIGURE 6 : CABLAGE BLUETOOTH HC 06

3. Conception du robot

3.1. Le bras robotique

Le bras robotique à 6 degrés de liberté offre un mouvement rapide, précis et reproductible. Il est réalisé en plastique (impression 3D) et composé de 6 servomoteurs puissants qui permettent d'obtenir un angle de rotation de °180 sur chaque articulation pour piloter le servo mètres nous avons utilisé Le module PCA9685.

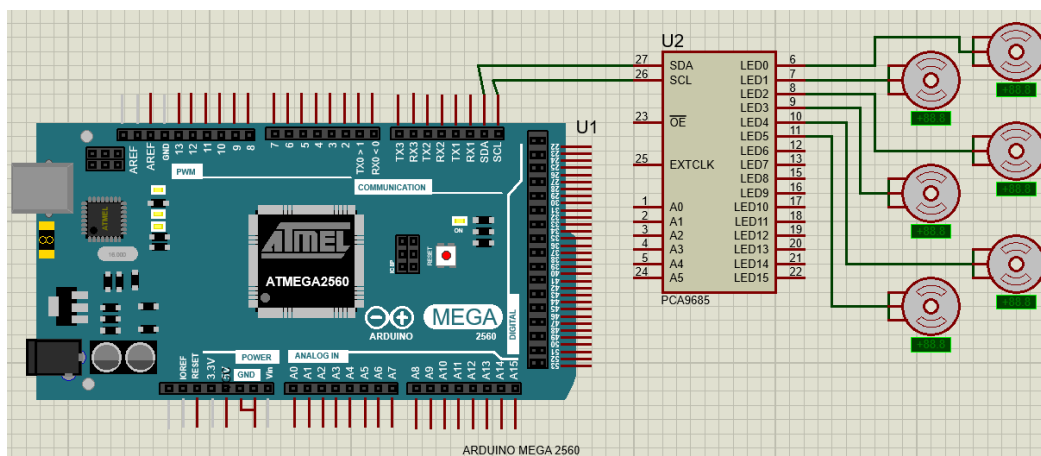


FIGURE 7 : LE SCHEMA ELECTRIQUE DE BRAS (PROTEUS)

La conception structurale à 6 DOF permet au bras robotique de bouger avec souplesse, de sorte qu'il puisse saisir des objets dans n'importe quelle direction.

Le bras robotique 6 axes Arduino est un outil révolutionnaire et qui une fois programmé peut accomplir diverses tâches plus ou moins complexes.

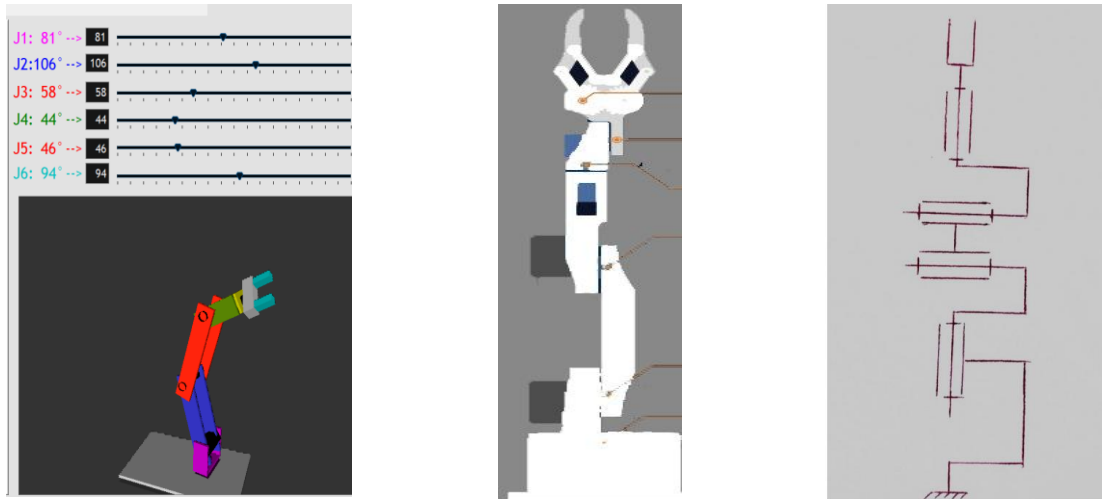


FIGURE 8 : LA CONCEPTION DE BRAS

3.2. Le modèle de chariot

Nous avons utilisé 4 Motors pas à pas Nema 17 pour avoir une meilleure précision en mouvement commandés par les 4 Driver A4988 et finalement nous avons attaché un dissipateur de chaleur, ainsi qu'un petit ventilateur 12V DC pour souffler de l'air, car le dissipateur de chaleur lui-même n'était pas suffisant pour refroidir le régulateur.

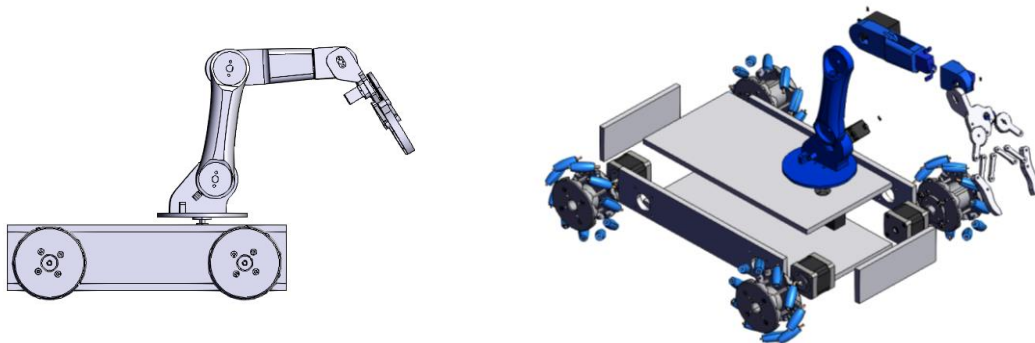


FIGURE 9 : LE CONCEPTION DE CHARIOT EN 3D

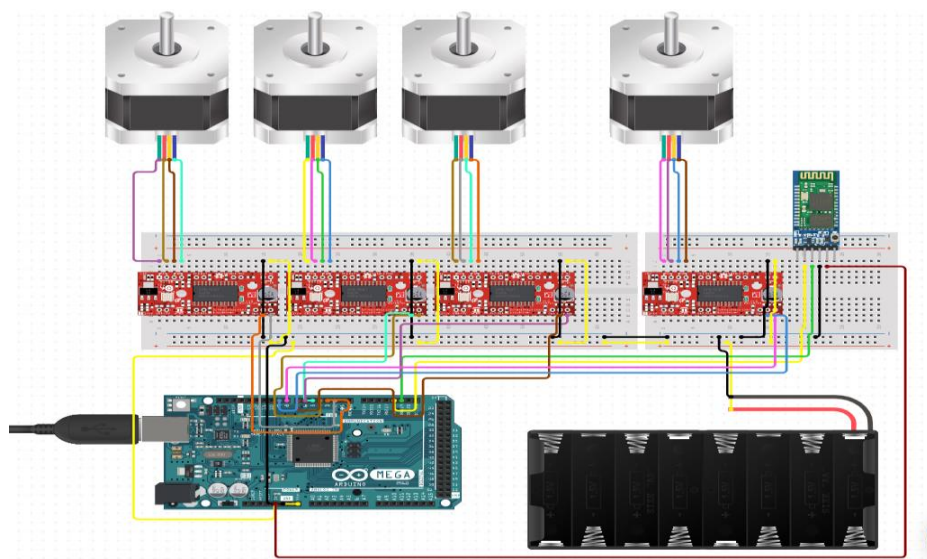


FIGURE 10 : LE SCHEMA ELECTRIQUE DE CHARIOT

3.3. Le schéma électrique

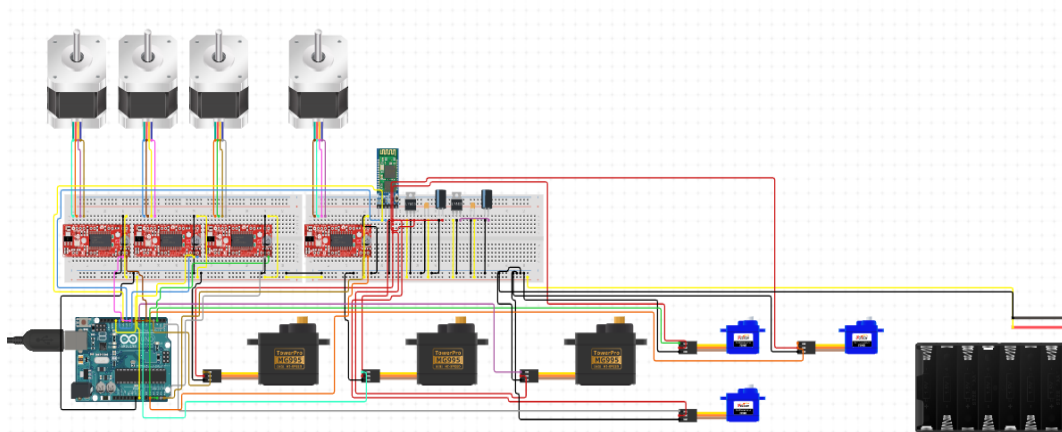


FIGURE 11 : LE SCHEMA ELECTRIQUE

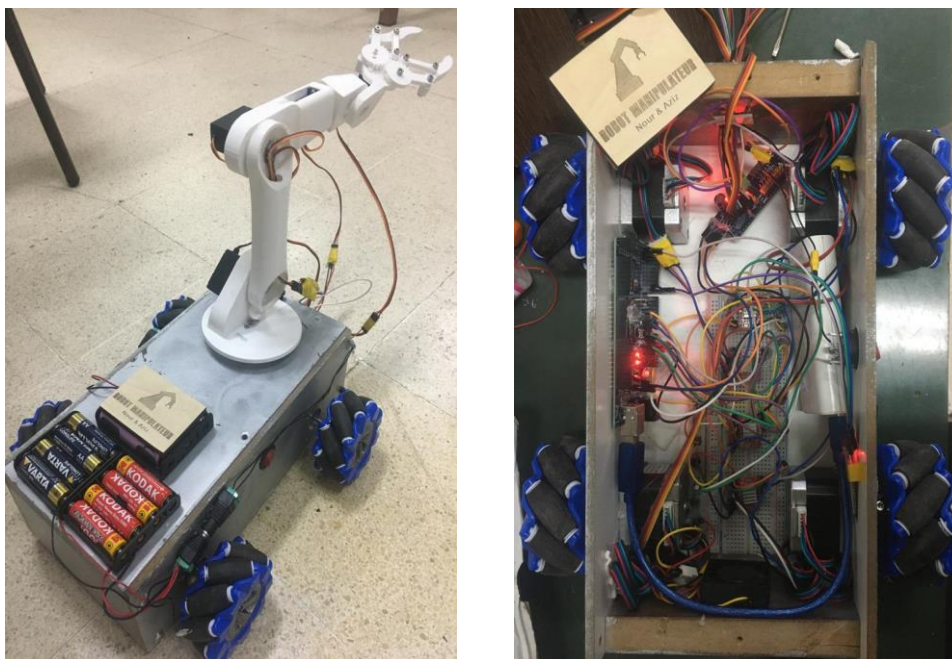


FIGURE 12 : LE CABLAGE ELECTRIQUE

4. Partie software

Reste à voir comment fonctionnent le code Arduino et l'application Android. Comme le code est un peu plus long, pour une meilleure compréhension, je posterai le code source du programme en sections avec une description pour chaque section.

Nous devons donc d'abord définir les 6 servomoteurs, les 4 moteurs pas à pas et la communication Bluetooth, le module PCA ainsi que définir certaines variables nécessaires au programme ci-dessous. Dans la section de configuration, nous définissons la vitesse maximale des steppers, définissons les broches auxquelles les servos sont connectés sur le pca, commençons la communication Bluetooth et mettons le bras du robot en position initiale.

```
#include "HCPA9685.h"
#include <SoftwareSerial.h>
#include <AccelStepper.h>
#define I2CAdd 0x40
HCPA9685 HCPA9685(I2CAdd);
```

```

SoftwareSerial Bluetooth(0, 1);

AccelStepper LeftBackWheel(1, 2, 3); // (Type:driver, STEP, DIR) -
Stepper1
AccelStepper LeftFrontWheel(1, 4, 5); // Stepper2
AccelStepper RightBackWheel(1, 6, 7); // Stepper3
AccelStepper RightFrontWheel(1, 8, 9); // Stepper4

#define led 14

int wheelSpeed = 1500;

int lbw[50], lfw[50], rbw[50], rfw[50]; // arrays for storing
positions/steps

int servo1Pos, servo2Pos, servo3Pos, servo4Pos, servo5Pos, servo6Pos; //
current position
int servo1PPos, servo2PPos, servo3PPos, servo4PPos, servo5PPos,
servo6PPos; // previous position
int servo01SP[50], servo02SP[50], servo03SP[50], servo04SP[50],
servo05SP[50], servo06SP[50]; // for storing positions/steps
int speedDelay = 20;
int index = 0;
int dataIn;
int m = 0;

void setup() {
    HCPCA9685.Init(SERVO_MODE);
    HCPCA9685.Sleep(false);

    // Set initial seed values for the steppers
    LeftFrontWheel.setMaxSpeed(400);
    LeftBackWheel.setMaxSpeed(400);
    RightFrontWheel.setMaxSpeed(400);
    RightBackWheel.setMaxSpeed(400);
    pinMode(led, OUTPUT);

    Bluetooth.begin(9600);
    Serial.begin(9600);
    // Move robot arm to initial position
    servo1PPos = 180;
    Serial.print( servo1PPos);
    HCPCA9685.Servo(0, servo1PPos);
    servo2PPos = 340;
    HCPCA9685.Servo(1, servo2PPos);
    servo3PPos = 0;
    HCPCA9685.Servo(2, servo3PPos);
    servo4PPos = 420;
    HCPCA9685.Servo(3, servo4PPos);
    servo5PPos = 220;
    HCPCA9685.Servo(4, servo5PPos);
    servo6PPos = 250;
    HCPCA9685.Servo(5, servo6PPos);
}

```

L'application Android est créée à l'aide de l'application en ligne MIT App Inventor. Il se compose de boutons simples qui ont des images appropriées en arrière-plan

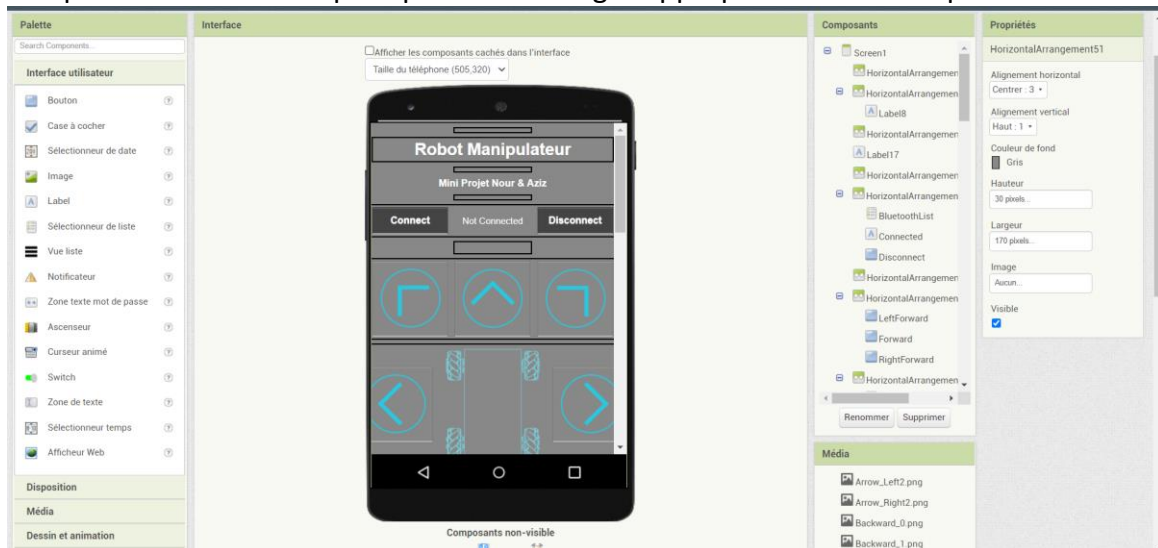


FIGURE 13 : INTERFACE APP INVENTOR

Si nous examinons les blocs de l'application, nous pouvons voir que tout ce qu'elle fait est d'envoyer des nombres à un octet lorsque les boutons sont cliqués.

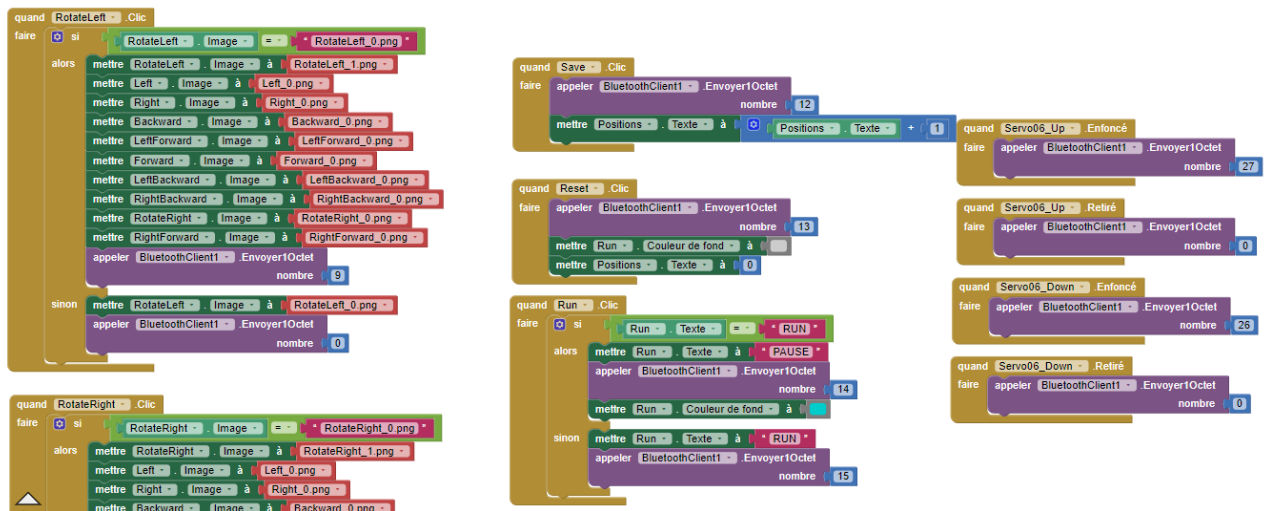


FIGURE 14 : CODE APP INVENTOR

Ainsi, en fonction du bouton cliqué, nous disons à l'Arduino quoi faire. Par exemple, si nous recevons le numéro « 2 », la plate-forme des roues mecanum avancera, en utilisant la fonction personnalisée moveForward.

```
if (dataIn == 2) {
  m = 2;
}
//
if (m == 2) {
  moveForward();
}
```


}

Cette fonction personnalisée règle les quatre moteurs pas à pas pour qu'ils tournent vers l'avant.

```
void moveForward() {
  LeftFrontWheel.setSpeed(wheelSpeed);
  LeftBackWheel.setSpeed(wheelSpeed);
  RightFrontWheel.setSpeed(wheelSpeed);
  RightBackWheel.setSpeed(wheelSpeed);
}
```

Pour se déplacer dans n'importe quelle autre direction, il suffit de faire tourner les roues dans les directions appropriées.

Pour contrôler le bras du robot, nous utilisons la même méthode. Encore une fois, nous avons des boutons dans l'application et lorsque vous maintenez les boutons enfoncés, les articulations du bras du robot se déplacent dans la direction particulière.

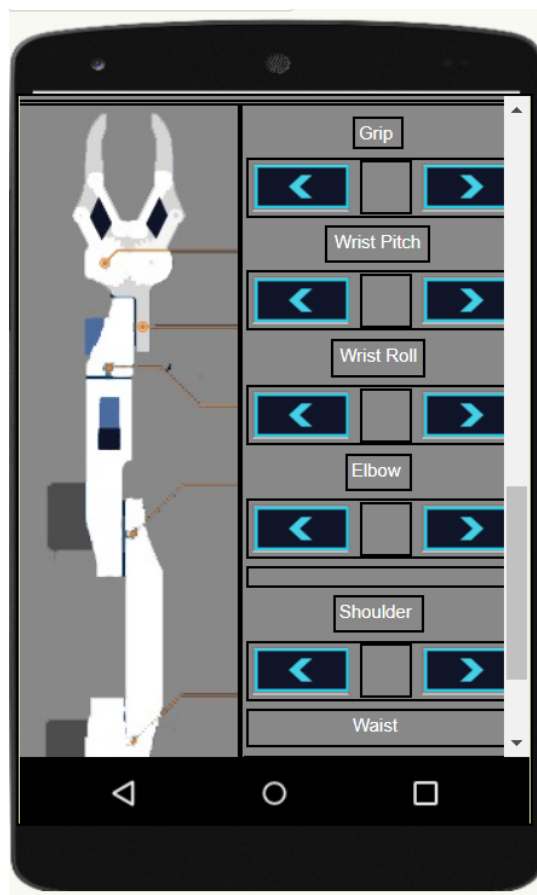
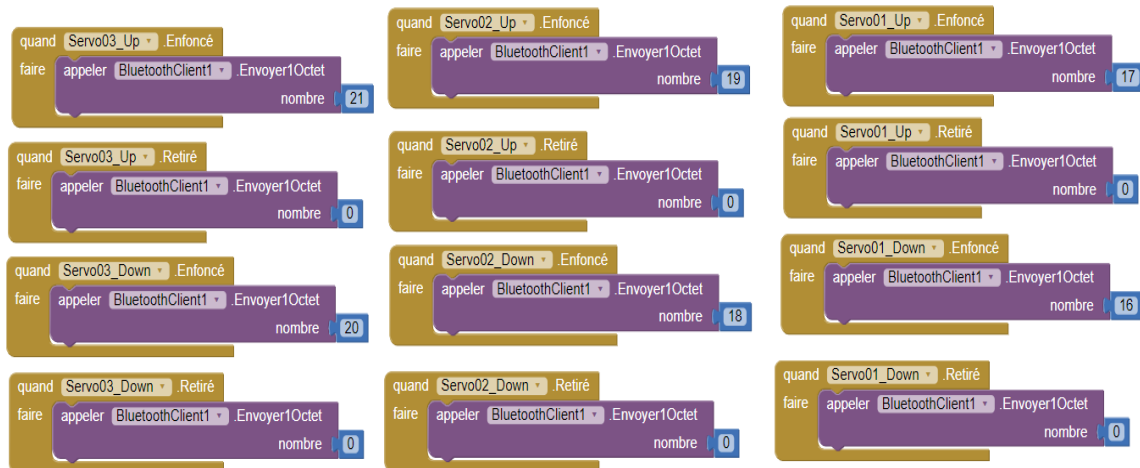


FIGURE 15 : APPLICATION MOBILE (COMMANDE DE BRAS)

De cette façon, nous envoyons simplement un seul numéro à 1 octet lorsqu'un bouton particulier est appuyé.



Le code Arduino entre dans la boucle while de ce numéro et y reste jusqu'à ce que nous touchions le bouton, car à ce moment-là, nous envoyons le numéro 0, ce qui signifie que le robot ne doit rien faire.

```
// Move servo 1 in positive direction
while (m == 16) {
  if (Bluetooth.available() > 0) {
    m = Bluetooth.read();
  }
  servo01.write(servo1PPos);
  servo1PPos++;
  delay(speedDelay);
}
// Move servo 1 in negative direction
while (m == 17) {
  if (Bluetooth.available() > 0) {
    m = Bluetooth.read();
  }
  servo01.write(servo1PPos);
  servo1PPos--;
  delay(speedDelay);
}
```

Ainsi, selon les boutons touchés, les servos se déplacent soit dans le sens positif, soit dans le sens négatif. Le même principe de fonctionnement s'applique à tous les servomoteurs. Pour modifier la vitesse de déplacement, nous utilisons les valeurs provenant du curseur qui vont de 100 à 250.

```
// If arm speed slider is changed
if (dataIn > 101 & dataIn < 250) {
  speedDelay = dataIn / 10; // Change servo speed (delay time)
}
```

En les divisant par 10, nous obtenons des valeurs de 10 à 25, qui sont utilisées comme retard en microsecondes dans les boucles whiles pour piloter les servos.

Pour stocker les mouvements du robot, nous enregistrons simplement les positions actuelles des servos et des moteurs pas à pas dans des tableaux, à chaque fois que le bouton Enregistrer est cliqué.

```
// If button "SAVE" is pressed
if (m == 12) {
//if it's initial save, set the steppers position to 0
if (index == 0) {
LeftBackWheel.setCurrentPosition(0);
LeftFrontWheel.setCurrentPosition(0);
RightBackWheel.setCurrentPosition(0);
RightFrontWheel.setCurrentPosition(0);
}
lbw[index] = LeftBackWheel.currentPosition(); // save position into the array
lfw[index] = LeftFrontWheel.currentPosition();
rbw[index] = RightBackWheel.currentPosition();
rfw[index] = RightFrontWheel.currentPosition();
servo01SP[index] = servo1PPos; // save position into the array
servo02SP[index] = servo2PPos;
servo03SP[index] = servo3PPos;
servo04SP[index] = servo4PPos;
servo05SP[index] = servo5PPos;
servo06SP[index] = servo6PPos;
index++; // Increase the array index
m = 0;
}
```

Ensuite, lorsque nous appuyons sur le bouton Exécuter, nous appelons la fonction personnalisée `runSteps()`. Cette fonction personnalisée parcourt toutes les étapes stockées à l'aide de certaines boucles `for` et `while`.

```
if (m == 14) {
runSteps();
// If button "RESET" is pressed
if (dataIn != 14) {
stopMoving();
memset(lbw, 0, sizeof(lbw)); // Clear the array data to 0
memset(lfw, 0, sizeof(lfw));
memset(rbw, 0, sizeof(rbw));
memset(rfw, 0, sizeof(rfw));
memset(servo01SP, 0, sizeof(servo01SP)); // Clear the array data to 0
memset(servo02SP, 0, sizeof(servo02SP));
memset(servo03SP, 0, sizeof(servo03SP));
memset(servo04SP, 0, sizeof(servo04SP));
memset(servo05SP, 0, sizeof(servo05SP));
memset(servo06SP, 0, sizeof(servo06SP));
index = 0; // Index to 0
}
}
```

Nous devrions noter qu'il commence à partir de la première position et va jusqu'à la dernière position, et le répète encore et encore. Par conséquent, lors de l'enregistrement des étapes, nous devons en fait positionner le robot de manière que la première étape ait la même position que la dernière étape. Tout en parcourant les étapes, nous pouvons également modifier la vitesse de la plate-forme et du bras du robot, ainsi que mettre en pause et réinitialiser toutes les étapes.

5. Conclusion

Le projet fonctionne bien, mais il est loin d'être parfait. Les mouvements automatiques peuvent ne pas être aussi précis en raison du patinage des roues mécaniques ainsi que des mauvaises performances des servomoteurs. Ces servomoteurs bon marché peuvent également trembler même lorsqu'ils ne bougent pas simplement parce qu'ils n'ont pas assez de force pour supporter le poids des pièces imprimées en 3D.

Néanmoins, notre projet peut être toujours évolué on peut ajouter une caméra ou même utiliser l'intelligence artificielle pour identifier les objets directement sans l'intervention humaine.