



# ROBOT MANIPULATEUR

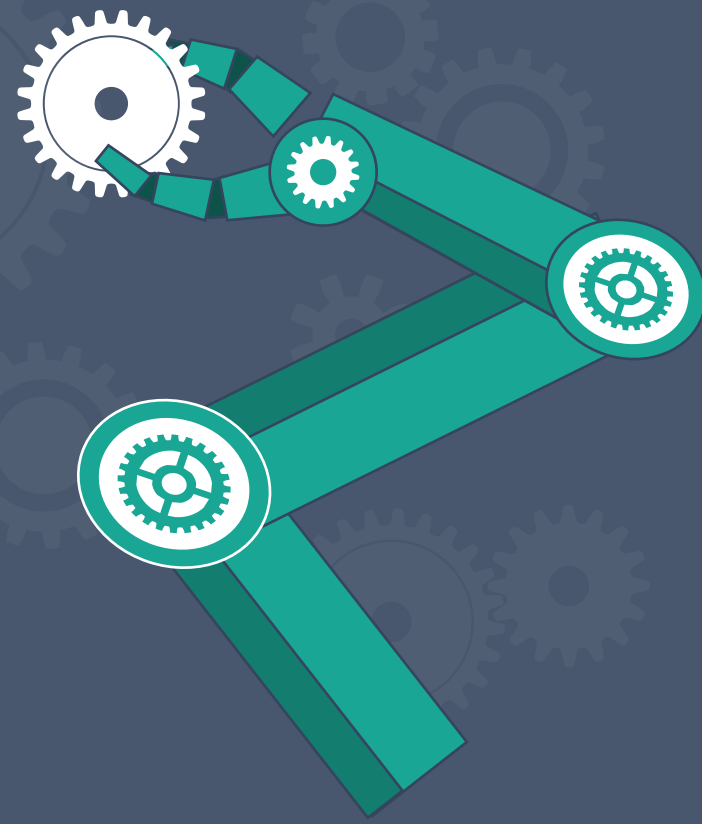
Elaboré par:

Nour Trabelsi  
Ahmed Aziz Gaddour

Classe :

L3AII

2021/2022



A stylized illustration of a teal robotic arm with three joints, each marked with a gear icon. The arm is positioned diagonally from the top left towards the bottom right. In the background, there are several dark grey gears of various sizes. The overall theme is industrial and mechanical.

# 4. INDUSTRY

## Plan

01

présentation

02

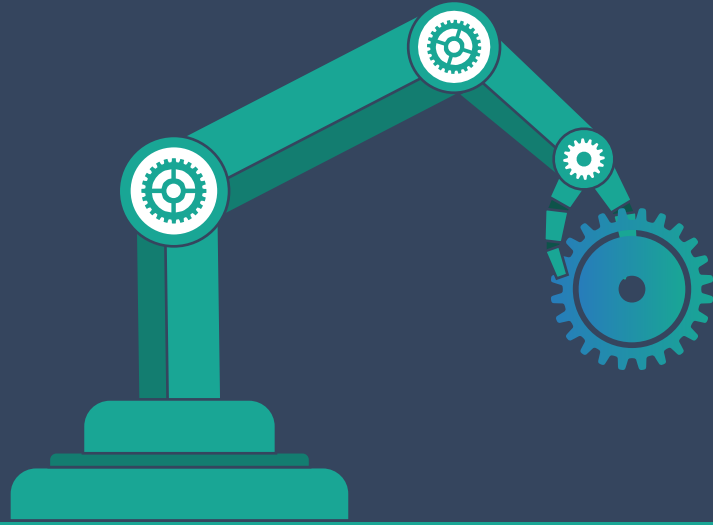
Etude pratique

03

Partie software

04

Conclusion



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

# ROBOT MANIPULATEUR



Notre robot manipulateur est composé d'un chariot mobile et un bras robotique imprimé en 3D piloter avec une 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.**

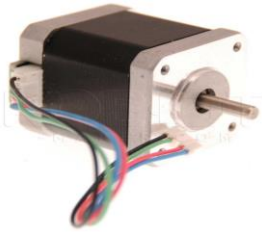


# ETUDE PRATIQUE



# Les composants du robot

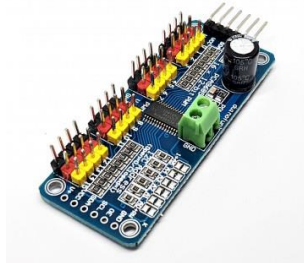
Motors pas à pas Nema 17



Stepper Driver A4988



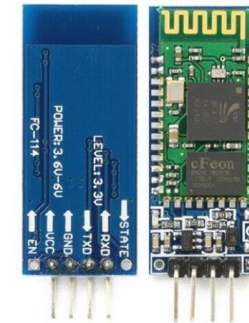
PCA 9685



Servo Motors (MG996R/ SG90 Micro)

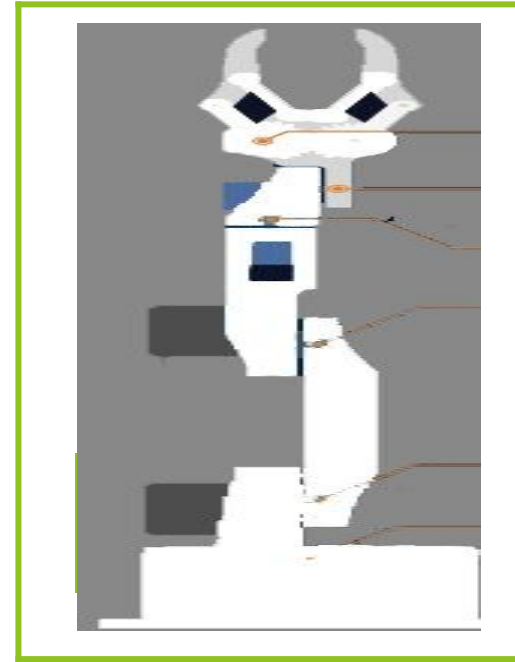
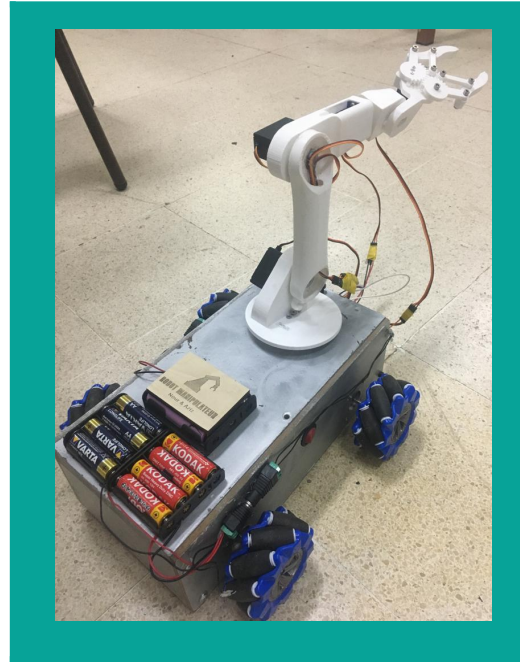
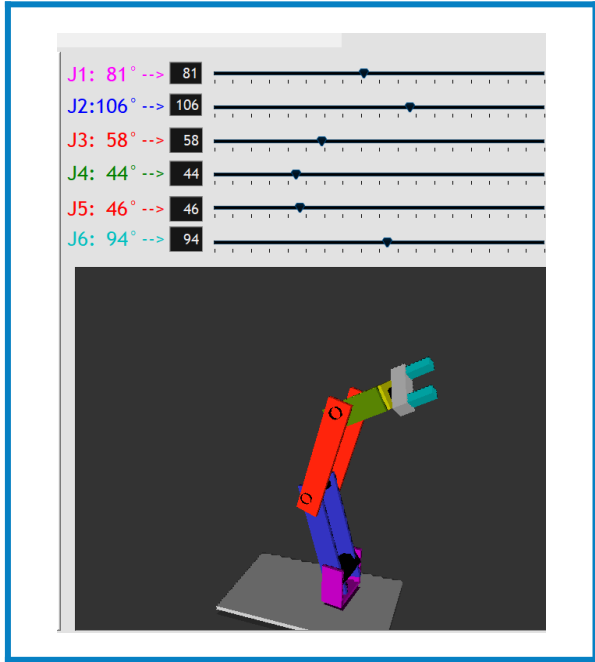


Bluetooth HC 06



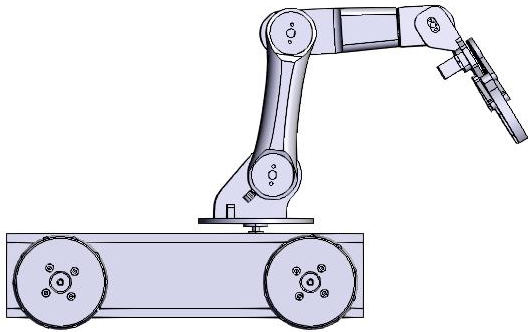
Arduino Mega





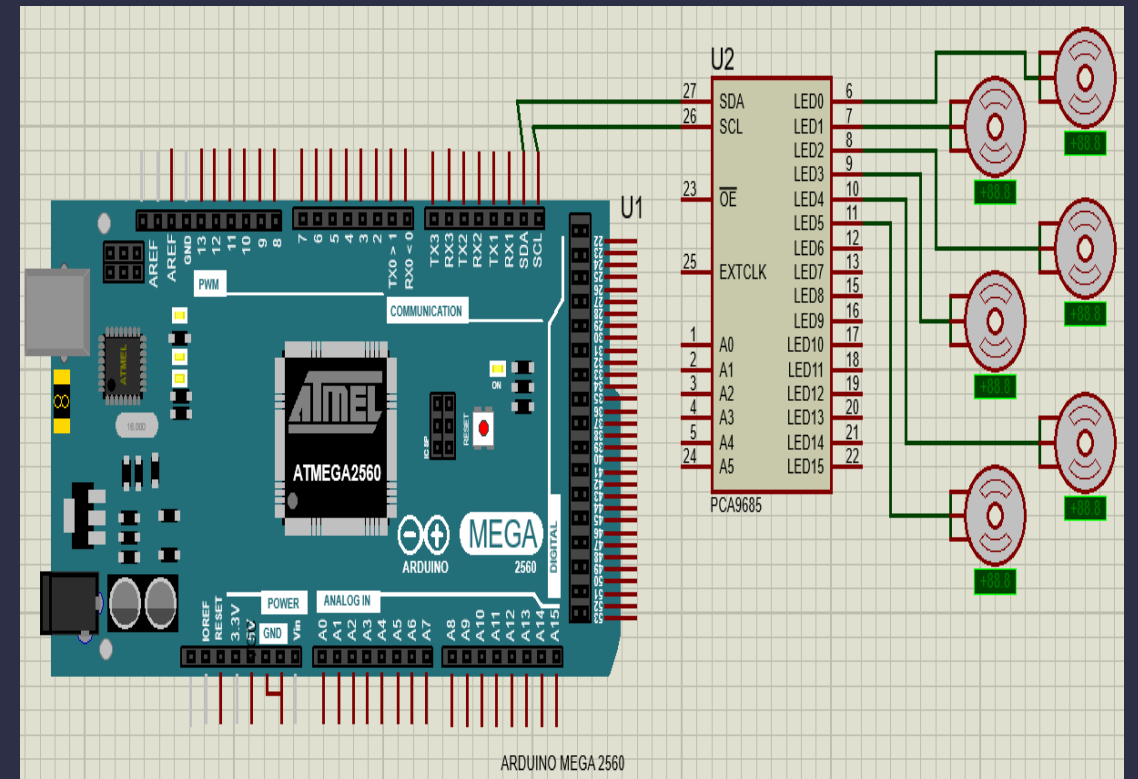
Notre bras robotique à 6 degrés de liberté offre un mouvement rapide, précis et reproductible.

**BRAS  
ELECTRIQUE**

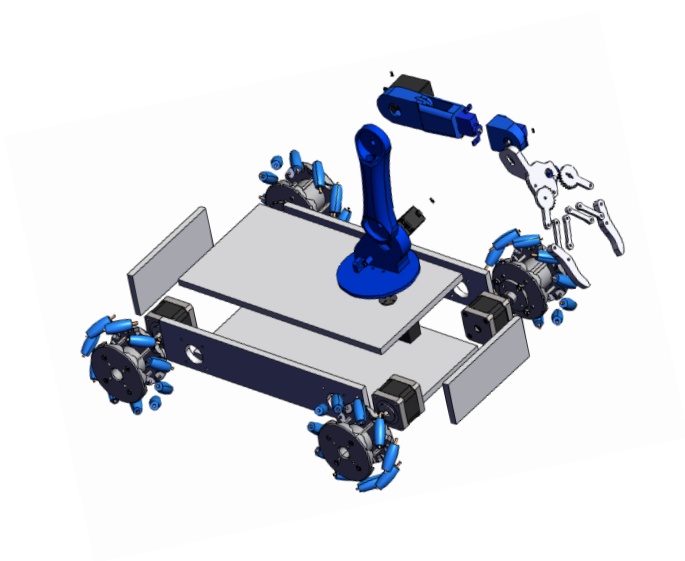


Il est réalisé en plastique (impression 3D) et composé de 6 servomoteurs puissant qui permettent d'obtenir un angle de rotation de °180 sur chaque articulation .

Pour piloter les servomoteurs nous avons utilisé Le module PCA9685.

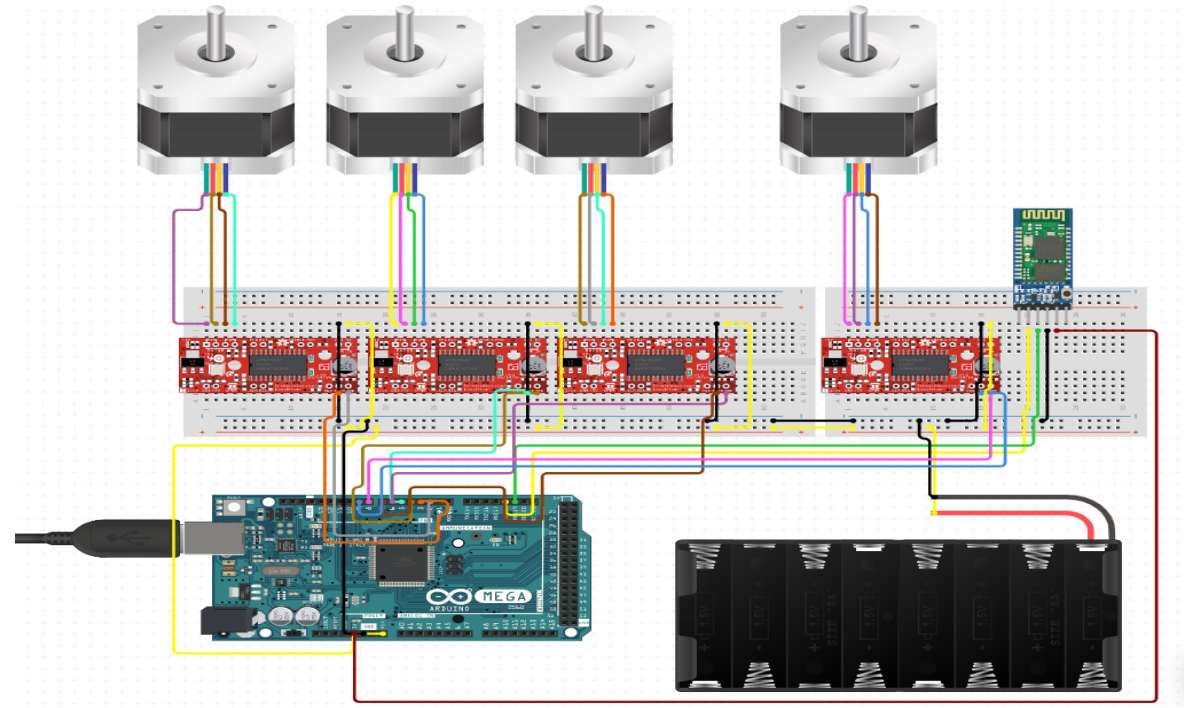
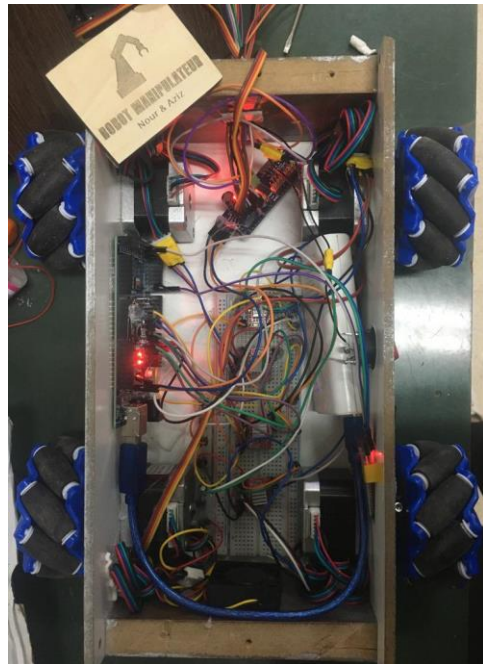






# CHARIOT

pour avoir une meilleure précision en mouvement nous avons utilisé 4 Motors pas à pas Nema 17 commandés par les 4 Driver A4988 et finalement nous avons attaché un dissipateur de chaleur, ainsi qu'un petit ventilateur 12V DC pour refroidir le régulateur





**PARTIE SOFTWARE**





## Code de Projet

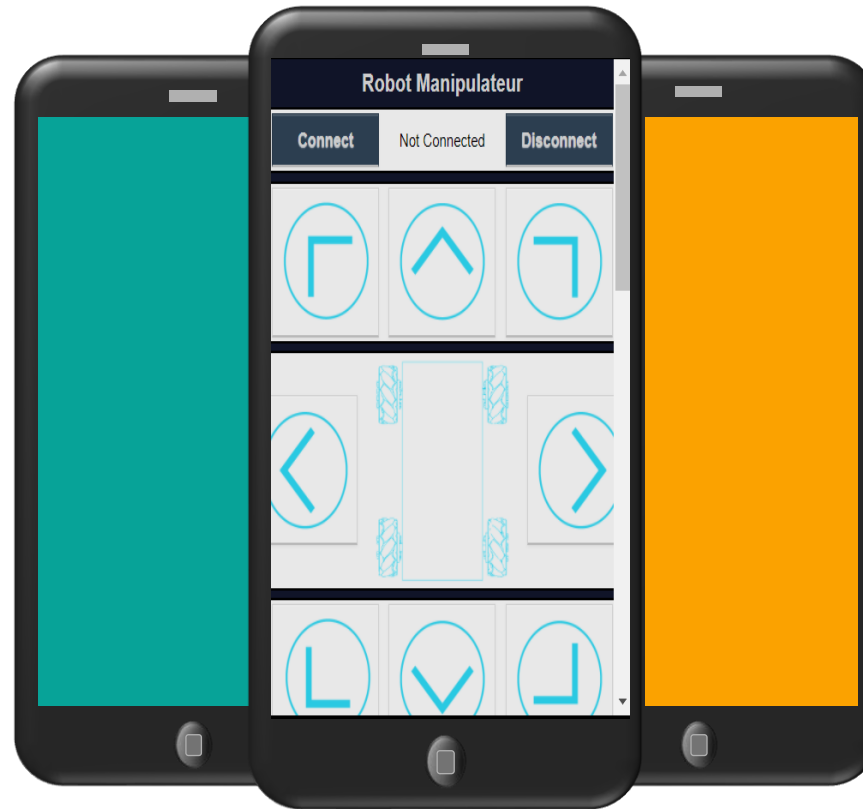
```
#include "HCPA9685.h"
#include <SoftwareSerial.h>
#include <AccelStepper.h>
#define I2CAdd 0x40
HCPA9685 HCPA9685(I2CAdd);
SoftwareSerial Bluetooth(A8, 38);

AccelStepper LeftBackWheel(1, 2, 3);
AccelStepper LeftFrontWheel(1, 4, 5);
AccelStepper RightBackWheel(1, 6, 7);
AccelStepper RightFrontWheel(1, 8, 9);
```

```
// 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);
    HCPA9685.Servo(0,    servo1PPos);
    servo2PPos = 340;
    HCPA9685.Servo(1,    servo2PPos);
    servo3PPos = 0;
    HCPA9685.Servo(2,    servo3PPos);
    servo4PPos = 420;
    HCPA9685.Servo(3,    servo4PPos);
    servo5PPos = 220;
    HCPA9685.Servo(4,    servo5PPos);
    servo6PPos = 250;
    HCPA9685.Servo(5,    servo6PPos);
```

# Application Mobile

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



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

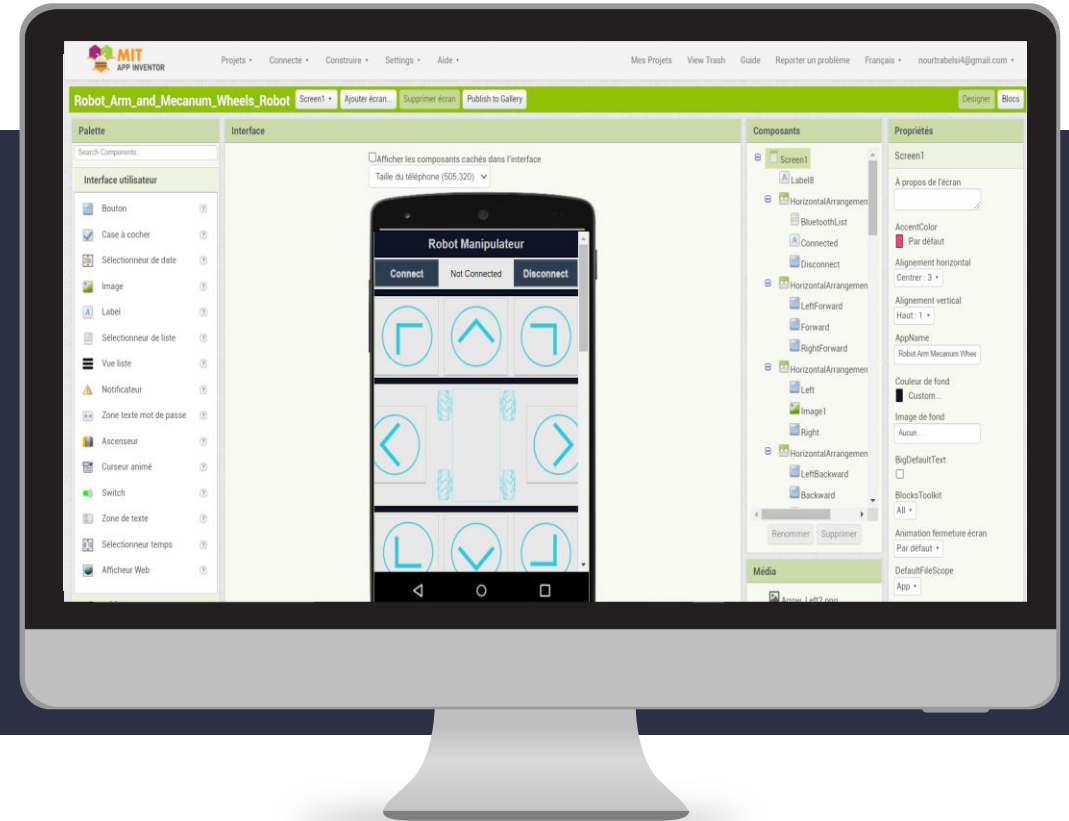


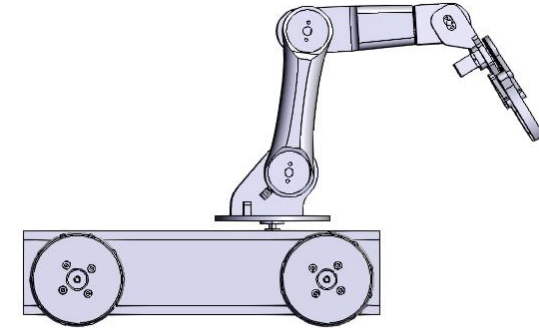
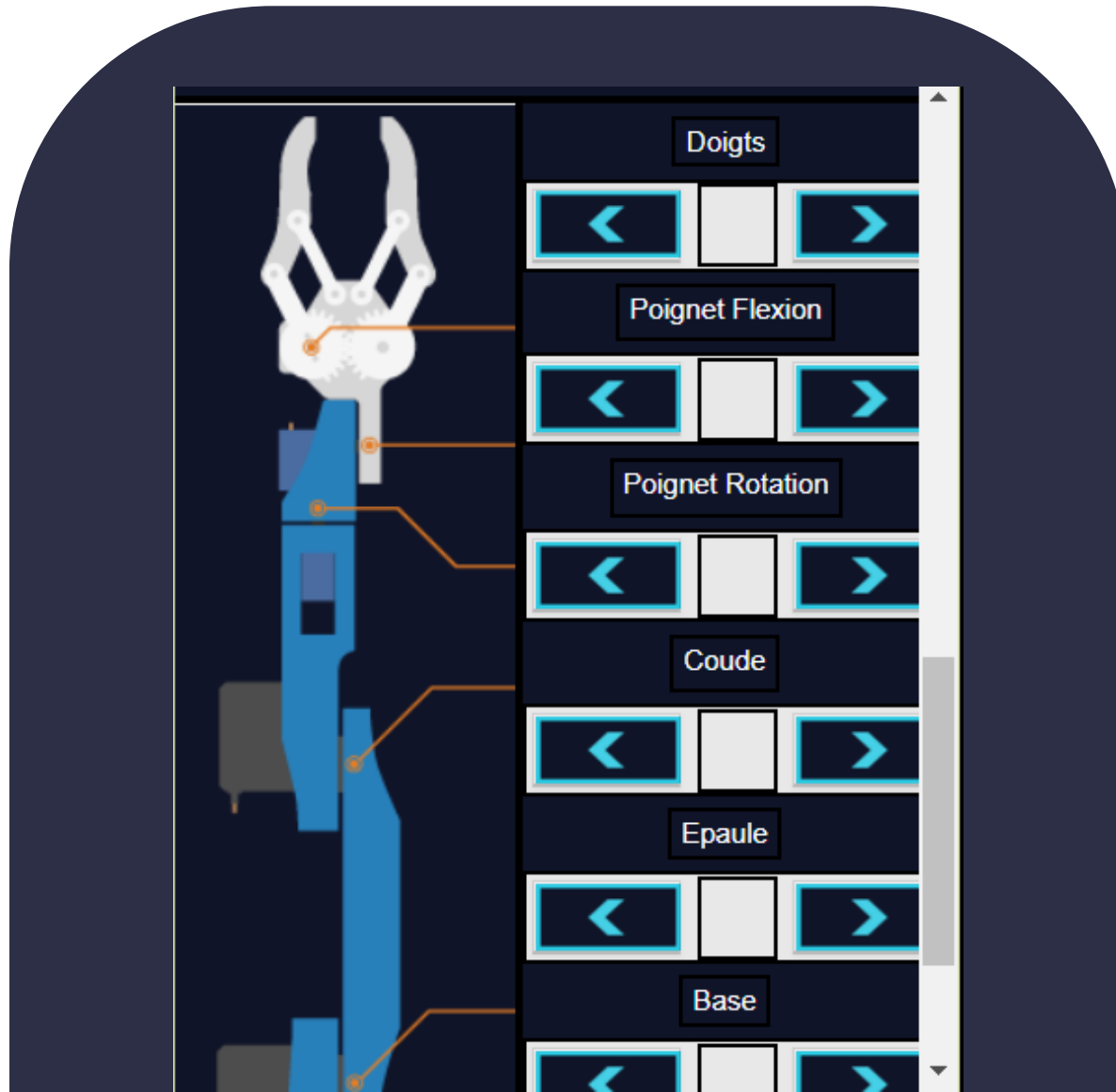
# Commande de robot



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

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

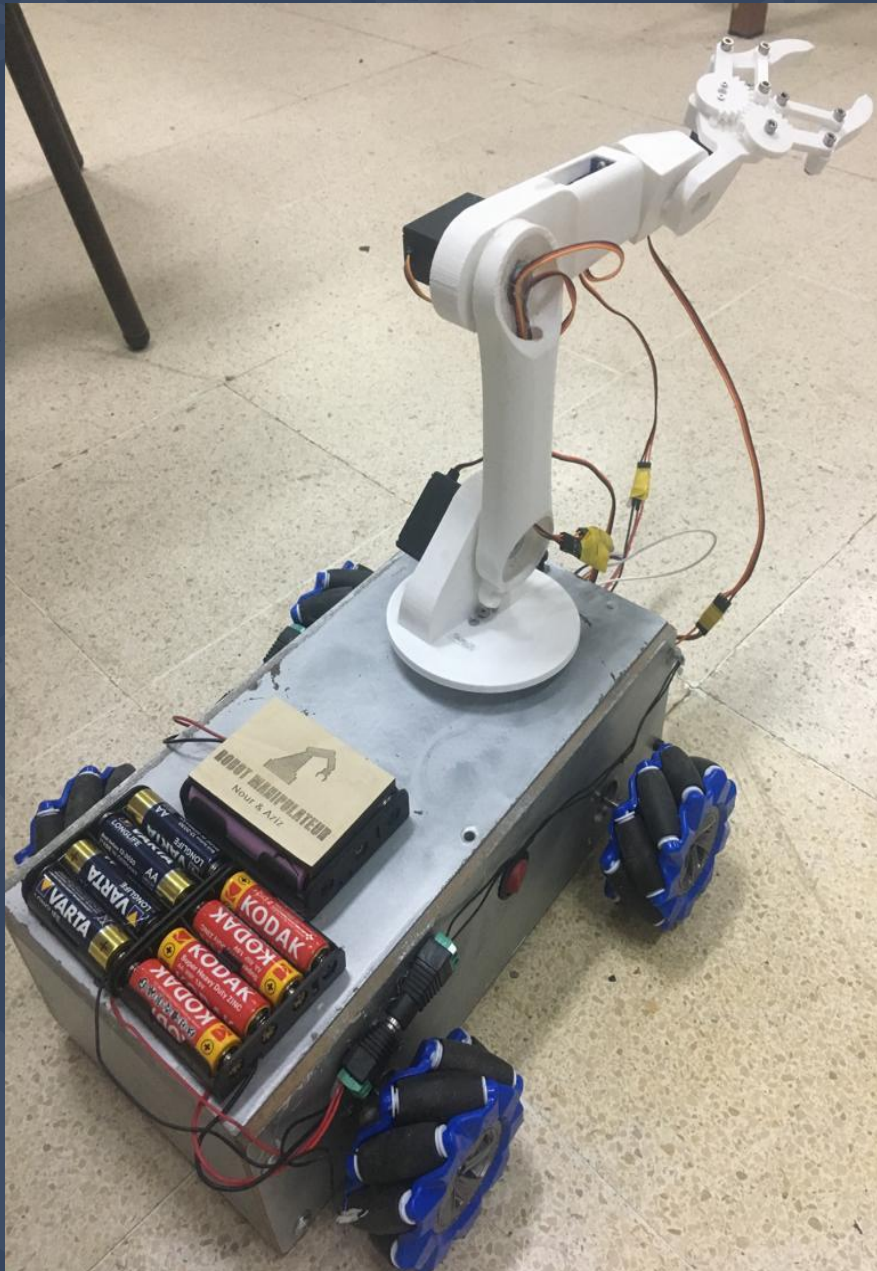




Pour contrôler le bras du robot, nous utilisons la même méthode.

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.





## 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 peuvent trembler 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.



**MERCI POUR VOTRE ATTENTION**

