

## Le bras robotisé

Pour le bras, nous avons décidés de réduire le nombre d'axe pour faciliter la manipulation de celui-ci si nous devons le faire bouger à l'aide du casque. Par ailleurs nous aimerions que le bras possède une puissance raisonnable et une vitesse de rotation de ses axes rapides mais au maximum précise.

Le bras sera le plus léger possible et aura une base rotative à 360° avec deux autre axe (coude poignet) avec une rotation max de 90°.

En attendant de pouvoir utiliser les commandes envoyez par le casque nous avons décidé d'utiliser une de guider le bras robotique a l'aide d'un smartphone pour simuler un écran tactile dans une entreprise. Pour sa nous avons utilisé un module Bluetooth HC-06

Pour les mouvements nous allons utiliser 5 moteurs HS-485-HB chacun alimenté en 5 V, nous avons ensuit tout relie sur un Arduino UNO



Motor HS-485-HB



Module Bluetooth



Arduino UNO



Photo du bras Monté



Photo du bras monté + support en bois  
découpée au FaLab

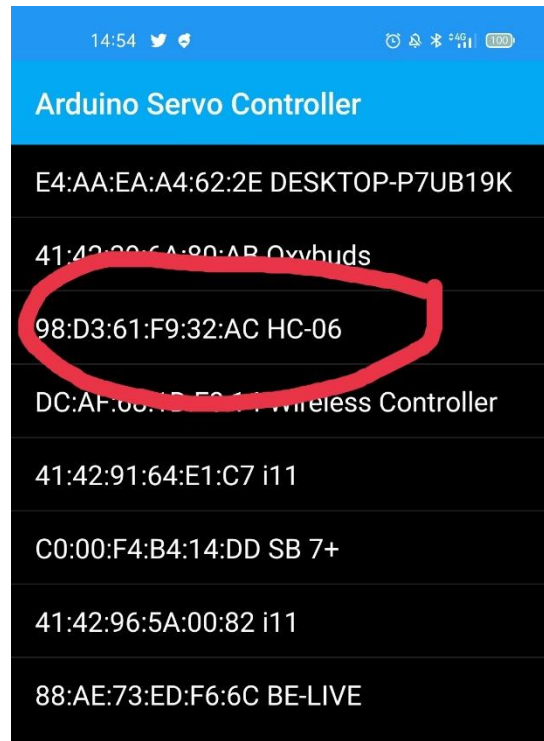
Pour contrôler les bras nous allons utiliser l'application sur smartphones Arduino Servo Controller :

<https://play.google.com/store/apps/details?id=com.arunpaul.arduinotronicscontroller&hl=fr&gl=US>

En suite mettre se connecter sur le module Arduino qui devra être allumé et connecté au Bluetooth du smartphone



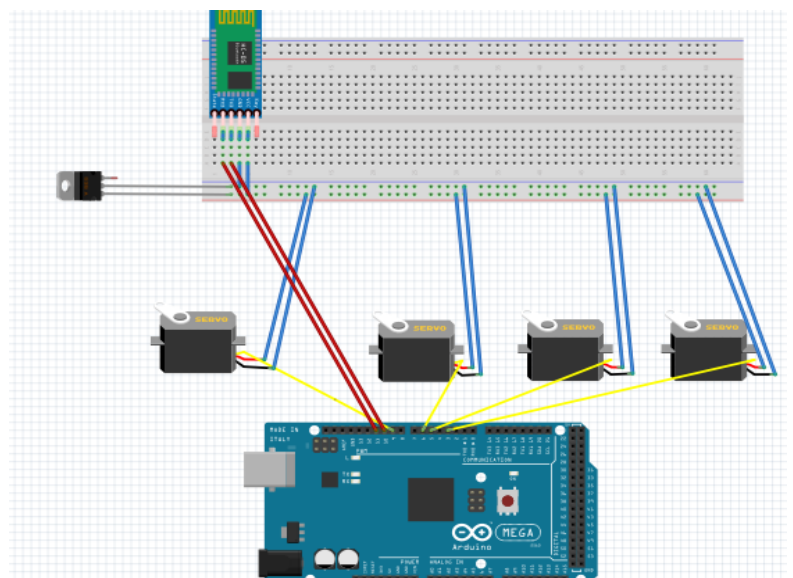
Interface de l'application



Connection au Module HC-06

Ici nous allons contrôler 4 moteurs.

Le sens de rotation va varier entre 0 et 180°. Les moteurs seront reliés aux Pin de l'Arduino. Le tout sera alimenté par un générateur de 5 V.



Montage réalisé sur le logiciel FRITZING

```
// PROJET BRAS ROBOTIQUE ADAMOU FATIHOU

#include <SoftwareSerial.h> // Bibliothèque logicielle TX RX pour Bluetooth

#include <Servo.h> // servo library
Servo myservo1, myservo2, myservo3, myservo4; // Le nom des moteur

int bluetoothTx = 10; // bluetooth tx connecté au 10 pin
int bluetoothRx = 11; // bluetooth rx connecté au 11 pin

SoftwareSerial bluetooth(bluetoothTx, bluetoothRx);

void setup()
{
  myservo1.attach(3); // attachez le fil du signal du servo à la broche 9
  myservo2.attach(5);
  myservo3.attach(6);
  myservo4.attach(9);
  //Setup connexion série usb à l'ordinateur
  Serial.begin(9600);

  //Configuration de la connexion série Bluetooth pour android
  bluetooth.begin(9600);
}

void loop()
{
  //Lecture depuis Bluetooth et écriture sur USB série
  if(bluetooth.available() >= 2 )
  {
    unsigned int servopos = bluetooth.read();
    unsigned int servopos1 = bluetooth.read();
    unsigned int realservo = (servopos1 * 256) + servopos;

    Serial.println(realservo);

    if (realservo >= 1000 && realservo < 1180) {
      int servol = realservo;
      servol = map(servol, 1000, 1180, 0, 180); // Le sense de rotation
      myservo1.write(servol);
      Serial.println("Servo 1 ON"); // Affiche dans le moniteur en serie l'etat du moteur
      delay(10);
    }
    if (realservo >= 2000 && realservo < 2180) {
      int servo2 = realservo;
      servo2 = map(servo2, 2000, 2180, 0, 180);
      myservo2.write(servo2);
      Serial.println("Servo 2 ON");
      delay(10);
    }
    if (realservo >= 3000 && realservo < 3180) {
      int servo3 = realservo;
      servo3 = map(servo3, 3000, 3180, 0, 180);
      myservo3.write(servo3);
      Serial.println("Servo 3 ON");
      delay(10);
    }
    if (realservo >= 4000 && realservo < 4180) {
      int servo4 = realservo;
      servo4 = map(servo4, 4000, 4180, 0, 180);
      myservo4.write(servo4);
      Serial.println("Servo 4 ON");
      delay(10);
    }
  }
}
```

Compilation terminée.

Ici vous pouvez voir le codes Arduino commenté pour commander le 4 moteurs. Pour pouvoir contrôler plus de moteur il faut ajouter le nom du moteur, lui attribuer un Pin ensuite recréer dans le **Void Loop** un « if » avec la configuration du moteur .