

```

#include <ArduinoBLE.h>
#include<Servo.h>

// =====
// Motor Shield R3
// =====

const int dirA = 12;
const int pwmA = 3;
const int brakeA = 9;

const int dirB = 13;
const int pwmB = 11;
const int brakeB = 8;

int vitesse = 180;
Servo servo1;
Servo servo2;

const int pinServo1 = 5;
const int pinServo2 = 6;

// Positions servo1
const int S1_POS1 = 20;
const int S1_POS2 = 120;

// Positions servo2
const int S2_POS1 = 20;
const int S2_POS2 = 120;

// =====
// BLE UART SERVICE
// =====

// Service UART Nordic
BLEService uartService("6E400001-B5A3-F393-E0A9-E50E24DCCA9E");

// RX = reçoit les données du smartphone
BLECharacteristic rxCharacteristic(
  "6E400002-B5A3-F393-E0A9-E50E24DCCA9E",
  BLEWrite | BLEWriteWithoutResponse,
  20
);

// TX = envoie des données au smartphone
BLECharacteristic txCharacteristic(
  "6E400003-B5A3-F393-E0A9-E50E24DCCA9E",
  BLENotify,

```

20

);

unsigned long lastCommandTime = 0;

const unsigned long timeout = 1000;

void setup() {

Serial.begin(115200);

// =====

// Configuration moteurs

// =====

pinMode(dirA, OUTPUT);

pinMode(pwmA, OUTPUT);

pinMode(brakeA, OUTPUT);

pinMode(dirB, OUTPUT);

pinMode(pwmB, OUTPUT);

pinMode(brakeB, OUTPUT);

stopMotors();

// =====

// BLE

// =====

if (!BLE.begin()) {

Serial.println("Erreur BLE");

while (1);

}

// =====

// Nom du robot

// =====

BLE.setLocalName("BattleBot_09");

BLE.setAdvertisedService(uartService);

uartService.addCharacteristic(rxCharacteristic);

uartService.addCharacteristic(txCharacteristic);

BLE.addService(uartService);

```

servo1.attach(pinServo1);
servo2.attach(pinServo2);

//position initiale
servo1.write(S1_POS1);
servo2.write(S2_POS1);


BLE.advertise();

Serial.println("BattleBot prêt");
}

void loop() {

  BLEDevice central = BLE.central();

  if (central) {

    Serial.println("Smartphone connecté");

    while (central.connected()) {

      if (rxCharacteristic.written()) {

        String message = "";

        int length = rxCharacteristic.valueLength();

        const uint8_t* data = rxCharacteristic.value();

        for (int i = 0; i < length; i++) {

          message += (char)data[i];
        }

        message.trim();

        Serial.print("Commande : ");
        Serial.println(message);

        lastCommandTime = millis();

        if (message == "F") {

          avancer();
        }
      }
    }
  }
}

```

```

    else if (message == "B") {

        reculer();
    }

    else if (message == "L") {

        gauche();
    }

    else if (message == "R") {

        droite();
    }

else if (message == "A") {

    servo1.write(S1_POS1);
}

else if (message == "Z") {

    servo1.write(S1_POS2);
}

else if (message == "E") {

    servo2.write(S2_POS1);
}
else if (message == "T") {

    servo2.write(S2_POS2);
}

    else if (message == "S") {

        stopMotors();
    }
}

// =====
// Sécurité timeout
// =====

if (millis() - lastCommandTime > timeout) {

    stopMotors();
}

```

```

    }

    stopMotors();

    Serial.println("Déconnecté");
}
}

// =====
// Fonctions moteurs
// =====

void avancer() {

    digitalWrite(brakeA, LOW);
    digitalWrite(brakeB, LOW);

    digitalWrite(dirA, HIGH);
    digitalWrite(dirB, LOW);

    analogWrite(pwmA, vitesse);
    analogWrite(pwmB, vitesse);
}

void reculer() {

    digitalWrite(brakeA, LOW);
    digitalWrite(brakeB, LOW);

    digitalWrite(dirA, LOW);
    digitalWrite(dirB, HIGH);

    analogWrite(pwmA, vitesse);
    analogWrite(pwmB, vitesse);
}

void gauche() {

    digitalWrite(brakeA, LOW);
    digitalWrite(brakeB, LOW);

    digitalWrite(dirA, LOW);
    digitalWrite(dirB, LOW);

    analogWrite(pwmA, vitesse);
    analogWrite(pwmB, vitesse);
}

```

```
void droite() {  
  
    digitalWrite(brakeA, LOW);  
    digitalWrite(brakeB, LOW);  
  
    digitalWrite(dirA, HIGH);  
    digitalWrite(dirB, HIGH);  
  
    analogWrite(pwmA, vitesse);  
    analogWrite(pwmB, vitesse);  
}
```

```
void stopMotors() {  
  
    digitalWrite(brakeA, HIGH);  
    digitalWrite(brakeB, HIGH);  
  
    analogWrite(pwmA, 0);  
    analogWrite(pwmB, 0);  
}
```