

Rapport Séance du 11/01 YANG Benjamin

- Premier fonctionnement du moteur à hélices avec alimentation sur prise. (avec mon binôme)
- Recherche et commande du moteur à hélices car emprunt de celui d'Alaeddine Rejeb.
- Liaison entre l'application bluetooth electronics et le code principal : autrement dit, utiliser des modules comme des jauges et vérifier la communication entre les deux appareils.
- Écriture du code, permettant de contrôler les moteurs à l'aide du téléphone : 1^{ère} étape à l'aide d'un slider qui permet de contrôler simplement la puissance des moteurs
- 2^{ème} étape : Code permettant d'utiliser un joystick : Nous avons décidé d'utiliser la forme Rayon/Angle, qui à l'aide de la trigonométrie nous permettra de gérer nos moteurs. Notre code nous donne donc une variable PWMA qui nous donne l'angle du joystick ainsi qu'une variable PWMR qui nous indique la « longueur » du joystick (càd si on le pousse au maximum). Voici le code, que j'ai codé sur la base du programme de mon binôme :

```
// Bluetooth
#include<SoftwareSerial.h>
#define RX 10
#define TX 11
SoftwareSerial BlueT(RX,TX);
char Data; //Pour stocker la lettre envoyée par le smartphone
int PWMR;
float PWMA;

//-- MOTEUR A --
int ENA=9; //Connecté à Arduino pin 9(sortie PWM)
int IN1=4; //Connecté à Arduino pin 4
int IN2=5; //Connecté à Arduino pin 5
//-- MOTEUR B --
int ENB=3; //Connecté à Arduino pin 3(Sortie PWM)
int IN3=6; //Connecté à Arduino pin 6
int IN4=7; //Connecté à Arduino pin 7

void setup() {
  pinMode(ENA,OUTPUT); // Configurer
  pinMode(ENB,OUTPUT); // les broches
  pinMode(IN1,OUTPUT); // comme sortie
  pinMode(IN2,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);

  //digitalWrite(ENA,LOW); // Moteur A - Ne pas tourner
  //digitalWrite(ENB,LOW); // Moteur B - Ne pas tourner

  // Direction du Moteur A
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);

  // Direction du Moteur B
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);

  Serial.begin(9600);
  delay(500);
  Serial.println("Bonjour -Pret pour les commandes AT");
  BlueT.begin(9600);
  delay(500);
}

void loop() {
  if (BlueT.available()) { //Si il y a un message dans le buffer Rx
    Data = BlueT.read();
    if (Data=='J') {
      PWMR = BlueT.parseInt();
      PWMA = BlueT.parseInt(); //renvoie le premier entier trouvé
      Serial.println(PWMR);
      Serial.print("--");
      Serial.print(PWMA);

      if (PWMR <= 90 || PWMR >= 270) {
        // Direction du Moteur A
        digitalWrite(IN1,LOW);
        digitalWrite(IN2,HIGH);

        // Direction du Moteur B
        digitalWrite(IN3,HIGH);
        digitalWrite(IN4,LOW);

        if (PWMR <= 90) {
          digitalWrite(ENA, HIGH);
          analogWrite(ENB, cos(PWMA)*PWMR);
        }
        else {
          digitalWrite(ENA, cos(PWMA)*PWMR);
        }
      }
      else {
        // Direction du Moteur A
        digitalWrite(IN1,HIGH);
        digitalWrite(IN2,LOW);

        // Direction du Moteur B
        digitalWrite(IN3,LOW);
        digitalWrite(IN4,HIGH);

        if (PWMR >= 270) {
          digitalWrite(ENB, HIGH);
          analogWrite(ENA, cos(PWMA)*PWMR);
        }
        else {
          digitalWrite(ENB, cos(PWMA)*PWMR);
        }
      }
    }
  }
}
```