

// Analyse des paramètres du véhicule

```
const int IN1 = 12; // direction A
const int IN2 = 9;  // brake A
const int PWM_A = 3;

const int IN3 = 13; // direction B
const int IN4 = 8;  // brake B
const int PWM_B = 11;

const int BOUTON_START = 7;
const int CAPTEUR_LIGNE = 6;

unsigned long Temps_course = 11157;
unsigned long debutCourse = 0;

int vitesse = 180;

void setup() {

    Serial.begin(9600);

    pinMode(BOUTON_START, INPUT_PULLUP);
    pinMode(CAPTEUR_LIGNE, INPUT_PULLUP);

    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(PWM_A, OUTPUT);

    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
    pinMode(PWM_B, OUTPUT);

    // moteurs arrêtés
    digitalWrite(IN2, HIGH);
    digitalWrite(IN4, HIGH);
}

void avancer() {

    digitalWrite(IN2, LOW); // release brake A
    digitalWrite(IN4, LOW); // release brake B
```

```

    digitalWrite(IN1, HIGH);
    digitalWrite(IN3, HIGH);

    analogWrite(PWM_A, vitesse);
    analogWrite(PWM_B, vitesse);
}

void stopMoteurs() {

    analogWrite(PWM_A, 0);
    analogWrite(PWM_B, 0);

    digitalWrite(IN2, HIGH);
    digitalWrite(IN4, HIGH);
}

void loop() {

    int start = digitalRead(BOUTON_START);
    int ligne = digitalRead(CAPTEUR_LIGNE);

    // STOP PRIORITAIRE
    if (debutCourse > 0 && ligne == LOW) {
        Serial.println("STOP LIGNE");
        stopMoteurs();
        while (1);
    }

    // START
    if (debutCourse == 0 && start == LOW) {
        delay(50);
        if (digitalRead(BOUTON_START) == LOW) {
            Serial.println("DEPART");
            debutCourse = millis();
        }
    }

    // RUN
    if (debutCourse > 0) {

        avancer();

        if (millis() - debutCourse > Temps_course) {

```

```
    Serial.println("FIN COURSE");  
    stopMoteurs();  
    while (1);  
  }  
}  
}
```