

Projet Arduino

Voiture robotisée & contrôlée par Bluetooth

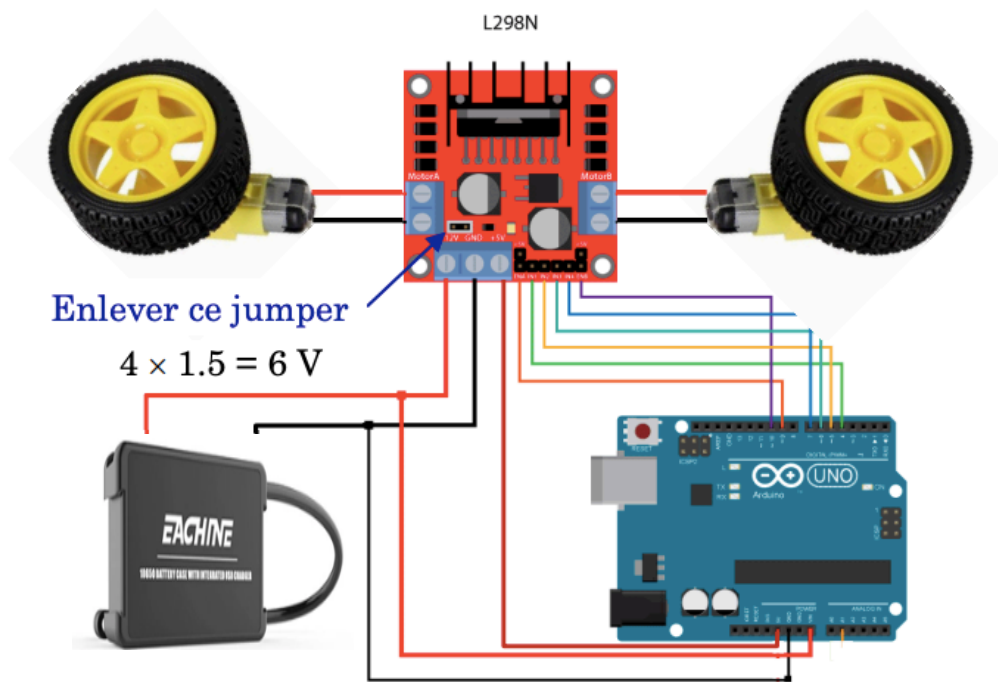
Bernard Anwar (en collaboration avec Chillat Quentin)

Rapport Semaine 4 (le 19/01/2024)

Matériels utilisés

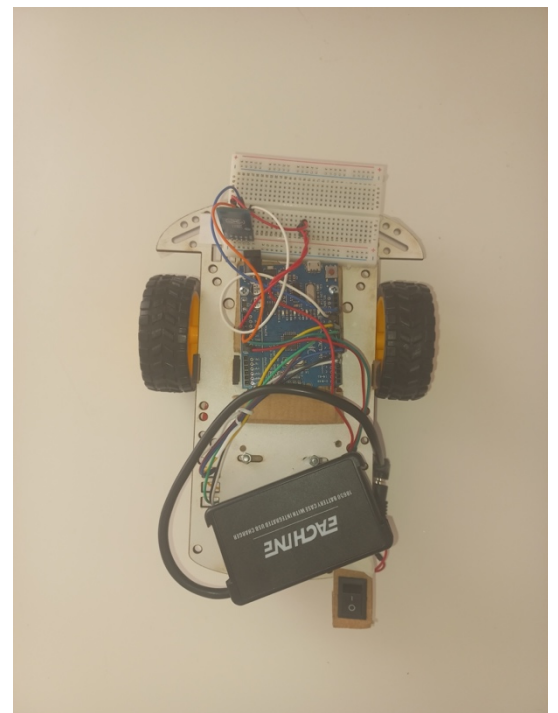
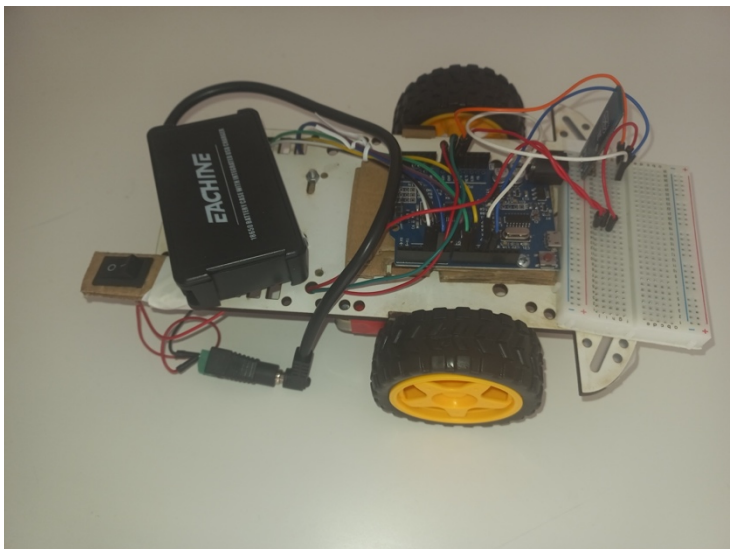
- Une batterie 18650 Eachine avec chargeur intégré
- Un connecteur (adaptateur)

Schéma du câblage



Assemblage

Ajout de la batterie 18650 Eachine à la place du porte-pile 4 piles.



Programmes de communication entre maître (HC-05) et esclave (HC-06) pour prendre en compte les données de l'accéléromètre

Codage côté maître (HC-05)

```
#include <SoftwareSerial.h>

SoftwareSerial bluetoothSerial(13, 12); // RX, TX (Connectez RX du HC-06 à 10 et TX à 11)

int16_t accelerometer_data[2] = {0};
String receivedString = "";           // Chaîne pour stocker les données reçues

int ENA=9;
int IN1=4;
int IN2=5;

int ENB=10;
int IN3=6;
int IN4=7;

void setup() {
  Serial.begin(38400);
  bluetoothSerial.begin(38400); // Initialisez le port série pour la communication Bluetooth

  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);

  analogWrite(ENA, 255);
  analogWrite(ENB, 255);
}

void loop() {
  // Vérifier s'il y a des données disponibles sur Bluetooth
  while (bluetoothSerial.available()) {
    char receivedChar = bluetoothSerial.read();

    // Si une virgule ou un saut de ligne est reçu, cela signifie la fin d'une valeur
    if (receivedChar == ',' || receivedChar == '\n') {
      // Convertir la chaîne reçue en entier
      int receivedValue = receivedString.toInt();

      // Mettre à jour les valeurs du tableau d'accéléromètre à partir des données reçues
      for (int i = 0; i < 2; i++) {
        if (accelerometer_data[i] == 0) {
          accelerometer_data[i] = receivedValue;
          break;
        }
      }
    }
  }
}
```

```

// Réinitialisez la chaîne pour la prochaine valeur
receivedString = "";

// Si le tableau est complètement construit, afficher chaque valeur
if (accelerometer_data[0] != 0 && accelerometer_data[1] != 0 ) {
    Serial.print("Tableau d'accéléromètre complet : ");
    Serial.print(accelerometer_data[0]);
    Serial.print(", ");
    Serial.println(accelerometer_data[1]);

    if(accelerometer_data[1]<-10000){ //avancer
        digitalWrite(IN1,HIGH);
        digitalWrite(IN2,LOW);
        digitalWrite(IN3,LOW);
        digitalWrite(IN4,HIGH);
    }
    else if(accelerometer_data[1]>10000){//reculer
        digitalWrite(IN1,LOW);
        digitalWrite(IN2,HIGH);
        digitalWrite(IN3,HIGH);
        digitalWrite(IN4,LOW);
    }

    else if(accelerometer_data[0]<-10000){//tourner a gauche
        digitalWrite(IN1,HIGH);
        digitalWrite(IN2,LOW);
        digitalWrite(IN3,HIGH);
        digitalWrite(IN4,LOW);
    }

    else if(accelerometer_data[0]>10000){//tourner a droite
        digitalWrite(IN1,LOW);
        digitalWrite(IN2,HIGH);
        digitalWrite(IN3,LOW);
        digitalWrite(IN4,HIGH);
    }
    else{ //static
        digitalWrite(IN1,LOW);
        digitalWrite(IN2,LOW);
        digitalWrite(IN3,LOW);
        digitalWrite(IN4,LOW);
    }

    // Réinitialisez le tableau pour la prochaine série de valeurs
    memset(accelerometer_data, 0, sizeof(accelerometer_data));
}
} else {
    // Ajoutez le caractère à la chaîne en cours de réception
    receivedString += receivedChar;
}
}
}

```

Codage côté esclave (HC-06)

```
#include <Wire.h>
#include <SoftwareSerial.h>

SoftwareSerial bluetoothSerial(10, 11); // RX, TX (Connectez RX du HC-05 à 10 et TX à 11)
#define MPU6050_ADDRESS 0x68           // Adresse I2C du MPU6050
int16_t accelerometer_x, accelerometer_y, accelerometer_z;
char tmp_str[7]; // variable temporaire utilisée dans la fonction de conversion

char* convert_int16_to_str(int16_t i) { // convertit int16 en chaîne de caractères. De
plus, les chaînes résultantes auront la même longueur dans le moniteur de débogage.
    sprintf(tmp_str, "%d", i);
    return tmp_str;
}

void setup() {
    Serial.begin(38400);
    Wire.begin();
    Wire.beginTransmission(MPU6050_ADDRESS); // Commence une transmission à l'esclave I2C
(carte GY-521)
    Wire.write(0x6B); // Registre PWR_MGMT_1
    Wire.write(0); // réglé à zéro (réveille le MPU-6050)
    Wire.endTransmission(true);
    bluetoothSerial.begin(38400);
}

void loop() {
    // Lire les données de l'accéléromètre (MPU6050)
    Wire.beginTransmission(MPU6050_ADDRESS);
    Wire.write(0x3B); // Adresse du registre de l'accélération X haute
    Wire.endTransmission(false);
    Wire.requestFrom(MPU6050_ADDRESS, 7 * 2, true);

    accelerometer_x = Wire.read() << 8 | Wire.read();
    accelerometer_y = Wire.read() << 8 | Wire.read();
    accelerometer_z = Wire.read() << 8 | Wire.read();

    // Envoyer les données à HC-06 sous forme de chaînes de caractères
    bluetoothSerial.print(convert_int16_to_str(accelerometer_x));
    bluetoothSerial.print(",");
    bluetoothSerial.println(convert_int16_to_str(accelerometer_y));
    /*bluetoothSerial.print(",");
    bluetoothSerial.println(convert_int16_to_str(accelerometer_z));*/

    // Attendre un court délai pour que les données soient envoyées
    delay(500);
}
```

Tâches effectuées :

Nous avons remplacé le porte-pile par une batterie.

Nous avons étudié les messages envoyés par l'accéléromètre selon l'inclinaison du poignet muni du gant. Ces données nous ont permis de corriger les programmes.

Nous avons modifié les programmes de communication Bluetooth (voir au-dessus).

Tests effectués :

Nous avons testé l'orientation des roues selon l'inclinaison du poignet muni du gant.

Lorsque le poignet est baissé, la voiture avance.

Lorsque le poignet est levé, la voiture recule.

Lorsque le poignet est incliné à droite, la voiture va à droite.

Lorsque le poignet est incliné à gauche, la voiture va à gauche.

Le temps de réaction est plutôt immédiat.

La voiture exécute les instructions même lorsqu'elle n'est pas branchée à l'ordinateur.

Réalisation à prévoir pour la semaine 5.

Perfectionner l'exécution des instructions par la voiture.