

Projet Arduino

Voiture robotisée & contrôlée par Bluetooth

Bernard Anwar (en collaboration avec Chillat Quentin)

Rapport Semaine 3 (le 12/01/2024)

Matériels utilisés

- Deux cartes Arduino UNO (et deux câbles USB),
- Deux modules Bluetooth 2.4 GHz (HC-05 maître et HC-06 esclave),
- Deux plaques d'essai et des fils pour câbler notre montage,
- Un accéléromètre
- Un gant

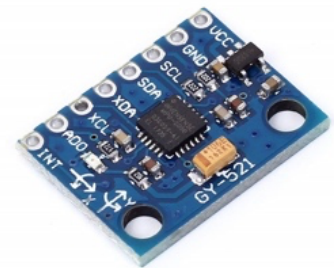
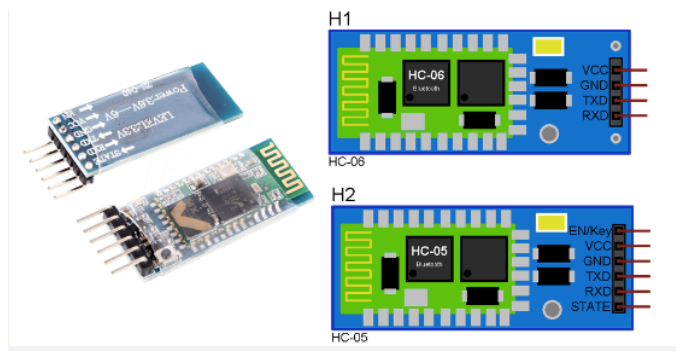
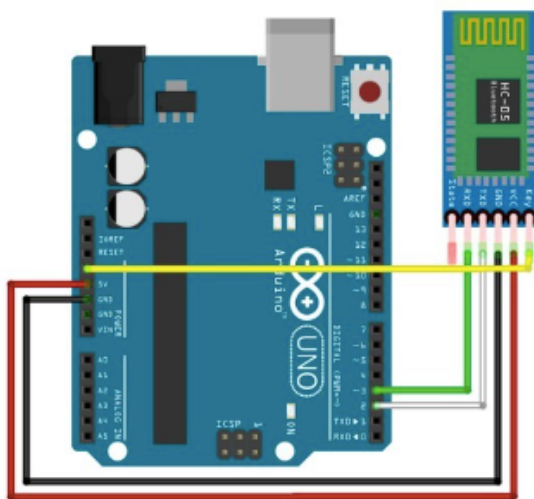
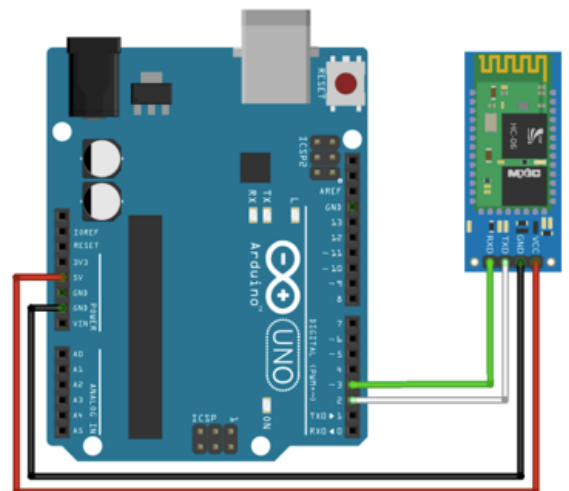


Schéma du câblage



HC-05



HC-06

Assemblage

Connexion de la plaque Arduino avec le module HC-05

Reliage des pins :

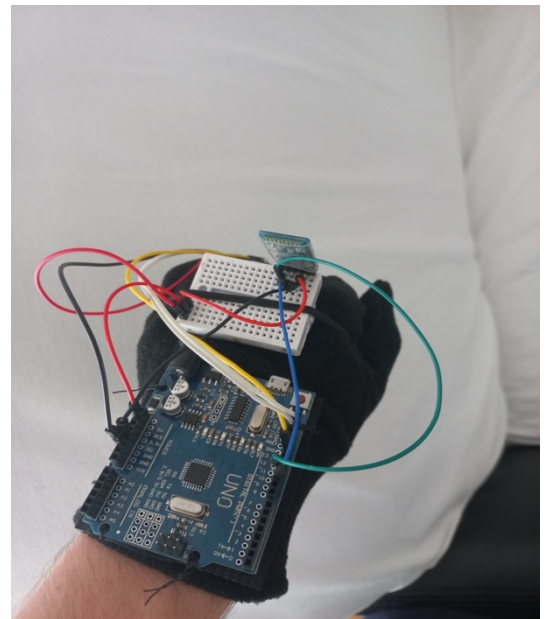
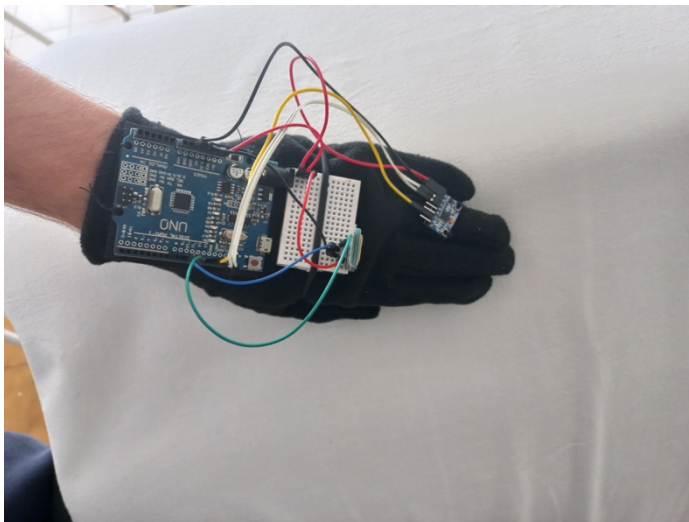
- RX→pin10
- TX→pin11

Connexion de la plaque Arduino avec le module HC-06

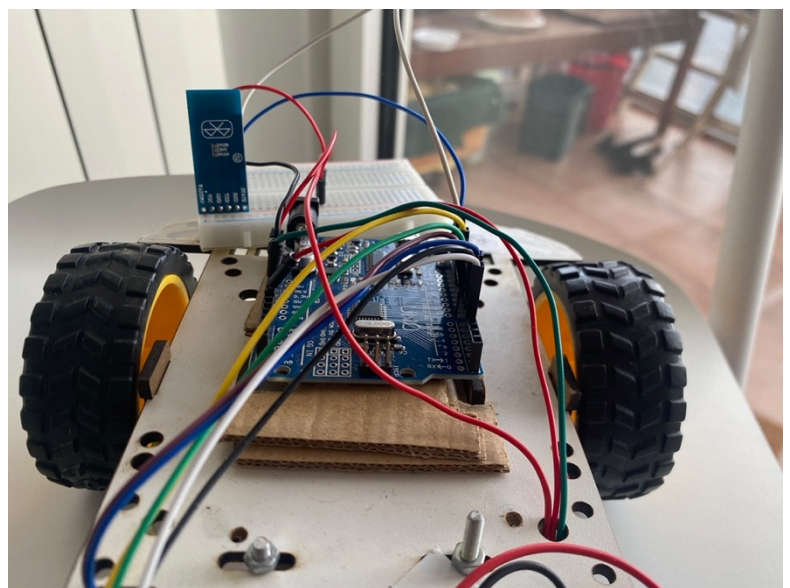
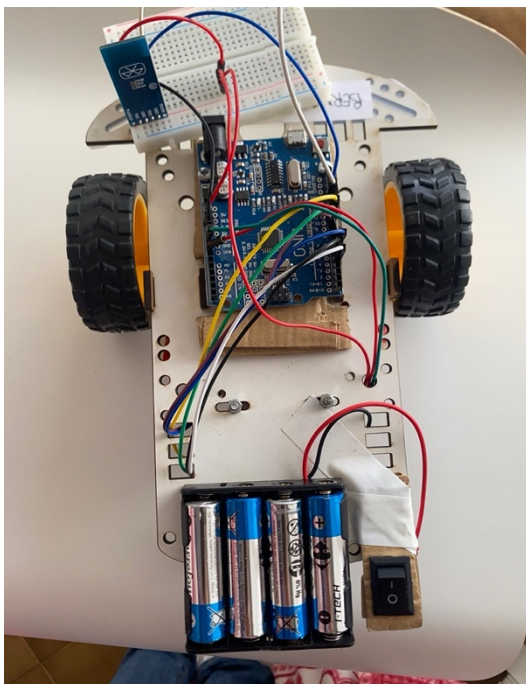
Reliage des pins :

- RX→pin13
- TX→pin12

Fixation du module HC-05 sur un gant.



Ajout de l'accéléromètre



Programmes de communication entre maître et esclave

Codage côté maître

```
#include<SoftwareSerial.h>

#define RX 10
#define TX 11

SoftwareSerial BlueT (RX,TX);

void setup() {
  Serial.begin(38400);
  delay(500);
  Serial.println("Bonjour -Pret pour les commandes AT");
  BlueT.begin(38400);
  delay(500);}

void loop() {
  while (BlueT.available()) {
    Serial.print(char(BlueT.read())); }
  while (Serial.available()) {
    BlueT.write(char(Serial.read())); }
}
```

Codage côté esclave

```
#include<SoftwareSerial.h>

#define RX 13
#define TX 12

SoftwareSerial BlueT(RX,TX);

void setup() {
  Serial.begin(38400);
  delay(500);
  Serial.println("Bonjour -Pret pour les commandes AT");
  BlueT.begin(38400);
  delay(500);}

void loop() {
  while (BlueT.available()) {
    Serial.print(char(BlueT.read())); }
  while (Serial.available()) {
    BlueT.write(char(Serial.read())); }
}
```

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

// Adresse I2C de l'accéléromètre GY-521
#define MPU_ADDR 0x68

SoftwareSerial bluetoothSerial(10, 11); // RX, TX pour le module Bluetooth HC-05

int16_t accelerometer_x, accelerometer_y, accelerometer_z; // variables for accelerometer raw data
char tmp_str[7]; // temporary variable used in convert function

char* convert_int16_to_str(int16_t i) { // converts int16 to string. Moreover, resulting strings will
have the same length in the debug monitor.
    sprintf(tmp_str, "%6d", i);
    return tmp_str;
}

void setup() {
    Serial.begin(9600);
    Wire.begin();
    Wire.beginTransmission(MPU_ADDR); // Begins a transmission to the I2C slave (GY-521
board)
    Wire.write(0x6B); // PWR_MGMT_1 register
    Wire.write(0); // set to zero (wakes up the MPU-6050)
    Wire.endTransmission(true);
    bluetoothSerial.begin(38400);
}

void loop() {
    Wire.beginTransmission(MPU_ADDR);
    Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H) [MPU-6000 and MPU-
6050 Register Map and Descriptions Revision 4.2, p.40]
    Wire.endTransmission(false); // the parameter indicates that the Arduino will send a restart. As a
result, the connection is kept active.
    Wire.requestFrom(MPU_ADDR, 7*2, true); // request a total of 7*2=14 registers

    // "Wire.read()<<8 | Wire.read();" means two registers are read and stored in the same variable
    accelerometer_x = Wire.read()<<8 | Wire.read(); // reading registers: 0x3B (ACCEL_XOUT_H)
and 0x3C (ACCEL_XOUT_L)
    accelerometer_y = Wire.read()<<8 | Wire.read(); // reading registers: 0x3D (ACCEL_YOUT_H)
and 0x3E (ACCEL_YOUT_L)
    accelerometer_z = Wire.read()<<8 | Wire.read(); // reading registers: 0x3F (ACCEL_ZOUT_H)
and 0x40 (ACCEL_ZOUT_L)

    // Envoyer les données via Bluetooth
    bluetoothSerial.print("X");
    bluetoothSerial.print(convert_int16_to_str(accelerometer_x));
    Serial.print("X"); Serial.print(convert_int16_to_str(accelerometer_x));
    bluetoothSerial.print(",Y");
    bluetoothSerial.print(convert_int16_to_str(accelerometer_y));
    Serial.print(",Y"); Serial.print(convert_int16_to_str(accelerometer_y));
    bluetoothSerial.print(",Z");
    bluetoothSerial.println(convert_int16_to_str(accelerometer_z));
    Serial.print(",Z"); Serial.print(convert_int16_to_str(accelerometer_z));
    Serial.println();
    delay(500); // Ajoutez un délai si nécessaire
}
```

Codage côté esclave (HC-06)

```
#include <SoftwareSerial.h>

SoftwareSerial bluetoothSerial(13, 12); // RX, TX pins

void setup() {
  Serial.begin(38400);
  bluetoothSerial.begin(38400);
}

void loop() {
  Serial.println("LOOP OK");
  while (bluetoothSerial.available()) {
    Serial.println("Bluetooth ok");
    // Lecture des données reçues via Bluetooth
    String receivedData = bluetoothSerial.readString();

    // Affichage des données reçues sur le moniteur série
    Serial.println("Données reçues : " + receivedData);
  }
}
```

Tests effectués :

Nous avons commencé par essayer de régler le problème de la transmission de l'accéléromètre sur le moteur A. Les modules de communication RF étant usés et sensibles aux perturbations extérieures, nous avons opté pour une communication Bluetooth.

Vérification de la connexion Bluetooth entre les modules HC-05 et HC-06 réussie. Tests réussis sur les formats des données émises par l'accéléromètre via le HC-05 et leurs réceptions par le HC-06.

Réalisation à prévoir pour la semaine 4.

Vérification que les directives données par l'accéléromètre soient correctement réalisées par le système embarqué de la voiture.