

## RAPPORT DE SEANCE DU 3/02/2020

Pour cette 5<sup>ème</sup> séance, j'ai voulu synchroniser pour une bonne fois pour toute, les deux modules Bluetooth pour pouvoir commencer à entrer les premiers codes pour commander la voiture avec le gant (que l'on commencera à construire la semaine prochaine). Mais on a rencontré quelques complications. En effet pendant que Thibault lui se chargeait de réfléchir à comment augmenter la puissance de nos moteurs, vous nous aviez fait prendre compte qu'on était en court-circuit donc notre module HC-06 était défectueux. Il y a donc eu ce problème, j'ai donc pas pu appliquer les commandes AT sur le module HC-06 (un coup ça marchait un coup non), dû à ce problème je n'ai donc pas pu synchroniser les 2 modules Bluetooth ou même on ne pouvait donc pas se connecter avec notre téléphone car on voulait faire les premiers test communication RF et commander la voiture avec notre téléphone pour s'assurer que tout fonctionne, mais on a pas réussi à établir la connexion entre le module HC-06.

Du côté du HC-05, c'est moins catastrophique, les commandes AT fonctionnent (AT+ROLE, AT+RMAAD, AT+BIND=0, AT+INQM etc), mais lorsque je veux demander l'affichage des modules esclaves détectés on m'affiche (ERROR(16)) car la librairie SPP n'a pas été initialisée mais je n'arrive pas à trouver cette librairie en question sur internet. A part ces problèmes de modules Bluetooth que l'on espère régler avant la prochaine séance, il nous manque à finir de configurer le Lilypad pour avoir 2 cartes Arduino opérationnelles (avec l'Arduino Uno) à notre projet et entrer les premiers code de commandes pour la voiture.

Donc pour ne pas prendre trop de retard, à partir de l'algorithme qu'on avait déjà préétabli la dernière fois, et des codes élémentaires que l'on a sur l'accéléromètre, la voiture, et les modules Bluetooth, j'ai commencé à assembler ces premiers codes ensemble pour pouvoir coder la transmission pour le gant et la réception pour la voiture, pour l'instant ce n'est que simpliste et synthétique, on s'assurera du bon fonctionnement lorsqu'on aura régler les problèmes précédents et qu'on pourra tester ces codes.

Thibault, lui s'est chargé de réfléchir à comment augmenter la puissance des moteurs, et a pu comprendre le fonctionnement des résistances flexibles, qui serviront à donner des mouvements supplémentaires à la voitures à partir de flexions de la main. A venir on réfléchira aussi à comment on va retourner la voiture à partir d'un système de propulser ou autre, système que l'on actionnera avec les flexions des doigts de la main.

# RAPPORT DE SEANCE DU 3/02/2020

```
GOLVE1 | Arduino 1.8.9
Fichier Edition Croquis Outils Aide

GOLVE1

#include <Wire.h> //BIBLIOTHEQUE WIRE POUR L'ACCELEROMETRE

const int MPU_addr = 0x68;
float Acc_X, Acc_Y, Acc_Z;
float Gy_X, Gy_Y, Gy_Z;
float An_X, An_Y, An_Z;
float rad_deg=180/3.1415926535897932384626433832795;
float Tmp;

#include <SoftwareSerial.h> //BIBLIOTHEQUE BLUETOOTH
SoftwareSerial BT(9,3); //MODULE HC-05 MASTER

void setup() {
  // put your setup code here, to run once:
  //PARTIE BLUETOOTH
  Serial.begin(9600);
  Serial.println("Bonjour -Pret pour les commandes AT");
  BT.begin(38400);

  //PARTIE ACCELEROMETRE
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);
  Wire.write(0);
  Wire.endTransmission(true);
  Wire.beginTransmission(0x68);
  Wire.write(0x1C);
  Wire.write(0x00000000);
  Wire.endTransmission();

  Enregistrement terminé.

47
Taper ici pour rechercher
Arduino Nano, ATmega328P (Old Bootloader) sur COM3
20:03
04/02/2020
```

```
GOLVE1 | Arduino 1.8.9
Fichier Edition Croquis Outils Aide

GOLVE1

void loop() {
  // put your main code here, to run repeatedly:
  //ON RECUPERE LES DONNEES DE L'ACCELEROMETRE
  Wire.beginTransmission(0x68);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(0x68,7*2,true);
  Acc_X=(Wire.read()<<8|Wire.read())/2048;
  Acc_Y=(Wire.read()<<8|Wire.read())/2048;
  Acc_Z=(Wire.read()<<8|Wire.read())/2048;

  Tmp = Wire.read() << 8 | Wire.read();

  Gy_X = Wire.read() << 8 | Wire.read();
  Gy_Y = Wire.read() << 8 | Wire.read();
  Gy_Z = Wire.read() << 8 | Wire.read();

  An_X=(atan((Acc_X)/sqrt(pow((Acc_Y),2)+pow((Acc_Z),2))) *rad_deg);
  An_Y=(atan((Acc_Y)/sqrt(pow((Acc_X),2)+pow((Acc_Z),2))) *rad_deg);
  An_Z=(atan((Acc_Z)/sqrt(pow((Acc_Y),2)+pow((Acc_X),2))) *rad_deg);

  Serial.print("Ax = "); Serial.print(Acc_X);
  Serial.print(" | Ay = "); Serial.print(Acc_Y);
  Serial.print(" | Az = "); Serial.print(Acc_Z);

  Serial.print(" | Gy_X = "); Serial.print(Gy_X);
  Serial.print(" | Gy_Y = "); Serial.print(Gy_Y);
  Serial.print(" | Gy_Z = "); Serial.print(Gy_Z);

  Enregistrement terminé.

47
Taper ici pour rechercher
Arduino Nano, ATmega328P (Old Bootloader) sur COM3
20:03
04/02/2020
```

# RAPPORT DE SEANCE DU 3/02/2020

```
00 GOLVE1 | Arduino 1.8.9
Fichier Édition Croquis Outils Aide

GOLVE1

Serial.print(" | An_Y = "); Serial.print(An_Y);
Serial.print(" | An_Z = "); Serial.println(An_Z);

delay(500);

//COMMUNICATION AVEC LE MODULE HC-05
while (BT.available()) {
  if ((An_X)<90 && An_Y)<10){
    BT.write('S') //POSITION NEUTRE
  }
  else if((An_X)<10 && An_Y<-10){
    BT.write('A') //AVANCER
  }
  else if(An_X<-10 && An_Y<-10){
    BT.write('D') //AVANCER DROITE
  }
  else if(An_X>10 && An_Y>-10){
    BT.write('G') //AVANCER GAUCHE
  }
  else if((An_X)<10 && An_Y>10){
    BT.write('R') //RECULER
  }
  else if(An_X<-10 && An_Y>10){
    BT.write('W') //RECULER DROITE
  }
  else if(An_X>10 && An_Y>10){
    BT.write('L') //RECULER GAUCHE
  }
}

Enregistrement terminé.

47 Arduino Nano, ATmega328P (Old Bootloader) sur COM3
Taper ici pour rechercher 20:03 04/02/2020
```

```
00 CAR1 | Arduino 1.8.9
Fichier Édition Croquis Outils Aide

CAR1

#include <SoftwareSerial.h>
SoftwareSerial BT(11,3); //MODULE HC-06 SLAVE
int ENA=9;
int ENB=10;
int IN1=4;
int IN2=5;
int IN3=6;
int IN4=7;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(ENA,OUTPUT); // Configurer
  pinMode(ENB,OUTPUT); // les broches
  pinMode(IN1,OUTPUT); // comme sortie
  pinMode(IN2,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);

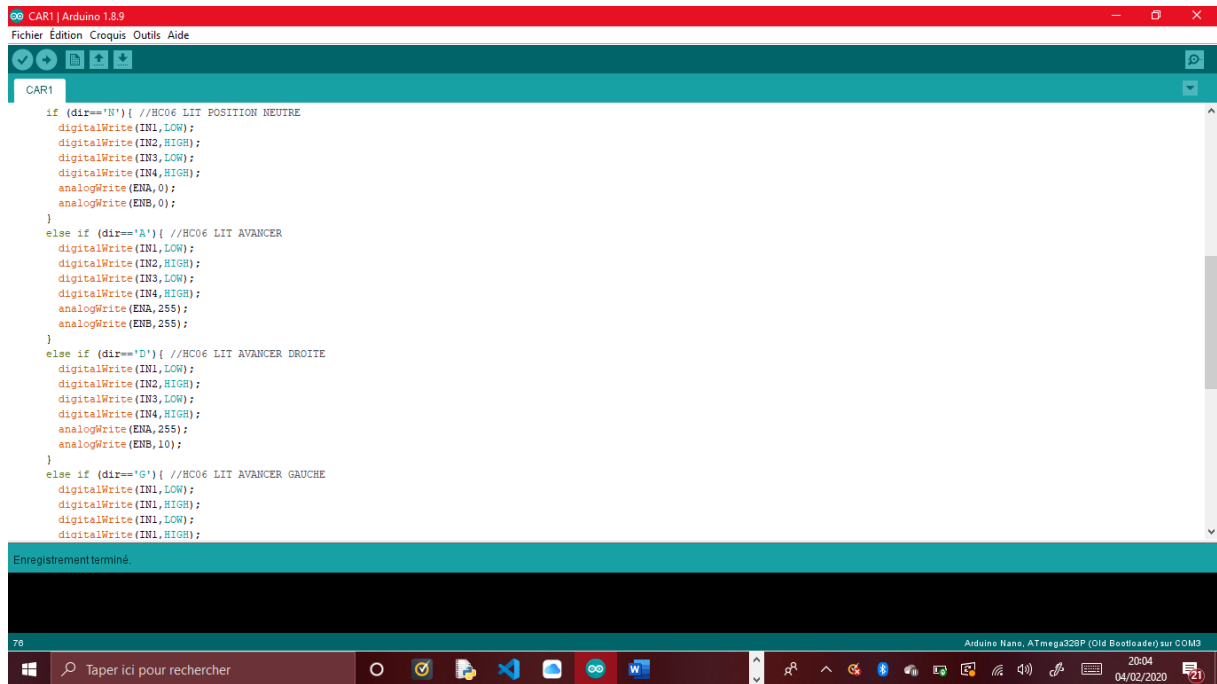
  digitalWrite(ENA,LOW); // Moteur A - Ne pas tourner
  digitalWrite(ENB,LOW); // Moteur B - Ne pas tourner
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
}

void loop() {
  // put your main code here, to run repeatedly:
  while (BT.available()) {

Enregistrement terminé.

76 Arduino Nano, ATmega328P (Old Bootloader) sur COM3
Taper ici pour rechercher 20:04 04/02/2020
```

# RAPPORT DE SEANCE DU 3/02/2020

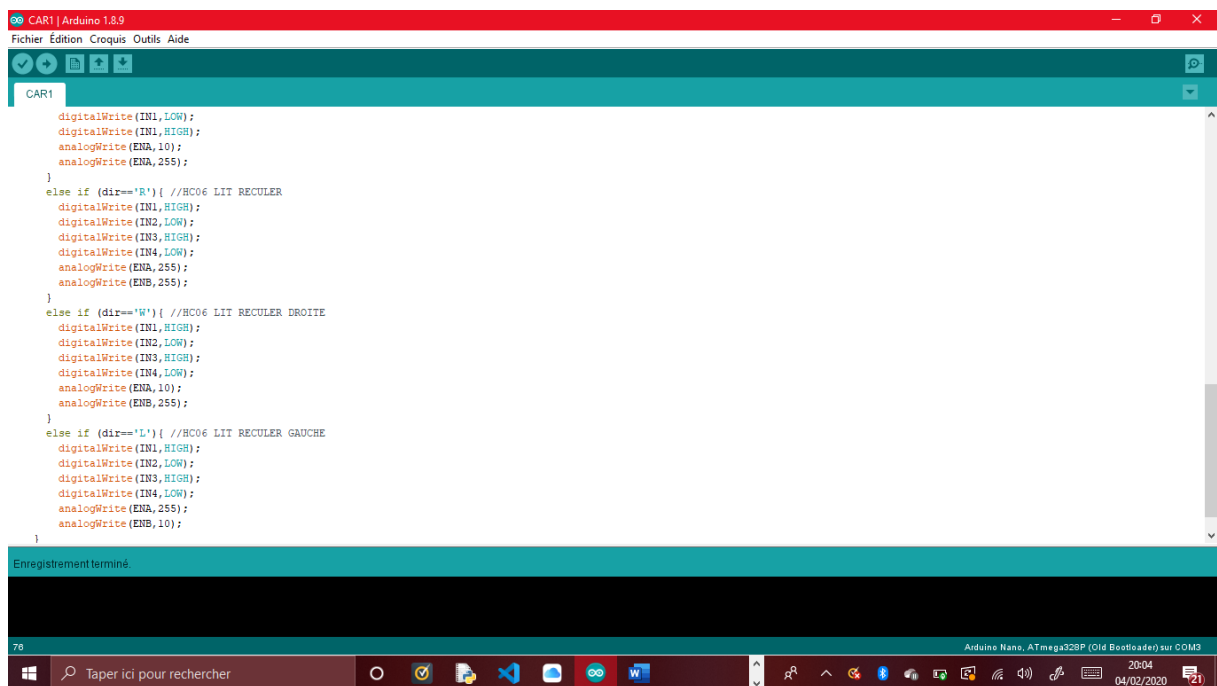


```
CAR1

if (dir=='N'){ //HC06 LIT POSITION NEUTRE
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);
  analogWrite(ENA,0);
  analogWrite(ENB,0);
}
else if (dir=='A'){ //HC06 LIT AVANCER
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);
  analogWrite(ENA,255);
  analogWrite(ENB,255);
}
else if (dir=='D'){ //HC06 LIT AVANCER DROITE
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);
  analogWrite(ENA,255);
  analogWrite(ENB,10);
}
else if (dir=='G'){ //HC06 LIT AVANCER GAUCHE
  digitalWrite(IN1,LOW);
  digitalWrite(IN1,HIGH);
  digitalWrite(IN3,LOW);
  digitalWrite(IN3,HIGH);
}
```

Enregistrement terminé.

76 Arduino Nano, ATmega328P (Old Bootloader) sur COM3



```
digitalWrite(IN1,LOW);
digitalWrite(IN1,HIGH);
analogWrite(ENA,10);
analogWrite(ENA,255);
}
else if (dir=='R'){ //HC06 LIT RECULER
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
  analogWrite(ENA,255);
  analogWrite(ENB,255);
}
else if (dir=='W'){ //HC06 LIT RECULER DROITE
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
  analogWrite(ENA,10);
  analogWrite(ENB,255);
}
else if (dir=='L'){ //HC06 LIT RECULER GAUCHE
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
  analogWrite(ENA,255);
  analogWrite(ENB,10);
}
}
```

Enregistrement terminé.

76 Arduino Nano, ATmega328P (Old Bootloader) sur COM3