

## Rapport du 06/01/2020 :

### Travail hors séances :

- Construction de la voiture :
  - Fixation des composants.
  - Utilisation de scotchs double face pour coller les moteurs aux planches (remplaçables par de la colle)
  - Fixation roues-moteurs avec de la colle.
- Défaut de la voiture :
  - Problèmes avec les roulettes si l'on veut retourner la voiture sur le mur.
  - Puissance trop faible des moteurs.
  - Pas de place pour insérer les piles sur la voiture.
  - Les roues qui ne tiennent pas.

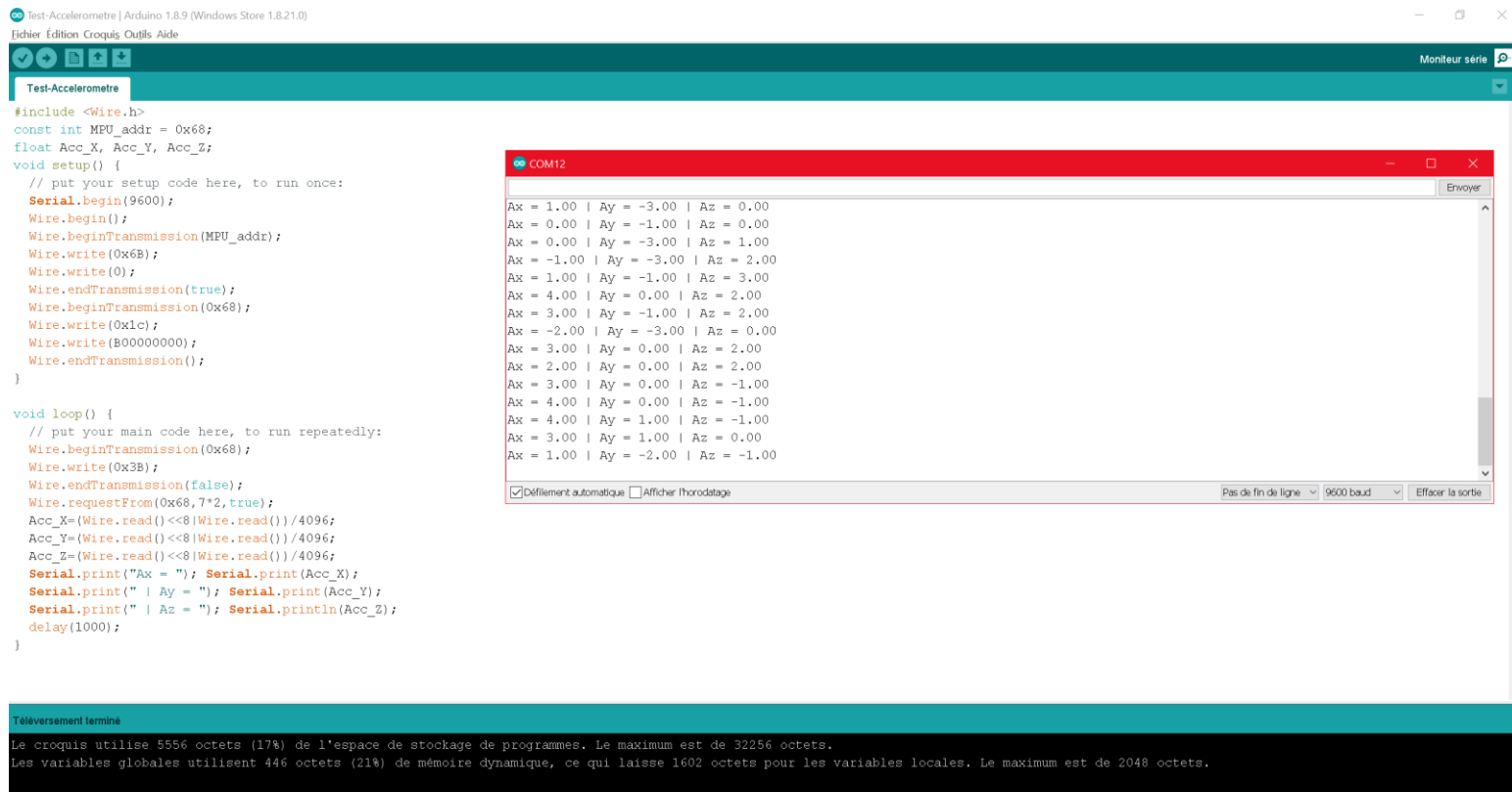


# Travail séance :

- Modification sur la voiture :

Essais de différents types de fixations pour les roues sur les moteurs(colle, plastique modelable...). On a aussi agrandi le trou de la planche en bois pour pouvoir accéder plus facilement aux piles et aux branchements sur la voiture.

- Test de code sur l'accéléromètre et compréhension de son fonctionnement. J'essaye d'abord comment le capteur réagit aux différents mouvements qu'on lui fait faire (rotation, accélération...), afin de pouvoir intégrer le code nécessaire avec celui de la voiture.
- Tsiory s'est occupé principalement de fixer les roues. Il a également pu initialiser les deux modules bluetooth( HC-05 et HC-06), et les connecter entre eux. Il a également lu la documentation nécessaire aux modules.



The screenshot shows the Arduino IDE interface. The main window displays the code for a test program. The code includes the `Wire` library and defines an MPU6050 sensor. It sets up the I2C bus and the serial port, then reads the sensor data in the `loop` function and prints it to the serial monitor.

```
#include <Wire.h>
const int MPU_addr = 0x68;
float Acc_X, Acc_Y, Acc_Z;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  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();
}

void loop() {
  // put your main code here, to run repeatedly:
  Wire.beginTransmission(0x68);
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(0x68, 7*2, true);
  Acc_X=(Wire.read() << 8 | Wire.read()) / 4096;
  Acc_Y=(Wire.read() << 8 | Wire.read()) / 4096;
  Acc_Z=(Wire.read() << 8 | Wire.read()) / 4096;
  Serial.print("Ax = "); Serial.print(Acc_X);
  Serial.print(" | Ay = "); Serial.print(Acc_Y);
  Serial.print(" | Az = "); Serial.println(Acc_Z);
  delay(1000);
}
```

The serial monitor window (COM12) shows the output of the program, displaying the acceleration values for Ax, Ay, and Az in g-units. The values are printed every second.

```
Ax = 1.00 | Ay = -3.00 | Az = 0.00
Ax = 0.00 | Ay = -1.00 | Az = 0.00
Ax = 0.00 | Ay = -3.00 | Az = 1.00
Ax = -1.00 | Ay = -3.00 | Az = 2.00
Ax = 1.00 | Ay = -1.00 | Az = 3.00
Ax = 4.00 | Ay = 0.00 | Az = 2.00
Ax = 3.00 | Ay = -1.00 | Az = 2.00
Ax = -2.00 | Ay = -3.00 | Az = 0.00
Ax = 3.00 | Ay = 0.00 | Az = 2.00
Ax = 2.00 | Ay = 0.00 | Az = 2.00
Ax = 3.00 | Ay = 0.00 | Az = -1.00
Ax = 4.00 | Ay = 0.00 | Az = -1.00
Ax = 4.00 | Ay = 1.00 | Az = -1.00
Ax = 3.00 | Ay = 1.00 | Az = 0.00
Ax = 1.00 | Ay = -2.00 | Az = -1.00
```

At the bottom of the IDE, a status bar indicates that the upload was successful and provides information about the memory usage of the sketch.

Téléversement terminé  
Le croquis utilise 5556 octets (17%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.  
Les variables globales utilisent 446 octets (21%) de mémoire dynamique, ce qui laisse 1602 octets pour les variables locales. Le maximum est de 2048 octets.

