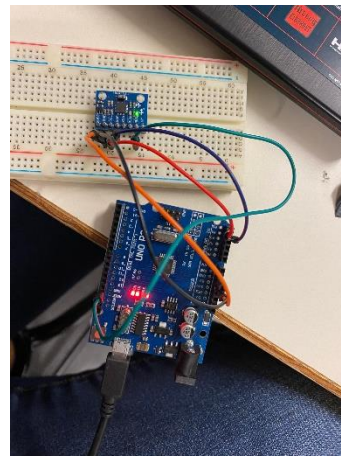
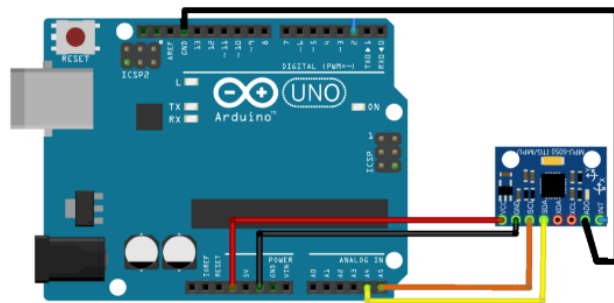


## Compte rendu séance 2

Pendant cette séance, je me suis penché sur le fonctionnement de l'accéléromètre et plus particulièrement le code permettant de prendre des mesures.

Pour le branchement de l'accéléromètre, j'ai copié celui du cours d'Arduino :



J'ai ensuite suivi la démarche sur comment obtenir des données dites « brutes » du site <http://gilles.thebault.free.fr/spip.php?article32> (installation bibliothèque puis code) :

Ici les données dites « brutes » :

```
Test_Valeur_msd | Arduino 1.8.15
Fichier Edition Croquis Outils Aide

// MPU-6050 Short Example Sketch
// By Arduino User @ohiochil
// August 17, 2014
// Public Domain

#include <Wire.h>

const int MPU_addr=0x68; // I2C address of the MPU-6050
int16_t AccX,AccY,AccZ,Tmp,OyX,OyY,OyZ;

void setup() {
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B); // PWR_MGMT_1 register
  Wire.write(0); // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);
  Serial.begin(9600);
}

void loop() {
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers
  AccX=Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AccY=Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AccZ=Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Tmp=Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
  OyX=Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
  OyY=Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
  OyZ=Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
  Serial.print("AccX = "); Serial.print(AccX);
  Serial.print(" | AccY = "); Serial.print(AccY);
  Serial.print(" | AccZ = "); Serial.print(AccZ);
  Serial.print(" | Tmp = "); Serial.print(Tmp/340.00+36.53); //reguation for temperature in degrees C from datasheet
  Serial.print(" | OyX = "); Serial.print(OyX);
  Serial.print(" | OyY = "); Serial.print(OyY);
  Serial.print(" | OyZ = "); Serial.print(OyZ);
  delay(333);
}
```

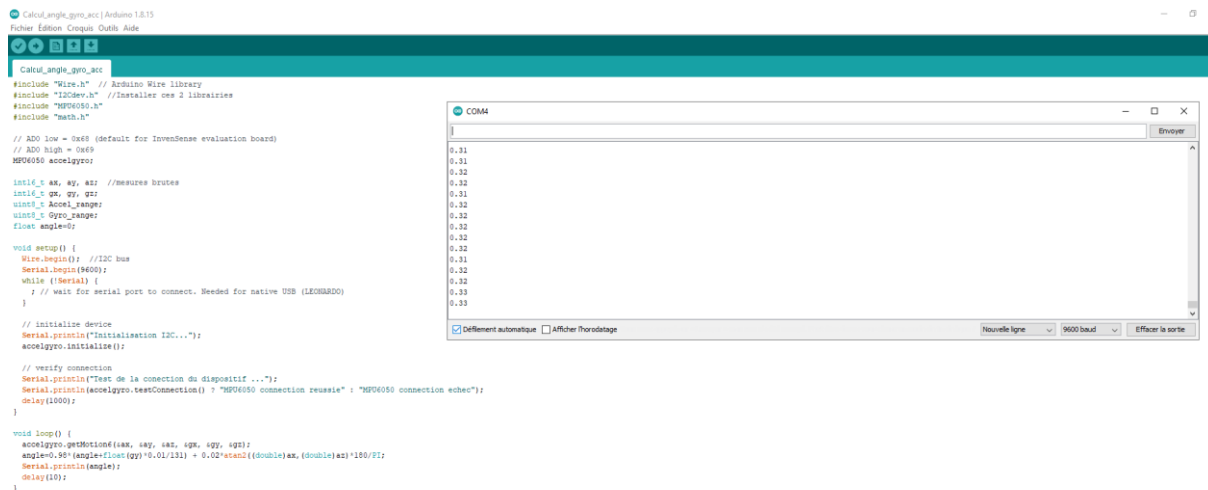
COMM

Avant mouvement

Après mouvement

Problème actuel : Je ne sais pas vraiment à quoi correspondent chaque composante et si les valeurs de bases (quand la GY-521 ne bouge pas) sont constantes ou changent à chaque fois que l'on met en marche la puce.

Ici les données obtenues (angle en combinant le gyropode et l'accéléromètre avec un "filtre complémentaire") :



The screenshot shows the Arduino IDE interface. The main window displays a C++ sketch for an Arduino Uno. The sketch includes libraries for Wire, I2Cdev, MPU6050, and Math. It defines variables for I2C addresses, sensor ranges, and a float for the angle. The setup function initializes the I2C bus, serial port, and sensors. The loop function reads sensor data and calculates the angle using a complementary filter formula.

```
#include <Wire.h> // Arduino Wire library
#include <I2Cdev.h> // Installer ces 2 librairies
#include <MPU6050.h>
#include <math.h>

// AD0 low = 0x69 (default for InvenSense evaluation board)
// AD0 high = 0x68
MPU6050 accelgyro;

int16_t ax, ay, az; //mesures brutes
int16_t gx, gy, gz;
uint16_t accel_range;
uint16_t gyro_range;
float angle=0;

void setup() {
  Wire.begin(); //I2C bus
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB (LEONARDO)
  }

  // initialize device
  Serial.println("Initialisation I2C...");
  accelgyro.initialize();

  // verify connection
  Serial.println("Test de la connexion du dispositif ...");
  Serial.println(accelgyro.testConnection() ? "MPU6050 connection réussie" : "MPU6050 connection echec");
  delay(1000);
}

void loop() {
  accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  angle=0.98x(angle+Gy*0.01/131) + 0.02x(angle de
  Serial.println(angle);
  delay(10);
}
```

The serial monitor window shows a list of numerical values, likely the angle calculated by the sketch:

```
0.31
0.31
0.32
0.32
0.31
0.32
0.32
0.32
0.32
0.32
0.31
0.32
0.32
0.33
0.33
```

➔ Voir vidéo essai prise de mesure de l'angle

On peut voir suite à cela que le code ci-dessus ne permet que d'obtenir les données que d'un angle

(via la formule :  $\text{angle} = 0.98 \times (\text{angle} + \text{Gy} \times 0.01 / 131) + 0.02 \times (\text{angle de l'accéléromètre en } ^\circ)$ )

Finalement, suite à cette séance, j'ai pu déterminer comment marchait la puce GY-521 couplé au code mais le problème est que maintenant reconnaître via ces données la position de la GY-521 suite à un mouvement, ce que je ne sais pas comment encore faire.