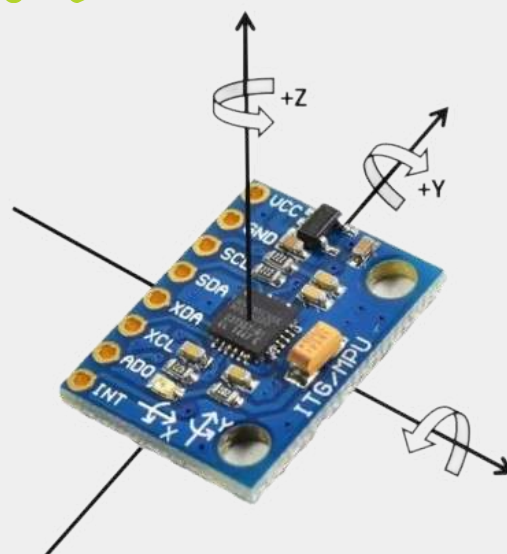


POLYTECH<sup>®</sup>  
NICE SOPHIA



UNIVERSITÉ  
CÔTE D'AZUR

# MPU-6050



on



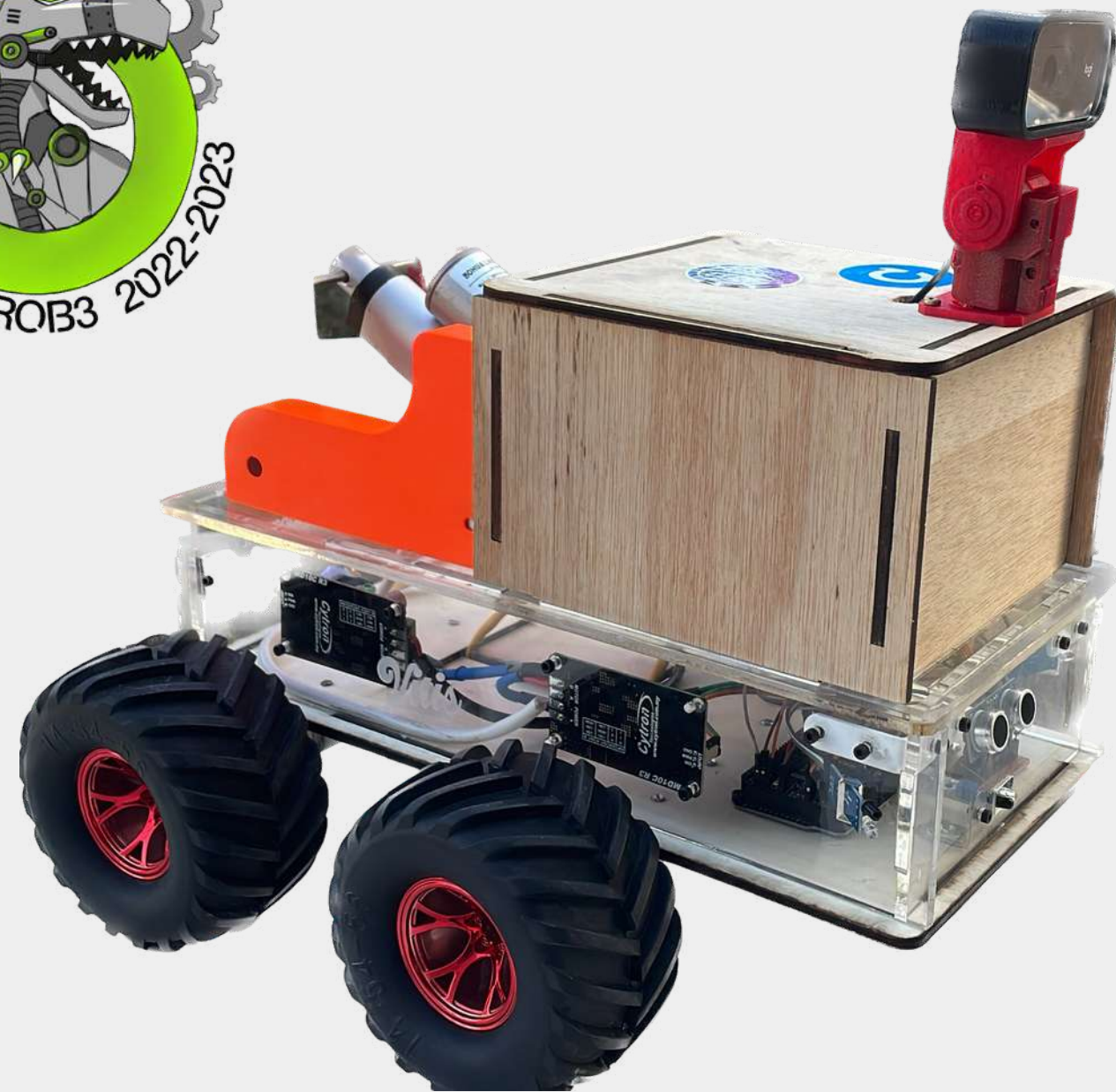
---

*Demoron Tanguy - Chavant Arthur*



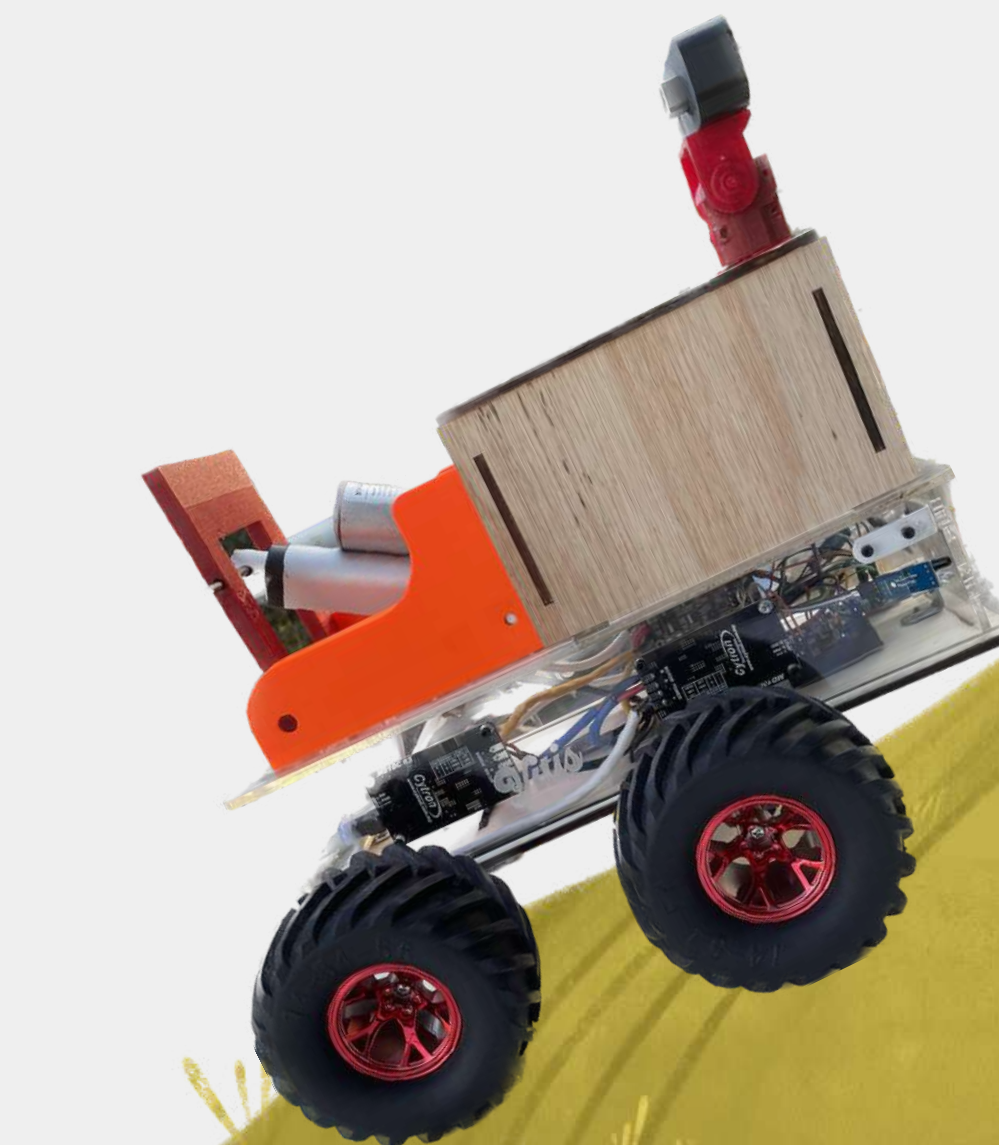
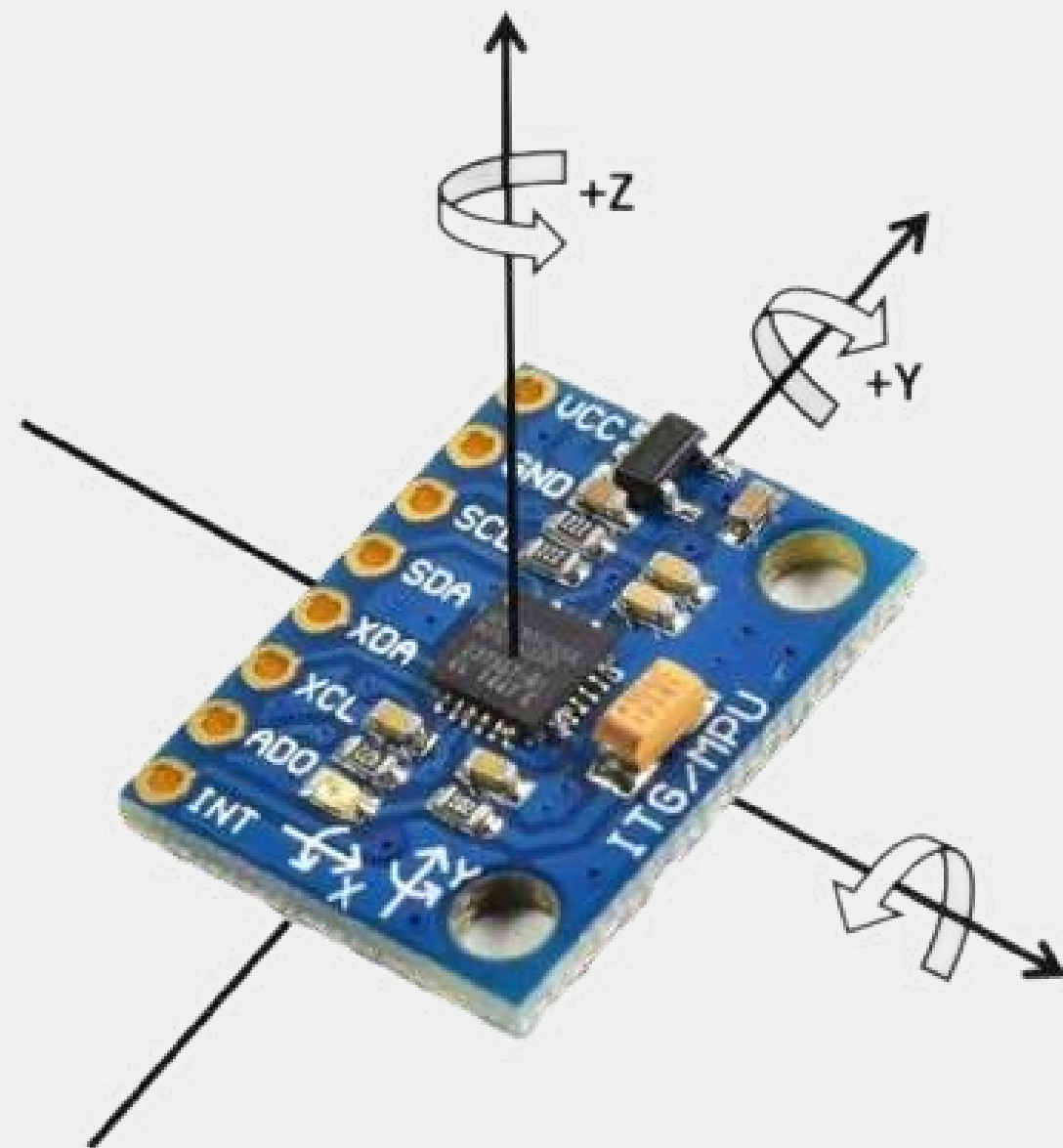


# WHAT IS VINS ?

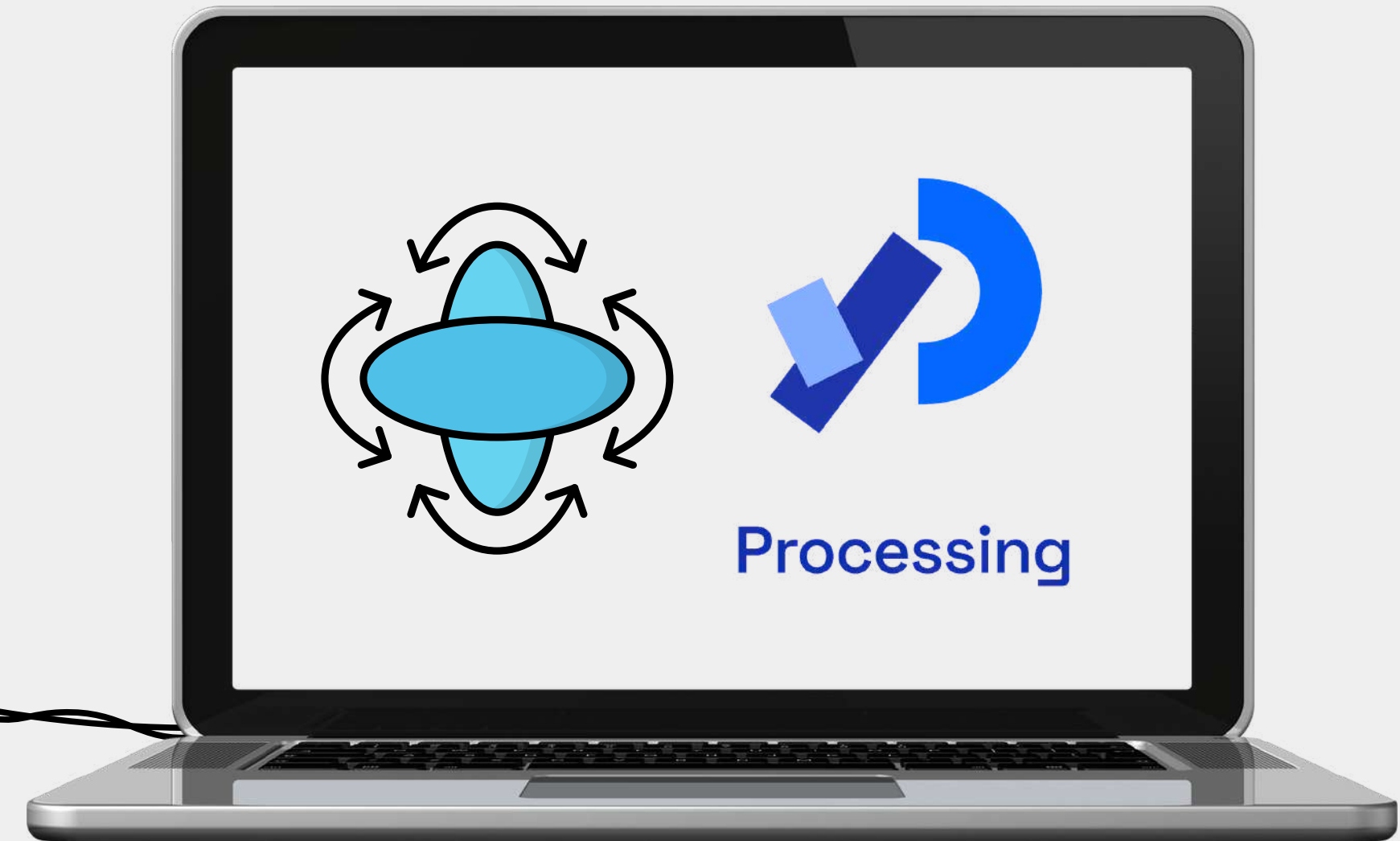
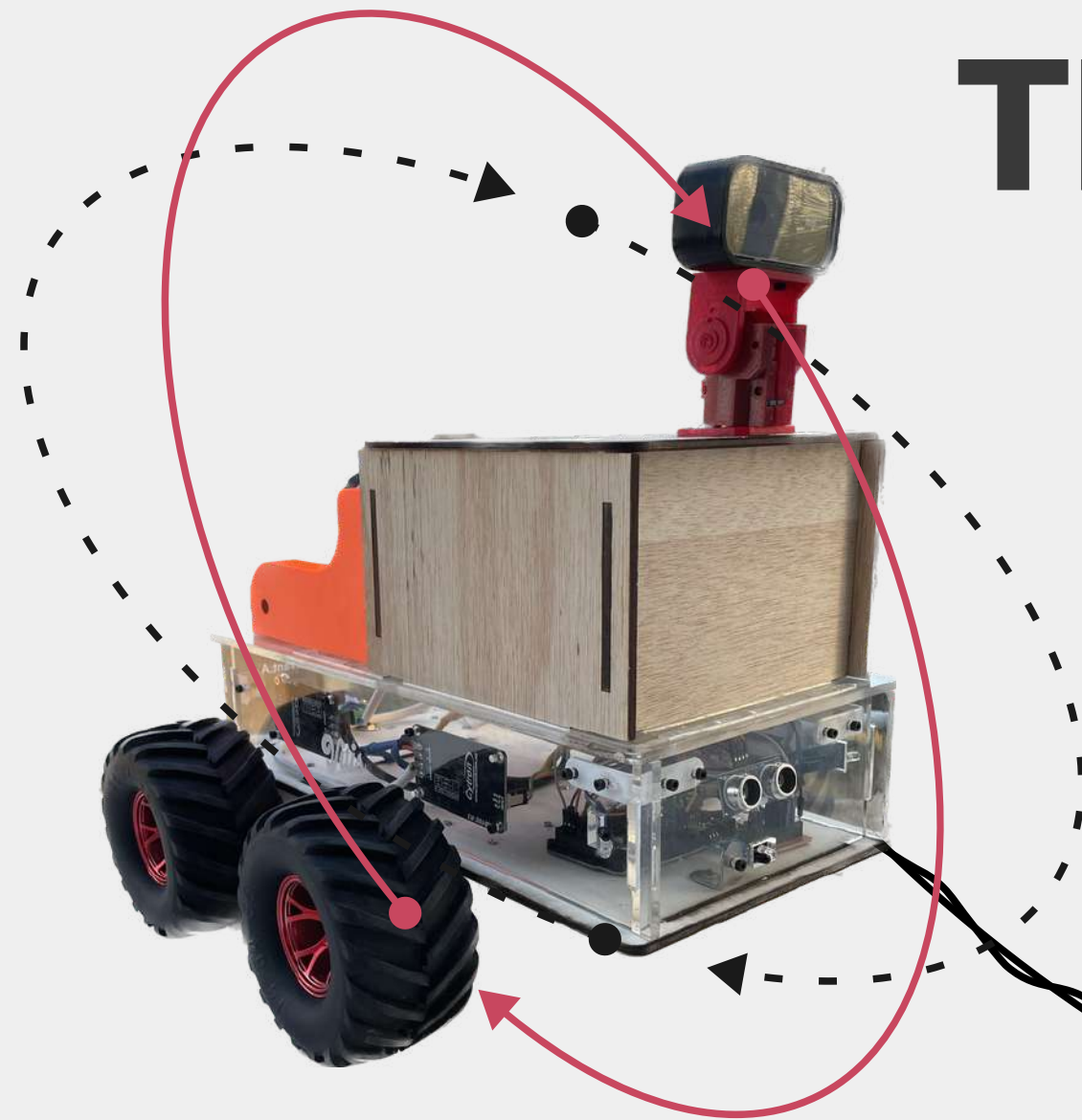




# WHY A MPU-6050 ?



# OUR OBJECTIVE FOR THIS COURSE



# ARDUINO PART



```
Arduino Uno
code_processing_acc.ino
1  #include <Wire.h>
2  #include <Adafruit_MPU6050.h>
3  #include <Adafruit_Sensor.h>
4
```

# ARDUINO PART



```
24 void loop() {  
25     unsigned long currentMillis = millis();  
26     if (currentMillis - lastMillis >= 10) { // Envoyer des données toutes les 10 ms  
27         lastMillis = currentMillis;
```

*Refresh every 10 ms*

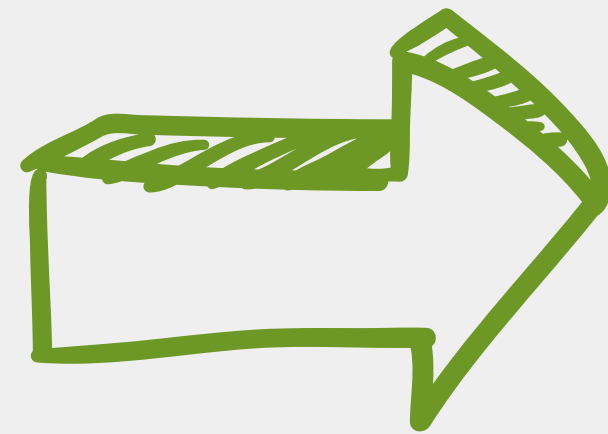
```
// Gyro data  
gyroX = g.gyro.x;  
gyroY = g.gyro.y;  
gyroZ = g.gyro.z;
```

```
// Acceleration data  
accelX = a.acceleration.x;  
accelY = a.acceleration.y;  
accelZ = a.acceleration.z;
```

# PROCESSING PART



Processing



```
1 import processing.serial.*;
2
3 Serial arduinoPort;
4 float angleX = 0.0;
5 float angleY = 0.0;
6 float angleZ = 0.0;
```

```
47 Serial.print(".");
Sortie Moniteur série X
```

---

Message (Enter to send message to 'Arduino Uno' on 'COM7')

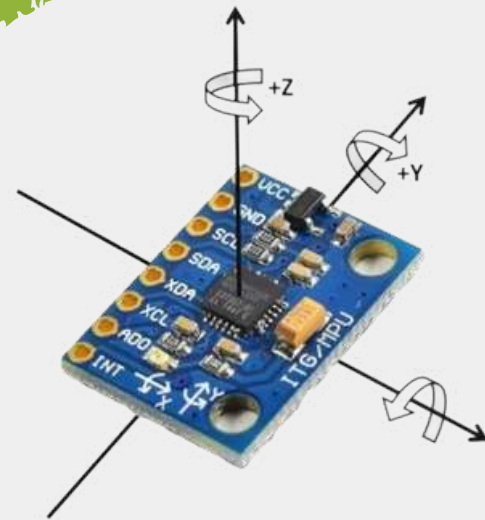
```
.16,0.01,-0.05,1.70,8.00,-6.13
0.17,0.01,0.07,1.88,8.08,-6.13
0.23,0.05,0.07,1.70,7.96,-6.35
```



# PROCESSING PART

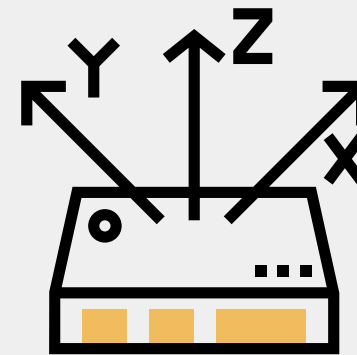


Processing



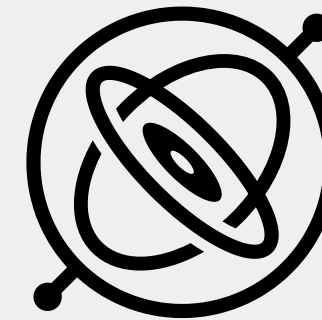
**MPU-6050**

=



**Accelerometer**

+



**Gyrometer**

*3 code suggestions :*

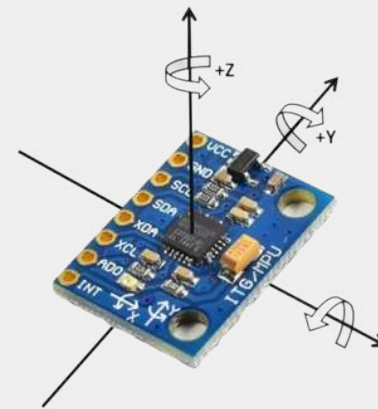
- **Only Accelerometer**
- **Only Gyrometer**
- **Accelerometer and Gyrometer**



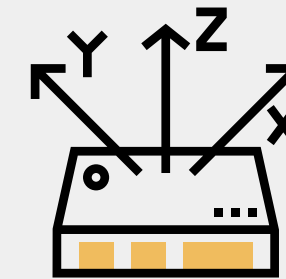
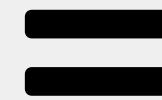
# PROCESSING PART



Processing



MPU-6050



Accelerometer

```
String[] values = split(data, ',');  
if (values.length == 6) {  
    float accelX = float(values[3]);  
    float accelY = float(values[4]);  
    float accelZ = float(values[5]);  
  
    anglesX = asin(accelX / g);  
    anglesY = atan(accelY / accelZ);  
}
```

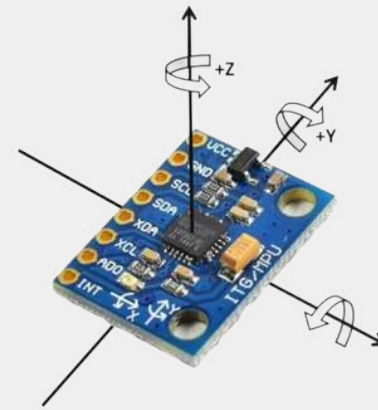


## DEMONSTRATION

# PROCESSING PART

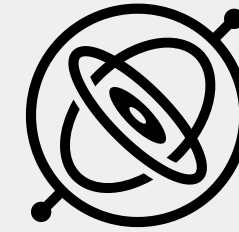


Processing



MPU-6050

=



Gyrometer

```
float gyroX = float(values[0]) - gyroXoffset;
float gyroY = float(values[1]) - gyroYoffset;
float gyroZ = float(values[2]) - gyroZoffset;

// Vérifier si les valeurs gyroscopiques sont supérieures au seuil
if (abs(gyroX) > gyroThreshold || abs(gyroY) > gyroThreshold || abs(gyroZ) > gyroThreshold) {
  // Ajustez la vitesse de rotation pour un mouvement plus rapide
  angleX += gyroX * 0.01;
  angleY += gyroY * 0.01;
  angleZ += gyroZ * 0.01;
}
```

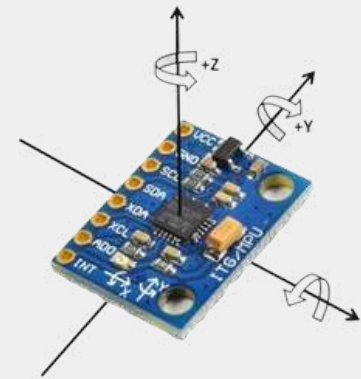
← speed

## DEMONSTRATION

# PROCESSING PART

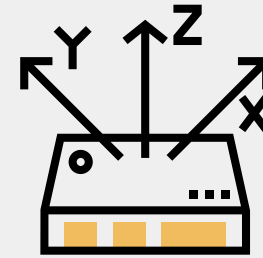


Processing



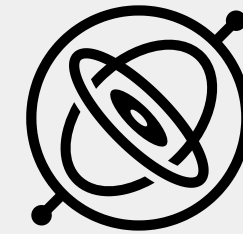
MPU-6050

=



Accelerometer

+



Gyrometer

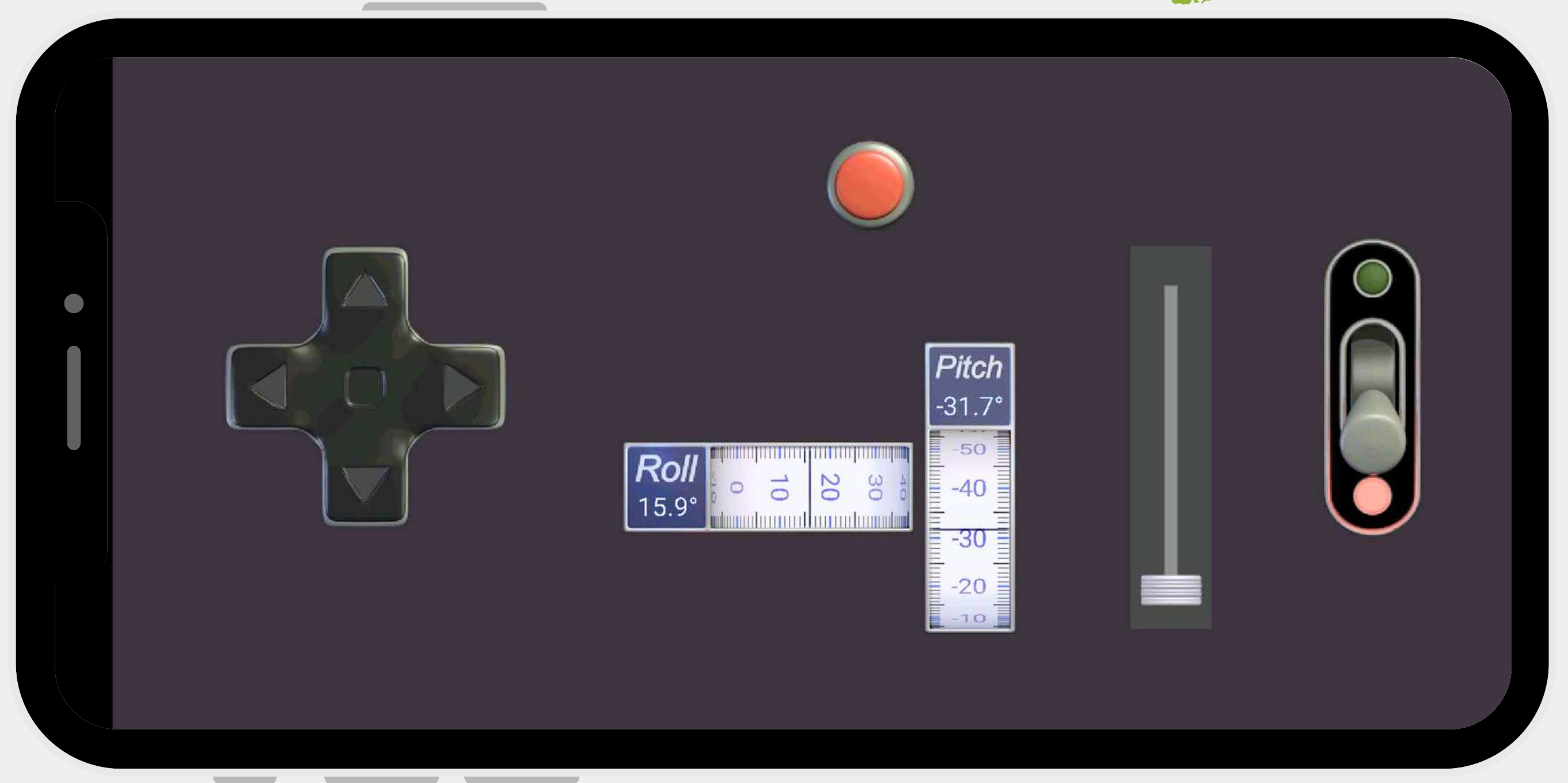
```
// Ajustez la vitesse de rotation pour un mouvement plus rapide
angleX += gyroX * 0.01;
angleY += gyroY * 0.01;
angleZ += gyroZ * 0.01;
}
// Faire la moyenne entre le gyro et l'accéléromètre
angleX = angleX/2 + asin(accelX / g);
angleY = angleY/2 + atan(accelY / accelZ);
```



## DEMONSTRATION



# BONUS !



# CONCLUSION



Processing

