Lesson 23 MPU-6050 Module

Introduction

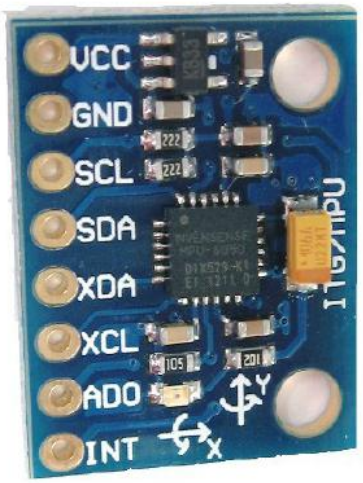
In this lesson, we will learn how to use the MPU6050 module which is one of the best IMU (Inertial Measurement Unit) sensors, compatible with Arduino.

Hardware Required

- ✓ 1 * RexQualis UNO R3
- ✓ 1 * MPU6050 module
- ✓ 4 * F-M Jumper Wires

Principle

MPU6050 SENSOR



The InvenSense MPU-6050 sensor contains a MEMS accelerometer and a MEMS gyro in a single chip. It is very accurate, as it contains 16-bits analog to digital conversion hardware for each channel. Therefore it captures the x, y, and z channel at the same time. The sensor uses the I2C-bus to interface with the Arduino.

The MPU-6050 is not expensive, especially given the fact that it combines both an accelerometer and a gyro.

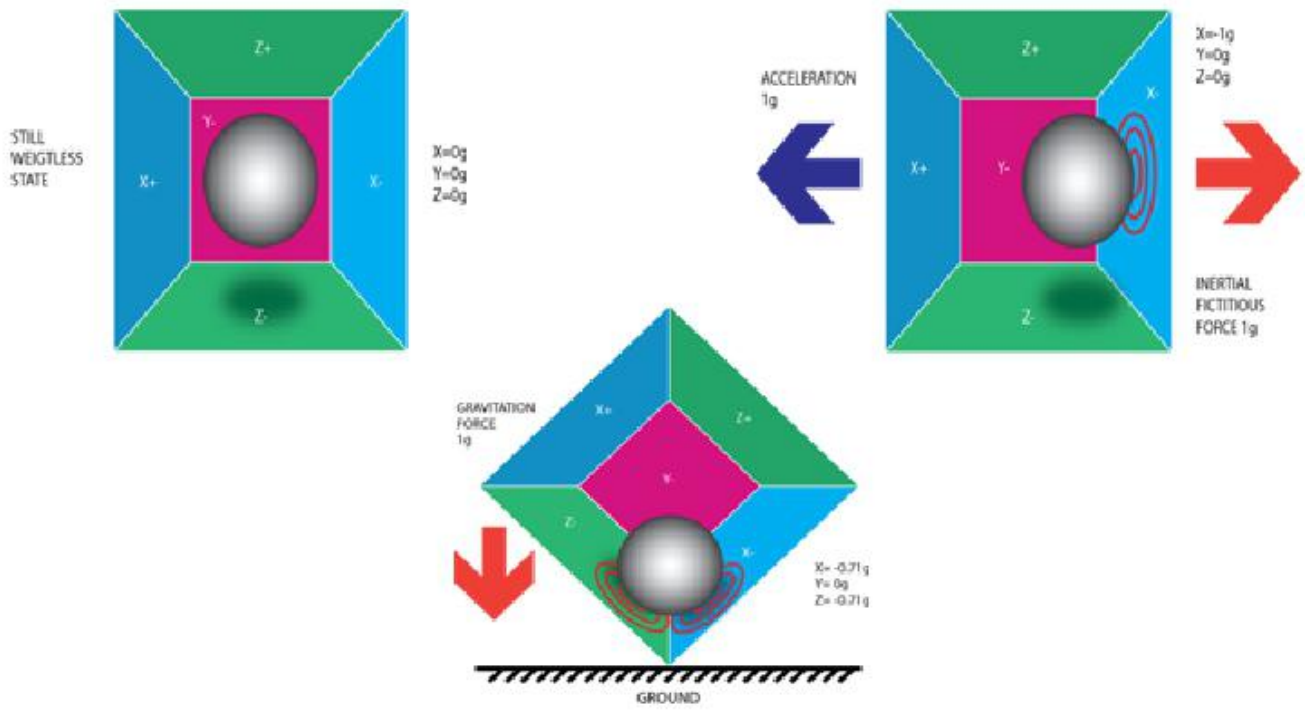
IMU sensors are one of the most inevitable type of sensors used today in all kinds of electronic gadgets. They are seen in smart phones, wearables, game controllers, etc. IMU sensors help us in getting the attitude of an object, attached to the sensor in three dimensional space. These values usually in angles, thus help us to determine its attitude. Thus, they are used in smart

phones to detect its orientation. And also in wearable gadgets like the nike fuel band or fit bit, which use IMU sensors to track movement.

How does it work?

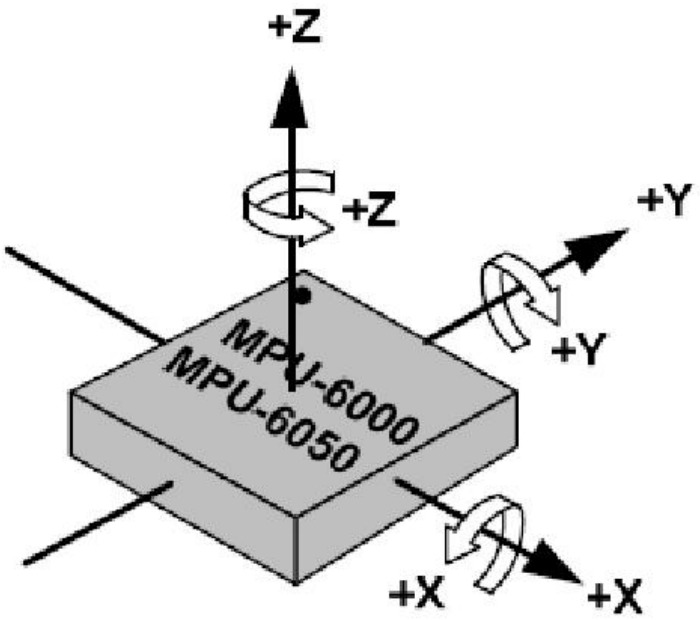
IMU sensors usually consists of two or more parts. Listing them by priority, they are : accelerometer, gyroscope, magnetometer and altimeter. The MPU 6050 is a 6 DOF (Degrees of Freedom) or a six axis IMU sensor, which means that it gives six values as output. Three values from the accelerometer and three from the gyroscope. The MPU 6050 is a sensor based on MEMS (Micro Electro Mechanical Systems) technology. Both the accelerometer and the gyroscope is embedded inside a single chip. This chip uses I2C (Inter Integrated Circuit) protocol for communication.

How does an accelerometer work?

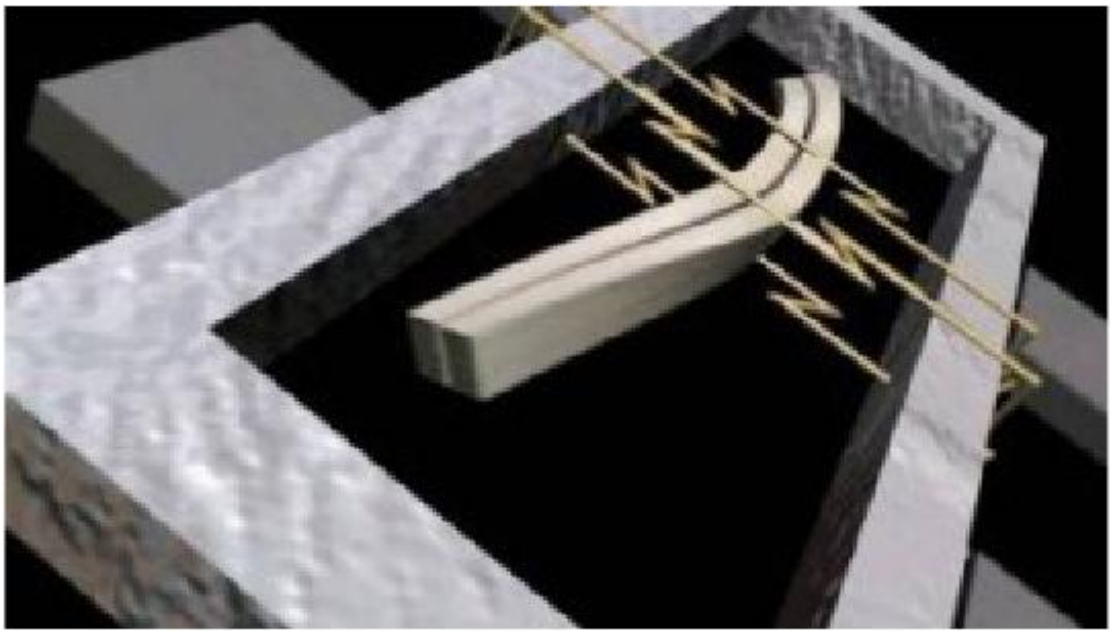


An accelerometer works on the principle of piezo electric effect. Here, imagine a cuboidal box, having a small ball inside it, like in the picture above. The walls of this box are made with piezo electric crystals. Whenever you tilt the box, the ball is forced to move in the direction of the inclination, due to gravity. The wall

with which the ball collides creates tiny piezoelectric currents. There are total, three pairs of opposite walls in a cuboid. Each pair corresponds to an axis in 3D space: X, Y and Z axes. Depending on the current produced from the piezoelectric walls, we can determine the direction of inclination and its magnitude. For more information check this.



How does a gyroscope work?



Gyroscopes work on the principle of Coriolis acceleration. Imagine that there is a fork like structure, which is in constant back and forth motion. It is held in

place using piezo electric crystals. Whenever, you try to tilt this arrangement, the crystals experience a force in the direction of inclination. This is caused as a result of the inertia of the moving fork. The crystals thus produce a current in consensus with the piezo electric effect, and this current is amplified. The values are then refined by the host microcontroller.

Code interpretation

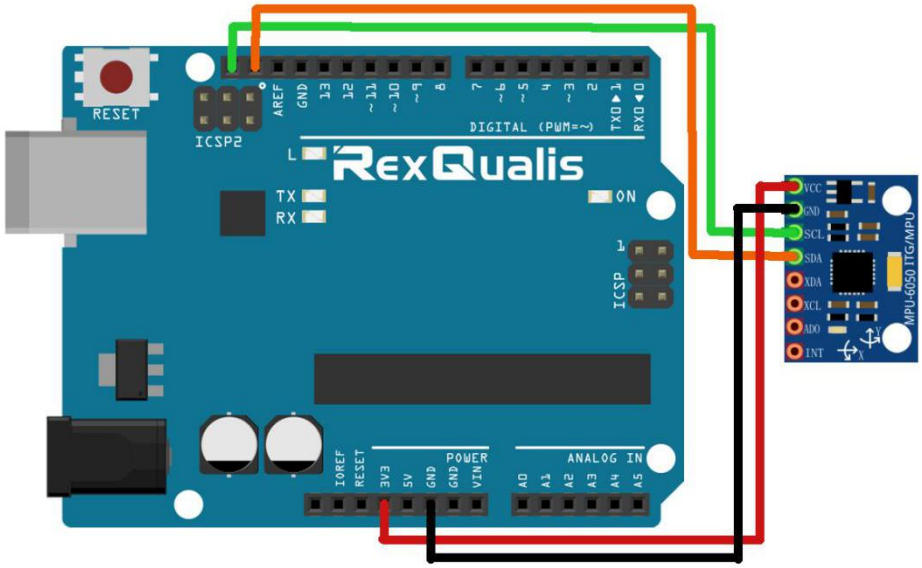
```
#include <MPU6050.h>

#include <Wire.h>

// MPU-6050 Short Example Sketch

const int MPU_addr=0x68;   // I2C address of the MPU-6050

int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

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
  AcX=Wire.read()<<8|Wire.read();  // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY=Wire.read()<<8|Wire.read();  // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ=Wire.read()<<8|Wire.read();  // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Tmp=Wire.read()<<8|Wire.read();  // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
  GyX=Wire.read()<<8|Wire.read();  // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
  GyY=Wire.read()<<8|Wire.read();  // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
  GyZ=Wire.read()<<8|Wire.read();  // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
  Serial.print("AcX = "); Serial.print(AcX);
  Serial.print(" | AcY = "); Serial.print(AcY);
  Serial.print(" | AcZ = "); Serial.print(AcZ);
  Serial.print(" | Tmp = "); Serial.print(Tmp/340.00+36.53);  //equation for temperature in degrees C from datasheet
  Serial.print(" | GyX = "); Serial.print(GyX);
  Serial.print(" | GyY = "); Serial.print(GyY);
  Serial.print(" | GyZ = "); Serial.println(GyZ);
```

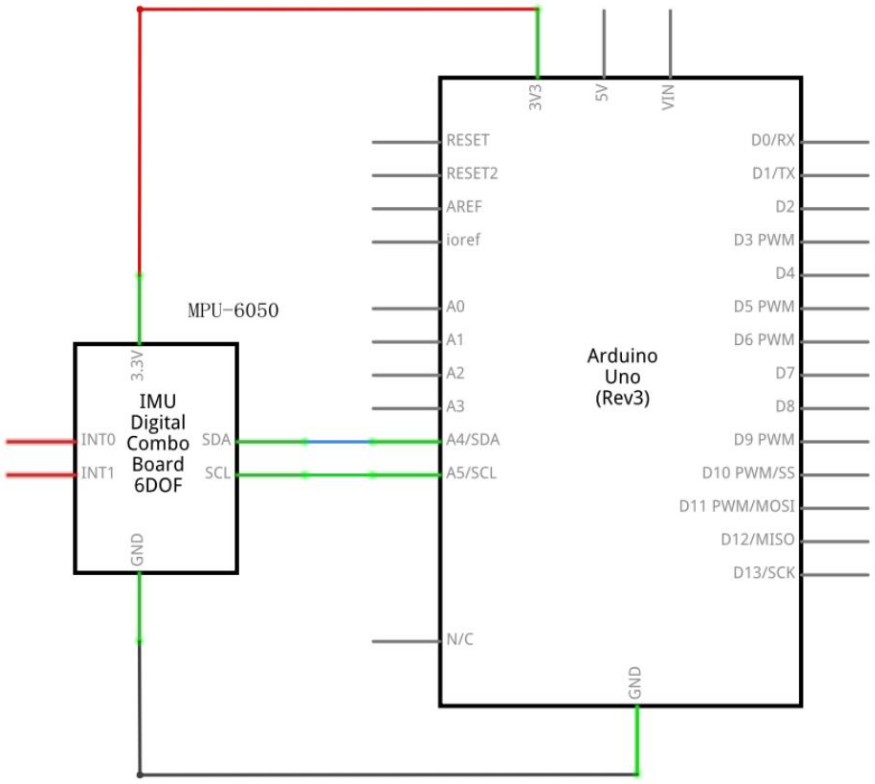
```
    delay(3000);

}
```

Experimental Procedures

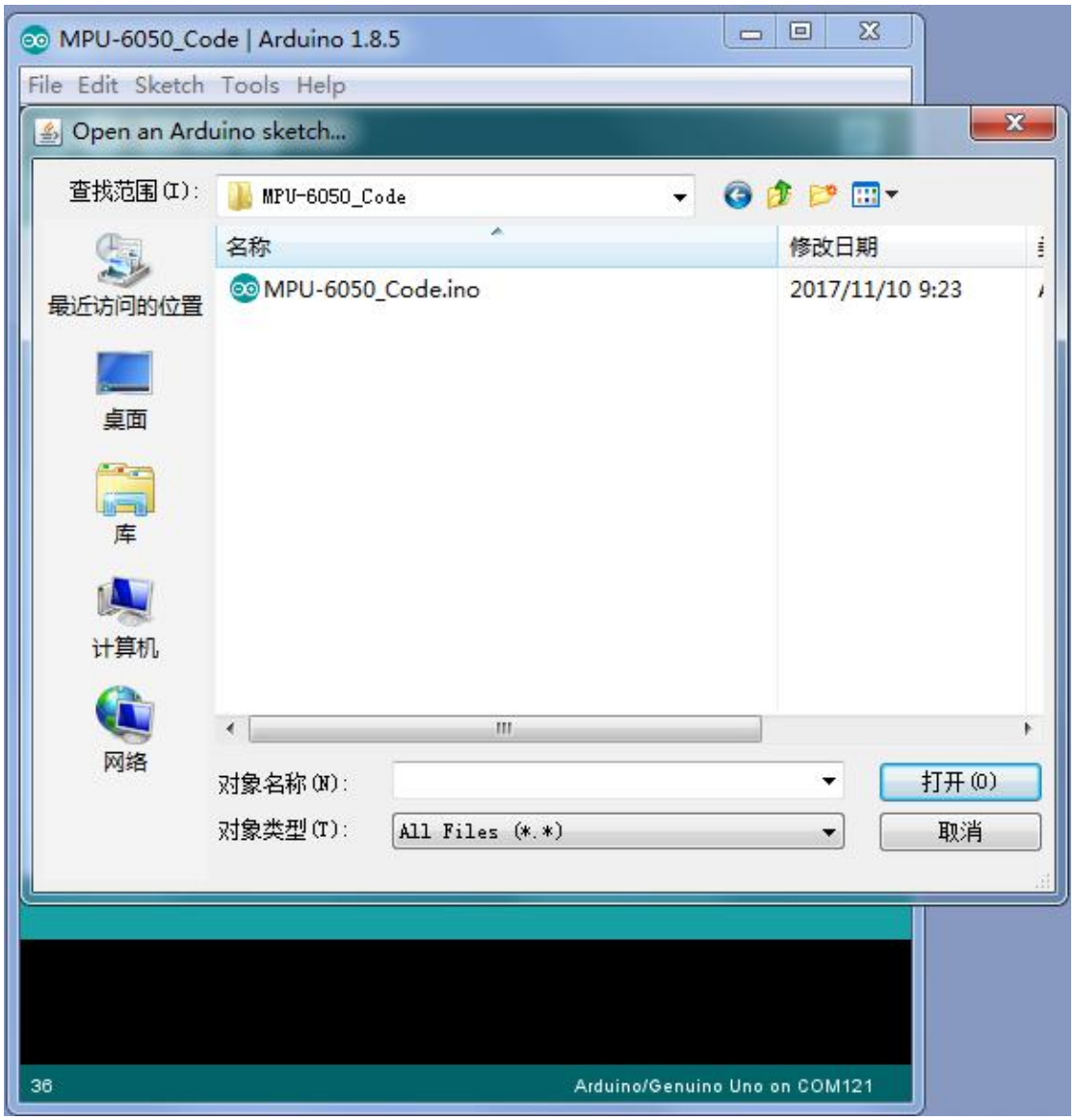
Step 1:Build the circuit



Schematic Diagram

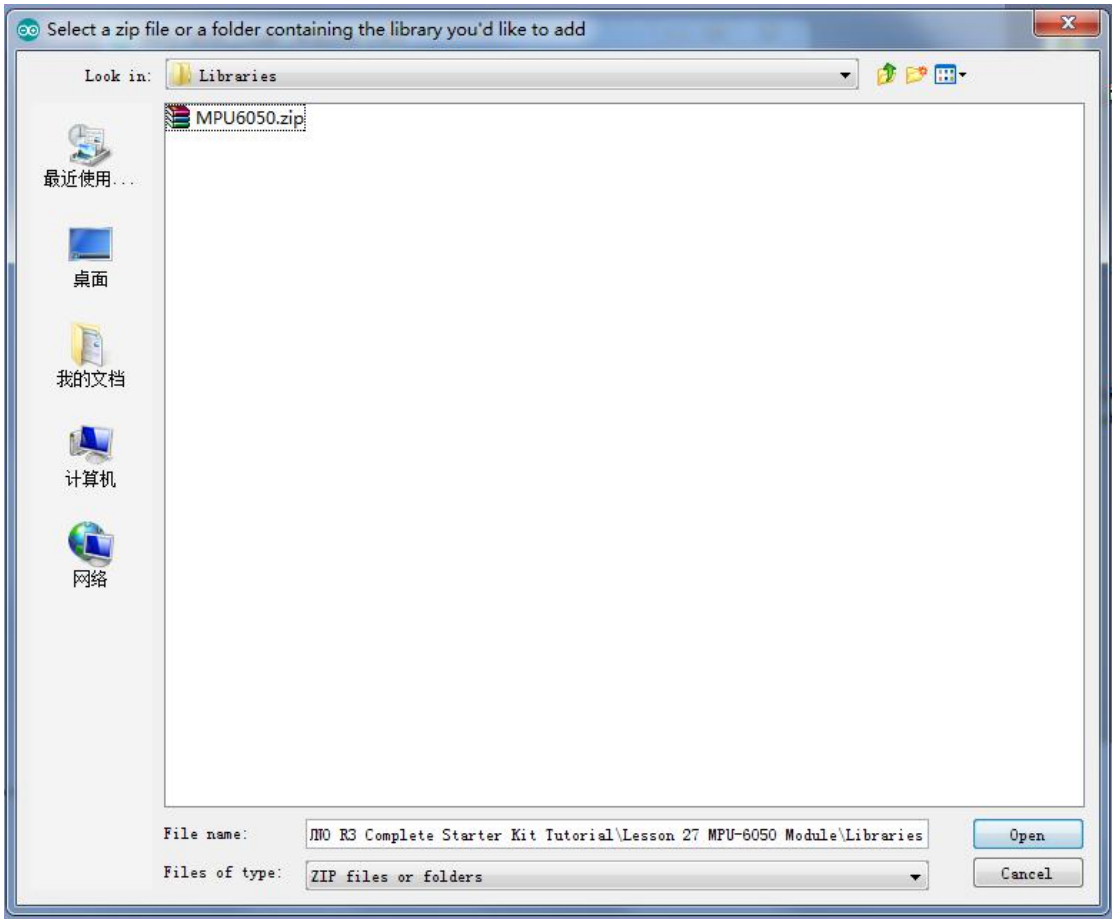


Step 2: Open the code:MPU-6050_Code



Step 3: Attach Arduino UNO R3 board to your computer via USB cable and check that the 'Board Type' and 'Serial Port' are set correctly.

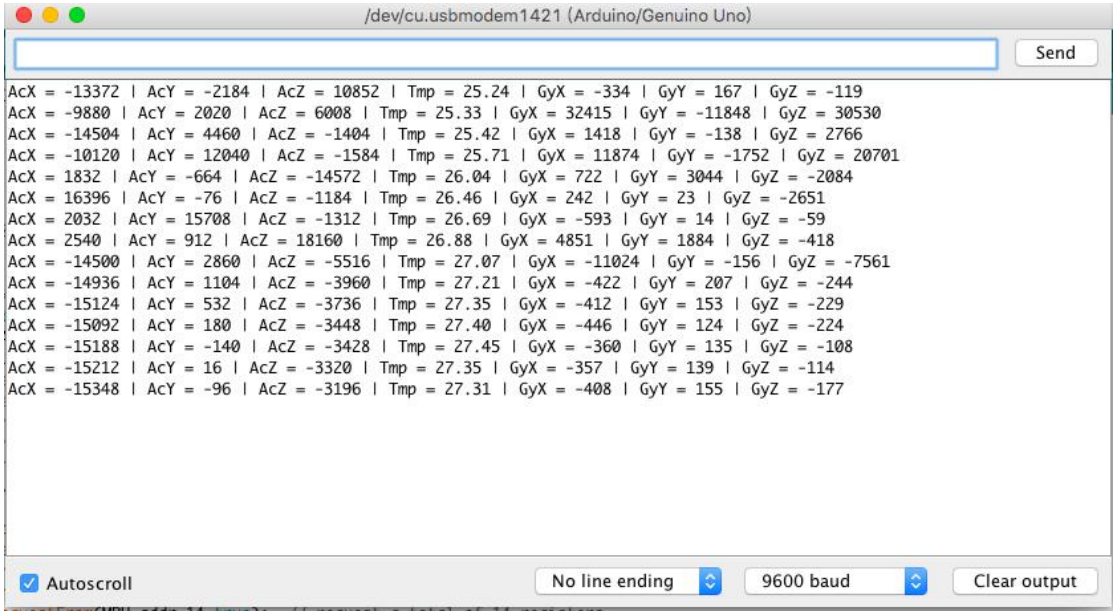
Step 4: Load the Library:MPU6050

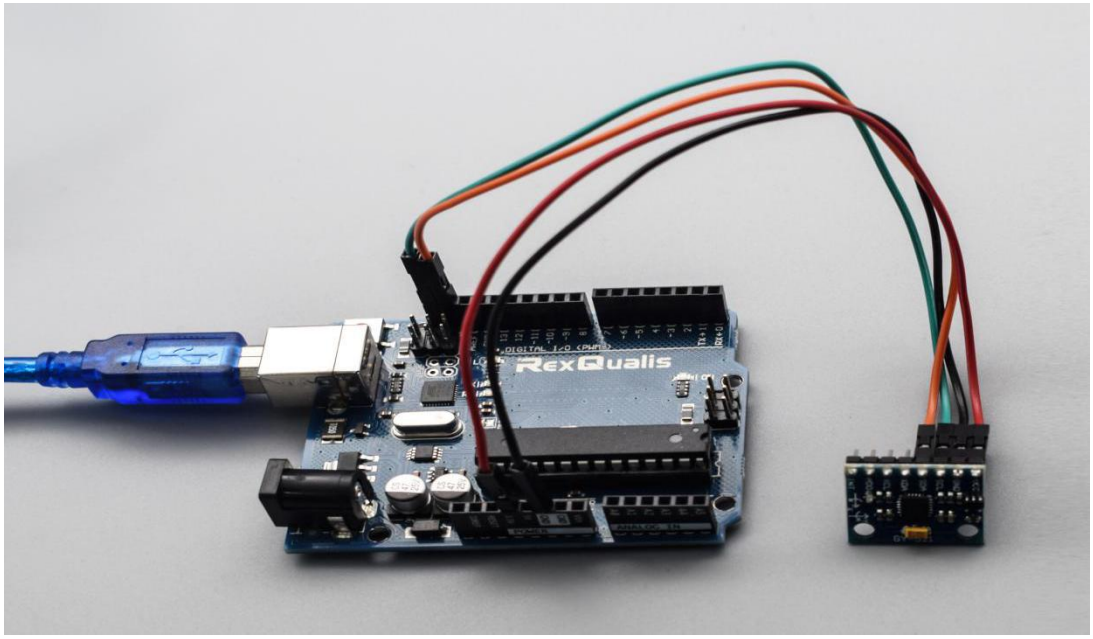


Step 5: Upload the code to the RexQualis UNO R3 board.

Step 6: Open the Serial Monitor,then you can see the data as below:

(How to use the Serial Monitor is introduced in details in Lesson 0 Preface)





If it isn’t working, make sure you have assembled the circuit correctly, verified and uploaded the code to your board. For how to upload the code and install the library, check Lesson 0 Preface.