Part 2

# lesson 15

## MPU-6050 Module

## Overview

In this lesson, we will learn how to use MPU-6050 module which is one of the best IMU (Inertia Measurement Unit) sensors, compatible with Arduino. IMU sensors like the MPU-6050 are used in self balancing robots, UAVs, smart phones, etc.

### Component Required:

(1) x Elegoo ESP32

(1) x MPU-6050 module

(4) x F-M wires

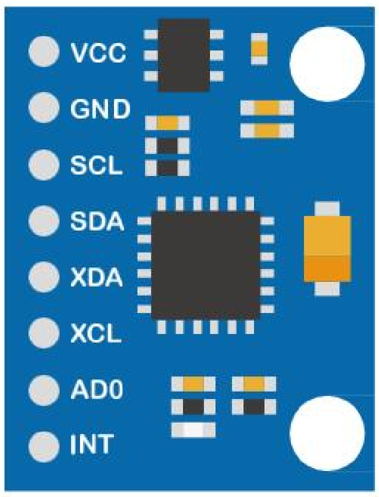
## Component Introduction

**MPU-6050 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.

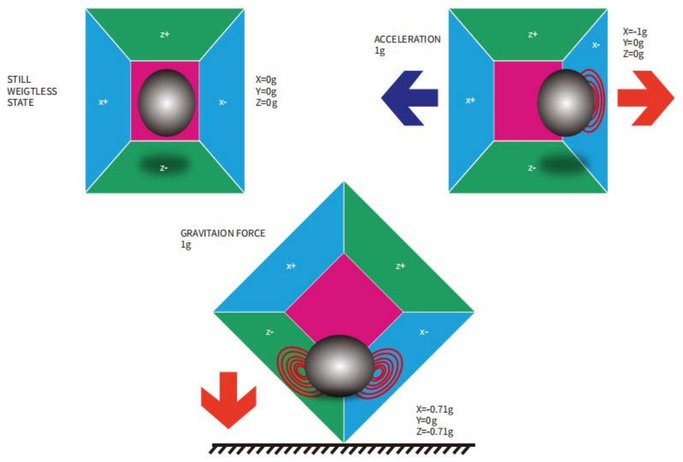


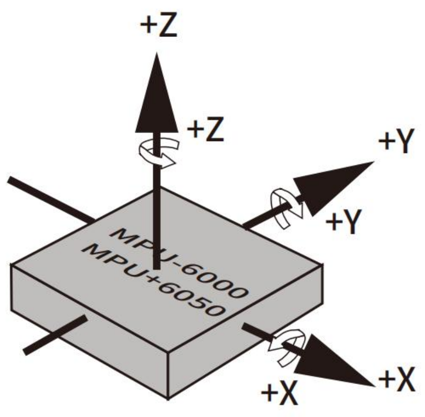
## How does it work?

**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 piezo electric currents. There are totally, 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 piezo electric walls, we can determine the direction of inclination and its magnitude. For more information check this.

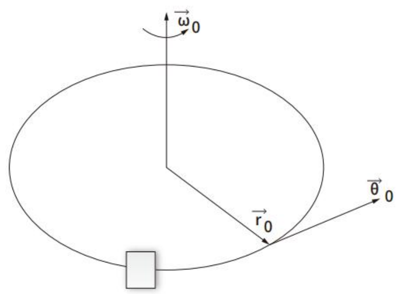


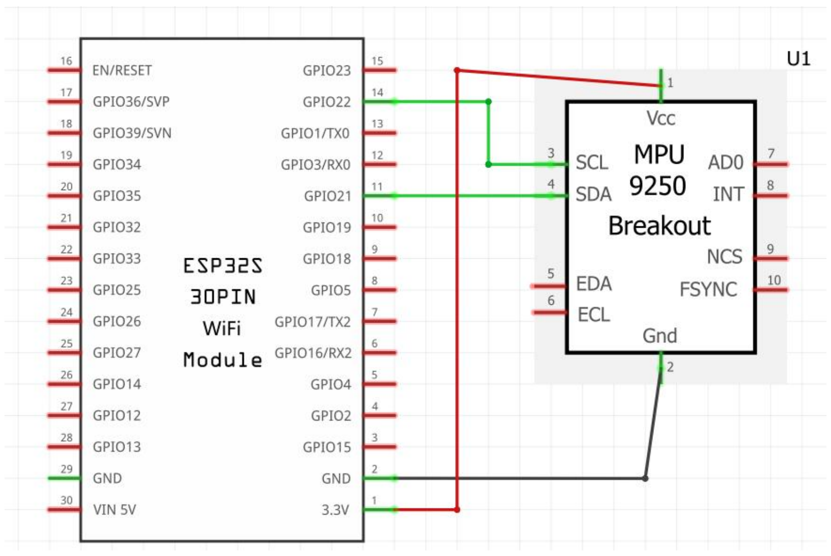


## How does a gyroscope work?

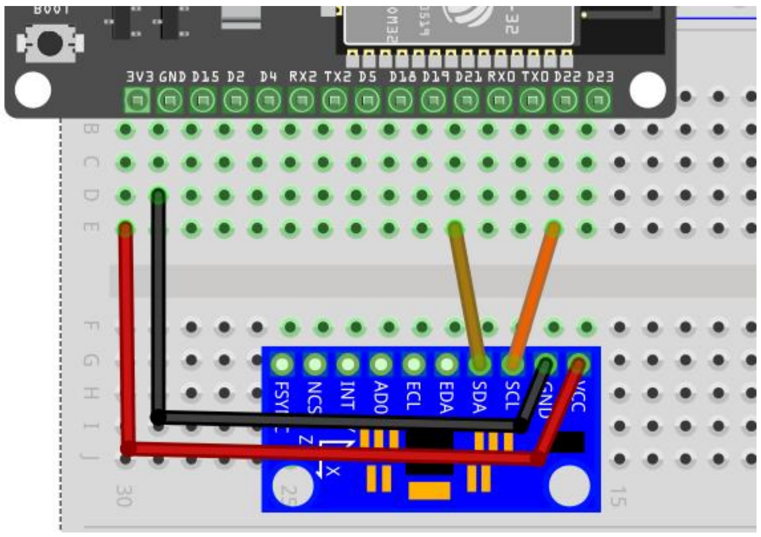
**Gyroscopes** work on the principle of Coriolis acceleration. Imagine that there is a fork- like structure, which is in constant back- and- forth motion. It is held in place using piezo electric crystals. Whenever you try to tilt this arrangement, the crystals experience a force in the direction of 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.

a (Coriolis) = -2vr * ω





**Connection Schematic**



BOOT
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23
FSYNC NCS INT AD0 ECL EDA SDA SCL GND VCC

Next, we need to set up the I2C lines. For this connect the pin labelled as SDA on the MPU-6050 to the ESP32's GPIO D21 (SDA). And the pin labelled as SCL on the MPU-6050 to the ESP32's GPIO D22(SCL). And that's it, you have finished wiring up the ESP32 to MPU-6050.

**Wiring diagram**

## Libraries needed

MPU-6050

## The Code

The short example sketch is a very short sketch and it shows all the raw values (accelerometer, gyro and temperature). It should work on Arduino Uno, Nano, ESP32,Leonardo, and also Due.

After wiring, please open the program in the code folder MPU 6050 and click UPLOAD to upload the program. See Lesson 5 in part 1 for details about program uploading if there are any errors.

Before you can run this, make sure that you have installed the < MPU 6050 > library or re-install it, if necessary. Otherwise, your code won't work.

For details about loading the library file, see Lesson 5 in part 1.

16-bit signed variables store raw data from gyroscope sensor.

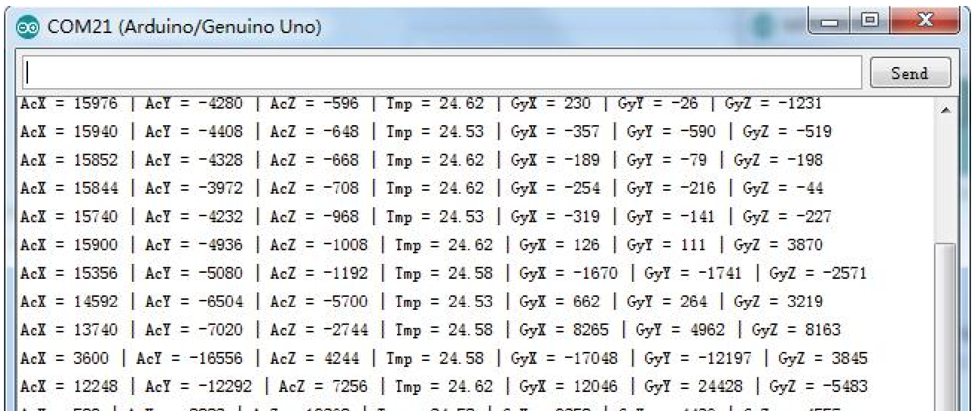
```
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
```

Sensor data is stored in consecutive registers starting from 0x3B; 14 bytes (7 16-bit values) are read in one I2C operation.

Data Merging: Each sensor's data occupies 2 bytes (high byte + low byte), which are combined into a 16-bit integer by shifting the high byte left by 8 bits using `<<8` and then performing a bitwise OR with the low byte using `|`.

```
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)
```

Open the monitor then you can see the data as blow:
Click the Serial Monitor button to turn on the serial monitor. The basics about the serial monitor are introduced in details in part 2 Lesson 4.



```
AcX = 15976 | AcY = -4280 | AcZ = -596 | Tmp = 24.62 | GyX = 230 | GyY = -26 | GyZ = -1231
AcX = 15940 | AcY = -4408 | AcZ = -648 | Tmp = 24.53 | GyX = -357 | GyY = -590 | GyZ = -519
AcX = 15852 | AcY = -4328 | AcZ = -668 | Tmp = 24.62 | GyX = -189 | GyY = -79 | GyZ = -198
AcX = 15844 | AcY = -3972 | AcZ = -708 | Tmp = 24.62 | GyX = -254 | GyY = -216 | GyZ = -44
AcX = 15740 | AcY = -4232 | AcZ = -968 | Tmp = 24.53 | GyX = -319 | GyY = -141 | GyZ = -227
AcX = 15900 | AcY = -4936 | AcZ = -1008 | Tmp = 24.62 | GyX = 126 | GyY = 111 | GyZ = 3870
AcX = 15356 | AcY = -5080 | AcZ = -1192 | Tmp = 24.58 | GyX = -1670 | GyY = -1741 | GyZ = -2571
AcX = 14592 | AcY = -6504 | AcZ = -5700 | Tmp = 24.53 | GyX = 662 | GyY = 264 | GyZ = 3219
AcX = 13740 | AcY = -7020 | AcZ = -2744 | Tmp = 24.58 | GyX = 8265 | GyY = 4962 | GyZ = 8163
AcX = 3600 | AcY = -16556 | AcZ = 4244 | Tmp = 24.58 | GyX = -17048 | GyY = -12197 | GyZ = 3845
AcX = 12248 | AcY = -12292 | AcZ = 7256 | Tmp = 24.62 | GyX = 12046 | GyY = 24428 | GyZ = -5483
```